



OPEN Generative super-resolution of turbulent flows via stochastic interpolants

Martin Schiødt¹✉, Nikolaj T. Mücke² & Clara M. Velte¹

Capturing the intricate multiscale features of turbulent flows remains a fundamental challenge due to the limited resolution of experimental data and the computational cost of high-fidelity simulations. In many practical scenarios only coarse representations of the flows are feasible, leaving crucial fine-scale dynamics unresolved. This study addresses that limitation by leveraging generative models to perform super-resolution of velocity fields and reconstruct the unresolved scales from low-resolution conditionals. In particular, the recently formalized stochastic interpolants are employed to super-resolve a case study of two-dimensional turbulence. Key to our approach is the iterative application of stochastic interpolants over local patches of the flow field, that enables efficient reconstruction without the need to process the full domain simultaneously. The patch-wise strategy is shown to yield physically consistent super-resolved flow snapshots, and key statistical quantities – such as the kinetic energy spectrum – are accurately recovered. Moreover, the patch-wise approach is observed to produce super-resolutions of a quality comparable to those produced using a full field approach, and, in general, stochastic interpolants are observed to outperform contesting generative models across a range of metrics. Although only demonstrated for a 2D case study, these results highlight the potential of using stochastic interpolants to super-resolve turbulent flows.

Super-resolution of turbulent flows is essential for bridging the gap between the limited resolution of experimental measurements or low-resolution simulations and the rich, multiscale dynamics inherent to turbulence. Many practical simulations, such as Large Eddy Simulations (LES) or low-cost numerical models, cannot afford to resolve all relevant scales due to computational constraints¹. Super-resolution techniques enable the reconstruction of fine-scale structures from low-resolution data, enhancing physical fidelity and enabling accurate analysis of quantities like energy spectra, vorticity, and dissipation. This is particularly valuable for data-driven modeling, control, and diagnostics of complex fluid systems².

In parallel with the growing influence of machine learning in imaging and language modeling, deep learning techniques have been increasingly adopted for super-resolving turbulent flows, with studies reporting significant performance gains over conventional methods³. Among these, deterministically trained convolutional neural networks (CNNs) are widely used due to their strong capabilities in feature extraction. Pioneering this approach in the field of turbulence, Fukami et al.^{4,5} applied deep CNNs to reconstruct various two-dimensional flows. While their model recovered flow statistics, such as the kinetic energy spectrum, fairly well, it exhibited non-physical artifacts and struggled to capture small-scale structures. To address these limitations, Liu et al.⁶ incorporated temporal information as a conditional input to the model. This extension yielded improved results, but the model continued to face challenges in regions dominated by viscous effects. Zhou et al.⁷ further enhanced the model by coupling it with an approximate deconvolution method⁸, and extended the analysis to a case study of three-dimensional turbulence.

Although deterministic methods, such as the aforementioned, have shown promise in super-resolving turbulent flows, recent efforts have increasingly focused on the application of generative models. Generative models constitute a class of algorithms designed to approximate the probability distribution underlying a given dataset. Once this, potentially conditional, distribution is learned, the model can synthesize new realizations by sampling from the learned distribution, yielding ensembles that are statistically consistent with the original data. Due to the stochastic nature of the sampling procedure, generative models are inherently non-deterministic. Among the most widely used generative frameworks are generative adversarial networks (GANs)⁹, and diffusion models (DMs)¹⁰. In the context of super-resolution, generative models produce high-resolution fields conditioned on corresponding low-resolution inputs.

¹DTU Construct, Technical University of Denmark, Kongens Lyngby 2800, Denmark. ²Centrum Wiskunde & Informatica, Amsterdam 1098 XG, The Netherlands. ✉email: maschi@dtu.dk

Inspired by the work of Ledig et al.¹¹, Deng et al.¹² applied GANs to super-resolve benchmark cases of two-dimensional velocity fields. Subsequently, Subramaniam et al.¹³ extended this methodology to reconstruct both pressure and velocity fields in three-dimensional homogeneous isotropic turbulence, enhancing the resolution from 16^3 to 64^3 . Later, Kim et al.¹⁴ used GANs to super-resolve slices of three-dimensional turbulent flow fields. Their results demonstrated a marked improvement in statistical accuracy relative to comparable CNN architectures. More recently, DMs have been shown to outperform GANs in augmenting incomplete or corrupted measurements of two-dimensional snapshots from three-dimensional turbulent flows¹⁵. Furthermore, an expanding body of work has successfully employed DMs to predict and super-resolve turbulent flows under a variety of configurations^{16–18}.

In this study, we employ stochastic interpolants (SIs)¹⁹ to perform super-resolution of the velocity field in a two-dimensional case study of the Kolmogorov flow. Compared to DMs, SIs offer a more direct mapping between two distributions, as their inference process is initialized with an observed data point rather than Gaussian noise. Although stochastic interpolants remain a relatively recent development, especially within the context of fluid dynamics, they have been applied in a few studies to forecast and super-resolve canonical two-dimensional flows^{20,21} and to recover state variables from sparse and noisy observations²².

We hypothesize that SIs provide improved performance over DMs due to their direct way of connecting two arbitrary distributions. This hypothesis is empirically supported for the 2D case study presented in this work, where we train a SI to map low-resolution samples to corresponding high-resolution samples. Moreover, with the potential to extend the applicability of SIs to more complex settings – specifically, three-dimensional turbulence – we introduce a patch-wise strategy that iteratively super-resolves localized subdomains of the full flow field. This localized approach effectively mitigates the computational burden associated with the increased input dimensionality that arises from finer grid resolutions, a challenge that becomes particularly acute in three-dimensional applications.

The paper is structured as follows: section [Preliminaries](#) presents the problem setting, provides a brief overview of the fundamentals of SIs, and details the simulation of training and test data. Our main contribution is introduced in section [Methodology: Stochastic interpolants for turbulence super-resolution](#), namely the full-field and patch-wise super-resolution methods using SIs. In section [Results & discussion](#), we evaluate our methodology and compare it with alternative approaches, including diffusion models and flow-matching. We examine both individual super-resolution snapshots and overall statistical performance. Finally, section [Conclusion](#) summarizes our findings and conclusions.

Preliminaries

This section presents the governing equations of motion for the case study considered in the current work. It provides a brief introduction to the stochastic interpolant framework, and describes the simulation methodology. Moreover, the procedure for generating the datasets used to train and evaluate the developed models is detailed.

Problem setting

Super-resolution via SIs is demonstrated on a two-dimensional Kolmogorov flow case study. The flow dynamics are governed by the incompressible Navier–Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} + \mathbf{f}, \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (1b)$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $p(\mathbf{x}, t)$ the pressure and $\mathbf{f}(\mathbf{u})$ the external forcing, specified as

$$\mathbf{f}(\mathbf{u}) = \sum_{k=4}^6 \sin(ky) \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0.1 \mathbf{u}. \quad (2)$$

Adopting the definition of the Reynolds number used by Lucas & Kerswell²³ for the two-dimensional Kolmogorov flow, we have

$$\text{Re} := \frac{\sqrt{\chi}}{\nu} \left(\frac{L_y}{2\pi} \right)^{3/2}, \quad (3)$$

where $\chi \approx 3$ denotes the forcing amplitude, ν the kinematic viscosity, and L_y the extent of the domain in the y -direction. The velocity field \mathbf{u} obtained through direct numerical simulation (DNS) of [1](#) serves as the reference target for training and evaluating the data-driven models developed in this study. These models aim to reconstruct the statistical features of \mathbf{u} from a filtered counterpart, $\hat{\mathbf{u}}$, which retains only the large-scale flow structures. Although presented here in two dimensions as a proof of concept, the SI approach detailed in the following sections is formulated with the aim of generalization to three-dimensional flows. Indeed, this extension is a key perspective of the present study.

