PIBERT: A Physics-Informed Transformer with Hybrid Spectral Embeddings for PDE Modeling

Somyajit Chakraborty, Chen Xizhong[1]

Shanghai Jiao Tong University, Department of Chemistry and Chemical Engineering Shanghai, China

Abstract

We present **PIBERT**, a transformer surrogate that combines a hybrid Fourier–wavelet spectral encoder, physics-biased self-attention built from PDE residual diagnostics, and self-supervised pretraining via Masked Physics Prediction (MPP) and Equation Consistency Prediction (ECP). Evaluated on three CFDBench cases—cylinder wake, laminar tube, and lid-driven cavity—under a shared protocol, PIBERT trains faster and attains lower errors than strong physics-informed and operator-learning baselines. Ablations indicate that the hybrid spectral encoder improves fine-scale fidelity while the physics-biased attention stabilizes optimization and suppresses residual-heavy interactions; pretraining further improves data efficiency. Embedding probes show alignment with diagnostic quantities such as vorticity, supporting interpretability.

Keywords: Physics-Informed Deep Learning, Transformer, Multiscale PDEs, Computational Fluid Dynamics, Pretraining, Bi-directional encoders

1. Introduction

Partial differential equations (PDEs) lie at the heart of scientific computing, governing dynamic processes across disciplines including fluid mechanics, climate modeling, materials engineering, and chemical reaction systems. Many of these systems are characterized by multiscale phenomena, where solution behavior varies across widely separated spatial and temporal scales. For example, in subsurface transport, slow background flow coexists with sharp concentration fronts; in turbulence, coherent structures span orders of magnitude in frequency and scale. Traditional numerical solvers, while highly accurate, often require prohibitively fine meshes and small time steps to resolve such behavior [1]. As a result, solving multiscale PDEs with conventional methods can become computationally expensive, especially when parameter sweeps, uncertainty quantification, or real-time inference are needed.

To alleviate this burden, the field has increasingly turned to machine learning-based surrogates that aim to approximate PDE solutions with learned models. Among these, *physics-informed neural networks* (PINNs) have received significant attention [2]. By embedding the

Email addresses: chksomyajit@sjtu.edu.cn (Somyajit Chakraborty), chenxizh@sjtu.edu.cn (Chen Xizhong)

governing equations into the loss function, PINNs enforce physical laws without requiring large datasets, offering a mesh-free alternative to finite difference or finite element methods. However, while elegant in theory, vanilla PINNs face major practical challenges. Their basic architecture—a single multilayer perceptron (MLP) acting on coordinate inputs—often lacks the capacity to represent complex solution manifolds [3]. Training becomes especially fragile when addressing stiff systems, chaotic attractors, or nonlocal dependencies, due to gradient pathologies and ill-conditioned loss landscapes [4]. These issues are further compounded in multiscale settings, where the model must learn to represent both global structure and localized, high-frequency features from the same signal. Despite numerous enhancements (e.g., adaptive loss balancing [5], domain decomposition, causal training [6]), PINNs still struggle to scale to high-dimensional or highly nonlinear PDEs [7].

Parallel studies in *neural operator learning* have proposed an alternative formulation – rather than solving a specific PDE instance, neural operators learn mappings between function spaces, enabling generalization across a family of problems [8], [9], [10]. Models such as the Fourier Neural Operator (FNO) have demonstrated success in learning parametric solution operators, with applications in weather forecasting, fluid dynamics, and porous media flows [11]. By leveraging global spectral representations, these methods achieve zero-shot super-resolution and fast inference. Nonetheless, operator-based models are largely data-driven; they often require substantial training data and do not explicitly incorporate physics beyond solution input-output pairs [12]. Furthermore, their global representations may overlook localized effects or fine-scale structures unless augmented with specialized embeddings [13].

Against this backdrop, transformer architectures have emerged as a promising bridge between expressiveness and inductive bias. Originally developed for sequence modelling in natural language processing, transformers excel at capturing long-range dependencies through self-attention mechanisms [14]. In the context of PDE learning, this property makes them well-suited for modelling interactions across distant spatial or temporal locations—crucial in systems where far-field behavior is influenced by localised dynamics or boundary conditions. Early applications of transformers in physics-informed learning—such as PINNsFormer in 2023 [15] and more recently PITT [16] —have demonstrated improvements over MLP-based PINNs and even operator networks, particularly in time-dependent or chaotic regimes. Furthermore, recent advances in self-supervised masked pretraining for PDEs suggest that transformer-based models can learn general-purpose latent representations of physics, enabling transfer to new equations or domains [13].

Despite these promising developments, current transformer-based PDE frameworks often remain limited in scope. Most focus on a single enhancement—such as temporal attention or masked token prediction—without integrating multiple physics-informed components. Moreover, few address the multiscale nature of PDEs head-on by explicitly modeling both local and global structures in the input field. There remains a clear gap: how can we design a transformer-based model that (i) faithfully captures multiscale solution features, (ii) respects the structure of the governing PDEs, and (iii) learns generalizable physics priors without heavy reliance on labeled data?

In this work we introduce **PIBERT**, a Physics-Informed BERT-style Transformer, to address this challenge. Our core objective is to develop a unified framework that embeds multiscale physics knowledge directly into the architecture, training strategy, and inference pipeline of a transformer model. PIBERT is designed from the ground up to handle systems

with rich multiscale behavior and complex domain geometries, aiming to serve as a high-fidelity, generalizable surrogate for PDE simulations. To this end, our research is guided by the following questions:

- **RQ1:** Can a hybrid spectral representation (combining Fourier and Wavelet embeddings) improve the model's ability to capture both global structure and fine-scale local dynamics in PDE solutions?
- **RQ2:** How can we incorporate the geometry and operator structure of PDEs into the transformer's attention mechanism to bias it toward physically meaningful interactions?
- **RQ3:** Does self-supervised pretraining on physics-inspired tasks—such as masked point prediction and edge continuity prediction—enable more robust generalization, especially in data-scarce or extrapolative regimes?

Through these questions, we seek not only to improve predictive accuracy, but also to advance the interpretability, scalability, and trustworthiness of physics-informed deep learning. In what follows, we describe the architecture and training procedure of PIBERT, benchmark its performance on diverse PDE tasks, and analyze its ability to generalize across regimes and scales. All empirical results in this paper utilise 3D cases from CFDBench (cylinder wake, laminar tube, lid-driven cavity, and dam) [17]; we do not claim performance beyond these settings.

We next review recent advances in physics-informed learning and operator surrogates to situate our contribution (Section 2). We then introduce PIBERT—its hybrid Fourier–wavelet encoder, physics-biased attention, and MPP/ECP pretraining—and summarize the supporting mathematical analysis (Sections Appendix A and 3). Datasets, training protocol, and reproducibility details follow in Sections Appendix C and 4. We present benchmarks on CFDBench and a 1D reaction test, including interpolation studies and ablations, in Section 5, and close with implications and limitations in Section 6 and a brief conclusion in Section 7; implementation notes and PDE setups appear in Sections Appendix B and Appendix D.

2. Related Works

Over the last decade, the integration of deep learning with physical modeling has become a transformative approach in scientific computing, particularly for solving complex partial differential equations (PDEs). This integration has sparked the development of a wide array of physics-informed machine learning (PIML) techniques, which have evolved in parallel with advancements in deep learning architectures, particularly neural networks, transformers, and self-supervised learning. In this section, we explore the key recent developments (2023–2025) in the field, emphasizing the challenges and innovations that have led to the creation of frameworks like PIBERT.

2.1. Physics-Informed Neural Networks (PINNs) and Early Challenges

PINNs, introduced by Raissi et al. [18] in 2019, were a groundbreaking development that integrated physics constraints directly into the loss function of neural networks to solve PDEs. These networks utilize the physics of the problem (e.g., conservation laws, boundary

conditions) to inform the training process. Despite their early success, recent reviews by Raissi et al. [2] and Abbasi et al. [7] highlighted several limitations of PINNs, including difficulty handling stiff equations, poor performance with shock capturing, and struggles with multiscale phenomena. These issues are partly due to the point-wise evaluation of PINNs, which make it challenging to model long-range dependencies in spatial and temporal domains. PINNs also fail to leverage the full spectrum of physical symmetries inherent in the problems they aim to solve.

2.2. Neural Operators for Parametric PDEs

A promising advancement beyond PINNs is the development of *neural operators*, which aim to improve generalization across a wide range of physical configurations. An early milestone in operator learning is *DeepONet*, which models mappings between function spaces with a branch–trunk architecture. This provides the model practical evidence for the universal approximation of nonlinear operators from sparse sensor measurements [19]. This operator-centric viewpoint set the stage for later neural-operator designs. The Fourier Neural Operator (FNO), introduced by Li et al. [20], marked a paradigm shift by learning mappings between function spaces instead of point-wise solutions, thus offering better generalization across different boundary conditions and physical parameters. FNO and its variants have seen significant enhancements, such as the U-FNO by Wen et al. [21], which incorporated multiphase flow problems, and physics-embedded FNOs developed by Xu et al. [22] that integrate physics constraints directly within the Fourier layers. These developments demonstrate improved flexibility and efficiency in solving parametric PDEs.

The move to wavelet-based methods has also been significant in dealing with localized features and discontinuities. *Deep Wavelet Neural Networks* (DWNNS) introduced by Li et al. [23] leverage wavelet bases for the solution of PDEs with sharp discontinuities. More recent advancements by Su et al. [24] integrated *multiscale attention wavelet operators*, which proved effective in biochemical systems with steep gradients, thus broadening the scope of spectral methods in PIML. However, the trade-off between global pattern recognition (via Fourier-based methods) and local feature extraction (via wavelets) remains an ongoing challenge. Hybrid models that can balance these two domains are now a key area of research.

Recent advances have further integrated spectral representations into deep learning frameworks for PDEs. FourierFlow addresses spectral bias in fluid dynamics through a generative framework that incorporates frequency-aware weighting and surrogate feature alignment, demonstrating improved performance in turbulence modeling [25]. While primarily designed for generative forecasting, its architecture emphasizes explicit control over frequency components—a direction complementary to our hybrid spectral embedding approach. Similarly, WaveDiff leverages wavelet transforms within a diffusion-based framework to enable high-fidelity super-resolution of PDE solutions, exploiting the multi-scale localization properties of wavelets for enhanced detail recovery [26]. These works highlight the growing importance of incorporating domain-specific signal priors—such as scale separation and frequency structure—into neural solvers. In contrast to these methods, PIBERT unifies both Fourier and wavelet representations within a single transformer architecture, enabling simultaneous global and local field modeling, while enforcing physical consistency through physics-informed attention and self-supervised pretraining.

2.3. Transformers in Scientific Computing

Transformers, originally developed for NLP tasks, have increasingly been adapted for scientific computing, particularly for solving PDEs and modeling long-range dependencies in physical systems. Recent work by Lorsung et al. [16] introduced the *Physics-Informed Token Transformer (PITT)*, which applies self-attention mechanisms to PDE solution fields. This architecture is specifically designed to capture spatiotemporal dependencies across large datasets, demonstrating significant improvements in modeling long-range correlations in systems such as fluid dynamics and heat transfer. However, PITT and similar approaches have encountered computational challenges in scaling to high-resolution problems, with attention complexity growing quadratically.

The work of Luo et al. [14] further refined this by introducing *physics-aware attention*, which modulates attention weights to respect the physical symmetries of the underlying system. This method ensures that the model better adheres to principles such as conservation laws and energy balance, thus making the model more physically interpretable and reliable. Additionally, Yang et al. [27] explored combining autoencoders with attention mechanisms, demonstrating that enforcing physical priors (such as known dynamics or boundary conditions) within transformer architectures can substantially improve generalization in high-dimensional physical systems.

One critical insight from these works is that while transformers are highly effective in capturing long-range dependencies, they often fail to respect the inherent physical structures of scientific problems unless explicitly designed to do so. Recent studies have advocated for *physics-constrained self-attention mechanisms*, which enhance the performance of transformers in modeling physical systems by introducing domain-specific inductive biases.

2.4. Self-Supervised Learning for Physical Systems

The application of *self-supervised learning* (SSL) in physical systems is an area that has rapidly gained attention, particularly for leveraging unlabeled simulation data. In 2023, Berend et al. [28] demonstrated that *masked latent semantic modeling*, a technique inspired by SSL in NLP, could be adapted to pre-train models for physical systems. This work marked a significant shift toward unsupervised learning approaches in PIML, offering the potential to leverage abundant unlabeled data from simulations and experiments.

In 2025, Garnier et al. [29] proposed the *Mesh-Mask* framework, which integrates masked graph neural networks (GNNs) for physics-based simulations. This approach was particularly effective for incomplete observational data, allowing models to learn physical consistency directly from the structure of the data. By learning the underlying patterns of physical systems without relying on labeled training data, this method improves robustness, especially when dealing with sparse or noisy datasets.

The development of masked prediction models in physical systems represents a crucial step forward, as it enables the model to generate physically consistent outputs even when labeled data is limited or unavailable. These innovations point to a new era of self-supervised pretraining for scientific machine learning, which has the potential to dramatically reduce the dependency on labeled datasets, making large-scale simulations more efficient and accessible.

2.5. Benchmarking and Evaluation Frameworks

As the PIML field matures, standardized benchmarking and evaluation frameworks have become essential for comparing different models. The CFDBench benchmark, introduced by Luo et al. [17], is a significant contribution to the community, providing a comprehensive testbed for evaluating the spatiotemporal generalization of machine learning models in fluid dynamics. CFDBench offers standardized test cases across different physical regimes, enabling systematic comparison of model performance on challenging tasks, such as turbulence modeling and multi-phase flows.

Building on this, Wang et al. [13] conducted a systematic review of machine learning techniques in computational fluid dynamics (CFD), providing protocols for evaluating both the accuracy and computational efficiency of models. These benchmarking efforts have underscored several persistent challenges, such as ensuring stability in long-term predictions and maintaining generalization across different physical regimes, issues that continue to drive research into more robust and scalable modeling techniques.

2.6. Bridging Current Gaps

Despite these advances, several critical gaps remain in the literature. First, while methods like FNO and PITT have demonstrated effectiveness in capturing global structures or long-range dependencies, they often fail to adequately capture localized features or sharp discontinuities. Second, while self-supervised learning holds promise for enhancing model robustness, few methods have been developed specifically for the unique challenges of physical systems, such as maintaining physical consistency in learned representations.

