Physics-Consistent Diffusion for Efficient Fluid Super-Resolution via Multiscale Residual Correction

Supplementary Material

1. Related Work

Neural operators. Operator learning (e.g., FNO and its successors) models mappings between function spaces for families of PDEs with cross-grid generalization [14, 18]. The original Fourier neural operator (FNO) parameterizes the integral kernel directly in Fourier space so that a single network trained on one discretization can transfer to unseen meshes and PDE parameters, achieving resolution-invariant surrogates that are up to three orders of magnitude faster than classical solvers on Burgers, Darcy flow and Navier–Stokes benchmarks [18]. Building on this, the neural-operator framework in [14] formalizes such mappings as compositions of linear integral layers and nonlinearities, establishes universal approximation in infinite-dimensional Banach spaces, and instantiates several concrete operator families (graph, low-rank and Fourier neural operators) that show strong performance and discretization-invariance across standard PDE testbeds [14]. Recent variants refine spectral modeling and multiscale hierarchy. HiNOTE [22] introduces a hierarchical neural-operator transformer with Galerkin-style self-attention on multi-resolution grids and a learnable frequency-aware loss prior that re-weights supervision across spectral bands, enabling arbitrary-scale scientific SR while explicitly controlling the reconstructed frequency content. DiffFNO [21] couples a weighted FNO backbone—equipped with mode re-balancing to better capture critical Fourier modes—with an attention-based neural operator and an adaptive-step ODE sampler, forming a diffusion framework that strengthens high-frequency recovery for arbitrary-scale image SR while keeping the sampling budget moderate. Spectral-Refiner [1] focuses on turbulent flows and fine-tunes spatio-temporal FNOs using a spatiotemporal adaptation scheme together with negative Sobolev-type spectral regularization, which alleviates spectral bias and substantially improves long-horizon accuracy and stability of turbulent rollouts. For physics-field SR, Guo et al. [5] integrate an adaptively weighted FNO front-end with a residual-guided diffusion model: a Markov chain first maps high-resolution physical fields to low-resolution counterparts, and the reverse diffusion process is solved via an adaptive time-step ODE solver, yielding improved accuracy and spectral fidelity over purely data-driven SR networks and numerical interpolation baselines. Unlike these methods, which either realize a single feed-forward operator or use diffusion refinement in feature/image space without explicit multigrid structure, ReMD keeps the operator viewpoint but performs iterative multigrid residual correction inside the sampler. Each reverse step is updated under an explicit data-consistency operator on a multi-wavelet hierarchy, so ReMD remains equation-free while enforcing spectrum-faithful, low-divergence corrections that are structurally different from prior neural-operator and operator–diffusion designs.

Multigrid applications. Prior work has also brought multigrid ideas into learning-based PDE solvers. Hsieh et al. [11] learn neural PDE solvers with convergence guarantees by viewing the network as a learned correction to a classical iterative scheme, so that the resulting method still fits into a provable fixed-point framework while accelerating convergence and generalizing across meshes and geometries. MgNO [9] moves from using multigrid merely as a preconditioner to using it as a parameterization of bounded linear operators on hierarchies, defining neural-operator layers whose weights follow a multigrid structure; this design naturally accommodates varying boundary conditions and discretizations, and alleviates some of the training and generalization issues observed in earlier operator architectures. M2NO [19] further combines multiwavelet analysis with algebraic multigrid to build a multiresolution neural operator: multiwavelet transforms provide hierarchical decompositions that capture both global trends and local details of PDE solutions, while multigrid-style couplings between levels enable efficient multiscale propagation, which proves beneficial for high-resolution and super-resolution PDE tasks. UGrid [8] takes a more solver-oriented view and tightly integrates a U-Net architecture into a classical multigrid V-cycle, yielding a neural multigrid solver for linear PDEs with rigorous convergence analysis and strong generalization to different linear operators and domains. ReMD is inspired by these multigrid-based designs but serves a different role: instead of parameterizing operators or constructing stand-alone multigrid solvers, it repurposes multigrid as a time-gated residual corrector inside a diffusion sampler. Using fixed multiwavelet restriction/prolongation together with lightweight smoothers, each reverse step transports errors from coarse to fine scales under explicit data/physics consistency, resulting in an equation-free multigrid mechanism tailored to fluid super-resolution rather than a conventional PDE solver.