Stochastic interpolants

Here, we briefly outline the stochastic interpolant method, as presented in²⁰. The method was originally presented in¹⁹ and expanded in^{20,24}.

The SI framework provides an approach for sampling from a conditional distribution by constructing a generative model that transports a point mass to a sample from a target distribution. We aim to generate samples from the conditional distribution,

$$\rho(\mathbf{x}_1 | \mathbf{x}_0) = \frac{\rho(\mathbf{x}_0, \mathbf{x}_1)}{\rho(\mathbf{x}_0)}, \quad (4)$$

where $\rho(\mathbf{x}_0)$ is the marginal distribution of \mathbf{x}_0 and $\rho(\mathbf{x}_0, \mathbf{x}_1)$ represents the joint distribution of \mathbf{x}_0 and \mathbf{x}_1 . \mathbf{x}_0 samples are referred to as base samples and \mathbf{x}_1 samples are referred to as target samples.

The core of the method relies on the stochastic interpolant \mathbf{I}_τ , defined as

$$\mathbf{I}_\tau = \alpha_\tau \mathbf{x}_0 + \beta_\tau \mathbf{x}_1 + \sigma_\tau \mathbf{W}_\tau, \quad \tau \in [0, 1], \quad (5)$$

where τ is denoted *pseudo-time*. \mathbf{W}_τ is a standard Wiener process independent of $(\mathbf{x}_0, \mathbf{x}_1)$, and $\alpha_\tau, \beta_\tau, \sigma_\tau \in C^1([0, 1])$ are pseudo-time-dependent coefficients satisfying temporal boundary conditions:

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \sigma_1 = 0. \quad (6)$$

These boundary conditions ensure that $\mathbf{I}_0 = \mathbf{x}_0$ and $\mathbf{I}_1 = \mathbf{x}_1$, creating a bridge between the point mass at \mathbf{x}_0 and the conditional distribution $\rho(\mathbf{x}_1 | \mathbf{x}_0)$. The key insight is that there exists a drift term, b_τ , such that the conditional distribution of \mathbf{I}_τ given \mathbf{x}_0 can be generated by solving the stochastic differential equation (SDE):

$$d\mathbf{X}_\tau = b_\tau(\mathbf{X}_\tau, \mathbf{x}_0)d\tau + \sigma_\tau d\mathbf{W}_\tau, \quad \tau \in [0, 1], \quad \mathbf{X}_{\tau=0} = \mathbf{x}_0. \quad (7)$$

In particular, samples from the distribution $\rho(\mathbf{x}_1 | \mathbf{x}_0)$ correspond to samples from the target distribution $\rho(\mathbf{x}_1 | \mathbf{x}_0)$ owing to the construction of the interpolant.

It can be shown that the drift term that provides the desired property is the unique minimizer of the objective:

$$\arg \min_{b_\tau} \mathcal{L}(b_\tau) = \int_0^1 \mathbb{E} [\|b_\tau(\mathbf{I}_\tau, \mathbf{x}_0) - \mathbf{R}_\tau\|^2] d\tau, \quad (8)$$

with $\mathbf{R}_\tau = \dot{\alpha}_\tau \mathbf{x}_0 + \dot{\beta}_\tau \mathbf{x}_1 + \dot{\sigma}_\tau \mathbf{W}_\tau$. This objective can be estimated empirically using samples from the joint distribution, making the drift learnable using standard regression techniques with neural networks. Therefore, we parameterize the drift term as a neural network, b_θ with weights θ and minimize an approximation of Eq. 8 with respect to θ :

$$\arg \min_{\theta} L(\theta) = \frac{1}{N_\tau N_{\text{train}}} \sum_{i=1}^{N_\tau} \sum_{j=1}^{N_{\text{train}}} \|b_\theta(\mathbf{I}_{\tau_i}^j, \mathbf{x}_0^j, \tau_i) - \mathbf{R}_{\tau_i}^j\|^2, \quad \mathbf{I}_{\tau_i}^j = \alpha_{\tau_i} \mathbf{x}_0^j + \beta_{\tau_i} \mathbf{x}_1^j + \sigma_{\tau_i} \mathbf{W}_{\tau_i}, \quad \mathbf{R}_{\tau_i}^j = \dot{\alpha}_{\tau_i} \mathbf{x}_0^j + \dot{\beta}_{\tau_i} \mathbf{x}_1^j + \dot{\sigma}_{\tau_i} \mathbf{W}_{\tau_i}, \quad (9)$$

where $(\mathbf{x}_0^j, \mathbf{x}_1^j) \sim \rho(\mathbf{x}_0, \mathbf{x}_1)$, N_{train} is the number of training samples, and N_τ is the number of discrete pseudo-time points. For more details on training stochastic interpolants, see^{20,24}.

The architecture of b_θ , as well as the coefficients $\alpha_\tau, \beta_\tau, \sigma_\tau$, which together define our stochastic interpolant will be detailed in section [Methodology: Stochastic interpolants for turbulence super-resolution](#).

Generating training and test sets

In this work, stochastic interpolants are trained to reconstruct simulated velocity fields, \mathbf{u} , by super-resolving the filtered counterpart $\tilde{\mathbf{u}}$, which represents the corresponding low-resolution field. Thus, training and test sets are produced to consist of pairs $(\mathbf{x}_0, \mathbf{x}_1) = (\tilde{\mathbf{u}}, \mathbf{u})$. In the current study we generate 2,000 sample pairs of \mathbf{u} and $\tilde{\mathbf{u}}$ for training our models. An additional 400 sample pairs are generated for evaluating model performance. We do not create a separate validation set, as we do not perform hyperparameter tuning; instead, we adopt fixed, prior-chosen hyperparameter values throughout. The target and base samples are simulated via the procedure detailed in the following subsections.

Numerical simulation

To produce \mathbf{u} we first convert **1** to its vorticity–streamfunction formulation²⁵ and solve the governing equations on a fully periodic domain $(x, y) \in \Omega = [0, 2\pi]^2$ using the Fourier Galerkin method²⁶. Here, all fields are represented as truncated Fourier series, and spatial derivatives are computed exactly in spectral space due to the periodic boundary conditions.

The non-linear convective term, typically expressed in vorticity form as $\mathbf{u} \cdot \nabla \omega$ is evaluated pseudospectrally. This is done by transforming the gradient of the vorticity $\nabla \omega$ and the velocity components $\mathbf{u} = (u, v)$ from spectral space to physical space using an inverse Fourier transform. The non-linear product $\mathbf{u} \cdot \nabla \omega$ is then computed pointwise in physical space, and the result is transformed back to spectral space using a forward Fourier transform. This approach avoids the expensive convolution sums that would arise from computing the non-linear product directly in spectral space²⁶. To suppress aliasing errors in the non-linear term, a dealiasing technique, namely the 2/3-rule, is employed, where the highest one-third of wavenumbers are zeroed out after transforming back to spectral space²⁷. This ensures numerical stability and accuracy in the pseudospectral evaluation.

The simulation is initialized with a random seed in spectral space, where Hermitian symmetry is enforced. After evolving the simulation to a statistically steady state, the velocity field \mathbf{u} is sampled on a uniform 128×128

grid at temporally decorrelated intervals, determined via the autocorrelation function. The Reynolds number is set to $Re = 1000$, and time integration is performed using a fourth-order Runge–Kutta scheme with a fixed timestep $\delta t = 0.025$. For further details of the simulation, we refer to our code repository which is available online.

Note that the specific choice of numerical solver is not crucial for the presented methodology as it is only a means to produce training data. Thus, the only requirement is that the numerical simulation provides accurate training data with respect to the quantities of interest. E.g. the method described above would not be suitable for a 3D case with non-periodic boundary conditions.

