GRADIENT-GUIDED PHYSICS-INFORMED MULTI-FIDELITY U-MAMBA FOR DISCONTINUOUS FLOW FIELD PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Surrogate modeling of strong discontinuities, such as shock waves and contact discontinuities, in compressible flows remains challenging: high-resolution numerical schemes are computationally prohibitive on fine grids, purely data-driven operators lack physical consistency and induce Gibbs oscillations, and standard physics-informed neural networks (PINNs) break down where strong-form differential residuals become ill-defined. We propose PIMUG, a Gradient-Guided Physics-Informed Multi-Fidelity U-Mamba framework coupling three complementary branches: (1) a multi-fidelity trunk extracting spatio-temporal features from multi-resolution, multi-timestep inputs; (2) an explicit gradient branch supervising directional derivatives to sharpen discontinuity fronts without oversmoothing; and (3) a coordinate-conditioned physics branch with a divergence-adaptive residual loss that enforces Euler conservation while relaxing ill-posed penalties at shock fronts. An Input Compression Module (ICM) suppresses spatio-temporal redundancy across fidelity levels. On 2D Riemann benchmarks spanning slip lines, rarefactions, and shock collisions, PIMUG preserves sharp discontinuity profiles without visible Gibbs oscillations. As a multi-fidelity field corrector, PIMUG consumes low-fidelity numerical solutions at the target time—a richer input regime than instance-wise pointwise PINNs (PINN-WE), which solve from initial and boundary conditions alone; under this setting, PIMUG reduces self-enforced Euler residuals by two to four orders of magnitude relative to PINN-WE with millisecond amortized inference. Compared to Fourier Neural Operators (FNO) on identical inputs, PIMUG eliminates Gibbs blurring while retaining physical consistency; augmenting FNO with the same residual loss merely induces degenerate over-smoothing while degrading data accuracy by two orders of magnitude, isolating PIMUG's gains as architectural rather than a loss artifact.

1 INTRODUCTION

Accurate prediction of shocks and contact discontinuities is central to compressible-flow design and safety analysis. High-resolution shock-capturing schemes are reliable but computationally expensive, whereas learned surrogates can be fast yet fail at discontinuities in complementary ways: data-driven networks may violate conservation laws outside the training distribution, and physics-informed neural networks (PINNs) use strong-form residuals that are not well defined across sharp jumps (Godunov, 1959; Jiang & Shu, 1996; Harten, 1984; Cockburn & Shu, 1998; Raissi et al., 2019; Liu et al., 2024a).

Existing remedies address only part of this difficulty. Multi-fidelity networks exploit inexpensive low-resolution solutions but usually omit physical constraints, and representative composite multi-fidelity models such as MPINN (Meng & Karniadakis, 2020) remain point-wise and are designed primarily for smooth function approximation and inverse problems (Huang et al., 2023; Liu & Wang, 2019); discontinuity-aware PINNs down-weight ill-posed residual samples but remain predominantly point-wise autonomous solvers that map coordinates to a solution from initial and boundary data alone (Liu et al., 2024a); and convolutional or state-space backbones capture spatial context without explicitly reconciling data accuracy, discontinuity sharpness, and physical consistency (Gu & Dao, 2024; Ma et al., 2024). Our starting point is that these requirements should be handled

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

by complementary objectives. Image restoration shows that explicit gradient supervision effectively recovers sharp edges because gradient magnitudes concentrate at high-contrast structures (Sun et al., 2011; Ma et al., 2020; Yang et al., 2017). Flow discontinuities exhibit analogous behavior, so PIMUG[1] makes gradient reconstruction an explicit target rather than an implicit byproduct of pixelwise loss.

We propose a U-Mamba backbone with three coordinated branches: a trunk branch extracting global patterns from multi-timestep, multi-fidelity inputs; a gradient branch supervising flow-field gradients; and a physics-informed branch modulating features via spatiotemporal coordinates, alongside an input compression module that mitigates channel redundancy. Our contributions are:

(1) a multi-branch architecture combining multi-fidelity learning, explicit gradient supervision, and feature-level physical conditioning for discontinuous flow prediction;

(2) an input compression module tailored to multi-fidelity, multi-variable, and multi-timestep flow channels; and

(3) systematic evaluation across three Riemann configurations, including robustness tests, ablations, and a cross-case transfer study on a held-out fourth case exposing the limits of fixed-encoder adaptation.

2 Related Work

2.1 Physics-Informed and Multi-Fidelity Surrogates

PINNs embed governing equations into training via automatic differentiation (Raissi et al., 2019). While combining sparse observations with physical priors, strong-form residuals become noisy or undefined across shocks. gPINN adds residual-gradient constraints (Yu et al., 2022), artificial-viscosity methods stabilize shock neighborhoods (Coutinho et al., 2023), and PINN-WE down-weights samples in discontinuous regions (Liu et al., 2024a). Multi-fidelity extensions like MPINN (Meng & Karniadakis, 2020) learn from multi-fidelity data with composite networks, but are tailored to smooth approximation and inverse problems. Benchmarking an amortized multi-fidelity baseline (MPINN) demonstrates that multi-fidelity data alone cannot enforce governing equations on discontinuous flows ($R_{\text{disc}} \sim 2.7 \times 10^3$, $27\times$ higher than PIMUG; Appendix A.3), confirming that PIMUG's gains stem from its architectural physical and gradient mechanisms rather than data privilege alone. PIMUG retains a discontinuity-aware residual while introducing an architectural gradient stream.

2.2 Neural Operators, State-Space Models, and Gradient Guidance

Neural operators like DeepONet and FNO learn mappings between function spaces, amortizing inference across instances (Lu et al., 2021; Li et al., 2021). Physics-informed variants add residuals to training (Li et al., 2024), though their weighting protocols target coordinate rollouts rather than fixed-grid multi-fidelity reconstruction. To cleanly separate physical constraints from backbone capabilities, we benchmark against a *physics-augmented FNO* trained on the shared 16-channel input with the unified discrete Euler residual used in our evaluation; Section 5 shows that this loss causes degenerate over-smoothing rather than reproducing PIMUG, isolating our architectural contribution. Spectral global bases can also induce Gibbs smoothing near discontinuities (Raonić et al., 2023; Peyvan et al., 2024). Graph-based PDE solvers capture dynamics on unstructured meshes (Brandstetter et al., 2022; Pfaff et al., 2021); PIMUG provides a complementary linear-complexity state-space formulation on Cartesian grids, leaving unstructured graph models to future work. U-Mamba couples U-Net multiscale features with selective state-space Mamba blocks (Gu & Dao, 2024; Ma et al., 2024), extended to 2D scanning in vision (Zhu et al., 2024; Liu et al., 2024b) to capture long-range wave interactions with linear complexity. Finally, image restoration demonstrates that explicit gradient supervision improves sharp edges (Ma et al., 2020; Zhu et al., 2023; Cao et al., 2025). PIMUG transfers this concept to flow modeling via fixed difference operators, a dedicated gradient branch, and gradient-domain losses.

[1] Physics-Informed Multi-fidelity U-Mamba with Gradient guidance.

3 PROBLEM SETTING AND DATA

3.1 GOVERNING EQUATIONS AND DISCONTINUOUS RIEMANN CONFIGURATIONS

We consider inviscid compressible flow on the unit square. The flow state can be represented by the primitive variables $\mathbf{W} = (\rho, u, v, p)^\mathsf{T}$ (density, x-velocity, y-velocity, and pressure) or the conserved variables $\mathbf{U}(\mathbf{W}) = (\rho, \rho u, \rho v, E)^\mathsf{T}$. The two-dimensional Euler equations in conservation form are

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{1}$$

where the two Cartesian fluxes are

$$\mathbf{F}_1 = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{F}_2 = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}, \tag{2}$$

closed by the ideal-gas law $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2+v^2)$ ($\gamma = 1.4$). Low-fidelity inputs and high-fidelity references are generated by schemes detailed in Appendix A.1. Each benchmark is a 2D Riemann problem with four quadrant states (Lax & Liu, 1998): Cases 1–3 cover pure slip lines, slip-line–rarefaction coupling, and shock–slip-line interaction; Case 4 is a held-out slip-line case for transfer. Both grid levels store 800 aligned frames per run (799 adjacent pairs). Training uses nominal initial conditions with perturbations $\{-4\%, -2\%, +2\%, +4\%\}$, while the disjoint test set uses $\{-5\%, -3\%, -1\%, +1\%, +3\%, +5\%\}$.

3.2 MULTI-FIDELITY SPATIOTEMPORAL INPUT REPRESENTATION

Let $\mathbf{W}_1$ and $\mathbf{W}_2$ denote solutions on the coarse 200×200 and fine 400×400 grids, the latter downsampled to 200×200. Samples pair adjacent time levels, grouping channels by time and variable:

$$\mathbf{X} = [\mathbf{W}^{t-1}, \mathbf{W}^t] \in \mathbb{R}^{16\times 200\times 200}, \tag{3}$$

where each 8-channel block $\mathbf{W}^\tau$ ($\tau \in \{t-1, t\}$) stacks fine–coarse pairs for all primitive variables, $[\mathbf{W}^\tau_{2,\rho}, \mathbf{W}^\tau_{1,\rho}, \ldots, \mathbf{W}^\tau_{2,p}, \mathbf{W}^\tau_{1,p}]$. The prediction target is the high-fidelity (WENO-Z+) primitive field $\mathbf{W}^t_{\mathrm{ref}}$ at the later time level t; PIMUG thus performs multi-fidelity reconstruction of the current frame rather than forward forecasting. The model exploits grid-convergence cues from two numerical dissipation levels of an inexpensive scheme without invoking the costly WENO solver at inference.

4 PIMUG METHODOLOGY

Figure 1 summarizes PIMUG, which consumes the 16-channel multi-fidelity input of Section 3.2. The trunk maps aligned multi-fidelity fields to the target field; the gradient branch reconstructs gradient fields with fixed difference operators and injects high-frequency features into the trunk through skip connections; and the physics branch transforms (x, y, t) into feature-level conditioning signals that modulate trunk features layer by layer. Detailed dimensions and operator definitions are given in Appendix A.2.

4.1 NETWORK ARCHITECTURE

Trunk branch. The trunk consists of the input compression module (ICM; Section 4.2) followed by a four-stage U-Mamba encoder–bottleneck–decoder network. The encoder extracts local multiscale features, while the Mamba bottleneck models long-range dependencies among shocks, slip lines, and vortical structures (each feature map is flattened into a 1D sequence by a unidirectional raster scan; Appendix A.2). Skip connections carry encoder features to the corresponding decoder stages, and features from the gradient and physics branches are fused into the trunk representation along these skips. The final decoder output is concatenated with the gradient-branch output and mapped by a 1×1 convolution to the four predicted primitive variables (ρ, u, v, p).

