

# Generative AI for fast and accurate statistical computation of fluids

Roberto Molinaro <sup>1,3,\*</sup>, Samuel Lanthaler <sup>2,\*</sup>, Bogdan Raonić <sup>1,4,\*</sup>, Tobias Rohner <sup>1,\*</sup>, Victor Armegioiu <sup>1</sup>, Stephan Simonis <sup>5</sup>, Dana Grund <sup>1,6</sup>, Yannick Ramic <sup>1</sup>, Zhong Yi Wan <sup>7</sup>, Fei Sha <sup>7</sup>, Siddhartha Mishra <sup>1,4,†</sup>, Leonardo Zepeda-Núñez <sup>7,†</sup>

<sup>1</sup> Seminar for Applied Mathematics, D-MATH, ETH Zurich, Switzerland,

<sup>2</sup> University of Vienna, Vienna, Austria,

<sup>3</sup> Jua.ai, Zurich, Switzerland,

<sup>4</sup> ETH AI Center, Zurich, Switzerland,

<sup>5</sup> Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany,

<sup>6</sup> D-USYS, ETH Zurich, Switzerland,

<sup>7</sup> Google Research, Mountain View, CA 94043, USA,

\* Equal contribution, † Co-corresponding authors.

**ABSTRACT.** We present a generative AI algorithm for addressing the pressing task of fast, accurate, and robust statistical computation of three-dimensional turbulent fluid flows. Our algorithm, termed as *GenCFD*, is based on an end-to-end conditional score-based diffusion model. Through extensive numerical experimentation with a set of challenging fluid flows, we demonstrate that GenCFD provides an accurate approximation of relevant statistical quantities of interest while also efficiently generating high-quality realistic samples of turbulent fluid flows and ensuring excellent spectral resolution. In contrast, ensembles of deterministic ML algorithms, trained to minimize mean square errors, regress to the mean flow. We present rigorous theoretical results uncovering the surprising mechanisms through which diffusion models accurately generate fluid flows. These mechanisms are illustrated with solvable toy models that exhibit the mathematically relevant features of turbulent fluid flows while being amenable to explicit analytical formulae. Our codes are publicly available at <https://github.com/camlab-ethz/GenCFD>.

## 1 Introduction.

Fluids are ubiquitous in nature and in engineering [19], encompassing phenomena as diverse as atmospheric and oceanic flows in climate modeling, waves and tsunamis in hydrology, flows of gases in astrophysics, sub-surface flows in mineral reservoirs and in the Earth’s mantle, blood flow in the human body to flows past vehicles such as cars and airplanes. As such, understanding, predicting, and controlling fluids is indispensable for scientific discovery and engineering design. However, the study of fluid flows is very challenging as they span a vast range of spatio-temporal scales and encompass a rich phenomenology of states. In particular, flows at high *Reynolds*

*numbers* ( $Re$ ) can evolve chaotically into states containing energetic eddies that span a very large range of scales [19]. This exhibition of multi-scale complexity and sensitive dependence on inputs is often attributed to *turbulence* [19], considered by Richard Feynman as the *most important unsolved problem of classical physics* [12].

Fluids are mathematically modeled by (variants of) the famous *Navier–Stokes* equations. In the absence of analytical solution formulae for these nonlinear systems of partial differential equations (PDEs), *simulating* fluids in silico with numerical algorithms such as finite difference [4], finite element [25], finite volume [40] and spectral methods [77] etc., have emerged as the dominant paradigm for predicting fluid flows. Although highly successful in many contexts, this field of computational fluid dynamics (CFD) suffers from an intrinsic *curse of computational complexity* as the underlying computational cost scales as  $Re^3$ , where  $Re$  is large for many flows of interest [55]. Consequently, direct numerical simulations (*DNS*) of fluid flows are *prohibitive* in practice, and a variety of turbulence models [55] have been proposed as alternatives to DNS.

However, at best, these models represent incomplete approximations with ad hoc closures, often including undetermined and uncertain parameters. Even so, for several downstream applications like atmospheric flows, even high-quality models, such as large eddy simulations (*LES*) [68] entail a heavy computational burden.

Moreover, given the very high sensitivity of fluid flows to small perturbations in inputs such as initial and boundary conditions (see Fig. 1(A)), deterministic simulations, whether DNS or LES, have limited predictive power [13, 5]. Fortunately, the computation of statistical quantities of interest is much more stable to perturbations [13, 5] (see also Fig. 1 (A)), making *statistical computation*, often referred to as forward *uncertainty quantification* (UQ), imperative in computational fluid dynamics as well as the preferred paradigm for design and optimization in engineering applications [5].

Alas, statistical computation of fluid flows is extremely challenging: to compute the desired statistical quantities, one typically requires an *ensemble* of inputs sampled from an underlying probability distribution, where each member of this ensemble is numerically evolved with an already computationally expensive DNS or LES, resulting in an ensemble of trajectories from which the target statistics are estimated. Although the computational cost grows linearly in the number of ensemble members, due to the slow (square-root) convergence of random sampling, one needs a large ensemble for accurate statistical computation [5, 16, 64], making the overall pipeline virtually intractable. This renders the design of algorithms for the fast and accurate statistical computation of fluid flows a grand challenge of modern computational science [5].

Given their success at providing fast and accurate surrogates for solutions of many PDEs, machine learning (ML) algorithms, such as PINNs [60, 61], neural operators [43, 42, 63], graph neural networks [54] and transformers [24], are promising candidates for fast statistical computation of fluid flows, by replacing the expensive numerical solver with these much faster ML-based surrogates. Unfortunately, these deterministic neural networks, which are trained to minimize the mean square prediction errors, are observed to fail at accurate statistical computation of complex multiscale physical systems [58, 6]. As demonstrated in Fig. 1 (D), the ensembles



predicted by these algorithms collapse to the mean instead of learning the underlying probability distribution of the fluid flow.

Given this context, our main goal is to address the outstanding challenge of designing a fast and accurate framework for the statistical computation of fluid flows. To this end, we tailor the so-called *score-based diffusion models*, see Fig. 1 (B, C), which are particular examples of generative AI and were developed for and are widely used in image and video generation [62, 69, 65, 29, 7, 1], to the disparate task of computing the statistics of fluid flows. As Fig. 1 (D) already shows, we demonstrate, through extensive numerical experiments, that our method, termed as GenCFD, yields accurate approximations of statistical quantities of interest, while also producing very high-quality realizations of a variety of challenging fluid flows (see 6). At the same time, GenCFD is several orders of magnitude faster than CFD solvers (see SI Table 15). To be more specific, GenCFD takes approximately 1 second to generate a complex three-dimensional turbulent fluid flow. We also provide rigorous mathematical arguments and analytically tractable toy models to *explain* the success of GenCFD in the statistical computation of complex physical systems, *uncovering* the precise mechanisms through which a diffusion model, such as ours, can provide accurate statistical computation for complex dynamical systems such as turbulent fluid flows. Thus, with GenCFD, we present a generative AI algorithm which can transform the simulation of fluid flows and has the potential to significantly impact a large number of downstream tasks in physics, climate science, and engineering.

## 2 Problem formulation and setup

Fluid flows are modeled by (variants of) the Navier–Stokes equations (defined in **SI** Sec. 6), which can be written as an abstract nonlinear PDE of the form  $\mathcal{L}_{\bar{u}}[u] = 0$ , with a differential operator  $\mathcal{L}$ ,  $\bar{u} \in \mathcal{X}$  representing inputs to the PDE (such as initial and boundary conditions for the Navier–Stokes equations) and  $u \in \mathcal{Y}$  being the solution and  $\mathcal{X}, \mathcal{Y}$  suitable function spaces. The resulting *Solution Operator*  $\mathcal{S} : \mathcal{X} \mapsto \mathcal{Y}$  maps the inputs  $\bar{u}$  to the solution  $u$ . Given a distribution  $\mu \in \text{Prob}(\mathcal{X})$ , statistical computation (or forward UQ) entails the calculation of the so-called *push-forward measure*  $\mathcal{S}_\# \mu \in \text{Prob}(\mathcal{Y})$ , which describes how uncertainties in the inputs  $\bar{u}$  are transformed by the solution operator of a PDE [5]. This abstract problem is very challenging in view of the intrinsic infinite-dimensionality of the underlying function spaces. Hence, in **SI** Sec. 6, we derive how this problem of *statistical computation* for PDEs can be (approximately) recast in terms of computing a *conditional probability distribution* given by the (generalized) probability density  $p(u|\bar{u})$ , conditioned on inputs  $\bar{u} \sim \bar{p}(\bar{u})$  drawn from an input distribution with density  $\bar{p}$ , which is an approximation to  $\mu$ .

In Fig. 1 (A), we illustrate how this conditional probability distribution is approximated in current UQ algorithms for CFD [5, 16, 39, 64, 72]. In a first step, an ensemble of inputs (for instance, initial data) is drawn from the distribution  $\bar{p}$ . Each ensemble member is then evolved with a CFD solver that approximates the solution operator  $\mathcal{S}$ . Thereafter, the empirical measure of these evolved samples approximates the target distribution  $p(u|\bar{u})$  and statistical quantities such as mean and variance can be readily computed. However, this process is

prohibitively expensive as a large number of ensemble members need to be evolved with already computationally expensive CFD solvers.

ML algorithms for computing the target conditional distribution work by replacing the CFD solver by a neural network  $\Psi_\theta \approx \mathcal{S}$  in the afore-sketched UQ algorithm, where the parameters  $\theta$  are determined by minimizing the mismatch between  $\Psi_\theta$  and  $\mathcal{S}$  in the mean-square (or absolute) norm. However, as seen in Fig. 1 (D) and discussed previously in [58, 6], these ML ensembles are observed to regress to the mean of the underlying distribution and are not able to generate the variance of that distribution. These observations underscore the urgent need for the design of alternative AI approaches for the accurate statistical computation of fluids.

To this end, we propose a paradigm shift: instead of developing fast surrogates for ensemble based computations, we seek to learn the underlying distribution *directly*. In particular, we propose a *conditional diffusion* model to *generate* the probability distribution  $p(u|\bar{u})$ . As illustrated in Fig. 1 (B), a conditional diffusion model [78, 3] approximates the target conditional probability distribution with a two-step process. In the first *forward* step, given a pair of samples,  $\bar{u} \sim \bar{p}$  and  $u \sim p(u|\bar{u})$ , *noise* is iteratively added to  $u_0 = u$  in order to transform it to a sample  $u_K$  that follows a known distribution such as an *isotropic Gaussian* of the form  $p_K(u_K|\bar{u}) \sim \mathcal{N}(u_K; 0, \sigma_K^2 I)$ , with zero mean and a prescribed variance  $\sigma_K^2 I$ . In general, this iterative process is implemented by solving a suitable stochastic differential equation (SDE, see **SI** Sec. 6 for details) forward in time [31]. Next, the key step is the so-called *reverse step* (Fig. 1 (B)) where given  $\bar{u}$  and a *noisy sample*  $u_K \sim p_K(u_K|\bar{u})$ , the *reverse SDE*

$$du_\tau = -2\dot{\sigma}_\tau \sigma_\tau \nabla_{u_\tau} \log p_\tau(u_\tau|\bar{u}) d\tau + \sqrt{2\dot{\sigma}_\tau \sigma_\tau} d\widehat{W}_\tau \quad (1)$$

is solved backward in *pseudo-time*  $\tau \in [0, K]$  with a terminal distribution  $p_K$  and  $\widehat{W}_\tau$  is the Brownian motion in backward time. While postponing detailed notation for this SDE to **SI** Sec. 6, we would like to emphasize that solving it from  $\tau = K$  to  $\tau = 0$  recovers the target conditional distribution as  $p_0(u|\bar{u}) = p(u|\bar{u})$  [31].

However, solving the SDE (1) requires the explicit form of the so-called *score-function*  $\log p_\tau(u_\tau|\bar{u})$  of the underlying distribution at each  $\tau \in [0, K]$ , which is not available. Instead, we follow score-based diffusion models [31, 3] and approximate the score-function in terms of the infamous Tweedie's formula by

$$\nabla_u \log p_\tau(u_\tau|\bar{u}) \approx \frac{D_\theta(u_\tau(\bar{u}), \bar{u}, \sigma_\tau) - u_\tau}{\sigma_\tau^2}. \quad (2)$$

Here, the so-called *denoiser*  $D_\theta$ , a neural network with trainable parameters  $\theta$ , takes the condition  $\bar{u} \sim \bar{p}$ , the *noisy sample*  $u_\tau(\bar{u})$  (drawn from a Gaussian  $\mathcal{N}(\cdot; u, \sigma_\tau^2 I)$ ) and the noise level  $\sigma_\tau$  as inputs in order to output the *clean* underlying sample. Hence, as illustrated in Fig. 1 (C), we need to *train* the denoiser  $D_\theta$  to remove noise from the *noisy sample*  $u_\tau(\bar{u}) = u + \eta$ ,  $\eta \sim \mathcal{N}(0, \sigma_\tau^2 I)$  and output the clean underlying sample  $u$ . This is achieved by training the denoiser to minimize the *denoiser training objective or diffusion loss*

$$\mathcal{J}(D_\theta) = \mathbb{E}_{\bar{u} \sim \bar{p}} \mathbb{E}_{u|\bar{u}} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma_\tau^2 I)} \|D_\theta(u + \eta; \bar{u}, \sigma_\tau) - u\|^2. \quad (3)$$

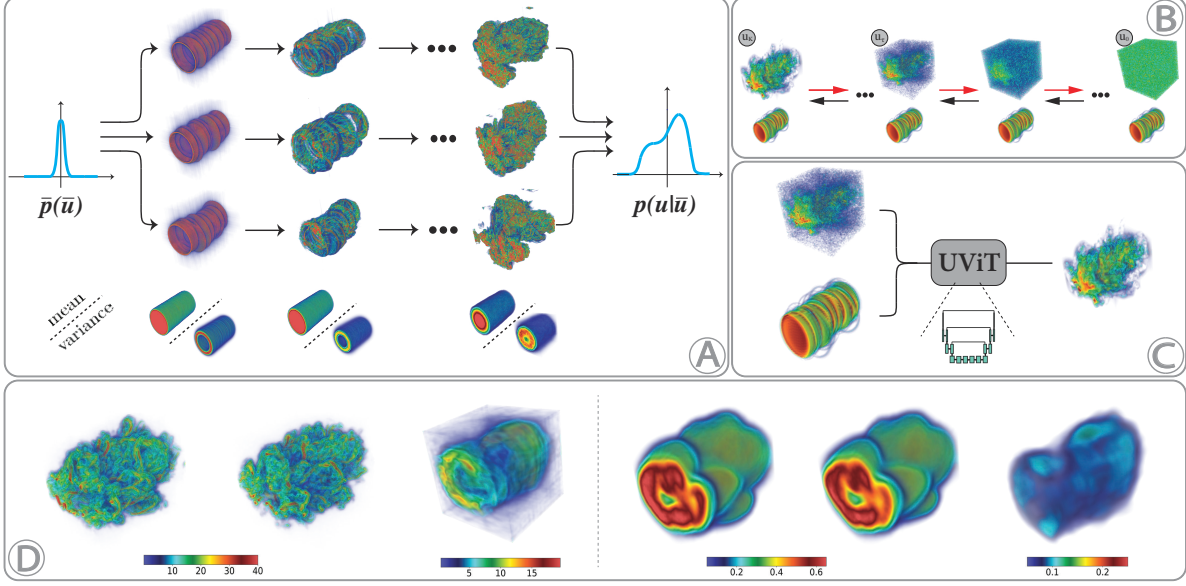
At inference, the reverse SDE (1), with its score-function replaced by the trained denoiser, is (numerically) integrated backward in time to generate samples from the target distribution  $p(u|\bar{u})$ , given the input condition  $\bar{u}$  and isotropic Gaussian noise  $u_K$ .

We chose a specific neural network architecture for the denoiser in our generative AI algorithm. As detailed in **SI** Sec. 6 and illustrated in Fig. 1 (C), it is a UViT [69] type neural operator specifically adapted for multiscale information processing.

Finally, it is essential to point out that several novel elements were incorporated into conditional score-based diffusion models in order to deal with the fact that our target distributions are push-forwards of the solution operators of time-dependent PDEs, rather than distributions over static data such as natural images. These include lead-time conditioning, all-to-all training [24] and special variance-capturing loss functions; for details, see **SI** Sec. 6.

We tested our proposed conditional score-based diffusion model, GenCFD, on a suite of five challenging fluid flows (see **SI** Sec. 6 for detailed description of datasets). To provide context to our results, we also tested ML baselines on the same suite of problems. To this end, we considered three state-of-the-art neural operators as baselines (defined in **SI** Sec. 6): the UViT model, which is also the architecture of the model underpinning GenCFD, the popular Fourier Neural Operator (FNO) [42] and a novel variant of it that adds *local convolutional* layers to the Fourier layers, which we term as C-FNO (see **SI** Sec. 6). All the baselines are neural networks  $\Psi_\theta$  that are trained to minimize the mean square error  $\mathbb{E}_{\bar{u} \sim \bar{p}} \|\Psi_\theta(\bar{u}) - \mathcal{S}(\bar{u})\|^2$ . Statistical computation is performed by generating ensembles of the form  $(\Psi_\theta)_{\#} \bar{p}$ , see also **SI** Sec. 6.

All the models are trained with data drawn from specific distributions as outlined in **SI** Sec. 6. However, at test time, we focus on input distributions  $\bar{p} \approx \delta_{\bar{u}^*}$ , for some  $\bar{u}^* \in \mathcal{X}$  (see Fig. 1 (A) for an illustration). We do this as i) it allows us to evaluate the ability of the models to generalize *out of distribution* and ii) it is well known that, even if the initial distribution is (approximately) a Dirac measure, the intrinsic chaotic evolution of turbulent fluids *spreads out the measure* [13, 39, 16] (see also Fig. 1 (A) and **SI** Sec. 6).



**Figure 1: Visual summary of this article.** (A): Our goal here is the statistical computation of a fluid flow, i.e., computing the push-forward of the distribution  $\bar{p}(\bar{u})$  on the inputs (initial and boundary conditions) with respect to the solution operator  $\mathcal{S}$  of a PDE to provide the target distribution  $p(u|\bar{u})$ , with  $u = \mathcal{S}(\bar{u})$  (at any desired time) as its output. In UQ for CFD, one draws samples  $\bar{u} \sim \bar{p}$ , evolves them in time with a CFD solver and computes statistics such as mean and variance from samples  $u \sim p(\cdot|\bar{u})$ . ML algorithms simply replace the CFD solver with a neural network surrogate, trained to minimize the mean square error with respect to each  $u$ . On the other hand, our method GenCFD is based on (B): A conditional diffusion model, which at inference (black arrows) generates  $u \sim p(\cdot|\bar{u})$ , given input  $\bar{u} \sim \bar{p}$  and isotropic Gaussian noise  $u_K$ , by solving the reserve SDE (1) backward in time. During training of the diffusion model, noise is added iteratively (red arrows) to transform any  $u \sim p$  to a noisy sample and (C): The denoiser (UViT for GenCFD) is a neural network that is trained to output a clean sample of the solution  $u$ , given input  $\bar{u}$  and noise. The denoiser replaces the score function in the reverse SDE (1). (D). Results for the cylindrical shear flow dataset: individual Realizations (Left sub-panel) of the vorticity intensity and standard deviation (Right sub-panel) of the pointwise kinetic energy at  $T = 1$  for the ground truth (Left), GenCFD (Center) and an ML baseline C-FNO (Right). The ground truth is generated by a DNS with a spectral hyperviscosity method. Please note the different ranges of the colorbars in (D).

### 3 Experimental results

Fig. 2 summarizes our experimental observations on the *Taylor–Green vortex* for the incompressible Navier–Stokes equations (see **SI** Sec. 6), a prototypical benchmark for three-dimensional turbulent fluid flows [19] which is widely used for validating CFD solvers as well as turbulence models.

As seen from Fig. 2 (A, B, F), the flow is highly intricate with a wide range of small scales. Our task is to (approximate) the underlying distribution at future times, conditioned on a Taylor–Green initial datum. From Fig. 2 (A) where we plot the pointwise kinetic energy at time  $T = 2$ , we see that GenCFD is able to generate *high-quality realistic* samples of the underlying fluid flow. In fact, it is not possible to visually distinguish between the ground truth and GenCFD-generated samples. This high sample quality of GenCFD is further demonstrated in Fig. 2 (B) where we plot the (pointwise) intensity of the fluid vorticity, computed by taking the curl of the generated velocity fields. Again, it is not possible to visually distinguish between the quality of the GenCFD-generated vorticity and the ground truth. This is particularly impressive as the model has never been trained on vorticity profiles. Nonetheless, GenCFD is able to accurately approximate the *multivariate* structures of the velocity profiles so that the derivatives give rise to the accurate vorticity profiles.

On the other hand, the samples generated by all the baselines (see Fig. 2 (A, B) for C-FNO which is the strongest ML baseline on this dataset) are of poor quality and do not capture the small scales of the flow. In particular, high-intensity vortex tubes are completely smeared out. Moreover, GenCFD excels at approximating statistical quantities such as the mean (of the pointwise kinetic energy shown in Fig. 2 (C)), the variance (of the kinetic energy shown in Fig. 2 (D)) and even point PDFs (of the  $x$ -velocity component shown in Fig. 2 (E)). In particular, the variance of the flow is very hard for deterministic neural operators to approximate as the initial condition is (nearly) a Dirac measure. Yet, GenCFD provides an excellent approximation of this statistical quantity, especially when compared to the baselines. Similarly, the point PDFs are well spread out by this time ( $T = 2$ ) and GenCFD still approximates them very well. On the other hand, the baselines completely fail at capturing the variance and the generated PDF collapses to a single point. These observations are reinforced by the quantitative results presented in Fig. 2 (G) and **SI** Table 6. In Fig. 2 (G), we present the  $L^1$ -errors in the mean and the standard deviation of the  $x$ -component of the velocity, as well as the spatially integrated 1-Wasserstein distance between the target distribution and the conditional distribution generated by GenCFD (see **SI** Sec. 6 for definitions). The quantitative results show how well GenCFD approximates the mean, the variance, and the underlying probability distribution. In particular, it is one order of magnitude more accurate at capturing the variance and approximating ground-truth in terms of Wasserstein distance when compared to the baselines.

These qualitative and quantitative results amply demonstrate the ability of GenCFD to accurately capture the statistics of the Taylor–Green vortex.

Capturing the correct spectral behavior is fundamental in the study of turbulent fluids as energy

cascades down to the smaller scales via a power law decay of the spectrum [19]. In Fig. 2 (F) we plot the energy spectrum for the ground truth, GenCFD, and the best-performing baseline (C-FNO), from which we observe that GenCFD approximates the energy spectrum (and its power law decay) accurately, all the way down to the smallest resolved scales. On the other hand, the spectrum generated by C-FNO and other baselines is highly inaccurate and decays way too fast (exponentially) to represent a turbulent flow. This is consistent with the observation of the lack of small-scale structures in the baselines presented here as well as in the literature [58, 6].

GenCFD’s superior performance in generating accurate turbulent fluid flows extends to the other four flows considered here. For example, for the three-dimensional *cylindrical shear flow* [64], we observe exactly the same qualitative and quantitative results obtained for the Taylor–Green vortex, as shown in Fig. 1 (D) where we present a sample of the vorticity intensity and the computed variance at time  $T = 1$ , for the ground truth, GenCFD, and C-FNO for a given initial conditions (see also **SI** Figures 2-6 and Table 7 for further results for this benchmark).

What does it cost for GenCFD to generate these samples and statistics of fluid flows ? In terms of compute, we see from **SI** Table 15 that it takes approximately 0.45 seconds for GenCFD to generate a single sample of Taylor-Green vortex or Shear flow on a GPU. In terms of *sample complexity*, we recall that the test task is out-of-distribution and GenCFD has only seen *one set of input-output pairs* per input condition during training. Nevertheless, it is able to generate *a large diversity in samples*, for the same input condition, as shown in **SI** Figures 2-3 and 24-26 for the Shear flow benchmark (see also Fig. 2 (A, B) for the Taylor–Green vortex), showcasing the very low sample-complexity of GenCFD in generating high-quality fluid flows.

In Figure 3, we present a representative glimpse of the experimental results for the other three very challenging datasets exhibiting different physics, boundary conditions and with different downstream applications. We start with a three-dimensional *nozzle flow* at a Reynolds number of up to  $Re = 20000$  for the Navier–Stokes equations (see **SI** Sec. 6). We consider this flow as a prototypical example of *turbulent jet flows* that are widely studied in engineering [55]. The simulated flow field differs from the other datasets in i) being both wall-bounded as well as having a freestream leading to ii) non-trivial wall boundary conditions in place of the previously considered periodic boundary conditions and iii) *the entire flow needs to be generated from a single scalar input, i.e., the injection velocity*, which is in the form of a boundary condition rather than the initial condition as in the Taylor–Green and the Cylindrical Shear Flow examples in Fig. 2 and Fig. 1 respectively. As visualized with a sample of (pointwise) vorticity intensity (see Fig. 3 (A)), the flow, with the ground truth generated by an LES (see **SI** Sec. 6), consists of an energetic jet emanating from the inlet and evolving in a turbulent manner to an intricate collection of multi-scale whirls and eddies further downstream. From Fig. 3 (A) (see also **SI** Figs. 10-11), we observe that GenCFD is able to generate samples of this complex flow realistically while the best-performing baseline (UViT in this case) fails completely in generating the small-scale features in the vorticity and collapses onto a (laminar) jet in the middle of the flow. Similarly, statistics of this complex flow are accurately approximated by GenCFD while the baselines fail to account for the variance (Fig. 3 (A)). Further qualitative and quantitative results in **SI** Figs. 10-12 and 19, and Table 9, show that GenCFD can accurately generate this



complex flow from just a single scalar input vastly outperforming the baselines that are, at best, only able to generate the mean behavior.

In Fig. 3 (B), we consider the three-dimensional *cloud-shock interaction problem*, which is a well-established benchmark for *compressible fluid flows* [40] (see SI Sec. 6). As visualized with the density profile in Fig. 3 (B), an incoming supersonic shock wave hits a high-density cloud and leads to the excitation of shock waves while creating a zone of turbulent mixing in their wake. Even though the underlying equations (compressible Euler vs. incompressible Navier–Stokes, see SI Sec. 6) and the flow dynamics (presence of discontinuous shock waves) are very different from the previously considered examples, GenCFD is able to generate realistic flow samples, while also yielding highly accurate approximations of statistical quantities of interest (Fig. 3 (B) and SI Figs. 7-9, 19 and Table 8). On the other hand, baselines such as C-FNO fail to capture the turbulent mixing zone, although the strong shock wave is accurately computed.

Finally, in Fig. 3 (C), we present results for the *dry convective planetary boundary layer*, a well-known benchmark in the atmospheric sciences [76], heavily used for understanding the statistics of boundary layer flows and validating and calibrating turbulence models in meteorology. This atmospheric flow corresponds to the dynamics of air under the effect of a surface heat flux (modeling radiative heating through a summer day) and a weak large-scale geostrophic wind which induces shear at the surface, leading to a complex combination of updraft and downdraft plumes driving (vertically) anisotropic turbulent motion (see Fig. 3 (C) for a visualization of the  $x$ -component of velocity). Not only are the underlying PDEs (anelastic flow equations, (see SI Sec. 6) different from the previous datasets, but the physics of this flow are far richer, due to the presence of heat transfer. Nevertheless, GenCFD generates high-quality samples of this flow and accurately approximates the variance, greatly outperforming the baselines (Fig. 3 (C)). We provide a detailed qualitative and quantitative analysis of this benchmark in the SI Figs. 14-17, 19 and Table 10, particularly for (horizontally averaged) statistics, which further showcase the excellent performance of GenCFD.

As mentioned earlier, the (approximately) Dirac test distribution for all the datasets is different from the underlying training distribution, highlighting the ability of GenCFD to generalize.

In fact, GenCFD is able to robustly generalize to unseen test distributions, either with no additional training (*zero shot*) or when fine-tuned with a few downstream samples (*few shot*), as shown in the SI (see SI Fig. 27 and Table 12)

Another avenue where GenCFD shines is its ability to generate the complex temporal dynamics of turbulent fluid flows, including transitions from laminar to turbulent regimes as shown in the SI. in Figs. 20-23 and Table 11. In those figures and tables, we present samples and statistics for the Taylor–Green vortex at time  $T = 0.8$ , when the flow is still laminar and not yet turbulent, showcasing that GenCFD provides accurate samples and approximations to time-varying statistical quantities. We attribute this ability to our *lead time conditioning* and an *all-to-all* training procedure, which leverages the semi-group property of the underlying solution operator (see SI Sec. 6).

Last but not least, the main premise for the design of surrogates for CFD solvers is their



computational speed. To this end, in **SI** Table 15, we compare the computational cost of generating the ground truth with state-of-the-art CFD solvers (on GPUs and CPUs) and the sample generation time of GenCFD at inference to find that GenCFD can provide anywhere between *one to five orders of magnitude speedup* over traditional solvers in generating fluid flows (see **SI** Table 15), depending on the dataset, underlying solver and hardware (GPU vs. CPU) used for CFD. In particular, GenCFD can generate a fluid flow in 1 to 4 seconds while it takes a standard CFD solver anywhere between minutes (on GPUs) to hours (on CPUs).

This massive speedup, coupled with its statistical accuracy, renders GenCFD particularly attractive for widespread downstream applications.

## 4 Theory

Why does GenCFD work so well in generating realistic fluid flows and accurately approximating their statistical and spectral behavior when baselines completely fail to do so? To address this question, we present theoretical arguments, based on rigorous mathematical analysis in the SI, with a heuristic summary here. Following **SI** Sec. 6 and for the setting considered here, the goal of statistical computation for a PDE with the (approximate) solution operator  $\mathcal{S}$  and any initial datum  $\bar{u}^* \in \mathcal{X}$ , is to compute the conditional distribution corresponding to the *Law* of the random variable  $\text{Law}_{\delta\bar{u}}\mathcal{S}(\bar{u}^* + \delta\bar{u})$ , over randomly chosen *very small* perturbations  $\delta\bar{u}$ , with  $\|\delta\bar{u}\|_{\mathcal{X}} \approx 0$ .

Given the sensitive dependence of turbulent fluid flows to inputs, it is reasonable to hypothesize that there exists a *sensitivity scale*  $\bar{\epsilon} \ll 1$ , such that for perturbations  $\|\delta\bar{u}\|_{\mathcal{X}} \sim \bar{\epsilon}$ , the outputs are well-separated, i.e.,  $\|\mathcal{S}(\bar{u}^* + \delta\bar{u}) - \mathcal{S}(\bar{u}^*)\|_{\mathcal{Y}} \gg 1$ .

On the other hand, we argue in the **SI** Sec. 7 that several factors including the empirically observed and theoretically argued fact that trained neural networks *operate at the edge of chaos* [11], the well-known *spectral bias of neural networks* [59] and the need for bounded gradients for training neural networks with gradient descent imply the *insensitivity of neural networks to small-scale perturbations*, i.e.,  $\Psi_{\theta}(\bar{u}^* + \delta\bar{u}) \approx \Psi_{\theta}(\bar{u}^*)$ , whenever  $\|\delta\bar{u}\|_{\mathcal{X}} < \bar{\epsilon}$ . Consequently, training such a neural network  $\Psi_{\theta}$  to learn the target conditional distribution using the *ensemble perturbation approach* amounts to minimizing

$$\begin{aligned} \mathbb{E}_{\delta\bar{u}} \|\Psi_{\theta}(\bar{u}^* + \delta\bar{u}) - \mathcal{S}(\bar{u}^* + \delta\bar{u})\|^2 &\approx \mathbb{E}_{\delta\bar{u}} \|\Psi_{\theta}(\bar{u}^*) - \mathcal{S}(\bar{u}^* + \delta\bar{u})\|^2 \quad (\text{insensitivity hypothesis}) \\ &= \|\Psi_{\theta}(\bar{u}^*) - \mathbb{E}_{\delta\bar{u}}\mathcal{S}(\bar{u}^* + \delta\bar{u})\|^2 + \text{Var}_{\delta\bar{u}}[\mathcal{S}(\bar{u}^* + \delta\bar{u})]. \quad (\text{bias-variance decomposition}) \end{aligned}$$

Note that we cannot replace  $\mathcal{S}(\bar{u}^* + \delta\bar{u})$  with  $\mathcal{S}(\bar{u}^*)$  above due to the sensitive dependence of  $\mathcal{S}$  to inputs. As the second term in the above sum is independent of  $\theta$ , the optimal neural network is given by  $\Psi_{\theta}(\bar{u}^*) = \mathbb{E}_{\delta\bar{u}}\mathcal{S}(\bar{u}^* + \delta\bar{u})$ , which is precisely the *mean* of the ensemble. This simple argument clearly explains the extensive empirical observation, both here and in the literature [6, 58], of why ensembles of neural networks trained to minimize least-square errors for learning turbulent fluids regress to the mean and fail to generate sufficient variance.

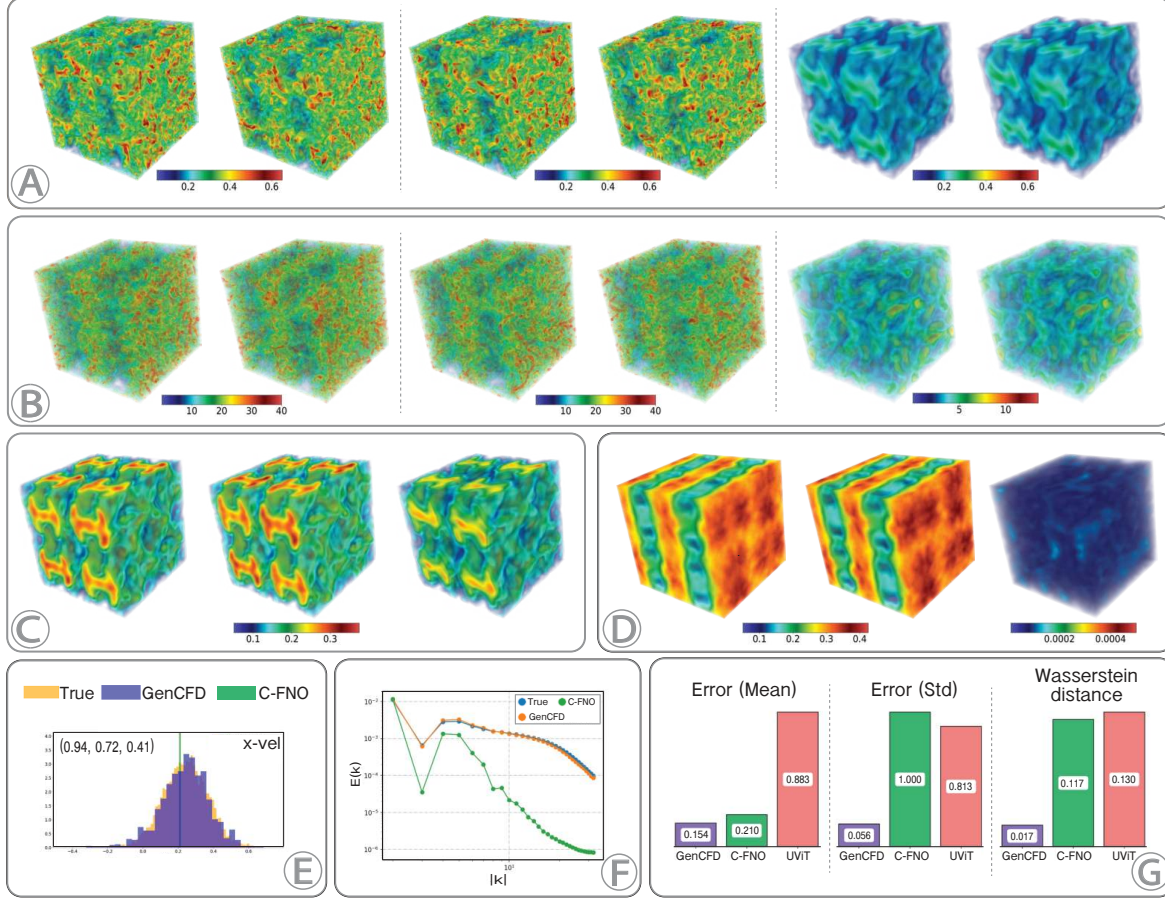
However, the same insensitivity hypothesis also implies that the denoisers  $D_{\theta}$  in a diffusion model such as GenCFD, being neural networks, will be *as insensitive to small input perturbations*,

i.e.,  $D_\theta(\bar{u}^* + \delta\bar{u}) \approx D_\theta(\bar{u}^*)$ , when  $\|\delta\bar{u}\|_{\mathcal{X}} < \bar{\epsilon}$ . Then, how are diffusion models such as GenCFD, based on the same neural networks, able to approximate the target distribution? The answer to this lies in the nature of the *loss function* (3) in training diffusion models as the following calculation shows. Starting with the specific form of the diffusion training loss (3) in our context as,

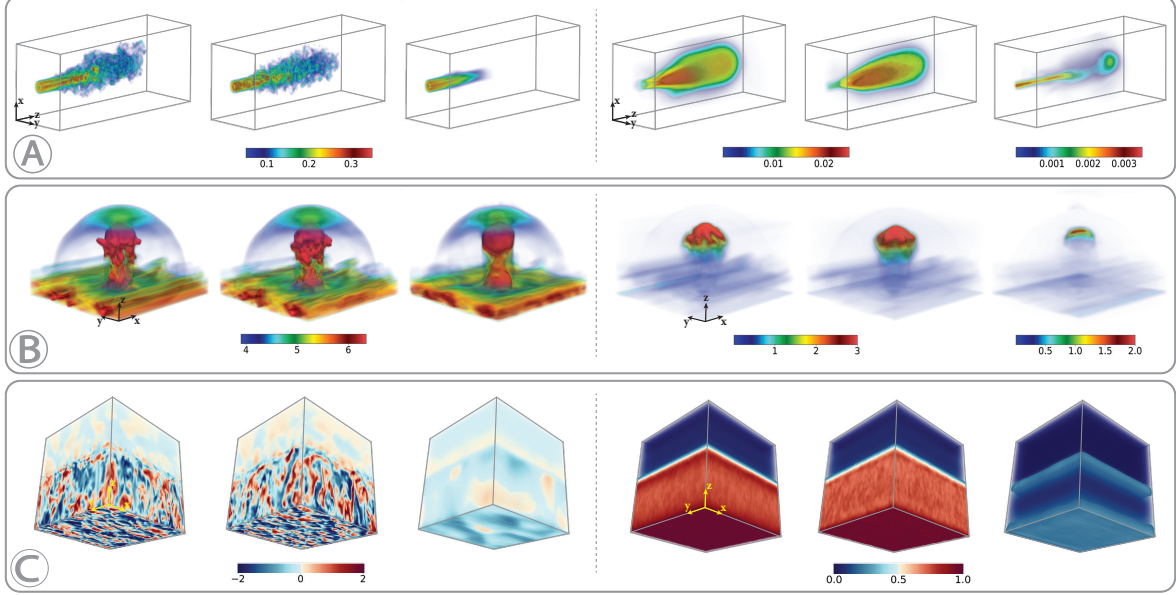
$$\begin{aligned} \mathcal{J}(D_\theta) &= \mathbb{E}_{\delta\bar{u}} \mathbb{E}_\eta [\|D_\theta(\mathcal{S}(\bar{u}^* + \delta\bar{u}) + \eta; \bar{u}^* + \delta\bar{u}, \sigma) - \mathcal{S}(\bar{u}^* + \delta\bar{u})\|^2] \\ &\approx \mathbb{E}_{\delta\bar{u}} \mathbb{E}_\eta [\|D_\theta(\mathcal{S}(\bar{u}^* + \delta\bar{u}) + \eta; \bar{u}^*, \sigma) - \mathcal{S}(\bar{u}^* + \delta\bar{u})\|^2] \quad (\text{insensitivity hypothesis}) \\ &= \mathbb{E}_u \mathbb{E}_\eta [\|D_\theta(u + \eta; \bar{u}^*, \sigma) - u\|^2], \end{aligned}$$

where the last line follows by a change of variables to  $u = \mathcal{S}(\bar{u}^* + \delta\bar{u})$ . Thus,  $\mathcal{J}$  is the *denoiser training objective or diffusion loss* for recovering the distribution corresponding to the  $\text{Law}_{\delta\bar{u}} \mathcal{S}(\bar{u}^* + \delta\bar{u})$ , conditioned on the input  $\bar{u}^*$ , which is precisely the goal of our statistical computation. This formal argument reveals the *surprising mechanism* through which a diffusion model can leverage the highly unstable nature of sensitive maps such as solution operators of fluid flows, to accurately approximate the conditional distributions, even when the underlying neural networks themselves are not sensitive to small perturbations, justifying the empirically observed performance of GenCFD.

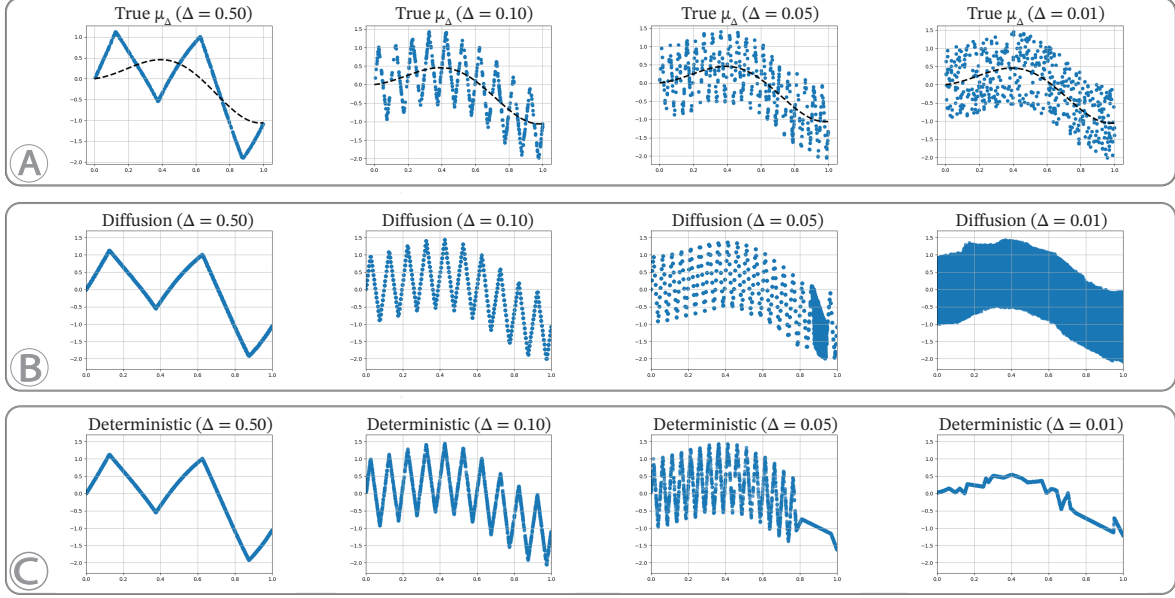
Given the formidable mathematical challenge of analytically characterizing the solution operators of the Navier–Stokes equations, we present *solvable toy models*, which capture essential features of turbulent fluid flows while still being analytically tractable. To this end, in **SI** Sec. 6 (See Fig. 4 (A) for visualization), we consider a sequence of simple maps between unit intervals, indexed by a small parameter  $\Delta$  that encodes input sensitivity. These maps contain oscillations on progressively finer and finer scales as  $\Delta \rightarrow 0$ . Consequently, the asymptotic limit of these maps can only be described in terms of (pointwise) statistics which are explicitly computed in the **SI** Sec. 7. As their Lipschitz constant blows up when  $\Delta \rightarrow 0$ , these maps are clearly very sensitive to small (spatial) input perturbations. On the other hand, the spectral bias of neural networks has been widely explored in the context of one-dimensional oscillatory maps [59] and it is well established that they fail to approximate high frequencies, making them *insensitive* to perturbations at such small scales. As expected from the theory presented above, Fig. 4 (C) shows how a multilayer perceptron (MLP) trained to minimize the mean square error between its prediction and the underlying map behaves as  $\Delta \rightarrow 0$  and increasingly higher frequency oscillations are introduced. We see that for relatively large  $\Delta$ , the ML model provides an accurate approximation, at least after a lot of training steps (see **SI** Fig. 28). However, when  $\Delta \ll 1$ , as predicted by the theory, the model fails to approximate the fine-scale oscillations and regresses to the mean. On the other hand, as shown in Fig. 4(B), a diffusion model (the same MLP but trained with the diffusion loss (3)) is able to approximate the underlying map for all values of  $\Delta$ , including when  $\Delta \ll 1$ , where it predicts the correct limit distribution. Both these observations are rigorously proved in the **SI** Sec. 7 by deriving explicit formulae for the optimal denoisers, putting our proposed theory on a firm mathematical footing for this toy problem. Moreover, in the **SI** Sec. 7, we also present and rigorously analyze a second toy problem which mimics the spectral behavior of fluid flows, reproducing energy spectra similar to Fig. 2 (F).



**Figure 2: Results for the Taylor–Green vortex dataset.** (A and B): Two Samples of the flow at time  $T = 2$  for the same initial condition with ground truth (Left), GenCFD (center) and C-FNO baseline (Right) for the pointwise kinetic energy (A) and vorticity intensity (B). (C): Mean and (D): Standard deviation, of the pointwise kinetic energy at time  $T = 2$  with ground truth (Left), GenCFD (Center) and C-FNO (Right). (E): PDF of the  $x$ -velocity component at the spatial point  $(0.94, 0.72, 0.41)$  and  $T = 2$  and (F): Energy spectrum at  $T = 2$ . (G): Errors in predicting the mean (Left), standard deviation (Center) and 1-Wasserstein distance (Right) at time  $T = 2$  with GenCFD, C-FNO and UVIT. Ground truth is generated by a DNS with a spectral hyperviscosity method. Please note the different ranges of the colorbars in (B) and (D).



**Figure 3: Results for the other flows: nozzle flow, cloud-shock interaction and dry convective boundary layer.** (A): A sample of the vorticity intensity (Left sub panel) and the standard deviation of the (pointwise) kinetic energy (Right sub panel) for the nozzle flow at time  $T = 1$  with ground truth (left), GenCFD (center) and UViT (right). (B): A sample of the density (Left sub panel) and standard deviation of the density (Right sub panel) at time  $T = 0.6$  for the cloud-shock interaction problem with the compressible Euler equations with ground truth (left), GenCFD (center) and C-FNO (right). (C): A sample of the  $x$ -component of the velocity (Left sub panel) and standard deviation of the  $x$ -velocity at time  $T = 2.4$  for the dry convective planetary boundary layer dataset with ground truth (Left), GenCFD (Center) and UViT (Right). The ground truth is generated by an LES with a lattice Boltzmann method (nozzle flow), a DNS with a high-resolution finite volume method (cloud-shock interaction) and an LES with a WENO finite difference method (convective planetary boundary layer). Please note the different ranges of the colorbars in Panels A and B (Right subpanels).



**Figure 4: Results for Toy Model #1.** This solvable toy model captures relevant aspects of turbulent fluids while being analytically tractable (for a detailed problem formulation, see **SI** Sec. 6). (A): Visualization of the underlying maps  $\mathcal{S}^\Delta$  (as in **SI** Sec. 6) for different values of  $\Delta$  representing oscillations at higher and higher frequencies (inversely proportional to  $\Delta$ ) being added to a mean function (in black), with the  $\Delta \rightarrow 0$  limit of  $\mathcal{S}^\Delta$  being uniform probability distributions at each point. (B): Results for different values of  $\Delta$  with a score-based diffusion model (with an MLP as the denoiser) after 10000 epochs of training to minimize the denoiser training objective or diffusion loss (3). (C): Results for different values of  $\Delta$ , with an MLP after 10000 epochs of training to minimize the mean square loss. Panel (B) clearly shows how the diffusion model can accurately approximate the underlying function for large  $\Delta$ , while approximating the underlying probability distribution for  $\Delta \approx 0$ . On the other hand, Panel (C) shows that the same neural network (MLP) trained to minimize the mean square loss is able to recover the underlying map for large  $\Delta$  but collapses to the mean as  $\Delta \approx 0$ . These observations are rigorously proved in the SI.



