

481 A Symbol Definitions

482 Table 5 summarizes the key symbols and notation used throughout the paper for clarity and reference.

Table 5: Definitions of symbols.

Symbol	Description
Input and function space variables	
\mathcal{X}	Input coordinate (spatial and/or temporal position).
x, t	Spatial and temporal components of the input coordinate X .
$f_a(\mathcal{X}; \theta_a)$	INR modeling a function in space \mathcal{A} (with parameters θ_a).
$f_u(\mathcal{X}; \theta_u)$	INR modeling a function in space \mathcal{U} (with parameters θ_u).
\mathcal{A}, \mathcal{U}	Input and output function spaces
$\mathcal{G}, \mathcal{G}^\dagger$	Function space mapping from $\mathcal{A} \rightarrow \mathcal{U}$ and $\mathcal{U} \rightarrow \mathcal{A}$
Neural network architecture parameters	
θ	Set of network parameters (weights and biases of an INR).
θ_a, θ_u	Network parameters for f_a and f_u , respectively.
L	Number of hidden layers in the INR.
h_0	Initial input to the network.
h_k	Hidden feature vector at layer k (h_0 is input, h_{L+1} is final hidden layer output).
W^k	Weight matrix of layer k in the network.
b^k	Bias vector of layer k in the network.
γ	Nonlinear activation function applied to hidden layers.
d	Dimension of each hidden layer (length of h_k and b_k).
Fourier reparameterization and spectral terms	
S^k	Trainable coefficient matrix for layer k under Fourier reparameterization.
Φ_k	Fixed Fourier basis for layer k .
p_m	m -th point in the uniform spatial grid used to construct Φ_k .
T_{\max}	Maximum coordinate value for the Fourier basis sampling interval.
$\omega = \{\omega_{low}, \omega_{high}\}$	Frequency component used in constructing the Fourier basis.
q	Phase shift applied in the Fourier basis construction.
D	Number of Fourier basis functions per layer.
M	Number of sample points per basis function.
F_1, F_2	High and low reference frequencies ($F_1 > F_2 > 0$) used in Theorem 1.
ϵ	Small positive constant (error tolerance in Theorem 1).
Modulation and training variables	
R^k	Shared base coefficient matrix for layer k .
α^k	Modulation vector for layer k
$\alpha^{k,W}, \alpha^{k,b}$	Components of α^k that modulate the weight and bias at layer k , respectively.
$\psi_k(\cdot)$	Modulation function applied at layer k
z	Latent modulation vector (sample-specific)
$g_a(z; \pi_a)$	Mapping for function space \mathcal{A} : generates $\{\alpha_k^a\}_{k=1}^L$ from latent z .
$g_u(z; \pi_u)$	Mapping for function space \mathcal{U} : generates $\{\alpha_k^u\}_{k=1}^L$ from latent z .
π_a, π_u	Learnable parameters of the modulation networks g_a and g_u .
$\eta_{inner}, \eta_{outer}$	Learning rates for inner/outer loop
\mathbb{L}	Training loss function.
PDE-specific parameters (experiment settings)	
β	PDE coefficient in the 1D convection equation.
a_1, a_2	PDE coefficients in the 2D Helmholtz equations.

483

484 B Limitation and Broader Impact

485 Our proposed GFM method allows explicit control over the ratio of high-frequency and low-frequency
486 components in the Fourier basis during weight construction. While optimal performance may require
487 careful hyperparameter tuning to select the appropriate frequency ratio, this flexibility enables our
488 framework to be adapted to a wide range of scientific problems and data characteristics. Additionally,
489 our proposed methods, GFM and PDEfunct, enable efficient compression and representation of
490 multiple PDE solution datasets, offering significant advantages in terms of memory efficiency. These
491 methods can facilitate effective storage and transmission of scientific data, potentially making large-
492 scale modeling and simulation more accessible across various domains.

493 C Comparison with Existing Neural PDE Solvers

494 PDEfunct demonstrates clear advantages over traditional Physics-Informed Neural Networks
495 (PINNs) and Neural Operators. (1) PINNs, while effective in integrating physical laws directly
496 into training, often struggle with scalability and generalization to complex, high-frequency PDE
497 solutions. (2) Neural Operators, including methods like FNO and DeepONet, address scalability
498 but typically require consistent discretizations and are limited to one-way function space mappings.
499 PDEfunct overcomes these challenges by leveraging Global Fourier Modulation, providing a com-
500 pact, efficient, and spectrally-aware representation capable of accurate forward and inverse mappings
501 without retraining. Its shared latent modulation vector further enhances generalization across varied
502 parameter spaces, demonstrating clear advantages in efficiency, generalization, and bidirectional
503 inference capabilities compared to both PINN and Neural Operator frameworks.

504 D Dataset Description

505 We provide detailed descriptions of all benchmark PDE datasets used in the main paper, including the
506 governing equations, boundary/initial conditions, parameter ranges, and scientific context for each.

507 D.1 Convection Equation

508 The convection dataset is based on the one-dimensional convection equation with a controllable
509 advection speed parameter β . The PDE governs the evolution of a field $u(x, t)$ as:

$$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} = 0, \quad x \in \Omega, \quad t \in [0, T], \quad (13)$$

510 where β is the convection coefficient. In our experiments, β varies within a range (e.g. $\beta \in [1, 50]$) to
511 represent solution fields of different characteristic speeds. We impose periodic boundary conditions
512 on the spatial domain Ω . An initial condition $u(x, 0)$ is $1 + \sin(x)$. It is a fundamental model for
513 wave propagation and convective transport.