Super-resolution with diffusion. Diffusion-based SR has progressed from continuous-scale formulations to few- and single-step samplers. Implicit Diffusion Models for Continuous Super-Resolution (IDM)[4] couple a diffusion prior with an implicit neural representation of images, so that a single model supports SR at arbitrary scaling factors in a continu-

ous spatial domain. ResShift[30] then reparameterizes the diffusion state as high–low-resolution residuals, shortening the noise schedule while maintaining competitive perceptual and distortion metrics.

Recent generative paradigms further push the speed limit: Flow Matching [20] straightens the generation probability paths via optimal transport to enable fast ODE integration, while Consistency Models [27] learn to map arbitrary time-states directly to the data origin, supporting one-step generation.

SinSR [31] similarly distills a multi-step diffusion SR teacher into a single-step student network via a consistency-style objective.

In geophysical SR, diffusion models have been adapted from natural images to flow and sea-surface-height (SSH) fields. A physics-informed diffusion model [25] reconstructs high-fidelity flow fields from coarse or sparse observations by augmenting the diffusion objective with PDE residuals, improving both reconstruction error and physical consistency. For SSH, DiffSRNet [16] and a Kuroshio-focused diffusion downscaler [7] use conditional diffusion to refine coarse SSH into high-resolution fields, recovering more realistic eddy structures and spectra than interpolation and deterministic deep SR baselines. The Multi-scale Boundary-Enhanced Diffusion Network (MSBED)[17] further introduces multi-scale feature paths and boundary-aware refinement modules tailored to SSH, sharpening fine-scale coastal and frontal structures. Tweedie-style samplers[24] finally reduce the number of function evaluations by combining truncated diffusion trajectories with iterative refinement grounded in Tweedie's formula, enabling few-step sampling on fluid benchmarks.

Unlike these diffusion SR approaches—which often rely on purely data-driven distillation (like [27, 31]) or latent trajectory straightening (like [20]) to accelerate sampling—ReMD embeds a time-gated multigrid V-cycle with fixed multi-wavelet restriction/prolongation inside the sampler as the per-step update direction. This effectively creates a structural shortcut for error removal across scales, allowing ReMD to achieve high-fidelity 2–5 step inference via explicit physical restriction consistency rather than learned abstract mappings.

Diffusion for scientific fields. Beyond images, diffusion models have been extended to PDE-governed fields via function-space and multiresolution formulations. Diffusion-PDE [13] casts PDE forward and inverse problems under partial observation as conditional generative modeling over joint fields, training a diffusion model that can recover hidden states and parameters more accurately than classical and purely data-driven baselines. WDNO [12] instead performs diffusion in a wavelet domain with multi-resolution training, modeling full trajectories so that cross-resolution simulation and control of dynamical systems become possible while preserving fine-scale structures. FUNDIFF [28] combines a function autoencoder with latent-space diffusion to model distributions over continuous physical fields, providing a function-space view of diffusion that naturally incorporates physics-informed constraints. ReMD is complementary to these lines: rather than designing a general function-space prior, it injects lightweight physics consistency directly into the discrete reverse dynamics via multiscale residual correction on a multi-wavelet multigrid hierarchy, yielding an efficient, spectrum-faithful fluid SR operator.

Implicit Neural Representations (INRs). Distinct from grid-based methods, INRs model signals as coordinate-based continuous functions, decoupling the parameterization from spatial resolution to enable arbitrary-scale super-resolution. LIIF [2] pioneered this approach by learning a shared local implicit function that is queried at continuous coordinates, though it exhibits a spectral bias that tends to blur high-frequency textures. Subsequent work has focused on mitigating this limitation. For instance, LTE [15] introduces local estimators conditioned on Fourier features to better recover fine details. Others, like DINeR [29], propose novel disorder-invariant architectures to faithfully represent signals defined on unordered data sets, which also helps preserve local high-frequency information. In scientific computing, INRs have been adapted to inherently model differential constraints. SIREN [26] replaces standard activations with periodic sine functions, enabling the accurate representation of complex physical fields and their derivatives. Building on this, WIRE [23] employs complex Gabor wavelets as activations to improve time-frequency localization and robustness. Our approach differs fundamentally from these implicit schemes. While INRs regress a single, deterministic continuous mapping—some using wavelets as micro-level neuron activations (WIRE)—ReMD integrates spectral separation into a discrete, generative process. Instead of learning one resolution-agnostic function, ReMD leverages fixed multiwavelets as macroscopic transfer operators within a time-gated multigrid corrector. This structure enforces explicit, physics-consistent residual corrections across a discrete scale hierarchy to achieve high-fidelity, stochastic super-resolution.