## 5 Concluding remarks

We address the outstanding challenge of designing very fast and accurate algorithms for the statistical computation of fluid flows by proposing GenCFD, a *conditional score-based diffusion model*. With extensive experimentation on a variety of three-dimensional datasets with comprehensive evaluation metrics, we demonstrate the capability of GenCFD to generate realistic and statistically accurate flows, while being several orders of magnitude faster in runtime than the optimized CFD solvers we tested (see **SI** Table 15). These empirical results hold equally well for popular academic benchmarks, such as the Taylor–Green vortex, as well as engineering flows, like turbulent round jets, and atmospheric flows, like the dry convective planetary boundary layer, showcasing the widespread applicability of GenCFD. Moreover, we provide theoretical arguments to support the strong performance of GenCFD: given the insensitivity of neural networks to very small perturbations, training the neural network to match trajectories of sensitive dynamical systems, such as the PDEs governing fluid flow, leads to a regression to the mean. However, we demonstrate that the same neural networks when trained to estimate underlying probability distributions in a Diffusion model are able to recover the statistical behavior of these sensitive dynamical systems.

Compared to traditional CFD solvers, the main advantage of GenCFD is its speed, which can be several orders of magnitude faster in accurately computing flow statistics (**SI** Table 15). Moreover, GenCFD is completely *data-driven* and agnostic to the underlying PDE, whereas a CFD solver needs explicit information about the latter. Compared to deterministic ML algorithms such as neural operators, the main advantage of GenCFD is its statistical accuracy, whereas ensembles based on these ML methods, albeit fast, are not statistically accurate as we have shown that they regress to the mean and fail to capture the underlying variance.

In the context of (turbulent) fluid flows, [20, 21, 22, 82, 51] do already consider diffusion models. However, these articles either focus on two-dimensional flows or on learning coarse-grained dynamics by using a diffusion model either for the decoding stage or for generating a sequence of coarse variables or by combining neural operators (for coarse graining) and diffusion models (for finer scales). Moreover, they mostly provide average metrics that do not demonstrate their ability to compute the entire target distribution, nor do they present any theoretical justification of why their methods work. In contrast, GenCFD is an end-to-end conditional diffusion model that can generate statistically accurate snapshots (of the whole trajectory) of turbulent flows given underlying inputs.

The algorithmic pipeline of GenCFD can be seamlessly modified to provide fast and accurate generation of turbulent flows to other high-impact applications, such weather and climate modeling [70], where atmospheric and oceanic flows are the cornerstone of earth system modeling. As GenCFD can accurately generate convective boundary layers, it is natural to extend it to moist flows such as clouds, which can have a large impact on cloud-resolving weather and climate simulations [70]. Along the same lines, the theoretical underpinning of GenCFD can also help understand the strong performance of diffusion model-based probabilistic weather emulators, such as [58], particularly when compared to ensembles generated by deterministic ML weather



emulators, which are accurate in short-term weather modeling, but they do not capture the true probabilistic nature of even medium-term, let alone long-term weather modeling [58, 6].

The general framework of GenCFD is very versatile, and it can be readily extended to other fluid flows, particularly around obstacles by either masking the obstacle [63] or adding graph neural network-based encoders and decoders [58]. Similarly, extensions to plasma flows, governed by magnetohydrodynamics equations, are relatively straightforward. Although diffusion models have been recently used in several applications such as statistical downscaling [47, 81], physical inverse problems [9, 83], ensemble augmentation [41], and data assimilation [67]; the theoretical analysis presented here provides further impetus for their adoption in an even wider variety of multiscale physical and engineering systems whose outputs depend sensitively on inputs. These include (but are not limited to): i) Computing invariant measures for the long-time limit of chaotic dynamical systems such as the Lorenz system; ii) Bayesian inverse problems [75], particularly for fluid flows, where the inverse operator is known to be unstable [37] even when the forward problem is well-posed and the goal of statistical computation is to sample from the posterior measure; iii) Non-convex variational problems in materials science [44] which model phase transitions in crystalline materials; iv) Homogenization of multi-scale materials, particularly in the modeling of composites [10].

Supplementary Information for:  
**Generative AI for fast and accurate  
statistical computation of fluids**

## Table of Contents

---

|           |                                                                                     |           |
|-----------|-------------------------------------------------------------------------------------|-----------|
| <b>6</b>  | <b>Methods</b>                                                                      | <b>18</b> |
| 6.1       | Problem Formulation . . . . .                                                       | 18        |
| 6.2       | Score-based Diffusion Models . . . . .                                              | 20        |
| 6.3       | The Denoiser . . . . .                                                              | 22        |
| 6.4       | Baselines . . . . .                                                                 | 28        |
| 6.5       | FNO . . . . .                                                                       | 28        |
| 6.6       | Training and Test Protocols . . . . .                                               | 29        |
| 6.7       | Datasets . . . . .                                                                  | 30        |
| 6.8       | Details of Models and Hyperparameters . . . . .                                     | 36        |
| 6.9       | Evaluation Metrics . . . . .                                                        | 38        |
| 6.10      | Details on Toy Model #1 of the Main Text . . . . .                                  | 40        |
| <b>7</b>  | <b>Detailed Theory</b>                                                              | <b>43</b> |
| 7.1       | Main results . . . . .                                                              | 43        |
| 7.2       | Toy Model #1: Illustrating the Consequences of Input Sensitivity Mismatch . . . . . | 47        |
| 7.3       | Toy Model #2: Illustrating Spectral Accuracy of Diffusion Models . . . . .          | 49        |
| 7.4       | Mathematical Derivation . . . . .                                                   | 52        |
| <b>8</b>  | <b>Further Experimental Results</b>                                                 | <b>74</b> |
| 8.1       | GenCFD Generates Very High-quality Samples of the Flow . . . . .                    | 74        |
| 8.2       | GenCFD Accurately Approximates Statistical Quantities of Interest . . . . .         | 75        |
| 8.3       | GenCFD Provides Excellent Spectral Resolution . . . . .                             | 76        |
| 8.4       | GenCFD Scales with Data . . . . .                                                   | 76        |
| 8.5       | The Statistical Computation with GenCFD is Robust . . . . .                         | 76        |
| 8.6       | Statistical Computation with GenCFD is Fast . . . . .                               | 77        |
| <b>9</b>  | <b>Supplementary Tables</b>                                                         | <b>80</b> |
| <b>10</b> | <b>Supplementary Figures</b>                                                        | <b>87</b> |

---

## 6 Methods

### 6.1 Problem Formulation

**Governing Equations.** The PDEs governing fluid flows are special cases of the generic time-dependent PDE

$$\begin{aligned}\partial_t u(x, t) + \mathcal{L}(u, \nabla_x u, \nabla_x^2 u, \dots) &= 0, \quad \forall x \in D \subset \mathbb{R}^d, t \in (0, T), \\ \mathcal{B}(u) &= u_b, \quad \forall (x, t) \in \partial D \times (0, T), \\ u(x, 0) &= \bar{u}(x), \quad x \in D,\end{aligned}\tag{4}$$

where,  $d$  is the spatial dimension,  $T$  is the time-horizon,  $u \in C(\mathcal{X}; [0, T])$  is the solution of (4), for a function space  $\mathcal{X} \subset L^p(D; \mathbb{R}^n)$  for some  $1 \leq p < \infty$ ,  $\bar{u} \in \mathcal{X}$  is the initial datum,  $u_b$  is the boundary datum, and  $\mathcal{L}, \mathcal{B}$  are the underlying differential and boundary operators, respectively.

Concrete examples of (4) are given by the well-known Navier–Stokes equations [46] for incompressible fluid flows, which take the form

$$\begin{aligned}\partial_t u + (u \cdot \nabla)u + \nabla p &= \nu \Delta u, \\ \operatorname{div} u &= 0,\end{aligned}\tag{5}$$

in a domain  $D \subset \mathbb{R}^d$  with suitable boundary conditions. Here,  $u : [0, T] \times D \rightarrow \mathbb{R}^d$  is the velocity field and  $p : [0, T] \times D \rightarrow \mathbb{R}$  is the pressure. The parameter  $\nu$  is the so-called *kinematic viscosity* of the fluid and is inversely proportional to the *Reynolds number* (Re).

Similarly, compressible fluid flow is modeled by the *compressible Navier–Stokes* equations [36]. Again, most compressible fluids of interest have  $\operatorname{Re} \gg 1$ . Consequently, one is interested in the corresponding infinite Reynolds number limit which yields the *compressible Euler equations* [36]. These nonlinear PDEs are special cases of so-called *hyperbolic systems of conservation laws*: a large class of PDEs of the generic form [8]

$$\partial_t u(x, t) + \nabla \cdot F(u(x, t)) = 0.\tag{6}$$

Here  $u : D \times [0, T] \rightarrow \mathbb{R}^m$  is the physical state with  $m$  components. The function  $F : \mathbb{R}^m \rightarrow \mathbb{R}^{d \times m}$  is the physical flux, which describes how the physical state variables are transported through the system, and  $\nabla_x \cdot F(u(x, t))$  is the (spatial) divergence of the vector field  $F \circ u : D \times [0, T] \rightarrow \mathbb{R}^{d \times m}$ ,  $(x, t) \mapsto F(u(x, t))$ , with components  $[\nabla_x \cdot F(u(x, t))]_j = \sum_{k=1}^d \partial_{x_k} (F_{k,j}(u(x, t)))$ .

In the specific example of the compressible Euler equations, the state variables are  $u = [\rho, \rho v, E]$ , with density  $\rho$ , velocity  $v = [v_{x_1}, v_{x_2}, \dots, v_{x_d}]$ , pressure  $p$  and total energy  $E$  related by the ideal gas equation of state

$$E = \frac{1}{2} \rho |u|^2 + \frac{p}{\gamma - 1},\tag{7}$$

with gas constant  $\gamma$ . The corresponding flux function is given by,

$$F = [\rho v, \rho v \otimes v + p \mathbf{I}, (E + p)v].\tag{8}$$

**Statistical Computation.** For simplicity of the exposition, we assume that the boundary conditions in the PDE (4) are fixed. Then, the solutions of the time-dependent PDE (4) are given in terms of the underlying *solution operator*  $\mathcal{S} : [0, T] \times \mathcal{X} \mapsto \mathcal{X}$  such that  $u(t) = \mathcal{S}^t(\bar{u}) = \mathcal{S}(t, \bar{u})$  is the solution of (4) at any time  $t \in [0, T]$ .

As mentioned in the Main Text, statistical computation of (4), also termed as forward uncertainty quantification (UQ), refers to the computation of the *push-forward measure*  $\hat{\mu}_t = \mathcal{S}_{\#}^t \bar{\mu}$  of some input measure  $\bar{u} \sim \bar{\mu} \in \text{Prob}(\mathcal{X})$  by the solution operator  $\mathcal{S}^t$  of (4).

Unfortunately, this solution operator may not be necessarily well-defined [13] (particularly when  $d = 3$ ), and even when  $\mathcal{S}^t$  is well-defined, it can be very sensitive to initial conditions, i.e.,  $\|\mathcal{S}^t(\bar{u}) - \mathcal{S}^t(\tilde{u})\|_{\mathcal{X}}$  can grow exponentially in time, even when  $\|\bar{u} - \tilde{u}\|_{\mathcal{X}} \ll 1$  [46]. Then, the question arises:

*How can we even define the push-forward measure  $\hat{\mu}_t = \mathcal{S}_{\#}^t \bar{\mu}$ ?*

To answer this question, we observe that, in practice, one cannot access the solution operator  $\mathcal{S}^t$  explicitly. Instead, one approximates  $\mathcal{S}^t$  with numerical simulations (or analytical approximations) resulting in an operator  $\mathcal{S}^{t,\Delta} \approx \mathcal{S}^t$  (in a suitable sense), for small enough values of a *discretization parameter*  $\Delta$ . As  $\mathcal{S}^{t,\Delta} : \mathcal{X}^{\Delta} \rightarrow \mathcal{X}^{\Delta}$  maps between finite dimensional spaces  $\mathcal{X}^{\Delta} \cong \mathbb{R}^N$  for  $N \gg 1$ , the push-forward  $\hat{\mu}_t^{\Delta} = \mathcal{S}_{\#}^{t,\Delta} \bar{\mu}$  is always well-defined. Given this, we consider the limit

$$\hat{\mu}_t = \lim_{\Delta \rightarrow 0} \hat{\mu}_t^{\Delta} = \lim_{\Delta \rightarrow 0} \mathcal{S}_{\#}^{t,\Delta} \bar{\mu}, \quad (9)$$

in a suitable topology. Clearly, if  $\mathcal{S}^{t,\Delta} \rightarrow \mathcal{S}^t$  as  $\Delta \rightarrow 0$ , the above limit is simply  $\hat{\mu}_t = \mathcal{S}_{\#}^t \bar{\mu}$ . The interesting case materializes when the deterministic approximation  $\mathcal{S}^{t,\Delta}$  does not converge to a well-defined limit as  $\Delta \rightarrow 0$  as in the case of computing unstable and turbulent fluid flows [14, 13, 39, 16]. Nevertheless, the limit (9) can still be well-defined and one can even observe strong convergence to the limit, in the sense that  $W_p(\hat{\mu}_t^{\Delta}, \hat{\mu}_t) \rightarrow 0$  as  $\Delta \rightarrow 0$ , for the appropriate  $p$ -Wasserstein metric on measures [39, 16]. These observations are formalized under the rubric of the theory of *statistical solutions* of PDEs [18, 15, 17] and the limit measure  $\hat{\mu}_t$  is termed as the statistical solution of the PDE (4).

Given the above discussion, the goal of statistical computation for fluid flows can be reformulated as follows: we are given initial data  $\bar{\mu} \in \text{Prob}(\mathcal{X})$ , which we approximate by a finite-dimensional projection  $\bar{\mu}^{\Delta}$  for a given discretization parameter  $\Delta > 0$ . With the approximate solution operator  $\mathcal{S}^{t,\Delta}$  given by a suitable finite-dimensional discretization of (4), and the map  $(\mathcal{S}^{t,\Delta} \times \text{ID}) : \mathcal{X} \rightarrow \mathcal{X} \times \mathcal{X}$ , such that  $(\mathcal{S}^{t,\Delta} \times \text{ID})(\bar{u}) = (\mathcal{S}^{t,\Delta}(\bar{u}), \bar{u})$ , we consider the distribution,  $\mu_t^{\Delta} := (\mathcal{S}^{t,\Delta} \times \text{ID})_{\#} \bar{\mu}^{\Delta}$  in order to explicitly condition the evolution in terms of the initial data. The corresponding limit distribution is given by

$$\mu_t = \lim_{\Delta \rightarrow 0} \mu_t^{\Delta} = \lim_{\Delta \rightarrow 0} (\mathcal{S}^{t,\Delta} \times \text{ID})_{\#} \bar{\mu}^{\Delta}. \quad (10)$$

Following [17, 39, 16], the limit can be well-defined under suitable hypotheses and  $W_p(\mu_t^{\Delta}, \mu_t) \rightarrow 0$ , even when the solution operator  $\mathcal{S}^{t,\Delta}$  does not converge. In general, this limit measure admits

a *conditional representation*

$$\mu_t(du, d\bar{u}) = P_t(du | \bar{u}) \bar{\mu}(d\bar{u}), \quad (11)$$

where  $P_t(du | \bar{u})$  represents the conditional probability distribution of  $u(t)$  given  $\bar{u}$ . If  $\bar{\mu} = \delta_{\bar{u}}$  and  $\mathcal{S}^{t,\Delta}$  converge to the solution operator  $\mathcal{S}^t$  of (4), then  $P_t(du | \bar{u}) = \delta_{\mathcal{S}^t(\bar{u})}$ . However, the interesting case of unstable and turbulent fluid flows corresponds to a *non-Dirac spread-out* conditional probability measure, even when the initial measure is concentrated on a function (See Main Text Fig. 1 (A) for an illustration).

Given the empirical fact that  $\mu_t^\Delta \rightarrow \mu_t$  as  $\Delta \rightarrow 0$ , we choose  $\Delta$  sufficiently small and rely on a disintegration property similar to (11) to realize the conditional probability measure  $P_t^\Delta(du | \bar{u})$  with

$$\mu_t^\Delta(du, d\bar{u}) = P_t^\Delta(du | \bar{u}) \bar{\mu}^\Delta(d\bar{u}). \quad (12)$$

This brings us to the goal for our statistical computation of fluid flows:

*For discretization parameter  $\Delta \ll 1$  and given initial measure  $\bar{\mu}^\Delta$ , compute the conditional probability measure  $P_t^\Delta(du | \bar{u})$  (12) that characterizes uncertainty in a fluid flow.*

For notational simplicity, we fix  $t$  and  $\Delta$  and suppress the dependence on  $t$  and  $\Delta$  in (12). We also observe that all the measures in (12) are supported in finite dimensions.

We will usually represent  $\bar{\mu}$  and  $P(du|\bar{u})$  by their (generalized) densities  $\bar{p}(\bar{u})$  and  $p(u | \bar{u})$ , respectively. Hence, as stated in the Main Text, the goal of statistical computation boils down to approximating the conditional distribution  $p(u | \bar{u})$ , given the initial measure  $\bar{p}(\bar{u})$  (Main Text Fig. 1 (A)).

## 6.2 Score-based Diffusion Models

As mentioned in the Main Text, we will adapt a specific generative AI algorithm, namely score-based diffusion models [26, 80, 73, 74] for computing the conditional distribution  $p(u | \bar{u})$ , conditioned on the initial (*prior*) measure  $\bar{p}(\bar{u})$ . We recall from the Main Text that diffusion models, [78] and references therein, learn probability distributions based on a very simple idea, realized in terms of a process with two steps (see Main Text Fig. 1 (B) for an illustration). In a forward step, *noise* is iteratively added to samples drawn from the target distribution in order to transform it to a known noisy distribution, typically of the Gaussian type. The key *reverse* step is based on *denoising*. In it, noise is iteratively removed from samples drawn from the known noisy distribution and they are transformed into samples that follow the target distribution. Different diffusion models differ in how the denoising step is performed in practice. Here, we adapt the widely used *score-based diffusion models*, [31] and references therein.

**Learning Unconditional Distributions with Score-based Diffusion Models.** For the ease of exposition, we will first consider the case of a target distribution  $p \in \text{Prob}(\mathbb{R}^N)$  that we wish to learn from data, the *forward step* in a score-based diffusion model consists of adding

noise to samples drawn from  $p$  by solving the *stochastic differential equation* (SDE) [31]

$$du_\tau = \frac{\dot{s}_\tau}{s_\tau} u_\tau d\tau + s_\tau \sqrt{2\dot{\sigma}_\tau \sigma_\tau} dW_\tau, \quad \text{for } \tau \in [0, K], \quad (13)$$

with time index  $\tau$ , which stands for the time variable in the *diffusion process* and is not related to the time  $t$  used to express the physical time evolution in the PDE (4). The drift and diffusion coefficients are given in terms of the *shape function*  $s_\tau$  and noise function  $\sigma_\tau$ , respectively, and  $W_\tau$  is the  $N$ -dimensional standard Wiener process. The shape and noise functions  $s_\tau, \sigma_\tau$  are chosen such that setting  $s_0 = 1, \sigma_0 = 0$  results in aligning the marginal distribution  $p_0 = p$  with the target distribution  $p$ .

Solving the SDE (13) forward in time  $\tau$  results in the addition of noise to the samples  $u_0 \sim p_0 = p$ , transforming them to samples drawn from a so-called *Gaussian Perturbation Kernel*

$$p_K(u_K) \sim \mathcal{N}(u_K; 0, s_K^2 \sigma_K^2 I), \quad (14)$$

leading to a terminal distribution which is indistinguishable from an *isotropic Gaussian with zero mean* at time  $\tau = K$ .

The *reverse step* in a score-based diffusion model consists of solving the so-called *reverse SDE*

$$du_\tau = \left[ \frac{\dot{s}_\tau}{s_\tau} u_\tau - 2\dot{\sigma}_\tau \sigma_\tau s_\tau^2 \nabla_u \log p_\tau(u_\tau) \right] d\tau + s_\tau \sqrt{2\dot{\sigma}_\tau \sigma_\tau} d\widehat{W}_\tau \quad (15)$$

*backward in time* with terminal distribution  $p_K$  (as defined in (14)), while  $p_\tau$  is the underlying distribution at any time  $\tau \in [0, K]$ . This reverse process yields the desired target distribution  $p_0 = p$  as the initial distribution at  $\tau = 0$ .

While the forward SDE (13) is straightforward to simulate, once the so-called *diffusion schedule* ( $s_\tau, \sigma_\tau$ ) is given, solving the reverse-SDE (15) needs the (approximate) knowledge of the so-called *score function*  $\nabla_u \log p_\tau(u)$ . The approximation of this score function lies at the heart of any diffusion model.

For our work, we will adopt the widely-used framework of [31] and approximate the score function in (15) via a *denoiser*  $D_\theta(u + \epsilon_\tau, \sigma_\tau)$ , which is a parametric function with parameters  $\theta \in \Theta \subset \mathbb{R}^M$ . Given a sample  $u \sim p$ , drawn from the target distribution  $p$  and the given noise level  $\epsilon_\tau = \epsilon \sigma_\tau$ , for the noise function  $\sigma_\tau$  and a parameter  $\epsilon$ , the parameters  $\theta$  of the denoiser are learned by minimizing the error in predicting the underlying *clean* sample  $u$ . The remarkable Tweedie's formula [78] then relates the score-function in (15) as

$$\nabla_u \log p_\tau(u_\tau) \approx \frac{D_\theta(\hat{u}_\tau, \sigma_\tau) - \hat{u}_\tau}{s_\tau \sigma_\tau^2}, \quad \text{with } \hat{u}_\tau := \frac{u_\tau}{s_\tau}, \quad (16)$$

enabling the solution of the reverse SDE (15). Thus, one needs to specify the diffusion schedule and the denoiser architecture in order to characterize a diffusion model.

**Learning Conditional Distributions.** Given that our goal of statistical computation of fluid flows entails computing conditional distributions  $p(u | \bar{u})$ , we need to adapt the score-based



diffusion model presented above. To this end, we follow the approach of [3] and modify the denoiser in (16) to take the form

$$D_\theta(u_\tau, \sigma_\tau) \rightarrow D_\theta(u_\tau(\bar{u}), \bar{u}, \sigma_\tau), \quad (17)$$

with noise  $\sigma_\tau$  now added to samples  $u(\bar{u})$  drawn from the underlying conditional distribution  $p(u | \bar{u})$ . Moreover, samples drawn from the prior distribution  $\bar{u} \sim \bar{p}$  are explicit inputs to the denoiser in (17). Theorem 1 of [3] shows that Tweedie’s formula (16) can be readily modified in this case to yield Formula (2) of the Main Text:

$$\nabla_u \log p_\tau(u_\tau | \bar{u}) \approx \frac{D_\theta(u_\tau(\bar{u}), \bar{u}, \sigma_\tau) - \hat{u}_\tau}{s_\tau \sigma_\tau^2}. \quad (18)$$

Samples from the target conditional distribution  $p(u | \bar{u})$  are now drawn by simulating the reverse SDE, Equation (1) of the Main Text, with the score function  $\nabla_u \log p_\tau(u_\tau | \bar{u})$  being approximated by the denoiser (18).

**Learning Time-Conditioned Distributions.** Given the time-dependent nature of our underlying PDE (4), we need to learn probability distributions  $p(u | (t, \bar{u}))$ , with  $t \in [0, T]$  being the time and  $\bar{u} \sim \bar{p}$ , the (finite-dimensional approximation of) initial data. Hence, the denoiser (17) has to be further modified to condition it on the time variable  $t$  so that the entire trajectory of the distribution can be generated. Moreover, given the results of [24] where a novel *all-to-all* training procedure was proposed for learning solution operators of time-dependent PDEs, we will similarly exploit the *semi-group* property of the solution operator of (4) to *condition the denoiser on lead times*. To this end, we further modify the denoiser (17) to

$$D_\theta(u_\tau(\bar{u}), \bar{u}, \sigma_\tau) \rightarrow D_\theta(t_n - t_\ell, u_\tau(t_n, \bar{u}), u(t_\ell, \bar{u}), \sigma_\tau), \quad (19)$$

with times  $0 \leq t_\ell \leq t_n \leq T$ , initial data  $\bar{u} \sim \bar{p}$  and  $u(t_\ell, \bar{u})$  being the state of the system at a previous time step  $t_\ell$  and noise  $\sigma_\tau$  added to the current state  $u(t_n, \bar{u})$  of the system at time  $t_n$ . Consequently, these intermediate conditional distributions can be chained together to learn the target conditional distribution by

$$p(u | \bar{u}) = \prod_{\ell=1}^L p(u(t_\ell, \bar{u}) | u(t_{\ell-1}, \bar{u})), \quad (20)$$

with  $0 \leq t_0 < t_1 < \dots < t_\ell < \dots < t_L = T$  being a set of monotonically increasing lead times and  $p(u | \bar{u})$  being the conditional distribution at the final time  $T$ , given the initial condition  $\bar{u} \sim \bar{p}$ .

### 6.3 The Denoiser

The main remaining step in specifying our conditional score-based diffusion model is to choose the architecture and training process for the denoiser in (19).

### 6.3.1 The Denoiser Architecture

We choose a UViT [69] as the model for our denoiser (19), see Fig. 5 for a schematic. As seen in this figure, the model takes the lead time, the noisy sample at the current time, the underlying sample at a previous time, and the noise level to output a *denoised* or clean sample. The inputs are lifted into a latent space and processed through a set of *convolutional hidden layers*, which are stacked together in an *encoder-decoder* form as suggested in the very popular U-Net architecture of [66] in order to enable *multi-scale* information processing. In contrast to the standard U-Net, UViT replaces a convolutional layer at the bottleneck (base of the U-Net) with a *global attention layer* such that global mixing can take place in latent space. In three space dimensions, *axial attention* blocks [27] replace the global attention layer for computational efficiency. Residual skip connections are added to transfer information between the encoder and decoder at all hidden layers. Finally, the noise level and lead time are conditioned into the model at all levels by incorporating them inside the conditional layer norms of the model. All these steps are further detailed below.

As illustrated in Fig. 5, the operations of the denoiser start with the input data  $\bar{u} \in \mathbb{R}^{(v \times h \times w)}$  being projected into an embedding space of dimension  $C_0$  through a convolutional layer. The data is then sequentially downsampled  $n$  times, reducing its resolution in each dimension by a factor of  $2^n$ . Each downsampling step is followed by a residual block composed of  $n_{res}$  layers, where each layer is structured as a sequence of a group normalization layer (denoted as GN in the figure), a non-linear activation function  $f$ , and a convolutional layer  $\mathcal{C}$ . This sequence is repeated twice. Meanwhile, the noise level  $\sigma_\tau$  and lead time  $t_n$  are embedded using two independent Fourier projections, concatenated, and processed by a multi-layer perceptron (MLP) comprised of two linear layers  $L$  and a non-linear activation function  $f$ . The MLP generates scale  $a$  and shift  $b$  parameters, which are used to condition the second group normalization in the residual blocks. Following the residual blocks, multi-head attention mechanisms are applied in each layer. To address the computational cost of global attention in 3D data, axial attention is implemented instead [28]. On the other hand, data upsampling is performed through a *depth-to-space* operation. In other words, a linear transformation is first performed to increase the number of channels by a factor of 4, transforming the input tensor of shape  $(h, w, c)$  to  $(h, w, 4c)$  and subsequently reshaped as  $(c, 2h, 2w)$ . Similarly to the downsampling stack, in the upsampling stack, each upsampling layer is followed by residual and attention blocks. Skip connections are used between downsampling and upsampling stacks.

Below, we rigorously formulate the building pieces of the main blocks of the denoiser architecture.

**Affine Transformation.** A linear layer in neural networks performs an affine transformation on the input data  $x \in \mathbb{R}^{h \times w \times c}$ , defined as:

$$L : \mathbb{R}^c \rightarrow \mathbb{R}^{\hat{c}}, \quad L(x) = xW + b, \quad (21)$$

where  $W \in \mathbb{R}^{c \times \hat{c}}$  is the weight matrix containing the learnable parameters, and  $b \in \mathbb{R}^{\hat{c}}$  is the bias vector, which also consists of learnable parameters.

**Convolution.** The discrete, multi-channel convolution with an  $s$ -stride of the input  $x$  is defined as follows:

$$\mathcal{C} : \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^{\hat{h} \times \hat{w} \times \hat{c}}, \quad \mathcal{C}(x) = (x \star K_w)[i, j, \hat{l}] := \sum_{m,n=0}^{k-1} \sum_{l=1}^c K_w[m, n, l, \hat{l}] \cdot x[is + m, js + n, l], \quad (22)$$

where  $l, \hat{l}$  correspond to the indices of the input and output channels, respectively, and  $i = 0, \dots, h-1, j = 0, \dots, w-1, \hat{l} = 1, \dots, \hat{c}$ . Downsampling in the architecture in Fig. 5 is performed with a convolution operation with kernel size 3 and stride 2.

**Group Normalization.** Group normalization (GN) is a technique used to normalize the features of an input tensor  $x \in \mathbb{R}^{h \times w \times c}$ . Unlike batch normalization, which normalizes across the batch dimension, group normalization divides the channels into groups and normalizes the features within each group. Specifically, the channels are split into  $G$  groups, each containing  $\frac{c}{G}$  channels. For each group  $g$ , the mean  $\mu_g$  and variance  $\sigma_g^2$  are computed as

$$\mu_g = \frac{1}{m} \sum_{i \in g} x_i, \quad \sigma_g^2 = \frac{1}{m} \sum_{i \in g} (x_i - \mu_g)^2, \quad (23)$$

respectively, where  $m = \frac{hwc}{G}$  is the number of elements in each group. The normalized output is then given by

$$\hat{x}_i = \frac{x_i - \mu_g}{\sqrt{\sigma_g^2 + \epsilon}}, \quad (24)$$

where  $\epsilon$  is a small constant for numerical stability. Finally, a learnable scale  $\gamma$  and shift  $\beta$  are applied to each normalized group, yielding the final output

$$\tilde{x}_i = \gamma \hat{x}_i + \beta. \quad (25)$$

**Fourier Embedding and Adaptive Scale.** Given an input tensor  $x \in \mathbb{R}^{h \times w \times 1}$ , where the last dimension typically represents the temporal lead time  $t$  or the diffusion noise  $\sigma$ , Fourier embeddings transform these coordinates into a higher-dimensional space as

$$\gamma(x) = [\sin(2\pi Bx), \cos(2\pi Bx)], \quad (26)$$

where  $B \in \mathbb{R}^{d_f \times 2}$  is a matrix of frequencies, chosen from a fixed grid, and  $d_f$  is the dimensionality of the Fourier feature space. The resulting embedding  $\gamma(x)$  has a shape of  $h \times w \times 2d_f$ , capturing both the sine and cosine components at multiple frequencies. Two independent Fourier embeddings are used for the lead time and diffusion noise. The embeddings  $\gamma_t(x)$  and  $\gamma_\sigma(x)$  are then concatenated together to

$$\gamma(x) = \text{Concat}(\gamma_t(x), \gamma_\sigma(x)) \quad (27)$$

and transformed through an MLP  $\mathcal{M}$ , defined as

$$\mathcal{M}(\gamma(x)) = \text{GeLU}(\gamma(x)W_1 + b_1)W_2 + b_2, \quad (28)$$

with  $W_1 \in \mathbb{R}^{2d_f \times C_0}$ ,  $b_1 \in \mathbb{R}^{C_0}$ ,  $W_2 \in \mathbb{R}^{C_0 \times 2C_0}$ ,  $b_2 \in \mathbb{R}^{2C_0}$ . The output of the MLP is then split into a scale  $a \in \mathbb{R}^{C_0}$  and shift  $b \in \mathbb{R}^{C_0}$  used to adjust suitable group normalization in the residual blocks of the UVit as

$$\tilde{x} = (a + 1)GN(x) + b. \quad (29)$$

**Multi-Head Attention.** Given an input tensor  $x \in \mathbb{R}^{h \times w \times c}$  ( $x \in \mathbb{R}^{h \times w \times d \times c}$  for 3D problems), following common practice in vision, the tensor is first reshaped into a new one of shape  $(hw, c)$  ( $(hwd, c)$  for 3D problems). To preserve spatial information, a positional embedding  $p \in \mathbb{R}^{hw \times c}$ , learned during training, is added to the reshaped tensor, yielding  $x' = x + p$ . The multi-head attention is then defined as

$$\text{MHA}(x') = \text{Concat}(\mathcal{A}_1(x'), \dots, \mathcal{A}_h(x'))W^O, \quad (30)$$

where  $W^O \in \mathbb{R}^{c \times c}$  is the output projection matrix, and  $\mathcal{A}_l(x')$ , for  $l = 1, \dots, h$ , represents the outputs of each attention head. Each head  $\mathcal{A}_k$  computes scaled dot-product attention as follows:

$$\mathcal{A}_k(x') = \text{Attention}(Q_k, K_k, V_k) = \text{softmax}\left(\frac{Q_k K_k^\top}{\sqrt{d_k}}\right) V_k, \quad (31)$$

where  $Q_k = x'W_k^Q$ ,  $K_k = x'W_k^K$ , and  $V_k = x'W_k^V$  are the query, key, and value projections for the  $k$ -th head.  $d_k$  is the dimension of each head's subspace (typically  $d_k = \frac{c}{h}$  for  $h$  heads), and  $W_k^{Q,K,V} \in \mathbb{R}^{d_k \times d_k}$  are the respective projection matrices. The attention block in the UVit architecture is defined as

$$\tilde{x} = x + \text{MHA}(\text{GN}(x)), \quad (32)$$

where GN is a group norm with 32 groups.

**Axial Attention.** The memory requirements of standard attention scale quadratically with the sequence length. In the case of multidimensional problems, it becomes prohibitively expensive also if performed at the bottleneck of a U-structure like the ones used in UVit. To circumvent this problem, axial attention has been proposed [28]. Axial attention performs attention sequentially over each axis  $i$  of the tensor  $x$ , mixing information only along the axis  $i$ . It is implemented by simply transposing all axes except  $i$  to the batch axis, and performing MHA as described above. In the case of 3D problems, given the input tensor  $x \in \mathbb{R}^{h \times w \times d \times c}$ , the axial attention block along the axis  $i$  can be defined as

$$\tilde{x} = x + \text{GN}_2(\text{MHAA}_i(\text{GN}_1(x))), \quad (33)$$

where  $\text{MHAA}_i$  denotes multi-head axial attention along the  $i$ -th axis, i.e.

$$\text{MHAA}_i(x) = \text{MHA}(\text{Transpose}_i(x)). \quad (34)$$

Here,  $\text{Transpose}_i$  is the operation transposing all axes except  $i$  to the batch axis. The block is repeated for each axis.

### 6.3.2 Denoiser Training

The parameters  $\theta \in \Theta \subset \mathbb{R}^M$  that define the denoiser (19) need to be determined from training data. To this end, we consider training data in the form of *trajectories*  $S_i = \{u(t_\ell, \bar{u}^i)\}_{\ell=1}^L$ , for  $1 \leq i \leq I$ , with all  $\bar{u}^i \sim \bar{p}$  and  $0 = t_0 < t_1 < \dots < t_\ell < \dots < t_L = T$ . Then, the denoiser parameters are given as the (local) minimizers for  $\theta \in \Theta$  for the *denoising loss function*:

$$L(D_\theta, \sigma) = \mathbb{E}_{\bar{u}^i} \mathbb{E}_{(u(t_\ell, \bar{u}^i), u(t_n, \bar{u}^i))} \mathbb{E}_\eta \|D_\theta(t_n - t_\ell, u(t_n, \bar{u}^i) + \eta, u(t_\ell, \bar{u}^i), \sigma) - u(t_n, \bar{u}^i)\|^2. \quad (35)$$

Here,  $\bar{u}^i \sim \bar{p}$ ,  $(u(t_\ell, \bar{u}^i), u(t_n, \bar{u}^i)) \sim S_i$ ,  $\eta \sim \mathcal{N}(0, \sigma^2 I)$ ,  $t_n > t_\ell$ , and  $0 < t_n \leq T$ . Thus, for any noise level  $\sigma$ , the denoiser is trained in order to remove the noise from a noisy sample and output the *clean* sample (see Main Text Fig. 1 (C) for an illustration).

Our training of the denoiser-based diffusion model largely follows the methodology proposed in [31]. For ease of notation, in the rest we will set  $\sigma_\tau = \sigma$ ,  $u_\tau = u$ , and focus without loss of generality and for the ease of presentation on the unconditional case. In this case, the loss function for training the denoiser  $D_\theta$  simplifies to

$$\mathcal{L}(D_\theta, \sigma) = \mathbb{E}_{u \sim p_{\text{data}}} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2 I)} \|D_\theta(u + \eta, \sigma) - u\|^2 \quad (36)$$

and

$$\mathcal{L}(D_\theta) = \mathbb{E}_{\sigma \sim p_{\text{train}}} [\lambda(\sigma) L(D_\theta, \sigma)], \quad (37)$$

where  $\lambda$  is a weight dependent on the noise level  $\sigma$  and  $p_{\text{train}} = p_{\text{train}}(\ln \sigma) = \mathcal{U}(\ln(\sigma_{\min}), \ln(\sigma_{\max}))$ ,  $\sigma_{\min} = 10^{-3}$ ,  $\sigma_{\max} = 80$ .

Moreover, in score-based diffusion models, it is common [31] to precondition the denoiser predictions as

$$D_\theta(u + \eta, \sigma) = c_{\text{skip}}(\theta)(u + \eta) + c_{\text{out}}(\theta) F_\theta(c_{\text{in}}(\sigma)(u + \eta); c_{\text{noise}}(\sigma)) \quad (38)$$

where  $F_\theta$  is the raw U-Net model. Upon defining

$$F_{\text{target}} = \frac{1}{c_{\text{out}}(\sigma)} (u - c_{\text{skip}}(\sigma)(u + \eta)), \quad (39)$$

the loss function can be rewritten as

$$\mathcal{L}(D_\theta) = \mathbb{E}_{\sigma, u, \eta} [w(\sigma) \|F_\theta(c_{\text{in}}(\sigma)(u + \eta), c_{\text{noise}}(\sigma)) - F_{\text{target}}\|_2^2]. \quad (40)$$

Here,  $c_{\text{noise}}$  is chosen to be  $c_{\text{noise}}(\sigma) = \frac{1}{4} \log(\sigma)$ . On the other hand,  $c_{\text{in}}$ ,  $c_{\text{out}}$ ,  $c_{\text{skip}}$  are derived by imposing the following requirements:

$$\text{Var}[c_{\text{in}}(\sigma)(u + \eta)] = 1 \quad \Rightarrow \quad c_{\text{in}}(\sigma) = \frac{1}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \quad (41)$$

$$\text{Var}[F_{\text{target}}] = 1 \quad \Rightarrow \quad c_{\text{out}}(\sigma) = \sigma \cdot \sqrt{\sigma^2 + \sigma_{\text{data}}^2}, \quad (42)$$

$$c_{\text{skip}}(\sigma) = \underset{c_{\text{skip}}}{\text{argmin}} c_{\text{skip}}(\sigma) \quad \Rightarrow \quad c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}}{\sigma^2 + \sigma_{\text{data}}^2}. \quad (43)$$

Finally, the weight  $\lambda$  is obtained by requiring the effective weight  $w(\sigma)$  to be uniform across noise levels, i.e.

$$w(\sigma) = 1 \quad \Rightarrow \quad \lambda(\sigma) = \frac{\sigma^2 + \sigma_{\text{data}}^2}{(\sigma \sigma_{\text{data}})^2}. \quad (44)$$

Details can be found in Appendix B.6 of [31]. In all the experiments addressed in this paper,  $\sigma_{\text{data}} = 0.5$ .

**Variance Capturing Loss.** In order to improve the approximation of the standard deviation of the generated samples, the following term can be added to the loss function:

$$\mathcal{L}_{sq}(D_\theta) = \mathbb{E}_{\sigma, u, \eta} [w(\sigma) \|D_\theta(u + \eta, \sigma)^2 - u^2\|_2^2]. \quad (45)$$

This is motivated by the fact that the variance of a random variable  $x$  is defined as

$$\mathbb{V}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2. \quad (46)$$

If we assume that the denoiser prediction  $\tilde{u} = D_\theta(u + \eta, \sigma)$  is a Gaussian perturbation of the noise-free data  $u$ , i.e.  $\tilde{u} = u + \psi$  (for instance, for large  $\sigma$ , the denoiser predictions are still affected by noise), then

$$\mathbb{V}[\tilde{u}] = \mathbb{V}[u + \psi] = \mathbb{E}[u^2] + \mathbb{E}[\psi^2] + 2\mathbb{E}[u\psi] - \mathbb{E}[u]^2 \quad (47)$$

and

$$\mathbb{V}[\tilde{u}] - \mathbb{V}[u] = \mathbb{E}[\psi^2] + 2\mathbb{E}[u\psi]. \quad (48)$$

Therefore, the error in the variance learned samples can be reduced by minimizing (45), weighted by a factor  $\lambda_{sq}$ , together with (37), i.e.

$$L(D_\theta) = \mathcal{L}(D_\theta) + \lambda_{sq} \mathcal{L}_{sq}(D_\theta). \quad (49)$$

### 6.3.3 Inference of the Diffusion Model

At the time of inference, samples can be generated by solving the following SDE:

$$du = 2 \left( \frac{\dot{\sigma}}{\sigma} + \frac{\dot{s}}{s} \right) d\tau - 2s \frac{\dot{\sigma}}{\sigma} D_\theta \left( \frac{u}{s}, \sigma \right) d\tau + s \sqrt{2\dot{\sigma}} dw. \quad (50)$$

Details can be found in [31], Appendix B. More specifically, we use the Euler–Maruyama method, where the time steps are defined according to a sequence of noise levels  $\{\sigma_i\}$ ,  $\tau_i = \sigma^{-1}(\sigma_i)$ ,  $i = 1, \dots, N$ ,  $N = 128$  and

$$\sigma_i = \left( \sigma_{\max}^{\frac{1}{\rho}} + \frac{i}{N-1} (\sigma_{\min}^{\frac{1}{\rho}} - \sigma_{\max}^{\frac{1}{\rho}}) \right)^{\rho}. \quad (51)$$

At this point, it only remains to define the  $\sigma$  and  $s$  scheduling.



**Sigma Scheduler.** Two most common choices of the noise scheduler  $\sigma$  are

- exponential:  $\sigma_\tau = \mathcal{O}(\exp(\tau))$ ,
- tangent:  $\sigma_\tau = \mathcal{O}(\tanh(\tau))$ .

In all experiments, the exponential noise scheduler is used, as it was empirically determined to be a more effective choice for our models.

**Diffusion Scheme.** We employ the variance-exploding (VE) schedule, which sets the forward scheduler  $s_\tau = 1, \forall t$ .

## 6.4 Baselines

In addition to our conditional score-based diffusion model, GenCFD, we consider the following ML baselines for all our experiments. Note that all these baselines are trained to minimize the mismatch between their predictions and the ground truth in the mean square or  $L^2$ -norm.

## 6.5 FNO

A *Fourier neural operator* (FNO)  $\mathcal{G}$  [42] is a composition

$$\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}, \quad \mathcal{G} = Q \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ R. \quad (52)$$

It has a *lifting operator*  $u(x) \mapsto R(u(x), x)$ , where  $R$  is represented by a linear function  $R : \mathbb{R}^{d_u} \rightarrow \mathbb{R}^{d_v}$  where  $d_u$  is the number of components of the input function and  $d_v$  is the *lifting dimension*. The operator  $Q$  is a non-linear projection, instantiated by a shallow neural network with a single hidden layer, 128 neurons and GeLU activation function, such that  $v^{L+1}(x) \mapsto \mathcal{G}(u)(x) = Q(v^{L+1}(x))$ .

Each *hidden layer*  $\mathcal{L}_\ell : v^\ell(x) \mapsto v^{\ell+1}(x)$  is of the form

$$v^{\ell+1}(x) = \sigma \left( W_\ell \cdot v^\ell(x) + \left( K_\ell v^\ell \right) (x) \right), \quad (53)$$

with  $W_\ell \in \mathbb{R}^{d_v \times d_v}$  a trainable weight matrix (residual connection),  $\sigma$  an activation function, corresponding to GeLU, and the *non-local Fourier layer*

$$K_\ell v^\ell = \mathcal{F}_N^{-1} \left( P_\ell(k) \cdot \mathcal{F}_N v^\ell(k) \right), \quad (54)$$

where  $\mathcal{F}_N v^\ell(k)$  denotes the (truncated)-Fourier coefficients of the discrete Fourier transform (DFT) of  $v^\ell(x)$ , computed based on the given  $s$  grid values in each direction. The maximal number of modes is set to  $M$ . Note that  $P_\ell(k) \in \mathbb{C}^{d_v \times d_v}$  is a complex Fourier multiplication matrix indexed by  $k \in \mathbb{Z}^d$ , and  $\mathcal{F}_N^{-1}$  denotes the inverse DFT.

For time-dependent problems, the lead time is conditioned into the FNO model at all levels by incorporating it inside the conditional instance norms of the model. Those normalization layers are applied before *each* Fourier layer.

### 6.5.1 C-FNO

We significantly enhanced the performance of the FNO model by incorporating *local convolution operations* in addition to the global convolutions performed in the original Fourier layer. Let  $\mathcal{C}_{m,d_v}$  be a discrete, local convolution operator applied with a kernel of size  $d_v \times d_v \times m^d$ , acting on the space of functions of  $v^\ell$ . The *modified* Fourier layer is then defined by

$$\tilde{v}^\ell(x) = \mathcal{C}_{3,d_v} \circ \sigma \circ \mathcal{C}_{5,d_v} \circ v^\ell(x), \quad (55)$$

$$v^{\ell+1}(x) = \sigma \left( W_\ell \cdot \tilde{v}^\ell(x) + \left( K_\ell \tilde{v}^\ell \right) (x) \right). \quad (56)$$

Thus, we alternate between global Fourier layers and local convolutions in this architecture.

### 6.5.2 UViT

For the UViT baseline, we use the same architecture as the UViT in the diffusion model. However, the learning objective differs since the baseline is deterministic. As a result, training this baseline is significantly different from training the diffusion model. Thus, the use of the UViT model can be considered as an *ablation* for the role of the diffusion training objective (35) in the performance of GenCFD.

## 6.6 Training and Test Protocols

GenCFD and the above-mentioned baselines are trained on all the datasets by sampling data  $\bar{u} \sim \mu_0$ , where the underlying probability distributions for each dataset are described below.

However, as mentioned in the Main Text, at *test time*, our goal is the statistical computation of fluid flow (4) with *Dirac* input distributions, i.e., the inputs (for simplicity, initial conditions) are given by

$$\hat{\mu}_0 = \delta_{\hat{u}}, \quad (57)$$

which implies that  $\bar{\mu}^\Delta = \delta_{\hat{u}^\Delta}$  in (11), with  $\hat{u}^\Delta \approx \hat{u}$ . Thus, we test the ability of our algorithm and the baselines to compute statistical solutions in the interesting case where the input is a Dirac and it is only the chaotic evolution of the flow that makes the conditional measure (11) *spread out*. To generate the ground truth with such Dirac initial conditions, we use an *ensemble perturbation* approach outlined below.

**Ensemble Perturbation Approach for Ground Truth Generation.** Given a distribution of inputs (say initial conditions)  $\bar{u} \sim \bar{\mu}$ , we approximate the ground truth distribution at time  $T = t$  by the empirical measure  $\hat{\mu}_t^\Delta$  induced by Monte Carlo samples on  $\bar{\mu}^\Delta$  which are propagated in time by a classical numerical method (See Main Text Fig. 1 (A) for an illustration). This strategy is used to approximate the conditional probability measure  $P_t^\Delta(du|\bar{u})$  as follows:

$$P_t^\Delta(du|\bar{u}) = \hat{\mu}_{\bar{u},t}^\Delta \quad \text{with} \quad \bar{\mu}_{\bar{u}}^\Delta \approx \bar{\mu}_{\bar{u}} = \delta_{\bar{u}}. \quad (58)$$

Due to the deterministic nature of the classical simulation, the initial distribution  $\bar{\mu}_{\bar{u}} = \delta_{\bar{u}}$  also needs to be approximated by another distribution  $\bar{\mu}_{\bar{u}}^{\Delta} \approx \bar{\mu}_{\bar{u}}$  with  $B_{\frac{\varepsilon}{2}}(\bar{u}) \subseteq \text{supp } \bar{\mu}_{\bar{u}}^{\Delta} \subseteq B_{\varepsilon}(\bar{u})$  for some  $\varepsilon > 0$ . This slight modification causes trajectories to diverge and approximate the underlying distribution. Note that this approximation can be made as accurate as desired by choosing a small enough value for  $\varepsilon$ . In general, we take  $\varepsilon \sim \Delta$  for the experiments performed in this paper.