514 D.2 Helmholtz Equation

515 The Helmholtz dataset involves a steady-state wave equation in two spatial dimensions. We generate
516 solutions of the form $u(x, y)$ by choosing forcing functions that yield analytic solutions. The PDE is
517 given by:

$$\begin{aligned} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} + k^2 u(x, y) - q(x, y) &= 0, \\ q(x, y) &= (-(a_1 \pi)^2 - (a_2 \pi)^2 + k^2) \sin(a_1 \pi x) \sin(a_2 \pi y), \end{aligned} \quad (14)$$

$$u(x, y) = k^2 \sin(a_1 \pi x) \sin(a_2 \pi y), \quad (15)$$

519 where k is the wavenumber and $q(x, y)$ is a source term. The coefficients a_1 and a_2 are PDE
520 coefficients that control the number of oscillations of the solution in the x and y directions, respectively.
521 In our dataset, we consider two difficulty settings: Helmholtz #1 with $\{a_1, a_2\} \in \{1, \dots, 5\}$ (limited
522 frequencies), and Helmholtz #2 with $\{a_1, a_2\} \in \{1, \dots, 10\}$.

523 D.3 Kuramoto-Sivashinsky Equation

524 The one-dimensional Kuramoto–Sivashinsky equation [43] is a prototypical model for spatio-temporal
525 chaos and pattern formation.

$$\begin{aligned} u_t + uu_x + u_{xx} + \nu u_{xxxx} &= 0, \quad (t, x) \in [0, T] \times [0, L], \\ u(0, x) &= u_0(x), \quad u_0(x) = \sum_{i=0}^{20} A_i \sin((2\pi k_i x/L) + \phi_i), \end{aligned} \quad (16)$$

526 where $u(x, t)$ is the evolving scalar field, and $\nu > 0$ is a viscosity (or hyperviscosity) parameter. In
527 our simulations we take periodic boundary conditions on the domain $[0, L]$ (so that derivatives in x
528 are periodic), which is natural given that initial conditions are composed of Fourier modes.

529 D.4 Navier-Stokes Equation

530 We use the 2D Navier–stokes equation data employed in the neural operator related studies [20, 21, 44].
531 This equation represents the dynamics of incompressible fluid within a rectangular domain and is
532 expressed as follows:

$$\frac{\partial w(x, t)}{\partial t} = -u(x, t) \cdot \nabla w(x, t) + \nu \Delta w(x, t) + f, \quad x \in (0, 1)^2, t \in (0, T] \quad (17)$$

$$w(x, t) = \nabla \times u(x, t), \quad x \in (0, 1)^2, t \in (0, T] \quad (18)$$

$$\nabla \cdot u(x, t) = 0, \quad x \in (0, 1)^2, t \in (0, T] \quad (19)$$

533 where u is the velocity field and w is the vorticity. ν is the viscosity and f is a forcing term. The
534 initial condition $\omega_0(x)$ is generated according to $\omega_0 \sim \mu$ where $\mu = \mathcal{N}(0, 7^{3/2}(-\Delta + 49I)^{-2.5})$.

535 We consider the spatial domain $x \in (0, 1)^2$ with the following boundary condition:

$$f(x_1, x_2) = 0.1(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2))). \quad (20)$$

536 D.5 Euler’s Equation (Airfoil)

537 In our study, we employ the Euler’s equation (airflow), which is a pivotal component of the discussed
538 benchmark data in [34, 22]. The governing equation is as follows:

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{v}) = 0, \quad \frac{\partial \rho_f \mathbf{v}}{\partial t} + \nabla \cdot (\rho_f \mathbf{v} \otimes \mathbf{v} + p \mathbb{I}) = 0, \quad \frac{\partial E}{\partial t} + \nabla \cdot ((E + p) \mathbf{v}) = 0, \quad (21)$$

539 where ρ_f , p , \mathbf{v} and E are the fluid density, the pressure, the velocity vector and the total energy
540 respectively. The viscous effect is neglected. The far-field boundary condition is specified as $\rho_\infty = 1$,
541 $p_\infty = 1.0$, $M_\infty = 0.8$, $AoA = 0$, where M_∞ represents the Mach number and AoA denotes the
542 angle of attack. In our experimentation, we use a total of 1,000 training data samples and 200 test
543 data samples. The computational grid utilized is the C-grid mesh with about 200×50 quadrilateral
544 elements.

545 D.6 Pipe

546 The pipe dataset [35] is derived from simulations of viscous incompressible flow in a pipe, governed
547 by the Navier–Stokes equations for velocity and pressure.

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} &= -\nabla p + \nu \nabla^2 \mathbf{v}, \\ \nabla \cdot \mathbf{v} &= 0 \end{aligned} \quad (22)$$

548 where \mathbf{v} is the velocity field, p is the pressure, and ν is the kinematic viscosity. The domain is a
549 cylindrical pipe, a long tube with circular cross section, though our dataset is represented on a mesh
550 (in the format of a 129×129 grid for a cross-sectional plane, as indicated by the input size).