Producing low-resolution samples

To generate the filtered state $\tilde{\mathbf{u}}$ from \mathbf{u} , a series of steps are applied. First, we apply a lowpass filter to \mathbf{u} with a cutoff frequency of $k_{cutoff} = 8$. This operation retains only the low-frequency modes of the velocity field, effectively removing small-scale variations. The filtering is defined by the Fourier coefficients:

$$\hat{\mathbf{u}}_{i,k\ell} = \begin{cases} \hat{\mathbf{u}}_{i,k\ell}, & \text{for } |k|, |\ell| \leq k_{cutoff} \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where $\kappa = [k \ \ell]^T$ is the spectral wavenumber and subscript $i = \{1, 2\}$ denotes the velocity field components u and v . Following the filtering, the field is downsampled onto a 16×16 grid by retaining every 8th grid point in both the x – and y –coordinates, discarding the remaining points. In essence, the velocity field on the 16×16 grid represents the limited resolution data available within a low-resolution simulation.

Methodology: Stochastic interpolants for turbulence super-resolution

This section develops two complementary models for turbulent flow super-resolution using stochastic interpolants: SI_{full} , which processes the entire velocity field simultaneously, and SI_{patch} , which employs a patch-wise strategy designed for computational scalability. Our objective is to construct a framework that allows reconstruction of high-fidelity velocity fields from limited experimental measurements or low-resolution simulation data. The SI models are trained to approximate the conditional distribution $\rho(\mathbf{u} | \tilde{\mathbf{u}})$, enabling generation of statistically consistent high-fidelity samples from coarse inputs.

A key constraint of the stochastic interpolant framework is that the base and target samples (\mathbf{x}_0 and \mathbf{x}_1) must reside in the same vector space²⁰. To satisfy this requirement, we upsample $\tilde{\mathbf{u}}$ prior to training. Specifically, we learn to sample from $\rho(\mathbf{u} | \tilde{\mathbf{u}}) = \rho(\mathbf{u} | \text{Up}(\tilde{\mathbf{u}}))$. Note that the equality holds due to the deterministic nature of the chosen upsampling operator, Up. We employ cubic interpolation to transform the filtered and downsampled velocity field from the 16×16 grid back to the original 128×128 resolution, though alternative interpolation schemes (e.g., linear) are equally viable.

In the following subsections, we detail the implementation of SI_{full} and SI_{patch} for sampling from $\rho(\mathbf{u} | \tilde{\mathbf{u}})$.

Full field super-resolution

The full field model, SI_{full} , directly super-resolves the entire velocity field in a single forward integration of the governing SDE (7). We define the stochastic interpolant with base samples $\mathbf{x}_0 = \tilde{\mathbf{u}} = \text{Up}(\tilde{\mathbf{u}})$ and target samples $\mathbf{x}_1 = \mathbf{u}$.

While conceptually straightforward, this approach faces computational limitations as grid resolution increases. The SDE integration required for sample generation scales poorly with domain size, which becomes particularly problematic for three-dimensional applications where memory and computational requirements are prohibitive. These scalability constraints motivate the development of the patch-wise strategy described below.

Patch-wise super-resolution

The patch-wise approach, SI_{patch} , addresses the computational limitations of SI_{full} by decomposing the super-resolution task into smaller, manageable subproblems. Rather than processing the entire domain simultaneously, SI_{patch} is applied iteratively to super-resolve localized patches of the velocity field, enabling application to high-resolution three-dimensional flows where the full field method becomes computationally intractable. Thus, where SI_{full} is applied to reconstruct \mathbf{u} from $\tilde{\mathbf{u}}$, SI_{patch} is applied to reconstruct any subfield \mathbf{u}_j from $\tilde{\mathbf{u}}_j$, where

$$\mathbf{u}_j(t, x, y) = \mathbf{u}(t, x, y), \quad \tilde{\mathbf{u}}_j(t, x, y) = \tilde{\mathbf{u}}(t, x, y), \quad \text{for } (x, y) \in \Omega_j, \quad (11)$$

and the subdomain, Ω_j , is defined through the partition

$$\Omega = \cup_j \Omega_j, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j, \quad (12)$$

with Ω denoting the full spatial domain. Although applying SI_{patch} iteratively across Ω generates statistically consistent super-resolved velocity fields, initial implementations exhibited shortcomings at patch boundaries. Specifically, non-physical discontinuities were observed at patch interfaces, a behavior that was especially pronounced in spatial-gradient fields, such as the vorticity computed from the generated velocity field. To address this issue we expand our model to consist of two separately trained patch models when seeking the generation of a full velocity field. The first patch model, which we term the *free*-generator, can be applied to super-resolve the field at any arbitrary patch, Ω_j , using the low-resolution neighboring patches as conditionals. The second patch model, which we term the *cond*-generator (conditional generator), may be applied to super-resolve patches, where the neighboring patches have been super-resolved using the free-generator.

With these distinct submodules of SI_{patch} , the process of super-resolving the full velocity field can be divided into two stages:

- **Stage 1:** The free-generator is applied to super-resolve patches arranged in a checkerboard pattern, conditioning each patch on neighboring low-resolution patches. This yields a velocity field that is partially super-resolved (see top and middle rows of Fig. 1);
- **Stage 2:** The cond-generator super-resolves the remaining patches conditioning on the high-resolution patches generated in Stage 1 (see middle and bottom rows of Fig. 1). Crucially, during training, this model uses neighbor patches from x_1 rather than x_0 , emulating the process of using super-resolved data from the free-generator as boundary conditions for the cond-generator.

In principle, the two-stage approach defines a sequential algorithm for super-resolving a given velocity field. However, because the patches processed within each stage are independent of one another, the super-resolution procedure can be carried out in parallel across patches in both stages, only requiring synchronization between the two stages.

The two-stage approach was observed to generate more physically consistent super-resolutions than when only the free-generator was used across Ω . Together, the free-generator and cond-generator thus define SI_{patch} , where boundary artifacts have effectively been mitigated while maintaining the same statistical objectives as SI_{full} , but with superior computational scalability. For the current work we choose a patch size of 32×32 , signifying that the full velocity field may be reconstructed by super-resolving 16 separate patches. The patch edge length, $\ell_{patch} = \pi/2$, relates to the flow characteristic length scale $\ell_{flow} = L_y/2\pi = 1^{23}$ by $\ell_{patch}/\ell_{flow} = \pi/2$. Hence, the chosen patch size should adequately resolve all relevant flow scales. A systematic assessment of how patch size influences the efficiency of the patch-wise method is left for future work.

Configuration of stochastic interpolants

Inspired by the results of²⁰, we choose, for both models, the interpolant coefficients

$$\alpha_\tau = 1 - \tau, \quad \beta_\tau = \tau^2, \quad \sigma_\tau = 0.1(1 - \tau), \quad (13)$$

such that they satisfy (6).

Network architecture

To parameterize b_θ each model employs a UNet architecture, which was originally introduced by Ronneberger et al.²⁸. Our UNet architecture (Fig. 2), largely based on the approach presented in²¹, is composed of a series of convolutional and ConvNeXt²⁹ layers. The UNets of SI_{full} and SI_{patch} differ solely in the state conditioning,