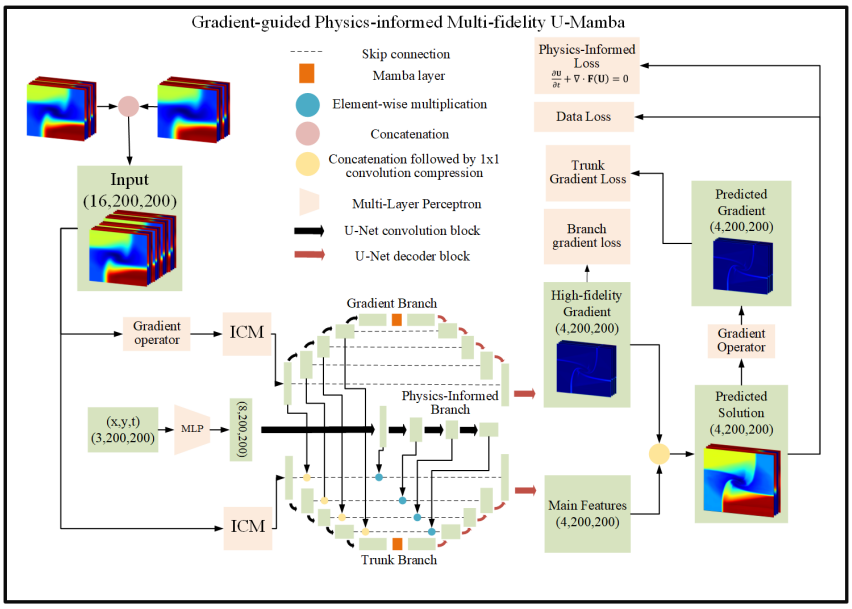


Figure 1: Overall framework of PIMUG: a multi-fidelity trunk branch, a fixed-operator gradient branch fused through skip connections, a coordinate-conditioned physics branch, and a final 1×1 fusion convolution.

Gradient branch. The gradient branch computes channel-wise spatial gradient magnitudes from the aligned input fields using two fixed 3×3 central difference kernels:

$$k_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad k_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{4}$$

where k_x and k_y compute scaled horizontal and vertical differences (the $1/(2\Delta x)$ factor is omitted; the uniform scaling does not affect the relative gradient magnitudes used for supervision). For any input field $\mathbf{W}$ with C channels, the discrete gradient magnitude is evaluated channel-wise as $\mathrm{Grad}(\mathbf{W})_c = [(k_x * \mathbf{W}_c)^2 + (k_y * \mathbf{W}_c)^2 + \epsilon]^{1/2}$, preserving the channel dimension while isolating steep spatial transitions. The resulting gradient fields pass through the ICM and a U-Mamba network of the same depth as the trunk; at each encoder level the gradient feature is concatenated with the corresponding trunk feature and compressed by a 1×1 convolution. The branch output also participates in the final fusion, but its main contribution is the high-frequency evidence it makes available to the encoder before decoding, rather than relying solely on a pixelwise reconstruction loss.

Physics branch. The physics branch maps coordinates (x, y, t) through an MLP and a four-stage residual encoder; its output features are aligned with the trunk encoder in spatial resolution and channel depth, and parameter conditioning multiplies them with the trunk skip features (Gupta & Brandstetter, 2023), as in related time-dependent U-Net solvers and Riemann operator networks (Oommen et al., 2024; Ovadia et al., 2023; Peyvan et al., 2024). The Autograd residual L_{Phy} differentiates the output only with respect to the coordinate feature inputs (x, y, t), measuring the sensitivity of the prediction along the conditioning path. It therefore acts as a continuous feature-space physical regularizer: it does not capture the spatial variation embedded in the discrete input grid processed by the trunk, and it does not by itself constitute a conservation statement on the grid—a role delegated to the unified discrete residual R_{disc} (Section 5; Appendix A.3).

4.2 Input Compression Module

The ICM is a shared front-end component rather than a fourth branch: two copies of it preprocess the 16-channel inputs of the trunk and the gradient branches before their respective encoders. Because

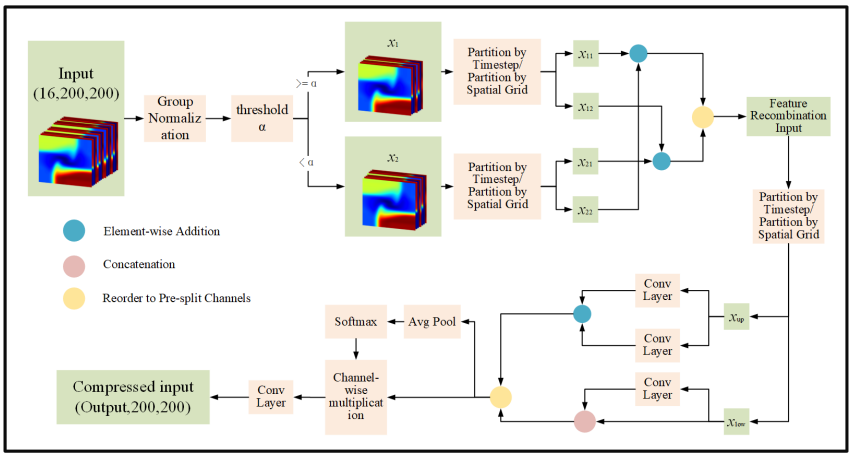


Figure 2: Detailed processing pipeline of the compression submodule within the ICM. Features undergo group-normalized gating, structured cross-recombination, dual-path multiscale convolution, and channel-attention fusion.

the input stacks four physical variables, two timesteps, and two fidelity levels ($4 \times 2 \times 2 = 16$ channels), standard convolutions would suffer from substantial inter-temporal and cross-resolution channel redundancy. Inspired by Spatial and Channel Reconstruction Convolution (SCConv) (Li et al., 2023), the ICM redesigns feature partitioning, cross-recombination, and channel attention to tailor them to multi-fidelity flow fields.

As illustrated in Figure 2, the ICM cascades two *specialized* compression submodules—TScConv, which maps the 16 input channels to 8, and MFScConv, which operates at 8 channels—that share one four-step pipeline and differ only in their channel-grouping criterion (Step 2). The $16 \to 8$ compression is realized inside the dual-path stage of the first submodule through 1×1 squeeze convolutions, not by a separate layer inserted between the two submodules. Each submodule (i) normalizes its input $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ by group normalization and thresholds a bounded nonlinearity of the normalized activation (scaled by the per-channel weight) at $\alpha = 0.5$ to produce an information mask and its complement—the masks partition feature *values* rather than channel indices, so both subsets retain the full channel depth C and no channel is discarded, reordered, or left with a mismatched depth; (ii) partitions each subset along channels into two sub-features ($\mathbf{x}_{11}, \mathbf{x}_{12}$ and $\mathbf{x}_{21}, \mathbf{x}_{22}$) and cross-combines them as $\mathbf{c}_1 = \mathbf{x}_{11} + \mathbf{x}_{22}$ and $\mathbf{c}_2 = \mathbf{x}_{12} + \mathbf{x}_{21}$, forcing exchange between differing time/fidelity sources; the grouping acts on fixed channel indices, and the cascaded submodules use complementary criteria—an interleaved (even/odd) split that separates the two fidelity sources under the fine–coarse interleaved layout of Eq. (3), followed by a contiguous half split that separates the two recombinant groups carrying the contributions of the two temporal blocks; (iii) splits the recombined feature $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2]$ into a primary stream passed through two parallel convolutions whose outputs are summed element-wise, and a secondary stream that is convolved and channel-self-concatenated; and (iv) concatenates the two processed streams and fuses them by Softmax channel attention obtained from global average pooling, $\mathbf{Y} = \mathbf{w} \otimes \mathbf{F}$.

This decomposition explicitly separates and exchanges complementary temporal tendencies and grid-scale variations instead of uniformly convolving redundant representations; the ablation study (Section 5.3) confirms the design quantitatively, with the ICM reducing the data error of a baseline U-Net by 62.6% under the $+3\%$ perturbation (1.88×10^{-3} to 7.03×10^{-4}) and a further 37.5% improvement in the full PIMUG (4.39×10^{-4}).

4.3 Loss Function

The composite objective is

$$L = L_{\text{Data}} + L_{\text{Phy}} + L_{\text{G1}} + L_{\text{G2}}, \tag{5}$$

where L_{Data} is the data reconstruction loss, L_{Phy} is the Euler residual loss, and L_{G1} and L_{G2} are the trunk and gradient-branch losses; all four terms are combined at unit weight. Following the discontinuity-aware weighting idea of PINN-WE (Liu et al., 2024a), the data and residual terms

share the pointwise adaptive weight

$$\lambda_1(x, y, t) = \frac{1}{k_1\big(|\nabla \cdot \mathbf{u}_{\mathrm{nn}}| - \nabla \cdot \mathbf{u}_{\mathrm{nn}}\big) + 1}, \tag{6}$$

built from the predicted velocity divergence $\nabla \cdot \mathbf{u}_{\mathrm{nn}}$ with $\mathbf{u}_{\mathrm{nn}} = (u_{\mathrm{nn}}, v_{\mathrm{nn}})^{\mathsf{T}}$, which distinguishes compression from expansion. In strongly compressed shock regions, $\nabla \cdot \mathbf{u}_{\mathrm{nn}}$ is large and negative, so $\lambda_1 \to 0$ and both terms are relaxed precisely where the strong-form differential residual becomes ill-defined across discontinuous jumps; in smooth or expanding regions $\lambda_1 \to 1$ and both terms retain their full strength. Crucially, shear layers and contact discontinuities (such as the slip lines in Cases 1 and 4) are divergence-free ($\nabla \cdot \mathbf{u}_{\mathrm{nn}} \approx 0$) and maintain $\lambda_1 \approx 1$; their sharpness is captured by the dedicated gradient branch ($L_{\mathrm{G1}}, L_{\mathrm{G2}}$) rather than by divergence relaxation. The constant $k_1 = 0.12$ is selected once on the Case 1 training perturbations and kept fixed across all cases, perturbations, and transfer experiments. Because λ_1 multiplies the pointwise errors inside the squared norms of Eqs. (7) and (8), its effective influence on both terms scales as λ_1^2. Two properties limit the risk of amplitude drift and of exploiting the weight: λ_1 stays above 0.5 unless the compression term exceeds $1/k_1 \approx 8.3$, so supervision is relaxed only in a narrow band around strong compressive cores while states on both sides of a shock remain fully anchored against a constant amplitude shift; and although λ_1 is *not* detached from the computation graph in our implementation, no degenerate divergence-inflating solution was observed in any training run. We note explicitly that λ_1 is an *inherited* element of the PINN-WE objective and not one of our contributions: the divergence that drives it is taken along the coordinate-conditioning path of the physics branch rather than from the grid-level divergence of the predicted field, so on our trained models it stays small and $\lambda_1 \in [0.997, 1.000]$ at every training stage we inspected (Appendix A.2); the weighted objective is therefore numerically indistinguishable from its unweighted counterpart ($< 0.05\%$, Section 5), and the sharpness of the predicted fronts is carried by the gradient branch rather than by this relaxation. A detailed discussion and the planned detached-variant ablation are deferred to Appendix A.2.