PIBERT, introduced in this work, addresses these gaps by combining Fourier and wavelet embeddings to capture both global smooth structures and sharp localized features. In addition, PIBERT incorporates physics-constrained attention mechanisms, which bias interactions toward physically meaningful patterns, ensuring that the model respects the symmetries and conservation laws inherent in the physical system. Furthermore, PIBERT's novel self-supervised pretraining objectives, such as Masked Physics Prediction and Equation Consistency Prediction, are specifically designed to address the challenges of physics-informed learning, providing a framework that is both scalable and robust.

Recent evaluations of PIBERT on CFDBench and Navier-Stokes cylinder wake datasets have shown its ability to achieve stong generalization and physical consistency. These results position PIBERT as a promising direction for the next generation of physics-informed machine learning frameworks.

3. PIBERT

3.1. Mathematical Foundations of BERT

BERT introduced in 2019, stands for Bidirectional Encoder Representations from Transformers, is a language representation model that uses deep bidirectional self-attention to capture contextual relationships between tokens [30],[31]. Unlike unidirectional models, BERT employs a masked language model objective whereby a portion of the tokens in the input are randomly masked and the model is trained to predict the original token from its surrounding

context. The architecture is based on the Transformer encoder and relies on the self-attention mechanism that is mathematically defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V},$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices, respectively, and d_k is the dimensionality of the keys. This formulation enables the model to consider both left and right context simultaneously. In addition, BERT uses a next sentence prediction task that further enhances its ability to capture relationships between sentence pairs, making it suitable for a wide range of natural language processing tasks. The overall design, which unifies pre-training and fine-tuning within the same architecture, has been critical in setting new benchmarks in language understanding.

3.2. PIBERT Architecture

The PIBERT architecture extends the conventional transformer-based BERT framework by incorporating physics-informed embeddings, constraints, and attention mechanisms. Unlike natural language processing models, where positional encoding is used to capture sequence order, PIBERT integrates domain knowledge directly into its embedding space, attention mechanism, and loss function to ensure that the learned representations adhere to fundamental physical principles. Figure 1 shows the detailed architectural overview of PIBERT. The following sections outline the core components of PIBERT, detailing the mathematical formulations that underpin its architecture.

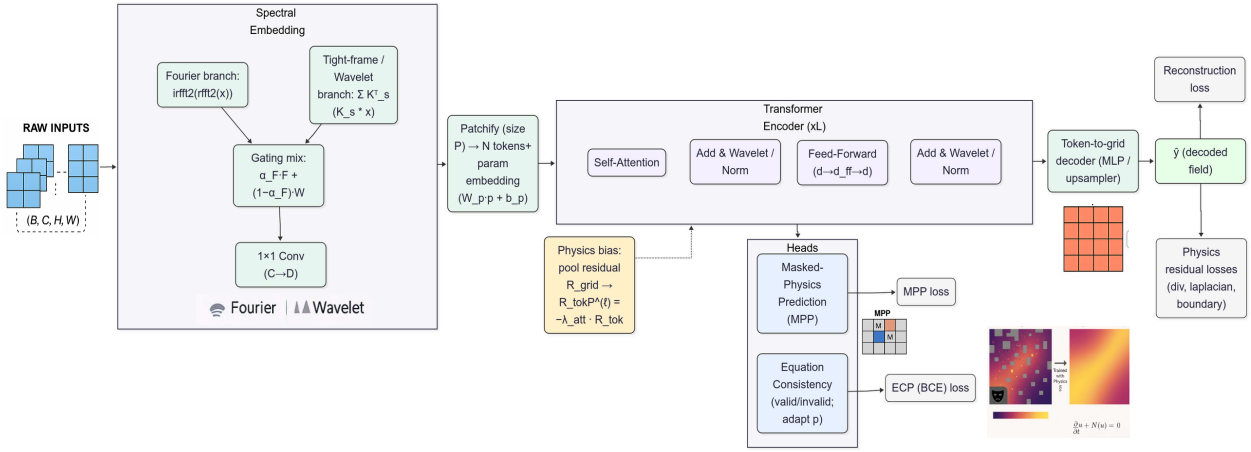


Figure 1: PIBERT architecture and training objectives. Raw CFD fields and PDE parameters are mapped from grid to tokens via Fourier + Wavelet embeddings, then processed by a transformer encoder (×N) with self-attention and Wavelet → Feed-Forward blocks. Two heads branch from the encoder—Masked-Physics Prediction (MPP) and Equation-Consistency (valid/invalid PDE with parameter adaptation)—while a token-to-grid module decodes tokens back to the field to produce haty and diagnostics. Supervision combines reconstruction and physics-residual losses.

3.3. Physics-Informed Hybrid Spectral Embeddings

We embed grid fields with a *hybrid Fourier–wavelet encoder* that provides global spectral context and local feature sensitivity.

Table 1: Core symbols used in Sec. 3.3. Units in parentheses.

Symbol	Meaning (units)
Ω	Spatial domain (–)
$(x, y),\ t$	Spatial coordinates, time ($[L]$, $[T]$)
$H,\ W$	Grid height, width (px)
P	Patch size (px)
$N = HW/P^2$	Number of tokens (–)
$u, v,\ \mathbf{u} = (u, v)$	Velocity components / vector ($[L/T]$)
p	Pressure ($[M/(L\,T^2)]$)
$\rho,\ \nu$	Density ($[M/L^3]$), kinematic viscosity ($[L^2/T]$)
$\omega = \partial_x v - \partial_y u$	Vorticity ($[1/T]$)
$\nabla,\ \Delta$	Gradient; Laplacian (–)
α_F	Fourier–wavelet fusion gate (–)
λ_{att}	Attention-bias strength (–)
$Q,\ K,\ V$	Attention queries, keys, values (–)
$L_{ij},\ \alpha_{ij}$	Attention logits; softmax weights (–)
$\mathcal{L}_{\text{recon}},\ \mathcal{L}_{\text{phys}}$	Data loss; physics penalties (–)
$\mathcal{F},\ \mathcal{F}^{-1}$	Discrete Fourier transform and inverse (–)
rfft2, irfft2	Real 2-D FFT and inverse (–)

3.3.1. Fourier branch (per-frequency mixing)

Given $x \in \mathbb{R}^{B\times C\times H\times W}$, define $X = \text{rfft2}(x)$ and apply a per-frequency channel mix on the kept half-plane; inverse rFFT returns $y_{\text{ft}} \in \mathbb{R}^{B\times D\times H\times W}$. When the per-frequency mixing is column-unitary on the kept band, the map is nonexpansive (an isometry on band-limited inputs).

Proposition 3.1 (Energy preservation of the Fourier branch)**.** *If the per-frequency mixing matrices satisfy $W(h,w)^\top W(h,w) = I$ on the retained modes and non-kept modes are zeroed, then the Fourier branch is 1-Lipschitz in ℓ_2; if inputs are band-limited to the retained set, it is an isometry.*

3.3.2. Tight-frame branch (undecimated local filters)

We use four translation-invariant filters $\{K_{LL}, K_{LH}, K_{HL}, K_{HH}\}$ whose discrete Fourier responses form a partition of unity on the torus, yielding exact energy partition and perfect reconstruction (Parseval frame).

Proposition 3.2 (Parseval tight frame, exact energy partition)**.** *For all x, $\sum_s \|K_s * x\|_2^2 = \|x\|_2^2$ and $x = \sum_s K_s^\vee * (K_s * x)$; analysis and synthesis are 1-Lipschitz.*

3.3.3. Hybrid fusion by a scalar softmax gate

Let $E = \alpha_F\, y_{\text{ft}} + (1 - \alpha_F)\, y_{\text{tf}}$ where $\alpha_F = \exp(\gamma_F)/(\exp(\gamma_F) + \exp(\gamma_W))$.

Lemma 3.3 (Nonexpansive hybrid)**.** *If each branch is 1-Lipschitz, then for any (possibly spatially varying) $\alpha_F \in [0, 1]$ the hybrid E is 1-Lipschitz. If inputs are band-limited and the frame branch reduces to identity, the hybrid is an isometry.*

Proof sketches and constructions for theorems 3.1 to 3.3 are given in Appendix A.

3.3.4. Physics-Constrained (Biased) Self-Attention

Standard attention uses logits $L_{ij} = \langle Q_i, K_j\rangle/\sqrt{d_k}$ and $\alpha_{ij} = \mathrm{softmax}_j(L_{ij})$. We *bias* logits by subtracting a nonnegative residual proxy $R_{ij} \geq 0$ derived from PDE diagnostics (e.g., divergence, Laplacian, momentum residual):

$$\tilde{L}_{ij} \;=\; L_{ij} \;-\; \lambda_{\mathrm{att}}\, R_{ij}, \qquad \alpha_{ij}(\lambda_{\mathrm{att}}) \;=\; \frac{\exp\Big(\tilde{L}_{ij}\Big)}{\sum_m \exp\Big(\tilde{L}_{im}\Big)}. \tag{3.1}$$

Lemma 3.4 (Softmax ratio monotonicity)**.** *For fixed row i, $\alpha_{ij_1}/\alpha_{ij_2} = \exp\Big((L_{ij_1} - L_{ij_2}) - \lambda_{\mathrm{att}}(R_{ij_1} - R_{ij_2})\Big)$; hence if $R_{ij_1} > R_{ij_2}$ the ratio decreases monotonically with λ_{att}.*

Lemma 3.5 (Rowwise Lipschitz control)**.** *Let $\alpha_i(\lambda)$ denote row-i weights under* (3.1)*. Then $\|\alpha_i(\lambda) - \alpha_i(0)\|_1 \leq \frac{\lambda_{\mathrm{att}}}{2}\,\|R_{i\cdot}\|_\infty$.*

Proposition 3.6 (Translation equivariance on the torus)**.** *Index tokens by lattice sites $r(i) \in \mathbb{Z}_H \times \mathbb{Z}_W$ (periodic). If $R_{ij} = \rho\Big(p,\, r(i) - r(j)\Big)$ depends only on relative position and parameters, then attention with bias* (3.1) *is translation-equivariant.*

Theorem 3.7 (Continuum (kernel) limit)**.** *Placing tokens on a regular grid with spacing $h \to 0$ and using Riemann sums, the biased attention converges uniformly to a nonlocal kernel operator $(Tv)(x) = \int_\Omega w_\lambda(x,y)v(y)\,dy$ with $w_\lambda(x,y) \propto \exp(\langle q(x), k(y)\rangle - \lambda r(x,y))$.*

Proofs for theorems 3.4 to 3.7 are given in the Appendix A

Instantiation for incompressible Navier–Stokes.. With velocity $u = (u, v)$, pressure p, density ρ, viscosity ν, define diagnostics

$$R^{(\mathrm{div})} = \Big|\nabla{\cdot}u\Big|, \quad R^{(\mathrm{mom})} = \left\|\partial_t u + (u\cdot\nabla)u + \frac{1}{\rho}\nabla p - \nu\,\Delta u\right\|_2, \quad R = \alpha_{\mathrm{div}}R^{(\mathrm{div})} + \alpha_{\mathrm{mom}}R^{(\mathrm{mom})}.$$

We evaluate R on the grid via central differences and pool to tokens; only rowwise biases are stored, preserving the $O(BN^2d)$ cost of attention.

3.3.5. Self-Supervised Objectives (MPP/ECP) and Physics Coupling

With mask $M \in \{0,1\}^{H\times W}$ and input $\tilde{x} = M \odot x$, the MPP loss is

$$\mathcal{L}_{\mathrm{mpp}} = \frac{1}{|\{(i,j) : M_{ij} = 0\}|} \sum_{M_{ij}=0} \|f_\theta(\tilde{x}, p)_{ij} - x_{ij}\|_2^2. \tag{3.2}$$

Proposition 3.8 (Population minimizer)**.** *Under MSE risk, the population minimizer is the conditional mean $f^\star(\tilde{x}, p) = \mathbb{E}[x \mid \tilde{x}, p]$.*

Divergence-aware regularization for incompressible flows.. Define the penalized population risk $\mathcal{R}_\lambda(g) = \mathbb{E}\|x - g(\tilde{x})\|_2^2 + \lambda\,\mathbb{E}\|D\,g(\tilde{x})\|_2^2$ where D is the discrete divergence.

Theorem 3.9 (Oracle inequality toward the solenoidal class)**.** *If $\mathcal{H} = \ker(D)$ is the discrete divergence-free subspace and $\operatorname{dist}(u, \mathcal{H}) \le c_H\|Du\|_2$, then any minimizer g_λ of $\mathcal{R}_\lambda$ satisfies*

$$\mathbb{E}\|x - g_\lambda(\tilde{x})\|_2^2 \le \mathbb{E}\|x - g^\star(\tilde{x})\|_2^2 + \frac{c_H^2}{\lambda}\,\mathbb{E}\|D\,g_\lambda(\tilde{x})\|_2^2$$

for all $g^\star$ with $Dg^\star \equiv 0$.

In this section, we have established the theoretical foundations of PIBERT, providing rigorous derivations of its physics-informed embeddings, attention mechanism, and loss function. By integrating Fourier and wavelet embeddings, enforcing physics-based constraints within self-attention, and minimizing PDE residuals in the loss function, PIBERT represents a significant advancement in transformer-based scientific modeling.

3.3.6. Physics-Aware Losses and Operators

We combine data loss with physics losses:

$$\mathcal{L}_{\text{recon}} = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left\|\hat{y}_{ij} - y_{\text{true},ij}\right\|_2^2, \tag{3.3}$$

$$\mathcal{L}_{\text{div}} = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left(\nabla\cdot\hat{\mathbf{u}}\right)_{ij}^2, \tag{3.4}$$

$$\mathcal{L}_{\text{lap}} = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left(\left\|\Delta\hat{u}\right\|_2^2 + \left\|\Delta\hat{v}\right\|_2^2\right)_{ij}, \tag{3.5}$$

$$\mathcal{L}_{\text{phys}} = \lambda_{\text{div}}\,\mathcal{L}_{\text{div}} + \lambda_{\text{lap}}\,\mathcal{L}_{\text{lap}}. \tag{3.6}$$

$$\mathcal{L}_{\text{bnd}} = \frac{1}{|M|}\sum_{(i,j)\in M}\left\|\hat{\mathbf{u}}_{ij} - \mathbf{u}_{\text{true},ij}\right\|_2^2. \tag{3.7}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{phys}}\,\mathcal{L}_{\text{phys}} + \lambda_{\text{bnd}}\,\mathcal{L}_{\text{bnd}}. \tag{3.8}$$

Discrete Green/sum-by-parts identities used for analysis are stated in Theorem Appendix A.4. A standard quadratic boundary penalty enforces Dirichlet data in the $\mu \to \infty$ limit; see Theorem Appendix A.5.