551 D.7 Seismic Wave Equation

552 The OpenFWI dataset [10] is a collection of seismic forward modeling simulations used for full-
 553 waveform inversion research. Each data sample consists of a subsurface velocity model (a spatial map
 554 of wave propagation speeds) and simulated seismic wavefields recorded from sources propagating
 555 through that medium.

$$\nabla^2 p - \frac{1}{v^2} \frac{\partial^2 p}{\partial t^2} = s, \quad (23)$$

556 where $p(x, z, t)$ is the pressure wavefield (or particle displacement) at spatial location (x, z) and time
 557 t , and $v(x, z)$ is the velocity model (spatially-varying wave speed). The term $s(x, z, t)$ represents the
 558 seismic source (for example, a Ricker wavelet point source on or near the surface). In our dataset, Ω
 559 is a 2D slice of the Earth’s subsurface (with x horizontal and z depth).

560 E Detailed experimental setup

561 This section provides the detailed experimental setup, including hyperparameter choices and compu-
 562 tational cost measurements. Experiments are conducted on a system running UBUNTU 18.04 LTS,
 563 PYTHON 3.9.7, PYTORCH 1.13.0, CUDA 11.6, i9 CPU, and NVIDIA RTX A5000.

564 E.1 Hyperparameters

565 For all experiments, we fixed the backbone INR to SIREN in order to isolate and compare the effects
 566 of different modulation methods. Across all models and experiments, we set $\eta_{inner} = 0.01$ and η_{outer}
 567 $= 0.0001$. All networks use a total of 5 layers, each with hidden dimension $M = 256$. The modulation
 568 mapping is implemented as a two-layer MLP with hidden dimension 512, and the dimension of the
 569 latent code z is fixed to 20 for all settings. For GFM, additional hyperparameter settings are required
 570 to construct the Fourier basis (cf. Appendix H). Specifically, the number of phase shifts (n_{phase}) is
 571 fixed to 32 for all datasets, while the numbers of high-frequency and low-frequency components
 572 (n_{high}, n_{low}) are set as follows: 128 and 32 for Convection and FWI, 128 and 128 for Helmholtz, 64
 573 and 16 for Kuramoto-Sivashinsky, and 8 and 128 for Navier-Stokes. The batch size and the number of
 574 training epochs for each dataset are as follows: batch size 32 and epoch 1,000 for Convection, batch
 575 size 16 and epoch 5,000 for Helmholtz, batch size 32 and epoch 5,000 for Kuramoto-Sivashinsky,
 576 batch size 10 and epoch 1,000 for Navier-Stokes, and batch size 16 and epoch 10,000 for FWI.

577 E.2 Computational cost

578 To assess the computational requirements of different
 579 modulation methods, we report peak GPU memory
 580 usage and average training time per epoch for each
 581 model, measured on a single NVIDIA A5000 GPU.
 582 As a representative case, we provide results for the
 583 Helmholtz#2 setting under the single-INR modula-
 584 tion configuration (cf. Section 5.2 and Appendix J).
 585 Results are summarized in Table 6. GFM achieves a
 586 strong trade-off between model complexity and com-
 587 putational efficiency, with memory consumption and training speed close to Shift, yet substantially
 588 better than Scale, FiLM, and SpatialFuncna. Notably, SpatialFuncna exhibits a much higher training
 589 time per epoch (0.59s), which is due to the extra computational cost of applying modulation at every
 590 spatial location, rather than globally, despite moderate memory usage. Other experimental settings
 591 showed similar trends in both memory and runtime.

Table 6: Computational cost

Method	Memory (GB)	Time (sec)
Shift	8.15	0.20
Scale	12.15	0.24
FiLM	12.15	0.29
SpatialFuncna	8.83	0.59
GFM (Ours)	8.28	0.27

F More Experimental Results on Experiments with Single-INR modulation

This section provides additional analyses and experimental results to supplement Section 5.2. We present training loss curves for every dataset, a detailed evaluation of the latent modulation space through interpolation analyses, and statistical comparisons across all benchmarks.

F.1 Training loss curves

Figure 8 presents the training loss (MSE) curves for Shift, Scale, FiLM, and GFM across all PDE benchmarks considered in Section 5.2. As discussed in Section 5.2, GFM consistently achieves faster and more stable convergence compared to other modulation strategies. This advantage is especially notable for challenging cases such as Helmholtz#2 and Kuramoto-Sivashinsky, where baselines exhibit slow or unstable training. These results further demonstrate the effectiveness of GFM in robust optimization and in mitigating the spectral bias typically observed in modulated INRs.

