Data Analytics and Machine Learning Group
Department of Computer Science
Technical University of Munich

From Zero to Turbulence: Generative Modeling for 3D Flow Simulation

Marten Lienen, David Lüdke, Jan Hansen-Palmus, Stephan Günnemann

TL;DR

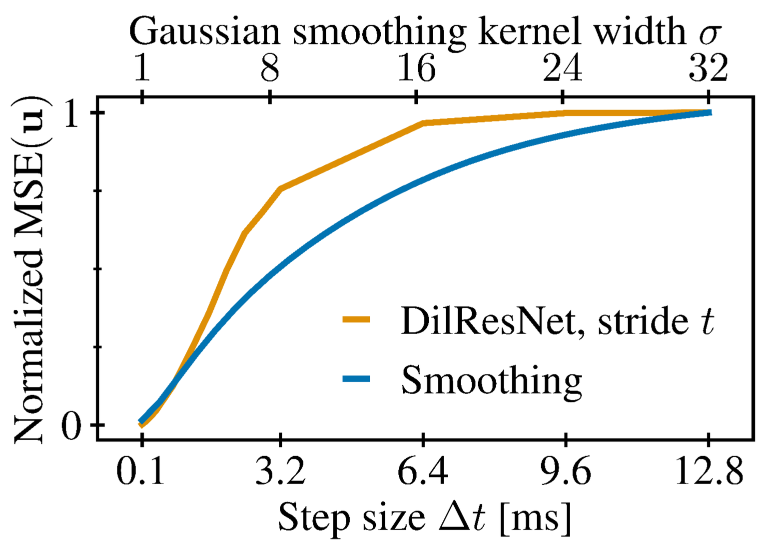
- Turbulence in 3D is more than "just one more dimension"
- Autoregressive models struggle to track intricate vortex structures through time
- Generative modeling lets us sample from the manifold of flow states directly, sidestepping the tracking problem

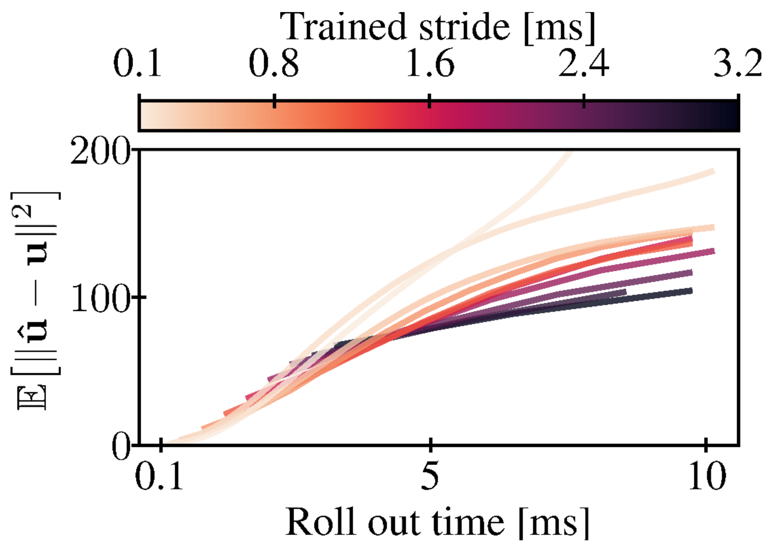
Turbulence in 2D and 3D

- 3D flows develop recursive, *fine-grained vortex structures* due to vortex stretching and strain self-amplification
- In 2D, energy cascade inverts due to vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u} = \mathbf{0}$, creating *homogeneous, long-lived structures*
- Smaller structures have shorter lifetimes but still influence larger ones through backscattering