Note that we now have two nested probability distributions. Firstly, the distribution of initial conditions  $\bar{\mu}$ , and secondly the approximations  $\bar{\mu}_{\bar{u}}^{\Delta}$  of  $\delta_{\bar{u}}$  for each concrete initial condition  $\bar{u}$ . The sampling is therefore also realized by two stages of Monte Carlo sampling. Note that  $|\text{supp } \bar{\mu}| \gg |\text{supp } \bar{\mu}_{\bar{u}}^{\Delta}|$ . This motivates the terminology of samples of  $\bar{\mu}$  being called *macro*-samples, while samples of  $\bar{\mu}_{\bar{u}}^{\Delta}$  approximating the Dirac-delta distribution are called *micro*-samples.

To summarize, we are interested in Monte Carlo samples of  $P_t(du|\bar{u})$  where  $\bar{u} \sim \bar{\mu}$ . The algorithm to draw them is as follows:

1. Draw *macro* samples  $\bar{u}^1, \bar{u}^2, \dots, \bar{u}^{M_{\text{macro}}} \sim \bar{\mu}$ ,
2. For each macro sample  $\bar{u}^i$  draw *micro* samples  $\bar{u}_{\bar{u}^i}^1, \bar{u}_{\bar{u}^i}^2, \dots, \bar{u}_{\bar{u}^i}^{M_{\text{micro}}} \sim \bar{\mu}_{\bar{u}^i}^{\Delta}$ ,
3. Evolve each sample  $\bar{u}_{\bar{u}^i}^j$  in time to get  $u_{\bar{u}^i}^j(t)$ ,
4. Approximate each  $P_t(du|\bar{u} = \bar{u}^i)$  by

$$P_t(du|\bar{u} = \bar{u}^i) \approx \frac{1}{M_{\text{micro}}} \sum_{j=1}^{M_{\text{micro}}} \delta_{u_{\bar{u}^i}^j(t)}. \quad (59)$$

This is the strategy that generates the ground truth distribution, with respect to which we test GenCFD and the baselines.

## 6.7 Datasets

In the Main Text, we have presented results with five challenging three-dimensional flow datasets, which we describe in detail below. Datasets are available in a Google cloud storage bucket [gs://gencfd](https://gencfd) under a CC BY 4.0 license.

### 6.7.1 Taylor–Green Vortex (TG)

We simulate the well-known Taylor–Green vortex [79] for the incompressible Navier–Stokes equations (5) in a probabilistic setting by adding small perturbations to the velocity field. The initial conditions are given by

$$\begin{aligned} \bar{u}_x(x, y, z) &= A \cos(1\pi x) \sin(2\pi y) \sin(2\pi z) + \varepsilon_x(x, y, z), \\ \bar{u}_y(x, y, z) &= B \sin(1\pi x) \cos(2\pi y) \sin(2\pi z) + \varepsilon_y(x, y, z), \\ \bar{u}_z(x, y, z) &= C \sin(1\pi x) \sin(2\pi y) \cos(2\pi z) + \varepsilon_z(x, y, z), \end{aligned} \quad (60)$$

where we choose  $A = 1$ ,  $B = -1$ ,  $C = 0$  to fulfill the incompressibility constraint. The perturbations  $\varepsilon_x$ ,  $\varepsilon_y$ , and  $\varepsilon_z$  are defined to be

$$\varepsilon_d(x, y, z) = \frac{1}{8} \sum_{(i,j,k) \in \{0,1\}^3} \delta_{d,i,j,k} \alpha_i(4\pi x) \alpha_j(4\pi y) \alpha_k(4\pi z), \quad \text{where } \alpha_i(x) = \begin{cases} \sin(x) & \text{if } i = 0, \\ \cos(x) & \text{if } i = 1, \end{cases} \quad (61)$$

where  $\delta_{d,i,j,k} \sim \mathcal{U}_{[-0.025, 0.025]}$ .

For simulating a Dirac distributed initial condition, we fix the values  $\delta_{d,i,j,k}$  and add a second perturbation of a similar form as  $\varepsilon_d$  to the flow field. The difference being that the frequency of the Fourier modes is doubled, and their amplitudes are chosen proportional to the mesh size.

Although the initial datum only contains a few large frequencies, the turbulent cascade into progressively smaller scales leads to the dynamic generation of higher frequencies, see Main Text Fig. 2 (A and B) for illustrations of the pointwise kinetic energy and vorticity intensity. Thus, the solution transitions from a laminar to a turbulent regime with time. We simulate this experiment with a spectral viscosity method implemented within the publicly available, state-of-the-art, highly optimized GPU-based **Azeban** code [64] at a spatial resolution of  $128 \times 128 \times 128$ .

Here, the underlying solution operator maps the initial data to (trajectories of) the solution at later times. In the experiments conducted, the total time duration was scaled to  $T = 2.0$ . We test the solution at  $T_{\text{test}} = 0.8$  and  $T_{\text{test}} = 2.0$ . The all-to-all training was performed using snapshots corresponding to the time points  $\{0, 0.4, 0.8, 1.2, 1.6, 2.0\}$ . There are 15 input-output pairs per trajectory in the training set.

### 6.7.2 Cylindrical Shear Flow (CSF)

The cylindrical shear flow for the incompressible Navier–Stokes equations (6) is heavily inspired by the flat vortex sheet experiment in [38] and is introduced as a 3D equivalent to the latter [64]. The initial conditions are given by

$$\begin{aligned} \bar{u}_x(x, y, z) &= \tanh\left(2\pi \frac{r - 0.25}{\rho}\right), \\ \bar{u}_y(x, y, z) &= 0, \\ \bar{u}_z(x, y, z) &= 0, \end{aligned} \quad (62)$$

where  $r^2 = (y - 0.5 + \sigma_y^j(x))^2 + (z - 0.5 + \sigma_z^j(x))^2$  and  $\rho$  is the smoothness parameter. We define the perturbations  $\sigma_y^j(x)$  and  $\sigma_z^j(x)$  in the following way: Let  $\alpha_k^y$  and  $\alpha_k^z$  be i.i.d. uniformly distributed on  $[0, 1]$  and let  $\beta_k^y$  and  $\beta_k^z$  be i.i.d. uniformly distributed on  $[0, 2\pi]$ . Then  $\sigma_y^j(x)$  and

$\sigma_z^j$  are given by

$$\begin{aligned}\sigma_y^j(x) &= \delta \sum_{k=1}^p \alpha_k^y \sin(2\pi kx - \beta_k^y), \\ \sigma_z^j(x) &= \delta \sum_{k=1}^p \alpha_k^z \sin(2\pi kx - \beta_k^z).\end{aligned}\tag{63}$$

These initial conditions are well defined in the limit  $\rho \rightarrow 0$  where the interface between the flow directions becomes discontinuous and are then equal to

$$\begin{aligned}\bar{u}_x(x, y, z) &= \begin{cases} -1 & \text{for } r \leq 0.25, \\ 1 & \text{otherwise,} \end{cases} \\ \bar{u}_y(x, y, z) &= 0, \\ \bar{u}_z(x, y, z) &= 0,\end{aligned}\tag{64}$$

where  $r$  is defined as above.

For simulating a Dirac distributed initial condition, we fix the values of  $\alpha_k^y$ ,  $\alpha_k^z$ ,  $\beta_k^y$ , and  $\beta_k^z$ . Then we extend the perturbation by an additional three modes with amplitude proportional to the mesh size.

We choose  $p = 10$  and  $\delta = 0$  as our default configuration. The initial shear flow is then evolved by numerically solving the three-dimensional incompressible Navier–Stokes equations with the spectral viscosity based **Azeban** code [64]. The resulting approximate solutions follow a very complex temporal evolution and contain a large range of small-scale eddies as seen in Main Text Fig. 1 (D). The dataset is also generated at the resolution of  $128^3$ .

Again, the underlying solution operator maps the initial velocity field to the velocity field at later times. In the experiments conducted, the total time duration was scaled to  $T = 1.0$ . The all-to-all training was performed using snapshots corresponding to the time points  $\{0, 0.25, 0.5, 0.75, 1.0\}$ . There are 10 input-output pairs per trajectory in the training set.

### 6.7.3 Cloud-Shock Interaction (CSI)

This dataset is the three-dimensional version of the well-known shock-bubble test case for compressible fluid flows [40, 48, 45] where a supersonic shock wave hits a high-density cloud (bubble) and leads to the excitation of shock waves while creating a zone of turbulent mixing in their wake.

For this experiment, we subdivide the domain  $[0, 1]^3$  into three subdomains. The initial condition is then initialized as a constant on each of these subdomains. We define

$$\begin{aligned}S &= \{(x, y) \in [0, 1]^3 \mid x \leq 0.05 + \sigma_x(y)\}, \\ C &= \{(x, y) \in [0, 1]^3 \mid r \leq 0.13 + \sigma_r(\text{atan2}(y - 0.5, z - 0.5))\}, \\ E &= [0, 1]^3 \setminus (S \cup C),\end{aligned}\tag{65}$$

where  $r^2 = (x - 0.25)^2 + (y - 0.5)^2 + (z - 0.5)^2$  and  $\sigma_x$ , and  $\sigma_r$  are defined as

$$\begin{aligned}\sigma_x(y) &= \frac{\delta}{\sum_{k=1}^p \alpha_k^x} \sum_{k=1}^p \alpha_k^x \cos(2\pi k(y + \beta_k^x)), \\ \sigma_r(\varphi) &= \frac{\delta}{\sum_{k=1}^p \alpha_k^r} \sum_{k=1}^p \alpha_k^r \cos(2\pi k(\varphi + \beta_k^r)).\end{aligned}\tag{66}$$

The parameters  $\alpha_k^x$ ,  $\alpha_k^r$ ,  $\beta_k^x$ , and  $\beta_k^r$  are all uniformly distributed in  $[0, 1]$ . Furthermore, for the in-distribution data, we use  $\delta = 0.06$  and  $p = 10$ . Combining all of this, the initial conditions for this experiment are given by

$$(\bar{\rho}, \bar{u}_x, \bar{u}_y, \bar{p}) = \begin{cases} (0.386859, 11.2536, 0, 167.345) & \text{if } (x, y, z) \in S, \\ (10, 0, 0, 1) & \text{if } (x, y, z) \in C, \\ (1, 0, 0, 1) & \text{if } (x, y, z) \in E. \end{cases}\tag{67}$$

For generating Dirac distributed initial conditions, the values of  $\alpha_k^x$ ,  $\alpha_k^r$ ,  $\beta_k^x$ , and  $\beta_k^r$  are fixed. Then the perturbations are extended by the next three higher modes with their amplitudes set to be proportional to the mesh size.

The dataset is simulated with the high-resolution finite volume GPU-optimized **Alsvinn** code of [45] at a resolution of  $256^3$ .

For this dataset, the solution operator maps the initial conditions (density, momenta and energy) to the solution at later times. In the experiments we conducted, the total time duration was scaled to  $T = 1.0$ . The all-to-all training was performed using snapshots corresponding to the time points  $\{0, 0.25, 0.5, 0.75, 1.0\}$ . There are 10 input-output pairs per trajectory in the training set.

#### 6.7.4 Nozzle Flow (NF)

This dataset is generated from a three-dimensional fluid flow through a nozzle geometry of diameter  $2l_c$  and length  $4l_c$  into a larger domain of diameter  $11l_c$  and length  $56l_c$  that is filled with the same medium. The injection with a randomized inflow velocity profile of maximal magnitude  $u_c$  generates a wall-bounded turbulent pipe flow at  $Re = 10000$  up to  $Re = 20000$  inside the nozzle, where  $Re := u_c l_c / \nu_c$ ,  $u_c = 1$  m/s to  $u_c = 2$  m/s,  $l_c = 1$  m, and  $\nu_c = 1.0 \times 10^{-4}$  m<sup>2</sup>/s. Consequently, in the larger domain a turbulent round jet is formed, see Main Text Fig. 3 (A). Turbulent jet flows appear in many engineering applications such as fluid mixing, combustion or acoustic control. The vortex generation method in the computational setup from [34] is modified such that random perturbations

$$u_{\text{vort}}(x, y, z, t) = \text{frac}(t) \frac{1}{2\pi} \sum_{i=1}^{n_{\text{vort}}} \Gamma_i \frac{\left(1 - \exp\left(-\frac{\| [x, y, z] - p_i \|^2}{2\sigma^2}\right)\right)}{\| [x, y, z] - p_i \|^2} ((p_i - [x, y, z]) \times d_{\text{in}})\tag{68}$$

are added to the inflow mean velocity

$$u_{\text{mean}}(x, y, z, t) = \text{frac}(t)[2.77u_c \left(1 - \left(\frac{1}{l_c} \left((x - 5.5l_c)^2 + (z - 5.5l_c)^2\right)^{\frac{1}{2}}\right)^{\frac{9}{8}}\right), 0, 0], \quad (69)$$

where

$$\Gamma_i = 4s_i \left( \frac{\pi}{6 \ln(3) - 9 \ln(2)} \frac{k_i A_{\text{in}}}{n_{\text{vort}}} \right)^{\frac{1}{2}} \quad (70)$$

and

$$k_i = \frac{3}{2} (|u_{\text{mean}}(p_i)| I_{\text{turb}})^2 \quad (71)$$

denote the circulations per vortex, and the turbulent kinetic energies, respectively, for  $i = 1, 2, \dots, n_{\text{vort}}$ , and  $\text{frac}(t) \in [0, 1]$  is polynomially increased to unity until  $t = 0.5$  s. Here,  $n_{\text{vort}} = 50$  is the number of vortices of size  $\sigma = 0.1l_c$  located at inlet positions

$$[0, r_i, \theta_i] = [0, l_c \left(\gamma_i^{(r)}\right)^{\frac{1}{2}}, 2\pi\gamma_i^{(\theta)}] \quad (72)$$

with the respective Cartesian locations  $p_i \in \mathbb{R}^3$  and signs  $s_i$ . The random parameters  $\gamma_i^{(r)}$  and  $\gamma_i^{(\theta)}$  are i.i.d. uniformly in  $[0, 1)$ , and  $s_i$  are i.i.d. discrete uniformly on  $\{-1, 1\}$ . Moreover,  $I_{\text{turb}} = 0.05$  is the turbulence intensity,  $A_{\text{in}}$  is the inlet area of the nozzle, and  $d_{\text{in}} \in \mathbb{R}^d$  is the normalized inflow direction. The mean velocity inlet profile (69) as well as the perturbation (68) are Langevin-combined [34] to synthetically produce a turbulent inflow velocity

$$u_{\text{in}}(x, y, z, t_n) := u_{\text{mean}}(x, y, z, t_n) + u_{\text{vort}}(x, y, z, t_n) - \frac{u_{\text{vort}}(x, y, z, t_n) \cdot \nabla u_{\text{in}}(x, y, z, t_{n-1})}{|\nabla u_{\text{in}}(x, y, z, t_{n-1})|} d_{\text{in}} \quad (73)$$

at a discrete timestep  $t_n \in N$ , where  $u_{\text{in}}(x, y, z, 0) = [0, 0, 0, 0]$ . We use a lattice Boltzmann method with an LES model to approximate the initial boundary value problem in the nozzle geometry that is based on the weakly compressible Navier–Stokes equations at a Mach number of  $Ma = 4.7 \times 10^{-3}$  with the velocity inlet condition (73), a constant pressure boundary at the outlet, and no-slip boundary conditions at the remaining cylinder walls. We simulate this experiment with the open-source highly parallel C++ library **OpenLB** [32, 35] that scales efficiently on hundreds of GPGPU nodes [33]. For the NF experiment, the domain is discretized with  $10.19 \times 10^6$  grid points at a resolution of  $124 \times 124 \times 663$  and we compute a time horizon until  $t = 130$  s with a step size  $\Delta t = 2.479 \cdot 10^{-4}$ .

Consequently, in this dataset, the solution operator maps the initial conditions and the boundary conditions (inflow velocity) to the velocity field at later times. In the experiments we conducted, the time horizon was rescaled and non-dimensionalized to  $T = 1.3$ , with the testing time set to  $T_{\text{test}} = 1.0$ . Note that the all-to-all training was performed using snapshots corresponding to the time points  $\{0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$ . Thus, there are 21 input-output pairs per trajectory in the training set.



### 6.7.5 Dry Convective Planetary Boundary Layer (CBL)

This test case describes the growth of a CBL as encountered during a summer day [76]. Forced at the surface by solar radiative heating and weak geostrophic winds [49], warm plumes ascend to the top of the boundary layer (*boundary layer height*  $z_i \approx 1$  km), balanced by larger scale downdrafts, resulting in turbulent dynamics spanning a wide range of scales. CBL dynamics are crucial to understanding the fundamental properties and sensitivities of the (strato) cumulus clouds that can form above a CBL as moisture is added to the simulation, which are in turn a major source of uncertainty in current climate projections [71].

In order to allow for larger time steps, an anelastic approximation to the fully compressible Navier–Stokes equations is adopted [53, 56], eliminating sound waves from the system. Given a hydrostatic reference state

$$\alpha_0 \frac{\partial p_0}{\partial x_3} = -g, \quad (74)$$

with reference profiles  $\alpha_0(x_3) = \rho_0(x_3)^{-1}$  for specific volume and  $p_0(x_3)$  for pressure, and gravity constant  $g = 9.80665 \text{ ms}^{-2}$ , density changes are neglected in thermodynamic and continuity equations. This leaves the dynamic pressure perturbation  $p'$  to be diagnosed by solving an elliptic equation at each time step.

The anelastic equations with velocities  $u_1, u_2, u_3$  and entropy  $s$  as prognostic equations are then given by

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{1}{\rho_0} \frac{\partial(\rho_0 u_i u_j)}{\partial x_i} &= -\frac{1}{\rho_0} \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} - \frac{\partial \alpha_0 p'}{\partial x_i} + b \delta_{13} - \epsilon_{ijk} \delta_{j3} f(u_k - U_{g,k}), \\ \frac{\partial s}{\partial t} + \frac{1}{\rho_0} \frac{\partial(\rho_0 u_i s)}{\partial x_i} &= -\frac{1}{\rho_0} \frac{\partial(\rho_0 \gamma_i)}{\partial x_i}, \\ \frac{\partial \rho_0 u_i}{\partial x_i} &= 0, \end{aligned} \quad (75)$$

within the domain  $D$  and the time span  $[0, T]$ , where we use standard conventions for summing, the Kronecker delta  $\delta_{ij}$ , and the Levi–Civita tensor  $\epsilon_{ijk}$ . The Coriolis parameter  $f = 0.376e - 4 \text{ s}^{-1}$  acts upon the difference to the large scale geostrophic wind  $U_g$ , and we refer to [53] for the definition of the buoyancy term  $b$ . As is characteristic for large eddy simulations (LES), which do not aim to resolve the smallest turbulent eddies, the effect of turbulent motion smaller than the grid size is modeled by the sub-grid scale (SGS) stresses  $\tau_{ij}$  for velocity and  $\gamma_i$  for entropy. The radiative forcing is prescribed as horizontally constant heat flux  $Q_*$  at the lower boundary.

We solve the anelastic equations with the open source Python and Cython based **PyCLES** library [56] at a resolution of  $128^3$  and a grid spacing of 40 m and 16 m horizontally and vertically, respectively. In agreement with previous findings [57], we omit the modeling of the SGS terms ( $\gamma_{s,i} = \tau_{ij} = 0$ ) and rely on the favorable dissipative properties of the WENO scheme which provide an *implicit large eddy simulation*.

The simulation is initialized with zero velocities and a horizontally constant profile of potential

temperature

$$\theta = T \left( \frac{p_0}{p} \right)^{\frac{R}{c_p}}, \quad (76)$$

given by a characteristic profile of the shape

$$\theta(x_3) = \begin{cases} T_g, & 0 < x_3 < z_a, \\ T_g + (x_3 - z_a)(\partial_3 \theta)_i, & z_a < x_3 < z_b, \\ T_g + (\partial_3 \theta)_i(z_a - z_b) + (x_3 - z_b)(\partial_3 \theta)_e, & \end{cases} \quad (77)$$

with dry constants  $R = 287.1 \text{ JK}^{-1}\text{kg}^{-1}$  and  $c_p = 1004.0 \text{ JK}^{-1}\text{kg}^{-1}$ , and reference pressure  $p_0$  at ground level. We fix the environmental profile, given by the lapse rate  $(\partial_3 \theta)_e = 0.003 \text{ K m}^{-1}$  and a fixed intercept, to match the original test case formulation [76], and define the boundary layer height to be  $z_i = (z_b - z_a)/2$ . This leaves as free parameters of the initial condition the boundary layer temperature  $T_g$ , the initial boundary layer height  $z_i$ , and the inversion lapse rate (sharpness)  $(\partial_3 \theta)_i$ . Together with the geostrophic wind  $U_g$  and the heat forcing  $Q_*$ , each realization of the CBL simulation is hence parameterized by five free parameters which we sample uniformly in the training distribution, as summarized in Table 1.

Based on this deterministic setup, convection is initialized by random perturbations at the lowest model levels. For the same initial profile of  $\theta_0$ , different realizations of the CBL are hence obtained by varying the random seed. For the experiments in this paper, all training samples were generated with the same random seed (and varying input parameters), since there is no dependence on the exact initial perturbation after the model spinup phase. In order to sample a Dirac measure, however, the random seed is varied while keeping the input parameters fixed.

The boundary layer height  $z_i$  is computed as in [76] by the *maximum gradient method* as the horizontal mean of the vertical location with the largest temperature gradient. Due to the constant  $Q_*$  forcing,  $z_i$  grows linearly in time. In Figure 21, we evaluate GenCFD against the numerical truth on horizontal statistics, including statistical moments of prognostic variables, the vertical temperature flux, and turbulent kinetic energy (TKE).

In the experiments conducted, the total duration of 7200 s in model time was scaled to  $T = 2.4$ . We test the solution at  $T_{\text{test}} = 2.4$ . The all-to-all training was performed using snapshots corresponding to the time points  $\{1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4\}$ . There are 21 input-output pairs per trajectory in the training set. During the testing phase, the model is conditioned on the time step  $t = 1.2$ , with the assumption that the spin-up phase has been completed by this point. Thus, the models aim to learn the statistics of the solution operator which maps initial conditions and parameters to solutions at later times.

## 6.8 Details of Models and Hyperparameters

Here, we describe the selection procedure for GenCFD and the baselines.

**GenCFD.** In all experiments, axial attention was applied at every layer of the UViT denoiser. Each of the tested models consistently used 4 axial attention blocks and 8 attention heads. The

**Table 1: Parameters in the convective boundary layer experiment.** Default values are as given in [76], the lower and upper bounds are used for uniform sampling of the training data. For testing with a Dirac distribution, the default values are used.

| Parameter              | Unit              | Type              | Lower bound | Default value | Upper bound |
|------------------------|-------------------|-------------------|-------------|---------------|-------------|
| $Q_*$                  | $\text{Kms}^{-1}$ | Forcing           | 0.1         | 0.24          | 0.3         |
| $U_g$                  | $\text{ms}^{-1}$  | Forcing           | 0.0         | 1.0           | 5.0         |
| $T_g$                  | K                 | Initial condition | 297.5       | 300.0         | 302.5       |
| $z_i$                  | m                 | Initial condition | 974.0       | 1024.0        | 1074.0      |
| $(\partial_3\theta)_i$ | $\text{Km}^{-1}$  | Initial condition | 0.03        | 0.08          | 0.15        |

UViT architecture (see Fig. 5) consisted of 3 downsampling layers, each with a downsampling ratio of 2. The intermediate channel dimension for the input and output spaces was set to 128, and the Fourier embedding dimension for physical (PDE) time and noise levels was also 128. In the CSF, TG, CSI, and NF experiments, the number of channels per downsampling layer followed the configuration (64, 128, 256). The resulting model is referred to as the *base* architecture. For the CBL benchmark, a smaller architecture, referred to as the *small* architecture, was used for improved memory efficiency and faster training. This version employed channels per layer in the sequence (48, 96, 192). Table 2 presents the sizes of the models used in our experiments. Note that the ground truth data is typically generated at a resolution that is higher than the resolution of the computational grid of the UViT denoiser (Table 2). Hence, the data is downsampled onto this computational grid using numerical downsampling.

**Table 2: Details of the GenCFD architectures used in the benchmarks.** The column *In/Out Ch.* corresponds to the number of input/output channels of the models.

| Benchmark | Model        | Size  | Resolution                | In/Out Ch. |
|-----------|--------------|-------|---------------------------|------------|
| CSF, TG   | <i>base</i>  | 70.2M | $64^3$                    | 3/3        |
| CSI       | <i>base</i>  | 70.2M | $64^3$                    | 4/4        |
| NF        | <i>base</i>  | 70.2M | $64 \times 64 \times 192$ | 4/3        |
| CBL       | <i>small</i> | 40.1M | $128^3$                   | 5/4        |

Moreover, in Table 3, we outline the experimental details for each dataset, including the number of training samples, batch sizes, gradient steps (and corresponding epochs), as well as the number of GPUs and their memory capacity used for training. Each model was trained for approximately 72 to 120 hours, except for the one used in the NF benchmark, which completed training in just 24 hours. It is worth noting that training times could be significantly reduced by utilizing multiple GPUs with larger memory.

**Table 3: Experimental details for the GenCFD models.** The first column indicates the benchmark, the second shows the number of trajectories (*Num. Traj.*) used for training, and the third lists the total number of training samples for all-to-all training (*Num. Sam.*). The fourth column specifies the batch size (*B. S.*), the fifth details the number of gradient steps (*Num. Grad. S.*) during training, and the sixth provides the approximate number of training epochs (*Epoch*). Finally, the seventh column notes the number of GPUs used for training along with their memory (*GPU (num:mem)*).

| Benchmark | Num. Traj. | Num. Sam. | B. S. | Num. Grad. S. | Epoch | GPU (num : mem) |
|-----------|------------|-----------|-------|---------------|-------|-----------------|
| TG        | 6.6K       | 99K       | 5     | 1.0M          | 50.5  | 1 : 24GB        |
| CSF       | 9.9K       | 99K       | 5     | 1.0M          | 50.5  | 1 : 24GB        |
| CSI       | 9.9K       | 99K       | 2     | 0.8M          | 16.2  | 1 : 24GB        |
| NF        | 9K         | 189K      | 4     | 0.1M          | 2.2   | 1 : 80GB        |
| CBL       | 7.5K       | 157.5K    | 4     | 0.25M         | 6.3   | 4 : 80GB        |

**Baselines.** The training and inference procedures differ between the baselines (FNO, C-FNO, and UViT) and the GenCFD. The GenCFD model was trained without using early stopping, whereas the deterministic models were trained until convergence using early stopping to prevent overfitting. In every benchmark, we observed that the training of deterministic models invariably collapsed to the mean of the output distribution. Even when the deterministic models were trained for extended durations and allowed to overfit the training set, the outcome remained unchanged. The deterministic models completed training in approximately 24 to 48 hours. Note that the same number of training samples was used as in the GenCFD trainings.

For each experiment and baseline (except for CBL, due to memory constraints), cross-validation was conducted using a random grid search over a range of hyperparameters – typically 10 to 20 for the FNO models and 4 to 10 for UViT, respectively. Since the deterministic UViT model shares the same architecture as GenCFD across all benchmarks, only the learning rate was varied during the hyperparameter sweeps.

## 6.9 Evaluation Metrics

Evaluation of the model’s performance employs a suite of metrics to quantify the fidelity and diversity of the generated samples relative to the ground truth distribution:

- **$L^2$ -norm error between the mean** of the ground truth ( $\mu_{\text{exact}}$ ) and the approximated distribution ( $\mu$ ):

$$e_{\mu} = \|\mu_{\text{exact}} - \mu\|_2. \quad (78)$$

- **$L^2$ -norm error between the standard deviation** of the ground truth ( $\sigma_{\text{exact}}$ ) and the

**Table 4: Details of the FNO architectures used in the benchmarks.** The  $d_v$  corresponds to the lifting dimension,  $L$  to the number of Fourier layers,  $M$  number of modes used in the Fourier layer,  $lr$  to the peak learning rate,  $B. S.$  to the batch size. The architectures are obtained using a random grid search over a range of hyperparameters (typically 10 to 20 configurations).

| Benchmark | $d_v$ | $L$ | $M$ | $lr$   | B. S. | Size  |
|-----------|-------|-----|-----|--------|-------|-------|
| TG        | 64    | 5   | 12  | 0.0001 | 2     | 42.1M |
| CSL       | 64    | 5   | 16  | 0.0001 | 5     | 95.2M |
| CSI       | 64    | 5   | 16  | 0.0001 | 2     | 95.2M |
| NF        | 64    | 5   | 12  | 0.0001 | 1     | 42.1M |
| CBL       | 48    | 4   | 16  | 0.0001 | 1     | 43.1M |

**Table 5: Details of the C-FNO architectures used in the benchmarks.** The  $d_v$  corresponds to the lifting dimension,  $L$  to the number of Fourier layers,  $M$  number of modes used in the Fourier layer,  $lr$  to the peak learning rate,  $B. S.$  to the batch size. The architectures are obtained using a random grid search over a range of hyperparameters (typically 10 to 20 configurations).

| Benchmark | $d_v$ | $L$ | $M$ | $lr$   | B. S. | Size   |
|-----------|-------|-----|-----|--------|-------|--------|
| TG        | 64    | 4   | 12  | 0.0001 | 2     | 40.0M  |
| CSL       | 64    | 4   | 16  | 0.0001 | 3     | 82.4M  |
| CSI       | 64    | 3   | 16  | 0.0001 | 2     | 62.1M  |
| NF        | 64    | 5   | 16  | 0.0001 | 1     | 102.7M |
| CBL       | 48    | 4   | 16  | 0.0001 | 1     | 46.9M  |

approximated distribution ( $\sigma$ ):

$$e_\sigma = \|\sigma_{\text{exact}} - \sigma\|_2. \quad (79)$$

The standard deviation error is normalized with respect to the ground truth norm.

- **Average 1-point Wasserstein distance** between the ground truth  $p_{\text{exact}}$  and the approximated distribution  $p$  (conditional and unconditional) computed over  $M$  spatial points:

$$W_s(p_{\text{exact}}, p) = \sum_{i=1}^M \left( \int_0^1 |F_{\text{exact}}^{-1}(u(x_i)) - F^{-1}(u(x_i))|^s dx \right)^{1/s}, \quad (80)$$

with  $F$  being the CDFs.

- **Continuous ranked probability score (CRPS):** Given an ensemble of predictions  $U = \{u\}_{m=1}^M$ ,  $x_m \sim p$ , and a single observation  $u_{\text{exact}}$  from the ground truth distributions  $u_{\text{exact}} \sim p_{\text{exact}}$ , the pointwise CRPS score is defined as

$$\text{CRPS}[U, u_{\text{exact}}] = \frac{1}{M} \sum_{m=1}^M \|u_m - u_{\text{exact}}\|_2^2 - \frac{1}{2M^2} \sum_{m=1}^M \sum_{j=1}^M \|u_m - u_j\|_2^2. \quad (81)$$

Given an ensemble of observations  $U_{\text{exact}} = \{u_{\text{exact}}\}_{n=1}^N$ , we can extend the definition of the CRPS as

$$\text{CRPS}[U, U_{\text{exact}}] = \frac{1}{N} \sum_{n=1}^N \text{CRPS}[U, u_{\text{exact},n}]. \quad (82)$$

It should be noted that  $\text{CRPS}[U, U_{\text{exact}}]$  is a function of the spatial coordinates  $x$ . To get a single global indicator of the ensembles' similarities, the (relative)  $L^2$ -norm of  $\text{CRPS}[U, U_{\text{exact}}]$  can be computed as

$$\text{CRPS}_G[U, U_{\text{exact}}] = \|\text{CRPS}[U, U_{\text{exact}}]\|_2^2. \quad (83)$$

Moreover, the CRPS (81) is also normalized with respect to the  $L^2$ -norm of the true observation  $u_{\text{exact}}$ .

**On Energy Spectra.** Given a flow field  $u : \mathbb{R}^d \rightarrow \mathbb{R}^d$  where  $d$  denotes the dimension of space, we denote its component-wise Fourier transform by  $\hat{u} : \mathbb{R}^d \rightarrow \mathbb{C}^d$ . The energy spectrum is then defined as

$$E_k = \frac{1}{2} \int_{|\xi|=k} \|\hat{u}(\xi)\|^2 d\xi. \quad (84)$$

As our solutions  $u^\Delta : \mathbb{T}^d \rightarrow \mathbb{R}^d$  lie on the  $d$ -dimensional torus  $\mathbb{T}^d$ , we make use of their discrete Fourier transforms  $\hat{u}_k^\Delta$  in order to compute the energy spectrum. Furthermore, we integrate over the ball in  $L^1$ , as that aligns with the computational grid. All in all, the energy spectra of our discrete solutions are computed as

$$E_k = \frac{\Delta^d}{2} \sum_{\|\xi\|_1=k} \|\hat{u}_\xi^\Delta\|^2. \quad (85)$$

## 6.10 Details on Toy Model #1 of the Main Text

In the toy problem mentioned in the Main Text, we aim to mimic essential aspects of the behavior of turbulent flows, in particular the sensitive dependence of outputs on small perturbations of the inputs. To this end, we fix  $\Delta = 1/N$  and consider a very simple one-dimensional model by setting  $u, \bar{u} \in \mathbb{R}$  and introducing a sequence of one-dimensional mappings

$$\mathcal{S}^\Delta(\bar{u}) = m(\bar{u}) + s_N(\bar{u}), \quad s_N(\bar{u}) := \Lambda(N\bar{u}), \quad (86)$$

with  $m : \mathbb{R} \mapsto \mathbb{R}$  being any *mean* function and  $\Lambda$  being the 1-periodic extension of the hat-function, with values  $\Lambda(0) = \Lambda(1) = -1$ ,  $\Lambda(1/2) = 1$ . The parameter  $\Delta = 1/N$  allows us to

control the input-sensitivity of the underlying map  $\mathcal{S}^\Delta$ , increasing sensitivity as  $\Delta \rightarrow 0$ . Fix the initial measure  $\bar{\mu} = \mathcal{U}[0, 1]$  to be the uniform measure on  $[0, 1]$ . Let  $\mu^\Delta = (\mathcal{S}^\Delta \times \text{Id})_{\#} \bar{\mu}$  denote the push-forward measure that we wish to approximate.

We observe from the Main Text Fig. 4 (A) how  $\mathcal{S}^\Delta$  becomes more and more oscillatory as  $\Delta \rightarrow 0$ . In particular, it does not seem possible to realize a deterministic limit. It is also easy to show that  $\text{Lip}(\mathcal{S}^\Delta) \sim 4N \rightarrow \infty$ . Nevertheless, we will later show that the  $\Delta \rightarrow 0$  limit is well-defined *statistically* and the resulting conditional distribution is given by the uniform distribution  $p(u | \bar{u}) = \mathcal{U}[m(\bar{u}) - 1, m(\bar{u}) + 1]$ , centered around the mean  $m$ .

Thus, this toy problem mimics several relevant features of turbulent fluid flow such as i) no deterministic limit under mesh refinement, ii) unstable behavior of the numerical approximation operator  $\mathcal{S}^\Delta$  in the limit, iii) the limit under mesh refinement is well-defined statistically and iv) the limit measure is not a Dirac but is *spread out*.

### 6.10.1 Numerical Results

We implemented toy problem #1, and trained both deterministic models and diffusion models for various values of the parameter  $\Delta > 0$ .

Our theoretical considerations assumed a bounded Lipschitz constant  $L^*$  as a mathematical proxy for the limitations in learning oscillatory functions. In practice, the approximation of neural networks is limited by (at least) three factors: (i) the available training data, (ii) the model capacity (architecture), (iii) the approximate optimization by stochastic gradient descent.

**Training Data.** We train all models with a total of  $N = 2048$  training samples, sampled uniformly on the interval  $[0, 1]$ . Since we focus on values of  $\Delta \geq 0.02 \gg 1/N$ , we expect the training samples to allow (in principle) for near-perfect interpolation of  $\mathcal{S}^\Delta$ .

**Architecture.** Our experiments are based on small models, all of which are chosen as vanilla dense, feedforward MLPs with ReLU activation. The deterministic model employs 2 hidden layers, and width 256, mapping a one-dimensional input (corresponding to  $\bar{u}$ ) to a one-dimensional output,

$$\bar{u} \mapsto \Psi_{\text{det}}(\bar{u}). \quad (87)$$

The diffusion model has depth 3 and width 512, mapping a three-dimensional input (corresponding to  $(u; \bar{u}, \sigma)$ ) to a one-dimensional output, i.e.

$$(\bar{u}, u, \sigma) \mapsto D(u; \bar{u}, \sigma). \quad (88)$$

From limited experimentation during the implementation, the qualitative results of the experiments are observed to be robust to changes in the hyperparameters of the networks. Our goal is to examine the qualitative behavior of the deterministic and diffusion models when  $\Delta \rightarrow 0$ , independently. No attempt is made to provide a quantitative comparison between the deterministic and diffusion models (which is probably meaningless for these toy problems).



**Training.** All models are implemented and trained in **PyTorch**. We use the Adam optimizer with the learning rate set to  $10^{-3}$ . Training is performed for a fixed number of epochs, for a maximum of 10000 epochs. The deterministic models are trained with MSE loss. For these one-dimensional, highly oscillatory toy problems, it is not always clear whether true stagnation of the training progress is observed. We therefore opt to illustrate not only the final results after 10000 epochs, but also the training progress. The observed difference between the deterministic and diffusion models during training is another interesting outcome of these toy problems.

**Illustration of Results.** To illustrate the trained deterministic models, we sample 2048 point in  $\bar{u}$ , and show a scatter plot of  $(\bar{u}, \Psi(\bar{u}))$ . Similarly, for the denoising models, we show a scatter plot as follows: We first sample 300 points uniformly in  $\bar{u}$ , and then generate 100 samples from the learned conditional distribution  $p(u | \bar{u})$  for each point in  $\bar{u}$ , giving a scatter plot of 30000 samples in total. The noise process is chosen as a variance-preserving process, as in [26]. The backward denoising process is run with 200 timesteps and a cosine noise schedule [50].

## 7 Detailed Theory

In this section, we present rigorous mathematical statements (and their proofs) that justify and expand on the theoretical discussion in the Main Text. Our aim is to explain, with rigorous mathematical analysis, the observations from our numerical experiments. In particular, we focus on explaining how i) diffusion models are able to learn the underlying probability distributions while deterministic ML baselines regress to mean fields and ii) diffusion models provide excellent spectral resolution (coverage) and are able to approximate small scales, right up to the smallest eddies in the data, while deterministic ML models have very poor spectral resolution.

### 7.1 Main results

**Characterization of Optimal Denoisers.** As the time-conditioning in (35) is not relevant for this theoretical discussion, we omit it and consider the *denoiser training objective* or Diffusion loss as

$$\mathcal{J}(D_\theta) = \mathbb{E}_{\bar{u} \sim \bar{p}} \mathbb{E}_{u|\bar{u}} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D_\theta(u + \eta; \bar{u}, \sigma) - u\|^2. \quad (89)$$

Hence, our aim is to remove noise from the noisy sample  $u_\sigma = u + \eta$ ,  $\eta \sim \mathcal{N}(0, \sigma^2)$ , conditioned on the input  $\bar{u}$ , during the training of the denoiser. It turns out (see Lemma 7.9) that we can explicitly characterize the optimal denoiser  $D_{\text{opt}} = \text{argmin}_\theta \mathcal{J}(D_\theta)$  for any noise level  $\sigma > 0$  by  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) = \mathbb{E}[u | (\bar{u}, u_\sigma)]$ . Therefore, if the noise process has ended up in a location  $u_\sigma$ , the optimal denoiser considers the distribution of the conditional random variable  $u$  given  $(\bar{u}, u_\sigma)$ , i.e. all possible origins  $u$  of the noise process conditioned on the input  $\bar{u}$  and the noised sample  $u_\sigma$ , and selects the most likely origin as the expected value in this distribution.

This key observation can be used to further characterize the optimal denoiser in the zero-noise ( $\sigma \rightarrow 0$ ) limit (see Proposition 7.11). In particular, we prove that, in this limit, the optimal denoiser  $D_{\text{opt}}(w; \bar{u}, \sigma = 0)$  evaluated at a point  $w$ , is identified with the closest point  $w^*$  in the support of  $p(u | \bar{u})$ , corresponding to a projection onto the data manifold. An immediate consequence of the identity  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) = \mathbb{E}[u | (\bar{u}, u_\sigma)]$  for  $\sigma > 0$  is the fact that

**Proposition 7.1.** *If  $u | \bar{u}$  is in fact deterministic, i.e.  $u = \mathcal{F}(\bar{u})$ , then  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) \equiv \mathcal{F}(\bar{u})$  for all  $\bar{u}, \sigma$ .*

The above proposition makes it clear that, if a conditional distribution is generated by an underlying deterministic map  $\mathcal{F}$ , then the optimal denoiser will simply collapse to this deterministic map. In practice, the diffusion model is trained on

$$\mathcal{J}^\Delta(D_\theta) = \mathbb{E}_{\bar{u} \sim \bar{p}} \mathbb{E}_{u^\Delta | \bar{u}} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D_\theta(u^\Delta + \eta; \bar{u}, \sigma) - u^\Delta\|^2, \quad (90)$$

with  $u^\Delta | \bar{u} = \mathcal{S}^\Delta(\bar{u})$  obtained from a numerical solver. Given that (90) stems from the *deterministic* approximate solution operator  $\mathcal{S}^\Delta$ , the optimal denoiser should be concentrated around  $\mathcal{S}^\Delta(\bar{u})$  for any  $\Delta > 0$ . Hence, there should be no difference between using a denoiser

training objective (90) and a purely deterministic loss,

$$\mathcal{J}_{\text{det}}^{\Delta}(\Psi_{\theta}) = \mathbb{E}_{\bar{u} \sim \bar{p}} \|\Psi_{\theta}(\bar{u}) - \mathcal{S}^{\Delta}(\bar{u})\|^2. \quad (91)$$

In other words, this result suggests *there should be very little difference between the simulations carried out using the ML baselines and our proposed conditional-diffusion model, seemingly contradicting our experimental observations*. Therefore, we need to formulate a more refined analysis in order to resolve this apparent contradiction.

**Input Sensitivity.** To this end, we recall that, for fluid flows, the approximate solution operator outputs solutions which contain energetic eddies across a very large range of scales, exhibiting chaotic dynamics. Consequently, the behavior of  $\mathcal{S}^{\Delta}$  asymptotically as  $\Delta \rightarrow 0$  is very oscillatory and unstable [16, 39]. It is precisely this lack of stability in the  $\Delta \rightarrow 0$  limit that could prevent us from realizing an *unconstrained* optimal denoiser within the class of neural networks. Our refined theoretical analysis is based on the assumption of a *mismatch in input sensitivity*: the underlying solution operator  $\mathcal{S}^{\Delta}$  is extremely sensitive to small perturbations  $\delta\bar{u}$ . Even perturbations of a small size  $\|\delta\bar{u}\|_{\mathcal{X}} \sim \tilde{\epsilon}$  can entail

$$\|\mathcal{S}^{\Delta}(\bar{u} + \delta\bar{u}) - \mathcal{S}^{\Delta}(\bar{u})\|_{\mathcal{Y}} \gg 1. \quad (92)$$

This is precisely the *sensitivity hypothesis* of the underlying operators of the main text.

In contrast, we argue that our trained neural network model could not be able to match this input sensitivity; i.e. a sufficiently small input perturbation  $\|\delta\bar{u}\|_{\mathcal{X}} \sim \tilde{\epsilon} \ll 1$  only leads to a small output perturbation,

$$\|\Psi_{\theta}(\bar{u} + \delta\bar{u}) - \Psi_{\theta}(\bar{u})\|_{\mathcal{Y}} \ll 1 \quad \text{and} \quad \|D_{\theta}(u_{\sigma}; \bar{u} + \delta\bar{u}, \sigma) - D_{\theta}(u_{\sigma}; \bar{u}, \sigma)\|_{\mathcal{Y}} \ll 1. \quad (93)$$

This is the *insensitivity hypothesis for neural networks*, that is discussed in the main text. Why does this hypothesis hold? Why are neural networks *insensitive to very small perturbations in inputs*?

A possible answer lies in the notion that *neural networks learn and generalize well at the edge of chaos* [11, 84]. In this framework based on statistical physics, the forward pass of neural networks is viewed as a dynamical system. The underlying principle states that optimal computational capability of neural networks (or other systems, including the brain) emerges when the dynamical system is at a critical point between order and chaos. A relevant *measure of chaos* is the Lyapunov exponent of the input-to-output map, defined as  $\gamma \approx \frac{1}{T} \log(|\delta x_T|/|\delta x_0|)$  [11, eq. (S27)], defined in terms of a temporal parameter  $T$  (which can be the depth for deep networks or the number of rollout steps for autoregressive neural network predictions) and the quotient of the magnitude of output perturbations  $|\delta x_T|$  versus the magnitude of (small) input perturbations  $|\delta x_0|$ . This quantity is equivalent to the Lipschitz constant  $\text{Lip}(\Psi_{\theta}) \approx |\delta x_T|/|\delta x_0|$ . Based on both theoretical considerations and extensive empirical evidence, it has been demonstrated that neural networks maximize their performance and generalization capability when  $\gamma \approx 0$  [11], i.e. when  $\text{Lip}(\Psi_{\theta}) \approx 1$ . Thus, this *edge of chaos* principle leads to an obvious tension

(or indeed contradiction) between the opposing goals of keeping Lipschitz constants of neural networks *bounded of order 1* to ensure optimal performance, and the requirement of having *exponentially large* Lipschitz constants as would be required to fit the exponential input-sensitivity of turbulent/chaotic flows.

A second and related motivation for the assumption of insensitivity of neural networks is by viewing it as a high-dimensional analogue of the well-known spectral bias of neural networks [59]. As observed for function regression in one dimension, neural networks are biased against fitting high-order Fourier modes. In that context, the lack of regularity of an underlying mapping, i.e. its input sensitivity, is encoded by a slow decay of the Fourier spectrum; fitting a highly input-sensitive function by a neural network necessitates the accurate approximation of high-order Fourier modes, and hence overcoming this observed spectral bias.

Finally, neural networks are trained by variants of stochastic gradient descent algorithms that require that the underlying gradients are well-behaved, i.e., are bounded and of order one. Otherwise, the well-known *exploding and vanishing gradient problem* [52] will be encountered and will lead to a failure of neural network training. Clearly, these gradients are related to the Lipschitz constants of the neural network  $\Psi_\theta$  with respect to the inputs. Hence, ensuring well-bounded gradients, necessary for neural network training, also adds weight to the insensitivity hypothesis that we propose here.

Thus, we argue that our assumption of a mismatch in input-sensitivity is natural, given the underlying chaotic dynamics of fluid flows or any sensitive map. While leaving a more rigorous investigation of this assumption for future work, we posit it here as a postulate, based on which theoretical consequences are to be derived below. This will allow us to explain several of our empirical observations.

