
Physics-Enhanced Reconstruction of High-Resolution Kolmogorov Flow from Low-Resolution Snapshots

Jiacheng Zhu

University of Pennsylvania
zhujch@seas.upenn.edu

Yifan Cai

University of Pennsylvania
cai103@seas.upenn.edu

Yiran Hu

University of Pennsylvania
yiranhu@seas.upenn.edu

Abstract

We study how to inject physics priors into neural networks when only single-frame observations are available. Using 2D Kolmogorov flow super-resolution task as a testbed, we evaluate CNNs, UNet, FNO, and diffusion models, and compare two forms of physics guidance: spectral feature augmentation and a Navier–Stokes-based consistency loss. Experiments show that feature-level physics offers little benefit, while the proposed physics-consistency loss consistently improves physical accuracy and accelerates diffusion model convergence. Our results suggest that simple consistency constraints provide an effective and scalable way to introduce physical priors into modern AI models without requiring temporal supervision. The full implementation of our method is open-sourced at <https://github.com/zhujch1/CIS-5200-Final-Project>.

1 Motivation

Deep neural networks have achieved strong performance in many reconstruction tasks, yet most existing methods in Computer Vision (CV) focus only on pixel fidelity, with little regard for underlying physics. On the other hand, Scientific Machine Learning (SciML) approaches such as physics-informed neural networks (PINNs) enforce physical laws but typically require temporal derivatives or sequential data — a luxury that many real-world scientific measurements lack.

This gap motivates a fundamental question: **How can we inject physical priors into modern neural networks when only single snapshots are available?**

To investigate this question, we use fluid super-resolution (SR)—the task of reconstructing high-resolution (HR) flow fields from low-resolution (LR) inputs—as a controlled testbed. Turbulent flows contain rich multi-scale structures that are challenging to resolve numerically, yet well suited for machine-learning methods that excel at learning nonlinear spatial mappings. This makes SR an ideal setting to observe whether a model truly learns physics or merely produces visually sharp fields. We choose 2D Kolmogorov flow, a classic turbulence benchmark, and evaluate several neural architectures — CNNs, UNet, FNO, and diffusion models — under different strategies for enforcing physics.

We investigate two ways of incorporating priors: **physics as learned features** and **physics as training constraints**. Our goal is to understand whether these approaches — individually or combined — can improve the physical correctness of reconstructed flow fields.

2 Related Work

2.1 Physics-Guided SR: PINNs and Hybrid Models - SciML Community

Physics-informed neural networks (PINNs) embed physical laws via PDE residuals, ensuring strong physical consistency and good performance with scarce HR labels [1]. Models like PINN-SR[2] and PINN–CNN hybrids[3] demonstrate that physics constraints can improve fluid super-resolution. However, PINNs are computationally expensive, scale poorly to HR turbulent flows, and PDE constraints make single-frame SR often ill-posed (our main focus).

2.2 Neural Operators for Continuous-Field SR (FNO) - SciML Community

Neural operators approximate mappings between infinite-dimensional function spaces, enabling discretization-invariant inference and strong cross-grid generalization. The Fourier Neural Operator (FNO)[4] , which performs global convolution in Fourier space, has demonstrated impressive capability in fluid tasks such as zero-shot super-resolution and large-scale weather prediction (FourCastNet [5]). However, neural operators can be memory-intensive and may oversmooth high-frequency structures when the retained modes are limited.

2.3 Image SR via CNN Architectures - CV Community

Classical image super-resolution methods treat LR-to-HR mapping as a pixel-wise regression task, using convolutional architectures such as SRCNN[6], EDSR[7], and RDN[8]. These models efficiently learn local spatial correlations but tend to over-smooth high-frequency textures and struggle to reconstruct multi-scale turbulent structures. For this reason, CNNs serve as lightweight baselines in our study rather than primary models.

2.4 Diffusion-Driven SR - CV Community

Diffusion models have recently achieved state-of-the-art performance in image super-resolution. SR3 [9] interpolates LR images to HR and concatenates them with noisy images as conditional input; ResDiff [10] and CDPMSR [11] use a pre-trained SR model or CNN to map LR to HR as conditional input; SRDiff [12] employs a pre-trained encoder to extract LR features and performs diffusion in the residual space. These models excel at preserving fine-scale textures and generating multi-scale stochastic structures, making them promising for turbulent flow reconstruction. However, despite their success in natural images, their application to physical fields remains limited, existing methods do not incorporate physics priors or PDE constraints—essential for turbulence super-resolution.