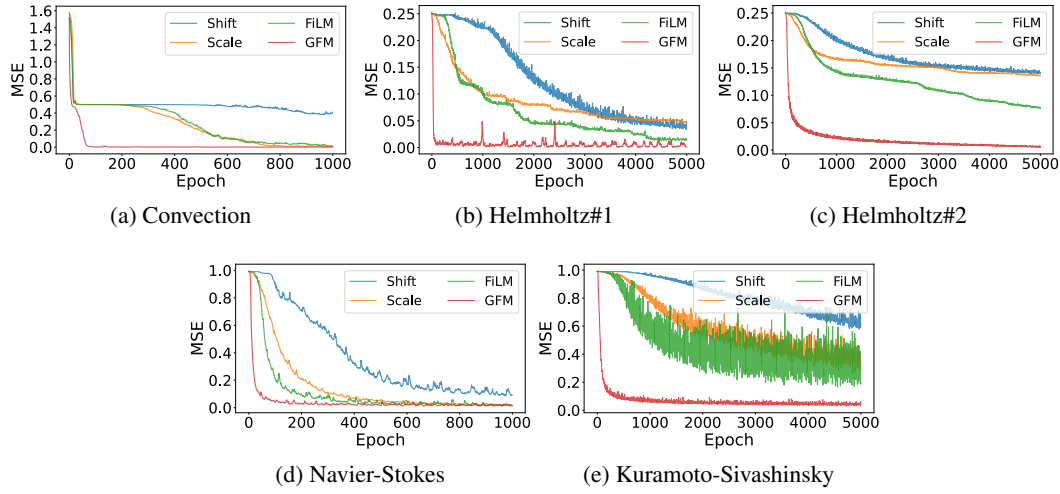


Figure 8: Training loss (MSE) curves for Shift, Scale, FiLM, and GFM on (a) Convection, (b) Helmholtz#1, (c) Helmholtz#2, (d) Navier-Stokes, and (e) Kuramoto-Sivashinsky. GFM consistently achieves faster and more stable convergence across all benchmarks.

F.2 Additional Analysis of Latent Modulation Space

To further analyze the properties of the learned latent modulation space, we present additional qualitative and quantitative results for the generalization experiments described in Section 5.2.1. Specifically, we visualize the dynamics of the interpolated latents for the convection equation and compare interpolation results across different modulation strategies.

Figure 9 shows the cubic interpolation trajectories of the latent code z for the convection equation. Each subplot corresponds to one dimension of the 20-dimensional latent modulation vector. Blue dots indicate latents obtained from the seen (training) coefficients, while red crosses denote interpolated latents for unseen test coefficients, estimated via cubic interpolation. This visualization suggests that the latent space encodes smooth and coherent functional variations as the PDE parameter is varied. The interpolated latents for unseen coefficients generally follow the manifold traced by the seen points, indicating that the latent space may provide a functionally meaningful representation aligned with the underlying parametric dependence of the PDE family.

To further investigate the effect of latent interpolation across different modulation strategies, Figure 10 presents the reconstruction results for an unseen coefficient ($\beta = 30.5$) obtained via cubic interpolation of the latent code for each modulation method (Shift, Scale, FiLM, and GFM). The ground truth solution is shown in Figure 10(a), and Figures 10(b)–(e) show the corresponding reconstructions. These results highlight that only GFM produces an accurate reconstruction that closely matches the ground truth, whereas Shift, Scale, and FiLM all fail to capture the solution structure.