**Lipschitz Continuity Quantifies Input Sensitivity.** The informal inequality (92) implies that discretized solution operators of interest have very large Lipschitz constants, i.e. that  $\text{Lip}(\mathcal{S}^\Delta) \gg 1$  for small  $\Delta$ . To capture this in our mathematical analysis, we argue that these Lipschitz constants are so large, that for all practical purposes the relevant regime is captured by assuming  $\text{Lip}(\mathcal{S}^\Delta) \rightarrow \infty$ , as  $\Delta \rightarrow 0$ ; in fact, for PDEs such as the Navier–Stokes equations, whether  $\text{Lip}(\mathcal{S}^\Delta)$  remains finite or not is a long-standing open problem, and the potential ill-posedness in the limit  $\Delta \rightarrow 0$  is a realistic possibility. In contrast, we surmise that training by stochastic gradient descent tends to bias neural networks towards stability and away from highly oscillatory multiscale mappings. We mathematically formulate this hypothesis as follows:

**Hypothesis 7.2.** *Practical minimization of the denoiser objective (89) is only possible within a subclass of mappings  $D_\theta$  satisfying a Lipschitz bound  $\text{Lip}(D_\theta) \leq L^*$ , for some cut-off  $L^* \geq 1$ .*

In fact, the Lipschitz bound is one possible mathematical requirement for ruling out wild oscillations and can be replaced by other equivalent criteria such as bounds on total variation or conditions on band-limited approximations [2]. Under this hypothesis, it is straightforward to show that *constrained minimization* of the deterministic training objective (91), within the model

class specified by hypothesis 7.2, *cannot approximate the true minimizer*. This merely reflects the fact that  $\text{Lip}(\mathcal{S}^\Delta) \gg L^*$ , as  $\Delta \rightarrow 0$ , and hence the optimal  $D_{\text{opt}}^\Delta = \mathcal{S}^\Delta$  (cp. Proposition 7.1) has a divergent Lipschitz constant in this limit. In particular, the unconstrained minimizer is unstable and cannot be approximated under Hypothesis 7.2, highlighting the potential role of instabilities and multiscale structure of the underlying operators in hindering the success of deterministic approximations of the solution operator of turbulent flows by neural networks.

**Statistical Computation with Diffusion Models is still Tractable.** On the other hand, how can training of the denoiser (89) within the model class of Hypothesis 7.2 lead to an accurate statistical computation of fluid flows? A positive answer is provided by the following proposition (see p. 57 for a proof):

**Proposition 7.3.** *Let  $\mu, \mu^\Delta$  be two probability measures with bounded support on  $\{|u| \leq M\}$ . Assume that the optimal conditional denoiser  $D_{\text{opt}}(u; \bar{u}, \sigma)$  for  $\mu$  is  $L^*$ -Lipschitz continuous for some  $L^* \geq 1$ . Let  $D^\Delta$  be the optimal constrained denoiser  $D^\Delta$  for  $\mu^\Delta$ ,*

$$D^\Delta(u; \bar{u}, \sigma) = \underset{\text{Lip}(D_\theta) \leq L^*}{\text{argmin}} \mathcal{J}^\Delta(D_\theta, \sigma). \quad (94)$$

*Then, we have*

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma)\|^2 \leq CL^* W_1(\mu^\Delta, \mu), \quad \forall \sigma > 0, \quad (95)$$

*with constant  $C$  depending on  $M$ , but otherwise independent of  $\mu^\Delta$ ,  $\mu$ , and independent of  $L^*$  and  $\sigma$ .*

As the whole premise of statistical computation of fluids, backed up by the theory of statistical solutions [14, 39], rests on the observation that  $W_1(\mu^\Delta, \mu) \rightarrow 0$  as  $\Delta \rightarrow 0$  [39, 16, 64], we see from (95) that the *constrained* denoiser achieves an almost optimal loss for  $\mu$ , even though in this setting, we may have  $\text{Lip}(D_{\text{opt}}^\Delta) = \text{Lip}(\mathcal{S}^\Delta) \rightarrow \infty$ , and  $D_\theta^\Delta$  will not be able to approximate the true optimizer  $D_{\text{opt}}^\Delta(u; \bar{u}, \sigma) = \mathcal{S}^\Delta(\bar{u})$  at any training resolution  $\Delta \ll 1$ .

Hence, Proposition 7.3 reveals the *surprising mechanism* through which a diffusion model can leverage the highly unstable and multiscale nature of the underlying operator (modeling fluid flow for instance) to enable accurate statistical computation even though deterministic approximation in this context is not tractable.

The key assumption in Proposition 7.3 is the Lipschitz continuity of the optimal denoiser. Due to our current lack of mathematical understanding of the fine properties of statistical solutions, an end-to-end rigorous proof guaranteeing this property for fluid flows remains out of reach with existing mathematical tools. Furthermore, a highly technical proof would not necessarily shed further light on the fundamental mechanisms through which diffusion models work. Instead, we here choose another approach and present two *solvable toy models* which mimic relevant aspects of the behavior of turbulent fluids while being analytically tractable. These problems shed further light on the fundamental difficulties encountered by deterministic models, and illustrate how such difficulties can be overcome by probabilistic diffusion models.

## 7.2 Toy Model #1: Illustrating the Consequences of Input Sensitivity Mismatch

We recall that toy model #1 is a one-dimensional model which mimics essential aspects of the behavior of turbulent fluid flows (cp. Section 6.10). At the same time, it is analytically tractable and allows for a rigorous mathematical analysis that we describe here. Recall that we fix  $\Delta = 1/N$  and that, for  $u, \bar{u} \in \mathbb{R}$ , we have introduced a sequence of one-dimensional mappings,

$$\mathcal{S}^\Delta(\bar{u}) = m(\bar{u}) + s_N(\bar{u}), \quad s_N(\bar{u}) := \Lambda(N\bar{u}),$$

where  $m : \mathbb{R} \mapsto \mathbb{R}$  is any *mean* function and  $\Lambda$  is a 1-periodic hat-function. We fix the initial measure  $\bar{\mu} = \mathcal{U}([0, 1])$  to be the uniform measure on  $[0, 1]$ . We observe from Figure 32 how  $\mathcal{S}^\Delta$  becomes more and more oscillatory as  $\Delta \rightarrow 0$ . In particular, it does not seem possible to realize a deterministic limit. It is also easy to show that  $\text{Lip}(\mathcal{S}^\Delta) \sim 4N \rightarrow \infty$ .

We now study  $L^*$ -Lipschitz minimizers of the deterministic loss

$$\mathcal{J}_{\text{det}}^\Delta(\Psi_\theta) = \mathbb{E}_{\bar{u} \sim \bar{\mu}} |\Psi_\theta(\bar{u}) - \mathcal{S}^\Delta(\bar{u})|^2, \quad (96)$$

and the conditional diffusion training objective

$$\mathcal{J}^\Delta(D_\theta) = \mathbb{E}_{\bar{u} \sim p_{\text{prior}}} \mathbb{E}_{u^\Delta | \bar{u}} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} |D_\theta(u^\Delta + \eta; \bar{u}, \sigma) - u^\Delta|^2, \quad (97)$$

where  $u^\Delta | \bar{u} = \mathcal{S}^\Delta(\bar{u})$ . The sensitivity hypothesis in the Main Text suggests that, for some sensitivity scale  $\tilde{\epsilon} > 0$ , and  $\delta \bar{u} \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}])$ , the Lipschitz optimizer of  $\mathcal{J}_{\text{det}}^\Delta$  satisfies  $\Psi^\Delta(\bar{u}) \approx \mathbb{E}_{\delta \bar{u}} [\mathcal{S}^\Delta(\bar{u} + \delta \bar{u})]$ , while the constrained optimizer of  $\mathcal{J}^\Delta$  is approximately equal to the optimal denoiser for  $\text{Law}_{\delta \bar{u}} [\mathcal{S}^\Delta(\bar{u} + \delta \bar{u})]$ . Our goal is to make this intuition rigorous, via asymptotic analysis as  $\Delta \rightarrow 0$ .

**Remark 7.4** (Leading-order Analysis). *With the notation and assumptions above, we have to leading order in  $\tilde{\epsilon}$ ,*

$$\begin{aligned} \mathcal{S}^\Delta(\bar{u} + \delta \bar{u}) &= \mathcal{S}^\Delta(\bar{u}) + O(N\tilde{\epsilon}), \\ m(\bar{u} + \delta \bar{u}) &= m(\bar{u}) + O(\text{Lip}(m)\tilde{\epsilon}), \\ \Psi_\theta(\bar{u} + \delta \bar{u}) &= \Psi_\theta(\bar{u}) + O(L^*\tilde{\epsilon}). \end{aligned}$$

*We will be interested in the regime  $\text{Lip}(m) \sim L^* \ll N = \Delta^{-1}$ , where the gap  $L^* \ll \Delta^{-1}$  corresponds to a sensitivity mismatch between  $\Psi_\theta$  and  $\mathcal{S}^\Delta$ . Our main observation is that for  $\Delta \ll \tilde{\epsilon} \ll 1/L^*$ , the remainder terms for  $\Psi_\theta$  and  $m$  can be neglected, but this is clearly not admissible for  $\mathcal{S}^\Delta$ . As a consequence,  $\Psi_\theta$  cannot accurately capture the variation of  $\mathcal{S}^\Delta$  at input scale  $\tilde{\epsilon}$ . To enable rigorous analysis while capturing this relevant regime, we will fix  $L^*, \tilde{\epsilon}$  and consider the asymptotic limit  $\Delta \rightarrow 0$ , in the following.*

**Deterministic Models Collapse to the Mean.** We denote the constrained optimizer of the deterministic problem formulation by,

$$\Psi^\Delta := \underset{\text{Lip}(\Psi_\theta) \leq L^*}{\text{argmin}} \mathcal{J}_{\text{det}}^\Delta(\Psi_\theta). \quad (98)$$

We note that the underlying map  $\mathcal{S}^\Delta$  has the smallest length-scale  $\Delta$ , whereas the smallest length scale of  $\Psi^\Delta$  is uniformly bounded due to the imposed Lipschitz bound. Thus, in the limit  $\Delta \rightarrow 0$ , the scale separation between the approximated maps  $\mathcal{S}^\Delta$  and the approximants  $\Psi^\Delta$  increases arbitrarily. As argued in Remark 7.4, it is in this limit that we can expect our leading-order analysis of the Main Text to be rigorously justified.

We now assume that the mean function  $m(\bar{u})$  is  $L^*$ -Lipschitz. We fix a (arbitrary) constant  $\tilde{\epsilon} > 0$ . The discussion in the Main Text is based on the approximate identity  $\Psi_\theta(\bar{u} + \delta\bar{u}) \approx \Psi_\theta(\bar{u})$  for  $|\delta\bar{u}| \leq \tilde{\epsilon}$ . Since we assume that  $\Psi_\theta(\bar{u})$  and  $m(\bar{u})$  obey the same Lipschitz bound, we also expect that  $m(\bar{u} + \delta\bar{u}) \approx m(\bar{u})$  to the same accuracy, and hence

$$\mathcal{S}^\Delta(\bar{u} + \delta\bar{u}) = m(\bar{u} + \delta\bar{u}) + s_N(\bar{u} + \delta\bar{u}) \approx m(\bar{u}) + s_N(\bar{u} + \delta\bar{u}).$$

This motivates the following definition:

$$\tilde{\Psi}^\Delta(\bar{u}) := m(\bar{u}) + \mathbb{E}_{\delta\bar{u}} [s_N(\bar{u} + \delta\bar{u})], \quad \delta\bar{u} \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]). \quad (99)$$

Thus, we have  $\tilde{\Psi}^\Delta(\bar{u}) \approx \mathbb{E}_{\delta\bar{u}} [\mathcal{S}^\Delta(\bar{u} + \delta\bar{u})] \approx \Psi^\Delta(\bar{u})$ , under the assumptions of the Main Text. We next confirm this intuition by showing that  $\Psi^\Delta$  and  $\tilde{\Psi}^\Delta$  are asymptotically equivalent, as  $\Delta \rightarrow 0$ .

**Proposition 7.5.** *With the definitions above, we have*

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{\bar{u} \sim \bar{\mu}} |\Psi^\Delta(\bar{u}) - \tilde{\Psi}^\Delta(\bar{u})|^2 = 0, \quad \text{and} \quad \lim_{\Delta \rightarrow 0} \mathbb{E}_{\bar{u} \sim \bar{\mu}} |\Psi^\Delta(\bar{u}) - m(\bar{u})|^2 = 0. \quad (100)$$

The last proposition rigorously justifies the approximate identity  $\Psi^\Delta(\bar{u}) \approx \mathbb{E}_{\delta\bar{u}} [\mathcal{S}^\Delta(\bar{u} + \delta\bar{u})]$ , provided  $\tilde{\epsilon}$  is chosen sufficiently small so that  $m(\bar{u} + \delta\bar{u}) \approx m(\bar{u})$ , and shows that the optimal constrained model  $\Psi^\Delta$  collapses to the mean, in the limit  $\Delta \rightarrow 0$ .

**Probabilistic Models Predict Uncertainty.** We next consider the probabilistic problem formulation of conditional diffusion models. We denote by

$$D^\Delta(u_\sigma; \bar{u}, \sigma) := \underset{\text{Lip}(D_\theta) \leq L^*}{\operatorname{argmin}} \mathcal{J}^\Delta(D_\theta), \quad (101)$$

the optimal constrained denoiser for  $\mu^\Delta(du | \bar{u}) = \delta(u - \mathcal{S}^\Delta(\bar{u}))$  and with  $\bar{u} \sim \mathcal{U}([0, 1])$ . We again assume that  $m(\bar{u})$  is Lipschitz continuous. Again, under the assumptions of the Main Text, we then have  $\mathcal{S}^\Delta(\bar{u} + \delta\bar{u}) \approx m(\bar{u}) + s_N(\bar{u} + \delta\bar{u})$ . We thus consider the conditional probability,

$$\nu^\Delta(du | \bar{u}) := \text{Law}_{\delta\bar{u}} [m(\bar{u}) + s_N(\bar{u} + \delta\bar{u})], \quad \delta\bar{u} \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]), \quad (102)$$

for which  $\nu^\Delta(du | \bar{u}) \approx \text{Law}_{\delta\bar{u}} [\mathcal{S}^\Delta(\bar{u} + \delta\bar{u})]$ , consistent with the discussion in the Main Text. We denote the optimal (unconstrained) denoiser of  $\nu^\Delta(du | \bar{u})$  by,

$$\tilde{D}^\Delta(u_\sigma; \bar{u}, \sigma) := \underset{D}{\operatorname{argmin}} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \mathbb{E}_{u \sim \nu^\Delta(\cdot | \bar{u})} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D(u + \eta; \bar{u}, \sigma) - u\|^2. \quad (103)$$



The difference between  $D^\Delta$  and  $\tilde{D}^\Delta$  is that  $u$  is sampled from  $\mu^\Delta(du|\bar{u})$  and  $\nu^\Delta(du|\bar{u})$ , respectively. In addition,  $D^\Delta$  is a *constrained* minimizer with  $\text{Lip}(D^\Delta) \leq L^*$  imposed, whereas  $\tilde{D}^\Delta$  is an *unconstrained* minimizer.

It turns out that  $\nu^\Delta$  is asymptotically equivalent to a simpler measure  $\mu$ , as  $\Delta \rightarrow 0$ : We thus finally define  $\mu \in \text{Prob}(\mathbb{R} \times [0, 1])$  as the uniform measure on

$$\mathcal{I}(m) := \{(u, \bar{u}) \in \mathbb{R} \times [0, 1] \mid u \in [m(\bar{u}) - 1, m(\bar{u}) + 1]\}, \quad (104)$$

so that  $\mu(du|\bar{u}) = \mathcal{U}([m(\bar{u}) - 1, m(\bar{u}) + 1])$ . It will be shown in Lemma 7.15 that  $\nu^\Delta \rightarrow \mu$ .

The following result shows that the optimal constrained denoiser for  $\mu^\Delta = \delta(u - \mathcal{S}^\Delta(\bar{u}))$  is asymptotically equivalent to the optimal (unconstrained) denoiser for  $\nu^\Delta(du|\bar{u}) = \text{Law}_{\delta\bar{u}}[m(\bar{u}) + s_N(\bar{u} + \delta\bar{u})] \approx \text{Law}_{\delta\bar{u}}[\mathcal{S}^\Delta(\bar{u} + \delta\bar{u})]$ . This rigorously justifies the conclusion drawn from the sensitivity hypothesis of the Main Text, in the regime  $\Delta \ll 1/L^*$ .

**Proposition 7.6.** *With the definitions above, and for  $L^*$  a constant sufficiently large depending only on the Lipschitz constant of  $m(\bar{u})$ , we have*

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^\Delta(u + \eta; \bar{u}, \sigma) - \tilde{D}^\Delta(u + \eta; \bar{u}, \sigma) \right\|^2 = 0, \quad (105)$$

uniformly in  $\sigma > 0$ .

### 7.3 Toy Model #2: Illustrating Spectral Accuracy of Diffusion Models

From the experimental results in the Main Text (Fig. 2 (F)), we observed that neural networks trained to minimize least square errors have a very small effective spectrum. The general ideas that went into the toy model of the last section can also be used to gain intuition regarding the success of diffusion models in reproducing correct energy spectra. The hypothesis is again that deterministically trained models cannot capture highly oscillatory behavior, causing them to collapse to the mean in the oscillatory limit.

#### 7.3.1 Motivation

We now consider a (translation-equivariant) equation like the Navier–Stokes equations. Assume we have an accurate approximation  $\Psi \approx \mathcal{S}$  of the corresponding solution operator (or of  $\mathcal{S}^\Delta$  for small  $\Delta > 0$ ). For a given input  $\bar{u}$ , let  $u = \mathcal{S}(\bar{u})$  be the corresponding solution. We define a parametric path in the input function space,  $h \mapsto \bar{u}_h := \bar{u}(\cdot + h)$ . We note that, by translation-equivariance of  $\mathcal{S}$ , we have  $\mathcal{S}(\bar{u}_h) = u_h := u(\cdot + h)$ . If  $\Psi$  is a good approximation of  $\mathcal{S}$ , then by assumption, we have

$$\Psi(\bar{u}_h) \approx u_h, \quad \forall h,$$

and for any continuous linear functional  $\ell : L^2(D) \rightarrow \mathbb{R}$ , we also have

$$\langle \ell, \Psi(\bar{u}_h) \rangle \approx \langle \ell, u_h \rangle, \quad \forall h.$$

But now, consider the Fourier expansion:

$$u(x) = \sum_{k \in \mathbb{Z}^d} \hat{u}(k) e^{ikx}$$

and define  $\ell$  as the projection onto the  $k$ -th Fourier mode. Then

$$h \mapsto \langle \ell, \Psi(\bar{u}_h) \rangle \approx \langle \ell, \mathcal{S}(\bar{u}_h) \rangle = \hat{u}(k) e^{ikh}.$$

If  $\|\Psi(\bar{u}_h) - \mathcal{S}(\bar{u}_h)\|_{L^2} \leq \epsilon \ll 1$ , then clearly, we must also have

$$\left| \langle \ell, \Psi(\bar{u}_h) \rangle - \hat{u}(k) e^{ikh} \right| \leq \|\Psi(\bar{u}_h) - \mathcal{S}(\bar{u}_h)\|_{L^2} \leq \epsilon.$$

There are only two options for this upper bound to hold: Either  $|\hat{u}(k)| \lesssim \epsilon$  is inherently small (in which case  $\langle \ell, \Psi(\bar{u}_h) \rangle \approx 0$  would do), or  $|\hat{u}(k)| \gg \epsilon$  is not small, in which case  $h \mapsto \langle \ell, \Psi(\bar{u}_h) \rangle / \hat{u}(k)$  must be a good approximation of the oscillatory function  $h \mapsto e^{ikh}$ . Just as in the previous section, if  $\Psi$  is constrained to be non-oscillatory, then it is impossible to achieve a highly accurate approximation of  $h \mapsto e^{ikh}$  for large  $k$ . Instead, we expect to see a collapse to the mean in the limit  $|k| \rightarrow \infty$ .

### 7.3.2 Model

The discussion above shows that for relevant solution operators  $\mathcal{S}$  in fluid dynamics, with solutions exhibiting slowly decaying Fourier spectrum, the mapping

$$\bar{u}_h \mapsto \mathcal{S}(\bar{u}_h),$$

can be considered oscillatory in a sense related to Fourier analysis. The following toy model replaces the dependence on the input function  $\bar{u}_h$  by a dependence on  $h \in [0, 1]$ , resulting in the following oscillatory model with parameter  $k \in \mathbb{N}$ :

$$h \mapsto \mathcal{S}^{(k)}(h), \quad \mathcal{S}^{(k)}(h) := (\cos(2\pi kh), \sin(2\pi kh)) \in \mathbb{R}^2.$$

This toy problem captures a core difficulty in accurately reproducing energy spectra for fluid flows: namely, Fourier modes of the solution at high wavenumber ( $k \gg 1$ ) are increasingly sensitive to small perturbations of the initial data. The initial data is here replaced by  $h \in [0, 1]$ . In comparison to the previous toy problem, an additional feature of this toy model lies in the fact that the outputs are constrained to lie on the unit circle, no matter what value the wavenumber  $k$  assumes. This *toy constraint* is designed to mimic real physical laws (constraints) such as energy balance in fluid flows, and represents a stable statistical property (akin to the robustness of energy spectra in fluid flows).

For this toy problem, our analysis suggests that a deterministically trained model will collapse to  $(0, 0)$  for large  $k$ . In contrast, the analysis presented below suggests that a practically trained conditional diffusion model will instead produce a denoiser,

$$D_\theta(u; h, \sigma) \approx \begin{cases} (\cos(2\pi kh), \sin(2\pi kh)), & (k \sim 1), \\ u/|u|, & (k \gg 1). \end{cases}$$

The conditional probability distribution corresponding to this denoiser is deterministic for small/moderate  $k$ , but it is non-deterministic for large  $k$ . The limiting denoiser as  $k \rightarrow \infty$  pushes the noise distribution toward a uniform distribution on the circle, which is the correct statistical limit of the above oscillatory map.

### 7.3.3 Theory

We will study  $L^*$ -Lipschitz minimizers of the deterministic loss,

$$\mathcal{J}_{\text{det}}^{(k)}(\Psi_\theta) = \mathbb{E}_{h \sim \bar{\mu}} |\Psi_\theta(h) - \mathcal{S}^{(k)}(h)|^2,$$

with  $\bar{\mu} = \mathcal{U}([0, 1])$  the uniform measure, and the conditional diffusion training objective,

$$\mathcal{J}^{(k)}(D_\theta) = \mathbb{E}_{h \sim \bar{\mu}} \mathbb{E}_{u|h} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} |D_\theta(u + \eta; h, \sigma) - u|^2,$$

where  $u|h = \mathcal{S}^{(k)}(h)$ . The length scale hypothesis in the Main Text suggests that, for some length scale  $\tilde{\epsilon} > 0$ , and  $\delta h \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}])$ , the Lipschitz optimizer of  $\mathcal{J}_{\text{det}}^{(k)}$  satisfies  $\Psi_{\theta^*}(h) \approx \mathbb{E}_{\delta h} [\mathcal{S}^\Delta(h + \delta h)]$ , and that the constrained optimizer of  $\mathcal{J}^{(k)}$  is approximately equal to the optimal denoiser for  $\text{Law}_{\delta h} [\mathcal{S}^\Delta(h + \delta h)]$ . Our goal is to make this intuition rigorous, via asymptotic analysis as  $k \rightarrow \infty$ .

**Deterministic Models Collapse to the Mean.** We denote the constrained optimizer of the deterministic problem formulation by,

$$\Psi^{(k)} := \underset{\text{Lip}(\Psi_\theta) \leq L^*}{\text{argmin}} \mathcal{J}_{\text{det}}^{(k)}(\Psi_\theta).$$

We note that the underlying map  $\mathcal{S}^{(k)}$  has length-scale  $1/k$ , whereas the smallest length scale of  $\Psi^{(k)}$  is uniformly bounded due to the imposed Lipschitz bound. We now denote

$$\tilde{\Psi}^{(k)}(h) := \mathbb{E}_{\delta h} [\mathcal{S}^{(k)}(h + \delta h)], \quad \delta h \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]).$$

We expect that  $\Psi^{(k)}(h) \approx \tilde{\Psi}^{(k)}(h)$ , under the assumptions of the Main Text. We next confirm this intuition, by showing that  $\Psi^{(k)}$  and  $\tilde{\Psi}^{(k)}$  are asymptotically equivalent, as  $k \rightarrow \infty$ .

**Proposition 7.7.** *With the definitions above, we have*

$$\lim_{k \rightarrow \infty} \mathbb{E}_{h \sim \bar{\mu}} |\Psi^{(k)}(h) - \tilde{\Psi}^{(k)}(h)|^2 = 0.$$

*In fact, both  $\Psi^{(k)}, \tilde{\Psi}^{(k)} \rightarrow 0$  collapse to 0 in  $L^2(\bar{\mu})$  as  $k \rightarrow \infty$ .*

The last proposition rigorously justifies the approximate identity  $\Psi^{(k)}(h) \approx \mathbb{E}_{\delta h} [\mathcal{S}^{(k)}(h + \delta h)]$ , and shows that the optimal constrained model  $\Psi^{(k)}$  collapses to the mean (zero), in the limit  $k \rightarrow \infty$ .

**Probabilistic Models Predict Uncertainty.** We next consider the probabilistic problem formulation of conditional diffusion models. We denote by

$$D^{(k)}(u_\sigma; h, \sigma) := \underset{\text{Lip}(D_\theta) \leq L^*}{\operatorname{argmin}} \mathcal{J}^{(k)}(D_\theta),$$

the optimal constrained denoiser for  $\mu^{(k)}(du | h) = \delta(u - \mathcal{S}^{(k)}(h))$ . Given the discussion in the Main Text, we consider the conditional probability,

$$\nu^{(k)}(du | h) := \text{Law}_{\delta h} \left[ \mathcal{S}^{(k)}(h + \delta h) \right], \quad \delta h \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]),$$

for which  $\nu^{(k)}(du | h) \approx \text{Law}_{\delta h} [\mathcal{S}^{(k)}(h + \delta h)]$ , consistent with the discussion in the Main Text. We define

$$\tilde{D}^{(k)}(u_\sigma; h, \sigma) := \underset{D}{\operatorname{argmin}} \mathbb{E}_{h \sim \bar{\mu}} \mathbb{E}_{u \sim \nu^{(k)}(\cdot | h)} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D(u + \eta; h, \sigma) - u\|^2.$$

to be the optimal (unconstrained) denoiser of  $\nu^{(k)}(du | h)$ . Similar to toy model #1, we will show that  $\nu^{(k)}(du | h)$  is asymptotically equivalent to  $\mu(du | h)$ , the conditional distribution arising from the joint uniform probability  $\mu = \mathcal{U}(\mathbb{S}^1) \otimes \mathcal{U}([0, 1])$  on  $\mathbb{S}^1 \times [0, 1]$ , with  $\mathbb{S}^1 \subset \mathbb{R}^2$  denoting the unit circle. The following result shows that the optimal constrained denoiser for  $\mu^{(k)} = \delta(u - \mathcal{S}^{(k)}(h))$  is asymptotically equivalent to the optimal (unconstrained) denoiser for  $\nu^{(k)}(du | h) = \text{Law}_{\delta h} [\mathcal{S}^{(k)}(h + \delta h)]$ , up to an error term that is exponentially small in the Lipschitz constant  $L^*$ :

**Proposition 7.8.** *Let  $\mu := \mathcal{U}(\mathbb{S}^1) \otimes \mathcal{U}([0, 1])$ . With the definitions above, and for constant  $L^*$  sufficiently large, we have*

$$\limsup_{k \rightarrow \infty} \mathbb{E}_{(u, h) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^{(k)}(u + \eta; h, \sigma) - \tilde{D}^{(k)}(u + \eta; h, \sigma) \right\|^2 \leq C e^{-L^*/8C},$$

with  $C$  and  $L^*$  independent of  $\sigma > 0$ .

We note that the appearance of the exponentially small additional term is due to the fact that the limiting denoiser is not uniformly Lipschitz continuous at the origin. However, since the origin is far from the data manifold, this mismatch only leads to a very small error contribution. Thus, we argue that also in this case, this analysis can justify the conclusion drawn from the length scale hypothesis of the Main Text, in the asymptotic regime  $k \rightarrow \infty$ .

## 7.4 Mathematical Derivation

### 7.4.1 Characterizing the Optimal Denoiser

The simple form of the forward process

$$u_\sigma = u + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I), \tag{106}$$

with  $\eta$  independent of  $u$  and  $\bar{u}$ , allows for explicit solution of the optimal denoiser, giving us insight into its mathematical properties. If  $p(u | \bar{u})$  is the conditional distribution of  $u$  given the initial data  $\bar{u}$ , then the diffusion process defines  $u_\sigma | (u, \bar{u})$  as a Gaussian random variable. The denoiser and its gradient are closely related to the posterior distribution of  $u | (u_\sigma, \bar{u})$ :

**Lemma 7.9.** *Assume that  $u \sim p(\cdot | \bar{u})$ , and  $u_\sigma$  is obtained by the forward process (106). Then the minimizer of  $D_{\text{opt}} = \operatorname{argmin}_D \mathcal{J}(D, \sigma)$  (cp. (89)) is given by*

$$D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) = \mathbb{E}[u | (u_\sigma, \bar{u})]. \quad (107)$$

The posterior distribution  $u | (u_\sigma, \bar{u}) \sim q_\sigma(u; u_\sigma, \bar{u})$  is given by the following mathematical expression,

$$q_\sigma(u; w, \bar{u}) = \frac{1}{Z_\sigma} e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}), \quad Z_\sigma = \int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du. \quad (108)$$

For completeness, we include a proof of Lemma 7.9 after the statement of Proposition 7.11, below. In words, the explicit formula in Lemma 7.9 tells us the following: Given that the noise process has ended up at location  $u_\sigma$  and given the additional information about  $\bar{u}$ , the denoiser considers the distribution of all possible origins  $u | (u_\sigma, \bar{u})$  of the noise process over the distribution  $p(u | \bar{u})$  and it singles out the most likely origin as the expected value over this distribution, i.e. the value  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) = \mathbb{E}[u | (u_\sigma, \bar{u})]$ .

We also remark the following corollary, which is immediate from Lemma 7.9:

**Corollary 7.10.** *If  $p(\cdot | \bar{u})$  is supported on a bounded set  $\{|u| \leq M\}$ , then  $|D_{\text{opt}}(u_\sigma; \bar{u}, \sigma)| \leq M$  for all  $u_\sigma$  and  $\sigma$ .*

It is interesting to consider the limit  $\sigma \rightarrow 0$  of the posterior (108). We fix  $w$  independently of  $\sigma$ , and consider the limiting behavior as  $\sigma \rightarrow 0$ , conditioned on the event  $u_\sigma = w$ . In this limit, the numerator and denominator individually (formally) converge to  $w p(w | \bar{u})$  and  $p(w | \bar{u})$ , respectively. Thus if we evaluate the optimal denoiser (107) at  $u_\sigma = w$ , we expect  $D_{\text{opt}}(w; \bar{u}, \sigma) \rightarrow w$  as  $\sigma \rightarrow 0$ . This is true, if  $p(w | \bar{u}) > 0$  and if e.g.  $w \mapsto p(w | \bar{u})$  is continuous. However, in general,  $p(u | \bar{u})$  may be 0 in some locations, or may even be supported on a lower-dimensional data manifold. In this case, we may have  $p(w | \bar{u}) = 0$  at  $w$ , and the behavior of  $\lim_{\sigma \rightarrow 0} D_{\text{opt}}(w; \bar{u}, \sigma)$  is unclear, at first sight. We next show that  $D_{\text{opt}}(w; \bar{u}, \sigma)$  converges to the closest point in the support of  $p(\cdot | \bar{u})$ :

**Proposition 7.11.** *Fix  $w \in \mathbb{R}^d$ . Assume that there exists a unique closest point  $w^* \in \mathbb{R}^d$  in the support of  $p(u | \bar{u})$ , i.e.  $w^* = \operatorname{argmin}_{u \in \operatorname{supp}(p(\cdot | \bar{u}))} |w - u|$ . Then,*

$$D_{\text{opt}}(w; \bar{u}, \sigma) = \mathbb{E}_{u \sim q_\sigma(\cdot; w, \bar{u})}[u] \rightarrow w^*, \quad \text{as } \sigma \rightarrow 0,$$

i.e. in this limit the optimal denoiser  $D_{\text{opt}}(w; \sigma = 0)$  evaluated at  $w$ , points to the closest point  $w^*$  in the support of  $p(u | \bar{u})$ .

We are often interested in comparing optimal denoisers between two probability measures  $\mu$  and  $\mu^\Delta$ . We end this section by stating two results that allow us to relate the distance between optimal denoisers to the Wasserstein distance between  $\mu$  and  $\mu^\Delta$ . A first result is provided by the previously stated Proposition 7.3. Calculations on specific examples (e.g. toy model #2) show that the optimal denoiser  $D_{\text{opt}}$  can be singular in the limit  $\sigma \rightarrow 0$ , in the sense that the local Lipschitz constant may blow up in certain locations. However, under suitable hypotheses on the data distribution, this only happens at a positive distance from the data distribution. The following Proposition generalizes Proposition 7.3 to allow for this possibility.

**Proposition 7.12.** *With the notation of Proposition 7.3, assume that there exists a set  $A \subset \mathbb{R}^d$  such that the restriction  $D_{\text{opt}}(\cdot; \cdot, \sigma)|_A$  is  $L^*$ -Lipschitz continuous for all  $\sigma > 0$ . Let  $D^\Delta$  be defined as before. Then*

$$\mathcal{E} := \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma)\|^2,$$

is upper bounded by

$$\mathcal{E} \leq C \left\{ L^* W_1(\mu^\Delta, \mu) + \text{Prob}_{\mu_\sigma} [\mathbb{R}^d \setminus A] + \text{Prob}_{\mu_\sigma^\Delta} [\mathbb{R}^d \setminus A] \right\}, \quad (109)$$

where  $C = C(M) > 0$  is a constant depending only on  $M$ .

#### 7.4.2 Proofs for Section 7.4.1.

*Proof of Lemma 7.9.* We recall that by definition,  $D_{\text{opt}}(u; \bar{u}, \sigma)$  minimizes the functional

$$\mathcal{J}(D, \sigma) = \mathbb{E}_{u \sim p(\cdot | \bar{u})} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D(u + \eta; \bar{u}, \sigma) - u\|^2.$$

We now replace the expectation over  $(u, \eta) \sim p(u | \bar{u}) \otimes \mathcal{N}(0, \sigma^2 I)$  by the expectation over  $(u, u_\sigma) | \bar{u}$ , where  $u_\sigma | (u, \bar{u}) = u + \eta$  is obtained from the noise process and  $u | \bar{u} \sim p(u | \bar{u})$ . Then,

$$\mathcal{J}(D, \sigma) = \mathbb{E}_{(u, u_\sigma) | \bar{u}} \|D(u_\sigma; \bar{u}, \sigma) - u\|^2.$$

Next, we note that  $D(u_\sigma; \bar{u}, \sigma)$  depends on  $u_\sigma$ , but not on  $u$ . This motivates splitting the expectation up as  $\mathbb{E}_{(u, u_\sigma) | \bar{u}} = \mathbb{E}_{u_\sigma \sim p_\sigma(\cdot | \bar{u})} \mathbb{E}_{u | (u_\sigma, \bar{u})}$ , to obtain,

$$\begin{aligned} \mathcal{J}(D, \sigma) &= \mathbb{E}_{u_\sigma \sim p_\sigma(\cdot | \bar{u})} \mathbb{E}_{u | (u_\sigma, \bar{u})} \|D(u_\sigma; \bar{u}, \sigma) - u\|^2 \\ &= \mathbb{E}_{u_\sigma \sim p_\sigma(\cdot | \bar{u})} \|D(u_\sigma; \bar{u}, \sigma) - \mathbb{E}_{u | (u_\sigma, \bar{u})}[u]\|^2 + \mathbb{E}_{u_\sigma \sim p_\sigma(\cdot | \bar{u})} \text{Var}_{u | (u_\sigma, \bar{u})}[u], \end{aligned}$$

where the last identity follows from a simple bias-variance decomposition. Since the last term is independent of  $D(u_\sigma; \bar{u}, \sigma)$ , it follows that  $\mathcal{J}(D, \sigma)$  is minimized by the choice  $D(u_\sigma; \bar{u}, \sigma) = \mathbb{E}_{u | (u_\sigma, \bar{u})}[u]$ . The formula for the posterior follows by a straightforward calculation from Bayes formula,

$$p(u | u_\sigma, \bar{u}) \propto p(u_\sigma | u, \bar{u}) p(u | \bar{u}).$$

□

*Proof of Proposition 7.11.* Let  $w \in \mathbb{R}^d$  be given, and let  $w^*$  denote the closest point to  $w$  in the support of  $p(\cdot | \bar{u})$ . Let  $q_\sigma(u; w, \bar{u})$  denote the posterior measure (108). Since  $D_{\text{opt}}(w; \bar{u}, \sigma) = \int u q_\sigma(u; w, \bar{u}) du$  and  $w^* = \int w^* q_\sigma(u; w, \bar{u}) du$ , we have

$$\begin{aligned} |D_{\text{opt}}(w; \bar{u}, \sigma) - w^*| &\leq \int |u - w^*| q_\sigma(u; w, \bar{u}) du \\ &= \frac{\int |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}. \end{aligned}$$

Denote  $r := |w - w^*|$ , and let  $\epsilon > 0$  be given. Since  $w^*$  is the unique closest point to  $w$ , in the support of  $p(\cdot | \bar{u})$ , it follows that there exists  $\delta > 0$ , such that  $|u - w| < r + \delta$  implies that  $|u - w^*| < \epsilon$ .<sup>1</sup> Then,

$$\begin{aligned} \int |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du &= \int_{r \leq |u-w| < r+\delta} |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du \\ &\quad + \int_{|u-w| \geq r+\delta} |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du \\ &\leq \epsilon \int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du + e^{-(r+\delta)^2/2\sigma^2} \mathbb{E}_{u \sim p(\cdot | \bar{u})}[|u - w^*|], \end{aligned}$$

and

$$\begin{aligned} \int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du &\geq \int_{r \leq |u-w| \leq r+\delta/2} e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du \\ &\geq e^{-(r+\delta/2)^2/2\sigma^2} \int_{r \leq |u-w| \leq r+\delta/2} p(u | \bar{u}) du \\ &\geq e^{-(r+\delta/2)^2/2\sigma^2} \int_{|u-w^*| \leq \delta/2} p(u | \bar{u}) du. \end{aligned}$$

We note that  $\int_{|u-w^*| \leq \delta/2} p(u | \bar{u}) du > 0$ , since  $w^*$  belongs to the support of  $p$ . It follows that

$$\begin{aligned} \frac{\int |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du} &\leq \frac{\epsilon \int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du} \\ &\quad + \frac{e^{-(r+\delta)^2/2\sigma^2} \mathbb{E}_{u \sim p(\cdot | \bar{u})}[|u - w^*|]}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du} \\ &\leq \epsilon + \frac{e^{-(r+\delta)^2/2\sigma^2} \mathbb{E}_{u \sim p(\cdot | \bar{u})}[|u - w^*|]}{e^{-(r+\delta/2)^2/2\sigma^2} \int_{|u-w^*| \leq \delta/2} p(u | \bar{u}) du}. \end{aligned}$$

Letting  $\sigma \rightarrow 0$ , the last term converges to 0 on account of the fact that

$$e^{-(r+\delta)^2/2\sigma^2} \ll e^{-(r+\delta/2)^2/2\sigma^2}.$$

Thus,

$$\limsup_{\sigma \rightarrow 0} \frac{\int |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du} \leq \epsilon.$$

<sup>1</sup>If not, then there exists a sequence  $u_n \in \text{supp}(p(\cdot | \bar{u}))$ , such that  $|u_n - w| \leq r + \frac{1}{n}$ , while at the same time  $|u_n - w^*| \geq \epsilon > 0$ . This sequence must have a limit point  $u^*$ , necessarily belonging to the (closed) support of  $p(\cdot | \bar{u})$ ,  $|u^* - w| \leq \limsup_n |u_n - w| = r$ , and  $|u^* - w^*| \geq \epsilon > 0$ ; thus,  $u^*$  is as close to  $w$  as  $w^*$ , contradicting the uniqueness of  $w^*$ .



Since  $\epsilon > 0$  was arbitrary, and the left-hand side is independent of  $\epsilon$ , we conclude that

$$\lim_{\sigma \rightarrow 0} |D_{\text{opt}}(w; \bar{u}, \sigma) - w^*| \leq \lim_{\sigma \rightarrow 0} \frac{\int |u - w^*| e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du}{\int e^{-|u-w|^2/2\sigma^2} p(u | \bar{u}) du} = 0.$$

This concludes our proof.  $\square$

The following lemma will be used in the proof of Proposition 7.3 and 7.12.

**Lemma 7.13.** *Let  $\mathcal{A} \subset H$  be a convex set in a Hilbert space  $H$ . Let  $\mathcal{J}(D) = \|D - F\|^2$  be a quadratic functional on  $\mathcal{A}$ , where  $F \in H$ . If  $D_{\text{opt}} \in \operatorname{argmin}_{D \in \mathcal{A}} \mathcal{J}(D)$ , then*

$$\mathcal{J}(D) - \mathcal{J}(D_{\text{opt}}) \geq \|D - D_{\text{opt}}\|^2, \quad \forall D \in \mathcal{A}.$$

*Proof of Lemma 7.13.* Fix  $D \in \mathcal{A}$  and let  $D_\tau := (1 - \tau)D_{\text{opt}} + \tau D$ . Since  $\mathcal{A}$  is convex, we have  $D_\tau \in \mathcal{A}$  for all  $\tau \in [0, 1]$ . Since  $D_{\text{opt}}$  is a minimizer of  $\mathcal{J}$ , it follows that  $\frac{d}{d\tau} \mathcal{J}(D_\tau) \big|_{\tau=0} \geq 0$ . Evaluating the derivative, this implies,

$$\frac{d}{d\tau} \bigg|_{\tau=0} \mathcal{J}(D_\tau) = 2\langle \dot{D}_\tau, D_{\text{opt}} - F \rangle = 2\langle D - D_{\text{opt}}, D_{\text{opt}} - F \rangle \geq 0, \quad \forall D \in \mathcal{A}.$$

Given  $D \in \mathcal{A}$ , we now obtain

$$\begin{aligned} \mathcal{J}(D) - \mathcal{J}(D_{\text{opt}}) &= \|D - F\|^2 - \|D_{\text{opt}} - F\|^2 \\ &= \langle (D - F) - (D_{\text{opt}} - F), (D - F) + (D_{\text{opt}} - F) \rangle \\ &= \langle D - D_{\text{opt}}, D + D_{\text{opt}} - 2F \rangle \\ &= \langle D - D_{\text{opt}}, D - D_{\text{opt}} \rangle + \underbrace{2\langle D - D_{\text{opt}}, D_{\text{opt}} - F \rangle}_{\geq 0} \\ &\geq \|D - D_{\text{opt}}\|^2. \end{aligned}$$

$\square$

The proof of Proposition 7.3 will also make use of the following:

**Lemma 7.14.** *Assume  $\phi : \mathcal{X} \rightarrow \mathcal{X}$  is a Lipschitz function, and  $\mu, \nu \in \operatorname{Prob}(\mathcal{X})$  are probability measures. Then,*

$$|\mathbb{E}_{u \sim \mu} \|\phi(u)\|^2 - \mathbb{E}_{u \sim \nu} \|\phi(u)\|^2| \leq 2\|\phi\|_{L^\infty} \operatorname{Lip}(\phi) W_1(\mu, \nu).$$

*Proof of Lemma 7.14.* Let  $\pi \in \text{Prob}(\mathcal{X} \times \mathcal{X})$  be an optimal  $W_1$ -coupling between  $\mu$  and  $\nu$ . Then,

$$\begin{aligned}
 \mathbb{E}_{u \sim \mu} \|\phi(u)\|^2 - \mathbb{E}_{u \sim \nu} \|\phi(u)\|^2 &= \int \|\phi(u)\|^2 d\mu(u) - \int \|\phi(v)\|^2 d\nu(v) \\
 &= \int \{\|\phi(u)\|^2 - \|\phi(v)\|^2\} d\pi(u, v) \\
 &= \int (\|\phi(u)\| + \|\phi(v)\|) \{\|\phi(u)\| - \|\phi(v)\|\} d\pi(u, v) \\
 &\leq \int 2\|\phi\|_{L^\infty} \{\|\phi(u) - \phi(v)\|\} d\pi(u, v) \\
 &\leq 2\|\phi\|_{L^\infty} \text{Lip}(\phi) \int \{\|u - v\|\} d\pi(u, v) \\
 &= 2\|\phi\|_{L^\infty} \text{Lip}(\phi) W_1(\mu, \nu).
 \end{aligned}$$

This proves the claimed bound if  $\mathbb{E}_{u \sim \mu} \|\phi(u)\|^2 \geq \mathbb{E}_{u \sim \nu} \|\phi(u)\|^2$ . For the reverse case, we can simply switch  $\mu$  and  $\nu$  in the above estimates. The claimed bound thus follows.  $\square$

### 7.4.3 Proof of Proposition 7.3

We now come to the proof of Proposition 7.3.

*Proof of Proposition 7.3.* The idea is to compare the optimal constrained denoiser  $D^\Delta = \text{argmin}_{\text{Lip}(D_\theta) \leq L^*} \mathcal{J}^\Delta(D_\theta)$  with the unconstrained denoiser  $D_{\text{opt}} = \text{argmin}_D \mathcal{J}(D)$ , for

$$\mathcal{J}(D) := \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D(u + \eta; \bar{u}, \sigma) - u\|^2.$$

By the assumptions of this proposition,  $D_{\text{opt}}$  is  $L^*$ -Lipschitz continuous. We note that, for any  $\sigma > 0$ ,

$$\begin{aligned}
 \mathcal{J}(D^\Delta, \sigma) &\leq \mathcal{J}^\Delta(D^\Delta, \sigma) + |\mathcal{J}(D^\Delta, \sigma) - \mathcal{J}^\Delta(D^\Delta, \sigma)| \\
 &\leq \mathcal{J}^\Delta(D_{\text{opt}}, \sigma) + |\mathcal{J}(D^\Delta, \sigma) - \mathcal{J}^\Delta(D^\Delta, \sigma)| \\
 &\leq \mathcal{J}(D_{\text{opt}}, \sigma) + 2 \max_{D=D_{\text{opt}}, D^\Delta} |\mathcal{J}(D, \sigma) - \mathcal{J}^\Delta(D, \sigma)|. \tag{110}
 \end{aligned}$$

To prove the claim, it thus suffices to show that there exists  $C > 0$ , independent of  $\Delta$ ,  $L^*$  and  $\sigma$ , such that

$$2 \max_{D=D_{\text{opt}}, D^\Delta} |\mathcal{J}(D, \sigma) - \mathcal{J}^\Delta(D, \sigma)| \leq CL^* W_1(\mu^\Delta, \mu).$$

To prove such an estimate, we first recall that

$$\mathcal{J}(D, \sigma) = \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D(u + \eta; \bar{u}, \sigma) - u\|^2,$$

and similarly for  $\mathcal{J}^\Delta$ , except that  $(u, \bar{u}) \sim \mu$  is replaced by  $(u, \bar{u}) \sim \mu^\Delta$ . Let us now momentarily fix  $\eta \in \mathcal{X}$ . Given a choice of either  $D = D^\Delta$  or  $D = D_{\text{opt}}$ , we introduce,

$$\phi_\eta(u, \bar{u}) := D(u + \eta; \bar{u}, \sigma) - u.$$

The following estimate will hold for either choice of  $D = D^\Delta, D_{\text{opt}}$ . By assumption on  $D^\Delta, D_{\text{opt}}$  being  $L^*$ -Lipschitz, it follows that  $\text{Lip}(\phi_\eta) \leq \text{Lip}(D) + 1 \leq L^* + 1$ . By Lemma 7.14, it therefore follows that

$$\begin{aligned}
|\mathcal{J}(D, \sigma) - \mathcal{J}^\Delta(D, \sigma)| &= \left| \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|\phi_\eta(u, \bar{u})\|^2 - \mathbb{E}_{(u, \bar{u}) \sim \mu^\Delta} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|\phi_\eta(u, \bar{u})\|^2 \right| \\
&\leq \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left| \mathbb{E}_{(u, \bar{u}) \sim \mu} \|\phi_\eta(u, \bar{u})\|^2 - \mathbb{E}_{(u, \bar{u}) \sim \mu^\Delta} \|\phi_\eta(u, \bar{u})\|^2 \right| \\
&\leq 2 \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|\phi_\eta\|_{L^\infty}] \text{Lip}(\phi_\eta) W_1(\mu, \mu^\Delta) \\
&\leq 2(L^* + 1) \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|\phi_\eta\|_{L^\infty}] W_1(\mu, \mu^\Delta).
\end{aligned} \tag{111}$$

Comparing with (95), we finally need to show that  $\mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|\phi_\eta\|_{L^\infty}] \leq B$  is bounded by a constant  $B$  independent of  $\Delta$ . Then (95) holds with constant  $2(L^* + 1)B \leq 4L^*B =: C$ , where we used  $L^* \geq 1$  to get a simpler bound.