3 Numerically Simulated Dataset

Since no public dataset perfectly matches our needs, we developed a pseudo-spectral Navier–Stokes solver to generate high-resolution Kolmogorov flow snapshots for systematic evaluation. The resulting dataset is publicly available at: <https://huggingface.co/datasets/skpy/PSFR/tree/main> , with source code in the /kmflow directory. It contains 14,400 samples (approximately 15 GB, .npy format). For visualization, we provide the script km_visualize.py which can generate image files. Figure 1 shows sample data collected along a trajectory over time.

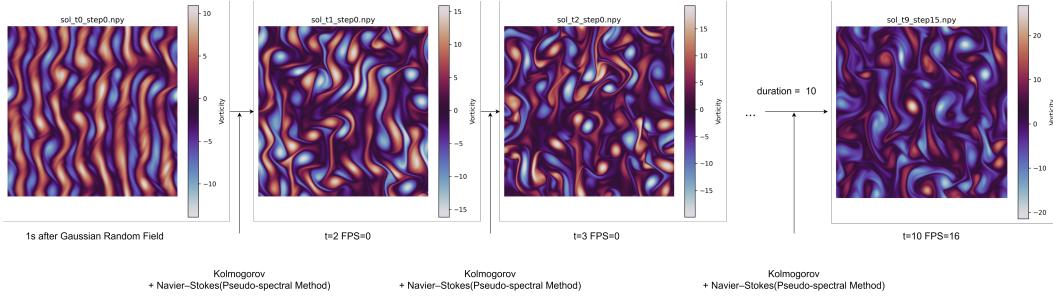


Figure 1: Example HR vorticity snapshots from the Kolmogorov flow dataset, illustrating the flow evolution after initialization. The snapshots are sampled during the recording window after the spin-up period.

3.1 Kolmogorov Flow Dataset Construction and Solver

Kolmogorov flow [13] is a classical periodic model in fluid dynamics, widely used to study Navier–Stokes dynamics, including turbulence transition and energy cascades. It is governed by the incompressible Navier–Stokes equations driven by a sinusoidal forcing field, with the velocity field $\mathbf{u} = (u, v)$ evolving on a periodic domain:

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{F}(k, y), \quad \nabla \cdot \mathbf{u} = 0, \quad (1)$$

or equivalently in terms of vorticity $\omega = (\nabla \times \mathbf{u})_z$ with a linear drag coefficient α for numerical stability:

$$\partial_t \omega + (\mathbf{u} \cdot \nabla) \omega = \frac{1}{Re} \nabla^2 \omega + (\nabla \times \mathbf{F})_z - \alpha \omega. \quad (2)$$

To capture a range of turbulence intensities and spatial complexities, we generate simulations across multiple Reynolds numbers ($Re = 1000, 2000, 3000$) and forcing wavenumbers ($n = 8, 12, 16$), ensuring diverse dynamical regimes.

To generate these flows, we implement a custom Fourier pseudo-spectral Navier–Stokes solver, reproducing the classical formulation by Orszag [14]. Each simulation is initialized from a Gaussian random vorticity field with prescribed spectral decay. The solver transforms the vorticity PDE into an ODE system in Fourier space, with spectral Poisson inversion, pseudo-spectral treatment of nonlinear advection, and Kolmogorov forcing. The vorticity evolution in Fourier space is given by

$$\partial_t \hat{\omega}(k_x, k_y) + ik_x \hat{u}\hat{\omega}(k_x, k_y) + k_y \hat{v}\hat{\omega}(k_x, k_y) = -\frac{1}{Re} (k_x^2 + k_y^2) \hat{\omega}(k_x, k_y) + \hat{f}_\omega(k_x, k_y), \quad (3)$$

where $\hat{\omega}$ is the Fourier-transformed vorticity, $\hat{u}\hat{\omega}, \hat{v}\hat{\omega}$ denote the Fourier transforms of the nonlinear advection terms, and \hat{f}_ω represents the Kolmogorov forcing. Time integration is performed using a Heun-type method. This solver produces high-fidelity data suitable for SR tasks.

