Unfolding Time: Generative Modeling for Turbulent Flows in 4D

Abdullah Saydemir [* 1]   **Marten Lienen** [* 1 2]   **Stephan Günnemann** [1 2]

Abstract

A recent study in turbulent flow simulation demonstrated the potential of generative diffusion models for fast 3D surrogate modeling. This approach eliminates the need for specifying initial states or performing lengthy simulations, significantly accelerating the process. While adept at sampling individual frames from the learned manifold of turbulent flow states, the previous model lacks the capability to generate sequences, hindering analysis of dynamic phenomena. This work addresses this limitation by introducing a 4D generative diffusion model and a physics-informed guidance technique that enables the generation of realistic sequences of flow states. Our findings indicate that the proposed method can successfully sample entire subsequences from the turbulent manifold, even though generalizing from individual frames to sequences remains a challenging task. This advancement opens doors for the application of generative modeling in analyzing the temporal evolution of turbulent flows, providing valuable insights into their complex dynamics.

1. Introduction

Computational Fluid Dynamics (CFD) has been a prominent research direction due to vast interest across fields and the challenge of developing new architectures grounded in its broad theoretical framework (Pope, 2001). Recent advancements in applied AI unlatched fast approximation and analysis tools for fluids, yielding plenty of models that predict the state of particles and the corresponding velocity and pressure fields at a future timestep, autoregressively (Li

---

[*]Equal contribution   [1]Department of Informatics, Technical University of Munich, Germany [2]Munich Data Science Institute, Technical University of Munich, Germany. Correspondence to: Marten Lienen <m.lienen@tum.de>.

et al., 2020; Chen et al., 2020). However, the primary objective of CFD simulations often involves exploring the manifold of all possible flow states rather than determining the exact trajectory of a particular flow (Lienen et al., 2024). Lienen et al. (2024) demonstrate that a generative model can effectively sample from this manifold directly, eliminating the need for iterative simulations over thousands of steps, saving time and computational resources.

Yet, in many situations turbulent flow simulations are used to investigate the development and decay of vortices, location of separation regions, and reattachment points, i.e. properties that are inherently linked to temporal variations. The model proposed by Lienen et al. (2024) is capable of providing high-quality snapshots of the flow both visually and according to the physically-motivated metrics they propose. However, it generates independent snapshots of the simulation instead of sequences, which are necessary to investigate dynamic phenomena. Furthermore, their model does not explicitly rely on any physical knowledge.

In theory, the physics of the simulation are encoded in the data and, therefore, a perfect generative model would have implicitly learned the Navier-Stokes equation. In practice, a generative model oblivious to the underlying physics may generate samples that are reasonably close to real data while violating the physics in important ways. To address such deviations from reality in the samples generated by the model, physical knowledge can be infused into the model in a variety of ways (Karniadakis et al., 2021) such as physically-enhanced loss functions (Oldenburg et al., 2022), integrating the model into a numerical solver (Kochkov et al., 2021), data augmentation (Lu et al., 2020) or integrating known equations into the structure of the model itself (Lienen & Günnemann, 2022).

We extend the approach from Lienen et al. to generate temporally coherent subsequences of simulations. In addition to extending the generative approach to simulation to dynamic, time-dependent phenomena, we also propose a physics-informed sampling procedure. In summary, our contributions are:

- We introduce an efficient generative diffusion model for 4D turbulent flow data by combining 3D U-Net with ConvGRU in Section 3.
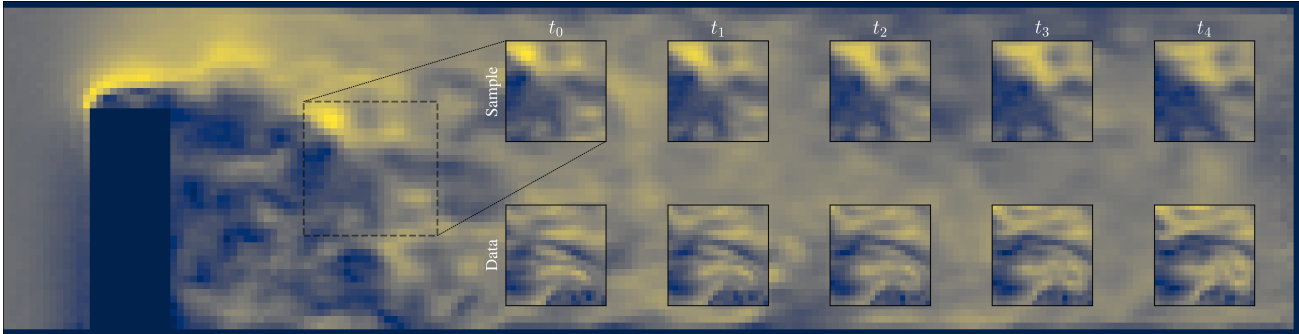- In Section 3.1, we derive a physics-informed sampling