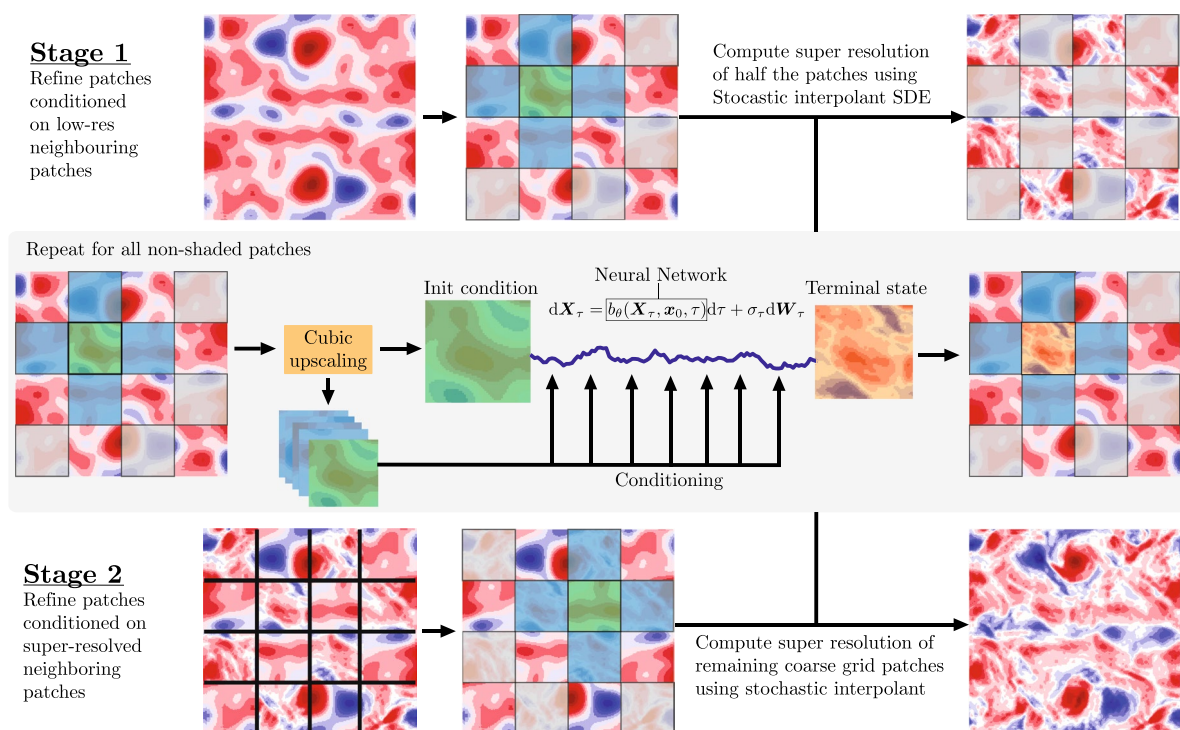


Fig. 1. Visualization of the two stages in SI_{patch} . Note that the states are visualized with a single channel for visual clarity. The actual data used consists of two channels – velocity in the horizontal and vertical direction.

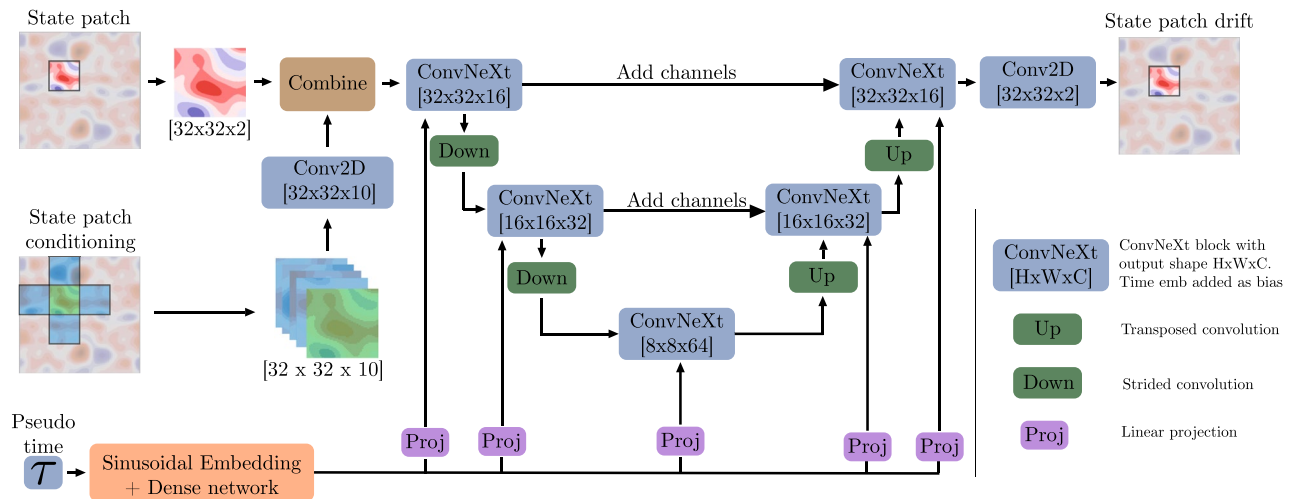


Fig. 2. Visualization of the UNet architecture within the free- and cond-generators of SI_{patch} .

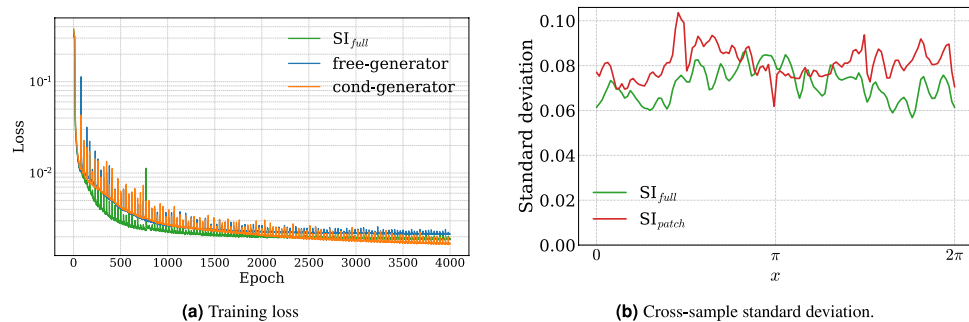


Fig. 3. (a) Training loss (9) evaluated at every epoch, illustrating convergence of each SI model. (b) Standard deviation of the velocity v at $y = \pi$, computed over 400 super-resolved samples generated from the same low-resolution input of a representative snapshot, indicating the models' ability to produce a diverse set of states.

where SI_{full} takes the full $x_0 \in \mathbb{R}^{128 \times 128 \times 2}$ as conditional input, whereas SI_{patch} takes only 5 field-patches, each of size $32 \times 32 \times 2$, as conditional input.

Throughout the network, we employ the GELU activation function³⁰. The pseudo-time variable τ is embedded using a sinusoidal positional encoding, which is then processed by a shallow neural network. This time embedding is incorporated as a conditioning input at each ConvNeXt layer as a bias within the UNet.

Divergence-free projection

As each model is unlikely to produce a divergence free field, the output $x_{\tau=1}$ is filtered using the Helmholtz-Hodge decomposition³¹. For any field F the method returns

$$F_{div=0} = F - \nabla \phi, \quad (14)$$

where ϕ solves

$$\nabla^2 \phi = \nabla \cdot F. \quad (15)$$

Since our velocity field is periodic, Eqs. (14)-(15) are solved in spectral space. In other flows, the decomposition may not be as effective, and other methods to remove non-zero divergence may be needed. We refer to²¹ for a discussion of alternative projection methods.

Training

Each stochastic interpolant is trained over 4000 epochs using a batch size of 40 (2% of the training set). We employ an AdamW optimizer³², and apply a linear warm-up learning rate scheduler for 50 epochs. The warm-up is succeeded by a cosine annealing learning rate scheduler³³, with a restart period of 30 epochs. Figure 3a displays the training loss pr. epoch for the SI_{full} model and the free- and cond-generators separately. Each model is observed to converge within 4000 epochs. Figure 3b shows the standard deviation of the velocity v at

$y = \pi$, $x \in [0, 2\pi]$, computed over 400 super-resolved samples generated from the same low-resolution input of a representative snapshot. The non-zero standard deviation demonstrates that the trained models are able to generate an ensemble of plausible states from a given input. For additional implementation details, please refer to the GitHub repository linked in this work.

Results & discussion

This section presents the results of applying SI_{full} and SI_{patch} to super-resolve the velocity field in the Kolmogorov flow case study. To represent the full field super-resolution of a snapshot we use the notation x_1^f and x_1^p for respectively SI_{full} and SI_{patch} . The trained models are evaluated on a test set consisting of 400 decorrelated flow snapshots. We first demonstrate that the models produce reasonable super-resolved versions of individual snapshots, followed by an analysis of statistical performance over the full test set.