3.2 High- and Low-Resolution Data

For each (Re, n) pair, we simulate Kolmogorov flow on a 512×512 grid for $6s$ time units, with the first $1s$ used as spin-up. During the remaining $5s$ units, snapshots are recorded at a uniform interval, yielding 160 high-resolution frames per trajectory. Also, each pair produces 10 independent trajectories. Across three Reynolds numbers and three forcing wavenumbers, this results in

$$3 \times 3 \times 10 \times 160 = 14,400$$

high-resolution vorticity samples. Low-resolution inputs are obtained by $4\times$ downsampling:

$$\omega_{LR} = \omega_{HR}[:, :: 4, :: 4] \quad (4)$$

The corresponding {LR, HR} serve as the input–target pair for supervised super-resolution.

4 Problem Formulation

4.1 Task Definition

The super-resolution task is to reconstruct a high-resolution (HR) vorticity field from a single low-resolution (LR) snapshot. Given an input field:

$$\omega_{LR} \in \mathbb{R}^{128 \times 128}, \quad (5)$$

our task is to recover its corresponding high-resolution counterpart (where fine-scale vortical structures are destroyed by downsampling):

$$\omega_{HR} \in \mathbb{R}^{512 \times 512}, \quad (6)$$

We intentionally adopt a **single-frame formulation** to isolate spatial reconstruction ability and to evaluate how much physical structure can be recovered without temporal information. This setting also reflects many real-world scenarios where only sparse or single-shot measurements are available—for example in PIV experiments, satellite observations, or 4D Flow MRI—making single-frame reconstruction both scientifically meaningful and practically important. The single-frame formulation also isolates the spatial reconstruction capability of each model without confounding temporal prediction errors, enabling a clearer comparison of architecture-level inductive biases.

4.2 Machine Learning Formulation

We frame the problem as learning a mapping between two discretized function spaces:

$$\hat{\omega}_{HR} = f_\theta(\omega_{LR}), \quad (7)$$

where f_θ denotes a neural network parameterized by θ .

Or equivalently, the model can be interpreted as approximating the conditional distribution

$$p(\omega_{HR} | \omega_{LR}), \quad (8)$$

Since the mapping must infer missing high-frequency details absent in the LR input, the model needs to capture meaningful spatial and multi-scale structures of turbulent Kolmogorov flows. We employ diverse architectures, including CNNs, UNets, Fourier Neural Operators, and diffusion models. The task is formulated as supervised regression, optionally regularized by physics constraints. We use an ℓ_2 reconstruction loss between $\hat{\omega}_{HR}$ and ω_{HR} for stability and robustness, and, when incorporating

physics priors, a PDE residual loss from the vorticity equation to enforce physical consistency. This ensures both pixel-level accuracy and adherence to the underlying dynamics.

4.3 Rationale for Treating SR as a Physics-Guided Mapping

(1) Downsampling destroys physical structure. The LR field removes fine-scale vortices essential for turbulence analysis. Thus the learning problem is inherently ill-posed and requires learning physical and statistical priors. **(2) Neural networks are effective function approximators.** Architectures such as CNNs, UNets, FNOs, and diffusion models excel at recovering missing high-frequency content and capturing nonlinear spatial correlations. **(3) The formulation is architecture-agnostic.** By defining SR as learning a mapping or conditional distribution, we can evaluate the impact of different model classes under a unified framework, which is one of the contributions of this work

4.4 Physics-Guided Formulation

Although the task is framed as supervised LR-to-HR mapping, purely data-driven models can produce visually plausible but physically inconsistent results. To address this, our framework incorporates physics priors in two ways: (1) augmenting LR inputs with physics-derived features (Laplacians, nonlinear terms, streamfunction, velocity) to provide dynamical cues; (2) for models estimating vorticity time derivatives, applying a physics-informed loss based on the Navier–Stokes residual to enforce dynamical consistency. These components, detailed in Section 5, extend the ML formulation introduced above.

5 Methods

We evaluate four model families—CNN, UNet, FNO, and Diffusion models as backbones—together with several physics-guided mechanisms. All architectures are **implemented from scratch** in PyTorch. Figure 2 presents the general framework through which physics information is integrated into all four model architectures, all the inputs are std-normalized.