Figure 1: A time-varying 3D turbulent flow simulation generated by our model. Visualization shows the magnitude of the velocity field $\boldsymbol{u}$. The insets show that the flow evolves coherently at a similar rate as a ground-truth sample of the same region.

procedure for diffusion models from the Navier-Stokes equation and classifier guidance.

- We show experimentally in Section 4 that simulations generated with our 4D model are of comparable quality to independent snapshots from a 3D model. Our proposed physics-informed guidance improves the sample quality in terms of their turbulent kinetic energy spectra.

2. Background

2.1. Dataset

We use the dataset of 3D flow simulations introduced by (Lienen et al., 2024). It consists of 45 simulations in a 0.4x0.1x0.1 cm channel, discretized into a regular grid of 192x48x48 cells. Each simulation has a differently shaped object placed near the inlet, blocking part of the flow and creating turbulent flow downstream. The simulations contain 0.5s of physical time sampled at regular steps of 10^{-4} seconds for 5000 flow states per simulation. At an inflow velocity of $20\,\mathrm{m\,s^{-1}}$, the fluid flows 25 channel lengths per simulation and with a low viscosity of $\nu = 10^{-4}$ achieves a Reynolds number of about 2×10^5, well above the threshold for turbulence. Yet, the sampling steps are sufficiently finely placed that the average travel distance of the fluid along the channel between two simulation snapshots is just 2mm, roughly the diameter of a cell on the simulation grid. This means that subsequences of the simulation show a coherent evolution of flow states and the dataset is well-suited to the study of dynamical phenomena in the flow.

For each cell $\mathbf{i} = (i, j, k)$ in the simulation grid and timestep t, we have a three-dimensional velocity vector $\boldsymbol{u}_{t,\mathbf{i}} \in \mathbb{R}^3$ and a scalar pressure $p_{t,\mathbf{i}}$.

2.2. Generative Turbulence Simulation

Lienen et al. (2024) define the task of generative turbulence simulation as sampling from the distribution of all possible

flow states

$$\mathrm{p}((\boldsymbol{u} \parallel p) \mid \mathcal{B}, t \geq t_{\mathrm{turb}}) \qquad (1)$$

beyond an initial transient phase to turbulence of length t_{turb} given the boundary conditions $\mathcal{B}$. Training a generative model $\mathrm{p}_{\boldsymbol{\theta}}$ on this distribution is a viable substitute for numerical simulations to explore the manifold of all possible flow states, because turbulent flows are independent of their initial conditions and ergodic, i.e. traverse all possible states in the infinite simulation time limit (Galanti & Tsinober, 2004).

2.3. Generative Diffusion Models

Denoising Diffusion Probabilistic Models (DDPM) are a recently developed class of generative models with great generative capacity and training stability (Ho et al., 2020; Sohl-Dickstein et al., 2015) that have been extended to diverse domains from molecules (Hoogeboom et al., 2022) over adversarial attacks (Kollovieh et al., 2023b) to point processes (Lüdke et al., 2023). DDPMs consist of a forward process $\mathrm{q}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$, which transforms a sample $\boldsymbol{x}_0 \sim \mathrm{p}$ from the data distribution iteratively into a sample $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ of Gaussian noise, and a learned reverse process $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, which is trained to reverse the forward process and remove noise. The training objective is to minimize the KL Divergence between the learnable reverse process $\mathrm{p}_{\boldsymbol{\theta}}$ and the forward process q.