The Autoregressive Dilemma

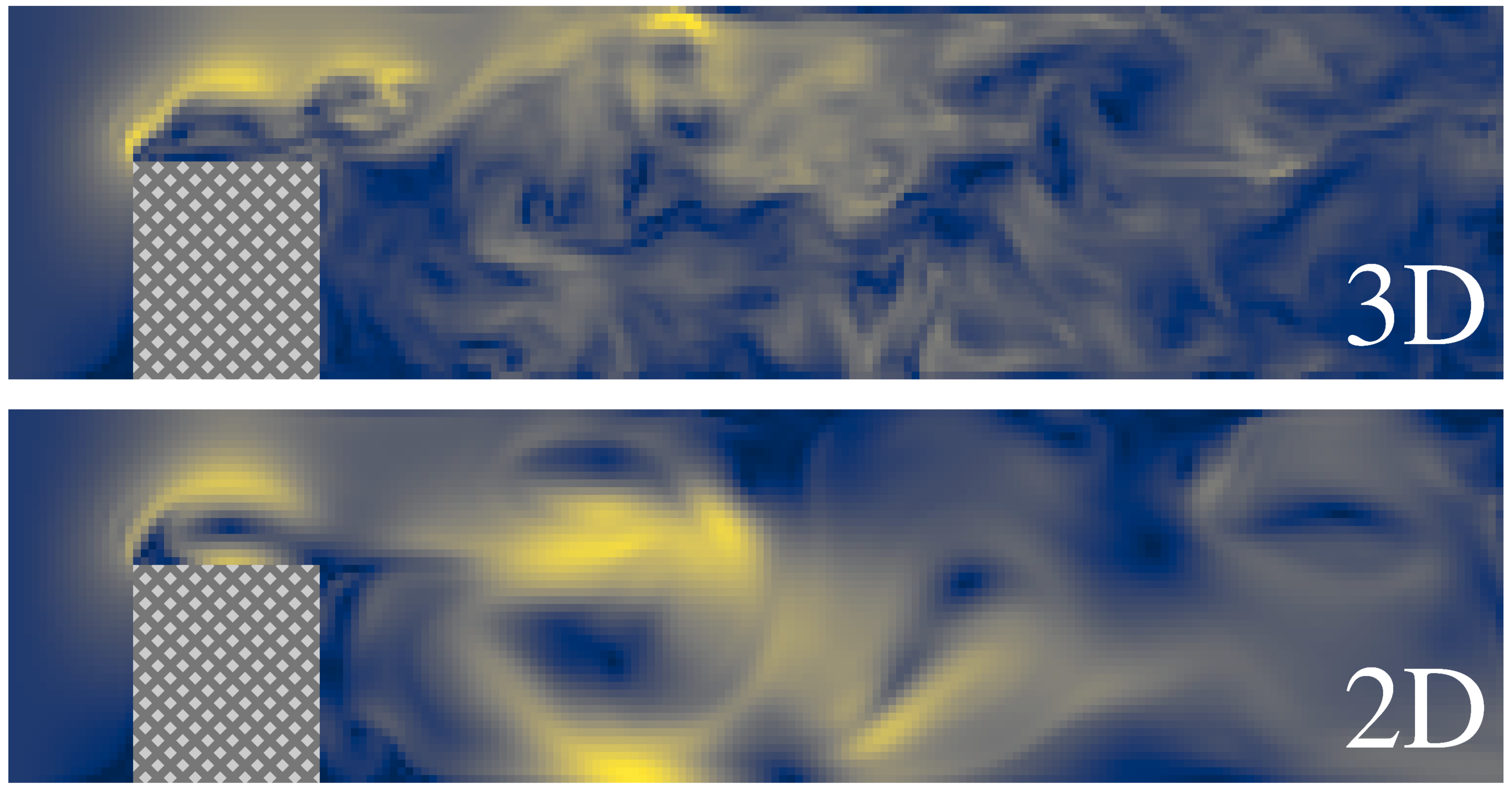
- Autoregressive models outpace numerical solvers by taking larger steps
- But: large time steps smooth out small-scale structures