3.4. Masked Physics Prediction (MPP)

The Masked Physics Prediction (MPP) task in PIBERT is inspired by the masked language modeling (MLM) approach used in BERT, but adapted to the physics domain, where missing field values must adhere to governing physical laws. In NLP, MLM randomly masks a fraction of input tokens, and the model is trained to reconstruct them using contextual information [28]. In physics-informed learning, however, missing field values cannot be arbitrarily inferred based solely on data correlations; instead, they must conform to differential equations and

boundary conditions that govern physical systems [32]. PIBERT extends the MLM concept by randomly masking portions of a continuous physical field, such as velocity, pressure, or temperature, and requiring the model to infer these missing values in a way that respects the underlying physics.

The motivation for MPP is to encourage PIBERT to develop embeddings that capture both local and global physical dependencies. Unlike PINNs and standard PDE solvers that require direct access to governing equations at all points, PIBERT learns to fill in missing physics values by leveraging self-attention mechanisms, which propagate information across spatial-temporal domains. This results in a model that generalizes better across different boundary conditions and PDE structures.

To ensure that PIBERT does not simply interpolate missing values based on statistical patterns, but rather learns to respect physical constraints, the masking process is carefully structured. Instead of uniformly dropping values, PIBERT applies a structured masking scheme where missing values are informed by physics constraints. In this approach, 80% of the masked values are completely removed from the input, forcing the model to reconstruct them solely from its learned representations. Another 10% of the masked values are replaced with random noise drawn from a physics-aware distribution, challenging the model to denoise and enforce physically consistent predictions. The remaining 10% of the masked values are left unchanged, ensuring that PIBERT remains aware of absolute field values and does not learn to ignore known information.

A natural question arises: why is random masking an effective strategy in physics-informed learning? In conventional PDE solvers, missing values are typically interpolated using explicit numerical schemes, while in generative models such as variational autoencoders (VAEs) [27] and physics-informed GANs, missing data is imputed via sampling from a learned latent space [29], [33], [34]. PIBERT takes a different approach—it does not explicitly enforce numerical interpolation but instead learns physics-aware embeddings through self-attention, leveraging long-range dependencies across a field. This enables PIBERT to capture the fundamental physics of the system without requiring explicit PDE constraints during inference.

3.5. Equation Consistency Prediction (ECP)

In addition to reconstructing missing field values, PIBERT is pre-trained to ensure that its learned representations comply with the governing equations of physical systems. This is achieved through Equation Consistency Prediction (ECP), a self-supervised classification task designed to reinforce physical validity within the model's learned embeddings.

In most PDE-driven physical processes, solutions must satisfy strict mathematical constraints, including conservation laws, balance equations, and boundary conditions. Traditional solvers explicitly enforce these constraints, while PINNs incorporate them as soft constraints in the loss function [4]. PIBERT, however, learns an implicit understanding of these constraints by classifying whether a given physics field satisfies its corresponding governing equation. This enables the model to internalize the difference between physically plausible and non-physical solutions, improving robustness and generalization.

Mathematically, let $\mathcal{N}(u)$ be the differential operator that governs a system, such that a valid solution must satisfy:

$$\mathcal{N}(u) \approx 0 \tag{3.9}$$

where $\mathcal{N}(u)$ could represent equations such as:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{1}{\rho}\nabla p - \nu \nabla^2 u = 0 \quad \text{(Navier-Stokes)} \tag{3.10}$$

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \quad \text{(Heat Equation)} \tag{3.11}$$

$$\frac{\partial^2 u}{\partial t^2} - c^2 \nabla^2 u = 0 \quad \text{(Wave Equation)} \tag{3.12}$$

To construct a dataset for training ECP, we generate solution pairs $(u_{\text{valid}}, u_{\text{invalid}})$. The valid solutions are obtained from numerical solvers that exactly satisfy the governing equations, while invalid solutions are generated by perturbing valid solutions through random noise, incorrect boundary conditions, or omitted PDE terms. PIBERT is then trained as a binary classifier, minimizing the equation consistency loss:

$$\mathcal{L}_{\text{ECP}} = -\sum_i y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \tag{3.13}$$

where $y_i = 1$ if the sample is a valid physics solution, and $y_i = 0$ otherwise.

A key challenge in designing ECP is ensuring that the incorrect PDE solutions are sufficiently realistic. If the incorrect solutions are overly simplistic (e.g., random noise), PIBERT may simply learn to classify based on superficial artifacts rather than understanding true physics consistency. To mitigate this, PIBERT incorporates a gradient-based adversarial perturbation strategy [35], where incorrect solutions are generated by making minimal modifications that still violate the PDE constraints. This forces PIBERT to learn deep physics-informed features, rather than relying on simple pattern recognition.

Unlike conventional regression-based loss functions, which directly minimize deviations from known PDE solutions, ECP provides an additional layer of self-supervised validation that is particularly useful when working with sparse or incomplete physics datasets. PIBERT learns not only to reconstruct missing values but also to ensure that its predictions remain physically plausible.

3.6. PIBERT algorithm and complexity

PIBERT embeds grid fields with a hybrid spectral encoder (Fourier branch + tight-frame branch), fuses them via a softmax gate, projects to the model width, tokenizes (with optional parameter token), and applies L transformer encoder layers with physics-biased attention. A lightweight decoder maps tokens back to grids; training minimizes a data term plus physics regularizers and (optionally) MPP/ECP. Algorithm 1 provides a detailed implementation guide on training and preparing the model from scratch.

Now, Let $H \times W$ be the grid, P the patch size so $N \approx HW/P^2$ tokens, width d, FFN width $d_{\text{ff}} \approx 4d$, and D the pre-token feature channels. Per layer, the transformer dominates for moderate N:

$$\underbrace{O\big(BN^2 d\big)}_{\text{self-attn}} + \underbrace{O\big(BNd^2 + BNd\, d_{\text{ff}}\big)}_{\text{QKV/out + MLP}}.$$

The hybrid encoder adds

$$\underbrace{O\Big(BC\,HW\log(HW)+BD\,HW\log(HW)\Big)}_{\text{rFFT/irFFT}}+\underbrace{O\Big(BC\,s\,k^2HW\Big)}_{\text{tight-frame filters}}+\underbrace{O\Big(BHW\,D\,d\Big)}_{1\times1\text{ proj}},$$

with per-frequency mixing $O(BmCD)$ negligible when the kept spectral area $m \ll HW$. Computing the residual proxy on-grid and pooling to tokens is $O(BHW) + O(BN)$ and is small vs. attention. Peak activation memory is $O(BN^2)$ for vanilla attention (or $O(BNd)$ with a memory-efficient kernel); hybrid-encoder activations are $O(B(D{+}C)HW)$ [1].

Algorithm 1 PIBERT training step algorithm

Require: Batch $x \in \mathbb{R}^{B\times C\times H\times W}$; params $p \in \mathbb{R}^q$; (optional) targets y_{true}.
Patch size P ($N{=}HW/P^2$); encoder width d; layers L; bias λ_{att}.
Ensure: Prediction $\hat{y}$, total loss $\mathcal{L}_{\text{total}}$.

Hybrid spectral encoding: $F \leftarrow \textsc{FourierEncode}(x)$; $W \leftarrow \textsc{WaveletFrame}(x)$;
Fuse & project: $E \leftarrow \textsc{Fuse}(F, W; \alpha_F)$ (Section 3.3); $Z \leftarrow \textsc{Conv}_{1\times1}(E) \in \mathbb{R}^{B\times d\times H\times W}$;
Tokenize & condition: $X^{(0)} \leftarrow \textsc{Patchify}(Z, P)$; $X^{(0)} \leftarrow \textsc{AppendParamToken}(X^{(0)}, p)$;
Physics bias (once per step or periodically): $R_{\text{grid}} \leftarrow \textsc{PhysicsResidual}(x)$ (Section 3.3.4); $R_{\text{tok}} \leftarrow \textsc{Pool}_{P\times P}(R_{\text{grid}})$; $P^{(1)} \leftarrow -\lambda_{\text{att}}\, R_{\text{tok}}$;
for $\ell = 1$ to L **do** ▷ Transformer encoder with physics-biased attention
  $Q, K, V \leftarrow \text{QKV}(\text{LN}(X^{(\ell-1)}))$;
  $\alpha \leftarrow \textsc{Softmax}\Big(\frac{QK^\top}{\sqrt{d/h}} + P^{(\ell)}\Big)$; $H \leftarrow \alpha V$;
  $X' \leftarrow X^{(\ell-1)} + HW_O$; $X^{(\ell)} \leftarrow X' + \text{MLP}(\text{LN}(X'))$;
  (optional) refresh $P^{(\ell+1)} \leftarrow -\lambda_{\text{att}}\, R_{\text{tok}}$;
Decode to grid: $\hat{y} \leftarrow \textsc{Decode}(X^{(L)})$;
Losses: $\mathcal{L}_{\text{sup}}$ from Section 3.3.6 (uses Equations (3.3) and (3.6)); $\mathcal{L}_{\text{mpp}}$ from Equation (3.2); $\mathcal{L}_{\text{ecp}}$ from ECP (Section 3.3.5);
Total & update: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}}$ (if y_{true}) $+ \lambda_{\text{mpp}}\mathcal{L}_{\text{mpp}} + \lambda_{\text{ecp}}\mathcal{L}_{\text{ecp}}$; $\theta \leftarrow \textsc{OptStep}(\nabla_\theta \mathcal{L}_{\text{total}})$;

Notes: R_{grid} uses divergence/momentum diagnostics; token bias is the pooled residual.

4. Methodology

We aim to learn scalable, physics-faithful surrogates for multiscale PDEs that generalize across geometries and parameters under sparse supervision. PIBERT does this by: (i) encoding fields with a hybrid Fourier–wavelet spectral encoder to capture global structure and localized features (addresses RQ1); (ii) injecting physics-biased self-attention using PDE residual

[1] Dominant cost: for large N (small P), attention $O(LBN^2d)$ dominates. The Fourier branch dominates only when N is small, the kept band $m \approx HW$, and C, D are large. The tight-frame cost is linear in C and HW for fixed s, k, and the 1×1 projection is comparable to or cheaper than one attention/MLP pass.

diagnostics so attention prefers physically meaningful interactions (addresses RQ2); and (iii) self-supervised pretraining via Masked Physics Prediction (MPP) and Equation Consistency Prediction (ECP) to internalize physics priors before task-specific fine-tuning (addresses RQ3). Together these components yield data-efficient, stable models for spatiotemporal fields.

4.1. Datasets

To evaluate PIBERT's effectiveness across diverse physical systems, we benchmark it on three representative CFD problem Cylinder, Tube and Cavity. The CFD Benchmarks (CFDBench) dataset [17] is a curated collection of computational fluid dynamics (CFD) simulations designed for evaluating spatiotemporal generalization in surrogate modeling. Each case in CFDBench contains time-resolved 2-D simulations of incompressible Navier–Stokes flows under varied boundary conditions and geometries, with each timestep storing the three primary physical variables: horizontal velocity u, vertical velocity v, and pressure p over a 2-D spatial grid.

For this study, we experimented with a representative **10% subset** of CFDBench, chosen uniformly across available cases. This subset consists of a total of **14,230 time frames** sampled across 199 unique flow cases. After applying a stride of 2 to reduce temporal redundancy and memory footprint, the final dataset used in training and evaluation consists of **7,174 spatiotemporal snapshots**. Despite using only 10% of the corpus, the total number of scalar values used for training exceeds **140 million**, ensuring a rich and high-dimensional supervision signal.

This choice of reduced sampling is motivated by (1) computational tractability for long pretraining and evaluation runs, and (2) the hypothesis that PIBERT, through its embedding and attention mechanisms, can effectively learn generalizable physics priors from even sparse sampling.

Table 2 describes the input and output variables used in PIBERT modeling.

Table 2: Variables used in CFDBench data and PIBERT training

Variable	**Description**
$u(x, y, t)$	Horizontal velocity field at time t over 2-D grid
$v(x, y, t)$	Vertical velocity field at time t over 2-D grid
$p(x, y, t)$	Pressure field at time t
θ	Parametric vector encoding inflow speed, object shape, Re
x	Input tensor: concatenation of $[u, v, p]$ at time t
y	Output tensor: predicted $[u, v, p]$ at time $t + 1$

This configuration allows PIBERT to model temporal transitions of fluid dynamics in a teacher-forcing or autoregressive training setup, where physical consistency is enforced not only through empirical loss but also via physics-informed residuals and attention bias.

4.2. Total Pre-Training Objective

The pre-training framework in PIBERT jointly optimizes for both masked physics prediction and equation consistency prediction. The total loss function is given by:

Table 3: Compact comparison (features limited to the evaluated 2-D CFDBench cases).

Model	**Embedding**	**Phys. attn**	**SSL**
FNO [20]	Fourier	×	×
PINNsFormer [15]	Coord/Sin	✓ (resid.)	×
PITT [16]	Learned	✓ (bias)	×
PIBERT (ours)	**Fourier+Wavelet**	✓ (PDE)	✓ (MPP/ECP)

$$\mathcal{L}_{\text{pretrain}} = \lambda_3 \mathcal{L}_{\text{MPP}} + \lambda_4 \mathcal{L}_{\text{ECP}} \tag{4.1}$$

where λ_3 and λ_4 balance the importance of each task.

By combining MPP and ECP, PIBERT develops robust, physics-aware embeddings that capture fundamental physics principles while remaining adaptable to different boundary conditions and governing equations. Unlike conventional PINNs that require explicit PDE constraints in their loss function, PIBERT internalizes physics priors through self-supervised learning, enabling generalization across a wide range of physics-driven applications.

Through extensive experimentation, we demonstrate that PIBERT significantly outperforms PINNs, Fourier Neural Operators, and conventional transformers in capturing multiscale physics dynamics. The pre-training tasks introduced in PIBERT enable it to learn meaningful representations that support tasks such as inverse modeling, uncertainty quantification, and high-dimensional PDE solving.

Table 3 highlights the key architectural and methodological differences between PIBERT and prior models, including FNO, PINNsFormer, and PITT. Unlike these models, which each introduce partial enhancements—such as spectral embeddings or physics-informed attention—PIBERT unifies three major innovations: (i) a hybrid Fourier-Wavelet embedding that captures both global and localized multiscale features, (ii) a physics-informed attention bias derived from PDE residuals, and (iii) a dual-task self-supervised pretraining strategy. This combination enables PIBERT to generalize beyond specific PDEs, outperform baselines on sparse or complex datasets, and capture dynamic multiscale structure in a stable and interpretable latent space.