We note that by the explicit formula for  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) = \mathbb{E}[u \mid (u_\sigma, \bar{u})]$  and the assumption that  $\mu$  is concentrated on  $B_M = \{\|u\| \leq M\}$ , it is immediate that  $\|D_{\text{opt}}(u_\sigma; \bar{u}, \sigma)\| \leq M$  for any choice of  $u_\sigma$ . In particular, this implies that for  $D = D_{\text{opt}}$ , we have

$$\mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|\phi_\eta\|_{L^\infty}] = \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|D_{\text{opt}}(u + \eta; \bar{u}, \sigma) - u\|_{L^\infty}] \leq 2M. \tag{112}$$

For  $D^\Delta$ , we can also show that  $\|D^\Delta(u + \eta; \bar{u}, \sigma)\| \leq M$ . To see this, let us introduce the  $M$ -truncated mapping,

$$D_M^\Delta(u_\sigma; \bar{u}, \sigma) := \begin{cases} D^\Delta(u_\sigma; \bar{u}, \sigma), & \text{if } \|D^\Delta(u_\sigma; \bar{u}, \sigma)\| \leq M, \\ \frac{M D^\Delta(u_\sigma; \bar{u}, \sigma)}{\|D^\Delta(u_\sigma; \bar{u}, \sigma)\|} & \text{if } \|D^\Delta(u_\sigma; \bar{u}, \sigma)\| > M. \end{cases}$$

Then  $D_M^\Delta$  is still  $L^*$ -Lipschitz. However, it is easy to see that for any  $\|u\| \leq M$  and  $\bar{u}, \eta \in \mathcal{X}$ , we have

$$\|D_M^\Delta(u + \eta; \bar{u}, \sigma) - u\| \leq \|D^\Delta(u + \eta; \bar{u}, \sigma) - u\|.$$

Upon taking expectations with respect to  $u, \bar{u}, \eta$ , this in turn implies that  $\mathcal{J}^\Delta(D_M^\Delta, \sigma) \leq \mathcal{J}^\Delta(D^\Delta, \sigma)$ . However,  $D^\Delta$  is by assumption the minimizer of the functional  $\mathcal{J}^\Delta$ , over the set of  $L^*$ -Lipschitz mappings. By the uniqueness of a minimizer over this (convex) set, and since  $D_M^\Delta$  is still  $L^*$ -Lipschitz, it follows that  $D^\Delta = D_M^\Delta$ , i.e.  $D^\Delta$  is uniformly bounded by  $M$ . Thus, also in this case, we have for  $D = D^\Delta$ :

$$\mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|\phi_\eta\|_{L^\infty}] = \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} [\|D^\Delta(u + \eta; \bar{u}, \sigma) - u\|_{L^\infty}] \leq 2M. \tag{113}$$

Combining (113), (112), (111) and (110), we conclude that

$$\mathcal{J}(D^\Delta, \sigma) \leq \mathcal{J}(D_{\text{opt}}, \sigma) + CL^*W_1(\mu^\Delta, \mu),$$

for  $C = 8M$ . Since  $D_{\text{opt}}$  is the optimizer of the quadratic functional  $\mathcal{J}$ , it follows from Lemma 7.13 that

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma)\|^2 \leq \mathcal{J}(D^\Delta) - \mathcal{J}(D_{\text{opt}}),$$

and hence

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma)\|^2 \leq CL^* W_1(\mu^\Delta, \mu),$$

by the previous bound. This completes the proof of Proposition 7.3.  $\square$

#### 7.4.4 Proof of Proposition 7.12

*Proof of Proposition 7.12.* We note that

$$\begin{aligned} \mathcal{J}(D^\Delta) &\leq \mathcal{J}^\Delta(D^\Delta) + |\mathcal{J}(D^\Delta) - \mathcal{J}^\Delta(D^\Delta)| \\ &\leq \mathcal{J}^\Delta(D_{\text{opt}}) + |\mathcal{J}(D^\Delta) - \mathcal{J}^\Delta(D^\Delta)| \\ &\leq \mathcal{J}(D_{\text{opt}}) + |\mathcal{J}(D^\Delta) - \mathcal{J}^\Delta(D^\Delta)| + |\mathcal{J}(D_{\text{opt}}) - \mathcal{J}^\Delta(D_{\text{opt}})|. \end{aligned}$$

By assumption  $D^\Delta$  is  $L^*$ -Lipschitz. Thus, (111) and (112) in the proof of Proposition 7.3 imply that

$$|\mathcal{J}(D^\Delta) - \mathcal{J}^\Delta(D^\Delta)| \leq 4M(L^* + 1)W_1(\mu, \mu^\Delta).$$

By assumption,  $D_{\text{opt}}$  is  $L^*$ -Lipschitz when restricted to  $A \subset \mathbb{R}^d$ . By the Kirszbraun theorem, there therefore exists  $D : \mathbb{R}^d \rightarrow \mathbb{R}^d$  with  $\text{Lip}(D) = \text{Lip}(D_{\text{opt}}|_A)$ ,  $\|D\|_{L^\infty} = \|D_{\text{opt}}\|_{L^\infty}$  and  $D|_A \equiv D_{\text{opt}}|_A$ . Given such a choice of  $D$ , we now bound

$$|\mathcal{J}(D_{\text{opt}}) - \mathcal{J}^\Delta(D_{\text{opt}})| \leq |\mathcal{J}(D_{\text{opt}}) - \mathcal{J}(D)| + |\mathcal{J}(D) - \mathcal{J}^\Delta(D)| + |\mathcal{J}^\Delta(D) - \mathcal{J}^\Delta(D_{\text{opt}})|.$$

Denote  $A^c = \mathbb{R}^d \setminus A$ . The first term can be bounded by observing that

$$\begin{aligned} |\mathcal{J}(D_{\text{opt}}) - \mathcal{J}(D)| &= |\mathbb{E}_u \|D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) - u\|^2 - \mathbb{E}_u \|D(u_\sigma; \bar{u}, \sigma) - u\|^2| \\ &= |\mathbb{E}_u [1_{A^c}(u_\sigma, \bar{u}) \|D_{\text{opt}}(u_\sigma; \bar{u}, \sigma) - u\|^2] - \mathbb{E}_u [1_{A^c}(u_\sigma, \bar{u}) \|D(u_\sigma; \bar{u}, \sigma) - u\|^2]| \\ &\leq (\|D_{\text{opt}}\|_{L^\infty} + 1)^2 \text{Prob}_{\mu_\sigma}[A^c]. \end{aligned}$$

Since  $D_{\text{opt}}$  is the optimal denoiser for  $\mu$ , it follows that  $\|D_{\text{opt}}\|_{L^\infty} \leq M$ , from Corollary 7.10. Thus,

$$|\mathcal{J}(D_{\text{opt}}) - \mathcal{J}(D)| \leq (M + 1)^2 \text{Prob}_{\mu_\sigma}[A^c].$$

Similarly, we can show that

$$|\mathcal{J}^\Delta(D_{\text{opt}}) - \mathcal{J}^\Delta(D)| \leq (M + 1)^2 \text{Prob}_{\mu_\sigma^\Delta}[A^c].$$

Finally, (111) and (112) in the proof of Proposition 7.3 imply that

$$|\mathcal{J}(D) - \mathcal{J}^\Delta(D)| \leq 4M(L^* + 1)W_1(\mu, \mu^\Delta).$$

Combining these estimates, it follows that

$$\mathcal{J}(D^\Delta) - \mathcal{J}(D_{\text{opt}}) \leq C \left\{ L^* W_1(\mu, \mu^\Delta) + \text{Prob}_{\mu_\sigma}[A^c] + \text{Prob}_{\mu_\sigma^\Delta}[A^c] \right\},$$

where  $C = C(M) > 0$  depends only on  $M$ .  $D_{\text{opt}}$  is the unconstrained optimizer of the quadratic functional  $\mathcal{J}$ . Thus, by Lemma 7.13, it follows that

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \|D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma)\|^2 \leq \mathcal{J}(D^\Delta) - \mathcal{J}(D_{\text{opt}}).$$

The claimed bound on the error thus follows.  $\square$

#### 7.4.5 Proofs for Section 7.2

We here give the detailed proof of Proposition 7.5 and Proposition 7.6.

**Deterministic Setting.** We start with the proof of Proposition 7.5.

*Proof of Proposition 7.5.* We first note that  $\tilde{\Psi}^\Delta(\bar{u}) = m(\bar{u}) + \mathbb{E}_{\delta\bar{u}}[s_N(\bar{u} + \delta\bar{u})]$  converges to  $m(\bar{u})$  as  $\Delta \rightarrow 0$ . This follows easily from well-known facts about weak limits, which imply in particular that the rapidly oscillating function  $\delta\bar{u} \mapsto s_N(\bar{u} + \delta\bar{u})$  satisfies,

$$\mathbb{E}_{\delta\bar{u}}[s_N(\bar{u} + \delta\bar{u})] = \frac{1}{2\tilde{\epsilon}} \int_{-\tilde{\epsilon}}^{\tilde{\epsilon}} \Lambda(N(\bar{u} + v)) dv \rightarrow \int_0^1 \Lambda(\xi) d\xi = 0,$$

where the last equality follows from our definition of  $\Lambda$ . Since this convergence holds pointwise for any fixed  $\bar{u}$ , by dominated convergence, it follows also in  $L^2([0, 1])$ . Thus, we conclude that  $\tilde{\Psi}^\Delta(\bar{u})$  and  $m(\bar{u})$  are asymptotically equivalent, in the sense that

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\tilde{\Psi}^\Delta(\bar{u}) - m(\bar{u})\|^2 = 0.$$

It will thus suffice to show,

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\Psi^\Delta(\bar{u}) - m(\bar{u})\|^2 = 0.$$

To this end, we first write

$$\begin{aligned} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\Psi^\Delta(\bar{u}) - m(\bar{u})\|^2 &= \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\Psi^\Delta(\bar{u}) - \mathcal{S}^\Delta(\bar{u})\|^2 \\ &\quad - 2\mathbb{E}_{\bar{u} \sim \bar{\mu}} \langle \Psi^\Delta(\bar{u}) - m(\bar{u}), s_N(\bar{u}) \rangle - \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|s_N(\bar{u})\|^2. \end{aligned}$$

Since  $\Psi^\Delta$  minimizes the first term over all  $L^*$ -Lipschitz functions, and since the mean function is  $L^*$ -Lipschitz by assumption, we obtain,

$$\begin{aligned} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\Psi^\Delta(\bar{u}) - m(\bar{u})\|^2 &\leq \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|m(\bar{u}) - \mathcal{S}^\Delta(\bar{u})\|^2 \\ &\quad - 2\mathbb{E}_{\bar{u} \sim \bar{\mu}} \langle \Psi^\Delta(\bar{u}) - m(\bar{u}), s_N(\bar{u}) \rangle - \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|s_N(\bar{u})\|^2 \\ &= -2\mathbb{E}_{\bar{u} \sim \bar{\mu}} \langle \Psi^\Delta(\bar{u}) - m(\bar{u}), s_N(\bar{u}) \rangle. \end{aligned}$$

It is a textbook exercise in analysis to show that  $\Psi(\bar{u}) := \Psi^\Delta(\bar{u}) - m(\bar{u})$  is  $2L^*$ -Lipschitz continuous, and that there exists a constant  $C > 0$ , such that

$$\sup_{\text{Lip}(\Psi) \leq 2L^*} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \langle \Psi(\bar{u}), s_N(\bar{u}) \rangle \leq \frac{C}{N} = C\Delta.$$

Thus, we conclude that  $\lim_{\Delta \rightarrow 0} \mathbb{E}_{\bar{u} \sim \bar{\mu}} \|\Psi^\Delta(\bar{u}) - m(\bar{u})\|^2 = 0$ , as claimed.  $\square$

**Probabilistic Setting.** We now consider the probabilistic setting of conditional diffusion models. Our asymptotic results will hold for any  $\tilde{\epsilon} > 0$ . We thus assume  $\tilde{\epsilon}$  to be fixed (arbitrarily). We recall that,

$$\nu^\Delta(du | \bar{u}) := \text{Law}_{\delta\bar{u}}[m(\bar{u}) + s_N(\bar{u} + \delta\bar{u})], \quad \delta\bar{u} \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]),$$

and  $\tilde{D}^\Delta(u_\sigma; \bar{u}, \sigma)$  denotes the optimal unconstrained conditional denoiser for  $\nu^\Delta$ . Since the derivation of Proposition 7.6 is more involved, we will first give an overview of the essential ingredients, and leave their proof for later paragraphs.

Our first result shows that  $\nu^\Delta \approx \mathcal{U}([m(\bar{u}) - 1, m(\bar{u}) + 1])$  is approximately equivalent to a uniform distribution, in a suitable sense:

**Lemma 7.15.** *Let  $\mu(du | \bar{u}) = \mathcal{U}([m(\bar{u}) - 1, m(\bar{u}) + 1])$  be a uniform measure. There exists a constant  $C > 0$ , independent of  $\Delta$ , such that*

$$(1 - C\Delta)\mu(du | \bar{u}) \leq \nu^\Delta(du | \bar{u}) \leq (1 + C\Delta)\mu(du | \bar{u}).$$

The result of the last lemma is important because it allows us to identify the limit  $D_{\text{opt}} = \lim_{\Delta \rightarrow 0} \tilde{D}^\Delta$ , owing to the following result.

**Lemma 7.16.** *Let  $\mu, \nu$  be probability measures on  $\mathbb{R}^d$ , supported on a bounded set  $\{|u| \leq M\}$  and suppose that for some  $\epsilon \in (0, 1)$ , we have*

$$(1 - \epsilon)\mu \leq \nu \leq (1 + \epsilon)\mu.$$

*Let  $D^\mu(u_\sigma; \sigma), D^\nu(u_\sigma; \sigma)$  denote the corresponding denoisers. Then*

$$\|D^\mu(\cdot; \sigma) - D^\nu(\cdot; \sigma)\|_{L^\infty(\mathbb{R}^d)} \leq 2M\epsilon.$$

Given the results of Lemma 7.15 and Lemma 7.16, the following corollary is now immediate:

**Corollary 7.17.** *Let  $\mu$  be the uniform measure on  $\mathcal{I} := \{(u, \bar{u}) \in \mathbb{R} \times [0, 1] \mid u \in [m(\bar{u}) - 1, m(\bar{u}) + 1]\}$ . Let  $D_{\text{opt}}$  denote the optimal (unconstrained) conditional denoiser for  $\mu$ . Then we have,*

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| \tilde{D}^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma) \right\|^2 \leq C\Delta.$$

Due to the simplicity of  $\mu$ , the optimal denoiser  $D_{\text{opt}}$  can be characterized quite explicitly, as shown next:

**Lemma 7.18.** *Let  $D_{\text{opt}}(u; \bar{u}, \sigma)$  denote the optimal denoiser for the uniform measure  $\mu$  on  $\mathcal{I}$  introduced above. Then  $D_{\text{opt}}$  is  $L^*$ -Lipschitz continuous, uniformly as  $\sigma \rightarrow 0$ , for some constant  $L^* > 0$ , and*

$$\lim_{\sigma \rightarrow 0} D_{\text{opt}}(u; \bar{u}, \sigma) = g(u - m(\bar{u})),$$

where

$$g(u) = \begin{cases} -1, & \text{if } u < -1, \\ u, & \text{if } -1 \leq u \leq +1, \\ +1, & \text{if } u > +1. \end{cases} \quad (114)$$

The proof of Lemma 7.18 is given below. Given the above results, we can now finally come to the proof of Proposition 7.6.

*Proof of Proposition 7.6.* We recall that our goal is to show that

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^\Delta(u + \eta; \bar{u}, \sigma) - \tilde{D}^\Delta(u + \eta; \bar{u}, \sigma) \right\|^2 = 0.$$

Corollary 7.17 shows that  $\tilde{D}^\Delta \rightarrow D_{\text{opt}}$  with  $D_{\text{opt}}$  the conditional diffusion model for  $\mu$ . It will thus be enough to show that

$$\lim_{\Delta \rightarrow 0} \mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma) \right\|^2 = 0.$$

Since  $D_{\text{opt}}$  is  $L^*$ -Lipschitz continuous by Lemma 7.18, it follows from Proposition 7.3 that

$$\mathbb{E}_{(u, \bar{u}) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^\Delta(u + \eta; \bar{u}, \sigma) - D_{\text{opt}}(u + \eta; \bar{u}, \sigma) \right\|^2 \leq CL^* W_1(\mu^\Delta, \mu).$$

Lemma 7.19 below shows that  $W_1(\mu^\Delta, \mu) \rightarrow 0$ , completing the proof.  $\square$

The following lemma identifies a robust statistical limit for this toy problem.

**Lemma 7.19.** *Let  $\mu \in \text{Prob}(\mathbb{R} \times [0, 1])$  be given by the uniform measure on*

$$\mathcal{I}(m) := \{(u, \bar{u}) \in \mathbb{R} \times [0, 1] \mid u \in [m(\bar{u}) - 1, m(\bar{u}) + 1]\}.$$

*Then,*

$$W_1(\mu^\Delta, \mu) = O(\Delta) \rightarrow 0, \quad \text{as } \Delta \rightarrow 0.$$

Interestingly, Lemma 7.19 shows that, even though  $\mathcal{S}^\Delta$  is highly oscillatory and cannot possess a limiting function  $\mathcal{S}^\Delta \not\rightarrow \mathcal{S}$ , the associated probability measure  $\mu^\Delta$  nevertheless converges in a statistical sense to a well-defined limit  $\mu$ . Disintegration of this limit  $\mu$  yields,

$$\mu(du, d\bar{u}) = p(u \mid \bar{u}) du d\bar{u},$$

where  $p(u \mid \bar{u}) = \mathcal{U}[m(\bar{u}) - 1, m(\bar{u}) + 1]$  is the uniform distribution on an interval centered around  $m(\bar{u})$ . In particular, *the limit is not a Dirac  $\delta$ -distribution.*

**Proofs of Lemma 7.15, Lemma 7.16, Lemma 7.18 and Lemma 7.19** In the following, we detail the proofs of Lemma 7.15, Lemma 7.16, Lemma 7.18 and Lemma 7.19.

*Proof of Lemma 7.15.* Note that  $\nu^\Delta$ , by definition, is the pushforward measure of  $\mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}])$  under the mapping  $f_N : [-\tilde{\epsilon}, \tilde{\epsilon}] \rightarrow \mathbb{R}$ ,  $f_N(\xi) = m(\bar{u}) + \Lambda(N\bar{u} + N\xi)$ . Since  $\Lambda$  is a hat function mapping onto the range  $[-1, 1]$ , and since its derivative  $|\Lambda'| = 1$  has magnitude 1 almost everywhere, it follows from the change of variables formula for pushforward measures that  $\nu^\Delta(du \mid \bar{u})$  is probability measure on  $[m(\bar{u}) - 1, m(\bar{u}) + 1]$  with a probability density  $q(u)$  whose value at a given  $u$  is proportional to the number of points in the pre-image of  $u$ , i.e.

$$q(u) = c \# \{\xi \in [-\tilde{\epsilon}, \tilde{\epsilon}] \mid f_N(\xi) = u\},$$

holds for almost every  $u \in [m(\bar{u}) - 1, m(\bar{u}) + 1]$ , where  $c$  is a normalization constant. Since  $f_N$  is  $1/N$ -periodic, there are  $\lfloor 2\tilde{\epsilon}N \rfloor$  completed periods over the interval  $[-\tilde{\epsilon}, \tilde{\epsilon}]$ . On each completed period, the equation  $f_N(\xi) = u$  has two solutions for almost every  $u \in [m(\bar{u}) - 1, m(\bar{u}) + 1]$ . Thus, we have

$$2\lfloor 2\tilde{\epsilon}N \rfloor \leq \#\{\xi \in [-\tilde{\epsilon}, \tilde{\epsilon}] \mid f_N(\xi) = u\} \leq 2\lfloor 2\tilde{\epsilon}N \rfloor + 1,$$

implying that

$$2c\lfloor 2\tilde{\epsilon}N \rfloor \leq q(u) \leq 2c\lfloor 2\tilde{\epsilon}N \rfloor + c, \quad \forall u \in [m(\bar{u}) - 1, m(\bar{u}) + 1].$$

Integration over  $u$ , and using the fact that  $\int q(u) du = 1$ , then implies that  $c \sim 1/4\lfloor \tilde{\epsilon}N \rfloor$ , and hence,

$$q(u) = \frac{1}{2} + O\left(\frac{1}{\tilde{\epsilon}N}\right) = \frac{1}{2} + O(\tilde{\epsilon}^{-1}\Delta),$$

with an absolute implied constant in the big-O notation. Since  $q_0(u) \equiv \frac{1}{2}$  for the considered values of  $u$  is the density of the uniform distribution  $\mu = \mathcal{U}([m(\bar{u}) - 1, m(\bar{u}) + 1])$ , we conclude that there exists a constant  $C = C(\tilde{\epsilon}) > 0$ , proportional to  $1/\tilde{\epsilon}$ , such that

$$(1 - C\Delta)\mu \leq \nu^\Delta \leq (1 + C\Delta)\mu.$$

□

*Proof of Lemma 7.16.* We recall that

$$D^\mu(w; \sigma) = \frac{\int u e^{-|w-u|^2/2\sigma^2} \mu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)}, \quad D^\nu(w; \sigma) = \frac{\int u e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \nu(du)}.$$

We now compute

$$\begin{aligned} D^\mu(w; \sigma) - D^\nu(w; \sigma) &= \frac{\int u e^{-|w-u|^2/2\sigma^2} [\mu(du) - \nu(du)]}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} + \frac{\int u e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \nu(du)} \left( \frac{\int e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} - 1 \right) \\ &= \frac{\int u e^{-|w-u|^2/2\sigma^2} \left[ 1 - \frac{d\nu}{d\mu} \right] \mu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} + D^\nu(w; \sigma) \left( \frac{\int e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} - 1 \right). \end{aligned}$$

By assumption, we have  $(1 - \epsilon)\mu \leq \nu \leq (1 + \epsilon)\mu$ . This implies that the Radon-Nikodym derivative  $d\nu/d\mu$  satisfies  $-\epsilon \leq \frac{d\nu}{d\mu} - 1 \leq \epsilon$ , implying that

$$\left\| 1 - \frac{d\nu}{d\mu} \right\|_{L^\infty(\mu)} \leq \epsilon.$$

Furthermore, taking convolution with  $e^{-|u|^2/2\sigma^2}$ , the inequalities between  $\mu$  and  $\nu$  also imply,

$$(1 - \epsilon) \int e^{-|w-u|^2/2\sigma^2} \mu(du) \leq \int e^{-|w-u|^2/2\sigma^2} \nu(du) \leq (1 + \epsilon) \int e^{-|w-u|^2/2\sigma^2} \mu(du),$$

and hence,

$$\left| \frac{\int e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} - 1 \right| \leq \epsilon.$$



Thus, we conclude that

$$\left| \frac{\int u e^{-|w-u|^2/2\sigma^2} \left[1 - \frac{d\nu}{d\mu}\right] \mu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} \right| \leq \frac{\int |u| e^{-|w-u|^2/2\sigma^2} \mu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} \left\| 1 - \frac{d\nu}{d\mu} \right\|_{L^\infty(\mu)} \leq M\epsilon,$$

and

$$\left| D^\nu(w; \sigma) \left( \frac{\int e^{-|w-u|^2/2\sigma^2} \nu(du)}{\int e^{-|w-u|^2/2\sigma^2} \mu(du)} - 1 \right) \right| \leq |D^\nu(w; \sigma)|\epsilon \leq M\epsilon,$$

where we have used the fact that  $\text{supp}(\mu), \text{supp}(\nu) \subset \{|u| \leq M\}$  in both estimates. Combining these estimates, we conclude that

$$\|D^\mu(\cdot; \sigma) - D^\nu(\cdot; \sigma)\|_{L^\infty(\mathbb{R}^d)} \leq 2M\epsilon,$$

as claimed.  $\square$

*Proof of Lemma 7.18.* Due to the problem setup, it is easy to see that the optimal conditional denoiser  $D_{\text{opt}}(u_\sigma; \bar{u}, \sigma)$  must be a shift by  $m(\bar{u})$  of the optimal denoiser for the uniform data distribution  $p(u) = \mathcal{U}[-1, 1]$  over  $[-1, 1]$ . It will therefore suffice to prove the statement for the optimal denoiser  $D_{\text{opt}}(u; \sigma)$  corresponding to this data distribution. We want to show:

1.  $D_{\text{opt}}(u; \sigma)$  is  $L^*$ -Lipschitz, uniformly as  $\sigma \rightarrow 0$ ,
2.  $\lim_{\sigma \rightarrow 0} D_{\text{opt}}(u; \sigma) = g(u)$ , given by (114).

To prove property (2.) we simply note that, by Lemma 7.9, the optimal denoiser converges to the closest point in the support of the data distribution  $p(u) = \mathcal{U}[-1, 1]$ . The formula (114) is then immediate.

It remains to prove the uniform  $L^*$ -Lipschitz bound. To this end, we recall that, by Lemma 7.9,  $D_{\text{opt}}(w; \sigma)$  is given by

$$D_{\text{opt}}(w; \sigma) = \frac{\int_{-1}^{+1} u e^{-(u-w)^2/2\sigma^2} du}{\int_{-1}^{+1} e^{-(u-w)^2/2\sigma^2} du}.$$

If we denote  $\underline{u} = D_{\text{opt}}(w; \sigma)$  for simplicity, then a short calculation implies that

$$|D'_{\text{opt}}(w; \sigma)| = \frac{\int_{-1}^{+1} (u - \underline{u})^2 e^{-(u-w)^2/2\sigma^2} du}{\sigma^2 \int_{-1}^{+1} e^{-(u-w)^2/2\sigma^2} du}.$$

We will prove that  $|D'_*(w; \sigma)| \leq L^*$  is uniformly bounded in  $\sigma$  and  $w$ . From the above formula, this is immediate for large  $\sigma$ ; e.g. for  $\sigma > 2$ , we have

$$|D'_{\text{opt}}(w; \sigma)| = \frac{\int_{-1}^{+1} (u - \underline{u})^2 e^{-(u-w)^2/2\sigma^2} du}{\sigma^2 \int_{-1}^{+1} e^{-(u-w)^2/2\sigma^2} du} \leq \frac{\int_{-1}^{+1} 4e^{-(u-w)^2/2\sigma^2} du}{4 \int_{-1}^{+1} e^{-(u-w)^2/2\sigma^2} du} = 1, \quad (\sigma > 2). \quad (115)$$

To establish an upper bound for  $\sigma \in (0, 2]$ , we will distinguish between exterior points  $\{w < -1\}, \{w > +1\} \subset [-1, 1]^c$ , and interior points  $\{-1 < w < +1\} \subset [-1, 1]$ .

**Exterior:** We first consider the exterior domain  $\{w < -1\}$ . Fix  $\xi > 0$ , and set  $w = -1 - \xi$ . Our goal is to bound  $D'_{\text{opt}}(-1 - \xi; \sigma)$  for all  $\xi > 0$  and  $\sigma \in (0, 2]$ . Under the current assumptions, we have

$$|D'_{\text{opt}}(-1 - \xi; \sigma)| = \frac{\int_{-1}^{+1} (u - \underline{u})^2 e^{-(u+1+\xi)^2/2\sigma^2} du}{\sigma^2 \int_{-1}^{+1} e^{-(u+1+\xi)^2/2\sigma^2} du}$$

After a change of variables  $u \rightarrow u - 1$  and noting that  $\underline{u}$  minimizes the quadratic variation, it follows that

$$|D'_{\text{opt}}(-1 - \xi; \sigma)| \leq \frac{\int_0^2 u^2 e^{-(u+\xi)^2/2\sigma^2} du}{\sigma^2 \int_0^2 e^{-(u+\xi)^2/2\sigma^2} du}.$$

Expanding  $(u + \xi)^2 = u(u + 2\xi) + \xi^2$ , we can write

$$|D'_*(-1 - \xi; \sigma)| \leq \frac{\int_0^2 u^2 e^{-u(u+2\xi)/2\sigma^2} du}{\sigma^2 \int_0^2 e^{-u(u+2\xi)/2\sigma^2} du} \leq \frac{\int_0^\infty u^2 e^{-u(u+2\xi)/2\sigma^2} du}{\sigma^2 \int_0^\infty e^{-u(u+2\xi)/2\sigma^2} du}$$

It will be convenient to estimate the denominator in terms of an integration over  $[0, \infty)$  instead of  $[0, 2]$ . To this effect, we note that

$$\int_2^\infty e^{-u(u+2\xi)/2\sigma^2} du = \int_0^\infty e^{-(2+u)(2+u+2\xi)/2\sigma^2} du \leq e^{-2/\sigma^2} \int_0^\infty e^{-u(u+2\xi)/2\sigma^2} du.$$

And thus,

$$\int_0^2 e^{-u(u+2\xi)/2\sigma^2} du = \int_0^\infty e^{-u(u+2\xi)/2\sigma^2} du - \int_2^\infty e^{-u(u+2\xi)/2\sigma^2} du \geq (1 - e^{-2/\sigma^2}) \int_0^\infty e^{-u(u+2\xi)/2\sigma^2} du.$$

It follows that

$$|D'_*(-1 - \xi; \sigma)| \leq \frac{\mathcal{I}}{1 - e^{-2/\sigma^2}}, \quad \mathcal{I} := \frac{\int_0^\infty u^2 e^{-u(u+2\xi)/2\sigma^2} du}{\sigma^2 \int_0^\infty e^{-u(u+2\xi)/2\sigma^2} du}.$$

We note that  $1 - e^{-2/\sigma^2} \geq 1 - e^{-1/2} > 0$  is uniformly lower bounded for all  $\sigma \in (0, 2]$ . Therefore, to prove a uniform upper bound on  $|D'_*(-1 - \xi; \sigma)|$ , it suffices to upper bound  $\mathcal{I}$ .

We now introduce  $z := \xi/\sigma^2$ , perform a change of variables  $u \rightarrow \sigma u$ , and write

$$\mathcal{I} = \frac{\int_0^\infty u^2 e^{-u(u+2z)/2} du}{\int_0^\infty e^{-u(u+2z)/2} du}.$$

Let us introduce  $x = u(u + 2z)$ , so that

$$u = \sqrt{x + z^2} - z = \frac{x}{\sqrt{x + z^2} + z},$$

and  $dx = 2(u + z)du = 2\sqrt{x + z^2} du$ . Then,

$$\mathcal{I} = \frac{\int_0^\infty \frac{x^2 e^{-x/2} dx}{(\sqrt{x + z^2} + z)^2 \sqrt{x + z^2}}}{\int_0^\infty \frac{e^{-x/2} dx}{\sqrt{x + z^2}}}.$$

To bound this independently of  $z \geq 0$ , we first consider  $z \in [0, 1]$ , and set  $z = 0$  in the numerator,  $z = 1$  in the denominator, to obtain the uniform bound:

$$\mathcal{I}|_{z \in [0,1]} \leq \frac{\int_0^\infty \sqrt{x} e^{-x/2} dx}{\int_0^\infty \frac{e^{-x/2} dx}{\sqrt{x+1}}}.$$

For  $z \geq 1$ , we observe that,

$$\int_0^\infty \frac{x^2 e^{-x/2} dx}{(\sqrt{x+z^2}+z)^2 \sqrt{x+z^2}} \leq z^{-3} \int_0^\infty x^2 e^{-x/2} dx,$$

and

$$\int_0^\infty \frac{e^{-x/2} dx}{\sqrt{x+z^2}} \geq \frac{1}{\sqrt{1+z^2}} \int_0^1 e^{-x/2} dx \geq (2z)^{-1} \int_0^1 e^{-x/2} dx.$$

It follows that

$$\mathcal{I}|_{z \in [1,\infty)} \leq \frac{2 \int_0^\infty x^2 e^{-x/2} dx}{z^2 \int_0^1 e^{-x/2} dx}.$$

The last term is uniformly bounded for  $z \in [1, \infty)$ , and  $\lesssim z^{-2}$  as  $z \rightarrow \infty$ . Recalling that  $z = \xi/\sigma^2$  and  $\xi = |w| - 1$ , these two estimates on  $\mathcal{I}$  imply an upper bound of the form,

$$|D'_{\text{opt}}(w, \sigma)| \leq C \left( \frac{\sigma^2}{|w| - 1 + \sigma^2} \right)^2, \quad (|w| > 1, \sigma \in (0, 2]). \quad (116)$$

Technically, we have only proved the above bound for  $w < -1$ . However, the same upper bound also holds for  $w > +1$ , by symmetry. From (116), we in fact observe that in the exterior domain, we have  $D'_{\text{opt}}(w; \sigma) \rightarrow 0$  except potentially at the boundary points  $\{-1, +1\}$ . This is consistent with the fact that  $D_{\text{opt}}(w; \sigma) \rightarrow \pm 1$  for  $|w| > 1$ .

**Interior:** Our final goal is to bound  $D'_{\text{opt}}(w; \sigma)$  in the interior, i.e. for all  $w \in (-1, 1)$ . By symmetry about the origin, we may in fact assume that  $w \in (-1, 0)$ . Under these assumptions, we have

$$|D'_{\text{opt}}(w; \sigma)| \leq \frac{\int_{-1}^+ (u-w)^2 e^{-(u-w)^2/2\sigma^2} du}{\sigma^2 \int_{-1}^+ e^{-(u-w)^2/2\sigma^2} du}.$$

Making the change of variables  $u \rightarrow u - w$  and noting the set inclusion  $w + [0, 1] \subset [-1, 1]$ , it follows that

$$|D'_{\text{opt}}(w; \sigma)| \leq \frac{\int_{-\infty}^\infty u^2 e^{-u^2/2\sigma^2} du}{\sigma^2 \int_0^{+1} e^{-u^2/2\sigma^2} du}$$

Making the change of variables  $\eta = u/\sigma$ , and recalling that we consider  $\sigma \in (0, 2]$ , we obtain

$$|D'_{\text{opt}}(w; \sigma)| = \frac{\int_{-\infty}^\infty \eta^2 e^{-\eta^2/2} d\eta}{\int_0^{1/\sigma} e^{-\eta^2/2} d\eta} \leq \frac{\int_{-\infty}^\infty \eta^2 e^{-\eta^2/2} d\eta}{\int_0^{1/2} e^{-\eta^2/2} d\eta}, \quad (w \in [-1, 1], \sigma \in (0, 2]). \quad (117)$$

The right-hand side is independent of  $w \in (-1, 1)$  and  $\sigma \in (0, 2]$ . Combining (115), (116) and (117), we have derived a uniform upper bound  $|D'_{\text{opt}}(w; \sigma)| \leq L^*$ , as desired.  $\square$

*Proof of Lemma 7.19.* Fix any  $\phi \in \text{Lip}_1(\mathbb{R} \times \mathbb{R})$ , such that  $\phi(0,0) = 0$ . We note that by Kantorovich duality,  $W_1(\mu^\Delta, \mu)$  is the supremum of

$$\mathcal{R}^\Delta(\phi) = \int \phi(u, \bar{u}) d\mu^\Delta - \int \phi(u, \bar{u}) d\mu,$$

over all such  $\phi$ . By definition of  $\mu^\Delta$  and  $\mu$ , we have

$$\mathcal{R}^\Delta(\phi) = \int_0^1 \phi(\mathcal{S}^\Delta(\bar{u}), \bar{u}) d\bar{u} - \int_0^1 \int_{-1}^1 \phi(m(\bar{u}) + y, \bar{u}) dy d\bar{u}.$$

To estimate  $\mathcal{R}^\Delta(\phi)$  from above, we first observe that for any bounded function  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,

$$\int_0^1 f(\bar{u}) d\bar{u} = \int_0^1 \mathbb{E}_{\eta \sim \mathcal{U}([0, \epsilon])} [f(\bar{u} + \eta)] d\bar{u} + r(f, \epsilon),$$

where the remainder  $r(f; \epsilon)$  can be bounded by  $|r(f; \epsilon)| \leq 2\epsilon \|f\|_{L^\infty}$ . This follows from the fact that, for uniformly distributed  $\bar{u} \in [0, 1]$  and  $\eta \in [0, \epsilon]$ , the sum  $\bar{u} + \eta$  has uniform density  $\equiv 1$  over  $[\epsilon, 1]$ , and is supported on  $[0, 1 + \epsilon]$ .

Thus, taking  $\bar{u} \sim \mathcal{U}[0, 1]$ ,  $\eta \sim \mathcal{U}[0, \epsilon]$ , it follows that

$$\int_0^1 \phi(\mathcal{S}^\Delta(\bar{u}), \bar{u}) d\bar{u} \leq \mathbb{E}_{\bar{u}} \mathbb{E}_\eta \phi(\mathcal{S}^\Delta(\bar{u} + \eta), \bar{u} + \eta) + 2\|\phi\|_{L^\infty} \epsilon.$$

We next recall that

$$\mathcal{S}^\Delta(\bar{u} + \eta) = m(\bar{u} + \eta) + s_N(\bar{u} + \eta).$$

The first term is  $L^*$ -Lipschitz continuous, implying that

$$\mathcal{S}^\Delta(\bar{u} + \eta) = m(\bar{u}) + s_N(\bar{u} + \eta) + O_1(L^* \epsilon),$$

with implied constant for the remainder term bounded by 1. Let us introduce,

$$\tilde{\phi}(\bar{u}, y) := \phi(m(\bar{u}) + y, \bar{u}).$$

The last bound combined with the 1-Lipschitz continuity of  $\phi$ , then implies that

$$\begin{aligned} \phi(\mathcal{S}^\Delta(\bar{u} + \eta), \bar{u} + \eta) &= \phi(m(\bar{u}) + s_N(\bar{u} + \eta), \bar{u}) + O(L^* \epsilon) \\ &= \tilde{\phi}(\bar{u}, s_N(\bar{u} + \eta)) + O(L^* \epsilon), \end{aligned}$$

where the only dependence on  $\phi$  of the last term is via the Lipschitz bound  $L^*$ . The importance of this last expressions is that, if  $\eta \sim \mathcal{N}[0, \epsilon]$  and if  $N\epsilon \in \mathbb{N}$  is integer, then the push-forward,

$$y := s_N(\bar{u} + \eta) = \Lambda(N\bar{u} + N\eta),$$

has uniform distribution  $y \sim \mathcal{U}[-1, 1]$ , independent of  $\bar{u}$ . Thus, we may choose  $\epsilon := 1/N = \Delta$ , for which it then follows that

$$\begin{aligned} \int_0^1 \phi(\mathcal{S}^\Delta(\bar{u}), \bar{u}) d\bar{u} &= \mathbb{E}_{\bar{u}} \mathbb{E}_\eta \phi(\mathcal{S}^\Delta(\bar{u} + \eta), \bar{u} + \eta) + O(\|\phi\|_{L^\infty} \Delta) \\ &= \mathbb{E}_{\bar{u}} \mathbb{E}_\eta \tilde{\phi}(\bar{u}, s_N(\bar{u} + \eta)) + O((L^* + \|\phi\|_{L^\infty}) \Delta) \\ &= \mathbb{E}_{\bar{u}} \mathbb{E}_y \tilde{\phi}(\bar{u}, y) + O((L^* + \|\phi\|_{L^\infty}) \Delta) \\ &= \int_0^1 \int_{-1}^1 \tilde{\phi}(\bar{u}, y) dy d\bar{u} + O((L^* + \|\phi\|_{L^\infty}) \Delta). \end{aligned}$$

Recalling the definition of  $\tilde{\phi}(\bar{u}, y) = \phi(m(\bar{u}) + y, \bar{u})$ , it follows that

$$\begin{aligned}\mathcal{R}^\Delta(\phi) &= \int_0^1 \phi(\mathcal{S}^\Delta(\bar{u}), \bar{u}) d\bar{u} - \int_0^1 \int_{-1}^1 \phi(m(\bar{u}) + y, \bar{u}) dy d\bar{u} \\ &= O((L^* + \|\phi\|_{L^\infty}) \Delta).\end{aligned}$$

Taking the supremum over all  $\phi \in \text{Lip}_1$  such that  $\phi(0, 0) = 0$ , we conclude that

$$W_1(\mu^\Delta, \mu) = \sup_{\phi} \mathcal{R}^\Delta(\phi) \leq CL^* \Delta.$$

This proves the claim.  $\square$

#### 7.4.6 Proofs for Section 7.3

We next provide the proof of Proposition 7.7 and Proposition 7.8.

**Deterministic Setting.** We start with the proof of Proposition 7.7.

*Proof of Proposition 7.7.* The proof is very similar to the proof of Proposition 7.5. We first note that  $\mathcal{S}^{(k)}(h)$  is highly oscillatory and has mean zero, implying that

$$\tilde{\Psi}^{(k)}(h) = \mathbb{E}_{\delta h} [\mathcal{S}^{(k)}(h + \delta h)] \rightarrow 0,$$

in  $L^2([0, 1])$ . To complete the proof, it will thus suffice to show,

$$\lim_{k \rightarrow \infty} \mathbb{E}_{h \sim \bar{\mu}} \|\Psi^{(k)}(h)\|^2 = 0.$$

To this end, we simply note that

$$\mathbb{E}_{h \sim \bar{\mu}} \|\Psi^{(k)}(h)\|^2 = \mathbb{E}_{h \sim \bar{\mu}} \|\Psi^{(k)}(h) - \mathcal{S}^{(k)}(h)\|^2 + 2\mathbb{E}_{h \sim \bar{\mu}} \langle \Psi^{(k)}(h), \mathcal{S}^{(k)}(h) \rangle - \mathbb{E}_{h \sim \bar{\mu}} \|\mathcal{S}^{(k)}(h)\|^2.$$

Since  $\Psi^\Delta$  minimizes the first term over all  $L^*$ -Lipschitz functions, we can compare with the 0 function to obtain,

$$\begin{aligned}\mathbb{E}_{h \sim \bar{\mu}} \|\Psi^{(k)}(h)\|^2 &\leq \mathbb{E}_{h \sim \bar{\mu}} \|0 - \mathcal{S}^{(k)}(h)\|^2 + 2\mathbb{E}_{h \sim \bar{\mu}} \langle \Psi^{(k)}(h), \mathcal{S}^{(k)}(h) \rangle - \mathbb{E}_{h \sim \bar{\mu}} \|\mathcal{S}^{(k)}(h)\|^2 \\ &= 2\mathbb{E}_{h \sim \bar{\mu}} \langle \Psi^{(k)}(h), \mathcal{S}^{(k)}(h) \rangle.\end{aligned}$$

It is straight-forward to show that there exists a constant  $C > 0$ , such that

$$\sup_{\text{Lip}(\Psi) \leq L^*} \mathbb{E}_{h \sim \bar{\mu}} \langle \Psi(h), \mathcal{S}^{(k)}(h) \rangle \leq \frac{C}{k}.$$

Thus, we conclude that  $\mathbb{E}_{h \sim \bar{\mu}} \|\Psi^{(k)}(h)\|^2 \leq C/k \rightarrow 0$ , as claimed.  $\square$

**Probabilistic Setting.** We now consider the probabilistic setting of conditional diffusion models. Our asymptotic results will hold for any  $\tilde{\epsilon} > 0$ . We thus assume  $\tilde{\epsilon}$  to be fixed (arbitrarily). We recall that,

$$\nu^{(k)}(du | h) := \text{Law}_{\delta h} \left[ \mathcal{S}^{(k)}(h + \delta h) \right], \quad \delta h \sim \mathcal{U}([-\tilde{\epsilon}, \tilde{\epsilon}]),$$

and  $\tilde{D}^{(k)}(u_\sigma; h, \sigma)$  denotes the optimal unconstrained conditional denoiser for  $\nu^{(k)}$ . The derivation of Proposition 7.8 is more involved, so we will first give an overview of the essential ingredients, and leave proofs for the next subsection.

Our first result shows that  $\nu^{(k)} \approx \mathcal{U}(\mathbb{S}^1)$  is approximately equivalent to a uniform distribution:

**Lemma 7.20.** *Let  $\mu(du | \bar{u}) = \mathcal{U}(\mathbb{S}^1)$  be a uniform measure. There exists a constant  $C > 0$ , independent of  $k$ , such that*

$$(1 - Ck^{-1})\mu(du | h) \leq \nu^{(k)}(du | h) \leq (1 + Ck^{-1})\mu(du | h).$$

Since the proof is completely analogous to the proof of Lemma 7.15, we will not discuss the details in this appendix. The result of the last lemma again allows us to easily identify the limit  $D_{\text{opt}} = \lim_{k \rightarrow \infty} \tilde{D}^{(k)}$ , as an immediate consequence of Lemma 7.16 and Lemma 7.20:

**Lemma 7.21.** *Let  $\mu = \mathcal{U}(\mathbb{S}^1) \otimes \mathcal{U}([0, 1])$  be the uniform measure on  $(u, h) \in \mathbb{S}^1 \times [0, 1]$ . Let  $D_{\text{opt}}$  denote the optimal (unconstrained) conditional denoiser for  $\mu$ . Then we have,*

$$\mathbb{E}_{(u, h) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| \tilde{D}^{(k)}(u + \eta; h, \sigma) - D_{\text{opt}}(u + \eta; h, \sigma) \right\|^2 \leq Ck^{-1}.$$

Due to the simplicity of  $\mu$ , the optimal denoiser  $D_{\text{opt}}$  can be computed explicitly, as shown next:

**Lemma 7.22.** *The optimal denoiser for  $\mu = \mathcal{U}(\mathbb{S}^1) \otimes \mathcal{U}([0, 1])$ , is given by*

$$D_{\text{opt}}(u; h, \sigma) = g_\sigma(|u|) \frac{u}{|u|}, \tag{118}$$

where  $g_\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is given by  $g_\sigma(t) = I_1(t/\sigma^2)/I_0(t/\sigma^2)$ , with  $I_\alpha(z)$  the modified Bessel function of the first kind and of order  $\alpha$ ,

$$I_\alpha(z) = \frac{1}{\pi} \int_0^\pi \cos(\alpha\theta) e^{z \cos(\theta)} d\theta \sim \frac{e^z}{\sqrt{2\pi z}}, \quad \text{as } z \rightarrow \infty.$$

In particular,

$$\lim_{\sigma \rightarrow 0} D_{\text{opt}}(u; h, \sigma) = \frac{u}{|u|}.$$

We include the details of the required calculation to prove Lemma 7.22 in the next subsection. As is clear from the limiting behavior as  $\sigma \rightarrow 0$ , we cannot have a uniform Lipschitz bound at the origin  $u = 0$  in this case. Thus, we need to better understand the (local) Lipschitz behavior of  $D_{\text{opt}}$  in this limit.

**Lemma 7.23.** *Assume  $D : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is of the form  $D(u) = g(|u|) \frac{u}{|u|}$ . Let  $A_\delta := \{|u| \geq \delta\}$ . Then for any  $\delta > 0$ ,*

$$\text{Lip}(D|_{A_\delta}) \leq \text{Lip}(g|_{[\delta, \infty)}) + \frac{2\|g\|_{L^\infty}}{\delta}.$$

Given the explicit form of  $D_{\text{opt}}$ , we want to apply the last lemma with  $g = g_\sigma$  to understand the Lipschitz behavior of  $D_{\text{opt}}$  near the origin. It remains to bound the Lipschitz constant of  $g_\sigma$  on  $[\delta, \infty)$ . This is the subject of the next lemma.