- Increasing time step of autoregressive model scales prediction error similar to smoothing larger and larger features from the target
- Small-scale structures are essential to turbulence and influence the overall trajectory of the flow

large time steps for *performance* vs. small time steps to *retain turbulence*



Generative Turbulence Simulation

- Simulations start from non-turbulent initial state $\boldsymbol{X}^{(0)}$ and boundary conditions $\boldsymbol{B}$ and reach turbulence after time t_{turb}
- 3D turbulent flows can be modeled as stochastic processes $\mathrm{p}\left(\boldsymbol{X}^{(t)} \mid \boldsymbol{X}^{(0)}, \boldsymbol{B}\right)$ because of their chaotic nature
- Turbulence flows are ergodic, i.e. flow state does not depend on $\boldsymbol{X}^{(0)}$

$$\mathrm{p}\left(\boldsymbol{X}^{(t)} \mid \boldsymbol{X}^{(0)}, \boldsymbol{B}, t > t_{\text{turb}}\right) = \mathrm{p}\left(\boldsymbol{X}^{(t)} \mid \boldsymbol{B}, t > t_{\text{turb}}\right)$$

- So, we can **simulate by sampling from a generative model**

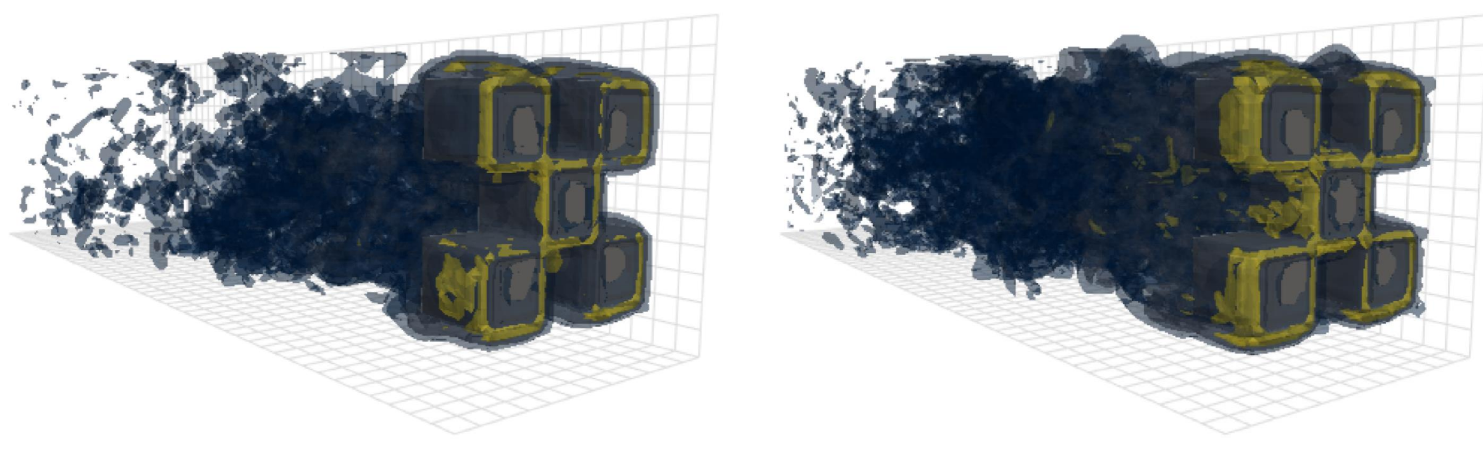
$$\mathrm{p}_{\boldsymbol{\theta}}\left(\boldsymbol{X}^{(t)} \mid \boldsymbol{B}\right) = \mathrm{p}\left(\boldsymbol{X}^{(t)} \mid \boldsymbol{B}, t > t_{\text{turb}}\right)$$

- Generative model *does not require an initial state* $\boldsymbol{X}^{(0)}$

Our Model

- *TurbDiff* is based on denoising diffusion probabilistic models (DDPM)
- Iteratively transforms Gaussian noise into a sample from simulation
- Conditions sampling process on $\boldsymbol{B}$ by fixing boundary cell values to true posterior