2. Model Implementation

This section details how we implement **ReMD**, covering the data pipeline, multigrid corrector with multiwavelet transfers, physics-consistent residuals, training losses and schedules, and practical efficiency considerations. We follow the notation of the preliminaries: the coarse input is $u^{\mathrm{LR}} \in \mathbb{R}^{C\times H_c\times W_c}$, the HR variables are $u \in \mathbb{R}^{C\times H\times W}$, and the restriction operator is a fixed linear map $R : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H_c\times W_c}$.

2.1. Data pipeline and normalization

Preprocessing. For ERA5 and Ocean, we crop HR tiles of size $H \times W$ and synthesize LR by applying the fixed restriction R (averaging or a fixed low-pass multiwavelet filter) followed by decimation. For NS we directly downsample from HR solutions. We store the fluid/land mask M when available and apply it consistently to residual terms that depend on boundaries.

Normalization. Each variable channel is standardized per-dataset using training-set mean and standard deviation. The same statistics are used at test time. When computing frequency-domain terms (Sec. 2.4), we re-scale back to physical units inside the FFT branch and then normalize the returned residual direction, which stabilizes the relative weight among residual components.

Initialization. We construct u_T by upsampling u^{LR} to HR (bicubic by default). The same R used for training is used to enforce restriction consistency at test time.

2.2. Time-gated multigrid residual corrector

At reverse step t, we form a residual

$$r(u_t) \;=\; u^{\mathrm{LR}} - R(u_t) + \lambda(t)\,\rho(u_t),$$

and map it to an update direction $e_t \;=\; S_t\big(r(u_t)\big)$. The corrector S_t is a learnable V-cycle with *fixed* inter-level transfers and lightweight, time-conditioned smoothers.

Fixed multiwavelet transfers. Let H_x, H_y denote 1D low-pass (scaling) filters from an orthonormal multiwavelet family (we use separable 2D tensors). On each level ℓ,

$$R_\ell \equiv I_h^{2h} = H_y^{(\ell)} \otimes H_x^{(\ell)}, \qquad P_\ell \equiv I_{2h}^{h} = R_\ell^{\top}.$$

Filters on higher levels are obtained by dyadic dilation/shift of the base filters. These transfers are *parameter-free* and kept fixed during training, yielding clean separation of low/high frequencies and numerically stable restriction/prolongation. In code, each transfer is implemented as a depthwise separable convolution with frozen kernels.

Level-wise smoothers with timestep modulation. Within each level we use a small depthwise 3×3 convolutional block $\mathrm{Smooth}_\ell(\cdot)$. To adapt the contribution of each level to the reverse timestep, we apply FiLM-style conditioning: given a sinusoidal timestep embedding $\mathrm{TE}(t)$, a tiny MLP produces per-level gates $w_\ell(t) \in (0,1)$ and per-channel scale/shift for the smoothers. Early steps favor coarse levels; late steps favor fine levels. This realizes the coarse-to-fine removal of low-frequency error before sharpening high-frequency structures.

Residual-aware input and stable output. We concatenate $[r,\, u_t]$ and project with a 1×1 convolution to form the input features for the V-cycle, which provides local state context to the corrector. The final 1×1 output projection to e_t is zero-initialized so the sampler starts from pure multigrid drift and gradually learns small data-driven refinements.

Computational cost. Because transfers are fixed and smoothers are depthwise, the per-step complexity is $\mathcal{O}(HW)$ per level with a small constant. Memory is comparable to a shallow CNN of similar width. Inference typically uses $2-5$ reverse steps.

2.3. Implementation Details

We summarize how M2NO is instantiated in practice—covering the discrete representations, multiwavelet transfers, multigrid blocks, and training protocol.

Discrete representation. All fields are stored as tensors $u \in \mathbb{R}^{C\times H\times W}$ on a dyadic hierarchy $\{\Omega^{h_\ell}\}_{\ell=0}^{L}$ with $h_{\ell+1} = 2h_\ell$. We keep channel dimension C (e.g., multiple variables) unchanged across levels and only down/upsample the spatial axes. Unless otherwise stated, we use reflection (Neumann) padding for bounded domains and circular padding for periodic problems; the same choice is used consistently by all components.