3. Method

Lienen et al. (2024) train a generative model to sample turbulent flow snapshots from the distribution in Equation (1). While this allows practitioners to explore the manifold of all possible flow states for some boundary conditions, e.g. a car engine design, their approach restricts users to exploring time-independent phenomena in the flow such as surface pressures. To make their approach viable for the analysis of dynamic phenomena such as mixing times between
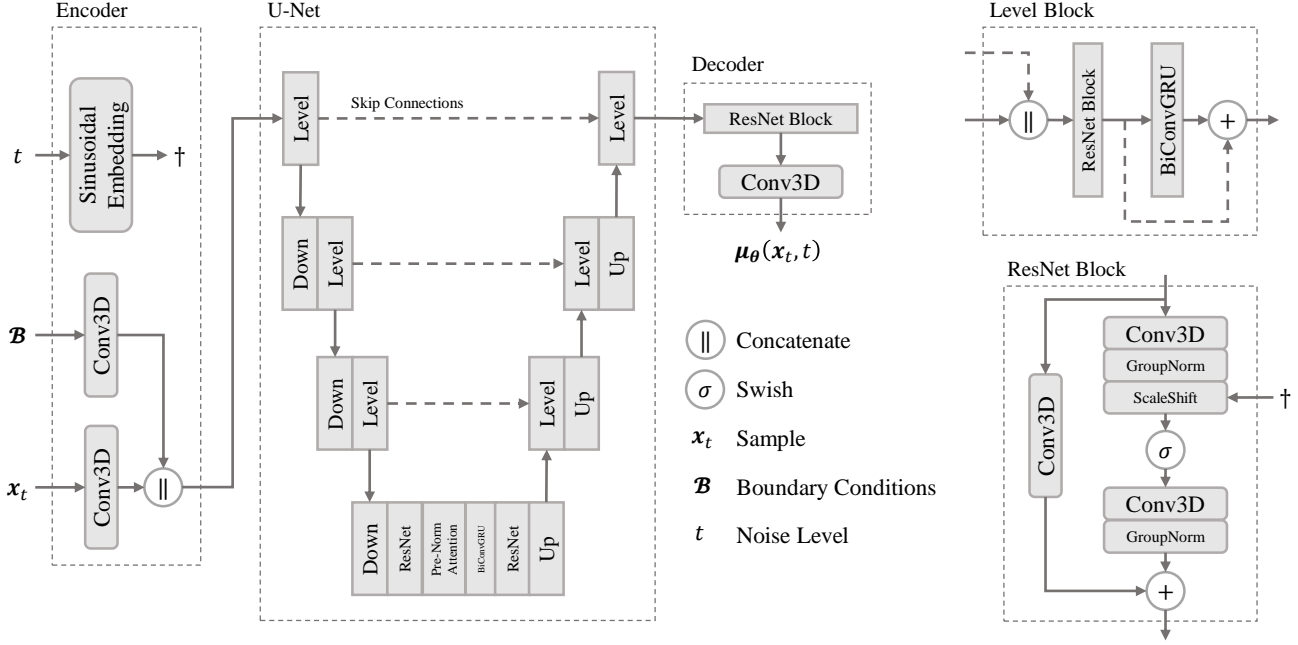
2

Figure 2: Architecture of our model for the denoised mean prediction $\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ in the DDPM framework. The model synchronizes the denoising process along the time dimension at each down- and up-sampling level of the U-Net with bi-directional ConvGRU layers.

chemicals, temporally coherent samples are required.

We extend their approach by considering the distribution

$$\mathrm{p}(\{(\boldsymbol{u}_{t'} \parallel p_{t'})\}_{t'=t+i \cdot \Delta t, i=0...k} \mid \mathcal{B}, t \geq t_{\mathrm{turb}}) \quad (2)$$

of k flow states separated by a time step of Δt. In effect, we are training a generative model $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x})$ for 4D data $\boldsymbol{x}_{tijk}$, i.e. 3D data varying across time, where $\boldsymbol{x} = (\boldsymbol{u} \parallel p)$.

To do so, we extend the TurbDiff model from Lienen et al. (2024). TurbDiff is an instance of DDPM that uses a 3D-U-Net (Ronneberger et al., 2015; Çiçek et al., 2016) to learn the reverse process $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$. At each level of the U-Net, TurbDiff uses 3D convolutions to be parameter efficient. While such an approach can be directly extended to 4D convolutions in principle (Giannopoulos et al., 2022), it is too costly in terms of runtime and memory. For the same reason, we also cannot rely on temporal attention as applied by Ho et al. (2022).

Instead, we stay with 3D convolutions and use a bi-directional sequence model along the time dimension. This can be interpreted as k 3D models communicating along the time dimension to ensure the temporal coherence of the sequence of generated 3D flow states. The bi-directionality increases the information flow so that state $\boldsymbol{x}_{t'}$ and $\boldsymbol{x}_{t'+1}$ can converge jointly towards a sample during the denoising process, instead of $\boldsymbol{x}_{t'+1}$ having to "follow" $\boldsymbol{x}_{t'}$ as it would be with uni-directional temporal communication.