**Lemma 7.24.** *Let  $g_\sigma(t) = g(t/\sigma^2)$  where  $g(z) := I_1(z)/I_0(z)$  is a quotient of modified Bessel functions of the first kind. There exists a constant  $C > 0$ , such that for any  $\delta > 0$ ,*

$$\text{Lip}(g_\sigma|_{[\delta, \infty)}) \leq \frac{C}{\delta}.$$

The following corollary is now immediate:

**Corollary 7.25.** *Let  $D_{\text{opt}}(u; \sigma) = g_\sigma(|u|) \frac{u}{|u|}$  be the optimal denoiser for  $\mu = \mathcal{U}(\mathbb{S}^1)$ . There exists a constant  $C > 0$ , such that for any  $\delta, \sigma > 0$ , and with  $A_\delta := \{|u| \geq \delta\}$ :*

$$\text{Lip}(D_{\text{opt}}|_{A_\delta}) \leq C \min\left(\frac{1}{\delta}, \frac{1}{\sigma^2}\right). \quad (119)$$

The last corollary is the first ingredient required to apply Proposition 7.12. The second ingredient is contained in the following lemma:

**Lemma 7.26.** *If  $\mu \in \mathcal{P}(\mathbb{R}^2)$  is a probability measure with  $\text{supp } \mu \subset \mathbb{S}^1$  and if  $\delta \in (0, 1/2]$ , then*

$$\text{Prob}_{\mu_\sigma} [B_\delta(0)] \leq \frac{\delta^2}{2\sigma} e^{-1/8\sigma^2},$$

where  $\mu_\sigma = \mu * \mathcal{N}(0, \sigma^2)$ .

Given the above results, we can now finally come to the proof of Proposition 7.8.

*Proof of Proposition 7.8.* We recall that our goal is to show that

$$\limsup_{k \rightarrow \infty} \mathbb{E}_{(u, h) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^{(k)}(u + \eta; h, \sigma) - \tilde{D}^{(k)}(u + \eta; h, \sigma) \right\|^2 \leq C e^{-L^*/8C}.$$

Lemma 7.21 shows that  $\tilde{D}^{(k)} \rightarrow D_{\text{opt}}$  with  $D_{\text{opt}}$  the conditional diffusion model for  $\mu$ . It will thus be enough to show that

$$\limsup_{k \rightarrow \infty} \mathbb{E}_{(u, h) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^{(k)}(u + \eta; h, \sigma) - D_{\text{opt}}(u + \eta; h, \sigma) \right\|^2 \leq C e^{-L^*/8}.$$

Let

$$\mathcal{E} := \limsup_{k \rightarrow \infty} \mathbb{E}_{(u, h) \sim \mu} \mathbb{E}_{\eta \sim \mathcal{N}(0, \sigma^2)} \left\| D^{(k)}(u + \eta; h, \sigma) - D_{\text{opt}}(u + \eta; h, \sigma) \right\|^2.$$

By Proposition 7.12, applied with  $A := A_\delta$  and  $\mathbb{R}^d \setminus A = B_\delta(0)$ , we have the upper bound,

$$\mathcal{E} \leq \limsup_{k \rightarrow \infty} C \left\{ L^* W_1(\mu^{(k)}, \mu) + \text{Prob}_{\mu_\sigma} [B_\delta(0)] + \text{Prob}_{\mu_\sigma^{(k)}} [B_\delta(0)] \right\},$$

under the constraint that  $\text{Lip}(D_{\text{opt}}|_{A_\delta}) \leq L^*$ . By Corollary 7.25, this is the case provided that  $C \min(\delta^{-1}, \sigma^{-2}) \leq L^*$ . An argument completely analogous to the proof of Lemma 7.19 then shows that  $\lim_{k \rightarrow \infty} W_1(\mu^{(k)}, \mu) = 0$ . And Lemma 7.26 implies that

$$\mathcal{E} \leq \frac{C\delta^2}{\sigma} e^{-(1-\delta)^2/2\sigma^2}.$$

Again, we emphasize our constraint on  $C \min(\delta^{-1}, \sigma^{-2}) \leq L^*$ . There are two cases, either (i)  $\sigma^2 \geq C/L^2$  and we are free to let  $\delta \rightarrow 0$ , or (ii)  $\sigma^2 < C/L^*$  and we will choose  $\delta = C/L^*$ . In the first case, we obtain

$$\mathcal{E} = 0.$$

In the second case, we have  $1/\sigma^2 \geq L^*/C$ , and we obtain

$$\mathcal{E} \leq \frac{C}{(L^*)^{3/2}} e^{-L^*/8C},$$

for sufficiently large  $L^*$ . It follows that, independent of  $\sigma$ , we have the following upper bound,

$$\mathcal{E} \leq C e^{-L^*/8C}.$$

In particular, it follows that

$$\limsup_{k \rightarrow \infty} \mathbb{E}_{h,u,\eta} \|D^{(k)}(u_\sigma; h, \sigma) - \tilde{D}^{(k)}(u_\sigma; h, \sigma)\|^2 \leq C e^{-L^*/8C}.$$

□

**Proofs of Lemma 7.22, Lemma 7.23, Lemma 7.24 and Lemma 7.26.** In the following we detail the proofs of Lemma 7.22, Lemma 7.23, Lemma 7.24 and Lemma 7.26.

*Proof of Lemma 7.22.* By the explicit formula for  $D_{\text{opt}}$ , we have

$$D_{\text{opt}}(w; h, \sigma) = \frac{\int u \exp(-(u-w)^2/2\sigma^2) p(u|h) du}{\int \exp(-(u-w)^2/2\sigma^2) p(u|h) du}.$$

Since the joint probability on  $(h, u)$  is a product measure, the conditioning on  $h$  is irrelevant, and  $p(u|h) = \mathcal{U}(\mathbb{S}^1)$ . By the symmetries of the problem, the expectation perpendicular to  $w$  vanishes, so that we can write

$$D_{\text{opt}}(w; h, \sigma) = g_\sigma(|w|) \frac{w}{|w|}.$$

For the computation of  $g_\sigma(|w|)$ , we may wlog assume that  $w = (|w|, 0)$  points in the first coordinate direction. We dot the above formula for  $D_{\text{opt}}$  by  $w/|w|$ , and parametrize  $u = (\cos \theta, \sin \theta)$  by the angular variable  $\theta \in [0, 2\pi]$ . It follows that

$$g_\sigma(|w|) = \frac{\int_0^{2\pi} \cos \theta \exp(-\{(\cos \theta - |w|)^2 + \sin^2 \theta\}/2\sigma^2) d\theta}{\int_0^{2\pi} \exp(-\{(\cos \theta - |w|)^2 + \sin^2 \theta\}/2\sigma^2) d\theta}.$$



After expansion of the squares and elementary simplifications, we obtain

$$g_\sigma(|w|) = \frac{\int_0^\pi \cos \theta e^{|w| \cos(\theta)/\sigma^2} d\theta}{\int_0^\pi e^{|w| \cos(\theta)/\sigma^2} d\theta} = \frac{I_1(|w|/\sigma^2)}{I_0(|w|/\sigma^2)}.$$

This is the claimed formula.  $\square$

*Proof of Lemma 7.23.* Given  $u, v \in \mathbb{R}^d$ , we have

$$\begin{aligned} |D(u) - D(v)| &\leq \left| g(|u|) \frac{u}{|u|} - g(|v|) \frac{u}{|u|} \right| + \left| g(|v|) \left( \frac{u}{|u|} - \frac{v}{|v|} \right) \right| \\ &\leq \text{Lip}(g|_{[\delta, \infty)}) |u - v| + \|g\|_{L^\infty} \left| \frac{u}{|u|} - \frac{v}{|v|} \right|. \end{aligned}$$

To estimate the last term, we note that

$$\begin{aligned} \left| \frac{u}{|u|} - \frac{v}{|v|} \right| &\leq \left| \frac{u}{|u|} - \frac{v}{|u|} \right| + \left| \frac{v}{|u|} - \frac{v}{|v|} \right| \\ &\leq \frac{1}{|u|} |u - v| + |v| \left| \frac{1}{|u|} - \frac{1}{|v|} \right| \\ &= \frac{1}{|u|} |u - v| + \frac{1}{|u|} ||v| - |u||. \end{aligned}$$

It thus follows that, for  $|u| \geq \delta$ ,

$$\left| \frac{u}{|u|} - \frac{v}{|v|} \right| \leq \frac{2}{\delta} |u - v|.$$

Substitution in the first inequality now yields the claimed bound on  $\text{Lip}(D|_{A_\delta})$ .  $\square$

*Proof of Lemma 7.24.* We have

$$\begin{aligned} \text{Lip}(g_\sigma|_{[\delta, \infty)}) &= \|g'_\sigma\|_{L^\infty([\delta, \infty))} \\ &= \frac{1}{\sigma^2} \|g'\|_{L^\infty([\delta/\sigma^2, \infty))} \\ &= \frac{1}{\delta} \frac{\delta}{\sigma^2} \|g'\|_{L^\infty([\delta/\sigma^2, \infty))} \\ &\leq \frac{1}{\delta} \|zg'(z)\|_{L^\infty([\delta/\sigma^2, \infty))} \\ &\leq \frac{1}{\delta} \|zg'(z)\|_{L^\infty([0, \infty))} \end{aligned}$$

The claim follows by observing that  $C := \|zg'(z)\|_{L^\infty([0, \infty))} < \infty$ . This last observation follows from the relationships  $I'_0(z) = I_1(z)$ ,  $I'_1(z) = I_0(z) - \frac{1}{z} I_1(z)$ , so that

$$g'(z) = \frac{I'_1(z)I_0(z) - I_1(z)I'_0(z)}{I_0(z)^2} = (1 - g(z)^2) - \frac{1}{z}g(z),$$

and the asymptotics of  $I_0(z)$ ,  $I_1(z)$  as  $z \rightarrow \infty$ ; Indeed, we have  $zg'(z) = z(1 - g(z)^2) - g(z)$ , and the asymptotic expansion of  $I_0(z)$ ,  $I_1(z)$  as  $z \rightarrow \infty$  implies that  $g(z) \rightarrow 1$  and  $1 - g(z)^2 \sim C/z$  for some constant  $C$ . Hence  $zg'(z)$  remains uniformly upper bounded as  $z \rightarrow \infty$ .  $\square$

*Proof of Lemma 7.26.* For any  $y \in \text{supp}(\mu)$  and  $x \in B_\delta(0)$ , we clearly have  $|x - y|^2 \geq (1 - \delta)^2$ . Hence,

$$\begin{aligned} \text{Prob}_{\mu_\sigma}[B_\delta(0)] &= \int \int 1_{B_\delta}(x) \frac{e^{-|x-y|^2/2\sigma^2}}{2\pi\sigma} \mu(dy) dx \\ &\leq \frac{e^{-(1-\delta)^2/2\sigma^2}}{2\pi\sigma} |B_\delta| = \frac{\delta^2}{2\sigma} e^{-(1-\delta)^2/2\sigma^2}. \end{aligned}$$

□

## 8 Further Experimental Results

In this section, we expand on the discussion about the experimental results in the Main Text by contextualizing additional results and better highlighting the ones briefly discussed in the Main Text.

### 8.1 GenCFD Generates Very High-quality Samples of the Flow

We start by recalling the ability of GenCFD to generate very high-quality samples, drawn from the conditional distribution, which was already discussed in the Main Text. To this end, in Fig. 2 (A) of the Main Text and Fig. 6 here, we present samples of the conditional kinetic energy (square of the norm of the velocity field), corresponding to the TG and CSF datasets, generated by GenCFD and compare them to ground-truth samples generated with the underlying CFD simulator. We observe from these figures that the pointwise kinetic energy samples are very realistic for both datasets, with very little to distinguish them visually from the ground truth samples. In particular, the small-scale eddies are captured very well in the generated velocity fields, and the diffusion model also provides a rich diversity of samples despite each of them corresponding to the *same initial condition*.

An even harder test of sample quality is investigated by computing the *vorticity* from the generated velocity fields by taking a *curl* and plotting the resulting pointwise vorticity intensity samples in Fig. 2 (B) of the Main Text and Fig. 7 here, for TG and CSF, respectively. When compared to the ground truth, we find the computed vorticity samples to be very accurate with a realistic rendition of small-scale features for both datasets, particularly of looping vortex tubes which are the characteristics of a turbulent fluid [46]. These realistic vorticity profiles, generated by our AI algorithm, are particularly impressive as the model itself has never been shown a vorticity field and has to infer it indirectly from the generated velocity field by taking its derivatives. This suggests that the covariate structures of the velocity fields are well captured by the AI algorithm.

In contrast to the high quality of samples generated by GenCFD, all the other ML baselines (see Figs. 2 (A and B) of the Main Text and Figs. 6 and 7, for C-FNO, which is the strongest baseline) only lead to very poor quality as well as very little diversity in terms of the generated samples, both for the kinetic energy and for the vorticity. In general, the samples generated by these baselines collapse to a field close to the mean velocity (and vorticity) field, rather than representing the statistical spread of the target conditional distribution.

This observation of very high-quality sample generation with GenCFD is further reinforced in Figs. 28 to 30, where samples with other initial conditions for the CSF dataset are presented.

Moreover, in Fig. 11, we present samples of the density for the CSI dataset, generated by GenCFD and compare it to ones obtained from the ground truth and C-FNO to observe that this high quality of sample generation by GenCFD is also present for compressible fluid flow. In particular, we observe that GenCFD is able to generate the leading shock wave, the rising

turbulent plume in its wake, and also the small-scale eddies marking the turbulent mixing zone near the trailing shocks. On the other hand, C-FNO is able to approximate the leading shock wave accurately but smoothens out the rising plume while completely failing to generate the small-scale turbulent eddies in the wake.

In Figs. 14 and 15, we present the pointwise kinetic energy and vorticity intensity, respectively, of the nozzle flow experiment. Even for this experiment that is prototypical of real-world engineering flows, we see that GenCFD provides very high quality and diverse samples of the flow whereas the best performing baseline (UViT) regresses to the mean. In particular, the ability of GenCFD to generate this very complex flow, with a very complicated distribution of eddies that are both wall-bounded and yet have a freestream component, is noteworthy.

Finally, in Fig. 18, we visualize the x-component of the velocity field in the convective boundary layer (CBL) experiment to again find that GenCFD is able to generate realistic, diverse samples of the flow field, whereas the best performing baseline (UViT) smears out all the detailed flow features, especially the plumes going up and down due to convection.

## 8.2 GenCFD Accurately Approximates Statistical Quantities of Interest

The high quality of AI-generated samples of fluid flow encourages us to test how well GenCFD approximates the statistical quantities of interest. We check this by computing the mean and the standard deviation of the conditional velocity field, generated by GenCFD and comparing them with the underlying ground truth and the statistics of the ML baselines. The results for all the datasets are presented in Main Text Fig. 2 (C and D) for TG, Figs. 8, and 9 for CSF, Fig. 12 for CSI, Fig. 16 and Main Text Fig. 3 (A) for NF and Fig. 19 and Main Text Fig. 3 (C) for CBL. We observe from these figures that GenCFD approximates the mean and variance very well. In contrast, the ML baselines (we present figures for the best performing baseline in each dataset) can approximate the mean fairly accurately but entirely fail at approximating the standard deviation. This (approximate) collapse to mean for the ML baselines is also seen when we plot the one-point PDFs in Main Text Fig. 2 (E) for TG, Fig. 10 for CSF, Fig. 13 for CSI, Fig. 17 for NF and Fig. 20 for CBL. In complete contrast, GenCFD very accurately and impressively approximates the underlying point PDFs. We would like to point out that this ability of GenCFD to accurately predict the PDFs is particularly noteworthy as the spread out PDFs have to be generated from inputs (initial conditions) that are (approximately) Dirac distributions.

This accurate approximation of statistical quantities of interest with GenCFD is not just qualitative but also quantitative. We demonstrate this accuracy by presenting the errors in computing the mean, the standard deviation and as well as (the first marginal of) the 1-Wasserstein distance between the conditional distributions and the CRP Scores (CRPS), computed by GenCFD and other ML baselines, and the ground truth computed with the CFD solvers in Tables 6, 7, 8, 9 and 10 for TG, CSF, CSI, NF and CBL, respectively. We see from these tables that GenCFD is *significantly more accurate* than the ML baselines, particularly with respect to the standard deviation and the Wasserstein distance with gains ranging up to one

order of magnitude for the Wasserstein distance and the standard deviation and demonstrating the ability of GenCFD for accurate statistical computation of turbulent fluid flows.

### 8.3 GenCFD Provides Excellent Spectral Resolution

Energy spectra are key quantities of interest for the theoretical, experimental and computational study of turbulent fluid flows [19]. In particular, the famous K41 theory of turbulence is based on predicting the decay of these spectra with respect to wave number. Hence, spectral resolutions are often used as proxies for judging the quality of physics- [23] or AI-based [6] simulators of turbulent fluid flows. We compute the energy spectra for the GenCFD and baseline generated fields for all the datasets and plot them in Main Text Fig. 2 (F) for TG and Fig. 23 here for the rest of the datasets. We clearly observe from these figures that the spectral accuracy of GenCFD is excellent and the energy content, up to the highest frequencies, is accurately generated. On the other hand, the deterministic ML baselines are only able to generate a small fraction of the spectrum accurately and lose spectral resolution very fast. This poor effective spectral resolution of deterministic ML models has also been observed in the context of weather and climate modeling, see for instance Fig. 1 of [6].

### 8.4 GenCFD Scales with Data

A key attribute of modern AI models is their ability to scale with data [30]. To test this, we compute the errors in standard deviation and with respect to the 1-Wasserstein metric for GenCFD as the number of training samples for the CSF dataset varies and plot the results in Fig. 22. We observe from this figure that these test errors with GenCFD decrease as the amount of training data increases.

### 8.5 The Statistical Computation with GenCFD is Robust

We recall that the test task for GenCFD is *out-of-distribution* as the test distribution is a Dirac measure whereas the training distribution is spread out. Yet, GenCFD computes the samples as well as the statistics accurately. We test this *generalization ability* further by choosing the functions, on which the test distribution is concentrated (57), from yet another distribution. To this end, we consider the CSF dataset and choose  $p \in \{8, 9, 10, 11, 12\}$  uniformly at random for each sample, rather than constant and equal to 10. Additionally, the perturbation functions  $\sigma_y^i$ , and  $\sigma_z^j$  are extended with another parameter  $\xi_y, \xi_z \sim \mathcal{U}_{[-0.0625, 0.0625]}$  by setting

$$\begin{aligned}\sigma_y^j(x) &= \delta \sum_{k=1}^p \alpha_k^y \sin(2\pi kx - \beta_k^y) - \xi_y \\ \sigma_z^j(x) &= \delta \sum_{k=1}^p \alpha_k^z \sin(2\pi kx - \beta_k^z) - \xi_z.\end{aligned}$$

This leads to a distribution  $\nu$ , which is different from the training distribution.

Zero-shot Evaluation is performed by sampling the initial condition  $\bar{u} \sim \nu$  and feeding it directly into the pretrained models (GenCFD and baselines). No adjustments or modifications to the pretrained models are needed for this evaluation.

For fine-tuning, the models are trained using the objectives (35) on 300 samples drawn from the distribution  $\nu$ . During this step, all pretrained model parameters are updated.

Even with no additional training (*Zero-Shot mode*), GenCFD is able to generate high quality samples (see Fig. 31) as well as approximate mean, standard deviation and probability distribution quantitatively (Table 12) to reasonable accuracy. For instance, the 1-Wasserstein distance between the ground truth and the generated distribution increases, on average, by a factor of 2 over the previously tested distribution (see Table 7), with this *zero-shot* evaluation. By further *fine-tuning* the model with merely 300 trajectories generated from initial data, drawn from the new training distribution  $\nu$ , the error is reduced to the previous levels, see Table 12 and compare with Table 7 further showcasing the ability of GenCFD to handle *distribution shifts*.

Another avenue for checking robustness arises when we check how the statistical errors with GenCFD increase over time. From Table 13, where we present the 1-Wasserstein distance between the ground truth and the GenCFD generated conditional distribution for the CSF dataset, we observe that after a modest increase in the beginning of evolution when the turbulence kicks in, the error is approximately constant for the time period when the turbulence is fully developed.

This demonstration of the robustness of our proposed algorithm to *time evolution* is particularly pertinent for the approximation of the Taylor–Green vortex as the flow starts laminar (in fact smooth) and dynamically transitions to turbulence over time. For instance, the flow at time  $T = 0.8$  is still markedly laminar while it has become turbulent by time  $T = 2$ . Can GenCFD still be robust with such transitions from laminar to turbulent flow? In particular, can it be accurate at also approximating deterministic flows? These questions are answered qualitatively in Figs. 24, 25, 26 and 27, from which we observe that GenCFD continues to provide realistic samples for both velocity and vorticity and approximates the mean and especially, the standard deviation, for the underlying laminar flow very accurately. This observation is further buttressed by the results in Table 11 where we see that GenCFD has lower errors in every single metric when compared to the ML baselines. Thus, this experiment clearly showcases the ability of GenCFD to accurately approximate both deterministic and stochastic fluid flow.

## 8.6 Statistical Computation with GenCFD is Fast

The computational cost of inference with the GenCFD algorithm scales *linearly* in the number of steps required for solving the reverse-SDE (15). In Table 14, we show how robust our algorithm is with respect to the number of steps in solving the reverse-SDE to find that as few as 16 steps suffice in ensuring acceptable statistical accuracy, with 32 steps being almost as good as 128 steps. Given this observation, we can deploy our model with 32 steps for the reverse SDE. The resulting inference time with GenCFD, in comparison to the underlying CFD solvers,

is presented in Table 15. We observe from the figure that GenCFD requires approximately 1 to 4 seconds for a single inference run on a NVIDIA RTX4090 24GB GPU, for all the test cases that we have considered. These inference times will be even faster for more powerful GPUs. In contrast, the run times for CFD solvers vary considerably based on the underlying numerical method and on the hardware used to run them. A highly optimized code such as **Azeban** can perform  $128^3$  CFD simulations on a periodic domain with spectral viscosity method in approximately 10 seconds on GPUs. However, this run time is significantly larger at approximately 20 minutes, even on state-of-the-art H100 96GB GPUs for a more complicated test case like the nozzle flow, even when a highly scalable solver such as **OpenLB** is employed. On the other hand, all the CFD simulations, which are performed on CPUs, required run times in the order of hours. This is indicative of the real world as most CFD codes run on CPUs. Hence, from Table 15, we see that GenCFD can provide a speedup, ranging from one to three orders of magnitude with respect to GPU-based CFD codes while providing an impressive three to five orders of magnitude speedup over CPU-based CFD codes. It is precisely this very high computational speed, coupled with accuracy, that makes GenCFD very attractive for applications in many areas of fluid dynamics.

**Numerical Results with the Toy Models.** Recalling Toy Model #1, which is given by the map  $S^\Delta$  (86), we present numerical results with a diffusion model and an underlying deterministic ML baseline in Fig. 32, from where we observe that the deterministic ML approximation does accurately approximate  $S^\Delta$ , for moderate values of  $\Delta \approx 0.1$ . But for even lower values of  $\Delta$ , the deterministic approximation collapses to the mean as predicted by the theory presented earlier. On the other hand, the diffusion model is able to approximate  $S^\Delta$  very well, for moderate values of  $\Delta \approx 0.05$  when the mapping is not too oscillatory in a deterministic manner, while at the same time, being able to approximate the statistical limit for very small values of  $\Delta \approx 0.002$ . Consistent with our theory, this ability of diffusion models to be robust with respect to any value of  $\Delta$  in this case is worth highlighting. It also matches the empirical observation that GenCFD was able to approximate the Taylor–Green vortex flow accurately for both the laminar and turbulent regimes.

In Fig. 32, we also illustrate the different modes through which deterministic ML models and diffusion models *learn during training*. A deterministic ML model first learns the mean and then adds oscillations as more and more gradient descent steps are taken, consistent with the well-known spectral bias of neural networks [59]. If the underlying map is too oscillatory, it simply does not add the oscillations and predicts the mean, which yields a (local) minimum for the  $L^2$  loss. In contrast, diffusion models do the opposite. Already, very early in the training, they capture the statistical limit measure and as more gradient descent steps are taken, they start *clearing out* the measure to reveal more of the underlying deterministic oscillations. If the underlying map is too oscillatory, the diffusion model sticks to the measure-valued output even when more training steps are taken, enabling it to capture both deterministic approximations as well as statistical information, as necessary.

Finally in Fig. 33, we present results with a diffusion model and the underlying deterministic baseline on Toy Model #2, which was described and rigorously analyzed in the previous section.

We observe from this figure that while the deterministic model is accurate for low wave numbers (around  $k \approx 10$ ), it collapses to the mean for  $k \geq 30$ , even with a lot of training steps. This also implies that the phase space approximation is very poor and the underlying constraint is violated at high wavenumber. In contrast, the diffusion model is able to learn the underlying map, both for small as well as large wavenumbers, even with a few training steps. The underlying constraint is satisfied for any of the tested wavenumbers and the contrast between the deterministic and diffusion models is nicely shown in the (synthetic) spectra plotted in Fig. 23 (bottom row). The diffusion model is able to capture structures at all wave numbers whereas the deterministic model has a limited spectral resolution, explaining the spectral findings for fluid flows in Fig. 23. To obtain these results, we used the same model specifications and training procedure as described in Section 6.10.1. In contrast to that section, the deterministic model now maps a one-dimensional input to a two-dimensional output,  $h \mapsto \Psi_{\text{det}}(h) \approx \mathcal{S}^{(k)}(h)$ . The diffusion model has four-dimensional input,  $(u_\sigma, h, \sigma)$ , where  $u_\sigma \in \mathbb{R}^2$ ,  $h \in [0, 1]$ ,  $\sigma \in \mathbb{R}$ , and outputs a two-dimensional (denoised) vector  $u$ ,  $(u, h, \sigma) \mapsto D(u; h, \sigma)$ . In each case, the model is trained on 2048 samples, with  $h \sim \mathcal{U}([0, 1])$  uniformly sampled over the input range  $[0, 1]$ . All other implementation details are identical to the ones specified in Section 6.10.1.



## 9 Supplementary Tables

|        |       | $e_\mu$      | $e_\sigma$   | $W_1$         | $\text{CRPS}_G$ |
|--------|-------|--------------|--------------|---------------|-----------------|
| GenCFD | $u_x$ | <b>0.154</b> | <b>0.056</b> | <b>0.0165</b> | <b>0.481</b>    |
|        | $u_y$ | <b>0.155</b> | <b>0.055</b> | <b>0.0172</b> | <b>0.479</b>    |
|        | $u_z$ | <b>0.282</b> | <b>0.053</b> | <b>0.0145</b> | <b>0.469</b>    |
| UViT   | $u_x$ | 0.883        | 0.813        | 0.130         | 0.768           |
|        | $u_y$ | 0.944        | 0.829        | 0.138         | 0.802           |
|        | $u_z$ | 1.016        | 0.881        | 0.110         | 0.648           |
| FNO    | $u_x$ | 0.359        | 0.999        | 0.121         | 0.690           |
|        | $u_y$ | 0.362        | 0.999        | 0.123         | 0.690           |
|        | $u_z$ | 0.756        | 0.998        | 0.119         | 0.671           |
| C-FNO  | $u_x$ | 0.210        | 1.000        | 0.117         | 0.670           |
|        | $u_y$ | 0.210        | 1.000        | 0.118         | 0.668           |
|        | $u_z$ | 0.402        | 1.000        | 0.115         | 0.653           |

**Table 6: Error metrics, defined in Materials and Methods, for different models for the Taylor–Green vortex experiment.** The metrics are defined in [SM 6.9](#) and computed at time  $T = 2.0$ . Results for the best performing model are in bold.

|        |       | $e_\mu$      | $e_\sigma$   | $W_1$        | $\text{CRPS}_G$ |
|--------|-------|--------------|--------------|--------------|-----------------|
| GenCFD | $u_x$ | <b>0.088</b> | <b>0.114</b> | <b>0.034</b> | <b>0.347</b>    |
|        | $u_y$ | <b>0.271</b> | <b>0.110</b> | <b>0.030</b> | <b>0.316</b>    |
|        | $u_z$ | <b>0.268</b> | <b>0.113</b> | <b>0.032</b> | <b>0.317</b>    |
| UViT   | $u_x$ | 0.604        | 0.562        | 0.253        | 0.708           |
|        | $u_y$ | 1.096        | 0.558        | 0.147        | 0.443           |
|        | $u_z$ | 1.038        | 0.663        | 0.150        | 0.475           |
| FNO    | $u_x$ | 0.301        | 0.957        | 0.169        | 0.547           |
|        | $u_y$ | 0.889        | 0.959        | 0.148        | 0.486           |
|        | $u_z$ | 0.815        | 0.962        | 0.150        | 0.485           |
| C-FNO  | $u_x$ | 0.217        | 0.864        | 0.133        | 0.452           |
|        | $u_y$ | 0.622        | 0.880        | 0.120        | 0.405           |
|        | $u_z$ | 0.641        | 0.880        | 0.124        | 0.417           |

**Table 7: Error metrics for different models at  $T = 1.0$  for the cylindrical shear flow experiment.** Results for the best performing model are in bold.

|        |        | $e_\mu$      | $e_\sigma$   | $W_1$        | $\text{CRPS}_G$ |
|--------|--------|--------------|--------------|--------------|-----------------|
| GenCFD | $\rho$ | <b>0.049</b> | <b>0.381</b> | <b>0.054</b> | <b>0.0035</b>   |
|        | $u_x$  | <b>0.015</b> | <b>0.332</b> | <b>0.093</b> | <b>0.0040</b>   |
|        | $u_y$  | <b>0.195</b> | <b>0.203</b> | <b>0.054</b> | 0.0033          |
|        | $u_z$  | <b>0.171</b> | <b>0.301</b> | <b>0.021</b> | <b>0.0012</b>   |
|        | $p$    | <b>0.015</b> | <b>0.316</b> | <b>1.525</b> | <b>0.0436</b>   |
| UViT   | $\rho$ | 0.252        | 0.993        | 0.400        | 0.0151          |
|        | $u_x$  | 0.095        | 0.985        | 0.631        | 0.0151          |
|        | $u_y$  | 0.849        | 0.992        | 0.283        | 0.0065          |
|        | $u_z$  | 0.722        | 0.992        | 0.158        | 0.0046          |
|        | $p$    | 0.127        | 0.980        | 13.68        | 0.2834          |
| FNO    | $\rho$ | 0.138        | 0.921        | 0.242        | 0.0085          |
|        | $u_x$  | 0.071        | 0.940        | 0.393        | 0.0111          |
|        | $u_y$  | 0.680        | 0.973        | 0.223        | 0.0054          |
|        | $u_z$  | 0.424        | 0.929        | 0.100        | 0.0027          |
|        | $p$    | 0.081        | 0.892        | 8.328        | 0.1782          |
| C-FNO  | $\rho$ | 0.081        | 0.798        | 0.133        | 0.0052          |
|        | $u_x$  | 0.037        | 0.688        | 0.264        | 0.0058          |
|        | $u_y$  | 0.399        | 0.613        | 0.127        | <b>0.0032</b>   |
|        | $u_z$  | 0.236        | 0.878        | 0.042        | 0.0017          |
|        | $p$    | 0.038        | 0.630        | 4.128        | 0.0811          |

**Table 8: Results for error metrics for different models at Time  $T = 1.0$  for the cloud-shock interaction experiment.** Results for the best performing model are in bold.

|        |       | $e_\mu$      | $e_\sigma$   | $W_1$          | $\text{CRPS}_G$ |
|--------|-------|--------------|--------------|----------------|-----------------|
| GenCFD | $u_x$ | 0.148        | <b>0.258</b> | <b>0.0073</b>  | <b>0.268</b>    |
|        | $u_y$ | N/A          | <b>0.207</b> | <b>0.0049</b>  | <b>0.508</b>    |
|        | $u_z$ | N/A          | <b>0.218</b> | <b>0.0050</b>  | <b>0.515</b>    |
|        | $E_k$ | <b>0.151</b> | <b>0.240</b> | <b>0.00065</b> | <b>0.230</b>    |
| UViT   | $u_x$ | <b>0.074</b> | 0.850        | 0.0109         | 0.324           |
|        | $u_y$ | N/A          | 0.926        | 0.0098         | 0.657           |
|        | $u_z$ | N/A          | 0.943        | 0.0100         | 0.669           |
|        | $E_k$ | 0.167        | 0.827        | 0.00128        | 0.276           |
| FNO    | $u_x$ | 0.176        | 0.864        | 0.0139         | 0.358           |
|        | $u_y$ | N/A          | 0.934        | 0.0111         | 0.678           |
|        | $u_z$ | N/A          | 0.956        | 0.0112         | 0.693           |
|        | $E_k$ | 0.237        | 0.848        | 0.00136        | 0.316           |
| C-FNO  | $u_x$ | 0.124        | 0.858        | 0.0116         | 0.333           |
|        | $u_y$ | N/A          | 0.923        | 0.0100         | 0.661           |
|        | $u_z$ | N/A          | 0.938        | 0.0102         | 0.673           |
|        | $E_k$ | 0.207        | 0.835        | 0.00134        | 0.287           |

**Table 9: Results for Error metrics for different models at Time  $T = 1.0$  for the nozzle flow experiment.** Results for the best performing model are in bold. Note that the term N/A for the mean error  $e_\mu$  for the  $u_y, u_z$  components signifies the fact that the ground truth means of these velocity components are 0 and the relative errors are not well-defined.

|        |       | $e_\mu$                             | $e_\sigma$   | $W_1$        | $\text{CRPS}_G$ |
|--------|-------|-------------------------------------|--------------|--------------|-----------------|
| GenCFD | $u_x$ | <b>0.223</b>                        | <b>0.072</b> | <b>0.036</b> | <b>0.557</b>    |
|        | $u_y$ | N/A                                 | <b>0.094</b> | <b>0.038</b> | <b>0.567</b>    |
|        | $u_z$ | N/A                                 | <b>0.059</b> | <b>0.037</b> | <b>0.553</b>    |
|        | $T$   | $10 \cdot 10^{-5}$                  | <b>0.091</b> | <b>0.025</b> | <b>0.00060</b>  |
|        | $E_k$ | <b>0.109</b>                        | <b>0.137</b> | <b>0.072</b> |                 |
| UViT   | $u_x$ | 0.235                               | 0.807        | 0.305        | 0.692           |
|        | $u_y$ | N/A                                 | 0.827        | 0.308        | 0.714           |
|        | $u_z$ | N/A                                 | 0.825        | 0.403        | 0.704           |
|        | $T$   | <b><math>6 \cdot 10^{-5}</math></b> | 0.890        | 0.108        | 0.00086         |
|        | $E_k$ | 0.956                               | 0.965        | 0.644        |                 |
| FNO    | $u_x$ | 0.345                               | 0.936        | 0.343        | 0.753           |
|        | $u_y$ | N/A                                 | 0.849        | 0.315        | 0.730           |
|        | $u_z$ | N/A                                 | 0.973        | 0.470        | 0.779           |
|        | $T$   | $18 \cdot 10^{-5}$                  | 0.612        | 0.080        | 0.00074         |
|        | $E_k$ | 0.978                               | 0.986        | 0.657        |                 |
| C-FNO  | $u_x$ | 0.293                               | 0.875        | 0.332        | 0.722           |
|        | $u_y$ | N/A                                 | 0.922        | 0.360        | 0.762           |
|        | $u_z$ | N/A                                 | 0.909        | 0.453        | 0.754           |
|        | $T$   | $44 \cdot 10^{-5}$                  | 0.807        | 0.143        | 0.00100         |
|        | $E_k$ | 0.977                               | 0.981        | 0.659        |                 |

**Table 10: Results for Error metrics for different models at Time  $T = 1.0$  for the dry convective boundary layer experiment.** Results for the best performing model are in bold. Note that the term N/A for the mean error  $e_\mu$  for the  $u_y, u_z$  components signifies the fact that the ground truth means of these velocity components are 0 and the relative errors are not well-defined.

|        |       | $e_\mu$      | $e_\sigma$   | $W_1$         | $\text{CRPS}_G$ |
|--------|-------|--------------|--------------|---------------|-----------------|
| GenCFD | $u_x$ | <b>0.030</b> | <b>0.228</b> | <b>0.0077</b> | <b>0.053</b>    |
|        | $u_y$ | <b>0.030</b> | <b>0.227</b> | <b>0.0075</b> | <b>0.050</b>    |
|        | $u_z$ | <b>0.030</b> | <b>0.251</b> | <b>0.0061</b> | <b>0.039</b>    |
| UViT   | $u_x$ | 0.843        | 1.219        | 0.203         | 0.832           |
|        | $u_y$ | 0.880        | 1.328        | 0.207         | 0.869           |
|        | $u_z$ | 0.957        | 1.207        | 0.175         | 0.762           |
| FNO    | $u_x$ | 0.458        | 0.989        | 0.100         | 0.489           |
|        | $u_y$ | 0.459        | 0.987        | 0.102         | 0.489           |
|        | $u_z$ | 0.556        | 0.978        | 0.096         | 0.473           |
| C-FNO  | $u_x$ | 0.151        | 0.997        | 0.0389        | 0.171           |
|        | $u_y$ | 0.146        | 0.997        | 0.0367        | 0.166           |
|        | $u_z$ | 0.166        | 0.997        | 0.0317        | 0.147           |

Table 11: Error metrics for different models at  $T = 0.8$  for the Taylor–Green vortex experiment.

|            |       | $e_\mu$      | $e_\sigma$   | $W_1$        | $\text{CRPS}_G$ |
|------------|-------|--------------|--------------|--------------|-----------------|
| 0-Shot     | $u_x$ | 0.230        | 0.240        | 0.100        | 0.422           |
|            | $u_y$ | 0.507        | 0.228        | 0.051        | 0.344           |
|            | $u_z$ | 0.468        | 0.227        | 0.051        | 0.343           |
| Fine-Tuned | $u_x$ | <b>0.097</b> | <b>0.136</b> | <b>0.037</b> | <b>0.339</b>    |
|            | $u_y$ | <b>0.309</b> | <b>0.133</b> | <b>0.034</b> | <b>0.309</b>    |
|            | $u_z$ | <b>0.309</b> | <b>0.134</b> | <b>0.032</b> | <b>0.310</b>    |

**Table 12: Error metrics for GenCFD for the cylindrical shear flow experiment at  $T = 1$ .** The tests were performed with data from a different distribution than the ones from Table 7.

|        |       | $T = 0.25$ | $T = 0.5$ | $T = 0.75$ | $T = 1$ |
|--------|-------|------------|-----------|------------|---------|
| GenCFD | $u_x$ | 0.016      | 0.022     | 0.027      | 0.034   |
|        | $u_y$ | 0.013      | 0.021     | 0.025      | 0.030   |
|        | $u_z$ | 0.014      | 0.022     | 0.024      | 0.032   |

**Table 13: Errors in Wasserstein metric for the distribution generated by GenCFD when compared to the ground truth distribution for the cylindrical shear flow experiment at different times in the evolution.**

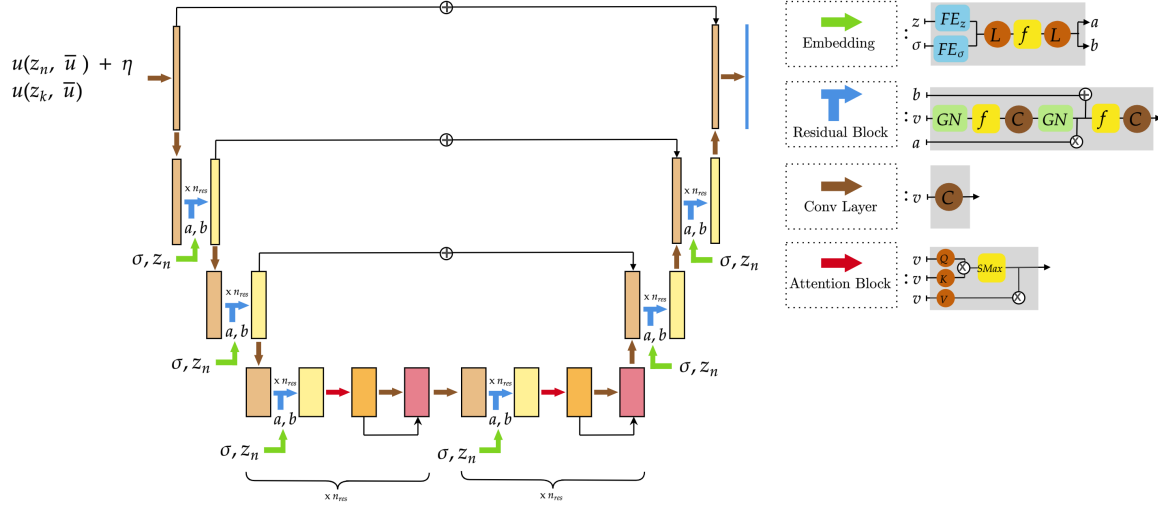
|        |       | $K = 8$ | $K = 16$ | $K = 32$ | $K = 64$ | $K = 128$ |
|--------|-------|---------|----------|----------|----------|-----------|
| GenCFD | $u_x$ | 1.736   | 0.051    | 0.035    | 0.034    | 0.034     |
|        | $u_y$ | 2.311   | 0.049    | 0.031    | 0.030    | 0.030     |
|        | $u_z$ | 2.087   | 0.045    | 0.033    | 0.032    | 0.032     |

**Table 14: Errors in Wasserstein metric for the distribution generated by GenCFD, with different number of steps of the reverse SDE (15) when compared to the ground truth distribution for the cylindrical shear flow experiment at  $T = 1$ .**

| Benchmark | Ground truth<br>(GPU)    | Ground truth<br>(CPU)   | GenCFD<br>(GPU)          | Speedup<br>(wrt GPU)                 | Speedup<br>(wrt CPU)                 |
|-----------|--------------------------|-------------------------|--------------------------|--------------------------------------|--------------------------------------|
| TG, CSF   | $1.07 \times 10^1$ secs  | $0.72 \times 10^3$ secs | $0.450 \times 10^0$ secs | <b><math>2.37 \times 10^1</math></b> | <b><math>1.60 \times 10^3</math></b> |
| CSI       | $0.390 \times 10^3$ secs | $1.80 \times 10^4$ secs | $0.450 \times 10^0$ secs | <b><math>0.87 \times 10^3</math></b> | <b><math>0.40 \times 10^5</math></b> |
| NF        | $1.20 \times 10^3$ secs  | $1.17 \times 10^4$ secs | $1.45 \times 10^0$ secs  | <b><math>0.83 \times 10^3</math></b> | <b><math>0.81 \times 10^4</math></b> |
| CBL       | N/A                      | $0.48 \times 10^5$ secs | $0.38 \times 10^1$ secs  | N/A                                  | <b><math>1.25 \times 10^4</math></b> |

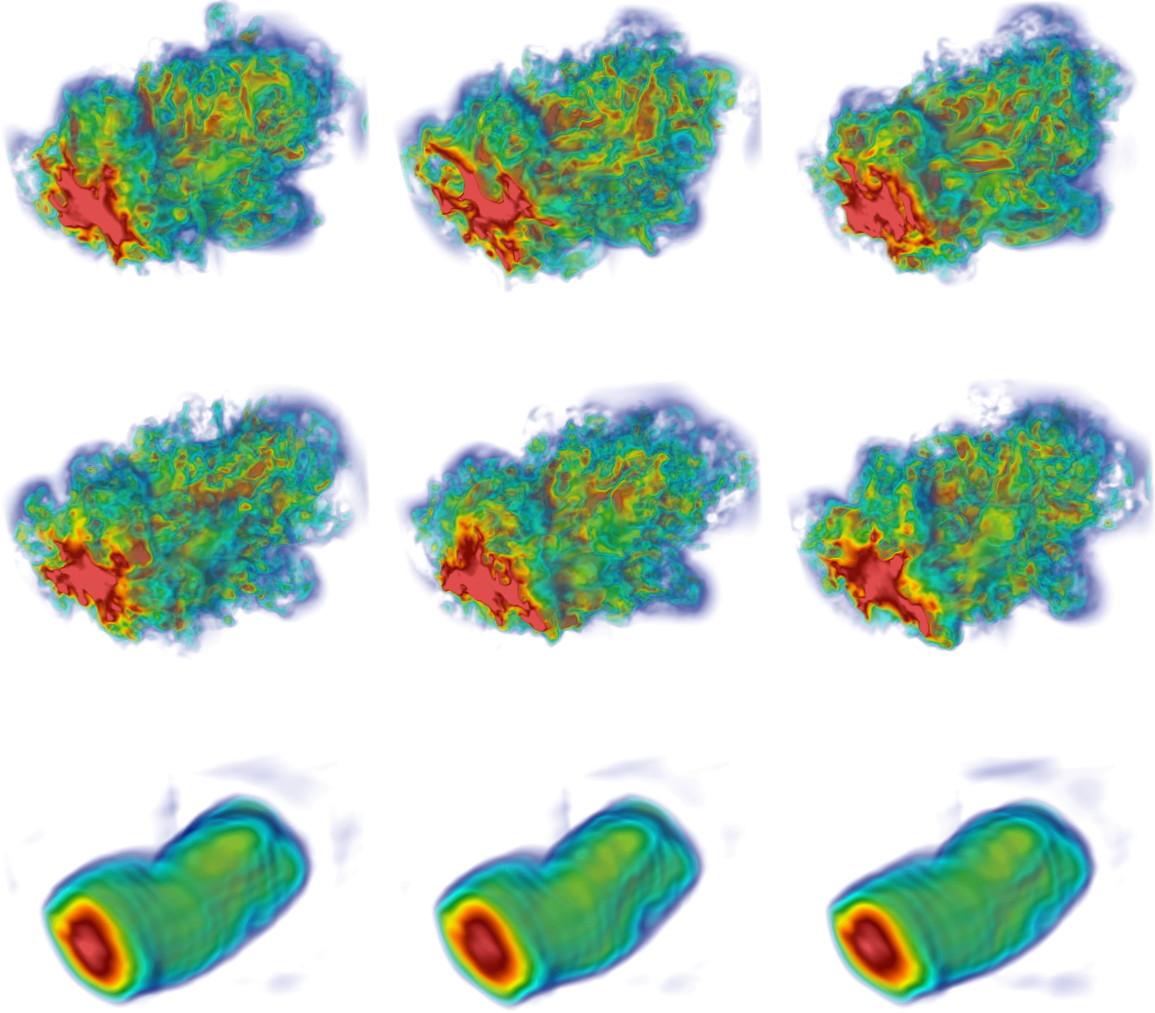
**Table 15: Runtimes and speedup for generating a single sample with the CFD solvers and with GenCFD.** The CFD solvers were used to simulate the ground truth (see Section 6.7 on which CFD solver is used for which experiment). The inference time to generate a single sample with GenCFD was measured on a NVIDIA RTX 4090 GPU with 24GB of memory. The term N/A implies that the corresponding GPU or CPU code is not available or has not been tested for the corresponding CFD solver. Note that in this comparison different machines have been used than for the sample generation. The computation time in seconds is rounded and includes I/O operations. For the TG, CSF, and CSI the ground truth (CPU) has been computed on a single Intel Core i7-9750H with 6 cores. The respective ground truth (GPU) was computed on an NVIDIA H100 GPU with 96GB of memory. For the NF the ground truth (CPU) has been computed on two Intel Xeon Gold 6326 with 16 cores each and the ground truth (GPU) has been computed on an NVIDIA H100 GPU, respectively. For CBL the ground truth data was generated on a cluster with diverse CPU hardware (mostly AMD EPYC 7H12 and AMD EPYC 7763 processors), and the mean runtime on a single core is reported.