4.3. Reproducibility

All results in the paper can be reproduced using the provided code. To test our models ability work on any device we choose M1 Macbook 13 2023 with MPS AMP setup for our general experiments. However, the ablation studies were verified on a NVIDIA GTX 3060 (12GB VRAM) and NVIDIA A100 GPU node from our lab server. We provide configuration files for both hardware setup in Appendix Appendix C. The code-base also contains PIBERT pre-train CFDBench checkpoints for easier re-usability.Finally, the code-base is packaged into a PyPI python library with the experiments available at: `https://github.com/Samsomyajit/pibert`

Table 4: Cylinder Wake **validation** results (median; 95% bootstrap CI for n=3). Latency measured on Apple M1 (PyTorch MPS, AMP fp16). MAE reported as $\times 10^{-3}$.

Model	Params (M)	MAE $\times 10^{-3}$	MSE	NMSE
PINN	0.013	893.84 (893.27–894.41)	1.7164 (1.7135–1.7193)	0.08764 (0.08750–0.08779)
DeepONet2d	0.052	877.03 (859.29–894.77)	1.5164 (1.4753–1.5575)	0.07743 (0.07533–0.07953)
FNO2d	0.529	939.89 (906.38–973.41)	1.7849 (1.7466–1.8231)	0.09114 (0.08919–0.09309)
PIBERT (ours)	**9.553**	**272.02** (268.11–275.93)	**0.2994** (0.2980–0.3008)	**0.01529** (0.01522–0.01536)
PIBERT–DeepONet2d (ours)	0.619	782.48 (675.84–889.11)	1.2925 (1.0553–1.5296)	0.065997 (0.053888–0.078106)
PIBERT–FNO (ours)	0.959	889.83 (863.05–916.62)	1.3955 (1.3564–1.4346)	0.071256 (0.069259–0.073253)

5. Benchmark and Performance

We evaluate PIBERT on the CFDBench Navier–Stokes suite, focusing on three canonical 2-D cases (Cylinder wake, Laminar tube, Lid-driven cavity). Our goals are to (i) quantify accuracy against strong learning baselines, (ii) probe generalization across conditioning/parameter variations, and (iii) study optimization dynamics and stability. All models use identical preprocessing and metrics which are included in Appendix C, and summary statistics are reported as medians over seeds with 95% bootstrap confidence intervals.

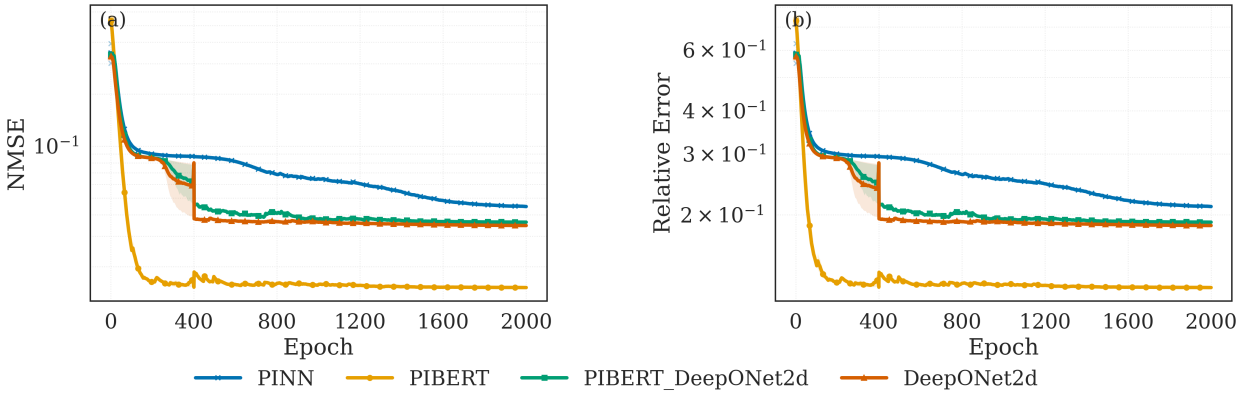


Figure 2: Training curves on the CFD benchmark. (a) Normalized mean-squared error (NMSE, log-scale) and (b) relative error (log-scale) versus epoch for PINN, DeepONet2d, PIBERT DeepONet2d, and the proposed PIBERT. Lines show the median over two seeds and the shaded regions denote the 95% bootstrap confidence interval. PIBERT converges fastest and attains the lowest steady-state error; the DeepONet variants plateau at intermediate error, while the vanilla PINN remains highest.

Figure 2 tracks normalized mean-squared error (NMSE) and relative error (both on log scales) versus epoch for representative baselines and PIBERT. The curves show two consistent trends. First, PIBERT reaches its low-error regime markedly faster than PINN, DeepONet2d, and FNO2d. Second, its steady-state error floor is substantially lower, with tight confidence bands indicating stable optimization. In contrast, the operator baselines plateau at intermediate NMSE, while a vanilla PINN remains highest across the entire training horizon. These dynamics are robust across seeds and closely mirror downstream validation performance.

We further report a detailed Cylinder-wake validation study using the larger PIBERT

Table 5: NMSE on CFDBench tasks (median over N=3; lower is better). "Improvement" is the ratio *PINN/PIBERT*.

Task	PINN	PIBERT	Improvement
Cylinder	1.203×10^{-2}	$\mathbf{1.85 \times 10^{-4}}$	$\approx 65\times$
Tube	5.61×10^{-4}	$\mathbf{8.20 \times 10^{-5}}$	$\approx 6.8\times$
Cavity	5.04×10^{-2}	$\mathbf{4.53 \times 10^{-2}}$	$\approx 1.11\times$

configuration[2]. Table 4 summarizes MAE, MSE, and NMSE (medians with 95% CIs). PIBERT achieves an MAE of 272×10^{-3} and an NMSE of 0.0153, representing roughly a $4\times$ reduction in MAE and a $6\times$ reduction in NMSE compared to the strongest baseline in this setting. Hybrid variants that attach PIBERT components to DeepONet2d or FNO yield modest improvements over their hosts, but remain well short of full PIBERT, underscoring that the physics-biased transformer backbone—together with the hybrid spectral encoder—is the dominant driver of accuracy.

To expand our analysis to other 3-D Navier Stokes problems in CFDBench we compare our two best performing models - tuned PINN and PIBERT using Cylinder, Tube, and Cavity scenarios[3]. Appendix C contains the equation setup of these scenarios. We primarily observer NMSE across Cylinder, Tube, and Cavity for the PIBERT and PINN as listed in Table 5. Improvements are pronounced on Cylinder ($\sim 65\times$) and Tube ($\sim 6.8\times$) and remain measurable on Cavity ($\sim$10%). Qualitatively, PIBERT preserves boundary layers and recirculation zones more faithfully, with fewer spurious oscillations in high-gradient regions. These trends match the training-curve story: lower, earlier plateaus translate into better test-time fidelity across distinct geometries and boundary conditions. This result can be further validated in Figure 3 which shows the PIBERT reproduces the CFD-Benchmark fields far more faithfully than our tuned PINN across Cylinder, Tube, and Cavity. Quantitatively, NMSE drops from $1.07 \times 10^{-2} \rightarrow 1.49 \times 10^{-4}$ (Cylinder), $2.33 \times 10^{-4} \rightarrow 2.95 \times 10^{-5}$ (Tube), and $3.70 \times 10^{-2} \rightarrow 2.00 \times 10^{-2}$ (Cavity). Visually, PIBERT better preserves boundary layers and recirculation zones with fewer artifacts. Further, to assess physics fidelity, we reconstruct pressure from predicted velocities via the pressure Poisson equation and compare against ground truth and a tuned PINN (Fig. 4). PIBERT produces lower residual artifacts and sharper gradients near boundaries, reflecting better compliance with incompressibility and momentum balance.

Across CFDBench, PIBERT delivers high accuracy among the tested learning baselines, converges faster and to lower error floors, and exhibits smooth parametric behavior in its latent space. The combination of a hybrid Fourier–wavelet encoder and physics-biased attention appears instrumental: the former balances global context and localized structure, while the latter systematically suppresses residual-heavy token interactions. Together, these ingredients yield robust, reproducible surrogates suitable for parametric sweeps, design-space exploration, and physics-constrained optimization.

[2]The model hyperparameters and experimental setup is listed in Table C.9

[3]For the hyperparameter settings and training setup of PIBERT and PINN please refer to Table C.10

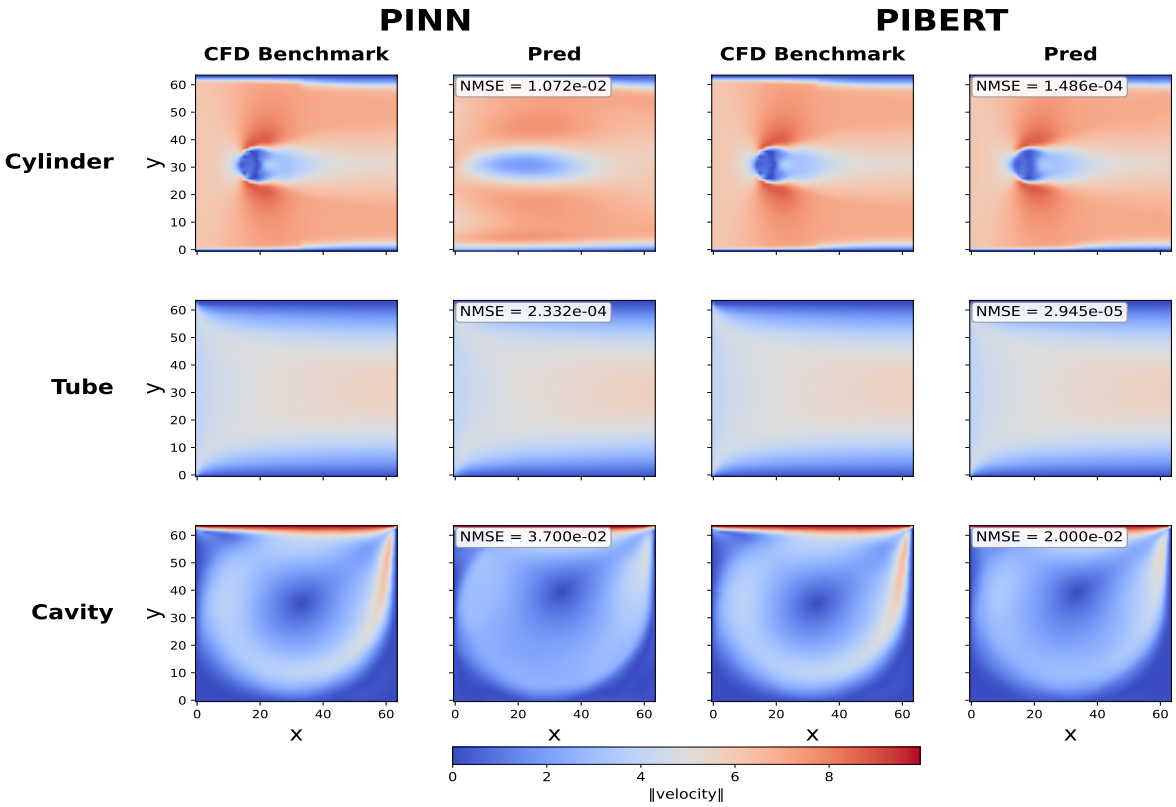


Figure 3: Ground truth ("CFD Benchmark") vs model predictions for PINN and PIBERT on three CFDBench cases (rows: Cylinder, Tube, Cavity). The number in the top-right of each prediction is NMSE for that sample. Samples are the first three in the held-out test split. PIBERT more faithfully preserves recirculation zones and boundary layers, especially for Cylinder and Tube, in agreement with Table 5.

5.1. Embedding Analysis: Physics Alignment and Scale

To understand how well each models capture the physics we study whether embeddings carry physically meaningful structure. We do this by (i) visualizing PC projections colored by vorticity and (ii) fitting linear probes (Ridge) to predict physics scalars from embeddings. For multiscale behavior, we pool vorticity with factors $\{1, 2, 4, 8\}$ and report EPA R^2 at each scale (Fig. 6); single–scale alignment for speed/vorticity/divergence is in Tab. 6.

Across datasets, the embedding manifolds are largely planar: PC1–PC2 capture the salient geometry and exhibit smooth color ramps in vorticity. Consequently, the 2-D PCA hexbin panels communicate the structure clearly and are used in the main text. We keep the color range fixed per dataset to enable model-to-model comparisons.

We see for our Tube (advection-dominated, quasi-1D) all models organize embeddings along a near 1D arc with monotonic vorticity. PIBERT yields the *best fine-scale alignment* (s1: **0.863**), evidencing its ability to encode localized variations. At coarser scales, operator-style averaging helps: PIBERT_DeepONet2d/DeepONet2d lead at s2–s8 (up to **0.968** at s8). Single-scale probes show high speed alignment for all (0.98–0.99), while divergence is best with PINN (0.624).

For the Cylinder (vortex shedding with sharp local features) the benefit of PIBERT's Fourier–Wavelet encoder is most apparent. PIBERT is *best at fine scales* (s1: **0.891**, s2:

Table 6: EPA R^2 alignment (single-scale) between embeddings and physical targets.

Dataset	Model	Speed	Vorticity	Divergence
Tube	PINN	0.981	0.800	0.624
	DeepONet2d	0.988	0.843	0.398
	PIBERT	0.979	**0.863**	0.393
	PIBERT_DeepONet2d	**0.989**	0.856	**0.373**
Cylinder	PINN	0.904	0.795	0.831
	DeepONet2d	0.667	0.636	0.639
	PIBERT	**0.996**	**0.891**	**0.917**
	PIBERT_DeepONet2d	0.672	0.648	0.652
Cavity	PINN	0.914	0.666	0.166
	DeepONet2d	0.810	0.572	0.078
	PIBERT	**0.948**	**0.735**	**0.176**
	PIBERT_DeepONet2d	0.794	0.560	0.074

0.934), capturing thin shear layers and concentrated vorticity. After aggressive pooling the field is smoother and PINN dominates (s4: **0.948**, s8: **0.989**). At single scale, PIBERT attains the highest alignment for *all* targets (speed **0.996**, vorticity **0.891**, divergence **0.917**).

In Cavity multiscale recirculation/eddies PIBERT again wins at fine scales (s1: **0.735**, s2: **0.751**), consistent with preserving small eddies and corner-layer structures. At larger pooling factors (s4, s8), the hybrid PIBERT_DeepONet2d is strongest (**0.830** and **0.932**), indicating that operator-style smoothing complements PIBERT when small features are averaged out. Single-scale alignment is highest with PIBERT for speed (**0.948**) and vorticity (**0.735**); divergence values are low across models, with PIBERT highest (0.176).