Let $N = 799$ be the number of adjacent-timestep samples, and let $\lambda_1(i)$ abbreviate $\lambda_1(x, y, t)$ evaluated on sample i. The data and physical residuals are

$$L_{\mathrm{Data}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \lambda_1(i) \big(\mathbf{W}_{\mathrm{nn}}(i) - \mathbf{W}_{\mathrm{ref}}(i) \big) \right\|^2, \tag{7}$$

$$L_{\mathrm{Phy}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \lambda_1(i) \left(\frac{\partial \mathbf{U}(\mathbf{W}_{\mathrm{nn}})}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{W}_{\mathrm{nn}}) \right) \right\|^2. \tag{8}$$

The gradient terms are

$$L_{\mathrm{G1}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathrm{Grad}(\mathbf{W}_{\mathrm{nn}}(i)) - \mathrm{Grad}(\mathbf{W}_{\mathrm{ref}}(i)) \right\|^2, \tag{9a}$$

$$L_{\mathrm{G2}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathrm{Grad}_{\mathrm{nn}}(i) - \mathrm{Grad}(\mathbf{W}_{\mathrm{ref}}(i)) \right\|^2. \tag{9b}$$

Here $\mathrm{Grad}_{\mathrm{nn}}$ denotes the output of the dedicated gradient branch. The objective therefore separates three roles: data fitting anchors the field, the residual term regularizes smooth physical evolution, and gradient supervision explicitly sharpens high-contrast structures.

5 Experiments

5.1 Experimental Setup

Training. PIMUG is trained with Adam at learning rate 5×10^{-4} and batch size 5 for at most 1000 epochs. The learning rate is multiplied by 0.7 after five non-improving epochs. The data and gradient losses are optimized at every epoch, while the Euler residual is evaluated every three epochs to reduce the cost of automatic differentiation. Full architecture and optimization details appear in Appendix A.2.

Baselines. PINN-WE (Liu et al., 2024a) is the physics-constrained point-wise baseline, reproduced with its original Adam plus L-BFGS protocol; as a point-wise PINN it consumes only coordinates and initial/boundary data, cannot take the 16-channel multi-fidelity input, and is therefore trained in its native instance-wise regime (Appendix A.3). FNO (Li et al., 2021) is the purely data-driven neural-operator baseline, receiving the same 16-channel input and the same data loss and split. We additionally train a *physics-augmented FNO* with identical input and data loss but appending the unified discrete Euler residual of our evaluation at unit weight, so that the effect of the residual term can be separated from that of the architecture; it is evaluated on Case 1. We stress the input-information asymmetry: PIMUG and FNO receive low-fidelity Leapfrog solutions at the target time whereas PINN-WE receives none, so PINN-WE comparisons contrast paradigms rather than input regimes; within the shared 16-channel regime the PIMUG–FNO pairing isolates physical regularization from amortized data fitting, and a multi-fidelity baseline (MPINN) is likewise evaluated under the identical 16-channel regime (Appendix A.3).

Metrics. We evaluate three dimensions: (1) *field reconstruction accuracy* (MSE_{data}), summed over the four primitive variables (ρ, u, v, p); (2) *unified discrete PDE consistency* (R_{disc}), computed with an identical second-order central-difference operator in space and first-order difference in time for all surrogates and for the WENO reference floor; and (3) *continuous PDE enforcement* (R_{cont}), the *unweighted* self-enforced automatic-differentiation Euler residual (total-energy component) of physics-informed surrogates; because the adaptive weights are removed for both methods, this column reads as a common-metric comparison under one weighting scheme rather than a restatement of each training objective (Appendix A.3). Table 1 reports the benchmark at the $+5\%$ perturbation. Because PINN-WE requires a separate multi-hour optimization from scratch for each initial condition, its native residual is reported at $+5\%$ across cases and its unified discrete trajectory replay on Case 1, whereas PIMUG and FNO amortize inference across all cases and perturbations in milliseconds (Appendix A.4).

Table 1: Multi-criteria benchmark on Cases 1–3 under $+5\%$ perturbation: data error (MSE_{data}), unified discrete residual (R_{disc} via second-order central differences against the WENO reference floor), and unweighted self-enforced continuous autograd residual (R_{cont}, total energy). Bold indicates best among learning surrogates in each column; lower is better.

		Data Error (MSE_{data})			**Unified Discrete Residual (R_{disc})**			**Unweighted Continuous Residual (R_{cont})**		
Method	Paradigm	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
WENO Reference	High-Order Solver	–	–	–	0.97	4.06	8.23×10^{1}	–	–	–
PINN-WE	Point-wise PINN	1.86×10^{-2}	–*	–*	2.51×10^{3}	–*	–*	4.70×10^{-3}	6.18×10^{-4}	1.10×10^{-3}
FNO	Spectral Operator	$\mathbf{8.04 \times 10^{-5}}$	$\mathbf{9.71 \times 10^{-6}}$	$\mathbf{2.99 \times 10^{-5}}$	**4.69**	**4.46**	7.46×10^{1}	– (unconstrained) –		
PIMUG (Ours)	Physics-Informed Operator	1.40×10^{-3}	1.28×10^{-3}	4.26×10^{-4}	1.00×10^{2}	1.19×10^{1}	$\mathbf{7.00 \times 10^{1}}$	$\mathbf{2.34 \times 10^{-5}}$	$\mathbf{3.33 \times 10^{-7}}$	$\mathbf{5.65 \times 10^{-8}}$

*PINN-WE is an instance-wise pointwise model without native grid trajectory outputs; trajectory replay and MSE are benchmarked on Case 1; native autograd residuals are reported on all three cases. PIMUG and FNO provide amortized inference across all cases in milliseconds.

5.2 Comparative Evaluation on 2D Riemann Problems

Case studies. Case 1 contains four interacting slip lines that develop complex roll-up vortical structures in the central core: PIMUG reproduces the global wave morphology and resolves the fine vortex cores, whereas PINN-WE severely distorts the roll-up region (1.86×10^{-2}, an order of magnitude worse than PIMUG) and FNO captures smooth outer waves but blurs the internal shear layers. Case 2 couples slip lines with expanding rarefaction fans: PINN-WE misses the subtle density depression near $x = 0.5$ at the $y = 0.75$ section because it over-penalizes high frequencies, while PIMUG tracks this low-amplitude variation—showing that the multi-fidelity Mamba trunk preserves low-amplitude structural details—and FNO interpolates smoothly but lacks constraint enforcement. Case 3 is shock dominated, featuring strong shock collisions and localized slip lines: PIMUG matches the reference density profiles across multiple sharp extrema (Figure 7, Appendix A.5), PINN-WE excessively smooths the shock fronts and misses discontinuous transitions in the core, and the 12-mode Fourier truncation of FNO smears the shock fronts across multiple grid cells and introduces unphysical Gibbs ripples.

Quantitative summary. Table 1 evaluates accuracy, discrete consistency, and continuous physical enforcement. Relative to PINN-WE, PIMUG achieves a **13.3-fold reduction in data error** on

Case 1 (1.40×10^{-3} vs 1.86×10^{-2}) and reduces the continuous PDE residual by two to four orders of magnitude (2.34×10^{-5} vs 4.70×10^{-3}; 10^{-8} on Case 3). This comparison spans different input regimes—PINN-WE consumes only initial and boundary conditions—so the data-error reduction reflects the combined effect of architecture and the richer multi-fidelity input rather than a like-for-like algorithmic comparison. Under the unified discrete operator PIMUG's residual (1.00×10^{2}) is **25 times lower** than PINN-WE (2.51×10^{3}), replacing instance-by-instance retraining with amortized inference.

Demystifying the FNO comparison. FNO's lower raw MSE_{data} reflects a Pareto trade-off between unweighted L_2 fitting and physical structure preservation: pixelwise MSE is dominated by low-frequency smooth regions and tolerant to edge blurring, whereas PIMUG trades a small amount of pixelwise accuracy for physically consistent structure, because its residual and gradient terms penalize unphysical drift and blurred fronts rather than low-frequency amplitude error.

Under the unified discrete operator the comparison must be read against the per-case reference floor (Table 4): in Cases 1–2 the floor is low (0.97, 4.06) and FNO's values (4.69, 4.46) genuinely track the ground truth, while PIMUG's higher residuals (1.00×10^{2}, 1.19×10^{1}) reflect genuine high-frequency error from its sharply reconstructed fronts—a limitation we report openly; in Case 3, where shock-induced $O(1/\Delta x)$ differencing spikes elevate the floor (82.3), both surrogates score below it (74.6 FNO, 70.0 PIMUG), so the discrete residual cannot rank the two methods in this regime. Total variation is complementary: FNO under-predicts the density variation (269.6 vs. reference 278.1), consistent with spectral smoothing, whereas PIMUG slightly over-predicts it (291.6; Appendix A.4). Overall, PIMUG preserves sharp shock transitions confined to 2–3 cells, self-consistent continuous constraints, and fine vortex roll-ups without visible Gibbs ripples.

Isolating the architectural contribution: a physics-augmented FNO. To separate the effect of the residual loss from that of the architecture we train a physics-augmented FNO: an identical spectral operator on the identical 16-channel input, with the unified discrete Euler residual (the evaluation operator of Table 4) appended to the data loss at unit weight—the same nominal setting as PIMUG's four loss terms. Training converges as cleanly as the data-only FNO (test loss 1.49×10^{-2} versus 1.34×10^{-4}), yet at the $+5\%$ perturbation R_{disc} collapses from 4.69 to 7.32×10^{-3}, two orders of magnitude *below the WENO reference floor itself* (0.97). This apparently excellent value is a degenerate optimum, not an accurate solution: the discrete Euler operator has a large null space of smooth fields, and since the residual is evaluated on the model's own predicted trajectory with no mechanism anchoring high-frequency structure, the optimizer lowers it by over-smoothing rather than by reproducing the flow. The data error therefore rises from 8.04×10^{-5} to 8.34×10^{-3} ($\sim 100\times$) and the trunk-gradient error from 1.45×10^{-5} to 1.16×10^{-3} ($\sim 80\times$), while the total variation is essentially unchanged (1145 against 1134 for the reference); the error is not a global bias but a displacement concentrated on the discontinuities (mean signed density bias -0.017, whereas $\max|\rho - \rho_{\text{ref}}|$ grows from 0.12 to 0.58), which Figure 3 localizes along the very shocks and slip lines the residual tolerance is meant to protect. Full per-perturbation results are in Table 5 (Appendix A.4). Two conclusions follow: (1) a spectral operator given the same physical residual cannot reproduce PIMUG—it improves consistency only by forfeiting data accuracy, whereas PIMUG achieves both simultaneously; and (2) a non-reference field can drive raw residuals below the discretization floor, so low discrete residuals alone do not certify physical accuracy.

5.3 Robustness, Ablation, and Cross-Case Transfer

Robustness across perturbations. Appendix A.4 shows that PIMUG remains stable across -5% to $+5\%$: its data error stays within 4.5×10^{-5} to 2.6×10^{-3} and its residual order is stable within each case, whereas FNO achieves small in-distribution interpolation errors but lacks physical constraints against unphysical drift on complex discontinuity topologies.