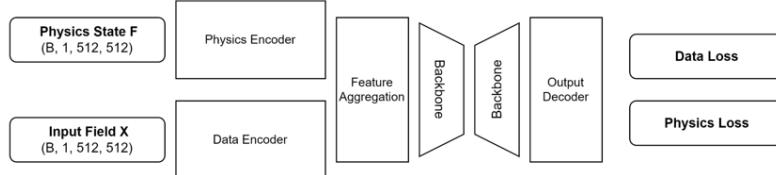


Figure 2: Architecture

Let $\omega_{LR} \in \mathbb{R}^{128 \times 128}$ denote the low-resolution vorticity field and $\omega_{LR}^\uparrow \in \mathbb{R}^{512 \times 512}$ its bicubic upsampled version. All models operate on ω_{LR}^\uparrow and produce a prediction $\hat{\omega}_{HR} \in \mathbb{R}^{512 \times 512}$.

5.1 Hierarchical CNN Baseline

The CNN baseline provides a lightweight reference architecture. Given ω_{LR}^\uparrow , the CNN applies a 3×3 convolution W_1 with stride 2 and GroupNorm, followed by a stack of Conv–GroupNorm–ReLU blocks at the reduced resolution, and finally a 3×3 convolution W_{out} and bilinear upsampling:

$$z_1 = \sigma(\text{GN}(W_1 * \omega_{LR}^\uparrow)), \quad (9)$$

$$z_k = \sigma(\text{GN}(W_k * z_{k-1})), \quad (10)$$

$$\hat{\omega}_{HR} = \text{Up}(W_{out} * z_K), \quad (11)$$

where σ denotes ReLU, GN is Group Normalization, $*$ is convolution, and Up is bilinear upsampling with scale factor 2. When input and output channels match, an optional residual skip $\omega_{LR}^\uparrow \mapsto \omega_{LR}^\uparrow + \hat{\omega}_{HR}$ is enabled. This baseline quantifies the difficulty of HR reconstruction without multi-scale pathways or operator learning.

5.2 UNet with ResNet Blocks and Optional Attention

To better capture multi-scale vortex structures, we adopt a custom UNet composed of modular ResNet blocks. Each block contains GroupNorm, SiLU activations, and 3×3 convolutions, with

optional self-attention at selected resolutions. Skip connections are recorded at every block, enabling fine-grained feature reuse in the decoder.

Let $x_{0,0} = \omega_{LR}^\uparrow$ be the input to the encoder. At encoder resolution level i and block index j , a ResNet block updates

$$x_{i,j} = x_{i,j-1} + F_{i,j}(x_{i,j-1}), \quad (12)$$

where $F_{i,j}$ is a sequence of GroupNorm \rightarrow SiLU \rightarrow Conv. Downsampling between resolutions is performed by a stride-2 convolution

$$x_{i+1,0} = D_i(x_{i,M_i}), \quad (13)$$

with M_i the number of blocks at level i .

In the decoder, the corresponding feature maps are upsampled and concatenated with encoder activations:

$$y_{i,0} = U_i(y_{i+1,M_{i+1}}), \quad (14)$$

$$y_{i,j} = y_{i,j-1} + G_{i,j}([y_{i,j-1}, x_{i,j}]), \quad (15)$$

where U_i denotes bilinear upsampling followed by convolution and $[\cdot, \cdot]$ denotes channel-wise concatenation. A final 3×3 convolution H maps the top-level decoder output to a single vorticity channel:

$$\hat{\omega}_{HR} = H(y_{0,M_0}). \quad (16)$$

When used as the backbone of the diffusion model, each ResNet block additionally receives a temporal embedding that modulates its activations.

5.3 Fourier Neural Operator (FNO)

To align with the spectral structure of Kolmogorov flow and our pseudo-spectral simulator, we implement a 2D Fourier Neural Operator (FNO). Each layer updates the input via a learned spectral kernel applied to a limited set of Fourier modes, combined with a 3×3 convolution for local refinement.