## 10 Supplementary Figures

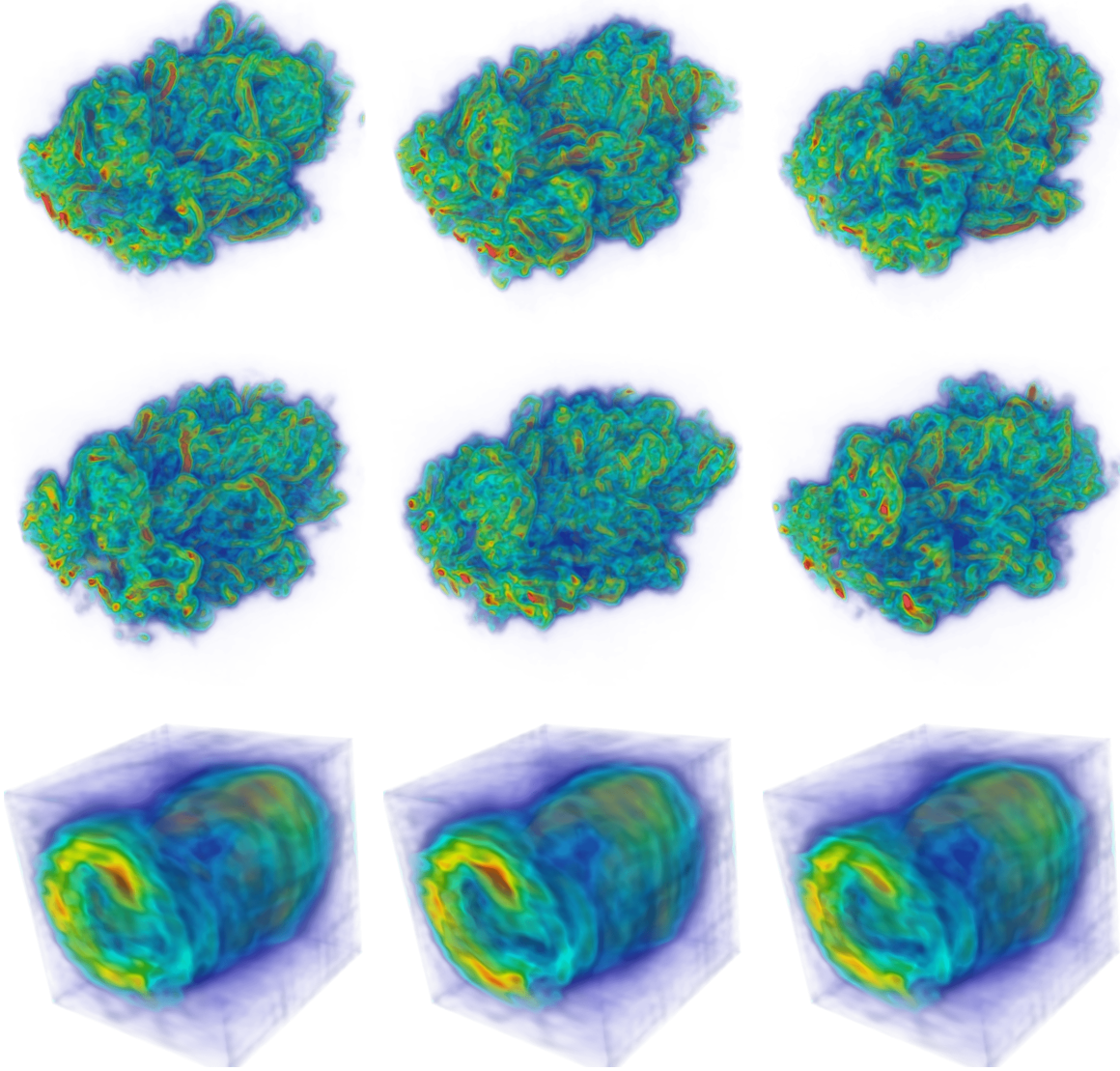


**Figure 5:** A schematic of the architecture of the UViT neural network which is used as the model for the denoiser in GenCFD. A detailed description of the model is provided in Materials and Methods.

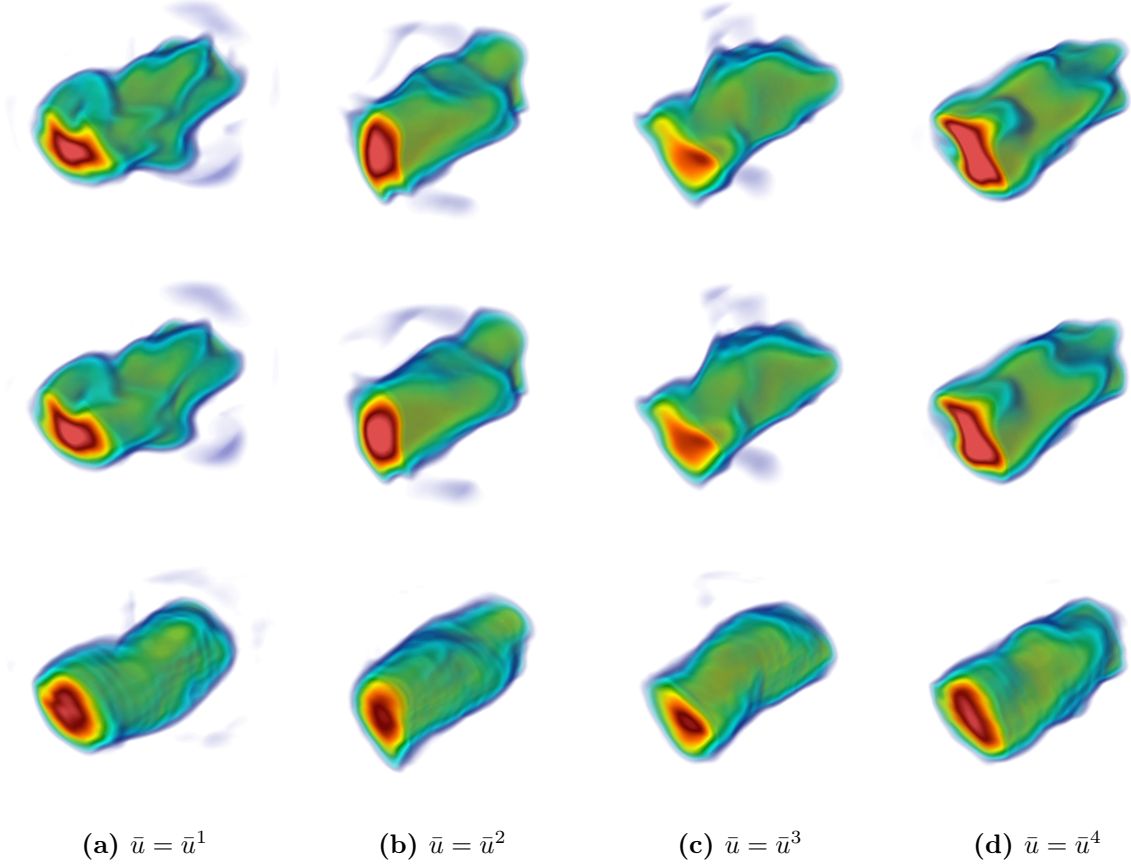




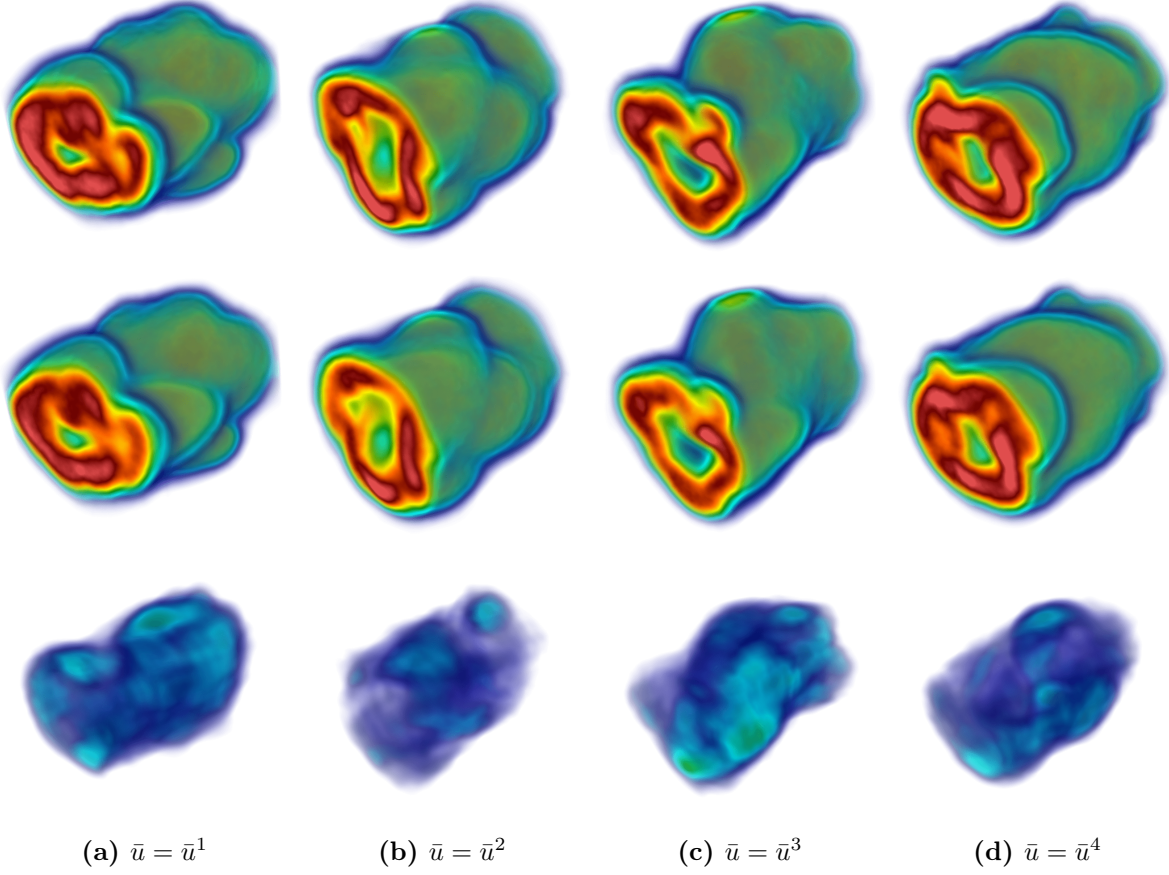
**Figure 6:** Visualization of pointwise kinetic energy for 3 randomly generated samples for the three-dimensional cylindrical shear flow experiment at time  $T = 1$ . Ground truth (Top Row), GenCFD (Middle Row) and C-FNO (Bottom Row). The colormap for all the figures ranges from 0.6 (dark blue) to 1.7 (dark red).



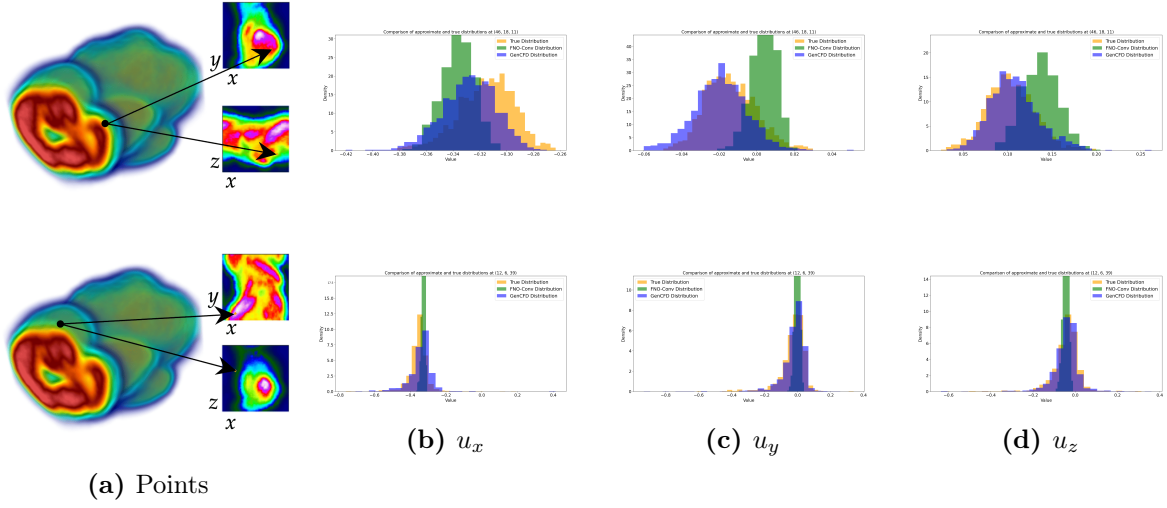
**Figure 7: Visualization of pointwise vorticity intensity for 3 randomly generated samples for the three-dimensional cylindrical shear flow experiment at time  $T = 1$ .** Ground truth (Top Row), GenCFD (Middle Row) and C-FNO (Bottom Row). The colormap for the top and middle rows ranges from  $10^{-4}$  (dark blue) to 40.0 (dark red), whereas for the bottom row, it ranges from 0.5 (filtering the low values) to 19.5.



**Figure 8: Visualization of the mean of the (pointwise) kinetic energy for the cylindrical shear flow experiment at time  $T = 1$ , for four different initial distributions.** Data generated by the ground truth (Top Row), GenCFD (Middle Row) and C-FNO (bottom Row). The colormap for all the figures ranges from 0.6 (dark blue) to 1.7 (dark red).

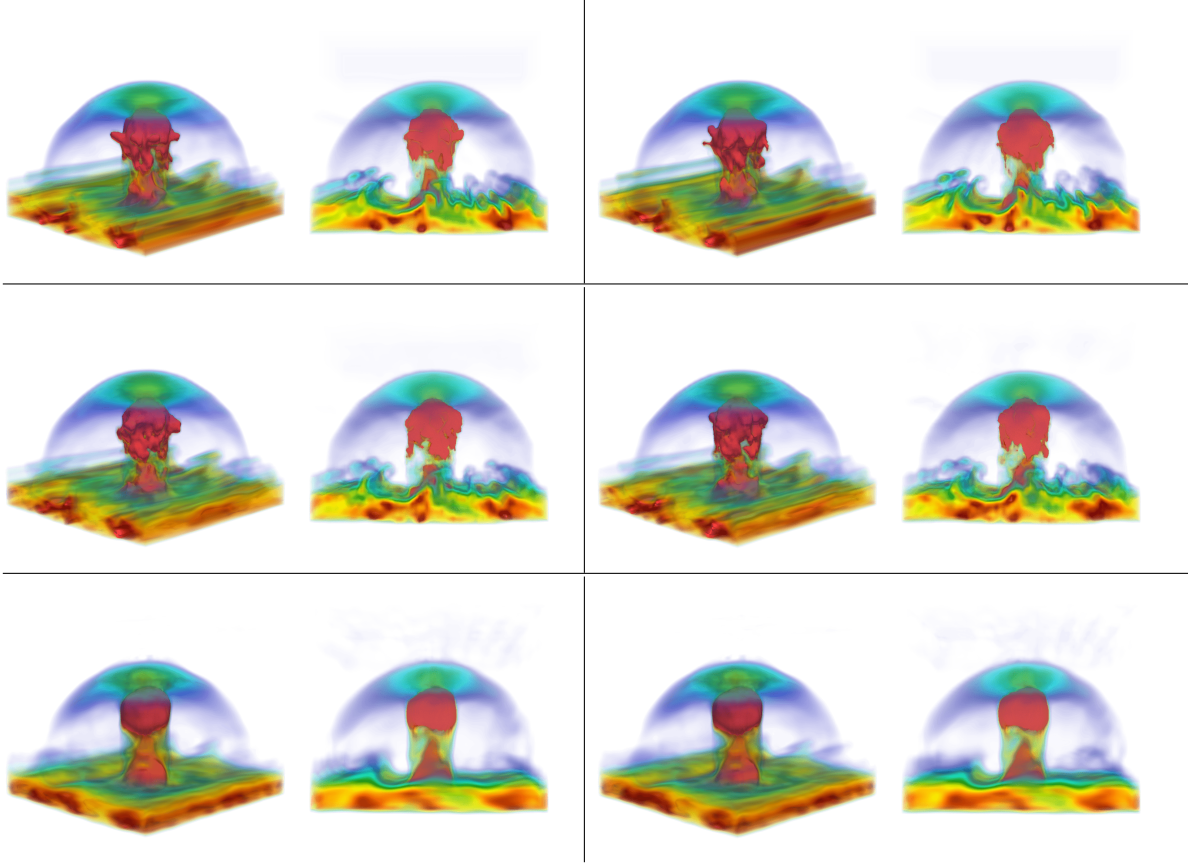


**Figure 9: Visualization of the standard deviation of the kinetic energy for the cylindrical shear flow experiment at time  $T = 1$ , for four different initial distributions.** Data generated by the ground truth (Top Row), GenCFD (Middle Row) and C-FNO (Bottom Row). The colormap for the top and middle rows ranges from 0.05 (dark blue) to 0.65 (dark red), whereas for the bottom row, it ranges from 0.05 to 0.25.

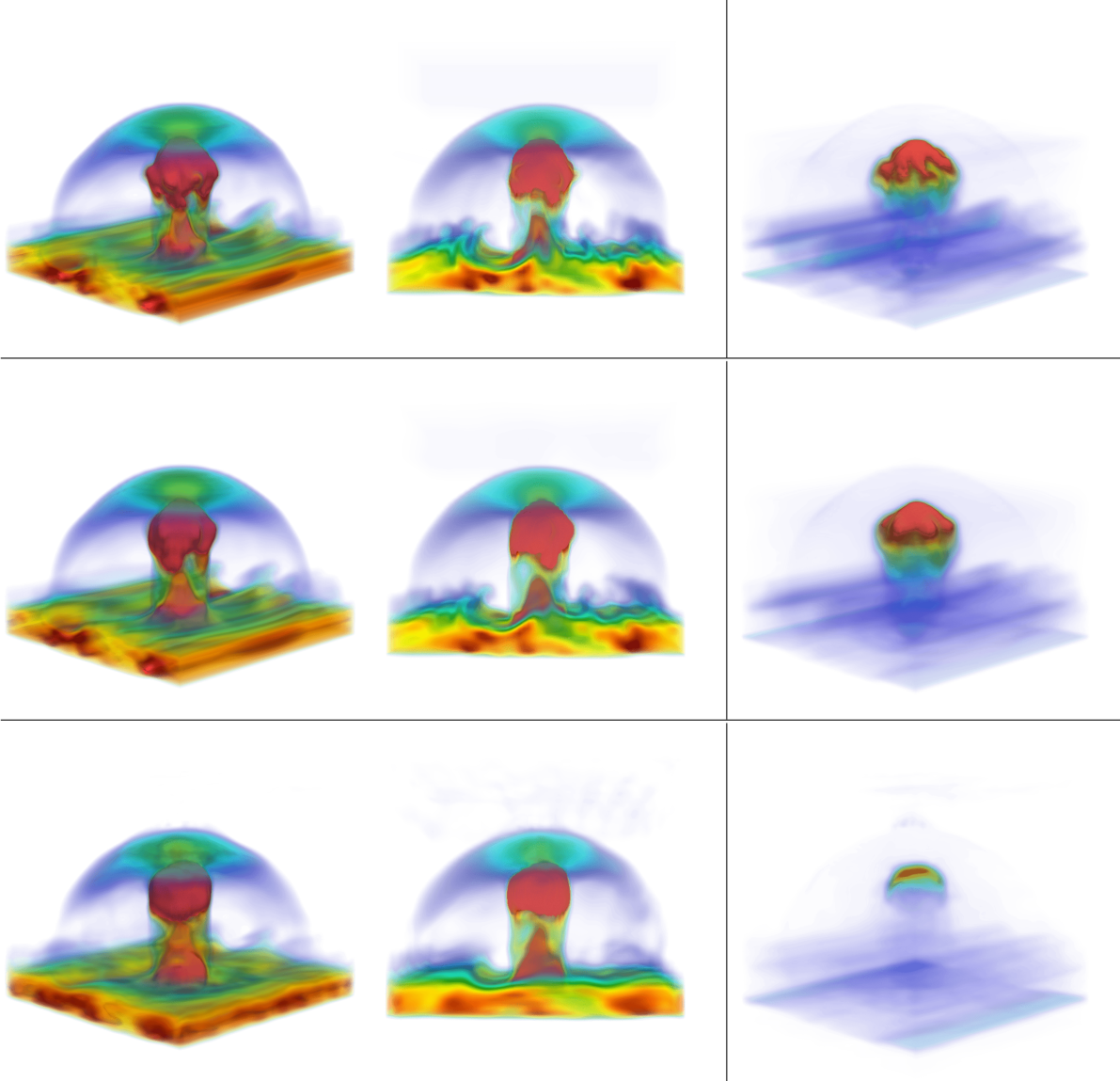


**Figure 10: Visualization of the point PDFs, at two different points (marked in the left column), of all the velocity components for the three-dimensional cylindrical shear flow experiment at time  $T = 1$ . Results generated by the ground truth, GenCFD and C-FNO.**

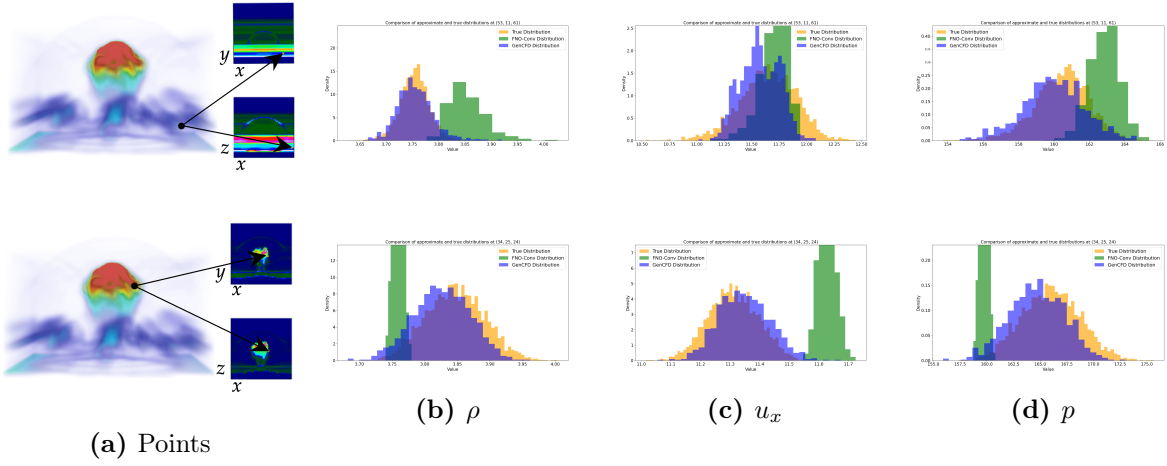




**Figure 11:** Visualization from two different angles of the density at time  $T = 0.06$  for the cloud-shock interaction experiment, generated by the ground truth (Top Row), GenCFD (Middle Row) and C-FN0 (Bottom Row). The colormap for all the figures ranges from 3.90 (dark blue) to 6.35 (dark red).

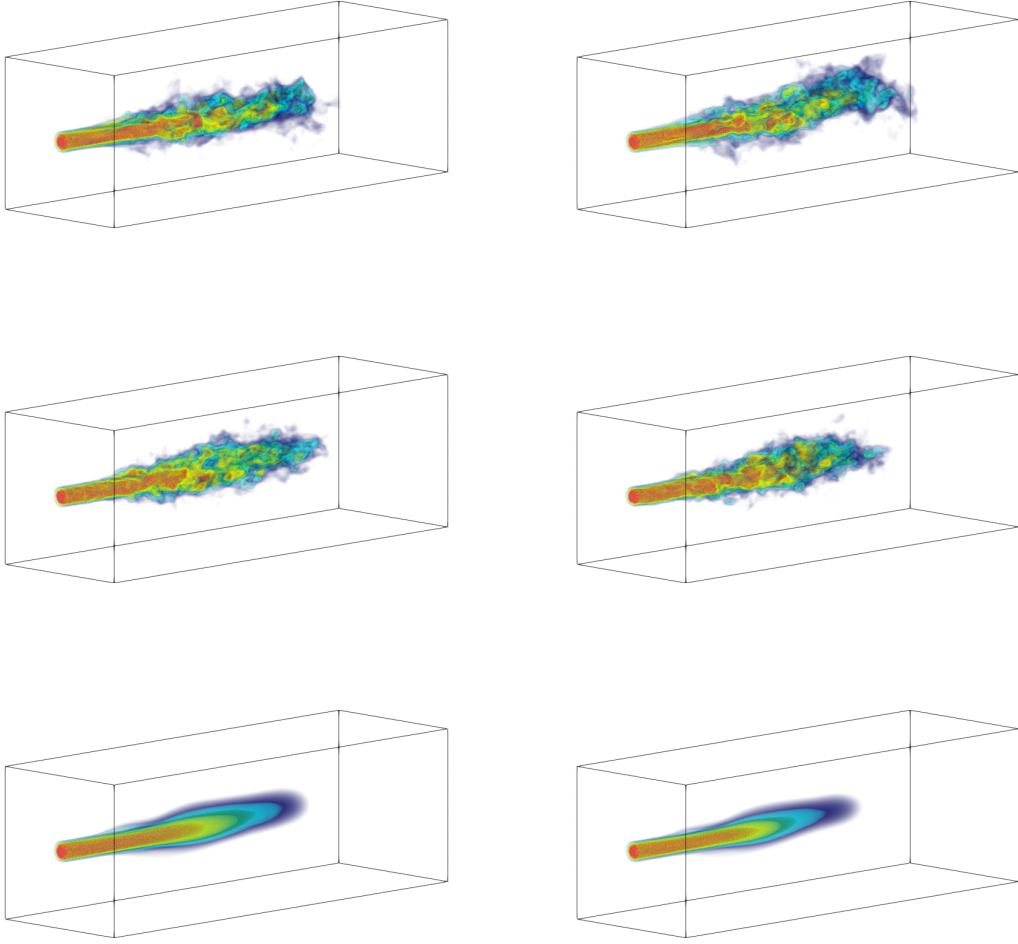


**Figure 12:** Mean (two different angles in Left and Center Column) and standard deviation (Right Column) of the density at time  $T = 0.06$  for the cloud-shock interaction experiment with ground truth (Top Row), GenCFD (Middle Row) and C-FNO (Bottom Row). The colormap for the figures representing the means ranges from 3.90 (dark blue) to 6.35 (dark red). The colormap for the figures representing the standard deviations ranges from 0.0075 to 3.0 for the top and middle rows, and from 0.0075 to 2.0 for the bottom row.

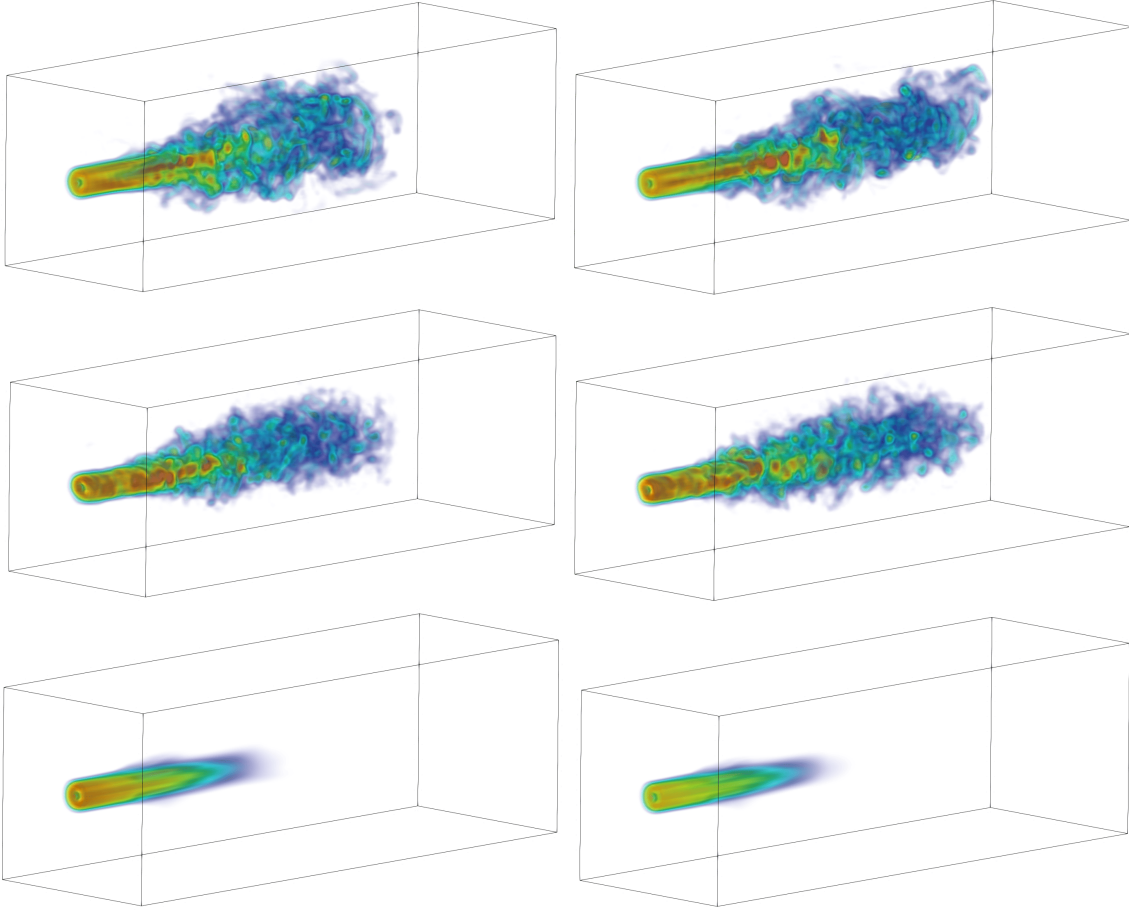


**Figure 13: Point PDFs at two different points (left most column) of the density,  $x$ -velocity and pressure, at time  $T = 0.06$  of the cloud-shock interaction experiment, generated by the ground truth, GenCFD and C-FNO.**

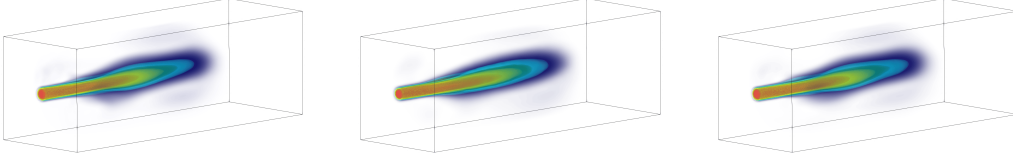




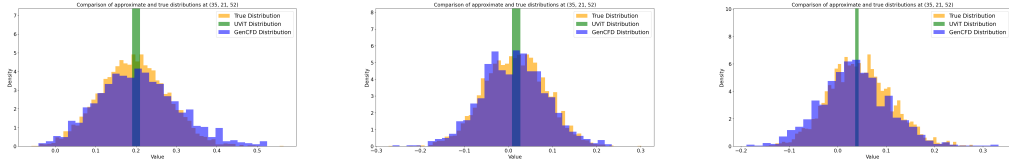
**Figure 14: Two random samples, for the same injection velocity, of the (pointwise) kinetic energy at time  $T = 130$  for the nozzle flow experiment. Data generated with ground truth (Top Row), GenCFD (Middle Row) and UViT (Bottom Row).**



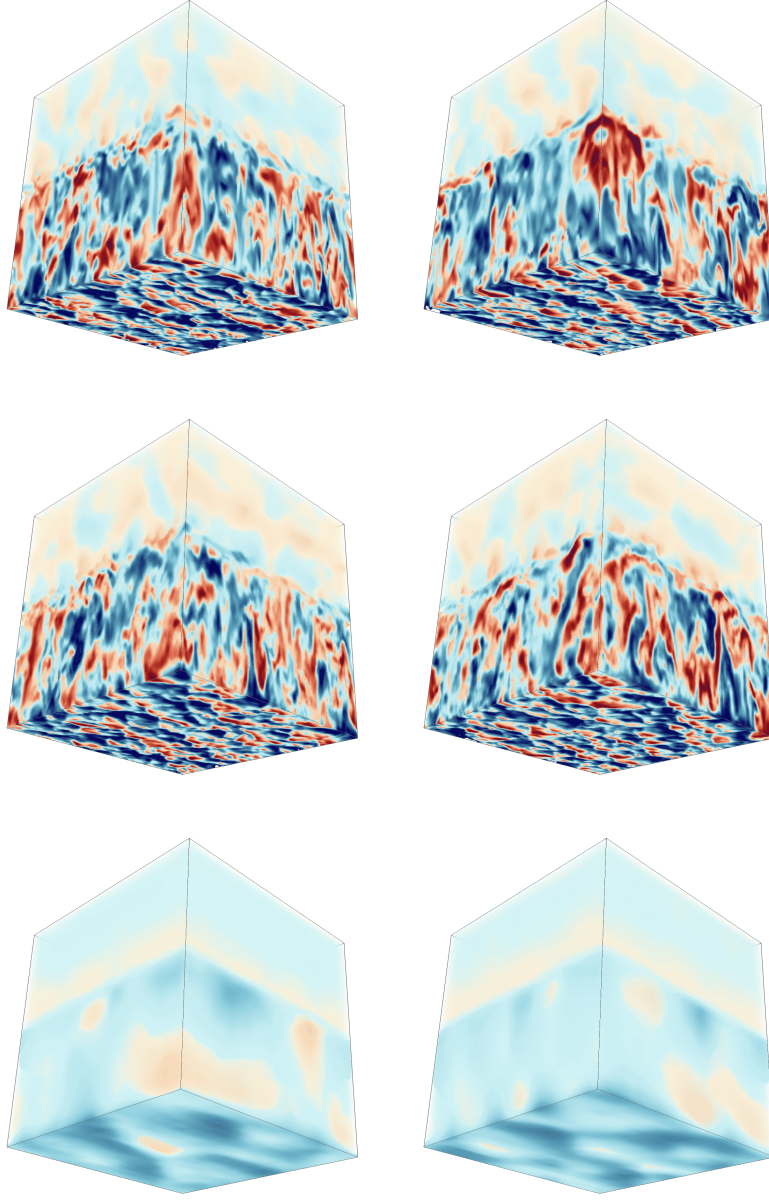
**Figure 15:** Two random samples, for the same injection velocity, of the (pointwise) vorticity intensity at time  $T = 130$  for the nozzle flow experiment. Data generated with ground truth (Top Row), GenCFD (Middle Row) and UViT (Bottom Row).



**Figure 16:** Mean of the pointwise kinetic energy at time  $T = 130$  for the nozzle flow experiment. Ground truth (Left), GenCFD (Center) and UViT. (Right).



**Figure 17:** Point PDFs of the three velocity components at the spatial point  $(0.547, 0.328, 0.271)$  at time  $T = 130$  for the nozzle flow experiment for Ground truth, GenCFD and UViT.



**Figure 18:** Two random samples, of the velocity component  $u_x$  at time  $T = 2.4$  for the convective boundary layer experiment with ground truth (Top Row), GenCFD (Middle Row) and UViT (Bottom Row).

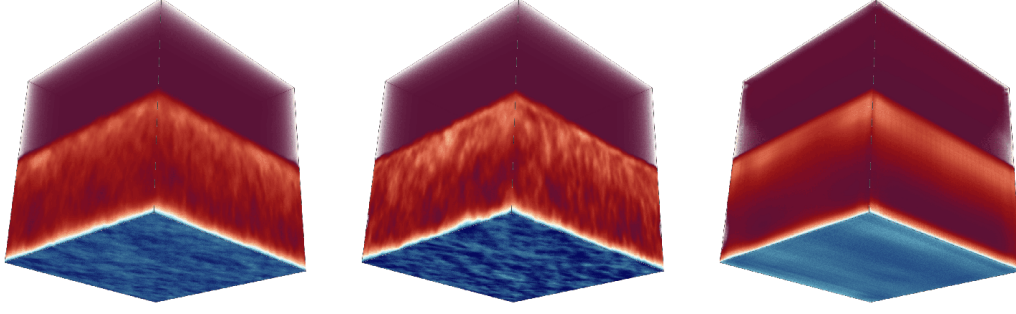


Figure 19: Mean of the velocity component  $u_x$  at time  $T = 2.4$  for the convective boundary layer experiment with ground truth (Left), GenCFD (Center) and UViT (Right).

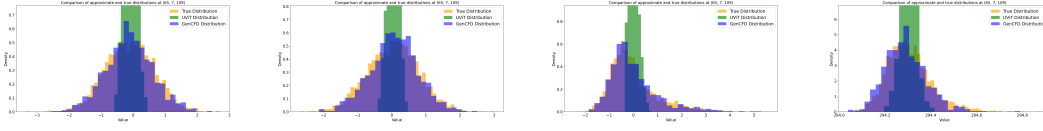


Figure 20: Point PDFs of the temperature (Left) and the three velocity components at the spatial point  $(0.508, 0.055, 0.852)$  at time  $T = 2.4$  for the convective boundary layer experiment for ground truth, GenCFD and UViT.

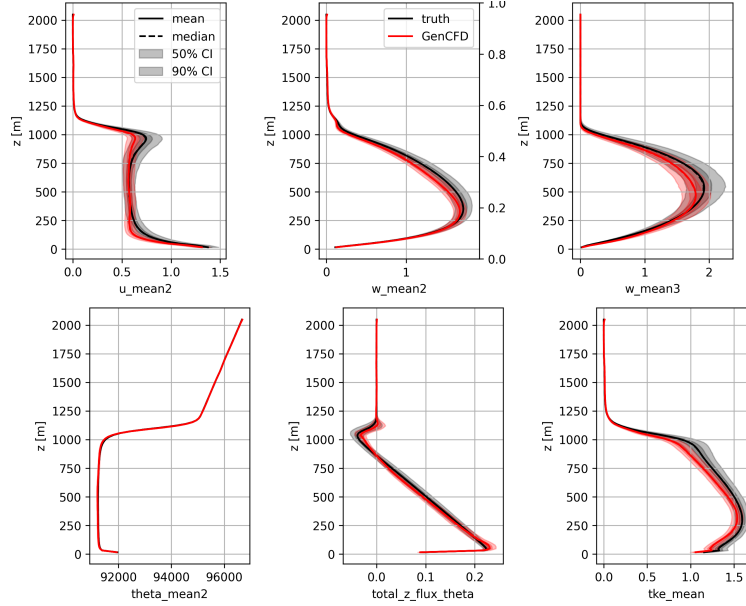
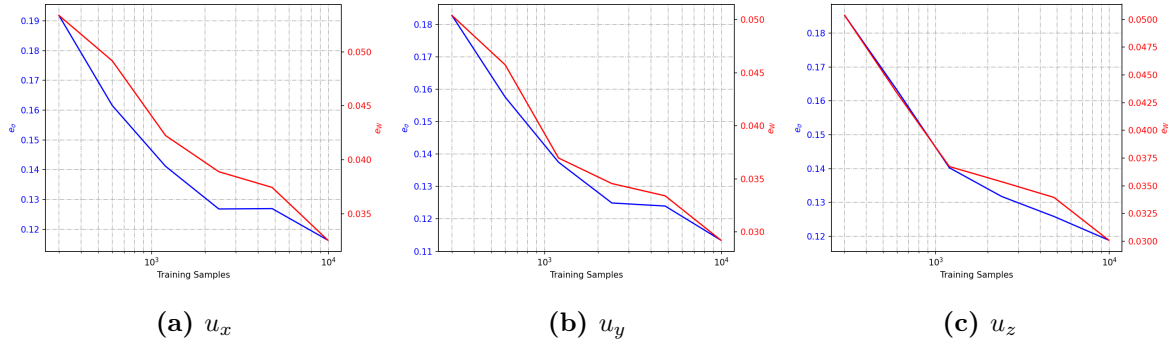
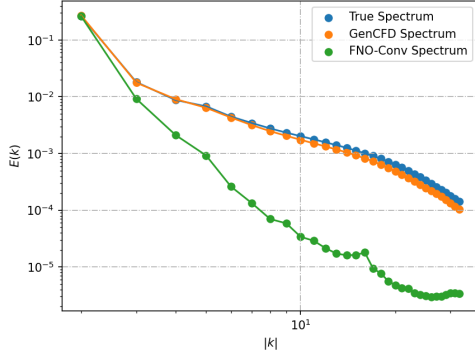


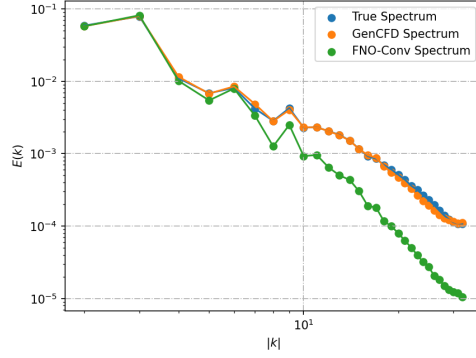
Figure 21: Profiles (horizontal statistics) for the convective boundary layer experiment for GenCFD and ground truth at final simulation time  $T = 7200$  s. From left to right and top to bottom:  $x_1$ -velocity variance, vertical velocity variance, vertical velocity skewness, potential temperature variance, vertical flux of potential temperature, and turbulent kinetic energy.



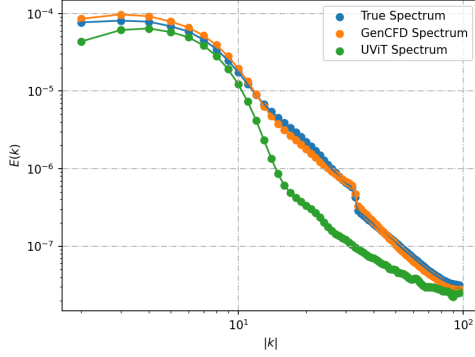
**Figure 22:** Scaling of standard deviation and Wasserstein metric error,  $e_\sigma$  and  $e_W$ , due to GenCFD, vs. number of training samples for the velocities  $u_x$ ,  $u_y$  and  $u_z$  for the cylindrical shear flow.



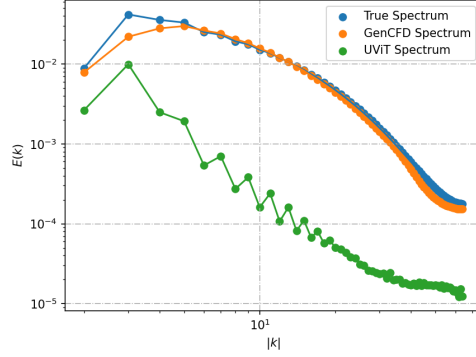
(a) Cylindrical shear flow



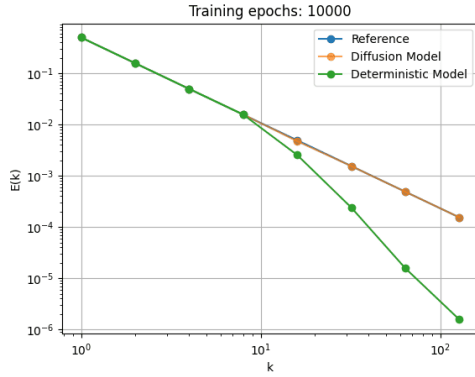
(b) Cloud-shock interaction



(c) Nozzle flow



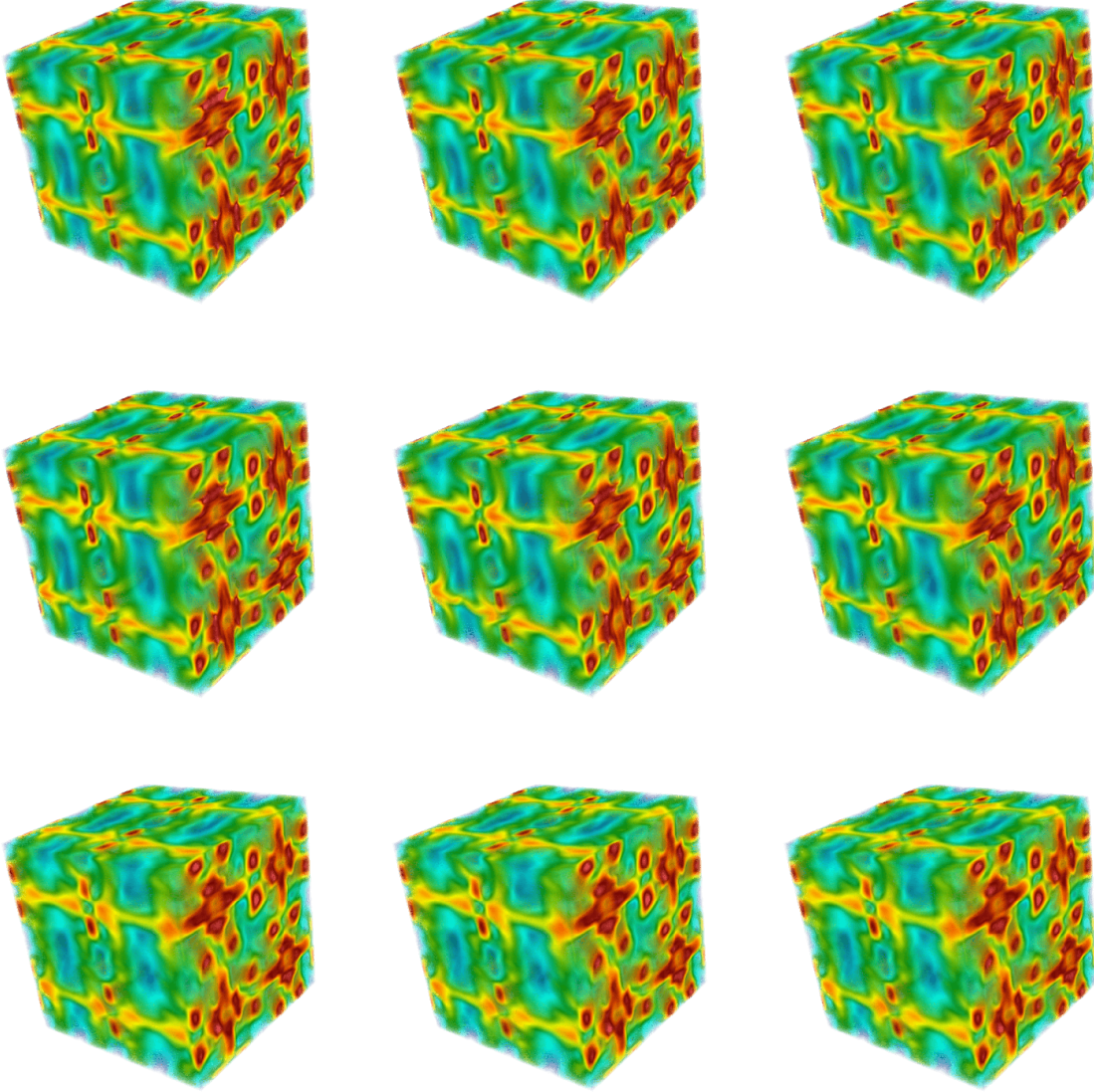
(d) Convective boundary layer



(e) Toy Problem 2

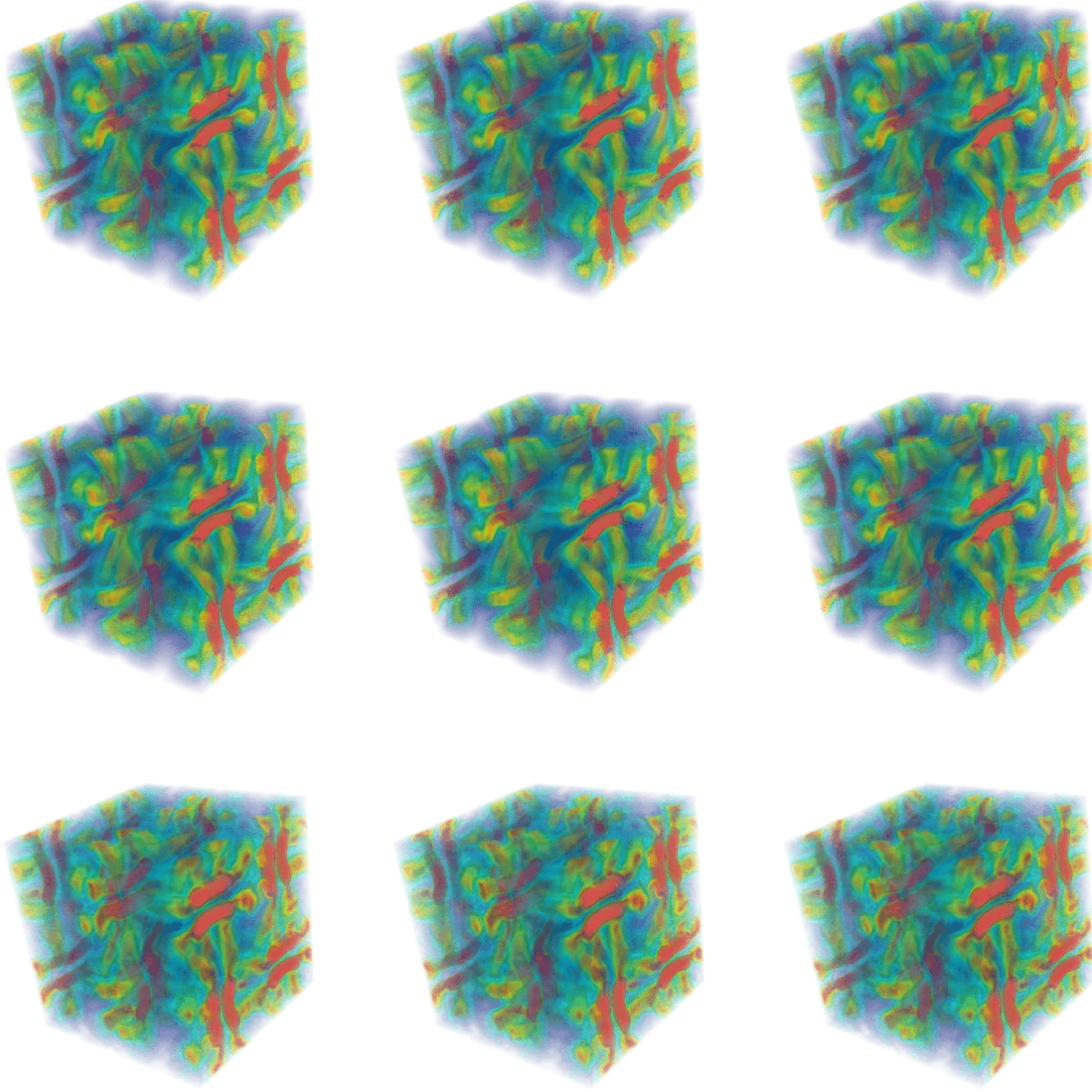
**Figure 23: Energy spectra, generated by the ground truth, GenCFD and the best-performing baseline for 4 of the 3D Datasets and the spectrum of Toy Model #2. Note that the spectrum shown here is the spectrum of the density for the cloud-shock interaction experiment.**



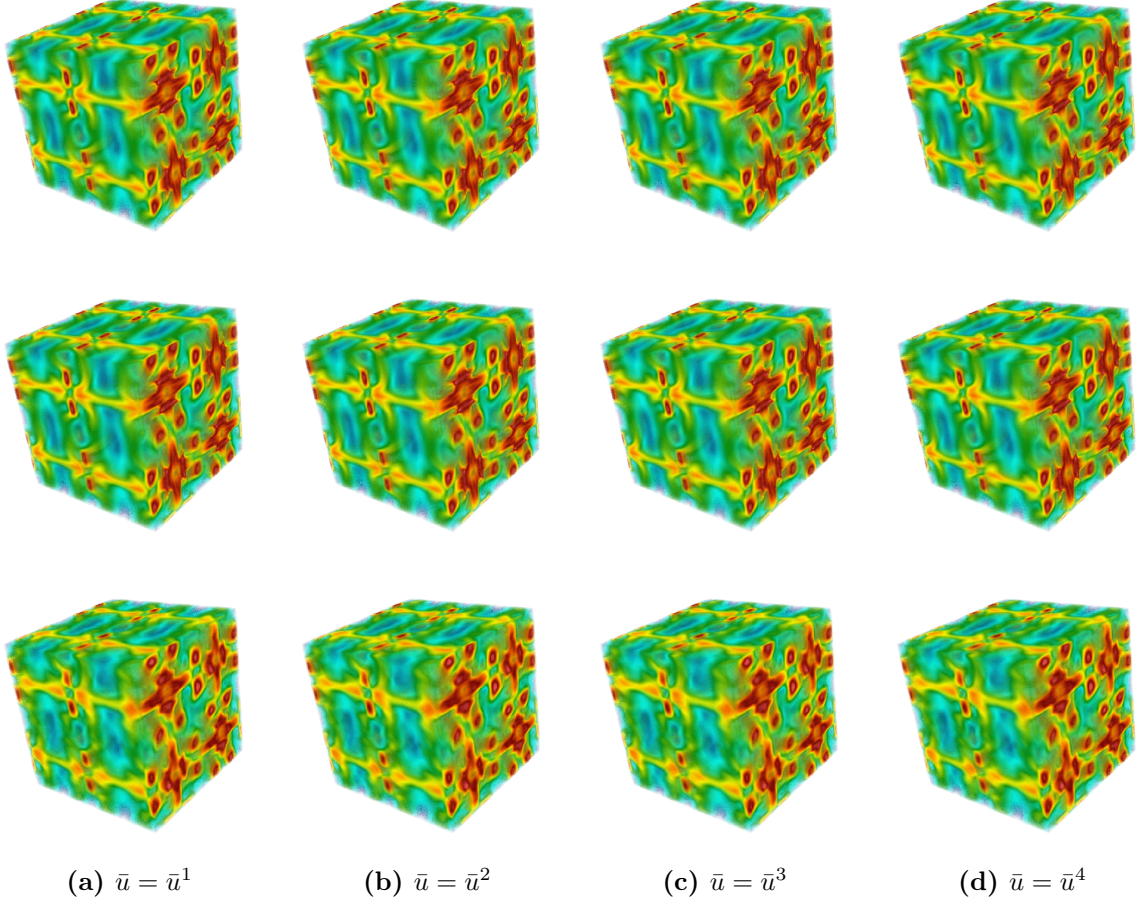


**Figure 24:** Visualization of pointwise kinetic energy for 3 randomly generated samples for the three-dimensional Taylor–Green experiment at time  $T = 0.8$  with ground truth (top row), GenCFD (middle row) and C-FNO (bottom row). The colormap for all the figures ranges from 0.0 (dark blue) to 1.0 (dark red).