Between ConvLSTM (Shi et al., 2015) and ConvGRU (Ballas et al., 2015), we choose ConvGRU as our sequence model building block for its lower count of parameters and operations. Our ConvGRU layer follows the equations in Ballas et al. (2015), with all 2D convolutions replaced with 3D convolutions. We make it bi-directional by applying two independent ConvGRU layers, one in the positive and in the negative time direction, to our sample $\boldsymbol{x}_t$ and then combining their outputs with a simple, learned linear layer.

See Figure 2 for a visual illustration of our model architecture.

3.1. Physics-inspired Guidance

Diffusion guidance is a mechanism to transform unconditional generative diffusion models into conditional models by guiding the generative process by conditioning the score function (Dhariwal & Nichol, 2021). If we want to condition our generative model on another random variable $\boldsymbol{y}$, we get the updated reverse process (Kollovieh et al., 2023a)

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) + \sigma_t^2 \boldsymbol{s}, \sigma_t^2 \boldsymbol{I})$$
$$\boldsymbol{s} = \nabla_{\boldsymbol{x}_t} \log \mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}_t)$$

$$(3)$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is the learned denoised mean and $\boldsymbol{s}$ is the guiding score.

The Navier-Stokes equation describes the motion of viscous fluids and as such forms the foundation of CFD and nu-

Table 1: Evaluation results w.r.t. the TKE spectra distance and the regional distribution distance evaluated over sample sets generated from noise with 3 different random seeds.

Dimension	$1/2\,\sigma_{\mathrm{NS}}^{-2}$	$W_{2,\mathrm{TKE}}$	$W_{2,\mathcal{R}}$
3	–	4.04(8)	1.470(2)
4	–	4.57(8)	1.392(3)
4	10^{-12}	4.48(2)	1.326(2)
4	10^{-11}	4.47(2)	1.326(2)
4	10^{-10}	4.43(2)	1.327(2)
4	10^{-9}	4.25(2)	1.332(2)
4	10^{-8}	4.25(2)	1.351(2)

merical solvers like OpenFOAM. For a given density ρ and viscosity ν, the velocity field $\boldsymbol{u}$ and pressure field p have to fulfill

$$\frac{\partial \boldsymbol{u}}{\partial t} = \nu \nabla^2 \boldsymbol{u} - \frac{1}{\rho} \nabla p - (\boldsymbol{u} \cdot \nabla) \boldsymbol{u}. \tag{4}$$

Through guidance, we can use this equation to inject physical knowledge into the generative process as follows. We define $\boldsymbol{y}$ as the residual of Equation (4),

$$\boldsymbol{y} = -\frac{\partial \boldsymbol{u}}{\partial t} + \nu \nabla^2 \boldsymbol{u} - \frac{1}{\rho} \nabla p - (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} \tag{5}$$

and assume that $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\mathrm{NS}}^2 \boldsymbol{I})$. An exact solution would have $\boldsymbol{y} = \boldsymbol{0}$, but the variance $\sigma_{\mathrm{NS}}^2 \boldsymbol{I}$ lets us introduce some slack to account for the unavoidable discretization error inherent to any numerical simulation.

This lets us simplify the guiding score in Equation (3) to

$$\boldsymbol{s} = -\frac{1}{2\sigma_{\mathrm{NS}}^2} \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y}\|^2. \tag{6}$$

This form of guidance would not have been possible in the original approach from Lienen et al. (2024), because the computation of $\boldsymbol{y}$ requires taking gradients with respect to time, which are not available in the generation of independent snapshots. However, since we are sampling coherent sequences, we can approximate both spatial and temporal derivatives in Equation (5) with finite differences from the current sample $\boldsymbol{x}_t = (\boldsymbol{u}_t \| p_t)$.

4. Experiments

We trained two models to evaluate our approach. The first is a 4D model with the architecture shown in Figure 2 trained to generate simulations of length $k = 5$. To compare against the established approach of generating independent snapshots of the simulation, we trained a second model to generate sequences of length $k = 1$, i.e. independent snapshots. For this model, we replaced all `BiConvGRU` layers with identity functions. Both models are trained for 5 epochs

and we chose the checkpoint with the lowest $W_{2,\mathrm{TKE}}$ on a validation set for further evaluation. Our training, validation and test set split follows the setup by Lienen et al. (2024).