Multiwavelet transfers (fixed). From a chosen orthonormal multiwavelet family, we precompute 1D scaling (low–pass) filters $H^{(0)}, H^{(1)}$ and wavelet (high–pass) filters $G^{(0)}, G^{(1)}$ for level transitions. In 2D we use separable tensors:

$$H \;=\; H_y \otimes H_x, \qquad G \;=\; \begin{bmatrix} G_y \otimes H_x \\ H_y \otimes G_x \\ G_y \otimes G_x \end{bmatrix}.$$

Restriction and prolongation are implemented by *fixed* depthwise (per–channel) convolutions with stride 2 and their adjoints:

$$I_h^{2h} \equiv R = H, \qquad I_{2h}^{h} \equiv P = R^{\top}.$$

When the base measure is not uniform, we include the standard correction terms for H, G; the kernels remain fixed during training.

Coarse operator via Galerkin. Given the fine–grid stencil A^h, the coarse operator is defined by the Galerkin product

$$A^{2h} \;=\; R\,A^h\,P,$$

which we realize as three depthwise convolutions with frozen R, P and learnable (or problem–specific) A^h. This preserves the multigrid variational property and ensures spectral consistency across levels.

Smoothers (learned). Each level ℓ uses a light smoother Smooth_ℓ applied to the residual:

$$u \leftarrow u + \mathsf{Smooth}_\ell\big(f - Au\big),$$

where Smooth_ℓ is a depthwise 3×3 convolution followed by a pointwise linear map and GELU. We typically apply a small, fixed number of pre-/post-smoothing steps; these parameters are part of the architecture and not tied to a specific PDE.

V-cycle layer. One two–level V-cycle is used as a network layer that maps $u^h \mapsto u^h_{\text{new}}$:

1. *Pre-smoothing:* $u^h \leftarrow u^h + \mathsf{Smooth}_h(f^h - A^h u^h)$.
2. *Residual & restrict:* $r^h = f^h - A^h u^h, \;\; r^{2h} = R\, r^h$.
3. *Coarse solve (approx.):* solve/relax $A^{2h} e^{2h} = r^{2h}$ using several Smooth_{2h} steps.
4. *Prolong & correct:* $u^h \leftarrow u^h + P\, e^{2h}$.
5. *Post-smoothing:* $u^h \leftarrow u^h + \mathsf{Smooth}_h(f^h - A^h u^h)$.

Stacking these V-cycles with skip connections yields the full M2NO mapping $\mathcal{G}_\theta$.

Operator parameterization. In the learnable surrogate setting, A^h is realized as a depthwise convolution (per channel) optionally followed by a pointwise 1×1 coupling across channels to capture weak cross–variable interactions. The Galerkin coarse $A^{2h} = RA^hP$ is recomputed on-the-fly (no extra parameters). The linear maps B^ℓ in Eq. (M2NO) are implemented with 1×1 convolutions to control width and stabilize training.

Masking and boundaries. When a binary mask M (fluid/solid) is available, we zero out residuals on solids and apply masked restriction/prolongation by multiplying outputs with the down/up–sampled mask; this prevents leakage across boundaries. For periodic problems we switch all paddings to circular and omit boundary masks.

Complexity. Transfers R, P are fixed depthwise kernels, so each level costs $\mathcal{O}(HW)$ per channel. The overall V-cycle is linear in the number of pixels times the number of levels, with a small constant dominated by a few 3×3 convolutions and two inter–level transfers.

Training objective. For supervised operator learning, we minimize an L_2 loss between the network output $u = \mathcal{G}_\theta(a)$ and reference solutions $u^\star$:

$$\mathcal{L}(\theta) = \|u - \; u^\star\|_2^2,$$

optionally combined with weak regularizers (e.g., Laplacian penalty for stability) that do not alter the multigrid structure. Optimization uses standard first–order methods; training/validation splits and PDE specifications follow the datasets' protocols.

Practical notes. (i) All depthwise convolutions can be grouped by channels for efficiency. (ii) We keep R, P frozen to preserve the variational property and numerical stability. (iii) Coarse solves are always "inexact" (a few relaxations), which is critical for speed and acts as an effective inductive bias. (iv) Hyperparameters (number of levels, smoother steps, width) are selected per dataset and reported in the experiment config.