For an input feature map $x \in \mathbb{R}^{C \times S \times S}$, let $\hat{x} = \mathcal{F}(x)$ be its 2D FFT. We retain only the lowest (m_x, m_y) modes and apply learned complex-valued weights, with separate parameters for positive and negative k_x :

$$\hat{z}_o(k_x, k_y) = \begin{cases} \sum_{c=1}^C W_{o,c}^{(1)}(k_x, k_y) \hat{x}_c(k_x, k_y), & (k_x, k_y) \in \mathcal{K}_+, \\ \sum_{c=1}^C W_{o,c}^{(2)}(k_x, k_y) \hat{x}_c(k_x, k_y), & (k_x, k_y) \in \mathcal{K}_-, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where \mathcal{K}_+ and \mathcal{K}_- denote the retained positive and negative wavenumber blocks. The spectral update is transformed back to physical space,

$$z = \mathcal{F}^{-1}(\hat{z}), \quad (18)$$

and combined with a parallel convolution branch

$$c = W_{\text{conv}} * x. \quad (19)$$

After GroupNorm and GELU, the layer output is

$$x' = \text{GELU}(\text{GN}(z + c)). \quad (20)$$

Stacking several such layers yields a global operator capable of capturing long-range interactions characteristic of forced turbulence. In the SR pipeline, ω_{LR}^\uparrow is first mapped into a feature space and then refined by a sequence of FNO layers before decoding back to $\hat{\omega}_{HR}$.

5.4 Physics-Aware Diffusion Model

We employ a conditional diffusion model whose backbone is the time-dependent UNet introduced above. Following the x -prediction DDPM formulation[15], the network predicts the clean HR vorticity x_0 directly from noisy samples.

Forward diffusion. With a linear noise schedule $\{\beta_t\}_{t=1}^T$, define

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s. \quad (21)$$

The noising process is

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad (22)$$

implemented by `q_sample`.

x-prediction network. At timestep t , the model receives x_t , the upsampled LR field ω_{LR}^\uparrow , and optionally physics-derived features $\phi(\omega_{LR})$. These inputs are encoded and concatenated, then processed by the time-conditioned UNet:

$$\hat{x}_0 = f_\theta(x_t, \omega_{LR}^\uparrow, t, \phi(\omega_{LR})). \quad (23)$$

Reverse sampling. Using DDPM posterior coefficients,

$$\mu_t = a_t \hat{x}_0 + b_t x_t, \quad x_{t-1} = \mu_t + \sigma_t \eta, \quad \eta \sim \mathcal{N}(0, I), \quad (24)$$

as implemented in `p_sample`.

Physics consistency loss. To encourage dynamical consistency, we employ the same pseudo-spectral Navier–Stokes operator used in data generation:

$$F(w; Re, n) = \partial_t \omega(w; Re, n). \quad (25)$$

The physics loss penalizes discrepancies in the induced time derivatives:

$$\mathcal{L}_{\text{deriv}} = \|F(\hat{x}_0) - F(x_0)\|_2^2, \quad (26)$$

When enabled, the diffusion objective becomes

$$\mathcal{L}_{\text{diff}} = \lambda_{\text{deriv}} \mathcal{L}_{\text{deriv}}, \quad (27)$$

promoting HR reconstructions that respect Navier–Stokes dynamics.

5.5 Physics-Guided Components For Four Models

Physics priors are incorporated via spectral operators that mirror the Navier–Stokes solver used for data generation.

Physics features. We augment the LR input with quantities derived from PDE induced operators, such as the Laplacian, streamfunction, and velocity components. These features summarize rotational and advective behavior not directly visible in the downsampled field.

PDE residual loss. We enforce physical consistency by penalizing the discrepancy between the predicted and true vorticity time derivatives, computed through the same pseudo-spectral operator used in data generation. The exact formulation of this loss is introduced in the diffusion model.

6 Experiments and Results

6.1 Training Configuration

All models are trained with Adam (learning rate 10^{-4} , $\beta_1=0.9$, $\beta_2=0.999$, no weight decay) using a batch size of 32 for 5 epochs. Mixed-precision training (FP16) is enabled via `GradScaler`. The dataset contains 14,400 samples, split 80/20 into training and validation sets. An additional set of 144 samples is generated for testing. Training is run on a single NVIDIA B200 GPU (MIG-90 profile).