We evaluate the sample quality of our model with the $W_{2,\mathrm{TKE}}$ and $W_{2,\mathcal{R}}$ metrics proposed by Lienen et al. (2024). The former compares the turbulent kinetic energy (TKE) spectra of the samples and the data, measuring if the kinetic energy of the flow is distributed realistically over the spatial scales. The latter measures if the samples exhibit the correct velocity, vorticity and pressure distributions in the each region of the simulation domain, e.g. low velocities and high pressures in a closed off corner and high velocities in the correct direction at the flow-facing edges of an object.

To compute these metrics, we draw the same number of samples, 16 from each model. Since the metrics compare snapshots invidually, the 4D model would have $16k$ samples with some of them highly correlated compared to the 16 independent samples of the 3D model. To ensure that the metrics can be compared between the models, we select only the middle frame of each sampled sequence for evaluation for the 4D model.

Figure 3 shows the velocity and pressure fields from a simulation generated with our model. See Appendix A for additional samples from several test cases.

Our evaluation results in Table 1 show first that our extended 4D model generates samples of comparable quality to the 3D baseline model. While the 4D samples are weaker in terms of their TKE spectra, they have a lower regional distribution distance $W_{2,\mathcal{R}}$. This means that the 4D samples are distributing the TKE across the spatial scales less accurately than the 3D samples. We attribute this to the fact that generating the smallest vortices in a flow seems to be particularly difficult for the model, as we can see in the sample in Figure 3. However, $W_{2,\mathcal{R}}$ being lower in 4D means that the overall distribution of velocities and pressures is more accurate. Overall, these results show that our 4D model is a viable extension of the 3D baseline model.

We attribute our metrics in the 3D case being overall slightly worse than the ones reported by Lienen et al. (2024) to the fact that we needed to reduce the model capacity, i.e. number of parameters and latent channel dimensions, to ensure that the 4D model could be trained on an NVIDIA A100.

4.1. Navier-Stokes Guidance

We also evaluate the effect of our physics-informed guidance (Section 3.1) in Table 1. The results show that increasing the strength of the guidance by decreasing the assumed variance σ_{NS}^2 on the residual $\boldsymbol{y}$ (Equation (5)) improves the TKE spectra of the samples while keeping the quality of the overall velocity and pressure distribution constant. This shows that our guidance is a viable approach to inject
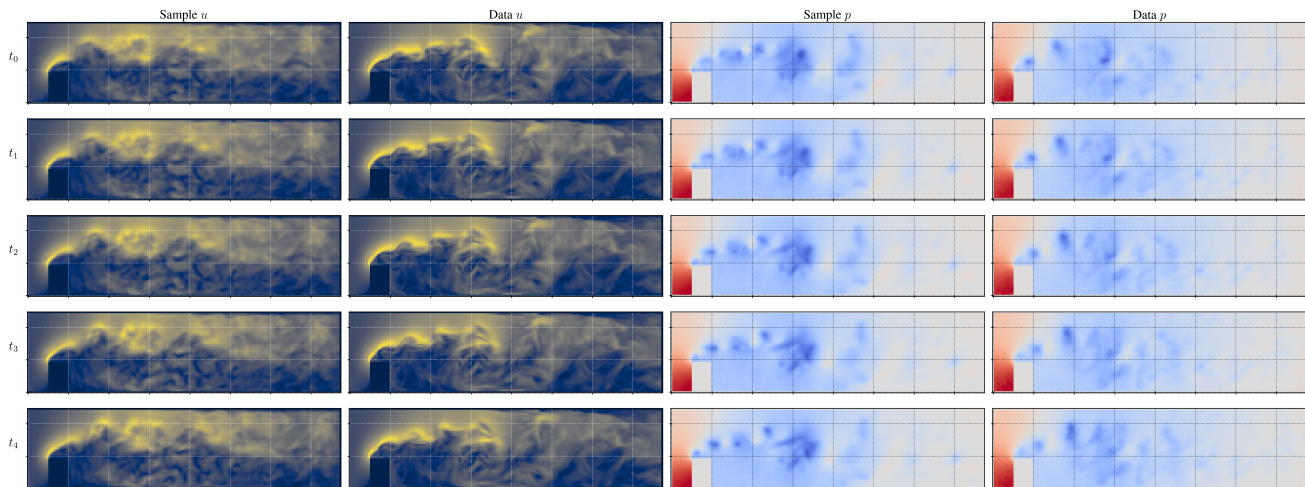
Figure 3: A generated sample for the `step-high` case compared to a ground-truth example. The dotted grid provides visual support to highlight the dynamics of the flow.

physical knowledge into the sampling process of diffusion models.