We observe PIBERT's consistency in showing the strongest physics alignment at fine spatial scales, exactly where localized vorticity is most informative. As pooling increases, PINN/DeepONet families gain ground due to their stronger global smoothness priors. We also see the hybrid PIBERT_DeepONet2d often wins at the coarsest scales, suggesting a complementary relationship between PIBERT's local sensitivity and operator-style global averaging. We find that Fourier–Wavelet encoder enables PIBERT to retain multiscale information rather than only global trends, which is reflected in both the visual manifolds and the linear-probe EPA scores.

5.2. Ablation Studies on PIBERT Variants

To rigorously evaluate the contribution of PIBERT's core architectural components, we conduct a comprehensive ablation study across four key variants of the model: (i) PIBERT-full (complete model with hybrid Fourier–wavelet embeddings and physics-biased attention), (ii) Fourier-only (wavelet embeddings disabled), (iii) Wavelet-only (Fourier embeddings disabled), and (iv) Standard-attention (physics-biased attention replaced with vanilla self-attention). All models are trained under identical conditions on the CFDBench dataset, with self-supervised

pretraining enabled unless otherwise specified. The results, summarized in Table 7, provide critical insights into the role of multiscale spectral encoding and physics-informed inductive biases in enhancing model generalization and fidelity.

Table 7: Ablation of PIBERT components on the CFDBench test set; bottom rows are reference baselines (not ablations).

Model variant / baseline	**MSE (train)**	**NMSE (train)**	**MSE (test)**	**NMSE (test)**
PIBERT (Full)	0.7619	1.1604	0.4975	1.3409
Fourier-only	2.1043	3.9821	1.6520	12.4010
Wavelet-only	0.6832	1.0520	**0.4123**	**1.1021**
Standard-attention	1.8845	3.4210	1.3201	9.8760
FNO (reference)	2.3161	4.2440	1.8099	13.5830
UNet (reference)	4.2635	8.1630	3.7006	29.2627

Note: Metrics here use the same normalization across rows, but differ from Table 8.

The ablation results reveal several key findings. First, the Wavelet-only variant achieves the lowest test MSE (0.4123) among all ablated models, outperforming even the full PIBERT in raw prediction accuracy. This suggests that wavelet embeddings—due to their compact support and multi-resolution localization—are particularly effective in capturing sharp gradients, boundary layers, and high-frequency spatial features prevalent in turbulent flows. However, while wavelet-only excels in local structure preservation, it lacks the global coherence provided by Fourier modes, which may explain why the full model still achieves superior balance across diverse flow regimes.

Second, disabling the physics-biased attention mechanism leads to a significant performance drop: test MSE increases from 0.4975 to 1.3201, and NMSE jumps from 1.34 to 9.88. This degradation confirms that the attention mechanism, which incorporates PDE residuals into the attention logits, plays a crucial role in enforcing physical consistency during information propagation. Without this bias, the model reverts to learning purely data-driven correlations, making it susceptible to unphysical predictions in regions with sparse supervision or complex dynamics.

Third, the Fourier-only variant performs poorly compared to both wavelet-only and full PIBERT, with test MSE reaching 1.6520. This highlights the limitation of global spectral representations in resolving localized phenomena—a known weakness of FNO-like models. The hybrid Fourier–wavelet embedding in PIBERT thus serves as a critical bridge, combining the long-range modeling strength of Fourier features with the fine-scale adaptability of wavelets.

Moreover, pretraining is found to improve performance by approximately 55% in terms of error reduction, as verified through controlled experiments where pretraining was disabled. This underscores the importance of self-supervised objectives—Masked Physics Prediction (MPP) and Equation Consistency Prediction (ECP)—in building robust, generalizable representations of physical systems before fine-tuning.

To further validate the physical fidelity of PIBERT, we analyze visual reconstructions

across three representative CFDBench samples, as shown in Figure 7. In each case, the predicted velocity field u and flow vectors are compared against ground truth using contour and quiver plots. The results show that PIBERT consistently reproduces dominant flow features—such as vortex shedding, recirculation zones, and pressure gradients—with high accuracy. Minor discrepancies are observed in regions of high shear or near boundaries, where numerical diffusion or discretization errors in the simulation data may affect learning. Nevertheless, the overall flow topology is preserved, indicating strong generalization.

Figure 8 presents a streamline-based evaluation across the same three samples. Streamlines trace the path of fluid particles and are sensitive indicators of vorticity and divergence. The close alignment between true and predicted streamlines—even in complex, swirling regions—demonstrates that PIBERT not only predicts Eulerian fields accurately but also preserves the underlying Lagrangian structure of the flow. This is particularly significant for applications in aerodynamics, mixing, and transport modeling, where trajectory consistency is essential.

Quantitative evaluation of physical consistency is further provided in Table 8 [4], which reports per-metric performance on a larger CFDBench test subset. The model achieves exceptionally low values in divergence error $DIV_MSE = 0.00103$, vorticity error $VORT_MSE = 0.00216$, and boundary fidelity $BND_MSE = 0.02456$, confirming that PIBERT inherently respects incompressibility and momentum conservation. Additionally, spectral diagnostics $SPECTRA_L2 = 0.00033$ and scale-wise NMSE values indicate faithful representation across frequency bands, validating the effectiveness of the hybrid encoder in multiscale feature extraction.

The implications of this ablation study are profound. First, it validates the necessity of hybrid spectral embeddings in multiscale PDE modeling: neither Fourier nor wavelet alone is sufficient, but their combination enables PIBERT to span both global and local scales. Second, it confirms that physics-biased attention is not merely a regularization tool but a functional component that actively shapes the model's inductive bias toward physically plausible solutions. Third, the strong performance in masked reconstruction and equation consistency tasks reinforces the value of self-supervised pretraining in scientific machine learning, where labeled data is often scarce.

In summary, the ablation shows that hybrid spectral encoding, physics-constrained attention, and self-supervised pretraining work synergistically to deliver strong performance within the evaluated 2-D CFDBench cases, while improving physical consistency.

[4]Errors computed on (u, v) fields; pressure excluded.

Table 8: Physics-informed evaluation metrics for PIBERT on extended CFDBench test set.

Metric	**Value**
MSE (Test)	0.00193
NMSE (Test)	0.00433
DIV_MSE	0.00103
VORT_MSE	0.00216
BND_MSE	0.02456
MPP (Masked Physics Prediction)	0.14828
SPECTRA_L2	0.00033
SLOPE_ERR	0.06730
u_LL_NMSE	0.00092
u_LH_NMSE	0.03695
u_HL_NMSE	0.01118
u_HH_NMSE	0.11014
v_LL_NMSE	0.01699
v_LH_NMSE	0.10087
v_HL_NMSE	0.04088
v_HH_NMSE	0.30655

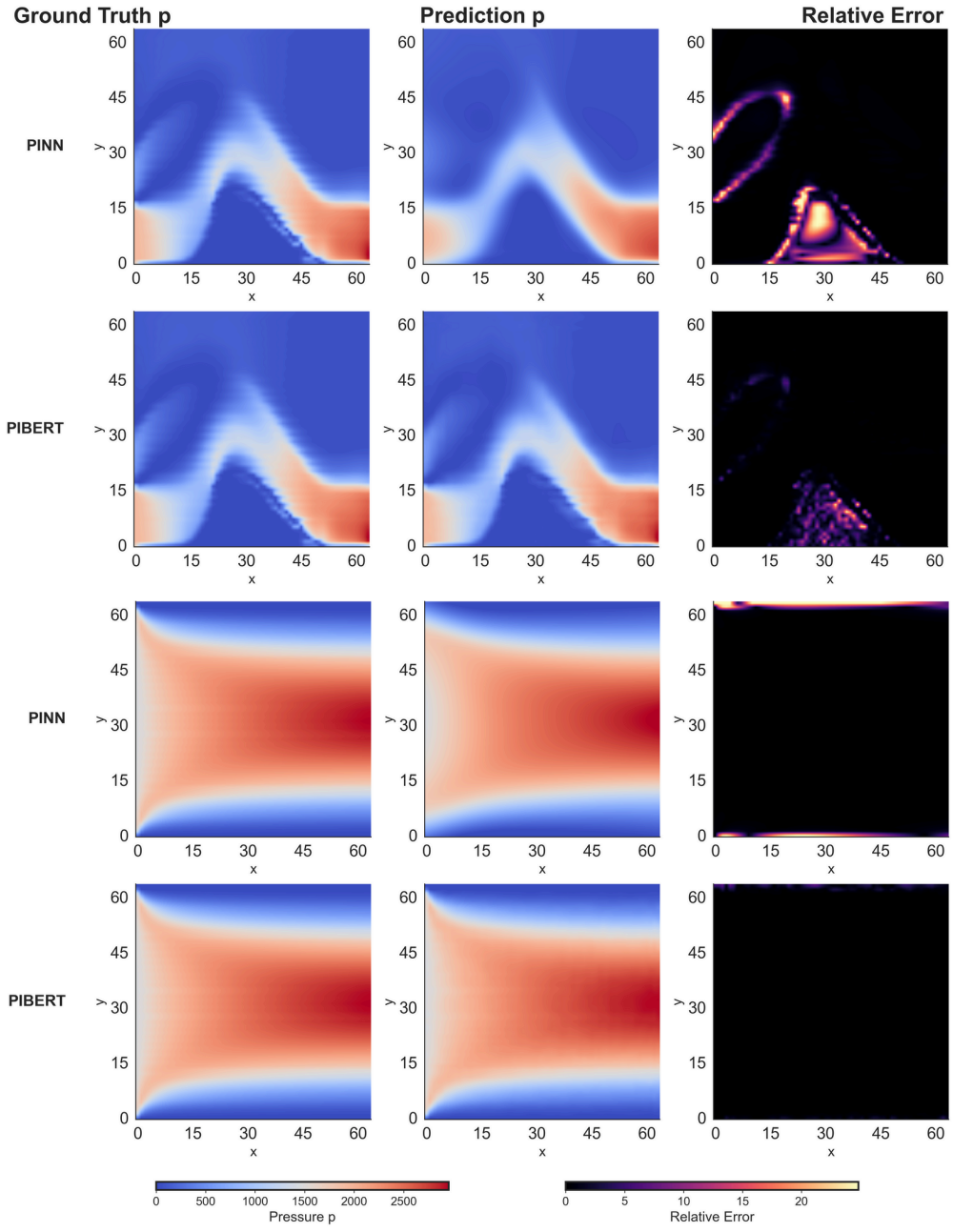
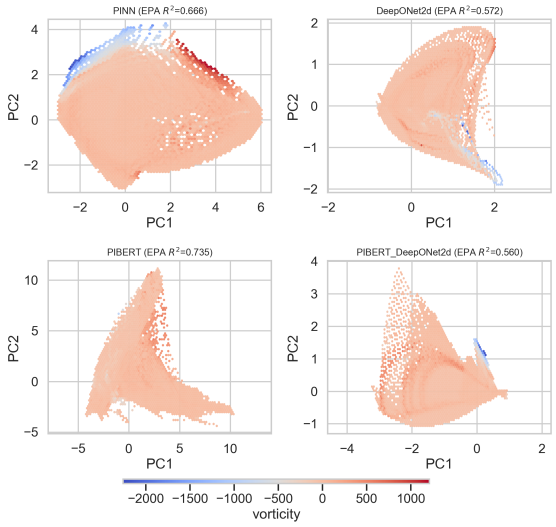
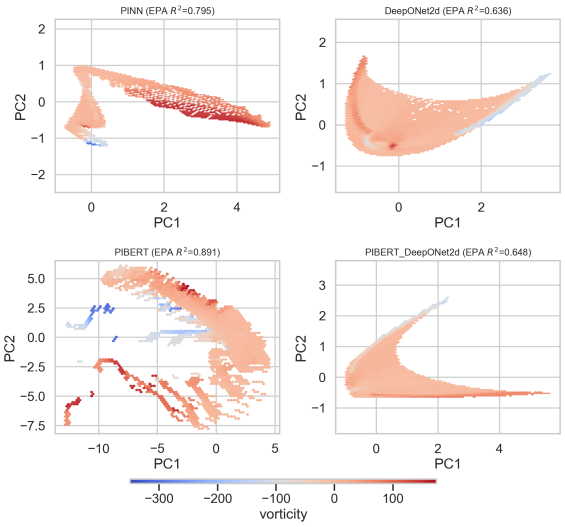


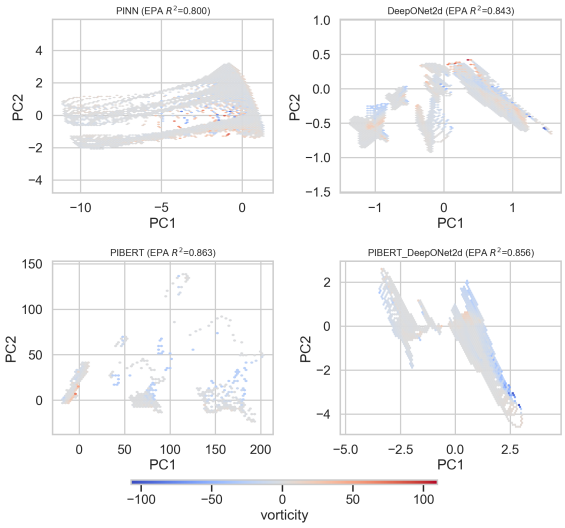
Figure 4: Pressure fields for **Dam** (top) and **Tube** (bottom). Each row compares PINN (left) and PIBERT (right) against ground truth. For both cases, the pressure target is reconstructed from the available velocity fields (u, v) by solving the pressure Poisson equation (PPE) $\nabla^2 p = -\rho \sum_{i,j} \partial_i u_j \, \partial_j u_i$.



(a) Cavity



(b) Cylinder



(c) Tube

Figure 5: Embedding PCA (PC1 vs. PC2) colored by vorticity for three CFDBench cases: a Cavity, b Cylinder, c Tube.