2.4. Physics-consistent residuals

We implement $\rho(u)$ as a sum of lightweight, differentiable directions, each returning a tensor with the same shape as u. All operators are implemented with local convolutions or FFT/iFFT and are fully backpropagable.

Smoothness residuals. Laplacian and biharmonic terms use fixed depthwise kernels:

$$\rho_{\text{lap}}(u) = -\Delta u, \qquad \rho_{\text{bi}}(u) = -\Delta(\Delta u).$$

The biharmonic variant more strongly suppresses checkerboard/ringing and is useful for derivative channels (e.g., u_x).

Anisotropic edge-preserving diffusion. Guided by an anchor u_a (default $u_a = u^{\text{LR}}$ upsampled), we use a Perona–Malik flux

$$\rho_{\text{aniso}}(u) = -\nabla \cdot \big(g(\|\nabla u_a\|)\, \nabla u\big), \quad g(s) = \tfrac{1}{1+(s/\kappa)^2}.$$

We modulate κ over time to emphasize coarse guidance early and relax later.

No-flux boundary residual. With a binary mask M (1 = fluid), we softly impose $\partial u/\partial n \approx 0$ on solids by

$$\rho_{\text{noflux}}(u) = -(1 - M)\, (\nabla u \cdot n), \qquad n \propto \nabla(1 - M),$$

where n is estimated via Sobel filters on the mask.

Spectrum alignment residual. We align the radial log-power spectrum of u to an anchor spectrum (from HR statistics or the upsampled coarse field). Let $\mathcal{F}$ be the FFT and $\mathcal{B}$ its inverse. We compute binned radial frequencies k, measure the discrepancy between $\log P(u)$ and the target, apply a robust Huber transform to obtain bin weights $W(k)$, and map back to pixels:

$$\rho_{\text{spec}}(u) \;=\; \mathcal{B}\big(W(k) \odot \mathcal{F}(u)\big).$$

We cache FFT frequency grids and radial bins for speed; complexity is $\mathcal{O}(HW \log(HW))$.

Composition and scheduling. The total physics residual is

$$\rho(u) = \sum_k w_k \, \rho_k(u; u^{\mathrm{LR}}, M),$$

with fixed w_k chosen by cross-validation. A cosine $\lambda(t)$ emphasizes stronger physics regularization early and decays later. For vector fields, divergence-related penalties can be added channel-wise; otherwise, scalar residuals are applied per channel.

2.5. Sampling updates and learned refinement

Each reverse step updates

$$u_{t-1} \;=\; u_t + \alpha_t \, e_t + \beta_t \, g_\theta(u_t, t) + \sigma_t \, \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I).$$

We use DDIM-style deterministic sampling by default ($\sigma_t = 0$). The schedules α_t, β_t are cosine-shaped and shared across datasets. The learned head g_θ is a shallow CNN with timestep conditioning (zero-initialized last layer) that compensates residual modeling error; $e_t = S_t(r(u_t))$ provides the dominant drift.

2.6. Training objective and optimization

Objective. We adopt the standard ε-prediction loss with a cosine noise schedule; the multigrid drift is used in the model's mean during training and inference. When training multiple variables jointly, we sample channels uniformly and sum losses.

Optimizer and schedules. We use Adam with a fixed learning rate 5×10^{-5}, batch size 64, and $> 10^5$ iterations. Gradients are clipped to a global norm of 1.0. We train with mixed precision (AMP) and, optionally, an EMA of parameters for evaluation.

Augmentation. We apply random flips and 90° rotations consistently to HR, LR, and masks. FFT-based residuals are computed after augmentation to maintain spectral consistency.

2.7. Architectural specifics and defaults

Timestep embedding. A sinusoidal embedding is projected by a two-layer MLP and used for: (i) FiLM modulation inside level-wise smoothers; (ii) per-level gates $w_\ell(t)$ (a linear projection plus sigmoid).

Projections. The input projection concatenates $[r,\, u_t]$ along channels and maps to a base width via a 1×1 convolution. The output projection maps the V-cycle output back to the state space with a zero-initialized 1×1 convolution.