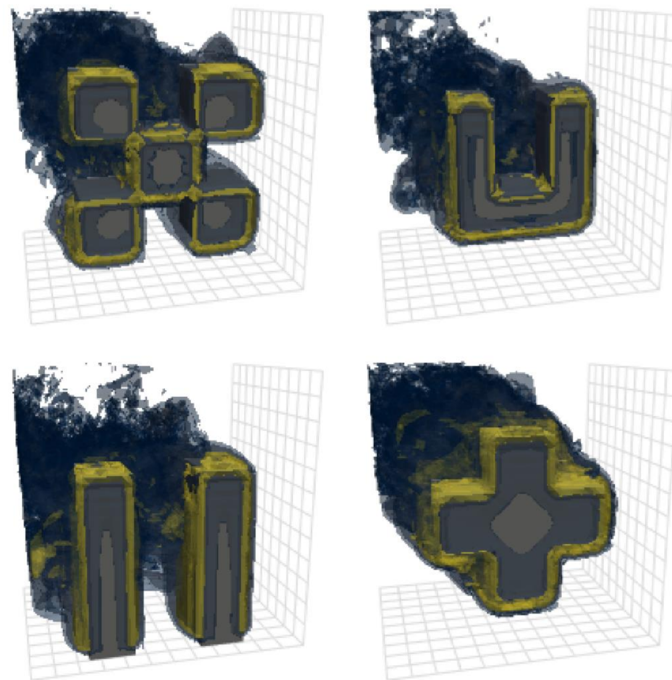
$$\mathrm{p}(\boldsymbol{X}_{n-1} \mid \boldsymbol{X}_n)_{\mathbf{i}} \sim \begin{cases} \mathrm{p}_{\theta}(\boldsymbol{X}_{n-1} \mid \boldsymbol{X}_0, \boldsymbol{X}_n, \boldsymbol{B})_{\mathbf{i}} & \text{if cell } \mathbf{i} \text{ is interior} \\ \mathrm{q}(\boldsymbol{X}_{n-1} \mid \boldsymbol{X}_0, \boldsymbol{X}_n)_{\mathbf{i}} & \text{otherwise} \end{cases}$$



Generated sample $\|\boldsymbol{\omega}\|_2$ Data

Dataset

- 45 shapes in a 3D flow
- 0.4x0.1x0.1m
- 20m/s flow velocity
- 192x48x48 cells
- 0.5s at 0.1ms steps
- 5000 steps
- OpenFOAM in LES mode
- 2TB of postprocessed data
- Horizontal flow distance per step roughly equal to 1 cell width (2mm)



Metrics

- Wasserstein distance W_2 between generated sample sets and subsamples of dataset
- One global and one local distance between samples

Turbulent Kinetic Energy (TKE)

- Measure the distance between the log TKE spectra

$$d_{\text{TKE}}(\boldsymbol{X}, \boldsymbol{X}') = \left\| \log E_{\boldsymbol{X}} - \log E_{\boldsymbol{X}'} \right\|_2$$

- Turbulent TKE spectra follow Kolmogorov's 5/3 law

Regional Distributions

- Divide domain into regions R of coherent behavior of ~500 cells via k-means clustering based on marginal velocity distribution
- In each region, compare W_2 distance between distributions of $\mathbf{v}_R \coloneqq \boldsymbol{u} \mid\mid \boldsymbol{\omega} \mid\mid p$

$$d_{\mathrm{R}}(\boldsymbol{X}, \boldsymbol{X}') = \left(\sum_R \frac{|R|}{\sum_{R'} |R'|} W_2^2(\boldsymbol{v}_{R,\boldsymbol{X}}, \boldsymbol{v}_{R,\boldsymbol{X}'}) \right)^{1/2}$$

- Balances distribution of $\boldsymbol{v}_R$ with their location in the domain

Results

- TurbDiff outperforms full surrogate baselines (-init)

	$W_{2,\text{TKE}}$	$W_{2,\mathcal{R}}$	Runtime [s]
TF-Net-init*	-	-	1.834
TF-Net-22*	189	493	602 + 0.23
DilResNet-init	60 ± 47	4.6×10^8	12.82
DilResNet-22	2.15 ± 0.06	1.240 ± 0.001	602 + 1.58
TurbDiff (ours)	3.9 ± 0.4	1.38 ± 0.04	20.63