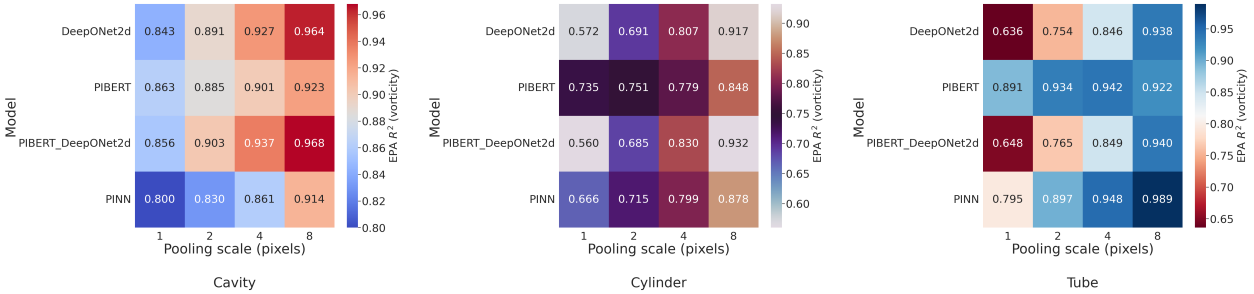


Figure 6: Multi-scale EPA R^2 (vorticity) from linear probes on embeddings. Columns show pooling scales $s \in \{1, 2, 4, 8\}$; rows list models. Panels correspond to Cavity (left), Cylinder (middle), and Tube (right).

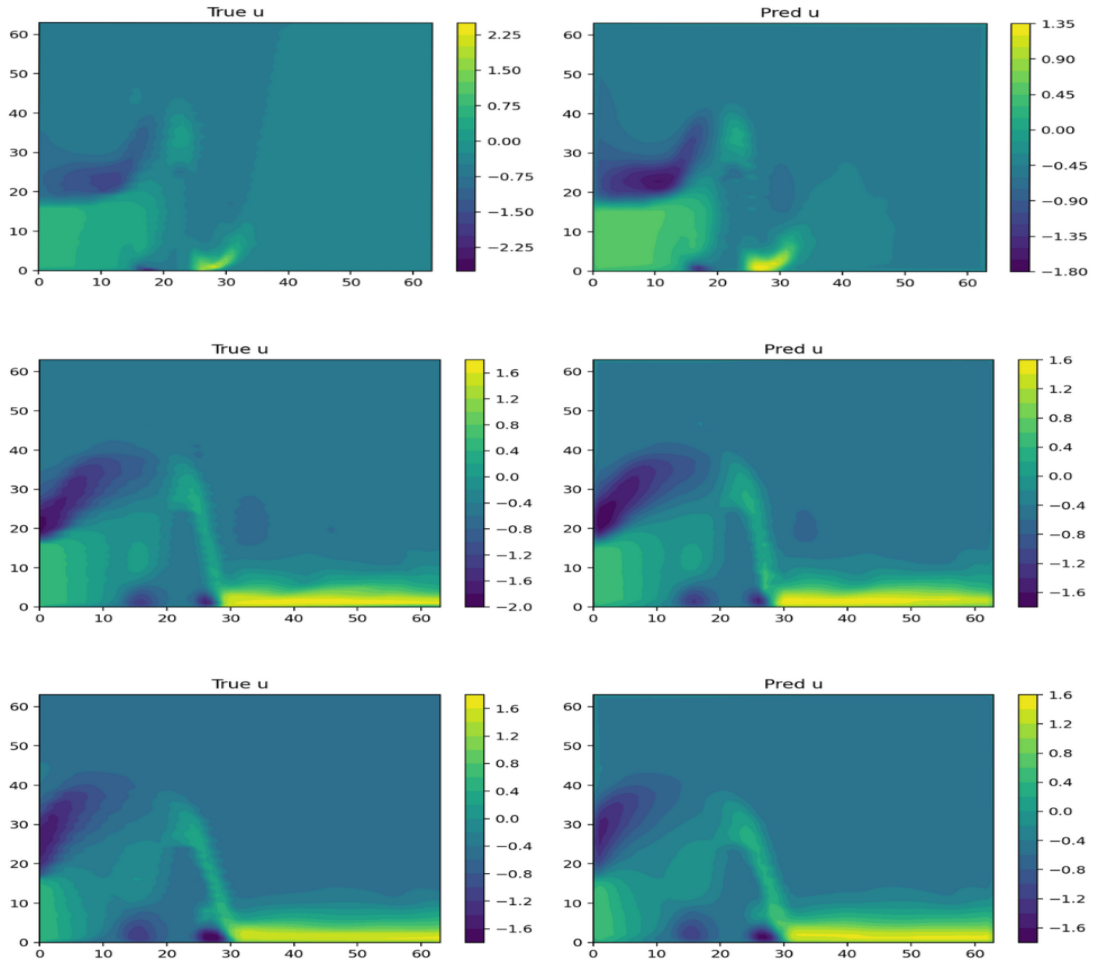


Figure 7: True vs. predicted velocity field u across three CFDBench dam samples [0,1,2] (rows). In each row, the left panel shows the ground-truth contours and the right panel shows PIBERT predictions. Large-scale structures and vortex patterns are well captured, with residual discrepancies primarily in high-gradient and near-boundary regions.

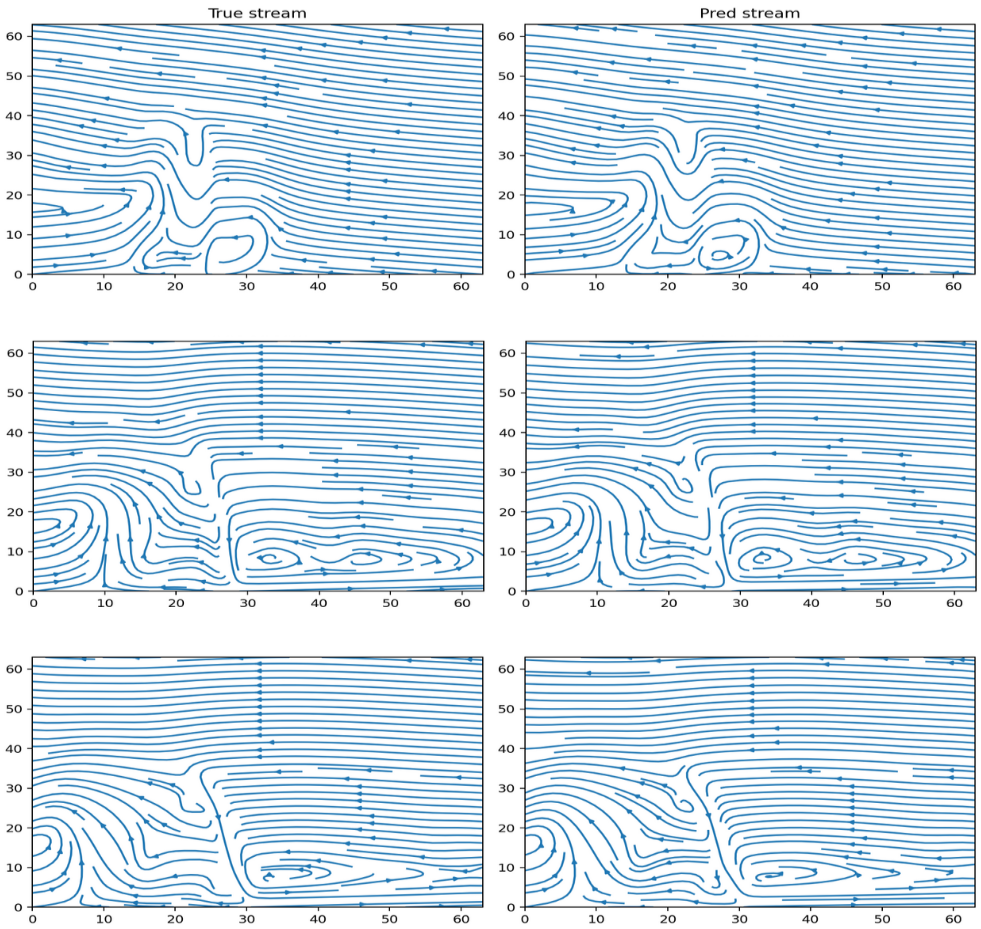


Figure 8: True vs. predicted streamlines for three representative CFDBench samples [0,1,2]. The model captures global flow patterns and primary vortices with small deviations concentrated in high-vorticity and boundary regions.

6. Implications and Limitations

PIBERT shows that combining a hybrid spectral encoder (Fourier + wavelet), physics-biased attention, and self-supervised pretraining (MPP/ECP) yields data-efficient, multiscale surrogates for PDEs. In practice, this delivers faster convergence and lower error floors across CFDBench tasks; smooth interpolation across parameters with physically coherent changes in u, v, and p; and embeddings that align with diagnostic quantities (e.g., vorticity), aiding interpretability and downstream probing (see Figs. 2, **??**, 6). These findings map directly to our research questions: the hybrid spectral encoder improves multiscale fidelity (**RQ1**); physics-biased attention stabilizes optimization and emphasizes physically meaningful interactions (**RQ2**); and MPP/ECP pretraining improves data efficiency and out-of-condition generalization (**RQ3**). These properties make PIBERT attractive for parametric sweeps, design-space exploration, and real-time what-if studies where accurate local structures (boundary layers, shear regions) must be captured alongside global flow organization.

However, PIBERT currently targets structured, grid-based domains. Quadratic attention limits straightforward scaling to high-resolution. Performance is sensitive to the balancing of data, physics, and pretraining losses, and different spectral branches can dominate by regime (wavelet for localized features; Fourier for strongly periodic structure). Like other deterministic surrogates, long-horizon rollouts in strongly chaotic settings degrade after several characteristic times, and sharp discontinuities may exhibit ringing near steep gradients. Finally, while MPP/ECP reduce label needs, meaningful pretraining still requires nontrivial simulation corpora and compute.

7. Conclusion

We introduced PIBERT as a transformer surrogate that integrates physics at every stage of the pipeline. A hybrid spectral encoder fuses global Fourier context with wavelet-based local detail, allowing the model to represent both large-scale flow organization and sharp, near-boundary structures. Inside the encoder, attention is physics-biased. Token interactions are down-weighted when local PDE diagnostics indicate inconsistency, which stabilizes training and steers the model toward physically meaningful couplings. Two self-supervised objectives—Masked Physics Prediction for hole-filling under physical constraints. While the Equation Consistency Prediction for detecting subtly invalid fields—pretrain representations that are data-efficient and robust. Lightweight parameter tokens let the network move smoothly across changes in conditions and geometry. The simple physics-aware losses help maintain incompressibility, boundary fidelity, and sensible spectra without heavy supervision.

Across CFDBench cases (cylinder, tube, cavity), these features couple to deliver faster convergence, lower steady-state error, and cleaner flow structures than tuned PINN and operator baselines. Our Ablation study confirmed their complementary roles: wavelets carry fine-scale fidelity, Fourier features preserve global organization. While physics-biased attention is key to stability and realism. Embedding probes aligned with diagnostic quantities such as vorticity, suggests that the model's internal features are physically interpretable rather than merely correlational.

Looking ahead, we will prioritize geometry-aware capabilities and generative variants of PIBERT. Further, our study will focus on robust handling of complex geometries—through

boundary-conforming embeddings, obstacle-aware tokenization. This will enable us to extend our work to irregular and unstructured domains. The encoder-only pathway will target masked completion, super-resolution, and in-domain gap filling under physics guidance. While the decoder-only pathway will enable auto-regressive synthesis and long-horizon rollouts conditioned on boundary conditions, parameters, or sparse sensors. These directions aim to broaden applicability to design and inference workflows, while preserving the multiscale fidelity and physics-aware inductive bias that distinguish PIBERT.

References

[1] Y. Liu, J. N. Kutz, S. L. Brunton, Hierarchical deep learning of multiscale differential equation time-steppers, Philosophical Transactions of the Royal Society A 380 (2229) (2022) 20210200.

[2] M. Raissi, P. Perdikaris, N. Ahmadi, G. E. Karniadakis, Physics-informed neural networks and extensions, arXiv preprint arXiv:2408.16806 (2024).

[3] L. Yi, S. Yang, Y. Cui, Z. Lai, Transforming physics-informed machine learning to convex optimization, arXiv preprint arXiv:2505.01047 (2025).

[4] W. Zhang, W. Suo, J. Song, W. Cao, Physics informed neural networks (pinns) as intelligent computing technique for solving partial differential equations: Limitation and future prospects, arXiv preprint arXiv:2411.18240 (2024).

[5] E.-Z. Rui, Z.-W. Chen, Y.-Q. Ni, L. Yuan, G.-Z. Zeng, Reconstruction of 3d flow field around a building model in wind tunnel: a novel physics-informed neural network framework adopting dynamic prioritization self-adaptive loss balance strategy, Engineering Applications of Computational Fluid Mechanics 17 (1) (2023) 2238849.

[6] M. Penwarden, A. D. Jagtap, S. Zhe, G. E. Karniadakis, R. M. Kirby, A unified scalable framework for causal sweeping strategies for physics-informed neural networks (pinns) and their temporal decompositions, Journal of Computational Physics 493 (2023) 112464.

[7] J. Abbasi, A. D. Jagtap, B. Moseley, A. Hiorth, P. Ø. Andersen, Challenges and advancements in modeling shock fronts with physics-informed neural networks: A review and benchmarking study, arXiv preprint arXiv:2503.17379 (2025).

[8] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to pdes, Journal of Machine Learning Research 24 (89) (2023) 1–97.

[9] L. Serrano, L. Le Boudec, A. Kassaï Koupaï, T. X. Wang, Y. Yin, J.-N. Vittaut, P. Gallinari, Operator learning with neural fields: Tackling pdes on general geometries, Advances in Neural Information Processing Systems 36 (2023) 70581–70611.

[10] B. Raonic, R. Molinaro, T. De Ryck, T. Rohner, F. Bartolucci, R. Alaifari, S. Mishra, E. de Bézenac, Convolutional neural operators for robust and accurate learning of pdes, Advances in Neural Information Processing Systems 36 (2023) 77187–77200.

[11] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, S. M. Benson, U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow, Advances in Water Resources 163 (2022) 104180.

[12] S. Sinha, B. Benton, P. Emami, On the effectiveness of neural operators at zero-shot weather downscaling, Environmental Data Science 4 (2025) e21.

[13] H. Wang, Y. Cao, Z. Huang, Y. Liu, P. Hu, X. Luo, Z. Song, W. Zhao, J. Liu, J. Sun, et al., Recent advances on machine learning for computational fluid dynamics: A survey, arXiv preprint arXiv:2408.12171 (2024).

[14] Q. Luo, W. Zeng, M. Chen, G. Peng, X. Yuan, Q. Yin, Self-attention and transformers: Driving the evolution of large language models, in: 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT), IEEE, 2023, pp. 401–405.

[15] Z. Zhao, X. Ding, B. A. Prakash, Pinnsformer: A transformer-based framework for physics-informed neural networks, arXiv preprint arXiv:2307.11833 (2023).