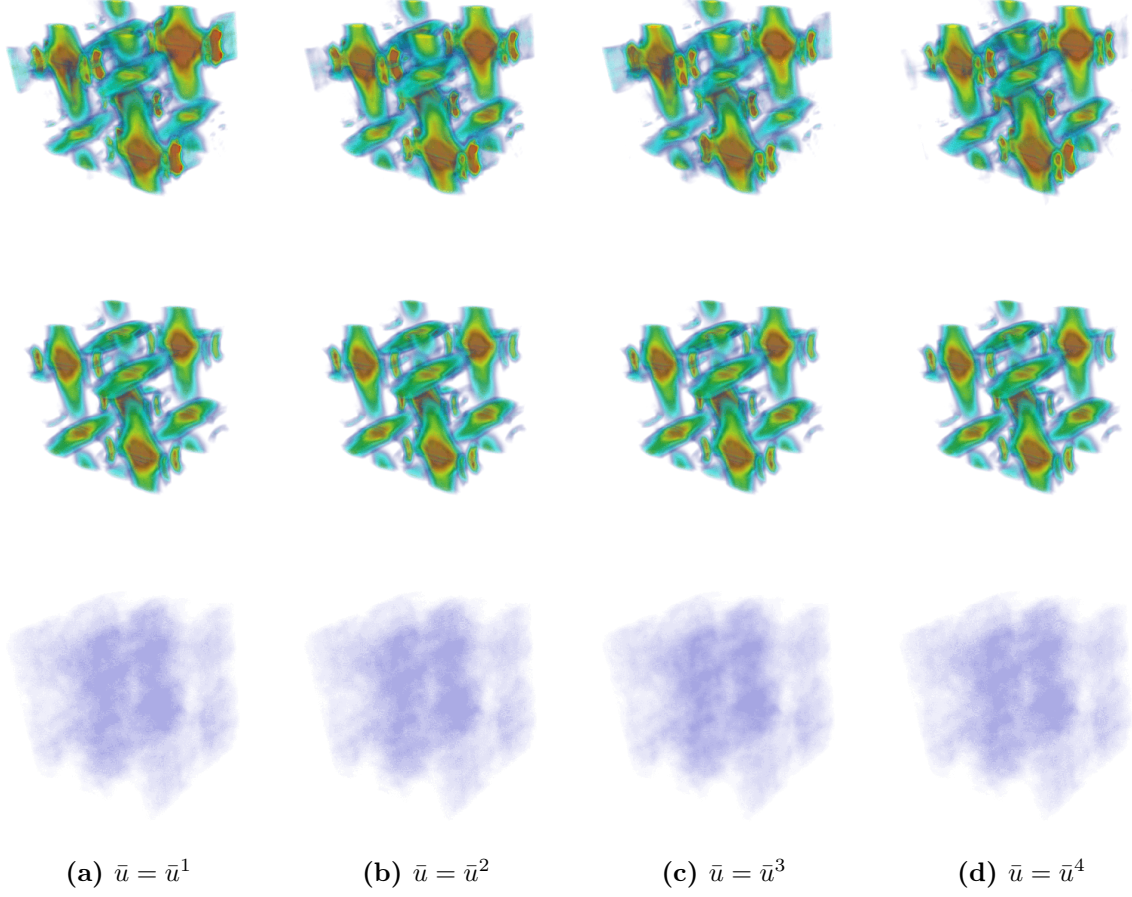




**Figure 25:** Visualization of pointwise vorticity intensity for 3 randomly generated samples for the three-dimensional Taylor–Green experiment at time  $T = 2$  with ground truth (top row), GenCFD (middle row) and C-FNO (bottom row). The colormap for the top and middle rows ranges from  $10^{-4}$  (dark blue) to 40.0 (dark red), whereas for the bottom row, it ranges from  $10^{-4}$  to 35.

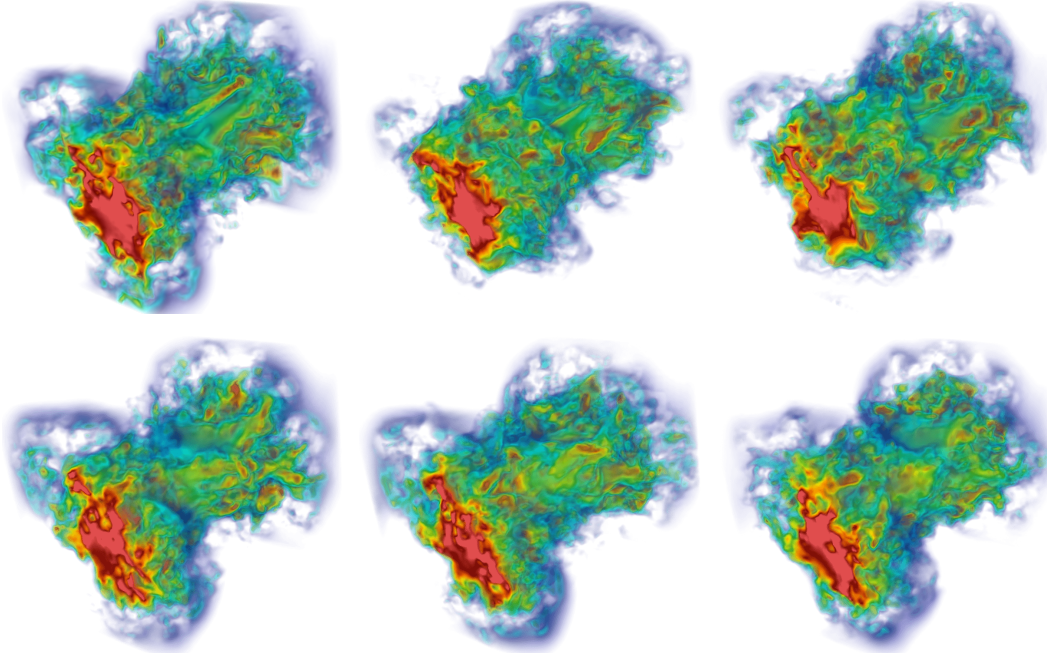


**Figure 26:** Visualization of the mean of the approximate statistical solution for the Taylor–Green experiment at time  $T = 2$ , for four different initial distributions, generated by the ground truth (top row), GenCFD (middle row) and C-FNO (bottom row). The colormap for all the figures ranges from 0.0 (dark blue) to 1.0 (dark red).

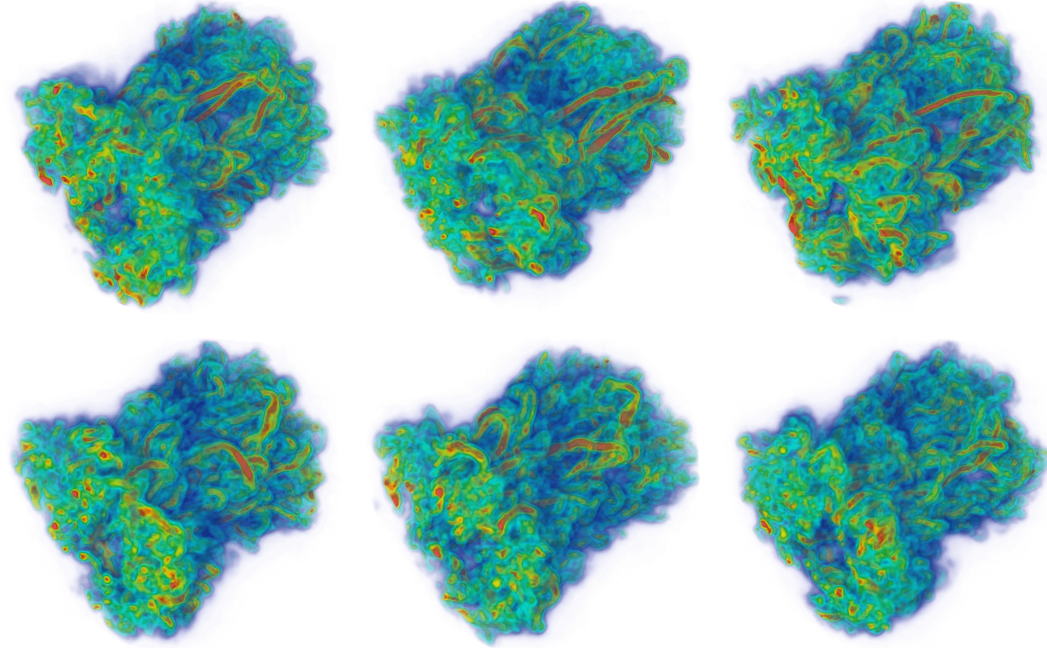


**Figure 27: Visualization of the standard deviation of the (pointwise) kinetic energy for the Taylor–Green experiment at time  $T = 0.8$ , for four different initial distributions, generated by the ground truth (top row), GenCFD (middle row) and C-FNO (bottom row). The colormap for the top and middle rows ranges from 0.05 (dark blue) to 0.15 (dark red), whereas for the bottom row, it ranges from  $1.3 \times 10^{-4}$  to  $3.5 \times 10^{-3}$ .**



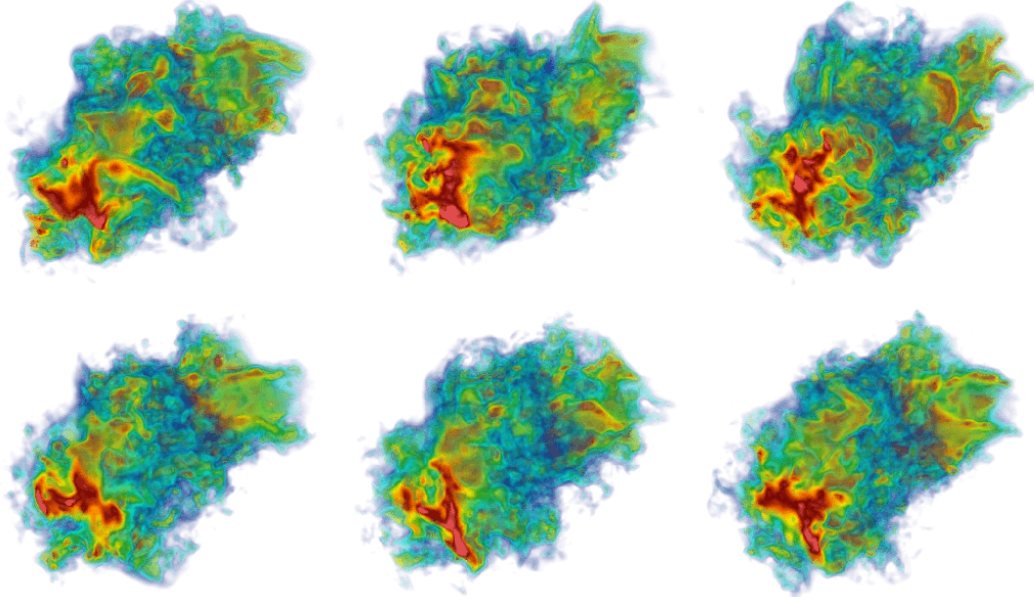


(a) Ground truth (top) and GenCFD (bottom)

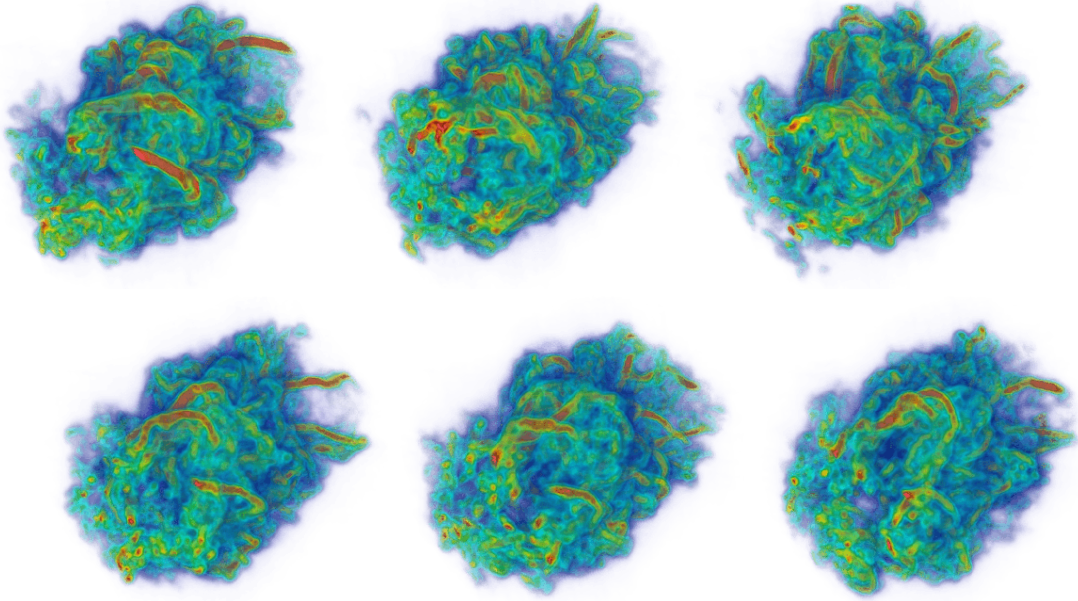


(b) Ground truth (top) and GenCFD (bottom)

**Figure 28:** Visualization of pointwise kinetic energy (a) and vorticity (b) for 3 randomly generated samples for the three-dimensional cylindrical shear flow experiment at time  $T = 1$  for an initial condition different from the one presented in Fig. 6. Colormaps are identical to the ones used in Fig. 6.



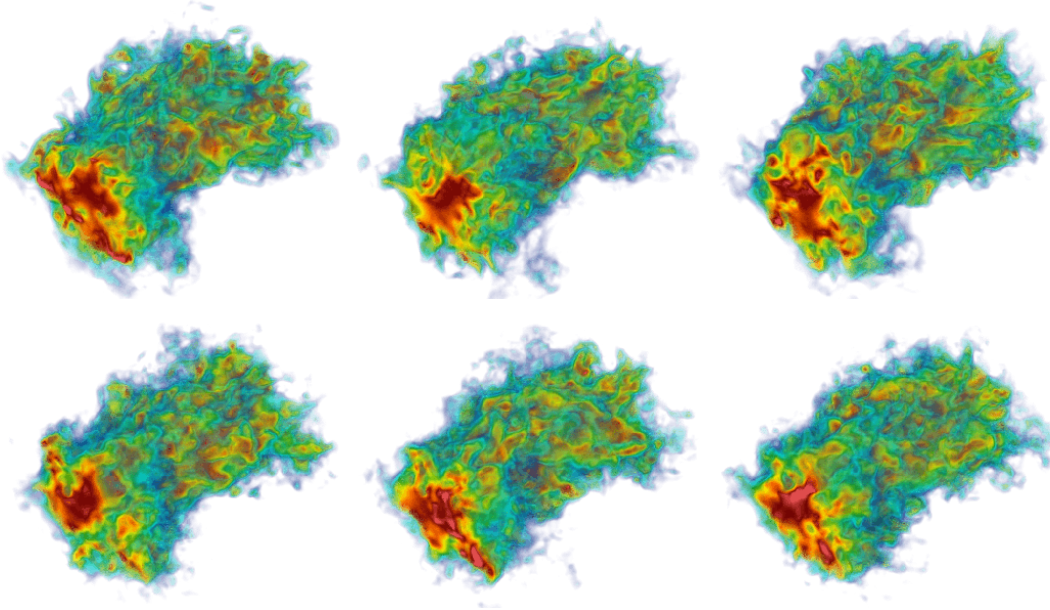
(a) Ground truth (top) and GenCFD (bottom)



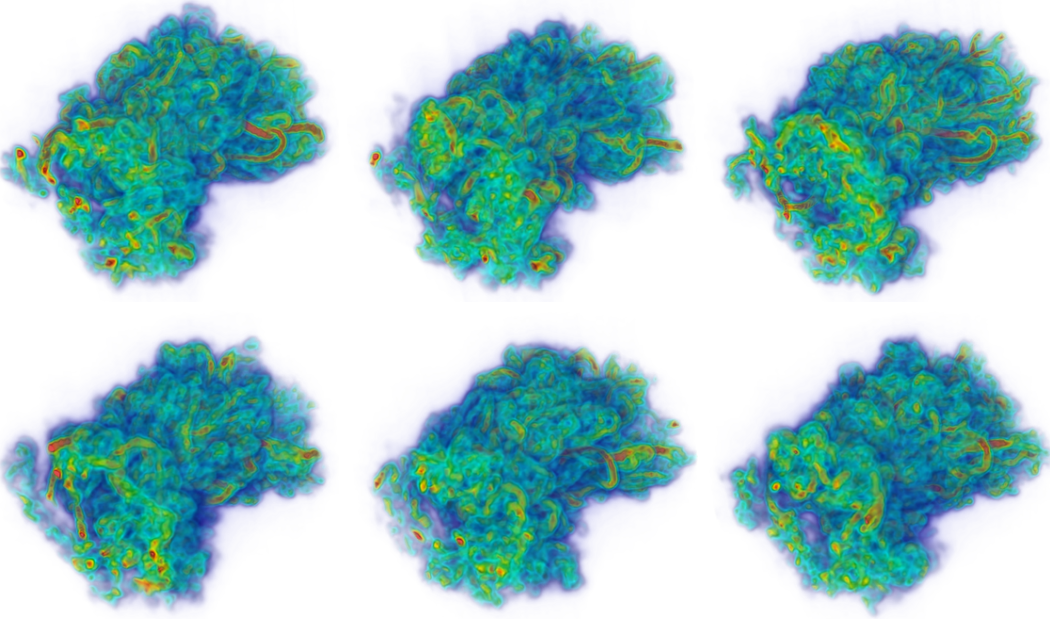
(b) Ground truth (top) and GenCFD (bottom)

**Figure 29:** Visualization of pointwise kinetic energy (a) and vorticity (b) for 3 randomly generated samples for the three-dimensional cylindrical shear flow experiment at time  $T = 1$  for an initial condition different from Figs. 6 and 28. Colormaps are identical to the ones used in Fig. 6.



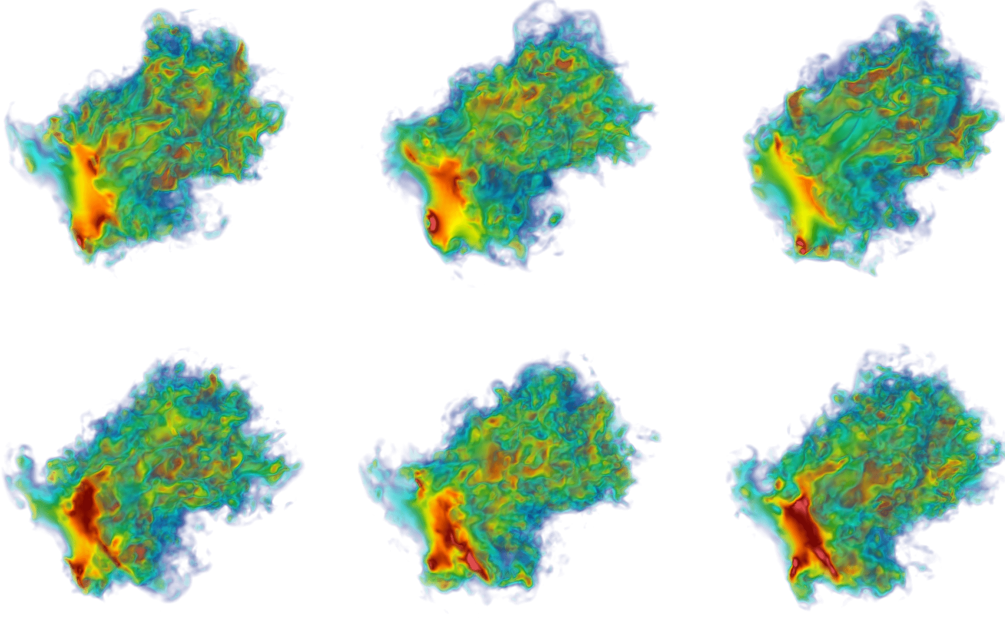


(a) Ground truth (top) and GenCFD (bottom)

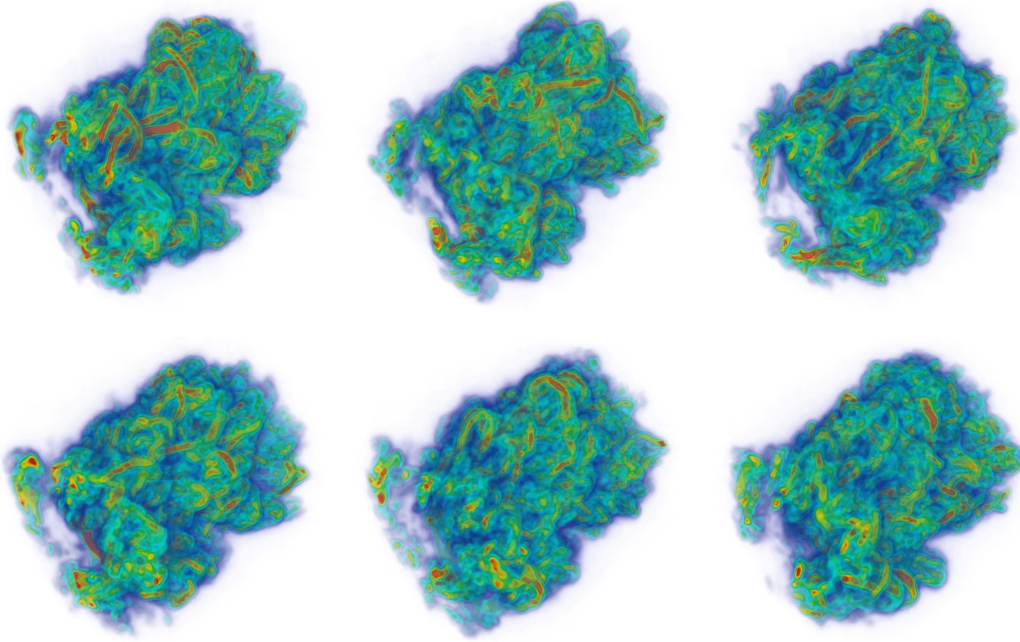


(b) Ground truth (top) and GenCFD (bottom)

**Figure 30:** Visualization of pointwise kinetic energy (a) and vorticity (b) for 3 randomly generated Samples for the three-dimensional cylindrical shear flow experiment at time  $T = 1$  for an initial condition different from Figs. 6, 28 and 29. Colormaps are identical to the ones used in Fig. 6.

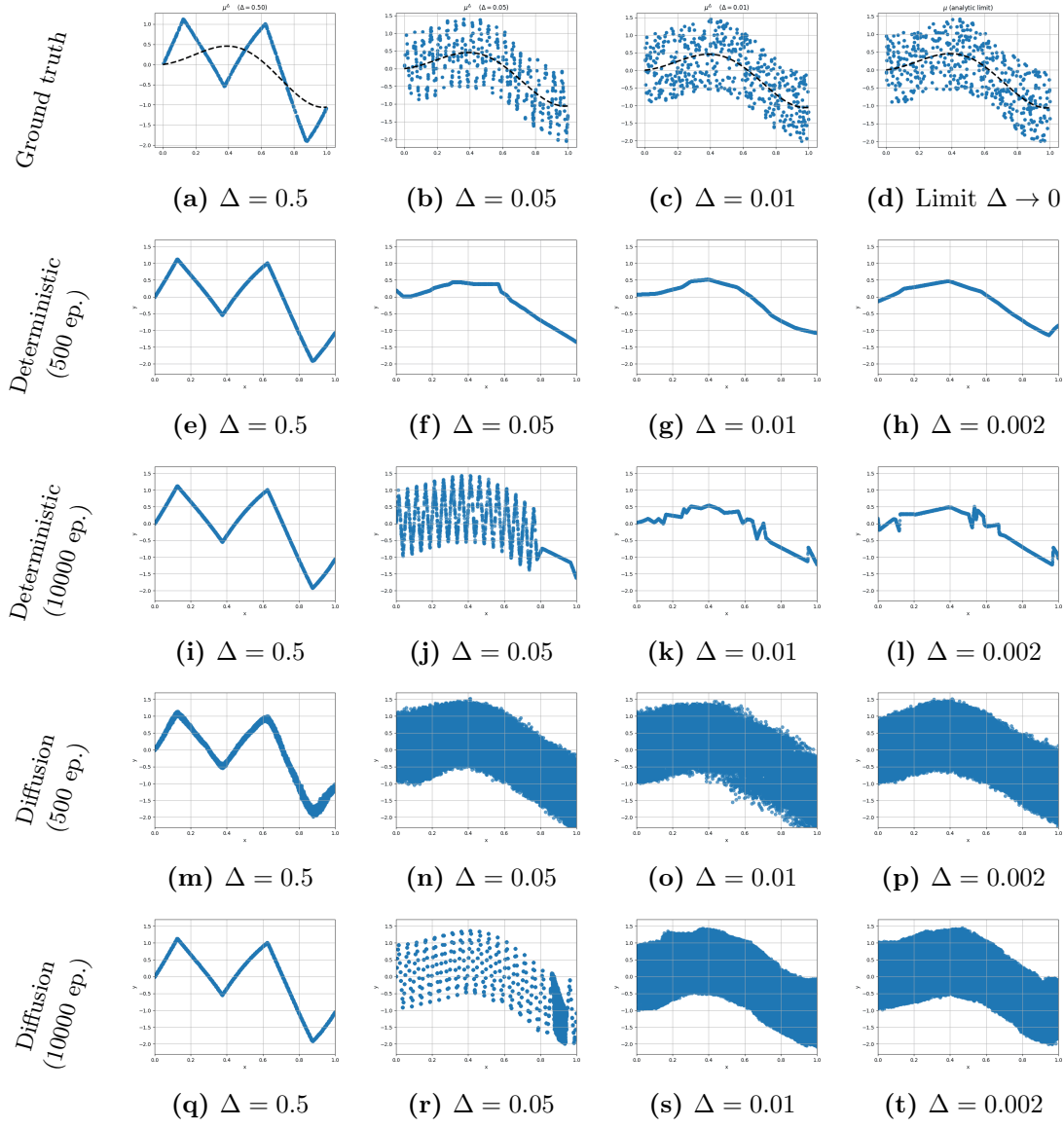


(a) Ground truth (top) and GenCFD (bottom)



(b) Ground truth (top) and GenCFD (bottom)

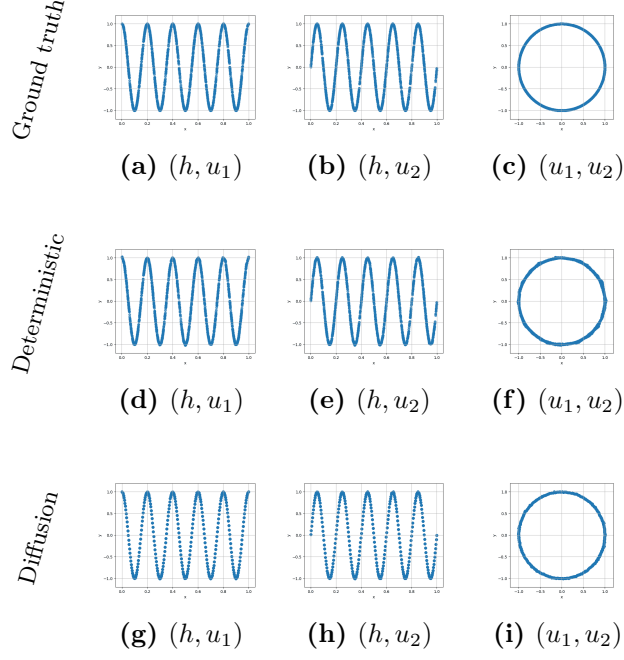
**Figure 31:** Visualization of the kinetic energy (a) of samples drawn from the ground truth and approximated conditional distribution  $p(u|\bar{u} = \bar{u}^2)$  and corresponding computed vorticity intensity (b). Colormaps are identical to the ones used in Fig. 6.



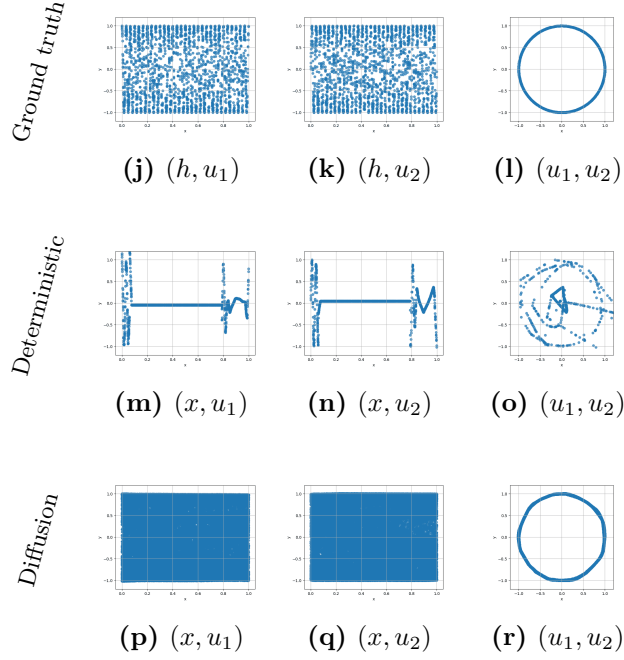
**Figure 32: Results for Toy Model #1 at different  $\Delta$ .** (Row 1): Ground truth, (Row 2): Deterministic ML model with 500 epochs of Training, (Row 3): Deterministic ML model with 10000 epochs, (Row 4): Diffusion model with 500 epochs of training, (Row 5): Diffusion model with 10000 epochs.



$k = 5$ :



$k = 30$ :



**Figure 33: Toy Model #2 at two different frequencies  $k = 5$  (Top 3 rows) and  $k = 30$  (Bottom 3 rows). Results with ground truth (Top), deterministic ML model (Middle) and diffusion model (Bottom) for each frequency.**

## References

- [1] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.
- [2] F. Bartolucci, E. de Bézenac, B Raonic, R Molinaro, S Mishra, and R Alaifari. Representation equivalent neural operators: a framework for alias-free operator learning. *arXiv:2305.19913*, 2023.
- [3] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.
- [4] J.B. Bell, P. Collela, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.
- [5] H Bijl, D Lucor, S Mishra, and C (Eds). Schwab. *Uncertainty quantification in computational fluid dynamics*, volume 92. Springer Lecture notes in Computational Science and Engineering, 2014.
- [6] Massimo Bonavita. On some limitations of current machine learning weather prediction models. *Geophysical Research Letters*, 51(12):e2023GL107377, 2024. e2023GL107377 2023GL107377.
- [7] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. Technical report, OpenAI, 2024.
- [8] Constantine M Dafermos and Constantine M Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 3. Springer, 2005.
- [9] Agnimitra Dasgupta, Harisankar Ramaswamy, Javier Murgoitio Esandi, Ken Foo, Runze Li, Qifa Zhou, Brendan Kennedy, and Assad Oberai. Conditional score-based diffusion models for solving inverse problems in mechanics. *arXiv preprint arXiv:2406.13154*, 2024.
- [10] B. Engquist and P. E. Souganidis. Asymptotic and numerical homogenization. *Acta Numerica*, 17:147–190, 2008.
- [11] Ling Feng, Lin Zhang, and Choy Heng Lai. Optimal machine intelligence at the edge of chaos, 2020.
- [12] Richard P Feynman, Robert B Leighton, and Matthew Sands. *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*, volume 1. Basic books, 2015.

- [13] U. S. Fjordholm, S. Mishra, and E. Tadmor. On the computation of measure-valued solutions. *Acta Numer.*, 25:567–679, 2016.
- [14] Ulrik S Fjordholm, Roger Käppeli, Siddhartha Mishra, and Eitan Tadmor. Construction of approximate entropy measure-valued solutions for hyperbolic systems of conservation laws. *Foundations of Computational Mathematics*, 17(3):763–827, 2017.
- [15] Ulrik S. Fjordholm, Samuel Lanthaler, and Siddhartha Mishra. Statistical solutions of hyperbolic conservation laws: Foundations. *Archive for Rational Mechanics and Analysis*, 226(2):809–849, Nov 2017.
- [16] Ulrik S. Fjordholm, Kjetil O. Lye, Siddhartha Mishra, and Franziska Weber. Statistical solutions of hyperbolic systems of conservation law: Numerical approximation. *Mathematical Models and Methods in Applied Sciences*, 30(3):539–609, 2020.
- [17] Ulrik S. Fjordholm, Siddhartha Mishra, and Franziska Weber. On the vanishing viscosity limit of statistical solutions of the incompressible navier-stokes equations. 2022.
- [18] Ciprian Foias, Oscar Manley, Ricardo Rosa, and Roger Temam. *Navier-Stokes Equations and Turbulence*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2001.
- [19] Uriel Frisch. *Turbulence: The Legacy of A.N. Kolmogorov*. Cambridge University Press, 1995.
- [20] Han Gao, Xu Han, Xiantao Fan, Luning Sun, Li-Ping Liu, Lian Duan, and Jian-Xun Wang. Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation. *Computer Methods in Applied Mechanics and Engineering*, 427:117023, 2024.
- [21] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning for forecasting the dynamics of complex systems. *arXiv preprint arXiv:2402.17157*, 2024.
- [22] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning of the solution of parametric partial differential equations using guided diffusion models and virtual observations. *arXiv preprint arXiv:2408.00157*, 2024.
- [23] S. Ghoshal. An analysis of numerical errors in large eddy simulations of turbulence. *J. Comput. Phys.*, 125(1):187–206, 1996.
- [24] Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes, 2024.
- [25] J. S. Hesthaven. *Numerical methods for conservation laws: From analysis to algorithms*. SIAM, 2018.
- [26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [27] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.
- [28] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers, 2019.
- [29] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
- [30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020. arXiv:2001.08361 [cs, stat].
- [31] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.
- [32] Mathias J. Krause, Adrian Kummerländer, Samuel J. Avis, Halim Kusumaatmaja, Davide Dapelo, Fabian Klemens, Maximilian Gaedtke, Nicolas Hafen, Albert Mink, Robin Trunk, Jan E. Marquardt, Marie-Luise Maier, Marc Haussmann, and Stephan Simonis. OpenLB—Open source lattice Boltzmann code. *Computers & Mathematics with Applications*, 81:258–288, 2021.
- [33] Adrian Kummerländer, Fedor Bukreev, Simon F. R. Berg, Marcio Dorn, and Mathias J. Krause. Advances in Computational Process Engineering using Lattice Boltzmann Methods on High Performance Computers. In Wolfgang E. Nagel, Dietmar H. Kröner, and Michael M. Resch, editors, *High Performance Computing in Science and Engineering '22*, pages 233–247, Cham, 2024. Springer Nature Switzerland.
- [34] Adrian Kummerländer, Fedor Bukreev, Dennis Teutscher, Marcio Dorn, and Mathias J. Krause. Optimization of Single Node Load Balancing for Lattice Boltzmann Methods on Heterogeneous High Performance Computers. *Available at SSRN*, 2024.
- [35] Adrian Kummerländer, Tim Bingert, Fedor Bukreev, Luiz Eduardo Czelusniak, Davide Dapelo, Nicolas Hafen, Marc Heinzelmann, Shota Ito, Julius Jeßberger, Halim Kusumaatmaja, Jan E. Marquardt, Michael Rennick, Tim Pertzel, František Prinz, Martin Sadric, Maximilian Schecher, Stephan Simonis, Pascal Sitter, Dennis Teutscher, Mingliang Zhong, and Mathias J. Krause. OpenLB Release 1.7: Open Source Lattice Boltzmann Code, February 2024.
- [36] L. D. Landau and E. M. Lipschitz. *Fluid Mechanics, 2nd edition*. Butterworth Heinemann, 1987.
- [37] S Lanthaler, S Mishra, and F Weber. On bayesian data assimilation for pdes with ill-posed forward problems. *Inverse Problems*, 38(8):085012, jul 2022.

- [38] Samuel Lanthaler, Siddhartha Mishra, and Carlos Parés-Pulido. Statistical solutions of the incompressible euler equations. *Mathematical Models and Methods in Applied Sciences*, 31(02):223–292, 2021.
- [39] Samuel Lanthaler, Siddhartha Mishra, and Carlos Parés-Pulido. Statistical solutions of the incompressible euler equations. *Mathematical Models and Methods in Applied Sciences*, 31(02):223–292, February 2021.
- [40] Randall J LeVeque. *Numerical methods for conservation laws*, volume 3. Springer, 1992.
- [41] Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024.
- [42] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.
- [43] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [44] M. Luskin. On the computation of crytalline microstructure. *Acta Numerica*, 5:191–257, 1996.
- [45] Kjetil O. Lye. *Computation of statistical solutions of hyperbolic systems of conservation laws*. PhD thesis, 2020.
- [46] Andrew J. Majda and Andrea L. Bertozzi. *Vorticity and Incompressible Flow*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2001.
- [47] Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual diffusion modeling for km-scale atmospheric downscaling, 2024.
- [48] S. Mishra, Ch. Schwab, and J. Šukys. Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *J. Comput. Phys.*, 231(8):3365–3388, 2012.
- [49] Chin-Hoh Moeng and Peter P. Sullivan. A comparison of shear- and buoyancy-driven planetary boundary layer flows. *Journal of the Atmospheric Sciences*, 51(7):999–1022, 1994. Publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences.
- [50] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

- [51] Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating neural operators with diffusion models improves spectral representation in turbulence modeling. *arXiv preprint arXiv:2409.08477*, 2024.
- [52] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *ICML '13*, page III–1310–III–1318. JMLR.org, 2013.
- [53] Olivier Pauluis. Thermodynamic consistency of the anelastic approximation for a moist atmosphere. *Journal of the Atmospheric Sciences*, 65(8):2719–2729, 2008.
- [54] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning Mesh-Based Simulation with Graph Networks, June 2021. *arXiv:2010.03409 [cs]*.
- [55] Stephen B Pope. *Turbulent flows*. Cambridge University Press, 2001.
- [56] Kyle G. Pressel, Colleen M. Kaul, Tapio Schneider, Zhihong Tan, and Siddhartha Mishra. Large-eddy simulation in an anelastic framework with closed water and entropy balances. *Journal of Advances in Modeling Earth Systems*, 7(3):1425–1456, 2015.
- [57] Kyle G. Pressel, Siddhartha Mishra, Tapio Schneider, Colleen M. Kaul, and Zhihong Tan. Numerics and subgrid-scale modeling in large eddy simulations of stratocumulus clouds. *Journal of Advances in Modeling Earth Systems*, 9(2):1342–1365, 2017.
- [58] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Gencast: Diffusion-based ensemble forecasting for medium-range weather, 2024.
- [59] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019.
- [60] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [61] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A Navier-Stokes informed deep learning framework for assimilating flow visualization data. *Science*, 367:1026–1030, 2020.
- [62] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.
- [63] Bogdan Raonić, Roberto Molinaro, Tobias Rohner, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators. *arXiv preprint arXiv:2302.01178*, 2023.

- [64] Tobias Rohner and Siddhartha Mishra. Efficient computation of large-scale statistical solutions to incompressible fluid flows. In *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '24*. ACM, June 2024.
- [65] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [67] François Rozet and Gilles Louppe. Score-based data assimilation. *Advances in Neural Information Processing Systems*, 36:40521–40541, 2023.
- [68] Pierre Sagaut. *Large Eddy Simulations for Incompressible Flows*. Springer, 2006.
- [69] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [70] T. Schneider, S. Lan, A. Stuart, and J. Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44:12396–12417, 2017.
- [71] Tapio Schneider, João Teixeira, Christopher S. Bretherton, Florent Brient, Kyle G. Pressel, Christoph Schär, and A. Pier Siebesma. Climate goals and computing the future of clouds. *Nature Climate Change*, 7(1):3–5, 2017.
- [72] S. Simonis and S. Mishra. Computing statistical Navier–Stokes solutions. In Rémi Abgrall, Mauro Garavello, Mária Lukáčová-Medvid’ová, and Konstantina Trivisa, editors, *Hyperbolic Balance Laws: Interplay between Scales and Randomness*, number 1 in Oberwolfach Report 21, pages 567–656. EMS Press, 2024.
- [73] Yang Song and Stefano Ermon. *Generative modeling by estimating gradients of the data distribution*. Curran Associates Inc., Red Hook, NY, USA, 2019.
- [74] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.
- [75] Andrew M Stuart. Inverse problems: a Bayesian perspective. *Acta numerica*, 19:451–559, 2010.



- [76] Peter P. Sullivan and Edward G. Patton. The Effect of Mesh Resolution on Convective Boundary Layer Statistics and Structures Generated by Large-Eddy Simulation. *Journal of the Atmospheric Sciences*, 68(10):2395–2415, 2011.
- [77] Eitan Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 26(1):30–44, 1989.
- [78] Wenpin Tang and Hanyang Zhao. Score-based diffusion models via stochastic differential equations – a technical tutorial, 2024.
- [79] G. I. Taylor and A. E. Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 158(895):499–521, 1937.
- [80] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
- [81] Zhong Yi Wan, Ricardo Baptista, Anudhyan Boral, Yi-Fan Chen, John Anderson, Fei Sha, and Leonardo Zepeda-Núñez. Debias coarsely, sample conditionally: Statistical downscaling through optimal transport and probabilistic diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47749–47763. Curran Associates, Inc., 2023.
- [82] Gefan Yang and Stefan Sommer. A denoising diffusion model for fluid field prediction. *arXiv preprint arXiv:2301.11661*, 2023.
- [83] Borong Zhang, Martín Guerra, Qin Li, and Leonardo Zepeda-Núñez. Back-projection diffusion: Solving the wideband inverse scattering problem with diffusion models. *arXiv preprint arXiv:2408.02866*, 2024.
- [84] Lin Zhang, Ling Feng, Kan Chen, and Choy Heng Lai. Edge of chaos as a guiding principle for modern neural network training, 2021.

## Acknowledgments

This work was supported by a computing grant from the Swiss National Supercomputing Centre (CSCS) under project ID 1217 as well as a part of the Swiss AI Initiative under project ID a01 on Alps. The authors thank Dr. Emmanuel de Bézenac (INRIA, Paris) and Prof. Sebastian Schemm (U. Cambridge) for their inputs. S. S. gratefully acknowledges the support from KHYS at KIT through a ConYS grant and the computing time (NHR Project ID p0023756) made available on the high-performance computer HoreKa funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research (Germany).