5. Limitations & Future Work

The main source of inaccuracy in our physics-informed guidance is the need to approximate the derivatives in Equation (5) with finite differences. First, there is the contribution to the inaccuracy of the finite difference approximation from the discretization of the simulation domain. However, further increasing the resolution of the simulation domain is challenging because the data is already challenging to handle at its current resolution. Second, the accuracy of the finite difference approximation on a gradient relies on the underlying function being smooth. However, the simulation data is inherently not smooth because the flow was not fully resolved, which is infeasible in any practical case.

A sophisticated approach to overcome the second source of inaccuracy might be to apply a Large-Eddy-Simulation (LES) technique to the residual in Equation (5). This is the same technique that was used with OpenFOAM to generate the dataset. In LES, one splits the dynamics into large and small scales and simulates the large scales, while applying a model for the dynamics at sub-grid resolution scale. A similar approach could enhance the effectiveness of our physics-informed guidance technique further.

6. Conclusion

We have extended the generative turbulence simulation approach by Lienen et al. (2024) to 4D and shown that our model can generate high-quality time-varying simulations of 3D turbulent flows by combining 3D U-Nets with bi-directional ConvGRU layers. Our Navier-Stokes guidance

mechanism lets us inject physical knowledge into the sampling procedure and improves the sample quality. We believe that our contributions are an important step to make neural networks viable surrogate models for 3D turbulent flows.

Acknowledgments

References

Ballas, N., Yao, L., Pal, C., and Courville, A. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

Chen, D., Gao, X., Xu, C., Chen, S., Fang, J., Wang, Z., and Wang, Z. Flowgan: A conditional generative adversarial network for flow prediction in various conditions. *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 315–322, 2020.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., and Wells, W. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Lecture Notes in Computer Science, pp. 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8. doi: 10.1007/978-3-319-46723-8_49.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs

on Image Synthesis. In *Neural Information Processing Systems*, 2021.

Galanti, B. and Tsinober, A. Is turbulence ergodic? *Physics Letters A*, 330(3):173–180, September 2004. ISSN 0375-9601. doi: 10.1016/j.physleta.2004.07.009.

Giannopoulos, M., Tsagkatakis, G., and Tsakalides, P. 4d u-nets for multi-temporal remote sensing data classification. *Remote Sensing*, 14(3), 2022. ISSN 2072-4292. doi: 10.3390/rs14030634.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *ArXiv*, abs/2204.03458, 2022.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

Kollovieh, M., Ansari, A. F., Bohlke-Schneider, M., Zschiegner, J., Wang, H., and Wang, Y. Predict, Refine, Synthesize: Self-Guiding Diffusion Models for Probabilistic Time Series Forecasting. In *Neural Information Processing Systems*. arXiv, 2023a. doi: 10.48550/arXiv. 2307.11494.

Kollovieh, M., Gosch, L., Scholten, Y., Lienen, M., and Günnemann, S. Assessing robustness via score-based adversarial image generation. *arXiv preprint arXiv:2310.04285*, 2023b.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Lienen, M. and Günnemann, S. Learning the Dynamics of Physical Systems from Sparse Observations with Finite Element Networks. In *International Conference on Learning Representations*, 2022.

Lienen, M., Lüdke, D., Hansen-Palmus, J., and Günnemann, S. From Zero to Turbulence: Generative Modeling for 3D Flow Simulation. In *International Conference on Learning Representations*, 2024.

Lu, L., Dao, M., Kumar, P., Ramamurty, U., Karniadakis, G. E., and Suresh, S. Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proceedings of the National Academy of Sciences*, 117(13):7052–7062, 2020.

Lüdke, D., Biloš, M., Shchur, O., Lienen, M., and Günnemann, S. Add and Thin: Diffusion for Temporal Point Processes. In *Neural Information Processing Systems*. arXiv, 2023. doi: 10.48550/arXiv.2311.01139.

Oldenburg, J., Borowski, F., Öner, A., Schmitz, K.-P., and Stiehm, M. Geometry aware physics informed neural network surrogate for solving navier–stokes equation (gap-inn). *Advanced Modeling and Simulation in Engineering Sciences*, 9, 2022.