Ablation study. All ablation variants are evaluated on the Case 1 test set under identical partitions and protocols (Appendix A.4). PIMUG attains the lowest data error at every tested perturbation: at $+3\%$, its error (4.39×10^{-4}) is about 35% below the isolated U-Mamba trunk (6.77×10^{-4}) and 38% below the ICM-augmented trunk (7.03×10^{-4}), which itself reduces error by 62.6% over the plain U-Net (1.88×10^{-3}). The physics-only variant attains the lowest Euler residual but at a data error 2.3 to 11 times higher, and PIMUG alone attains low data error, low residual, and the

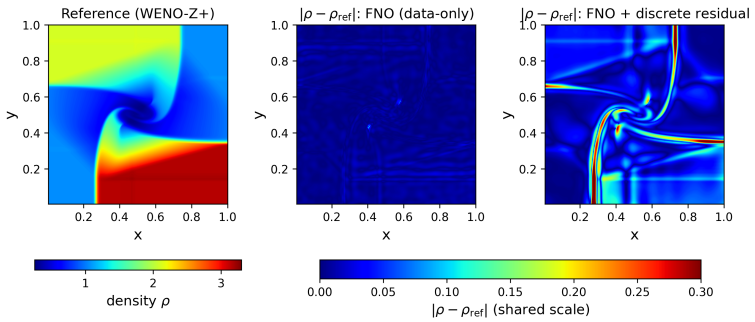


Figure 3: Case 1 density at $+5\%$ perturbation. Left: WENO-Z+ reference. Middle and right: absolute error $|\rho - \rho_{\text{ref}}|$ of data-only and physics-augmented FNOs on a shared scale. The residual penalty over-smooths shocks and slip lines: the degenerate solution quantified in Table 5 (Appendix A.4).

lowest trunk-gradient error at every perturbation, consistent with the intended division of labor: the gradient stream supplies feature-level evidence whose benefit emerges in combination with the trunk and physics branches. Single-run point estimates are reported; narrow ablation margins may fall within optimization variance, motivating future multi-seed runs.

Transfer to Case 4. PIMUG is pretrained on Cases 1–3 and then fine-tuned for 150 epochs at learning rate 8×10^{-5} on the held-out Case 4 with the trunk encoder, gradient encoder, and physics branch frozen. The model recovers the dominant slip-line structures visually, indicating that low-level multi-scale features are partially reusable across distinct flow configurations; however, this reusability is strictly limited to field appearance. As detailed in Appendix A.4, under this frozen-encoder regime the continuous energy residual degrades to $O(1)$ (Table 9), orders of magnitude worse than the PINN-WE baseline trained from scratch on Case 4; that is, the adapted model does not satisfy conservation laws at an acceptable level. The transfer study therefore carries a deliberately negative result: decoder-only fine-tuning trades physical consistency for rapid spatial alignment, and restoring strict PDE consistency under substantial wave-system topology shifts motivates future exploration of lightweight adapters or partial encoder fine-tuning.

6 Discussion and Conclusion

PIMUG addresses discontinuous flow prediction by harmonizing multi-fidelity learning, explicit gradient guidance, and feature-level physical conditioning. On 2D Riemann problems, this multi-branch architecture reduces Euler residuals relative to PINN-WE by two to four orders of magnitude while delivering millisecond amortized inference. Comparisons with data-driven and physics-augmented FNOs confirm that data fitting alone fails to enforce governing equations, while naively penalizing residuals in spectral operators merely induces degenerate over-smoothing; PIMUG's gains are thus architectural rather than a loss artifact. Case 1 ablations validate the synergy of all three branches, while the Case 4 transfer study reveals that frozen-encoder adaptation preserves visual fronts but degrades continuous energy residuals to $O(1)$—a deliberately negative result motivating future exploration of lightweight adapters. Future work includes conservation-preserving transfer strategies, per-problem residual weighting, graph-based PDE solvers (Brandstetter et al., 2022; Pfaff et al., 2021), community-standard benchmarks like PDEBench (Takamoto et al., 2022), realistic aerodynamic meshes (Pfaff et al., 2021), and extensions to viscous, three-dimensional, and multiphase flows.

AI USE STATEMENT

In this work, generative AI tools were used for language editing (grammar, clarity, and wording) and manuscript organization. They were not used to generate experimental data, implement the scientific method, or draw the scientific conclusions. The authors reviewed all AI-assisted text and take full responsibility for the final content of this work.

ETHICS STATEMENT

This study uses synthetic numerical simulations of classical compressible-flow test problems and does not involve human subjects, personal data, or deployment in safety-critical systems. Learned surrogates should not replace validated numerical solvers or physical review in safety-critical applications without additional verification.

REPRODUCIBILITY STATEMENT

Appendix A.1 defines the numerical schemes, the four Riemann cases with their full quadrant states, and the perturbation protocol; Appendix A.2 gives the complete architecture, optimization, input pipeline, and training protocol, including the channel layout of the multi-fidelity input; and Appendix A.3 describes baseline implementations and defines all reported metrics with their per-method evaluation operators. These specifications are designed to be sufficient to re-implement the experiments, and an anonymized source distribution containing the model, training, evaluation, and data-generation code accompanies the submission as supplementary material. Each reported number comes from a single training run per configuration; repeated runs under varying random seeds were not performed, so the values should be read as single-run estimates. We acknowledge that single-run point estimates cannot resolve the narrow gaps observed in some ablation comparisons (Table 7), which may fall within the variance of stochastic optimization; multi-seed repetitions reporting mean and standard deviation are planned for the revision. Full quantitative results and qualitative examples are provided in Appendices A.4–A.5.

REFERENCES

Felipe Acker, Rafael Borges, and Bruno Costa. An improved WENO-Z scheme. *Journal of Computational Physics*, 313:726–753, 2016. doi: 10.1016/j.jcp.2016.01.038.

Rafael Borges, Monique Carmona, Bruno Costa, and Wai-Sun Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227(6):3191–3211, 2008. doi: 10.1016/j.jcp.2007.11.038.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Ping Cao, Shuran Lin, Yanwu Yang, and Chunjie Zhang. Dual-branch interactive guided network based on gradient prior for image super-resolution. *Journal of Visual Communication and Image Representation*, 113:104607, 2025. doi: 10.1016/j.jvcir.2025.104607.

Yong-Sik Cho and Sung Bum Yoon. A modified leap-frog scheme for linear shallow-water equations. *Coastal Engineering Journal*, 40(2):191–205, 1998. doi: 10.1142/S0578563498000121.

Bernardo Cockburn and Chi-Wang Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998. doi: 10.1137/S0036142997316712.

Emilio Jose Rocha Coutinho, Marcelo Dall'Aqua, Levi McClenny, Ming Zhong, Ulisses Braga-Neto, and Eduardo Gildin. Physics-informed neural networks with adaptive localized artificial viscosity. *Journal of Computational Physics*, 489:112265, 2023. doi: 10.1016/j.jcp.2023.112265.

Sergei Konstantinovich Godunov. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Matematicheskii Sbornik*, 47(3):271–306, 1959.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Jayesh K. Gupta and Johannes Brandstetter. Towards Multi-Spatiotemporal-Scale Generalized PDE Modeling. *Transactions on Machine Learning Research*, 2023.

Ami Harten. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM Journal on Numerical Analysis*, 21(1):1–23, 1984. doi: 10.1137/0721001.

Haoxiang Huang, Vigor Yang, and Yingjie Liu. Neural networks with local converging inputs (NNLCI) for solving conservation laws, Part I: 1D problems. *Communications in Computational Physics*, 34(2):290–317, 2023. doi: 10.4208/cicp.oa-2022-0285.

Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996. doi: 10.1006/jcph.1996.0130.

Peter D. Lax and Xu-Dong Liu. Solution of Two-Dimensional Riemann Problems of Gas Dynamics by Positive Schemes. *SIAM Journal on Scientific Computing*, 19(2):319–340, 1998. doi: 10.1137/S1064827595291819.

Jiafeng Li, Ying Wen, and Lianghua He. SCConv: Spatial and channel reconstruction convolution for feature redundancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6153–6162, 2023. doi: 10.1109/CVPR52729.2023.00596.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

Dehao Liu and Yan Wang. Multi-Fidelity Physics-Constrained Neural Network and its Application in Materials Modeling. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume Volume 2A: 45th Design Automation Conference, pp. V02AT03A007, 2019. doi: 10.1115/DETC2019-98115.

Li Liu, Shengping Liu, Hui Xie, Fansheng Xiong, Tengchao Yu, Mengjuan Xiao, Lufeng Liu, and Heng Yong. Discontinuity computing using physics-informed neural networks. *Journal of Scientific Computing*, 98(1):22, 2024a. doi: 10.1007/s10915-023-02412-1.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103031–103063, 2024b. doi: 10.52202/079017-3273.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7766–7775, 2020. doi: 10.1109/CVPR42600.2020.00779.

Jun Ma, Feifei Li, and Bo Wang. U-Mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Robert W. MacCormack. The Effect of Viscosity in Hypervelocity Impact Cratering. In *AIAA 4th Aerodynamic Testing Conference / Hypervelocity Impact Conference*, pp. AIAA Paper 69–354, 1969. doi: 10.2514/6.1969-354.

Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 401:109020, 2020. doi: 10.1016/j.jcp.2019.109020.

Vivek Oommen, Khemraj Shukla, Saaketh Desai, Rémi Dingreville, and George Em Karniadakis. Rethinking materials simulations: Blending direct numerical simulations with neural operators. *npj Computational Materials*, 10(1):145, 2024. doi: 10.1038/s41524-024-01319-1.

Oded Ovadia, Vivek Oommen, Adar Kahana, Ahmad Peyvan, Eli Turkel, and George Em Karniadakis. Real-time inference and extrapolation via a diffusion-inspired temporal transformer operator (DITTO). *arXiv preprint arXiv:2307.09072*, 2023.

Ahmad Peyvan, Vivek Oommen, Ameya D. Jagtap, and George Em Karniadakis. RiemannONets: Interpretable neural operators for Riemann problems. *Computer Methods in Applied Mechanics and Engineering*, 426:116996, 2024. doi: 10.1016/j.cma.2024.116996.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Maziar Raissi, Paris Perdikaris, and George 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*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 66899–66915, 2023.

Jian Sun, Zongben Xu, and Heung-Yeung Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE Transactions on Image Processing*, 20(6):1529–1542, 2011. doi: 10.1109/TIP.2010.2095871.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022.

Wenhan Yang, Jiashi Feng, Jianchao Yang, Fang Zhao, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing*, 26(12):5895–5907, 2017. doi: 10.1109/TIP.2017.2750403.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022. doi: 10.1016/j.cma.2022.114823.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

Xiancheng Zhu, Detian Huang, Xiaorui Li, Danlin Cai, and Daxin Zhu. GPSR: Gradient-prior-based network for image super-resolution. *Applied Sciences*, 13(2):833, 2023. doi: 10.3390/app13020833.