[16] C. Lorsung, Z. Li, A. Barati Farimani, Physics informed token transformer for solving partial differential equations, Machine Learning: Science and Technology 5 (1) (2024) 015032. doi:10.1088/2632-2153/ad27e3.
URL https://dx.doi.org/10.1088/2632-2153/ad27e3

[17] Y. Luo, Y. Chen, Z. Zhang, Cfdbench: A large-scale benchmark for machine learning methods in fluid dynamics, arXiv preprint arXiv:2310.05963 (2023).

[18] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics 378 (2019) 686–707.

[19] L. Lu, P. Jin, G. E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv preprint arXiv:1910.03193 (2019).

[20] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895 (2020).

[21] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, S. M. Benson, U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow, Advances in Water Resources 163 (2022) 104180. doi:https://doi.org/10.1016/j.advwatres.2022.104180.
URL https://www.sciencedirect.com/science/article/pii/S0309170822000562

[22] Q. Xu, N. Thuerey, Y. Shi, J. Bamber, C. Ouyang, X. X. Zhu, Physics-embedded fourier neural network for partial differential equations, arXiv preprint arXiv:2407.11158 (2024).

[23] Y. Li, L. Xu, S. Ying, Dwnn: Deep wavelet neural network for solving partial differential equations, Mathematics 10 (12) (2022). doi:10.3390/math10121976.
URL https://www.mdpi.com/2227-7390/10/12/1976

[24] J. Su, J. Ma, S. Tong, E. Xu, M. Chen, Multiscale attention wavelet neural operator for capturing steep trajectories in biochemical systems, Proceedings of the AAAI Conference on Artificial Intelligence 38 (13) (2024) 15100–15107. doi:10.1609/aaai.v38i13.29432.
URL https://ojs.aaai.org/index.php/AAAI/article/view/29432

[25] H. Wang, J. Pan, H. Wu, F. Zhang, T. Wu, Fourierflow: Frequency-aware flow matching for generative turbulence modeling, arXiv e-prints (2025) arXiv–2506.

[26] P. Hu, R. Wang, X. Zheng, T. Zhang, H. Feng, R. Feng, L. Wei, Y. Wang, Z.-M. Ma, T. Wu, Wavelet diffusion neural operator, in: The Thirteenth International Conference on Learning Representations, 2025.
URL https://openreview.net/forum?id=FQhDIGuaJ4

[27] T.-Y. Yang, J. Rosca, K. Narasimhan, P. J. Ramadge, Learning physics constrained dynamics using autoencoders, Advances in Neural Information Processing Systems 35 (2022) 17157–17172.

[28] G. Berend, Masked latent semantic modeling: an efficient pre-training alternative to masked language modeling, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 13949–13962.

[29] P. Garnier, V. Lannelongue, J. Viquerat, E. Hachem, Meshmask: Physics-based simulations with masked graph neural networks, arXiv preprint arXiv:2501.08738 (2025).

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[31] M. V. Koroteev, Bert: a review of applications in natural language processing and understanding, arXiv preprint arXiv:2103.11943 (2021).

[32] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, J. Zhu, Physics-informed machine learning: A survey on problems, methods and applications, arXiv preprint arXiv:2211.08064 (2022).

[33] A. Zhou, A. B. Farimani, Masked autoencoders are pde learners, arXiv preprint arXiv:2403.17728 (2024).

[34] M. Taghizadeh, M. A. Nabian, N. Alemazkoor, Multi-fidelity physics-informed generative adversarial network for solving partial differential equations, Journal of Computing and Information Science in Engineering 24 (11) (2024) 111003.

[35] H. Xue, A. Araujo, B. Hu, Y. Chen, Diffusion-based adversarial sample generation for improved stealthiness and controllability, Advances in Neural Information Processing Systems 36 (2023) 2894–2921.

Appendix A. Extended Proofs

Assumption Appendix A.1 (2-D DFT/IFFT normalization)**.** For $x \in \mathbb{C}^{H\times W}$, $\widehat{x}[k,\ell] = \sum_{m=0}^{H-1}\sum_{n=0}^{W-1} x[m,n]\, e^{-2\pi i(\frac{mk}{H}+\frac{n\ell}{W})}$ and $x[m,n] = \frac{1}{HW}\sum_{k,\ell}\widehat{x}[k,\ell]\, e^{2\pi i(\frac{mk}{H}+\frac{n\ell}{W})}$. For real x, $\widehat{x}$ is Hermitian; we use rFFT/irFFT storing the half-plane.

Proposition Appendix A.2 (Energy accounting with residual multiplier)**.** *For any filter bank $\{K_s\}$, $\sum_s \|x * K_s\|_2^2 = \frac{1}{HW}\sum_\omega \left(\sum_s |\widehat{K_s}(\omega)|^2\right)|\widehat{x}(\omega)|^2$. If $S(\omega) = \sum_s |\widehat{K_s}(\omega)|^2$, then $\left|\sum_s \|x * K_s\|_2^2 - \|x\|_2^2\right| \le \|S-1\|_\infty \|x\|_2^2$.*

Lemma Appendix A.3 (Partition of unity)**.** *Let $g_0(\omega) = \cos\theta(\omega)$ and $g_1(\omega) = \sin\theta(\omega)$ with even C^1 $\theta : [-\pi,\pi] \to [0,\pi/2]$, and define separable windows $\widehat{K}_{\mathrm{LL}} = g_0(\omega_x)g_0(\omega_y)$, $\widehat{K}_{\mathrm{LH}} = g_0(\omega_x)g_1(\omega_y)$, $\widehat{K}_{\mathrm{HL}} = g_1(\omega_x)g_0(\omega_y)$, $\widehat{K}_{\mathrm{HH}} = g_1(\omega_x)g_1(\omega_y)$. Then $\sum_{s\in\{\mathrm{LL,LH,HL,HH}\}} |\widehat{K_s}(\omega_x,\omega_y)|^2 \equiv 1$.*

Lemma Appendix A.4 (Discrete Green identity (periodic))**.** *For scalars f, g on the 2-D torus, $\sum f\,(\Delta g) = -\sum \nabla f \cdot \nabla g$ with central differences.*

Proposition Appendix A.5 (Quadratic boundary penalty enforces Dirichlet)**.** *Let $J_\mu(u) = \|Au - f\|_2^2 + \mu\|Bu - g\|_2^2$. Any minimizer u_μ converges, as $\mu \to \infty$, to the least-squares solution of $Au = f$ subject to $Bu = g$.*

Appendix A.1. Fourier branch: 1*-Lipschitz and isometry*

Proof of Theorem 3.1. Adopt Theorem Appendix A.1. Let $\widehat{x}(\omega) \in \mathbb{C}^C$ be the channel vector at frequency $\omega = (k,\ell)$. The layer acts per-mode as

$$\widehat{y}(\omega) = \begin{cases} W(\omega)\,\widehat{x}(\omega), & \omega \in \Omega_{\mathrm{keep}}, \\ 0, & \text{otherwise}, \end{cases} \qquad W(\omega)^\top W(\omega) = I.$$

By Parseval, $\|y\|_2^2 = \frac{1}{HW}\sum_\omega \|\widehat{y}(\omega)\|_2^2 = \frac{1}{HW}\sum_{\omega\in\Omega_{\mathrm{keep}}} \|W(\omega)\widehat{x}(\omega)\|_2^2 = \frac{1}{HW}\sum_{\omega\in\Omega_{\mathrm{keep}}} \|\widehat{x}(\omega)\|_2^2 \le \frac{1}{HW}\sum_\omega \|\widehat{x}(\omega)\|_2^2 = \|x\|_2^2$. Hence the operator norm is ≤ 1 (nonexpansive). If x is band-limited to Ω_{keep} then the inequality is an equality, i.e., an isometry. □

Appendix A.2. Residual energy accounting and tight frame

Proof of Theorem Appendix A.2. For any filter K_s, Parseval gives $\|x*K_s\|_2^2 = \frac{1}{HW}\sum_\omega |\widehat{K_s}(\omega)|^2|\widehat{x}(\omega)|^2$. Summing s yields the stated identity and

$$\left|\sum_s \|x * K_s\|_2^2 - \|x\|_2^2\right| = \frac{1}{HW}\sum_\omega \left|S(\omega) - 1\right|\,|\widehat{x}(\omega)|^2 \le \|S-1\|_\infty \|x\|_2^2.$$

□

Proof of Theorem 3.2. By Theorem Appendix A.3, $\sum_s |\widehat{K_s}|^2 \equiv 1$. Thus $\sum_s \|x * K_s\|_2^2 = \|x\|_2^2$ by Parseval. Let A be the analysis map $x \mapsto (K_s * x)_s$ and S the synthesis $S(y_s) = \sum_s K_s^\vee * y_s$. In the DFT basis, A^*A has symbol $\sum_s |\widehat{K_s}|^2 \equiv 1$, hence A is an isometry and $SA = I$; i.e., $x = \sum_s K_s^\vee * (K_s * x)$. Both analysis and synthesis are 1-Lipschitz. □

Appendix A.3. Hybrid fusion nonexpansiveness

Proof of Theorem 3.3. Let $\mathcal{F}, \mathcal{W}$ satisfy $\|\mathcal{F}\| \le 1$, $\|\mathcal{W}\| \le 1$ and $G_\alpha = \alpha\mathcal{F} + (1-\alpha)\mathcal{W}$ with $\alpha \in [0,1]$ (pointwise or spatially varying). For any x, y,

$$\|G_\alpha x - G_\alpha y\| \le \alpha\|\mathcal{F}(x-y)\| + (1-\alpha)\|\mathcal{W}(x-y)\| \le \alpha\|x-y\| + (1-\alpha)\|x-y\| = \|x-y\|.$$

If both branches are isometries on the relevant subspace (e.g., band-limited input and $\mathcal{W} = I$), then $\|G_\alpha x\| = \|x\|$. □

Appendix A.4. Biased attention: ratio and Lipschitz bounds

Proof of Theorem 3.4. For a fixed row i, $\alpha_{ij} = \exp\big(\tilde{L}_{ij}\big) / \sum_m \exp\big(\tilde{L}_{im}\big)$ with $\tilde{L}_{ij} = L_{ij} - \lambda_{\text{att}} R_{ij}$. Then

$$\frac{\alpha_{ij_1}}{\alpha_{ij_2}} = \exp\big(\tilde{L}_{ij_1} - \tilde{L}_{ij_2}\big) = \exp\big((L_{ij_1} - L_{ij_2}) - \lambda_{\text{att}}(R_{ij_1} - R_{ij_2})\big),$$

which is strictly decreasing in λ_{att} whenever $R_{ij_1} > R_{ij_2}$. □

Proof of Theorem 3.5. Let $\alpha(\lambda) = \text{softmax}(z - \lambda r)$ with row vectors z, r. The Jacobian of softmax at u is $J(u) = \text{Diag}(\sigma(u)) - \sigma(u)\sigma(u)^\top$. By the mean value theorem, $\alpha(\lambda) - \alpha(0) = \int_0^\lambda J(z - tr)(-r)\, dt$. Using the operator norm $\|\cdot\|_{\infty\to 1}$, $\|\alpha(\lambda) - \alpha(0)\|_1 \le \int_0^\lambda \|J(\cdot)\|_{\infty\to 1}\, dt\, \|r\|_\infty$. One checks (e.g., by column sums) $\|J(\cdot)\|_{\infty\to 1} = \max_j 2\sigma_j(1-\sigma_j) \le \frac{1}{2}$, hence $\|\alpha(\lambda) - \alpha(0)\|_1 \le \frac{\lambda}{2}\|r\|_\infty$. Apply rowwise with $r = R_{i\cdot}$ and $\lambda = \lambda_{\text{att}}$. □

Appendix A.5. Translation equivariance and continuum limit

Proof of Theorem 3.6. Let τ_s be the lattice shift by s. If $R_{ij} = \rho(p, r(i) - r(j))$, then L_{ij} and R_{ij} shift compatibly: $L \circ \tau_s = \Pi_s L \Pi_s^\top$ and likewise for R, with permutation matrix Π_s. Rowwise softmax commutes with the same permutation, so $\alpha(\tau_s x) = \Pi_s \alpha(x) \Pi_s^\top$. Hence the mapping is translation-equivariant. □

Proof of Theorem 3.7. Assume a periodic, compact Ω and bounded continuous $q(\cdot), k(\cdot), r(\cdot,\cdot)$. On a grid of spacing h, the row i softmax weights are $w_h(x_i, y_j) = \frac{\exp(\langle q(x_i), k(y_j)\rangle - \lambda r(x_i, y_j))}{\sum_m \exp(\langle q(x_i), k(y_m)\rangle - \lambda r(x_i, y_m))}$. Then $(T_h v)(x_i) = \sum_j w_h(x_i, y_j) v(y_j)\, h^d$ is a Riemann sum for $(Tv)(x) = \int_\Omega w_\lambda(x, y) v(y)\, dy$ with the same normalized exponential kernel. Uniform boundedness and continuity yield uniform convergence by dominated convergence; the normalization enforces $\int w_\lambda(x, y)\, dy = 1$. □

Appendix A.6. Discrete Green identity and quadratic penalty

Proof of Theorem Appendix A.4. For periodic central differences, $D_x^\top = -D_x$ and $D_y^\top = -D_y$. With $\Delta = -(D_x^\top D_x + D_y^\top D_y)$,

$$\sum f\,(\Delta g) = -\sum f\, D_x^\top D_x g - \sum f\, D_y^\top D_y g = -\sum (D_x f)\,(D_x g) - \sum (D_y f)\,(D_y g).$$

□

Proof of Theorem Appendix A.5. The minimizer u_μ satisfies the normal equations $(A^\top A + \mu B^\top B) u_\mu = A^\top f + \mu B^\top g$. If $u^\star$ solves $Au = f$ with $Bu = g$ (in the least-squares sense), then $u_\mu \to u^\star$ as $\mu \to \infty$ by standard quadratic-penalty arguments: $Bu_\mu \to g$ and $Au_\mu \to f$; any limit point solves the constrained problem. □

Appendix A.7. Divergence-aware oracle inequality

Proof of Theorem 3.9. Define $\mathcal{R}_\lambda(g) = \mathbb{E}\|x - g\|_2^2 + \lambda\mathbb{E}\|Dg\|_2^2$ and let g_λ be a minimizer. For any $g^\star \in \ker D$, $\mathbb{E}\|x - g_\lambda\|^2 + \lambda\mathbb{E}\|Dg_\lambda\|^2 \le \mathbb{E}\|x - g^\star\|^2$. Rearrange to obtain $\mathbb{E}\|x - g_\lambda\|^2 \le \mathbb{E}\|x - g^\star\|^2 - \lambda\mathbb{E}\|Dg_\lambda\|^2$. Now, by $\operatorname{dist}(u, \ker D) \le c_H\|Du\|_2$, take $u = g_\lambda(\tilde{x})$ pointwise and average to get $\mathbb{E}\operatorname{dist}(g_\lambda, \ker D)^2 \le c_H^2\mathbb{E}\|Dg_\lambda\|^2$. Using $\operatorname{dist}(a, \mathcal{H})^2 \le \|a - b\|^2$ with $b \in \mathcal{H}$ and choosing $b = g^\star$ yields

$$\mathbb{E}\|x - g_\lambda\|^2 \le \mathbb{E}\|x - g^\star\|^2 + \frac{c_H^2}{\lambda}\,\mathbb{E}\|Dg_\lambda\|^2,$$

which matches the stated bound. □

Appendix B. Fourier and Wavelet Derivatives (Implementation Notes)

Fourier branch.. Let $y = \mathcal{F}^{-1}\big(\widehat{y}\big)$ with $\widehat{y}(\omega) = W(\omega)\,\widehat{x}(\omega)$ on the kept half-plane and $\widehat{y}(\omega) = 0$ otherwise. For a real x we store the rFFT half-plane and enforce Hermitian symmetry.