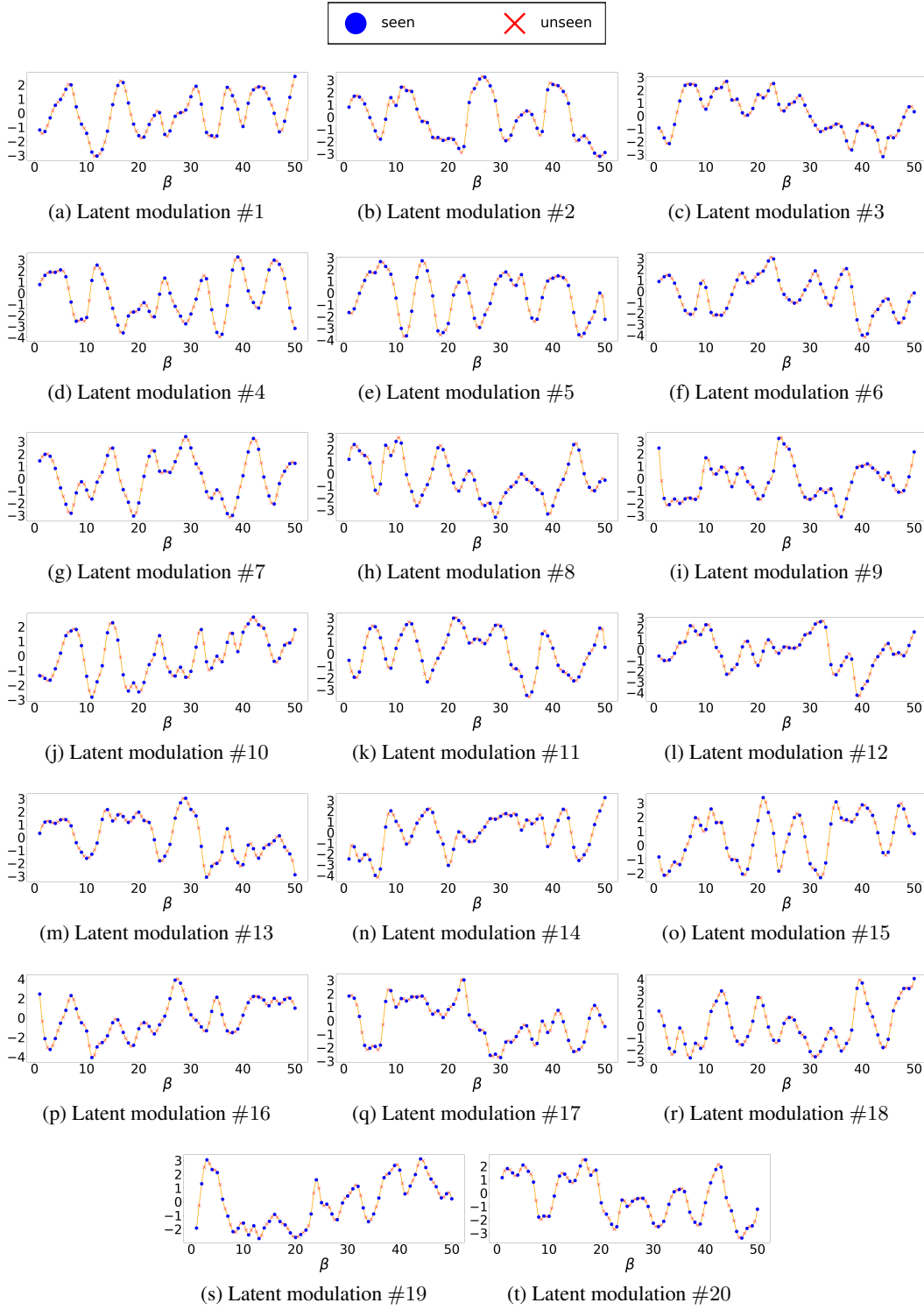


Figure 9: Dynamics of Latents (convection equations)

F.3 Statistical Comparison Across Modulation Methods

Table 7 provides the mean and standard deviation of PSNR across three trials for each modulation method and PDE benchmark. While Table 1 in the main paper presents the mean values, this supplementary table highlights the consistency and robustness of the results by reporting the standard deviations.

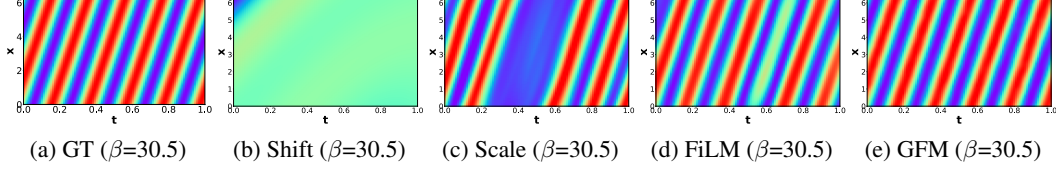


Figure 10: **[Convection] Predicted solution snapshots for unseen coefficient $\beta = 30.5$ from interpolated latent codes using different modulation methods.** (a) Ground truth, (b) Shift, (c) Scale, (d) FiLM, and (e) GFM. For each method, the latent code for $\beta = 30.5$ is obtained via cubic interpolation between neighboring training latents.

Table 7: Comparison with existing modulation methods with mean and standard deviation (PSNR \uparrow).

Modulation	Convection	Helmholtz #1	Helmholtz #2	Navier-Stokes	Kuramoto-Sivashinsky
Shift [17]	13.225 \pm 0.597	25.467 \pm 0.925	18.249 \pm 0.901	25.767 \pm 0.635	16.962 \pm 0.819
Scale [17]	27.410 \pm 0.608	25.669 \pm 0.492	18.183 \pm 0.683	31.469 \pm 0.507	20.395 \pm 0.997
FiLM [22]	27.330 \pm 1.254	33.611 \pm 0.506	22.771 \pm 0.533	31.904 \pm 0.502	22.297 \pm 1.896
GFM	42.474 \pm 0.412	39.139 \pm 0.530	29.033 \pm 0.556	34.754 \pm 0.779	29.827 \pm 0.290

627 G More Experimental Results on Bidirectional Inference

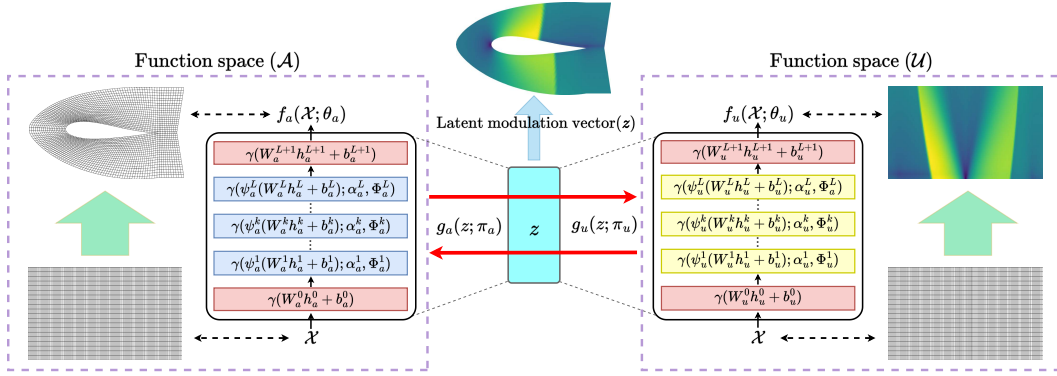


Figure 11: **Visualization of the PDEfuncta pipeline for the Airfoil dataset.** Two modalities in airfoil data are represented as separate INRs f_a and f_u , and are jointly compressed and reconstructed through a shared latent modulation vector z .

628 This section provides supplementary explanations for Section 5.3. First, we include a schematic
 629 illustration to visually describe how the Airfoil dataset is used in the unseen neural operator setting
 630 (cf. Section 5.3.2). Figure 11 illustrates the overall architecture of PDEfuncta applied to the Airfoil
 631 dataset. The two modalities—geometry and flow field—are modeled using separate INRs f_a and f_u ,
 632 while a shared latent modulation vector enables joint compression and bidirectional reconstruction
 633 between them. Additionally, we present FWI reconstruction results for seen samples using different
 634 modulation methods in Figures 12 and 13.