A Appendix

A.1 Numerical Schemes and Case Descriptions

Low-fidelity scheme. The low-fidelity solutions are computed with a Leapfrog-based two-step scheme (MacCormack, 1969; Cho & Yoon, 1998). A predictor step advances the state, and a corrector step introduces numerical dissipation proportional to the grid spacing, yielding first-order spatial accuracy. On the coarse 200×200 grid the solver takes 800 steps over the case horizon ($\Delta t = T/800$) and stores every step; the fine 400×400 solver takes 1600 steps at half the step size and stores every second step after resampling to 200×200, so both grids provide the same 800 aligned frames.

High-fidelity scheme. Reference solutions are generated with the seventh-order WENO-Z+ scheme (Acker et al., 2016), an improved variant of WENO-Z (Borges et al., 2008). The nonlinear weights are modified by a global-smoothness term of the form $\Delta x^P/(\beta_k + \epsilon)$ in addition to the WENO-Z smoothness indicator τ_7/β_k, which reduces dissipation near discontinuities. Time integration uses the third-order strong-stability-preserving Runge–Kutta scheme with 2400 steps, of which every third step is stored, again yielding 800 frames per run.

Two-dimensional Riemann cases. All cases are defined on $[0, 1]^2$, divided into four quadrants by $x = 0.5$ and $y = 0.5$ (Lax & Liu, 1998). Both the Leapfrog and WENO-Z+ solvers apply zero-gradient (transmissive/outflow) boundary conditions, implemented by copying the adjacent interior cell into each boundary cell; the same boundary data are supplied to the PINN-WE baseline. This unconfined setting lets waves leave the domain without reflection. Table 2 lists the initial quadrant states (ρ, u, v, p) of all four cases together with the evolution horizon T.

Table 2: Initial quadrant states (ρ, u, v, p) and evolution horizon T of the four Riemann cases, following the configuration catalogue of Lax & Liu (1998).

Case	Upper-left	Upper-right	Lower-left	Lower-right	T
Case 1 (Config. 6)	(2.0, 0.75, 0.5, 1.0)	(1.0, 0.75, −0.5, 1.0)	(1.0, −0.75, 0.5, 1.0)	(3.0, −0.75, −0.5, 1.0)	0.3
Case 2 (Config. 8)	(1.0, −0.6259, 0.1, 1.0)	(0.5197, 0.1, 0.1, 0.4)	(0.8, 0.1, 0.1, 1.0)	(1.0, 0.1, −0.6259, 1.0)	0.2
Case 3 (Config. 11)	(0.5313, 0.8276, 0.0, 0.4)	(1.0, 0.1, 0.0, 1.0)	(0.8, 0.1, 0.0, 0.4)	(0.5313, 0.1, 0.7276, 0.4)	0.3
Case 4 (Config. 4)	(0.5065, 0.8939, 0.0, 0.35)	(1.1, 0.0, 0.0, 1.1)	(1.1, 0.8939, 0.8939, 1.1)	(0.5065, 0.0, 0.8939, 0.35)	0.3

Wave-structure summary. Case 1 (Configuration 6) is a pure slip-line case whose four alternating contact discontinuities develop complex vortical structures. Case 2 (Configuration 8) couples two negative slip lines with two rarefaction waves, the low pressure in the first quadrant inducing expansion. Case 3 (Configuration 11) contains two shocks and two slip lines with large pressure jumps and velocity gradients. Case 4 (Configuration 4) is a held-out pure slip-line case with four negative contact discontinuities, used only for transfer learning.

A.2 Network and Training Configuration

The trunk and gradient branches use symmetric four-stage encoder–Mamba–decoder architectures. Feature channels are 16, 32, 64, and 128; kernel sizes are 3, 5, 5, and 5. Each encoding stage contains two residual convolutional blocks, except the two coarsest stages, which contain one; each decoding stage contains two blocks, except the finest stage, which contain one; all blocks use LeakyReLU activations. Downsampling uses a stride-2 3×3 convolution; upsampling applies nearest-neighbor interpolation with the matching scale factor of 2 followed by a 1×1 convolution, which avoids checkerboard artifacts. Each Mamba bottleneck block applies LayerNorm followed by a selective state-space model with state dimension $d_{\text{state}} = 16$, convolution width $d_{\text{conv}} = 4$, and expansion factor 2; the 2D feature map is flattened into a 1D sequence by a unidirectional row-major (raster) scan, and no bidirectional or multi-directional cross-scan is used. The physics branch is an MLP followed by a four-stage residual convolutional encoder. The ICM of each branch reduces the 16 input channels to 8 before the corresponding encoder.

In the 16-channel input, channels are grouped by time level: each time level contributes, in the primitive variable order (ρ, u, v, p), a fine–coarse channel pair per variable, i.e.,

$[\text{fine}_\rho^{t-1}, \text{coarse}_\rho^{t-1}, \ldots, \text{fine}_p^{t-1}, \text{coarse}_p^{t-1}]$, followed by the same eight channels at level t, where "fine" and "coarse" denote the resampled 400×400 and native 200×200 Leapfrog solutions. At the first frame the two time levels coincide, so the earlier block duplicates the initial frame. The scalar time coordinate τ fed to the physics branch is normalized uniformly across all benchmarks as $\tau \in [0, \tau_{\max}]$ (discretized uniformly into 800 steps with $\tau_{\max} = 0.3$), serving as a standardized dimensionless temporal coordinate across cases. This dimensionless parameterization ensures an identical temporal feature scale for the shared MLP physics conditioning across diverse Riemann evolution horizons while leaving the physical input fields and prediction targets unchanged. When evaluating temporal derivatives with respect to physical time $t_{\text{phys}} \in [0, T]$, the chain rule provides the exact Jacobian scaling factor $\frac{\partial}{\partial t_{\text{phys}}} = \frac{\tau_{\max}}{T} \frac{\partial}{\partial \tau}$ ($\tau_{\max}/T = 1.5$ for Case 2 where $T = 0.2$, and 1.0 for Cases 1, 3, and 4), ensuring dimensional consistency. Training uses Adam with initial learning rate 5×10^{-4} and batch size 5. If the total loss does not improve for five epochs, the learning rate is multiplied by 0.7. The maximum number of epochs is 1000. The physics loss is evaluated every three epochs to reduce the cost of automatic differentiation; data and gradient losses are evaluated at every epoch.

Computational cost. PIMUG has 3.54M parameters and requires 4.6 ms per 200×200 frame at inference (batch size 1, RTX 4090); FNO has 4.75M parameters and requires 0.7 ms per frame under the same setting. PIMUG is thus the smaller model but roughly six times slower per frame—the price of the multi-branch architecture and the sequential Mamba bottleneck—while both models remain orders of magnitude faster than the numerical solvers that generated their evaluation references.

Input pipeline and ICM details. Figure 4 shows the multi-fidelity input pipeline, and Figure 5 illustrates the two partitioning criteria applied by the cascaded submodules of the ICM, complementing the submodule architecture presented in Section 4.2.

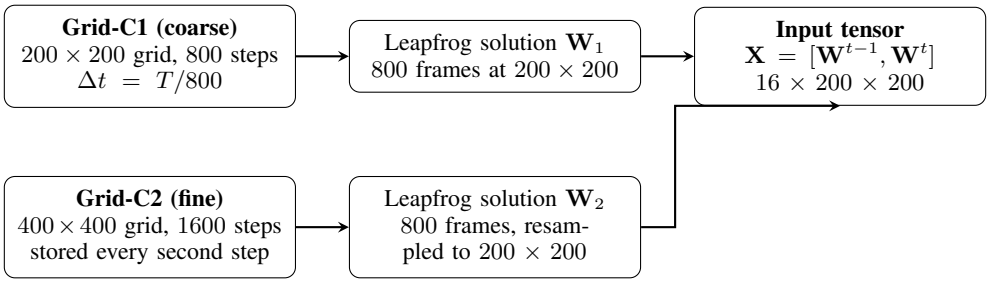
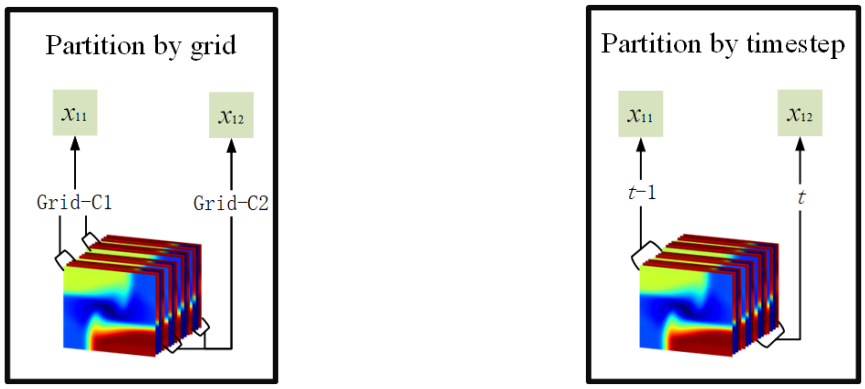


Figure 4: Construction of the multi-fidelity input. Both grid levels store 800 aligned frames of primitive variables (ρ, u, v, p); each sample stacks fine–coarse pairs across adjacent time levels ($t-1$ and t) into a 16-channel tensor as in Eq. (3).



(a) Partition by spatial grid/fidelity.

(b) Partition by timestep.

Figure 5: Two feature partitioning strategies in the ICM.

Notes on the adaptive weight λ_1. First, λ_1 is a bounded and slowly varying function of the compression term: with $k_1 = 0.12$ it stays above 0.5 unless $|\nabla \cdot \mathbf{u}_{\mathrm{nn}}| - \nabla \cdot \mathbf{u}_{\mathrm{nn}}$ exceeds $1/k_1 \approx 8.3$, so wherever the mechanism does engage it relaxes supervision only in a narrow band around strong compressive cores, while the states on both sides of a shock—which occupy $\lambda_1 \approx 1$ regions—remain fully anchored; a constant amplitude shift across a shock would incur a full data penalty outside the relaxed band. Measured behaviour, however, is more conservative than this mechanism suggests. The divergence that drives the weight is evaluated along the coordinate-conditioning path of the physics branch rather than as the grid-level divergence of the predicted field, and it stays small on our trained models: $\lambda_1 \in [0.997, 1.000]$ on every evaluated frame of Cases 1–3, and also at intermediate training stages (epochs 106, 243 and 445 of Case 1, against $[0.9972, 1.0000]$ for the converged model). The weight is therefore effectively inactive throughout optimization rather than only at convergence, the weighted and unweighted objectives coincide to within 0.05%, and no amplitude-drift or weight-inflation pathology can arise in practice. This is why we treat λ_1 as a safeguard inherited from PINN-WE rather than as a mechanism our results depend on, and why the R_{cont} values of PIMUG can be reported under the same unweighted scheme as PINN-WE (Appendix A.3). Second, λ_1 is computed from the predicted divergence and is *not* detached from the computation graph in our implementation, so gradients may flow through the weight itself; in principle the optimizer could exploit this coupling by inflating negative divergence to suppress both weighted terms, but no such degenerate solution was observed in any training run—all reported models converge to low data error with physically plausible divergence fields. Detaching λ_1 is a one-line safeguard, and a controlled ablation quantifying its effect is planned for the revision.