6.2 Performance Metrics

We use three complementary metrics to evaluate the reconstruction quality:

(1) Mean Squared Error (MSE). A standard pixel-wise measure of reconstruction accuracy:

$$\text{MSE} = \|\hat{\omega}_{HR} - \omega_{HR}\|_2^2.$$

it is the standard measure of pointwise reconstruction accuracy in super-resolution tasks. Although it does not fully capture small-scale turbulence, MSE provides a consistent baseline for comparing models and ensures that the global vorticity field is numerically accurate.

(2) Physics Consistency Error (PCE). To assess physical consistency, we compare the predicted and ground-truth vorticity fields through their temporal derivatives implied by the Navier–Stokes

equations. Let $\mathcal{N}(\cdot)$ denote the pseudo-spectral operator computing

$$\frac{\partial \omega}{\partial t} = \mathcal{N}(\omega) = -\mathbf{u} \cdot \nabla \omega + \frac{1}{Re} \Delta \omega + f,$$

using the same Fourier stencil employed in dataset generation. The derivative residual loss is

$$\mathcal{L}_{\text{deriv}} = \|\mathcal{N}(\omega_{\text{pred}}) - \mathcal{N}(\omega_{\text{gt}})\|_2^2.$$

This term penalizes discrepancies in the implied dynamical evolution, encouraging reconstructions that are physically plausible, not merely numerically similar.

(3) Energy Spectrum Error (ESE). To evaluate how well the model reconstructs the fine-scale structures that are typically lost during downsampling, we compute the relative error in the isotropic energy spectrum over a selected range of mid–high wave numbers:

$$\text{Err}(k) = \frac{|E_{\text{pred}}(k) - E_{\text{gt}}(k)|}{E_{\text{gt}}(k)},$$

where $E_{\text{pred}}(k)$ and $E_{\text{gt}}(k)$ denote the predicted and ground-truth spectra.

Instead of evaluating all wave numbers, we focus on mid–high frequencies because low-frequency structure is already preserved in the LR input, whereas SR must recover the small-scale vortical features encoded in higher k . This makes the spectral error a targeted measure of fine-scale turbulence reconstruction that pixel-wise metrics like MSE cannot capture.

6.3 Performance Results Across All Models

Table 1: Comparison of SR model performance.

Model	MSE ↓	Consistency Error ↓	Energy Spectrum Error ↓
CNN-vis	0.074325	44.42137	0.044681
CNN-loss	0.059858	27.23335	0.028782
CNN-feat	0.059803	87.276901	0.081832
CNN-both	0.026098	29.812644	0.020574
FNO-vis	0.021097	43.753108	0.039619
FNO-loss	0.020333	16.731096	0.015712
FNO-feat	0.020775	55.339843	0.044689
FNO-both	0.016933	18.256484	0.018389
UNet-vis	0.053052	101.093528	0.112314
UNet-loss	0.033025	24.422718	0.031949
UNet-feat	0.050432	96.757049	0.100919
UNet-both	0.035860	38.494117	0.038818
Diffusion-vis	0.036409	46.311275	0.007826
Diffusion-loss	0.018609	6.935647	0.007030
Diffusion-feat	0.053068	39.950759	0.011965
Diffusion-both	0.111310	396.946990	0.007866

The four variants indicate different uses of physics: *-vis* (no physics), *-loss* (physics-informed loss), *-feat* (physics-derived features), and *-both* (combined).

Overall, adding physics through a loss term provides the most consistent improvements, markedly reducing the physical metrics, and also lowering MSE for most architectures. In contrast, physics features are not consistently beneficial and sometimes even degrade performance, likely because the additional channels introduce high-frequency or numerically stiff information that the network cannot easily integrate, thereby making optimization more difficult. The Diffusion-*both* model exhibits strong instability, as simultaneously enforcing physics features and physics-based losses substantially increases training complexity beyond what the available compute can stably support. On the other hand, the Diffusion-*loss* results demonstrate that introducing a single physics-based loss term significantly accelerates diffusion model convergence.

6.4 Qualitative Analysis

Reconstruction Visualization. Figure 3 shows the ground truth, the low-resolution input, and the four CNN-based super-resolution variants. This layout highlights that even a simple CNN architecture can visually reconstruct the large-scale flow patterns with reasonable fidelity. However, closer inspection reveals discrepancies in the fine-scale structures—particularly in vortex shape, orientation, and small-scale filamentary features.