Snapshot evaluation

For a given snapshot, the super-resolved velocity field is inferred by forward-integrating the SDE in Eq. (7) using the Heun SDE integrator³⁴, with 100 pseudo-timesteps and $x_0 = \bar{u} = \text{Up}(\bar{u})$ as the initial condition. The resulting high-resolution velocity field for a representative snapshot is shown in Fig. 4a. While the models SI_{full} and SI_{patch} are not designed to exactly reproduce x_1 , they are observed to produce super-resolutions that match

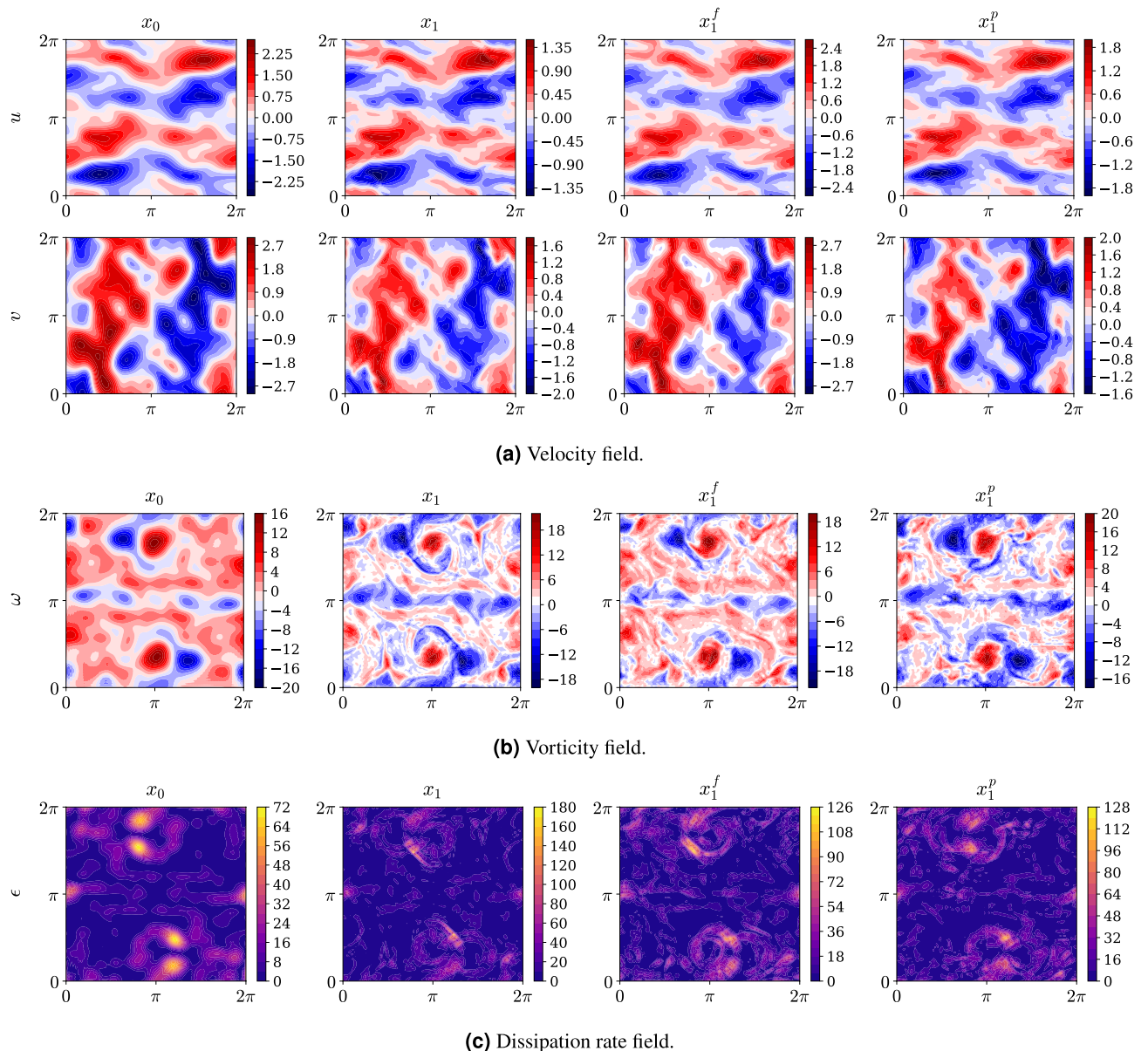


Fig. 4. Representative snapshot showing: (a) the velocity field, (b) the vorticity field, and (c) the dissipation rate field. In all panels, from left to right, the images display the low-resolution base field, the high-resolution target field, the SI_{full} super-resolution, and the SI_{patch} super-resolution.

the target field fairly well. Moreover, close inspection shows that the fine-scale structures seen in \mathbf{x}_1 are better matched by the SI super-resolutions, than the cubically upscaled field \mathbf{x}_0 .

The distinction becomes more apparent when examining the vorticity field, $\omega = \nabla \times \mathbf{u}$ (4b). While \mathbf{x}_0 exhibits smooth, low-detail contours, both \mathbf{x}_1 and the super-resolved fields produced by the SI models display finer-scale structures. Notably, the patch-based model avoids introducing sharp discontinuities at patch boundaries (see the patch mask in Fig. 1), indicating that the cond-generator in SI_{patch} produces super-resolved patches that are consistent with those from the free-generator. Such consistency is particularly important at patch boundaries, where discontinuities in spatial gradients might otherwise arise. Fortunately, the model maintains coherent transitions across patches.

The dissipation rate field (Fig. 4c), which also depends on spatial gradients, likewise shows no discontinuities at patch boundaries. The dissipation rate is evaluated at each spatial point by computing

$$\epsilon = 2\mu s_{ij}s_{ij}, \quad s_{ij} = \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad i, j \in \{1, 2\}. \quad (16)$$

Since we solve the non-dimensionalized Navier–Stokes equations, we simply set $\mu = 1/2$ to ease the computation of ϵ . Consistent with the behavior seen in the velocity and vorticity fields, the SI super-resolved dissipation rate fields show qualitative features that more closely match \mathbf{x}_1 than \mathbf{x}_0 .

Statistical performance

We have seen that the models generate reasonable super-resolutions for a representative snapshot. We now evaluate the statistical performance over the full test set.

Figure 5a and Fig. 5b display respectively the radially averaged spectra of energy, E , and enstrophy, Z , of the base (\mathbf{x}_0) and target (\mathbf{x}_1) sets, and compares them to the corresponding spectra of the SI_{full} and SI_{patch} super-resolved fields. A close alignment between the model and target spectra is observed, particularly at low to intermediate wavenumbers, highlighting a marked improvement compared to the base spectrum. At high wavenumbers the energy spectra diverge, however, the associated energy at these scales is minimal relative to the system energy, and the impact on overall statistical measures is therefore considered negligible. Overall, both models recover the target spectrum of energy and enstrophy well. Related studies, that apply generative models for super-resolution, such as the work by¹⁸, which uses DMs for full-field reconstruction, observe similarly shaped spectra for their Kolmogorov flow, and also report a divergence at high wavenumbers.