A.3 Baseline Implementation and Evaluation Metrics

Baseline implementations. For PINN-WE (Liu et al., 2024a), we use the MLP backbone and two-stage optimization from the original implementation: 10,000 Adam epochs followed by 1,000 L-BFGS epochs. As in the original design, a separate network is optimized from scratch for each initial-condition instance, consuming only coordinates and the initial and boundary data of that instance; no multi-fidelity field input is available to it. Evaluating all six test perturbations on all three cases would therefore require eighteen full re-trainings instead of three, so PINN-WE is reported at the representative $+5\%$ perturbation. Because PINN-WE has no access to multi-fidelity field inputs, its data-error comparison against PIMUG and FNO contrasts input regimes rather than algorithmic quality alone; this structural advantage of the multi-fidelity methods is acknowledged explicitly in Section 5. For FNO (Li et al., 2021), we use four Fourier layers with 12 Fourier modes per spatial dimension and width 64, trained with Adam at learning rate 10^{-3} for 500 epochs under the same plateau schedule. FNO receives the same 16-channel input as PIMUG and is trained with the same data loss and data partition. The physics-augmented variant uses the same network, input, optimizer, and schedule, and additionally minimizes a discontinuity-aware mean-squared discrete Euler residual on consecutive predicted frames—computed with the unified finite-difference operator of Appendix A.3 and the same divergence-adaptive weight form λ_1 used by PIMUG, at unit nominal weight—and is trained and reported on Case 1. For MPINN (Meng & Karniadakis, 2020), we adapt the composite multi-fidelity architecture (linear and nonlinear correlation paths: $\hat{\mathbf{W}} = \hat{g}_L(\mathbf{X}) + F_{H1}(\mathbf{X}) \odot \hat{g}_L(\mathbf{X}) + F_{H2}(\mathbf{X}) \odot \Delta_t \mathbf{X}_{\mathrm{LF}}$) from its original coordinate-pointwise MLP formulation into an amortized convolutional operator sharing the identical 16-channel multi-fidelity input, Adam optimizer, plateau schedule, and 500-epoch training protocol as FNO and PIMUG. Specifically, $\Delta_t \mathbf{X}_{\mathrm{LF}}$ extracts the temporal difference of the low-fidelity 200×200 grid solutions, while the projection layers of F_{H1} and F_{H2} are zero-initialized to stabilize optimization. This adaptation enables a rigorous, like-for-like comparison against PIMUG under identical multi-fidelity inputs and amortized inference settings.

Evaluation metrics. The data error is the sum over the four primitive variables (ρ, u, v, p) of the per-variable spatial mean-squared error between the predicted and reference fields. Reported residual values evaluate the Euler conservation equations (with the total-energy component isolated in Table 1 of the main text). Three residual evaluations coexist and are kept separate throughout. For PIMUG, the residual is the *unweighted* strong-form residual of Eq. (8) (the adaptive weight set to $\lambda_1 \equiv 1$), with time and space derivatives obtained by automatic differentiation through the coordinate feature inputs (x, y, t). We emphasize the scope of this quantity: the autograd derivatives treat the discrete multi-fidelity input grid $\mathbf{X}$ as constant and thus measure the sensitivity of the pre-

diction along the coordinate-conditioning path, not the total spatial derivative of the predicted field; the residual is the training objective itself and certifies self-consistency of the physics-enforcement mechanism rather than grid-level conservation, which is instead measured by the unified discrete residual below. For PINN-WE it is the unweighted autograd strong-form residual of its own MLP representation. The two physics-informed entries therefore share one weighting scheme—no adaptive weighting is applied to either—which is what makes the R_{cont} columns of Table 1 mutually comparable. We verified that the change is numerically negligible for PIMUG: over every evaluated frame of Cases 1–3 the trained models place λ_1 in $[0.997, 1.000]$, because the coordinate-path divergence that drives the weight stays small, so the λ_1-weighted and unweighted values agree to within 0.05%. For the FNO baseline and its physics-augmented variant, which have no coordinate inputs, we discretize Eqs. (1) and (2) with second-order central differences in space and first-order differences in time on the predicted trajectory. This paradigm asymmetry is deliberate, but the two physics-informed columns are now evaluated with identical (absent) weighting, so no cross-method deflation remains within the R_{cont} column; the unified discrete residual (Table 4) nevertheless provides the independent common ground for grid-level consistency. Appendix A.4 therefore discusses the comparison jointly with the residual floor of the reference solution. The trunk gradient error is computed with the fixed kernels of Eq. (4); the gradient-branch error is defined analogously for the dedicated branch output and is applicable only to models equipped with a gradient branch (PIMUG and the gradient-guided variant), just as it is not applicable to FNO. One property of these gradient metrics deserves note: because smoothing removes large gradient peaks, a fixed-kernel L_2 gradient error can favor over-smoothed predictions, so the gradient errors should be read jointly with the qualitative sections of Appendix A.5 rather than as standalone sharpness scores.

A.4 Extended Quantitative Results

Per-perturbation robustness. In Table 3, the Euler-residual column uses each method's own evaluation operator defined in Appendix A.3: the unweighted autograd residual for PIMUG and the unified finite-difference residual for FNO. The two blocks of rows are therefore not directly comparable to each other; the unified discrete-operator comparison follows below.

Interpreting discrete residuals. Table 4 places all surrogates and the WENO reference under one unified discrete operator—first-order time differences, second-order central differences in space, no weighting, evaluated on the final predicted frame and summed over the four conserved variables—on the $+5\%$ perturbation test sets. The PIMUG rows use the same single-frame evaluation protocol as all PIMUG results in Section 5; the operator and aggregation are identical to those used for FNO and PINN-WE. The $t-1$ state entering the first-order time difference is the surrogate's own predicted frame at $t-1$ (trajectory replay over the test set); the reference floor is computed analogously from consecutive WENO-Z+ frames, and the residual is evaluated from the second frame onward.

Two observations follow from this unified comparison. First, in Case 1 PIMUG reduces the discrete residual by a factor of 25 relative to PINN-WE (1.00×10^2 versus 2.51×10^3), consistent with its 13.3-fold data error reduction (1.40×10^{-3} versus 1.86×10^{-2}), demonstrating superior discrete fidelity over point-wise physics-informed formulations. Furthermore, benchmarking the multi-fidelity baseline MPINN (Table 4 and Table 6) isolates the role of the multi-fidelity data input versus PIMUG's architectural mechanisms. Despite consuming the identical 16-channel coarse/fine input pairs as PIMUG, MPINN yields a discrete Euler residual of 2.71×10^3 at $+5\%$ (and up to 4.76×10^3 at -3%)—virtually identical to the single-fidelity PINN-WE (2.51×10^3). PIMUG reduces this residual by a factor of 27 (1.00×10^2). Moreover, MPINN's trunk gradient error is up to $5.2\times$ worse than PIMUG on challenging negative perturbations (e.g., 3.02×10^{-4} vs. 5.82×10^{-5} at -3%). While MPINN achieves small pixelwise data error by focusing entirely on unweighted L_2 regression in smooth regions, it severely violates governing conservation laws on the grid and softens discontinuity fronts. This controlled comparison confirms that multi-fidelity data alone cannot guarantee physical consistency or sharp wave resolution without PIMUG's dedicated gradient branch and feature-level physical regularization.

In Cases 1–2, FNO's discrete residuals (4.69 and 4.46) track the low reference floors (0.97 and 4.06), whereas PIMUG's higher values (1.00×10^2 and 1.19×10^1) reflect genuine high-frequency discrete error from its sharply reconstructed fronts; by contrast, in Case 3, where the elevated floor (82.3) stems from $O(1/\Delta x)$ differencing spikes across genuine shocks, both surrogates score be-

Table 3: Prediction errors of PIMUG and FNO for Cases 1–3 under all six test perturbations. Each method's Euler residual is evaluated with its own operator defined in Appendix A.3 (unweighted continuous autograd residual for PIMUG, unified finite-difference residual for FNO); reported values are aggregated over all four conservation equations.

Case	Method	Perturbation	Data error	Euler residual	Trunk gradient error	Branch gradient error
Case 1	PIMUG	-5%	1.70×10^{-3}	6.78×10^{-5}	7.95×10^{-5}	2.02×10^{-3}
		-3%	6.03×10^{-4}	7.16×10^{-5}	5.82×10^{-5}	2.06×10^{-3}
		-1%	3.61×10^{-4}	8.69×10^{-5}	5.14×10^{-5}	2.11×10^{-3}
		1%	5.01×10^{-4}	9.19×10^{-5}	5.42×10^{-5}	2.18×10^{-3}
		3%	4.39×10^{-4}	8.49×10^{-5}	6.21×10^{-5}	2.27×10^{-3}
		5%	1.40×10^{-3}	7.68×10^{-5}	8.39×10^{-5}	2.37×10^{-3}
	FNO	-5%	2.99×10^{-4}	1.64	1.82×10^{-5}	–
		-3%	1.31×10^{-4}	1.47	1.35×10^{-5}	–
		-1%	1.31×10^{-4}	1.47	1.07×10^{-5}	–
		1%	1.12×10^{-4}	1.68	8.65×10^{-6}	–
		3%	4.45×10^{-5}	2.46	8.66×10^{-6}	–
		5%	8.04×10^{-5}	4.69	1.45×10^{-5}	–
Case 2	PIMUG	-5%	2.60×10^{-3}	9.60×10^{-7}	5.20×10^{-5}	7.06×10^{-4}
		-3%	1.39×10^{-3}	9.53×10^{-7}	3.85×10^{-5}	7.26×10^{-4}
		-1%	6.52×10^{-4}	9.69×10^{-7}	3.19×10^{-5}	7.50×10^{-4}
		1%	3.68×10^{-4}	1.04×10^{-6}	3.25×10^{-5}	7.81×10^{-4}
		3%	5.65×10^{-4}	1.15×10^{-6}	4.00×10^{-5}	8.18×10^{-4}
		5%	1.28×10^{-3}	1.28×10^{-6}	5.82×10^{-5}	8.61×10^{-4}
	FNO	-5%	1.08×10^{-4}	2.49	1.92×10^{-6}	–
		-3%	6.43×10^{-5}	2.50	1.25×10^{-6}	–
		-1%	4.30×10^{-5}	2.80	1.04×10^{-6}	–
		1%	2.49×10^{-5}	3.32	1.00×10^{-6}	–
		3%	7.09×10^{-6}	3.90	1.00×10^{-6}	–
		5%	9.71×10^{-6}	4.46	1.33×10^{-6}	–
Case 3	PIMUG	-5%	2.98×10^{-4}	1.51×10^{-7}	1.94×10^{-5}	1.92×10^{-3}
		-3%	7.55×10^{-5}	1.25×10^{-7}	1.28×10^{-5}	1.99×10^{-3}
		-1%	5.26×10^{-5}	1.18×10^{-7}	1.09×10^{-5}	2.08×10^{-3}
		1%	4.46×10^{-5}	1.15×10^{-7}	1.17×10^{-5}	2.14×10^{-3}
		3%	6.15×10^{-5}	1.31×10^{-7}	1.32×10^{-5}	2.27×10^{-3}
		5%	4.26×10^{-4}	1.43×10^{-7}	2.30×10^{-5}	2.32×10^{-3}
	FNO	-5%	1.19×10^{-4}	3.78×10^{1}	2.96×10^{-6}	–
		-3%	2.70×10^{-5}	7.31×10^{1}	1.47×10^{-6}	–
		-1%	2.75×10^{-5}	2.45×10^{1}	1.48×10^{-6}	–
		1%	3.74×10^{-5}	8.36×10^{1}	1.91×10^{-6}	–
		3%	1.62×10^{-5}	3.20×10^{1}	2.22×10^{-6}	–
		5%	2.99×10^{-5}	7.46×10^{1}	3.92×10^{-6}	–

Table 4: Data error and unified discrete-operator Euler residual on the $+5\%$ perturbation test sets (final frame, summed over the four conserved variables). The reference floor is the residual of the WENO solution itself under the same operator.