While all four reconstructions appear visually plausible, their underlying physical correctness differs substantially. This disconnect between visual quality and physical fidelity motivates the need for physics-aware evaluation metrics and modeling strategies.

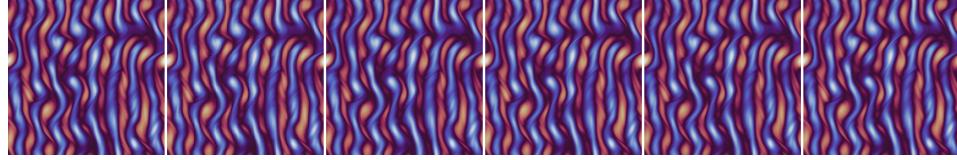


Figure 3: From left to right, the images are: ground truth, LR, -vis, -feat, -loss, -both

Validation Loss. Figure 4 shows the validation loss curves. The green line denotes the pure vision model (without physics features or physics constraints); the orange line corresponds to the model with physics constraints only; the blue line represents the model with physics features only; and the red line combines both components. We observe that when trained for the same number of steps and epochs, incorporating physics features **without** physics-based loss can increase the data loss (pixel-wise error), while the other three approaches achieve similar loss. The diffusion loss recorded here is computed by a single step sampling for efficiency.

Examining the physics-consistency loss reveals that models using physics constraints—either alone or in combination with physics features—generally achieve lower physics-consistency errors. In contrast, using physics features alone without enforcing physics constraints can degrade performance (see CNN, FNO, and Diffusion results).

We can conclude that physics consistency loss is the key to generate physically-correct results.

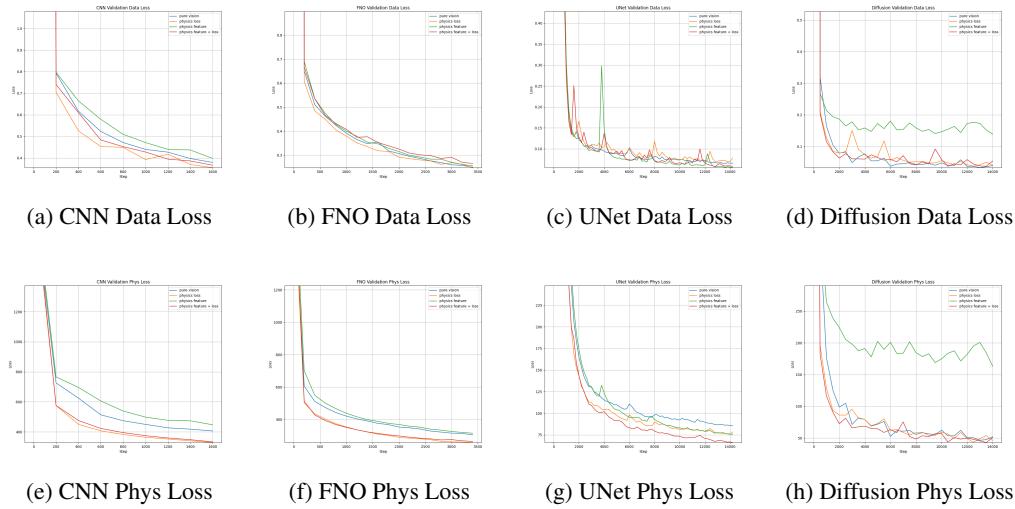


Figure 4: Data & Physics Loss

7 Conclusion and Discussion

7.1 Summary of Findings

In this project, we discovered how to effectively incorporate physics priors into modern neural network architectures from both SciML and CV community, even when only temporal PDE constraints are not available. Through our experiments, we gained several practical and conceptual insights:

- **Positive Findings**

- Physics-consistency constraints improve physical accuracy, measured by both the consistency loss and the energy spectrum error — metrics that are of primary interest to the physics community.
- When temporal information is missing, physics-consistency constraints act as a strong substitute for PINN-style regularization.

- **Negative Findings / Limitations**

- Simply augmenting the input with physics-derived features does not necessarily improve performance.
- Combining physics-consistency constraints with physics-based feature inputs can lead to higher accuracy, but may also introduce training instability, especially under limited computational resources (as seen in our diffusion experiments).