Smoothers. Each smoother is a depthwise 3×3 convolution followed by pointwise scale/shift from FiLM, with lightweight normalization where needed (LayerNorm in feature space for stability on coarse levels).

Padding and boundaries. Convolutions use reflection padding; restriction/prolongation respect masks when present by zeroing solid regions after transfer. For coastal data, spectrum residuals use optional apodization windows to reduce ringing.

2.8. Efficiency and memory

Each S_t pass costs one pre-smoothing, per-level restrict–smooth–prolong with fixed kernels, and one post-smoothing. Transfers dominate neither FLOPs nor memory. With AMP and gradient checkpointing on the corrector, memory fits comfortably for $H = W = 256$ at batch size 64. Inference with $2-5$ steps takes a small multiple of a single CNN pass and is substantially faster than many-step diffusion samplers.

2.9. Ablations and toggles

We expose the following switches: (i) disable the multigrid corrector (use $e_t = 0$) to recover a pure diffusion baseline; (ii) remove individual physics terms ($\rho_{\mathrm{spec}}, \rho_{\mathrm{lap}}/\rho_{\mathrm{bi}}, \rho_{\mathrm{aniso}}, \rho_{\mathrm{noflux}}$); (iii) freeze the level gates $w_\ell(t) = 1$ to test time-gating; (iv) replace multiwavelet transfers with averaged pooling/nearest interpolation to test the role of spectral separation.

3. Experiment Details

3.1. Details for Benchmarks

3.1.1. Navier-Stokes Equation

Navier-Stokes equations describe the motion of viscous fluid substances [6, 9, 18], and are used to model ocean currents, weather,and air flow. We experiment on the 2D Navier-Stokes equation for a viscous, incompressible fluid in vorticity form on the unit torus,which takes the following form:

$$\begin{aligned} \frac{\partial w(x,t)}{\partial t} + u(x,t) \cdot \nabla w(x,t) - \nu \Delta w(x,t) &= f(x), \\ \nabla \cdot u(x,t) &= 0, \\ w_0(x) &= w(x, t=0), \\ \text{for} \quad x \in (0,1)^2,\ t \in (0,T]. & \end{aligned}$$

where $w(x,t)$ is the vorticity,$u(x,t)$ is the velocity field,v is the kinematic viscosity, and $f(x)$ is a forcing function.

Dataset Construction. The NS2D datasets comprises a total of 24,000 instantaneous snapshots generated under the governing equations.For empirical evaluation,the dataset was divided following a 0.98/0.01/0.01 split for training, validation, and testing, respectively. This partitioning ensures a diverse coverage of spatial structures while providing sufficient data volume for effective super-resolution reconstruction.

Table 1. **Hyperparameter configurations for baselines and ReMD on benchmarks.** All models are carefully tuned within a comparable computational budget to ensure fair evaluation.

HYPERPARAMETER	NS	ERA5	OCEAN
BASE_CHANNELS	64	64	64
C	4	4	4
GRID_LEVEL	[8,8,4,2,1]	[8,8,4,2,1]	[8,8,4,2,1]
K	4	4	4
TARGET	'UNETMODELSWIN'	'UNETMODELSWIN'	'UNETMODELSWIN'
MODEL_CHANNELS	160	160	160
CHANNEL_MULT	[1,2,2,4]	[1,2,2,4]	[1,2,2,4]
NUM_RES_BLOCKS	[2,2,2,2]	[2,2,2,2]	[2,2,2,2]
ATTENTION_RESOLUTION	[64,32,16,8]	[64,32,16,8]	[64,32,16,8]
STEPS	2 STEPS/5 STEPS	2 STEPS/5 STEPS	2 STEPS/5 STEPS
KAPPA	2	2	2
SCHEDULE_NAME	'EXPONENTIAL'	'EXPONENTIAL'	'EXPONENTIAL'
LATENT_FLAG	'FALSE'	'FALSE'	'FALSE'

3.1.2. ERA5

The ERA5 [10] dataset, developed by the European Centre for MediumRange Weather Forecasts (ECMWF), provides hourly estimates of various atmospheric, land, and oceanic climate variables on a global scale. For our experiments, we focus on the key variables components U.This global climate reanalysis dataset consisting of 20 days of data,each with 24 temporal slices at a spatial resolution of 721×1440. In this study, the raw data were preprocessed through variable extraction, temporal flattening, and sliding-window cropping (128×128 patches with a stride of 64), resulting in standardized samples of shape (100800, 128, 128, 1) for model training and evaluation.