Case	Method	Data error	Discrete residual
Case 1	PINN-WE	1.86×10^{-2}	2.51×10^{3}
	MPINN	2.34×10^{-4}	2.71×10^{3}
	PIMUG	1.40×10^{-3}	1.00×10^{2}
	FNO	8.04×10^{-5}	4.69
	FNO (physics-aug.)	8.34×10^{-3}	7.32×10^{-3}
	Reference floor	–	0.97
Case 2	PIMUG	1.28×10^{-3}	1.19×10^{1}
	FNO	9.71×10^{-6}	4.46
	Reference floor	–	4.06
Case 3	PIMUG	4.26×10^{-4}	7.00×10^{1}
	FNO	2.99×10^{-5}	7.46×10^{1}
	Reference floor	–	8.23×10^{1}

Table 5: Isolating the architectural contribution on the Case 1 test perturbations: the data-only FNO versus an identical FNO trained with the unified discrete Euler residual at unit weight (physics-augmented). Adding the residual lowers R_{disc} by two to three orders of magnitude—at $+5\%$ it falls below the WENO reference floor (0.97)—but degrades the data and trunk-gradient errors by roughly two orders of magnitude, because the residual is reduced by over-smoothing rather than by matching the flow. Operators and evaluation protocol are identical to Tables 1 and 4; lower is better.

	FNO (data-only)			**FNO (physics-augmented)**		
Perturbation	Data error	R_{disc}	Trunk grad	Data error	R_{disc}	Trunk grad
-5%	2.99×10^{-4}	1.64	1.82×10^{-5}	2.28×10^{-2}	1.88×10^{-1}	8.55×10^{-4}
-3%	1.31×10^{-4}	1.47	1.35×10^{-5}	1.47×10^{-2}	1.15×10^{-1}	8.43×10^{-4}
-1%	1.31×10^{-4}	1.47	1.07×10^{-5}	9.12×10^{-3}	6.25×10^{-2}	8.67×10^{-4}
1%	1.12×10^{-4}	1.68	8.65×10^{-6}	6.14×10^{-3}	2.43×10^{-2}	9.27×10^{-4}
3%	4.45×10^{-5}	2.46	8.66×10^{-6}	5.85×10^{-3}	4.42×10^{-3}	1.02×10^{-3}
5%	8.04×10^{-5}	4.69	1.45×10^{-5}	8.34×10^{-3}	7.32×10^{-3}	1.16×10^{-3}

Table 6: Evaluation of the multi-fidelity MPINN baseline on the Case 1 test perturbations under the shared 16-channel input regime. Without explicit physical constraints or gradient guidance, the composite architecture achieves small pixelwise data error by fitting smooth regions but incurs severe governing-equation violations ($R_{\text{disc}} \sim 10^3$, $27\times$ higher than PIMUG) and degraded gradient sharpness. Operators and evaluation protocol are identical to Table 3; lower is better.

Perturbation	Data error	R_{disc}	Trunk gradient error
-5%	6.06×10^{-4}	2.80×10^{3}	2.63×10^{-4}
-3%	5.24×10^{-4}	4.76×10^{3}	3.02×10^{-4}
-1%	2.24×10^{-4}	1.24×10^{3}	9.47×10^{-5}
1%	9.95×10^{-5}	1.33×10^{3}	3.94×10^{-5}
3%	1.31×10^{-4}	1.76×10^{3}	5.91×10^{-5}
5%	2.34×10^{-4}	2.71×10^{3}	1.19×10^{-4}

low the floor (74.6 and 70.0), so the discrete residual cannot rank the two methods in this regime and we do not claim superiority from these numbers. The Case 1 block also lists the physics-augmented FNO of Section 5, whose full per-perturbation results are given in Table 5: at $+5\%$ its residual (7.32×10^{-3}) falls far *below* the reference floor itself, exactly the degenerate over-smoothed regime analyzed there, which shows that this column alone cannot certify a correct prediction. Total variation analysis provides a complementary check. The TV of the density field is the anisotropic discrete sum $\mathrm{TV}(\rho) = \sum_{i,j}\big(|\rho_{i+1,j} - \rho_{i,j}| + |\rho_{i,j+1} - \rho_{i,j}|\big)$ over the 200×200 grid. FNO under-predicts the density variation (269.6 versus reference 278.1), consistent with spectral smoothing. PIMUG's TV (291.6) exceeds the reference by about 5%; in a hyperbolic problem, TV above the high-fidelity reference indicates small new extrema, i.e., slight grid-scale dispersive overshoots near the reconstructed fronts, so PIMUG is not strictly TV-diminishing and we do not claim it to be. "Oscillation-free" throughout this paper refers to the absence of visible Gibbs-type ripples in the rendered profiles and field maps (Appendix A.5), not to a strict TVD guarantee. Overall, raw discrete central-difference residuals certify grid-level conservation but are not a monotonic ranking of physical correctness; we therefore read them jointly with the continuous residuals, TV, and qualitative profiles.

Full ablation results. Table 7 reports the complete ablation grid on the Case 1 test perturbations. The trunk-gradient error is evaluated with the fixed kernels of Eq. (4) under the protocol of Section 5; the branch-gradient error applies only to the variants equipped with a gradient branch.

Table 7: Complete ablation results on the Case 1 test perturbations: positive (top) and negative (bottom). The Euler-residual column aggregates the residual over all four conservation equations (for PIMUG under its unweighted operator as defined in Appendix A.3).

Perturbation	Model	Data error	Euler residual	Trunk gradient error	Branch gradient error
1%	U-Net	8.11×10^{-4}	–	7.66×10^{-5}	–
	Gradient-guided U-Net	1.05×10^{-3}	–	6.68×10^{-5}	2.67×10^{-3}
	ICM U-Net	1.01×10^{-3}	–	8.99×10^{-5}	–
	Physics U-Net	1.59×10^{-3}	2.06×10^{-5}	9.11×10^{-5}	–
	U-Mamba	2.21×10^{-3}	–	8.71×10^{-5}	–
	PIMUG	5.01×10^{-4}	9.19×10^{-5}	5.42×10^{-5}	2.18×10^{-3}
3%	U-Net	1.88×10^{-3}	–	9.83×10^{-5}	–
	Gradient-guided U-Net	9.26×10^{-4}	–	7.68×10^{-5}	2.75×10^{-3}
	ICM U-Net	7.03×10^{-4}	–	1.00×10^{-4}	–
	Physics U-Net	1.00×10^{-3}	1.73×10^{-5}	9.65×10^{-5}	–
	U-Mamba	6.77×10^{-4}	–	8.38×10^{-5}	–
	PIMUG	4.39×10^{-4}	8.49×10^{-5}	6.21×10^{-5}	2.27×10^{-3}
5%	U-Net	6.12×10^{-3}	–	1.54×10^{-4}	–
	Gradient-guided U-Net	3.98×10^{-3}	–	1.14×10^{-4}	2.83×10^{-3}
	ICM U-Net	2.25×10^{-3}	–	1.44×10^{-4}	–
	Physics U-Net	3.59×10^{-3}	1.55×10^{-5}	1.34×10^{-4}	–
	U-Mamba	2.12×10^{-3}	–	1.14×10^{-4}	–
	PIMUG	1.40×10^{-3}	7.68×10^{-5}	8.39×10^{-5}	2.37×10^{-3}
-1%	U-Net	2.65×10^{-3}	–	8.58×10^{-5}	–
	Gradient-guided U-Net	4.15×10^{-3}	–	8.99×10^{-5}	2.59×10^{-3}
	ICM U-Net	1.28×10^{-3}	–	7.97×10^{-5}	–
	Physics U-Net	4.01×10^{-3}	3.13×10^{-5}	1.06×10^{-4}	–
	U-Mamba	4.40×10^{-3}	–	9.02×10^{-5}	–
	PIMUG	3.61×10^{-4}	8.69×10^{-5}	5.14×10^{-5}	2.11×10^{-3}
-3%	U-Net	7.01×10^{-3}	–	1.16×10^{-4}	–
	Gradient-guided U-Net	1.01×10^{-2}	–	1.43×10^{-4}	2.51×10^{-3}
	ICM U-Net	1.18×10^{-3}	–	8.22×10^{-5}	–
	Physics U-Net	6.25×10^{-3}	5.76×10^{-5}	1.26×10^{-4}	–
	U-Mamba	4.08×10^{-3}	–	8.96×10^{-5}	–
	PIMUG	6.03×10^{-4}	7.16×10^{-5}	5.82×10^{-5}	2.06×10^{-3}
-5%	U-Net	1.31×10^{-2}	–	1.65×10^{-4}	–
	Gradient-guided U-Net	1.93×10^{-2}	–	2.25×10^{-4}	2.45×10^{-3}
	ICM U-Net	2.05×10^{-3}	–	9.57×10^{-5}	–
	Physics U-Net	7.30×10^{-3}	3.29×10^{-5}	1.50×10^{-4}	–
	U-Mamba	5.53×10^{-3}	–	1.19×10^{-4}	–
	PIMUG	1.70×10^{-3}	6.78×10^{-5}	7.95×10^{-5}	2.02×10^{-3}

Full transfer results on Case 4. Table 8 reports the transfer-learning errors of PIMUG on Case 4 across all perturbations. The perturbation grid includes the nominal 0% condition, which is unseen for Case 4 (it belongs to the training set of Cases 1–3 only). The transferred model recovers dominant structures quickly, but the frozen encoders cannot adjust to the feature distribution of the new case, so decoder fine-tuning improves field reconstruction without restoring the residual quality obtained by full training. One implementation detail is worth recording: the adaptive weight does engage in this regime. Because the frozen encoders leave the predicted velocity field far from the distribution of the new case, the coordinate-path divergence that drives λ_1 is no longer negligible, and the weight drops to $\lambda_1 \in [0.36, 0.49]$ with 2.7–7.7% of grid cells below 0.9—in contrast to the fully trained Cases 1–3, where λ_1 stays within 0.3% of unity. The transfer regime therefore does exercise the relaxation mechanism, and its λ_1-weighted residuals are correspondingly 20–30% lower than the unweighted ones. All residuals reported in this section are nevertheless *unweighted*, matching the definition used for Cases 1–3 and for PINN-WE (Appendix A.3), so that the same operator is applied throughout. In Table 9, PINN-WE is trained from scratch on Case 4, whereas PIMUG is transferred with frozen encoders; the contrast therefore compares full training from scratch against frozen-encoder adaptation rather than two equivalent regimes.