7.2 Qualitative Interpretation

We analyze the role of the physics consistency constraint through the lens of training dynamics. Let \mathcal{F} be a non-linear and differentiable operator that maps a physical quantity to a meaningful physical state. We examine the gradient of the loss function with respect to the network parameters:

$$\begin{aligned} & \frac{\partial}{\partial \theta} (\|x_{\text{pred}} - x_{\text{gt}}\|_2^2 + \lambda \|\mathcal{F}(x_{\text{pred}}) - \mathcal{F}(x_{\text{gt}})\|_2^2) \\ &= \frac{\partial}{\partial \theta} \|x_{\text{pred}} - x_{\text{gt}}\|_2^2 + \lambda \frac{\partial}{\partial \theta} \|\mathcal{F}(x_{\text{pred}}) - \mathcal{F}(x_{\text{gt}})\|_2^2 \\ &= 2 \left[\underbrace{(x_{\text{pred}} - x_{\text{gt}})^\top}_{\text{Data Gradient}} + \lambda \underbrace{(\mathcal{F}(x_{\text{pred}}) - \mathcal{F}(x_{\text{gt}}))^\top \mathbf{J}_{\mathcal{F}}}_{\text{Physics Gradient}} \right] \frac{\partial x_{\text{pred}}}{\partial \theta} \end{aligned}$$

When \mathcal{F} is non-linear and differentiable, the Jacobian $\mathbf{J}_{\mathcal{F}}$ is non-diagonal, meaning it can modify not only the magnitude but also the direction of the gradient. As a result, the optimization is nudged toward regions of the solution space that respect the underlying physical laws, rather than solely minimizing pixel-wise error. This effectively guides the training trajectory along manifolds that are more physically plausible.

7.3 Future Directions

Several natural extensions arise from this work:

- **3D Extension:** If computational resources allow, extending the physics-enhanced super-resolution method to fully 3D Navier–Stokes turbulence would be a major step forward. This could reveal how well the model scales to more realistic flows and help test its ability to recover complex vortex structures beyond 2D Kolmogorov flow.
- **Extension to Complex Scenarios.** Real-world flows often involve far more complex boundary conditions, geometries, and topologies — for example, blood flow in branching vessels or turbulent structures in irregular domains. Applying the method in such settings would provide stronger evidence of its robustness, and may require new strategies for handling geometry-aware physics constraints.
- **Extension to Other Fields.** In the robotics domain, robots must obey contact dynamics, energy limits, and safety constraints. A natural future step is to explore whether similar physics-guided approaches can be applied to policy learning, allowing robots to better understand physical interactions and potentially reducing the sim-to-real gap.

References

- [1] Maziar Raissi, Paris Perdikaris, and George Em 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.
- [2] Cheng Wang, Enrico Bentivegna, Wenxiao Zhou, Paris Perdikaris, and Daniel Tartakovsky. Physics-informed neural network super resolution for advection-diffusion models. *arXiv preprint arXiv:2011.02519*, 2020.
- [3] Han Gao, Lei Sun, and Jian-Xun Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7):073603, 2021.
- [4] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [5] Jaideep Pathak, Sandeep Subramanian, Peter Harrington, Safa Raja, Morteza Mardani, Kamyar Azizzadenesheli, and Anima Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.
- [6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015.
- [7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.
- [8] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.
- [9] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022.
- [10] Shuai Shang, Zhe Shan, Guangyuan Liu, Yifan Meng, and Li Yuan. Resdiff: Combining cnn and diffusion model for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8975–8983, 2024.
- [11] Ao Niu, Kai Zhang, Trung X Pham, Yu Liu, Luc Van Gool, and Radu Timofte. Cdpmr: Conditional diffusion probabilistic models for single image super-resolution. *arXiv preprint arXiv:2302.12831*, 2023.
- [12] Haoyu Li, Yinheng Yang, Mingbao Chang, Chongyi Ma, Zhiyuan Liu, Zhiyong Luo, Zhenyu Zeng, and Wangmeng Cheng. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.
- [13] E. Fylladitakis. Kolmogorov flow: Seven decades of history. *Journal of Applied Mathematics and Physics*, 6:2227–2263, 2018.
- [14] Steven A. Orszag. Numerical simulation of incompressible flows within simple boundaries: accuracy. *Journal of Fluid Mechanics*, 49(1):75–112, 1971.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020.