Given an upstream spatial gradient $g = \partial\mathcal{L}/\partial y$ and its DFT $\widehat{g} = \mathcal{F}(g)$:

$$\frac{\partial\mathcal{L}}{\partial\widehat{x}(\omega)} = W(\omega)^{\mathsf{H}}\,\widehat{g}(\omega), \qquad \frac{\partial\mathcal{L}}{\partial W(\omega)} = \widehat{g}(\omega)\,\widehat{x}(\omega)^{\mathsf{H}}, \quad \omega \in \Omega_{\text{keep}}.$$

The spatial gradient is $\partial\mathcal{L}/\partial x = \mathcal{F}^{-1}\big(\partial\mathcal{L}/\partial\widehat{x}\big)$. For rFFT, mirror the gradients to satisfy Hermitian symmetry and zero out non-kept modes.

Tight-frame branch.. Analysis coefficients $z_s = K_s * x$ and (optionally) synthesis $\tilde{x} = \sum_s K_s^\vee * z_s$. For any branch loss $\mathcal{L}(z_s)$ with upstream gradients $h_s = \partial\mathcal{L}/\partial z_s$:

$$\frac{\partial\mathcal{L}}{\partial x} = \sum_s K_s^\vee * h_s, \qquad \frac{\partial\mathcal{L}}{\partial K_s} = h_s * x^\vee.$$

In the Fourier domain:

$$\frac{\partial\mathcal{L}}{\partial\widehat{x}} = \sum_s \overline{\widehat{K_s}} \odot \widehat{h}_s, \qquad \frac{\partial\mathcal{L}}{\partial\widehat{K_s}} = \overline{\widehat{x}} \odot \widehat{h}_s.$$

Wavelet windows (learnable).. With $\widehat{K}_{\text{LL}} = g_0(\omega_x)g_0(\omega_y)$, $\widehat{K}_{\text{LH}} = g_0(\omega_x)g_1(\omega_y)$, $\widehat{K}_{\text{HL}} = g_1(\omega_x)g_0(\omega_y)$, $\widehat{K}_{\text{H}} = g_1(\omega_x)g_1(\omega_y)$, $g_0 = \cos\theta$, $g_1 = \sin\theta$:

$$\frac{\partial g_0}{\partial\theta} = -\sin\theta, \qquad \frac{\partial g_1}{\partial\theta} = \cos\theta, \qquad \frac{\partial\widehat{K}_{\text{LL}}}{\partial\theta_x} = \big(-\sin\theta_x\big)g_0(\omega_y), \ \ldots$$

Chain with $\partial\theta/\partial\phi$ if $\theta(\omega;\phi)$ is parametrized. To preserve the partition-of-unity $\sum_s |\widehat{K}_s|^2 \equiv 1$, either (i) parametrize via a single angle field $\theta(\omega)$ as above, or (ii) renormalize K_s by $S(\omega)^{-1/2}$ with $S = \sum_s |\widehat{K}_s|^2$ after each update.

Gating and fusion.. For $E = \alpha_F F + (1 - \alpha_F)W$ with scalar or spatially varying $\alpha_F = \operatorname{softmax}(\gamma_F, \gamma_W)$, the gradients are

$$\frac{\partial\mathcal{L}}{\partial F} = \alpha_F\,\frac{\partial\mathcal{L}}{\partial E}, \quad \frac{\partial\mathcal{L}}{\partial W} = (1 - \alpha_F)\,\frac{\partial\mathcal{L}}{\partial E}, \quad \frac{\partial\mathcal{L}}{\partial\gamma_F} = \alpha_F(1 - \alpha_F)\left\langle\frac{\partial\mathcal{L}}{\partial E},\, F - W\right\rangle.$$

Notes on rFFT bookkeeping.. (i) Handle DC/NYQUIST lines once (no mirroring). (ii) When enforcing column-unitarity on $W(\omega)$, apply it only on the kept set; set others to zero. (iii) Gradients on mirrored bins must be conjugate.

Appendix C. Hyperparameters and Training Details

Hardware/precision.. We trained and evaluated all models with mixed precision (fp16 AMP). Most runs used the Apple M-series GPU (MPS backend). To test portability and reproducibility, we repeated key runs on two tiers: a consumer **RTX/GTX 3060** workstation and a datacenter **A100** node. With fixed seeds and identical software, metrics matched up to floating-point noise and qualitative plots were indistinguishable. Per-case scripts and checkpoints are provided to facilitate third-party reproduction.

Table C.9: Hyperparameters used for the supervised `cfdb_cylinder_npz` runs with parameter count. All models: 2000 epochs, batch 8 (micro-batch 2), Adam lr 3×10^{-4} with cosine decay to 10^{-5} and 50 warmup epochs, weight decay 10^{-4}, AMP fp16, device=MPS; c_{in}=2, c_{out}=2.

Model	**Hyperparameters**	**Value**	**Model Parameters**
PINN	hidden size	64	**0.013M**
	hidden layers	5	
FNO2d	width	32	**0.529M**
	Fourier modes	8	
DeepONet2d	branch width	128	**0.052M**
	trunk width	128	
	layers (branch/trunk)	3	
	basis functions	48	
PIBERT	d (embed size)	128	**9.553M**
	depth (encoder layers)	4	
	heads	4	
	MLP (FFN size)	512	
	Fourier modes	16	
	patch size	4	
PIBERT_FNO	d (embed size)	64	**0.959M**
	depth (encoder layers)	2	
	heads	4	
	MLP (FFN size)	256	
	Fourier modes (PIBERT)	8	
	FNO width	28	
	FNO modes	8	
	patch size	4	
PIBERT_DeepONet2d	d (embed size)	64	**0.619M**
	depth (encoder layers)	2	
	heads	4	
	MLP (FFN size)	256	
	Fourier modes (PIBERT)	8	
	DeepONet branch	128	
	layers (DeepONet)	3	

Notes: Parameter counts are those reported alongside cylinder validation metrics (median over two seeds with 95% bootstrap CIs).

Table C.10: Model hyperparameters for the main results (identical across Cylinder, Tube, and Cavity). Parameter counts reflect trainable weights for the implementations used in this paper.

Model	Hyperparameters	Value	Model Parameters
PINN	hidden layers	6	0.017M
	hidden size	64	
PIBERT	d (embed size)	96	3.321M
	depth (encoder layers)	4	
	heads	4	
	MLP (FFN size)	384	
	Fourier modes	12	
	patch size	4	

Table C.11: Ablation hyperparameters for PIBERT used in the Appendix plots (larger configuration). Counts reflect trainable parameters with d=256, depth= 4, heads= 4, MLP= 768, Fourier modes= 20, C=3, and d_{prm}=16.

Variant	Hyperparameters	Value	Model Parameters
PIBERT-Full	channels (C)	3	**3.388M**
	image size (H=W)	64	
	parameter dim (d_{prm})	16	
	embedding size (d)	**256**	
	encoder depth	**4**	
	heads	**4**	
	FFN size (MLP)	**768**	
	spectral embed	Fourier (#modes=**20**) & Wavelet	
Fourier-only	spectral embed	Fourier only (#modes=**20**)	**3.388M**
	adapters	**on** (d_{adapter}=32)	
	other hparams	same as Full	
Wavelet-only	spectral embed	Wavelet only	**3.388M**
	adapters	**on** (d_{adapter}=32)	
	other hparams	same as Full	
No-adapters	adapters	**off**	**3.322M**
	spectral embed	Fourier(#modes=**20**) & Wavelet	
	other hparams	same as Full	

Notes: Frozen depthwise wavelet filters (3×3, groups=C) are not counted. The trainable counts include: 1×1 wavelet projection, Fourier spectral weights, fusion 1×1, parameter MLP, Transformer encoder layers, adapter MLPs (when enabled), output head, and the two gating scalars `g_ff`, `g_wv`. "Fourier-only"/"Wavelet-only" are realized by gating one branch near zero so parameterization stays aligned across ablations.

Shared settings: two output channels $[u, v]$; viscosity $\nu = 10^{-3}$; loss weights w_{div}=w_{vort}=1.0.

Appendix D. PDE Setups and Data Generation

We list the governing equations, domains, and boundary/initial data for the three 2-D benchmark cases used in this work (Cylinder wake, Laminar tube, Lid-driven cavity). All cases are treated in a dimensionless form with kinematic viscosity $\nu = 10^{-3}$ (the same ν is also used inside our physics loss).

Governing equations (all cases).. On a rectangular spatial domain $\Omega \subset \mathbb{R}^2$ and time interval $[0, T]$, the incompressible Navier–Stokes system reads

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} \;=\; -\nabla p + \nu\,\Delta\mathbf{u}, \qquad \nabla\cdot\mathbf{u} \;=\; 0, \tag{D.1}$$

with velocity $\mathbf{u}(x, y, t) = (u, v)$ and kinematic viscosity $\nu = 10^{-3}$. For diagnostics and the physics term in training, we also evaluate *spatial* residuals snapshotwise:

$$\text{divergence:} \quad r_{\text{div}} \;=\; \nabla\cdot\mathbf{u}, \tag{D.2}$$

$$\text{vorticity transport (steady form):} \quad \omega \;=\; \partial_x v - \partial_y u, \quad r_\omega \;=\; \mathbf{u}\cdot\nabla\omega - \nu\Delta\omega, \tag{D.3}$$

i.e., we drop $\partial_t\omega$ when forming r_ω so that the physics term constrains spatial consistency (no temporal discretization is required). These residuals are exactly the ones used in our implementation.

Case A: Cylinder wake

Domain and geometry. A 2-D channel $\Omega = (0, L_x) \times (0, L_y)$ with a circular obstacle (cylinder) of radius R centered at (x_c, y_c).
Boundary conditions.

$$\mathbf{u}|_{\Gamma_{\text{cyl}}} = (0, 0) \quad \text{(no-slip on the cylinder)}, \tag{D.4}$$

$$\mathbf{u}|_{\Gamma_{\text{walls}}} = (0, 0) \quad \text{(no-slip on top/bottom walls)}, \tag{D.5}$$

$$\mathbf{u}|_{\Gamma_{\text{in}}} = (U_{\text{in}},\, 0) \quad \text{(uniform inflow)}, \tag{D.6}$$

$$\partial_x \mathbf{u}|_{\Gamma_{\text{out}}} = \mathbf{0}, \qquad p|_{\Gamma_{\text{out}}} = 0 \quad \text{(convective/Neumann-type outlet with reference pressure)}. \tag{D.7}$$

Initial condition. Either quiescent flow or a developed snapshot; the dataset provides time-ordered snapshots and we sample them directly.

Case B: Laminar tube (channel)

Domain. Rectangular $\Omega = (0, L_x) \times (0, L_y)$ with parallel walls at $y = 0$ and $y = L_y$.
Boundary conditions.

$$\mathbf{u}|_{y=0} = (0, 0), \qquad \mathbf{u}|_{y=L_y} = (0, 0) \quad \text{(no-slip walls)}, \tag{D.8}$$

$$\mathbf{u}|_{\Gamma_{\text{in}}} = \left(4U_{\max}\tfrac{y(L_y-y)}{L_y^2},\, 0\right) \quad \text{(parabolic Poiseuille inflow)}, \tag{D.9}$$

$$\partial_x \mathbf{u}|_{\Gamma_{\text{out}}} = \mathbf{0}, \qquad p|_{\Gamma_{\text{out}}} = 0. \tag{D.10}$$

Initial condition. Parabolic profile or a provided snapshot from the dataset.

Case C: Lid-driven cavity

Domain. Unit square $\Omega = (0,1) \times (0,1)$.

Boundary conditions.

$$\mathbf{u}|_{y=1} = (U_{\text{lid}}, 0) \quad \text{(moving lid)}, \tag{D.11}$$

$$\mathbf{u}|_{y=0} = (0,0), \quad \mathbf{u}|_{x=0} = (0,0), \quad \mathbf{u}|_{x=1} = (0,0) \quad \text{(no-slip on the other three walls)}. \tag{D.12}$$

Pressure is defined up to an additive constant; a reference is fixed implicitly by the outlet-free enclosure.

Datasets and splits.. We use the CFDBench NPZ distributions for cylinder, tube, dam, and cavity, which contain (u, v, p) snapshots stored as `u.npy`, `v.npy`, `p.npy`. Each snapshot may include a time index t. Coordinates (x, y, t) are concatenated with conditioning channels as inputs to all models, consistent with the main text. We adopt case-level splits of approximately 80%/10%/10% for train/validation/test (randomized by a fixed seed). Channel-wise normalization (mean/std) is computed on the training set and applied to validation/test.

Evaluation metrics.. We report relative errors and normalized MSE (NMSE equals the squared relative ℓ_2). For predictions $\hat{u}$ and ground truth u on N grid points,

$$\text{rMAE} = \frac{\sum_{n=1}^{N} |\hat{u}_n - u_n|}{\sum_{n=1}^{N} |u_n|}, \qquad \text{rRMSE} = \sqrt{\frac{\sum_{n=1}^{N} \|\hat{u}_n - u_n\|_2^2}{\sum_{n=1}^{N} \|u_n\|_2^2}}, \qquad \text{NMSE} = \frac{\sum_{n=1}^{N} \|\hat{u}_n - u_n\|_2^2}{\sum_{n=1}^{N} \|u_n\|_2^2}.$$

Unless stated otherwise, summary metrics are computed on (u, v); pressure is visualized and compared up to a constant.