Table 8: Transfer-learning errors of PIMUG on Case 4. The Euler-residual column aggregates the unweighted continuous autograd residual over all four conservation equations (Appendix A.3).

Perturbation	Data error	Euler residual	Trunk gradient error	Branch gradient error
-5%	1.92×10^{-2}	1.82×10^{1}	1.27×10^{-3}	1.82×10^{-2}
-3%	7.30×10^{-3}	1.17×10^{1}	4.76×10^{-4}	1.87×10^{-2}
-1%	4.42×10^{-3}	1.26×10^{1}	3.73×10^{-4}	1.95×10^{-2}
0%	2.52×10^{-3}	1.08×10^{1}	3.19×10^{-4}	1.99×10^{-2}
1%	1.45×10^{-3}	9.40	3.59×10^{-4}	2.05×10^{-2}
3%	2.06×10^{-3}	9.34	5.70×10^{-4}	2.16×10^{-2}
5%	9.44×10^{-3}	8.91	2.01×10^{-3}	2.34×10^{-2}

Table 9: Comparison of continuous autograd residuals on Case 4 under the $+5\%$ perturbation (per-component unweighted autograd residuals; their sum corresponds to the Euler residual reported for PIMUG in Table 8). PIMUG uses frozen-encoder transfer; PINN-WE is trained from scratch.

Method	Density	x-momentum	y-momentum	Total energy
PIMUG	1.27	1.92	6.60×10^{-1}	5.06
PINN-WE	4.02×10^{-3}	1.09×10^{-2}	8.21×10^{-3}	1.18×10^{-2}

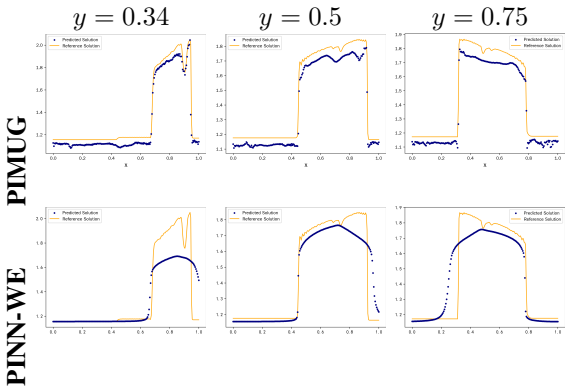


Figure 6: Case 4 density profiles after transfer learning. In each panel, the solid orange curve is the WENO-Z+ reference and the navy dots are the corresponding surrogate prediction.

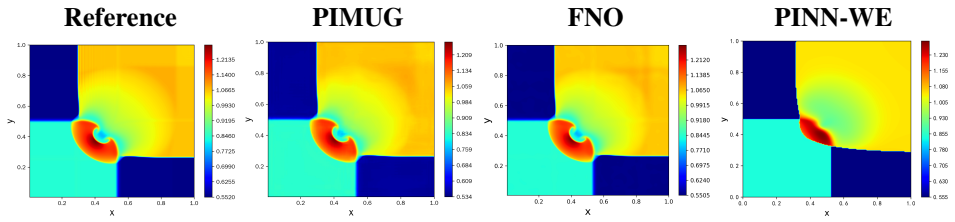


Figure 7: Case 3 density field under $+5\%$ perturbation. PIMUG reproduces the shock–slip-line pattern of the reference; PINN-WE smooths part of the discontinuous structure, whereas FNO fits the field but lacks governing-equation enforcement (Table 1). Detailed comparisons and cross-sections follow in the remaining figures of this appendix.

A.5 Additional Qualitative Results

Figure 8 compares all four predicted variables for Case 1, Figure 9 shows density profiles for Cases 1–3, and Figure 6 in Appendix A.4 shows the transferred Case 4, all under the 5% perturbation.

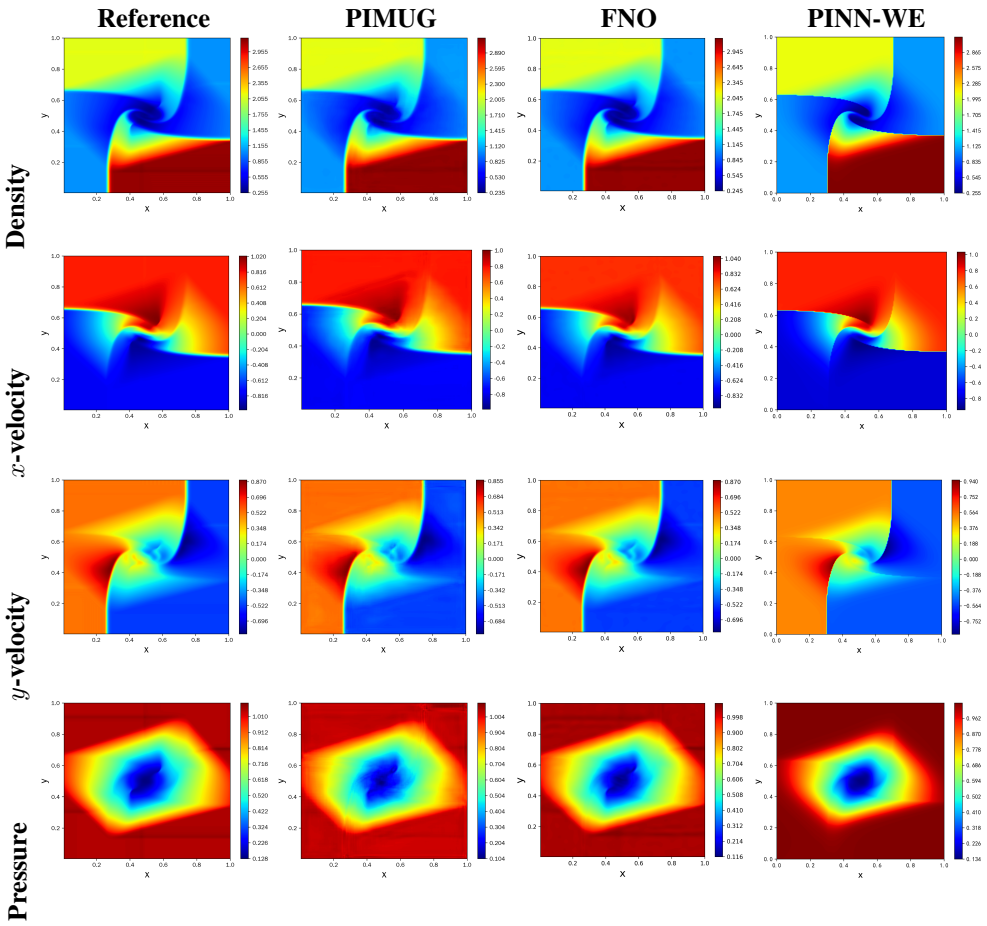
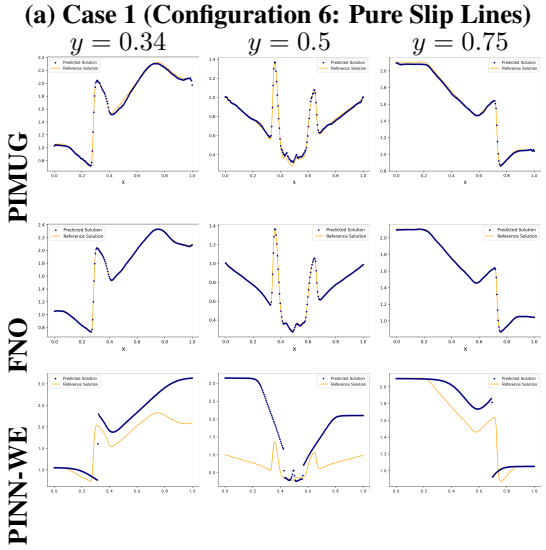
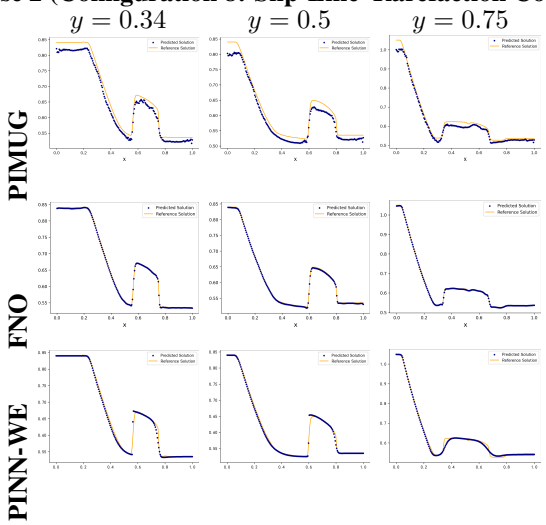


Figure 8: Case 1 predictions of all four primitive flow-field variables (ρ, u, v, p) under the $+5\%$ perturbation.

(a) Case 1 (Configuration 6: Pure Slip Lines)



(b) Case 2 (Configuration 8: Slip-Line–Rarefaction Coupling)



(c) Case 3 (Configuration 11: Shock–Slip-Line Interaction)

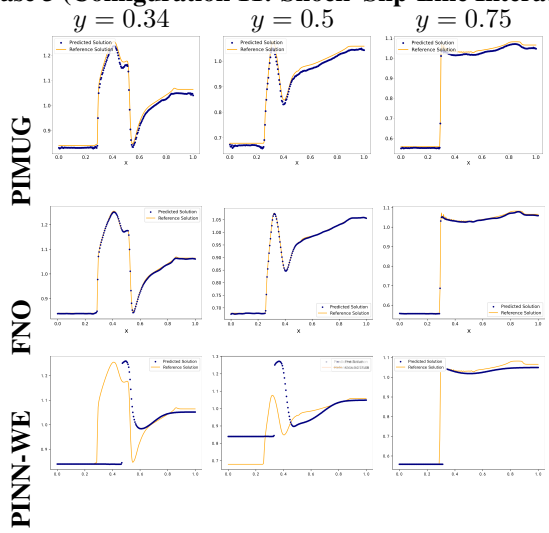


Figure 9: Density profiles of Cases 1–3 under the $+5\%$ perturbation. In each panel, the solid orange curve is the WENO-Z+ reference and the navy dots are the corresponding surrogate prediction.