In Fig. 5c, the flatness of vorticity increments, F_ω , is shown as a function of the physical separation $\ell \in (0, \pi]$. The flatness is evaluated as

$$F_\omega(\ell) = \frac{\langle (\delta_\ell \omega)^4 \rangle}{\langle (\delta_\ell \omega)^2 \rangle^2}, \quad \delta_\ell \omega = \frac{1}{2}[(\omega(x + \ell, y) + \omega(x, y + \ell) - 2\omega(x, y))], \quad (17)$$

where $\langle \cdot \rangle$ denotes the spatial average. For the target field, the flatness profile exhibits a clear scale dependence. At small separations ℓ , the flatness takes values of ≈ 5 , indicating non-Gaussian statistics and intermittency associated with sharp vorticity gradients and coherent structures. As ℓ increases, the flatness decreases and approaches values ≈ 4 , reflecting partial Gaussianization due to spatial averaging. However, the persistence of flatness values above the Gaussian value of 3 at large scales indicates that large-scale vorticity fluctuations remain correlated and influenced by coherent flow structures. While deviations between the model and target flatness profiles are observed, the target statistic is nevertheless reasonably well recovered from the base field, which exhibits larger discrepancies that can be directly attributed to the filtering, subsampling, and interpolation procedure described in section Preliminaries.

The probability density functions shown in Fig. 6 are estimated using a Gaussian kernel density estimator^{35,36}. They describe the distributions of the kinetic energy, vorticity, and the dissipation rate of the base, the target, SI_{full} , and SI_{patch} fields. For every realization in the test set, each quantity is computed at all grid points, and the density functions are estimated over the entire test set. The kinetic energy is computed as

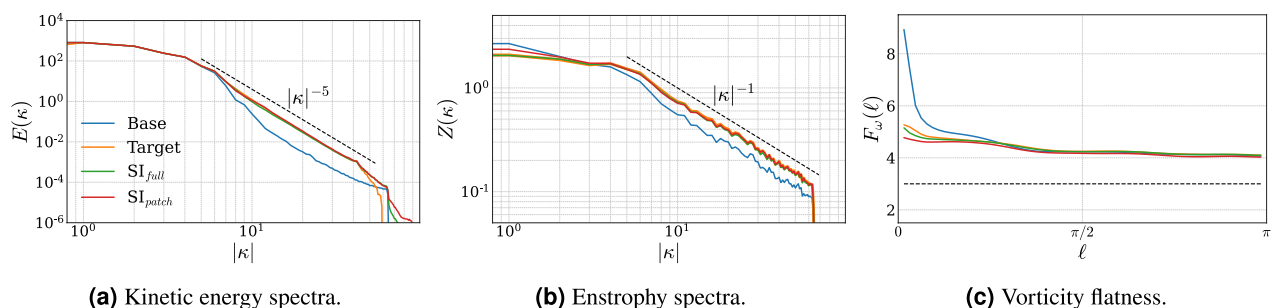


Fig. 5. Radially averaged spectra of (a) kinetic energy and (b) enstrophy. Each spectrum is computed as the average over the test set, and the base and target spectra are compared to the corresponding spectra of the SI_{full} and SI_{patch} super-resolved fields. Figure (c) similarly shows the flatness of vorticity increments.

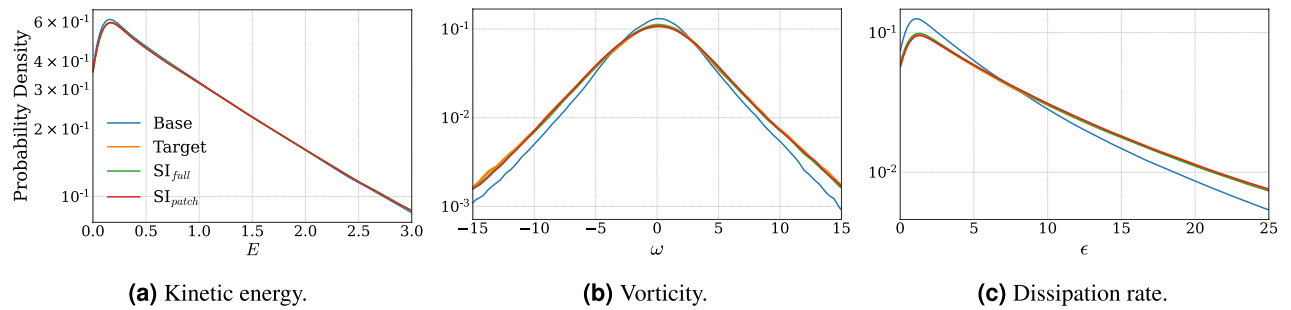


Fig. 6. Probability density functions of the (a) kinetic energy (b) vorticity and (c) dissipation rate, computed pointwise over the entire test set, and estimated using a Gaussian kernel density estimator. The plots compare the densities of SI_{full} and SI_{patch} with those of the base, x_0 , and the target, x_1 .

| | KL divergence ↓ | | | Wasserstein-1 distance ↓ | | |
|--------------|-----------------|---------------|---------------|--------------------------|---------------|---------------|
| | E | ω | ϵ | E | ω | ϵ |
| x_0 | 0.0121 | 0.0998 | 0.1470 | 1.1046 | 3.1063 | 9.8719 |
| DM_{full} | 0.0011 | 0.0068 | 0.0063 | 0.2057 | 0.7321 | 1.6198 |
| DM_{patch} | 0.0006 | 0.0182 | 0.0153 | 0.1880 | 1.2716 | 3.1738 |
| FM_{full} | 0.0015 | 0.0073 | 0.0062 | 0.4024 | 0.7250 | 1.7037 |
| FM_{patch} | 0.0009 | 0.0173 | 0.0140 | 0.1951 | 1.2142 | 2.9304 |
| SI_{full} | 0.0003 | 0.0011 | 0.0006 | 0.0528 | 0.1875 | <i>0.6181</i> |
| SI_{patch} | 0.0007 | <i>0.0049</i> | <i>0.0011</i> | <i>0.0803</i> | <i>0.4779</i> | 0.5889 |

Table 1. Comparison of Kullback–Leibler divergence and Wasserstein-1 distance between the target distribution and the distributions of x_0 , DM_{full} , DM_{patch} , FM_{full} , FM_{patch} , SI_{full} and SI_{patch} , evaluated for the probability densities of kinetic energy, vorticity, and dissipation rate. For easier readability, the distances computed for E have been scaled by a factor 10^2 , whereas those computed for ω and ϵ have been scaled by a factor 10. A downward-pointing arrow signifies that lower values are better. The best and second-best results are highlighted in boldface and italics, respectively.

$$E = u^2 + v^2, \quad (18)$$

Each quantity captures a distinct aspect of the flow. As shown in Fig. 6a, both models accurately recover the probability density function of the kinetic energy. This outcome is expected, since the models are trained to super-resolve the velocity fields, which are directly related to the kinetic energy. The densities of the vorticity (Fig. 6b) and dissipation rate (Fig. 6c) fields show that these quantities are also well recovered, although SI_{patch} exhibits a slight deviation from the target in both cases. The authors note in particular the recovery of the dissipation rate as a significant result, as ϵ is a key quantity commonly used to characterize turbulent flows³⁷. Moreover, it is a notoriously difficult parameter to experimentally measure³⁸, and if future measurements or low-resolution simulations can apply generative models to recover dissipation accurately, it would represent a meaningful advancement.

To quantify the deviations observed in Fig. 6, we utilize the Kullback–Leibler (KL) divergence³⁹ and the Wasserstein-1 distance⁴⁰. The KL divergence is defined as

$$D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx, \quad (19)$$

where p denotes the reference distribution, and q is the distribution being compared or approximated. The Wasserstein-1 distance is defined as

$$W_1(p, q) = \inf_{\pi \in \Gamma(p, q)} \mathbb{E}_{(x, y) \sim \pi} [|x - y|] \quad (20)$$

where $\Gamma(p, q)$ is the set of all joint distributions with marginals p and q .