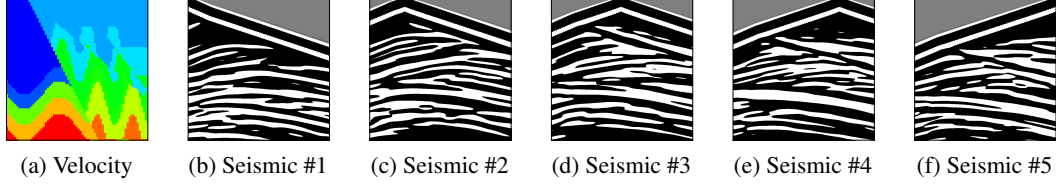


Figure 12: **Ground truth for a representative FWI sample.** (a) Velocity map and (b–f) corresponding seismic data.

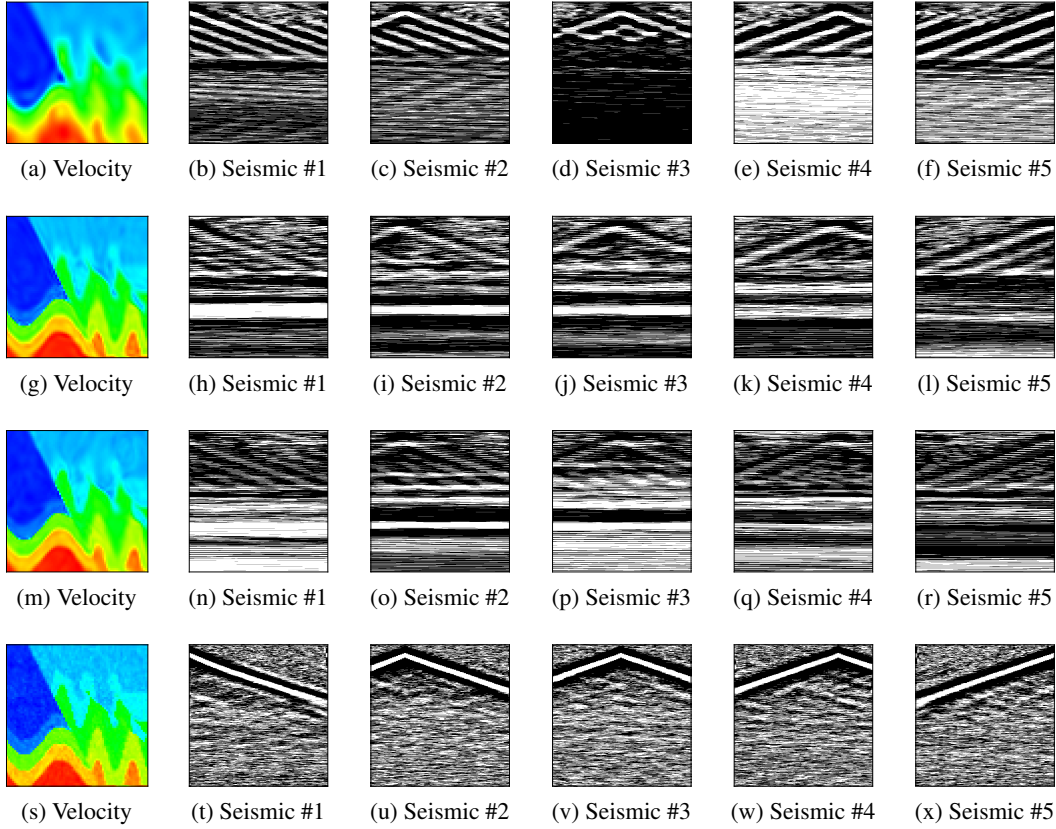


Figure 13: **FWI bidirectional reconstruction results for seen samples using different modulation methods.** Each row shows a velocity map and its corresponding seismic data reconstructed by a different modulation method: (a–f) Shift, (g–l) Scale, (m–r) FiLM, and (s–x) GFM

H Implementation Details of Fourier Basis Construction

This section provides additional details regarding the construction and implementation of the fixed Fourier bases used in Global Fourier Modulation (GFM), supplementing Section 3.1 of the main paper. The fixed Fourier bases for each layer are constructed following the procedure introduced in [37]. In all experiments, the fixed Fourier basis Φ^k for each layer is constructed by combining a set of frequencies and phase shifts (cf. Equation 3.1). To ensure spectral diversity, we employ n_{low} low-frequency components ($\omega_{\text{low}} = \{\frac{1}{n_{\text{low}}}, \frac{2}{n_{\text{low}}}, \dots, 1\}$) and n_{high} high-frequency components ($\omega_{\text{high}} = \{1, 2, \dots, n_{\text{high}}\}$). For each frequency, we use n_{phase} phase shifts uniformly spaced over $[0, 2\pi]$. Here, n_{low} , n_{high} , and n_{phase} are hyperparameters, and the total number of basis vectors per layer is thus $D = (n_{\text{low}} + n_{\text{high}}) \times n_{\text{phase}}$. Each basis vector is evaluated at uniformly spaced M points within the interval $[-T_{\text{max}}/2, T_{\text{max}}/2]$, where $T_{\text{max}} = 2\pi n_{\text{low}}$. The number of sampling points M is chosen to match the input dimension of the corresponding layer, ensuring compatibility with weight reparameterization. In our implementation, these bases are precomputed, fixed throughout training, and not updated by gradient descent.

I Meta-learning based Training Algorithm

In this section, we outline the meta-learning-based training and inference procedures for PDEfunctn. Algorithm 1 describes the training phase, where both network parameters and sample-specific latent vectors are jointly optimized via a nested inner–outer loop.