Pope, S. B. Turbulent flows. *Measurement Science and Technology*, 12(11):2020–2021, 2001.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention*, Lecture Notes in Computer Science, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

A. Samples from the 4D model

Here, we depict samples generated by our model in 4 different scenarios. Black and white grids are added to aid the visibility of small changes in time. Exact depictions of the shapes that block the flow are available in the appendix of Lienen et al. (2024).
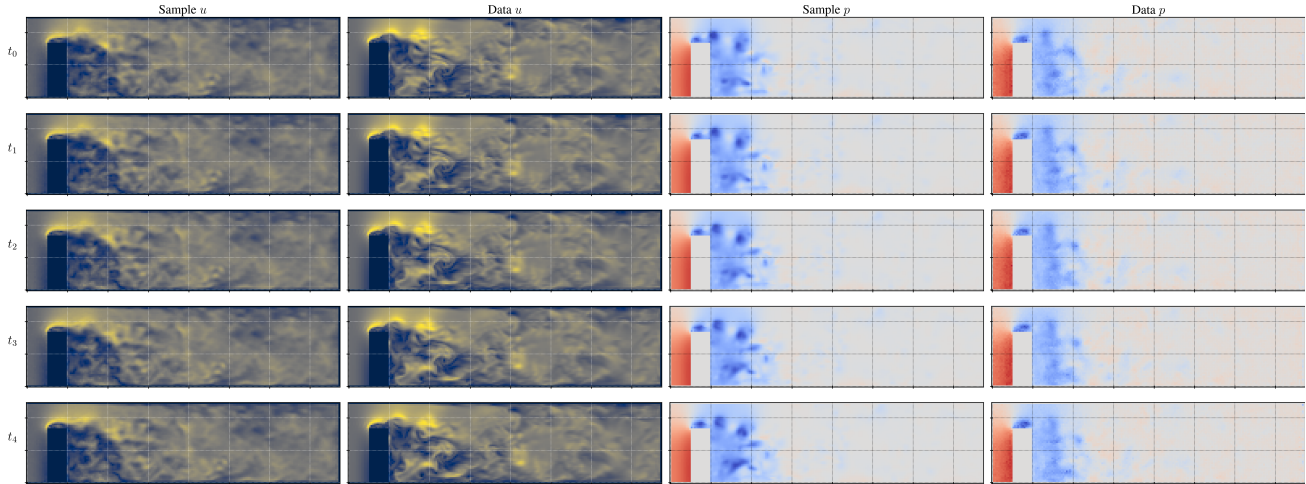

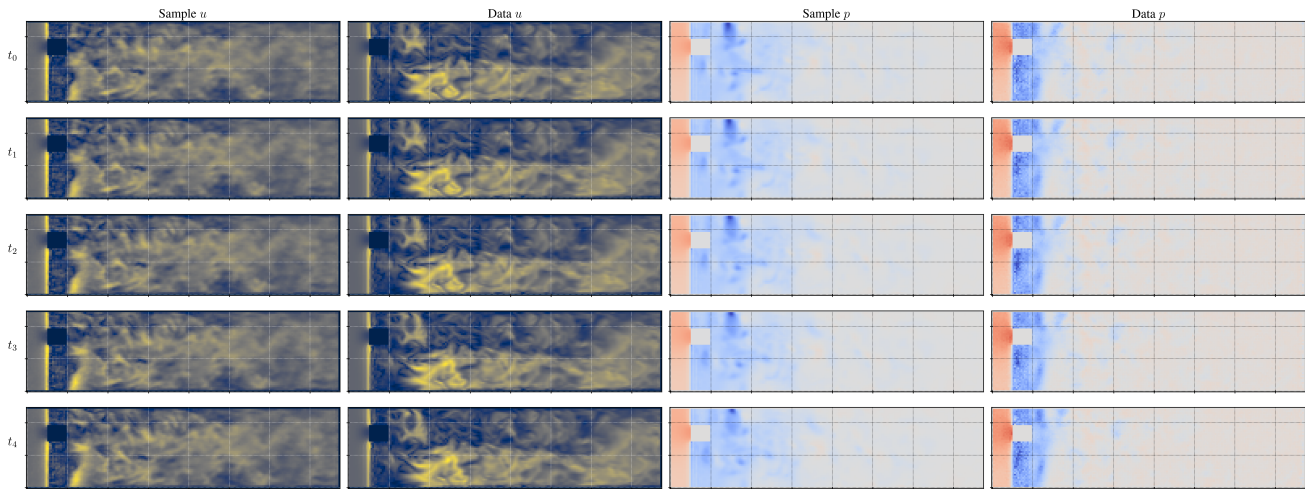Figure 4: A generated sample from case `wide-elbow`. `wide-elbow` is an "L" shape rotated 90 degrees clockwise.