Dataset Construction. To form a computationally manageable benchmark, a random subset of 10,000 samples was drawn from the full collection and used as our experimental dataset. This subset was further divided into training, validation, and test sets using a 0.98/0.01/0.01 allocation. The sampling process ensured uniform coverage across different spatial locations and temporal indices.

3.1.3. The Global Ocean Surface Velocity Dataset

The Global Ocean Surface Velocity Dataset – 1997 [3] is derived from the Copernicus Marine Service (CMEMS) Global Ocean Physics Reanalysis product (GLOBAL_MULTIYEAR_PHY_001_030 [3]), providing northward (vo) velocity fields for the entire year of 1997.The original data have a spatial resolution of $1/12°$ (approximately 8 km) and represent surface-layer conditions at a depth of about 0.494 m.During data preprocessing, the original global surface velocity fields were rigorously filtered and spatially segmented. Regions containing any land pixels were removed based on a land–sea mask, ensuring that only 100% oceanic areas were retained for physical consistency and dynamical purity. The remaining oceanic domain was then partitioned into overlapping patches of size 128×128 with a stride of 64, providing high-resolution spatial sampling and enhanced data diversity.

Dataset Construction. From this collection, 10,000 samples were randomly selected to construct the benchmark used in our experiments. These samples were subsequently divided into training, validation, and test sets following a 0.98/0.01/0.01 ratio. The random selection ensured that the subset preserved the geographical diversity and multiscale variability inherent in global ocean circulation patterns.

3.2. Details of Degradation Model

To synthesize realistic Low-Resolution (LR) inputs from the High-Resolution (HR) ground truth, we apply a standard degradation pipeline $\mathcal{D}$, which models the image acquisition process through blurring and downsampling. Formally, given an HR sample $\mathbf{x}_{\text{HR}} \in \mathbb{R}^{C\times H\times W}$, the corresponding LR counterpart $\mathbf{x}_{\text{LR}}$ is obtained via:

$$\mathbf{x}_{\text{LR}} = (\mathbf{x}_{\text{HR}} \circledast \mathbf{k}_\sigma) \downarrow_s, \tag{1}$$

where $\circledast$ denotes the depthwise convolution, $\mathbf{k}_\sigma$ is an isotropic Gaussian kernel, and $\downarrow_s$ represents the bicubic downsampling operation with a scale factor s.

Gaussian Kernel. The blur kernel $\mathbf{k}_\sigma$ of size $K \times K$ is

defined as:

$$\mathbf{k}_\sigma(i,j) = \frac{1}{Z}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), \quad \text{where } i,j \in \left[-\frac{K-1}{2}, \frac{K-1}{2}\right] \tag{2}$$

Here, Z is a normalization constant ensuring $\sum_{i,j}\mathbf{k}_\sigma(i,j) = 1$. In our implementation, we set the kernel size $K = 7$ and the standard deviation $\sigma = 1.2$.

Implementation Details. In our pipeline,the Gaussian filter is applied individually to each single-channel input.This variable-independent processing strategy preserves the physical purity of the data,ensuring that the degradation of one physical quantity(e.g.,zonal velocity) is not influenced by others. Following the blurring step, the features are downsampled using bicubic interpolation (with antialiasing enabled). The specific scale factors used for each dataset are detailed in Table 2.

Table 2. Degradation settings and resolution changes for the datasets used in this study.

Dataset	Scale Factor (s)	Resolution Change
NS	$\times 2$	$64 \times 64 \rightarrow 32 \times 32$
ERA5	$\times 4$	$128 \times 128 \rightarrow 32 \times 32$
Ocean	$\times 4$	$128 \times 128 \rightarrow 32 \times 32$

References

[1] Shuhao Cao, Francesco Brarda, Rui Peng Li, and Yuanzhe Xi. Spectral-refiner: Accurate fine-tuning of spatiotemporal fourier neural operator for turbulent flows. In *International Conference on Learning Representations (ICLR)*, 2025. 1

[2] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, 2021. 2

[3] E.U. Copernicus Marine Service Information (CMEMS). Global Ocean Physics Reanalysis. Marine Data Store (MDS), 2023. 6