Table 1 reports the KL divergence and W_1 distance for the densities of E , ω and ϵ . In each case, the SI models show an evident improvement over the base, with accuracy gains of approximately one to two orders of magnitude. SI_{full} is observed to outperform SI_{patch} across all but one of the evaluated metrics, for which the

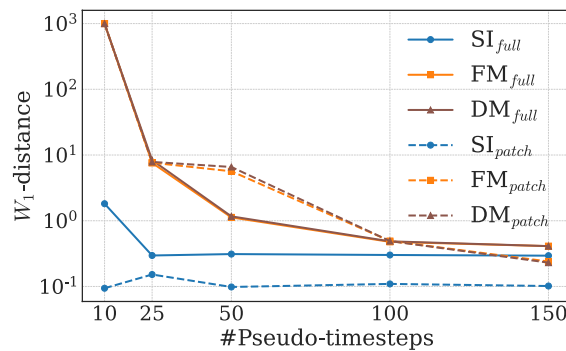


Fig. 7. Convergence of the Wasserstein-1 distance of the density of $\langle \epsilon \rangle$ as a function of the number of pseudo-timesteps used to infer super-resolutions. The SI-models are observed to converge for a lower number of pseudo-timesteps compared to their contesting counterparts. For the FM- and DM-models, the W_1 -value 10^3 is used as a placeholder, as the models failed to produce meaningful solutions when limited to only 10 timesteps during the inference stage.

two models show comparable performance. This finding aligns with the intuition that the model with access to full-domain information has an advantage over the one that receives only partial information. However, SI_{patch} performs largely on par with the full field model for reconstructing the considered densities, demonstrating SI_{patch} as a scalable alternative to the full-field method.

The table also compares the performance of the SI-models to equivalent flow-matching (FM) and diffusion (DM) models (see [Appendix A](#) for details). For the current configuration of using 100 pseudo-timesteps to produce super-resolutions within each model, the SI-models are observed to outperform the contesting methods in the considered metrics. This suggests that stochastic interpolants are indeed the better option for super-resolving 2D Kolmogorov turbulent flows. The conclusion is further supported as we consider the convergence of the W_1 distance for the density of $\langle \epsilon \rangle$ in [Fig. 7](#). Here $\langle \cdot \rangle$ denotes the spatial average, and $\langle \epsilon \rangle$ is computed for each snapshot in the test set, after which the corresponding probability density functions are estimated as before, over the whole test set, with the W_1 distance denoting the distance to the target density. Convergence is displayed as a function of the number of pseudo-timesteps used to infer super-resolutions. It is evident that the SI-models require significantly fewer timesteps to reasonably reproduce the flow statistic. Thus the SI framework is favored as the inference time decreases proportionally to the number of pseudo-timesteps. The primary reason for this is the fact that the SI model initiates the reconstruction from the low-resolution state, while the DM and FM start from a Gaussian sample. As a result, the SI base sample is already close to the target field, making the required transformation simpler and allowing the model to converge with fewer SDE steps. Furthermore, it is worth noting that the FM generates via an ODE while the SI generates via an SDE. Extensive comparisons between SDE- and ODE-based generation are performed in [Ma et al.⁴¹](#) showing that SDE-based sampling generally yields better results.

Extending the framework

The statistics presented in the previous section demonstrate that stochastic interpolants provide a viable approach for super-resolving the 2D Kolmogorov flow. While these results are promising, the framework still needs to be evaluated in three-dimensional turbulence and across a broad range of flow configurations, such as wall-bounded turbulence, to establish its applicability in more practical settings. Extending the framework to three-dimensional turbulence is non-trivial, and given its computational demands, the full-field approach is unlikely to be practical in this setting. For this reason, we focus on how the patch-wise strategy may be generalized to 3D flows.

The efficacy of the SI method for turbulent flows depends on the availability of suitable training data. Such data may be obtained either through numerical simulation or experimental investigations. Sample datasets may be found e.g. in Johns Hopkins Turbulence Database⁴². While training the full-field method on three-dimensional snapshots is computationally infeasible, particularly when large batch sizes are required, the patch-wise approach mitigates this limitation by operating on localized chunks of each snapshot. Once suitable data have been obtained, the patch model can be trained. The authors propose two possible strategies: 1. training a model tailored to a specific flow type, such as homogeneous isotropic turbulence, where, for instance, the Reynolds number can be used as a conditional input to the neural network for broader applicability, or 2. a more general model may be developed by training on patches drawn from multiple flow configurations.

With regard to network architecture, retaining a UNet-type model in three dimensions would require replacing all two-dimensional convolutional layers with their three-dimensional counterparts (i.e., $\text{Conv2D} \rightarrow \text{Conv3D}$ in PyTorch), along with potential adjustments to the network depth and overall parameterization. Alternatively, transformer-based models^{43,44} may offer improved scalability and flexibility in high-dimensional settings.

An additional consideration is the choice of patch size. In three-dimensional turbulence, the patch dimensions may need to be aligned with characteristic flow scales, such as the integral length scale, to ensure that all dynamically relevant features are adequately represented within each patch. Moreover, the interpolant

coefficients used in the present work are not guaranteed to be optimal for all flow configurations. We adopted the coefficients proposed by Chen et al.²⁰ without modification, but their generality across different turbulence regimes remains uncertain. Identifying optimal hyperparameters, such as those mentioned here, is an important topic for future study. Equally crucial is assessing whether the combination of low-resolution simulation and super-resolution offers a net computational advantage over direct high-resolution simulation.

Despite the remaining open questions, the results presented here provide motivation to further investigate the patch-wise approach in higher-dimensional flow settings.

Conclusion

We have introduced stochastic interpolants as a generative method for super-resolving fluid flows. Designed to enhance low-resolution simulations, LES, or experimental data, the approach can be applied either to reconstruct the full field in a single pass or to super-resolve smaller patches, enabling iterative recovery of the full domain or targeted regions of interest. For both configurations, the method effectively captures key flow statistics in the 2D case study, including the energy spectrum and the probability density functions of the kinetic energy, vorticity, and dissipation rate.

While the models developed meets the performance requirements within the studied setting, further investigation is required to evaluate the applicability to three-dimensional turbulence and generalizability to different flow regimes, for instance, how a model trained on one type of flow behaves when applied to another. Moreover, a rigorous evaluation of inference cost relative to the computational expense of high-resolution simulations is essential to justify the use of stochastic interpolants for fluid flow super-resolution.

Compared to other state-of-the-art generative methods, such as flow-matching and diffusion models, the proposed stochastic interpolant models demonstrate superior or at least comparable performance. This highlights their potential for turbulent flow super-resolution, and offers a promising perspective for future applications.

Data availability

The code for data generation and setting up the model stochastic interpolants, is available at https://github.com/martinschiodt/Turbulence_Stochastic_Interpolants. The repository also contains scripts for training the models, performing super-resolution, conducting analyses, and the implementations of the flow and diffusion models used for comparison.

Appendix A Implementation of flow and diffusion model

The flow-matching (FM) and diffusion (DM) models used for comparison in this work are developed according to the framework prescribed in⁴⁵. Here, a flow/diffusion model is defined by the ODE/SDE used for inference (super-resolution), i.e.

$$\mathbf{X}_0 \sim \mathcal{N}(0, 1), \quad d\mathbf{X}_\tau = b_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau)dt, \quad (\text{Flow model}) \quad (21)$$

$$\mathbf{X}_0 \sim \mathcal{N}(0, 1), \quad d\mathbf{X}_\tau = \tilde{b}_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau)dt + \sigma_\tau d\mathbf{W}_\tau. \quad (\text{Diffusion model}) \quad (22)$$

The architecture of the drift model, b_θ , in FM_{full} is identical to that used in SI_{full} and equivalently for the patch-models. To train b_θ we follow Algorithm 3 in⁴⁵, where the loss

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau \in U_{[0,1]}, \epsilon \in \mathcal{N}(0,1)} [\|b_\theta(\mathbf{x}_\tau, \mathbf{x}_0, \tau) - \mathbf{R}_\tau\|^2], \quad (23)$$

is minimized for batch samples of \mathbf{x}_0 and \mathbf{x}_1 under the same configurations as detailed in section [Methodology: Stochastic interpolants for turbulence super-resolution](#). Here

$$\mathbf{x}_\tau = \alpha_\tau \epsilon + \beta_\tau \mathbf{x}_1, \quad (24)$$

and for the noise-schedulers $\alpha_\tau = 1 - \tau^2$, $\beta_\tau = \tau$, the target is given by

$$\mathbf{R}_\tau = -2\tau\epsilon + \mathbf{x}_1. \quad (25)$$

After training, the flow model ODE can be solved to produce a FM super-resolution. For this purpose we use Heuns method.