Algorithm 1 Training of the proposed method

```

1: /* Training */
2: Input: Spatio-temporal grid  $\mathcal{X} = (x, t)$ 
3: Randomly initialize  $\theta = \{\theta_a, \theta_u\}$ ,  $\pi = \{\pi_a, \pi_u\}$  and set  $z_{\text{train}} \leftarrow 0$  (zero vector)
4: while not done do
5:   Sample batch  $\mathcal{B}$  of output  $\{a^i, u^i\}_{i \in \mathcal{B}}$ 
6:   Sample  $K$  examples from batch  $\mathcal{B}$ 
7:   /* Inner loop */
8:   for  $j = 1$  to  $K$  do
9:      $\mathbb{L}_j = \mathbb{L}_{\mathcal{X}_j}(a^j, f_a(\mathcal{X}; \theta_a, g_a(z_{\text{train}}^j; \pi_a))) + \mathbb{L}_{\mathcal{X}_j}(u^j, f_u(\mathcal{X}; \theta_u, g_u(z_{\text{train}}^j; \pi_u)))$ 
10:     $z_{\text{train}}^j \leftarrow z_{\text{train}}^j - \eta_{\text{inner}} \nabla_{z_{\text{train}}^j} \mathbb{L}_j$ 
11:   end for
12:   /* Outer loop */
13:    $\mathbb{L}_{\mathcal{B}} = \sum_{i \in \mathcal{B}} (\mathbb{L}_{\mathcal{X}_i}(a^i, f_a(\mathcal{X}; \theta_a, g_a(z_{\text{train}}^i; \pi_a))) + \mathbb{L}_{\mathcal{X}_i}(u^i, f_u(\mathcal{X}; \theta_u, g_u(z_{\text{train}}^i; \pi_u))))$ 
14:    $\theta \leftarrow \theta - \eta_{\text{outer}} \nabla_{\theta} \mathbb{L}_{\mathcal{B}}$ 
15:    $\pi \leftarrow \pi - \eta_{\text{outer}} \nabla_{\pi} \mathbb{L}_{\mathcal{B}}$ 
16: end while

```

653 J Comparison with Spatial Functa

Table 8: Performance comparison (PSNR \uparrow /MSE \downarrow) between spatial functa and GFM on Convection, Helmholtz, and Kuramoto-Sivashinsky equations.

Modulation	Convection		Helmholtz #1		Helmholtz #2		Kuramoto-Sivashinsky	
	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
SpatialFuncta [36]	20.390	0.0739	28.908	0.0068	25.976	0.0141	20.728	3.4297
GFM	42.474	0.0004	39.139	0.0006	29.033	0.0056	29.827	0.4198

654

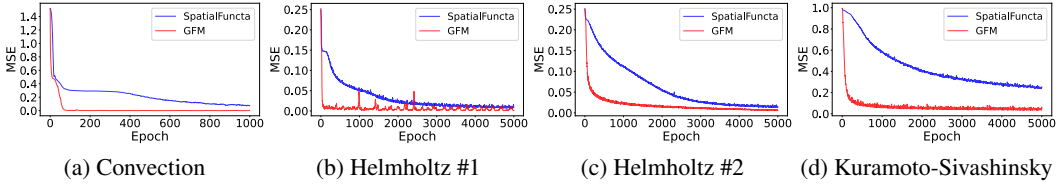


Figure 14: Train loss (MSE) curves comparing spatial functa and GFM across epochs for PDE benchmarks (Convection, Helmholtz#1, Helmholtz#2, Kuramoto-Sivachinsky)

655 To evaluate the performance of our proposed GFM in comparison with Spatial Functa [36], we
 656 conduct experiments under the same conditions to those reported in Table 1. Unlike global modulation
 657 approaches such as Shift, Scale, FiLM and GFM, which inject instance-specific information via
 658 a low-dimensional global latent vector (e.g., $z \in \mathbb{R}^{20}$ in our experimental setup), Spatial Functa
 659 utilizes a high-dimensional spatially structured latent code (e.g., $z \in \mathbb{R}^{8 \times 8 \times 16}$ in our implementation)
 660 designed to modulate local regions of the underlying function space. Note that these latent dimensions
 661 are representative of the settings used in our experiments and may vary depending on the specific
 662 model architecture or dataset. Due to the structural difference in latent space dimensionality, a
 663 direct comparison under fixed latent size is infeasible. We therefore present Spatial Functa as a
 664 separate baseline, following the original implementation and latent dimension, and report results
 665 on the convection, Helmholtz, and Kuramoto-Sivashinsky benchmarks. We exclude Navier-Stokes
 666 since its 3D-coordinates are incompatible with the 2D latent grid arrangement used by Spatial Functa.

667 Table 8 reports the reconstruction accuracy for both methods, and Figure 14 shows the training loss
 668 curves. Figures 15 and 16 provide the reconstructed solutions on convection and Helmholtz dataset,
 669 respectively. Notably, in Figures 15 and 16, we observe that as β and a_1 increases, the reconstruction
 670 quality of Spatial Functa (second row) deteriorates significantly, whereas GFM (third row) maintains
 671 high fidelity even in challenging high-frequency regimes. While Spatial Functa achieves moderate
 672 improvements over basic global modulation methods such as Shift, Scale, and FiLM (cf. Table 1),
 673 our GFM method consistently achieves superior PSNR and lower reconstruction errors across all
 674 considered datasets. This empirical advantage highlights the effectiveness of explicitly injecting
 675 frequency-aware priors into the modulation space, allowing GFM to better capture both global
 676 structure and fine-scale, which global modulation was previously thought to be insufficient.