[4] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[5] Yanan Guo, Junqiang Song, Xiaoqun Cao, Chuanfeng Zhao, and Hongze Leng. Physics field super-resolution reconstruction via enhanced diffusion model and fourier neural operator. *Theoretical and Applied Mechanics Letters*, 2025. Open access; available online 14 June 2025; CC BY-NC-ND 4.0. 1

[6] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021. 5

[7] Qiuchang Han, Xingliang Jiang, Yang Zhao, Xudong Wang, Zhijin Li, and Renhe Zhang. Generative diffusion model-based downscaling of observed sea surface height over kuroshio extension since 2000. *arXiv preprint arXiv:2408.12632*, 2024. 2

[8] Xi Han, Fei Hou, and Hong Qin. Ugrid: An efficient-and-rigorous neural multigrid solver for linear pdes. In *ICML*. OpenReview.net, 2024. 1

[9] Juncai He, Xinliang Liu, and Jinchao Xu. Mgno: Efficient parameterization of linear operators via multigrid. In *ICLR*. OpenReview.net, 2024. 1, 5

[10] Hans Hersbach, Bill Bell, Paul Berrisford, Gionata Biavati, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Iryna Rozum, Dick Schepers, Adrian Simmons, Côme Soci, Dick Dee, and Jean-Noël Thépaut. ERA5 hourly data on pressure levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS), 2023. Accessed on 04-Oct-2025. 6

[11] Jun-Ting Hsieh, Shengjia Zhao, Stephan Eismann, Lucia Mirabella, and Stefano Ermon. Learning neural PDE solvers with convergence guarantees. In *ICLR (Poster)*. OpenReview.net, 2019. 1

[12] Peiyan Hu, Rui Wang, Xiang Zheng, Tao Zhang, Haodong Feng, Ruiqi Feng, Long Wei, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Wavelet diffusion neural operator. In *International Conference on Learning Representations (ICLR)*, 2025. ICLR 2025 Poster; OpenReview. 2

[13] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 130291–130323, 2024. 2

[14] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. 1

[15] Jaewon Lee and Kyong Hwan Lee. Local texture estimator for implicit representation function. In *CVPR*, 2022. 2

[16] Xiaowei Li, Na Tao, Dan Zhang, Yan Peng, and Yaoran Chen. Refined sea surface height reconstruction using a diffusion-based super-resolution method. *SSRN Electronic Journal*, 2024. 2

[17] Xiaowei Li, Na Tao, Dan Zhang, Wenhu Liu, Yan Peng, Yong Cao, and Yaoran Chen. Multi-scale boundary-enhanced diffusion network for high-resolution sea surface height reconstruction. *Physics of Fluids*, 37(2):026610, 2025. 2

[18] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020. 1, 5

[19] Zhihao Li, Zhilu Lai, Xiaobo Zhang, and Wei Wang. M2no: An efficient multi-resolution operator framework for dynamic multi-scale pde solvers. 2024. 1

[20] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 2

[21] Xiaoyi Liu and Hao Tang. Difffno: Diffusion fourier neural operator for arbitrary-scale image super-resolution. *arXiv preprint arXiv:2411.09911*, 2024. to appear in CVPR 2025. 1

[22] Xihaier Luo, Xiaoning Qian, and Byung-Jun Yoon. Hierarchical neural operator transformer with learnable frequency-aware loss prior for arbitrary-scale super-resolution. *arXiv preprint arXiv:2405.12202*, 2024. 1

[23] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *CVPR*, 2023. 2

[24] Youssef Shehata, Benjamin Holzschuh, and Nils Thuerey. Improved sampling of diffusion models in fluid dynamics with tweedie's formula. In *International Conference on Learning Representations (ICLR)*, 2025. ICLR 2025 Poster; OpenReview. 2

[25] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow reconstruction. *Journal of Computational Physics*, 478:111972, 2023. 2

[26] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 2

[27] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 2

[28] Sifan Wang, Zehao Dou, Tong-Rui Liu, and Lu Lu. Fundiff: Diffusion models over function spaces for physics-informed generative modeling. *arXiv preprint arXiv:2506.07902*, 2025. 2

[29] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. Diner: Disorder-invariant implicit neural representation. In *CVPR*, 2023. 2

[30] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36:13294–13307, 2023. 2

[31] Wang Yufei, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C. Kot, and Bihan Wen. Sinsr: Diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2