Figure 5: A generated sample from case `cross-offset`. `cross-offset` is a cross but the lines intersect each other on the side rather than in the middle.
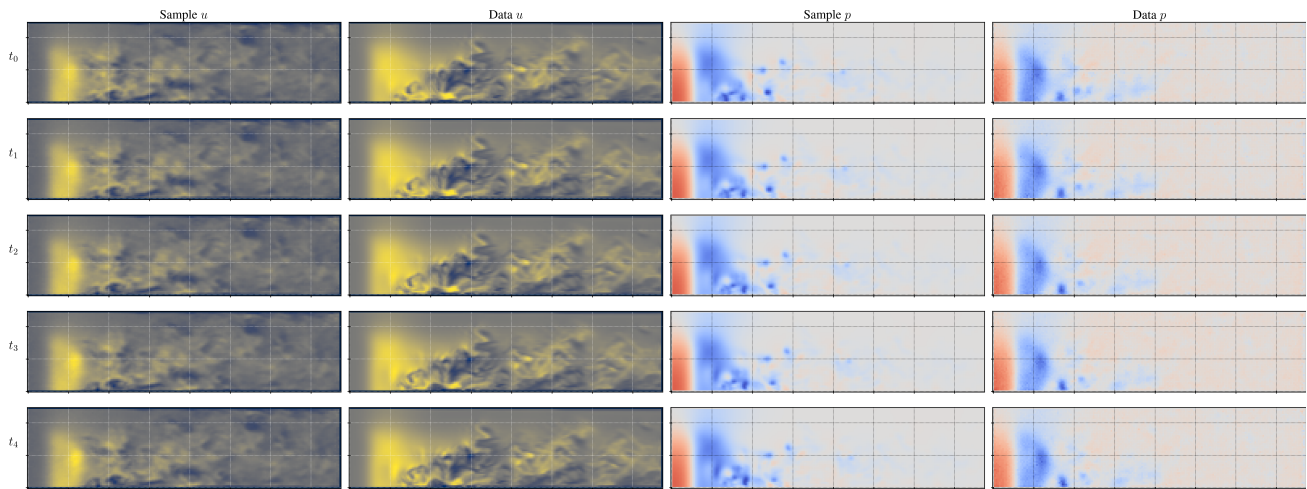
Figure 6: A generated sample from case `double-pillar`. `double-pillar` is two pillars standing side by side.
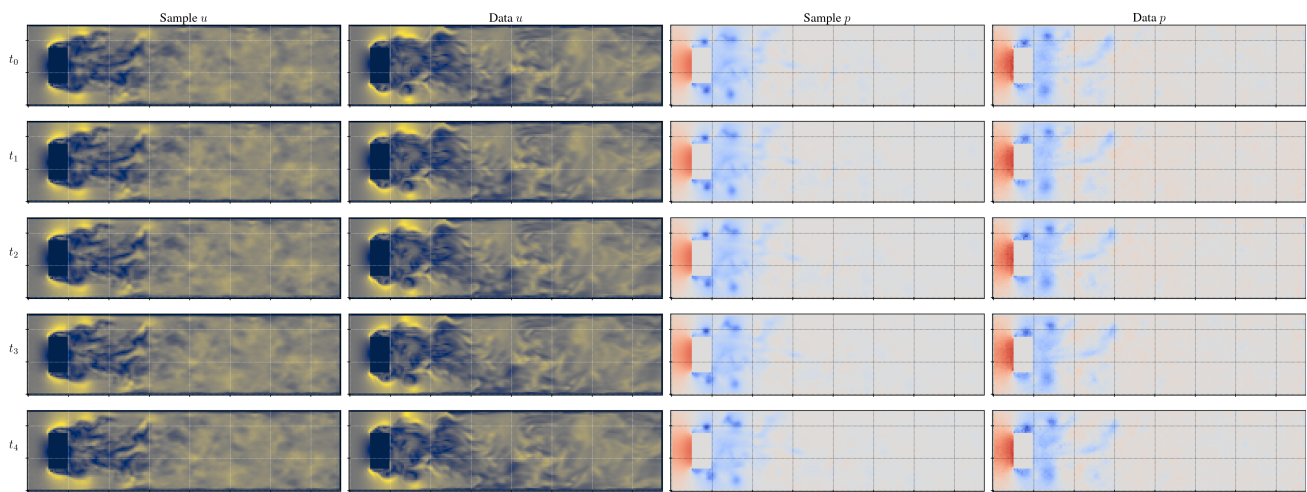

Figure 7: A generated sample from case `square-large`. `square-large` is a large square in the middle of the flow.