677 We also observe that Spatial Functa requires significantly higher computational cost compared to
 678 global modulation methods. The use of high-dimensional spatial latent codes results in longer training
 679 times (see Section E.2). In contrast, GFM requires less computational cost, leveraging compact latent
 680 codes for faster training and inference.

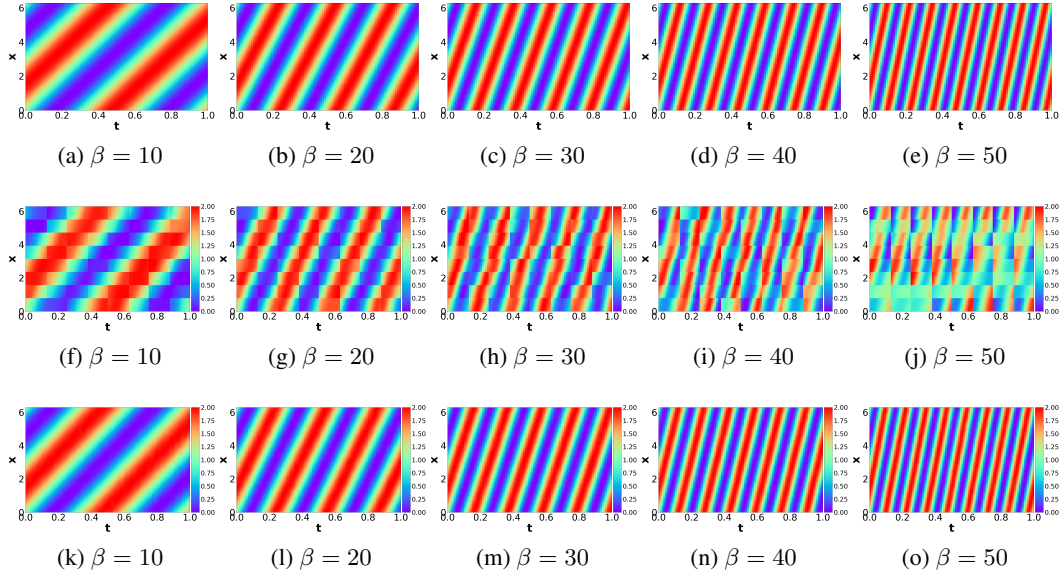


Figure 15: Reconstruction results for spatial function and GFM on the convection equation with $\beta = \{10, 20, 30, 40, 50\}$. Rows represent ground truth (top), spatial function (middle), and GFM (bottom).

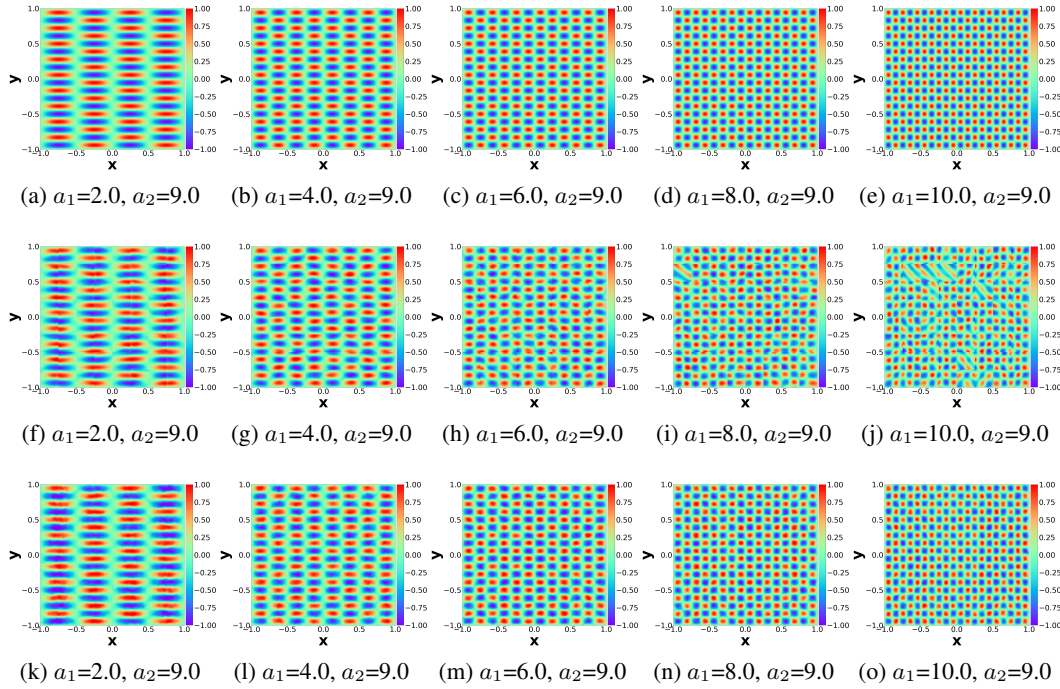


Figure 16: Reconstruction results for spatial function and GFM on the convection equation with $(a_1, a_2) = \{(2.0, 9.0), (4.0, 9.0), (6.0, 9.0), (8.0, 9.0), (10.0, 9.0)\}$. Rows represent ground truth (top), spatial function (middle), and GFM (bottom).