For the diffusion model, we set

$$\tilde{b}_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau) = b_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau) + \frac{\sigma_\tau}{2} s_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau), \quad (26)$$

where b_θ is the drift from the FM-models, and the score network, s_θ , is evaluated directly from b_θ via

$$s_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau) = \frac{\beta_\tau b_\theta(\mathbf{X}_\tau, \mathbf{x}_0, \tau) - \dot{\beta}_\tau \mathbf{X}_\tau}{\alpha_\tau^2 \dot{\beta}_\tau - \beta_\tau \dot{\alpha}_\tau}. \quad (27)$$

The diffusion coefficient is set to $\sigma_\tau = 0.1(1 - \tau)$. With \tilde{b}_θ defined, the diffusion model SDE may be integrated forward in time to produce super-resolved velocity fields. For this purpose we use Heuns SDE integrator.

Received: 21 August 2025; Accepted: 28 December 2025

References

- Goc, K. A., Lehmkuhl, O., Park, G. I., Bose, S. T. & Moin, P. Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. *Flow* **1**, E14 (2021).
- Brunton, S. L., Noack, B. R. & Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020).
- Nista, L. et al. Influence of adversarial training on super-resolution turbulence reconstruction. *Phys. Rev. Fluids* **9**, 064601 (2024).
- Fukami, K., Fukagata, K. & Taira, K. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120 (2019).
- Fukami, K., Fukagata, K. & Taira, K. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* **909**, A9 (2021).
- Liu, B., Tang, J., Huang, H. & Lu, X.-Y. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* **32**, 025105 (2020).
- Zhou, Z., Li, B., Yang, X. & Yang, Z. A robust super-resolution reconstruction model of turbulent flow data based on deep learning. *Computers & Fluids* **239**, 105382 (2022).
- Stolz, S. & Adams, N. A. An approximate deconvolution procedure for large-eddy simulation. *Phys. Fluids* **11**, 1699–1701 (1999).
- Goodfellow, I. J. et al. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.*, **27**, 2672–2680 (2014).
- Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **33**, 6840–6851 (2020).
- Ledig, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690 (2017).
- Deng, Z., He, C., Liu, Y. & Kim, K. C. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys. Fluids*, **31**, 125111 (2019).
- Subramaniam, A., Wong, M. L., Borker, R. D., Nimmagadda, S. & Lele, S. K. Turbulence enrichment using physics-informed generative adversarial networks. [arXiv:2003.01907](https://arxiv.org/abs/2003.01907) (2020).
- Kim, H., Kim, J., Won, S. & Lee, C. Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.* **910**, A29 (2021).
- Li, T., Lanotte, A. S., Buzzicotti, M., Bonaccorso, F. & Biferale, L. Multi-scale reconstruction of turbulent rotating flows with generative diffusion models. *Atmosphere* **15**, 60 (2023).
- Kohl, G., Chen, L.-W. & Thurey, N. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. [arXiv:2309.01745](https://arxiv.org/abs/2309.01745) (2023).
- Lienen, M., Lüdke, D., Hansen-Palmus, J. & Günnemann, S. From zero to turbulence: Generative modeling for 3d flow simulation. [arXiv:2306.01776](https://arxiv.org/abs/2306.01776) (2023).
- Sardar, M., Skillen, A., Zimoń, M., Draycott, S. & Revell, A. Spectrally decomposed denoising diffusion probabilistic models for generative turbulence super-resolution. *Phys. Fluids*, **36**, 115179 (2024).
- Albergo, M. S. & Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. [arXiv:2209.15571](https://arxiv.org/abs/2209.15571) (2022).
- Chen, Y. et al. Probabilistic forecasting with stochastic interpolants and föllmer processes. [arXiv:2403.13724](https://arxiv.org/abs/2403.13724) (2024).
- Mücke, N. T. & Sanderse, B. Physics-aware generative models for turbulent fluid flows through energy-consistent stochastic interpolants. [arXiv:2504.05852](https://arxiv.org/abs/2504.05852) (2025).
- Chen, S., Jia, Y., Qu, Q., Sun, H. & Fessler, J. A. Flowdas: A stochastic interpolant-based framework for data assimilation. [arXiv:2501.16642](https://arxiv.org/abs/2501.16642) (2025).
- Lucas, D. & Kerswell, R. R. Recurrent flow analysis in spatiotemporally chaotic 2-dimensional kolmogorov flow. *Phys. Fluids*, **27**, 045106 (2015).
- Albergo, M. S., Boffi, N. M. & Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. [arXiv:2303.08797](https://arxiv.org/abs/2303.08797) (2023).
- Peyret, R. *Spectral methods for incompressible viscous flow*, vol. 148 (Springer, 2002).
- Kopriva, D. A. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers* (Springer Science & Business Media, 2009).
- Boyd, J. P. *Chebyshev and Fourier spectral methods* (Courier Corporation, 2001).
- Ronneberger, O., Fischer, P. & Brox, T. U-net Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention-MICCAI In 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, 234–241 (Springer, 2015).
- Liu, Z. et al. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986 (2022).
- Hendrycks, D. & Gimpel, K. Gaussian error linear units (gelus). [arXiv:1606.08415](https://arxiv.org/abs/1606.08415) (2016).
- Suda, T. Application of helmholtz-hodge decomposition to the study of certain vector fields. *J. Phys. A: Math. Theor.* **53**, 375703 (2020).
- Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017).
- Loshchilov, I. & Hutter, F. Sgdr. *Stochastic gradient descent with warm restarts* [arXiv:1608.03983](https://arxiv.org/abs/1608.03983) (2016).
- Thygesen, U. H. *Stochastic differential equations for science and engineering* (Chapman and Hall/CRC, 2023).
- Scott, D. W. *Multivariate density estimation: theory, practice, and visualization* (John Wiley & Sons, 2015).
- Silverman, B. W. *Density estimation for statistics and data analysis* (Routledge, 2018).
- Kolmogorov, A. N. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Proc. R. Soc. A: Math. Phys. Sci.* **434**, 9–13 (1991).
- Lai, C. C. & Socolofsky, S. A. Budgets of turbulent kinetic energy, reynolds stresses, and dissipation in a turbulent round jet discharged into a stagnant ambient. *Environ. Fluid Mech.* **19**, 349–377 (2019).
- Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951).
- Ramdas, A., García Trillos, N. & Cuturi, M. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy* **19**, 47 (2017).
- Ma, N. et al. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 23–40 (Springer, 2024).
- Li, Y. et al. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *J. Turbul.* **9**, 1–29 (2008).
- Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 6000–6010 (2017).
- Liu, Z. et al. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3202–3211 (2022).
- Holderrieth, P. & Erives, E. Introduction to flow matching and diffusion models. [arXiv:2506.02070](https://arxiv.org/abs/2506.02070) (2025).

Author contributions

M.S.: Conceptualization, Methodology, Software, Analysis, Discussion, Writing, Visualization. N.T.M.: Conceptualization, Methodology, Discussion, Writing, Visualization. C.M.V.: Supervision, Project administration,

Revisional writing.

Funding

C.M.V. acknowledges financial support from the European Research council: This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant agreement No 803419). C.M.V. and M.S. acknowledge financial support from the Poul Due Jensen Foundation: Financial support from the Poul Due Jensen Foundation (Grundfos Foundation) for this research is gratefully acknowledged. N.T.M acknowledges financial support from the National Growth Fund of the Netherlands administered by the Netherlands Organisation for Scientific Research (NWO) under the AINed XS grant NGF.1609.242.037.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2026