Generative Modelling for Fluid Simulations

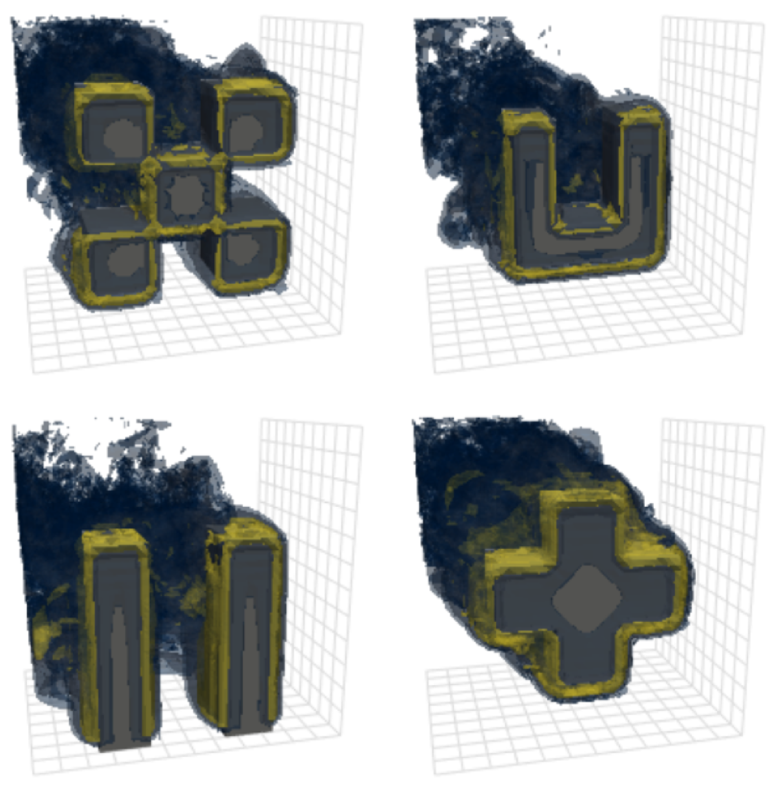
Artem Alekberov, Diogo Soares, Vuk Bibic, Marten Lienen, Stephan Günnemann

DAML Group, Technical University of Munich

Technical University of Munich

Motivation & Problem

- Significant amount of compute is spent everyday to model fluid dynamics in turbulent states
- Turbulent flows exhibit non-Gaussian velocity and pressure distributions, with values extending up to 10 standard deviations from the mean [1]
- Heavy Tailed Distributions are hard to model with generative models that leverage Gaussian noise [2]



The goal is to solve the following generative problem, where $\mathcal{B}$ denotes the boundary conditions:

$$p_\theta(X \mid \mathcal{B}) \approx p_{t_{\text{turb}}}(X \mid \mathcal{B}) \quad (1)$$

Ideas

- Use non-Gaussian noising procedures with heavier tails
- Map the original distribution to a distribution with smaller tails, by performing dimensionality reduction

Metrics

For our results we focus on two Wasserstein-2 distances:

- **Turbulent Kinetic Energy (TKE)** that captures the global patterns in the flow velocity by measuring the energy contained in various spatial scales [1]
- **Distributional Distance (DD)** that incorporates both velocity and pressure variations across space by focusing on regions with similar distributional characteristics [1]

Non Gaussian Denoising Diffusion

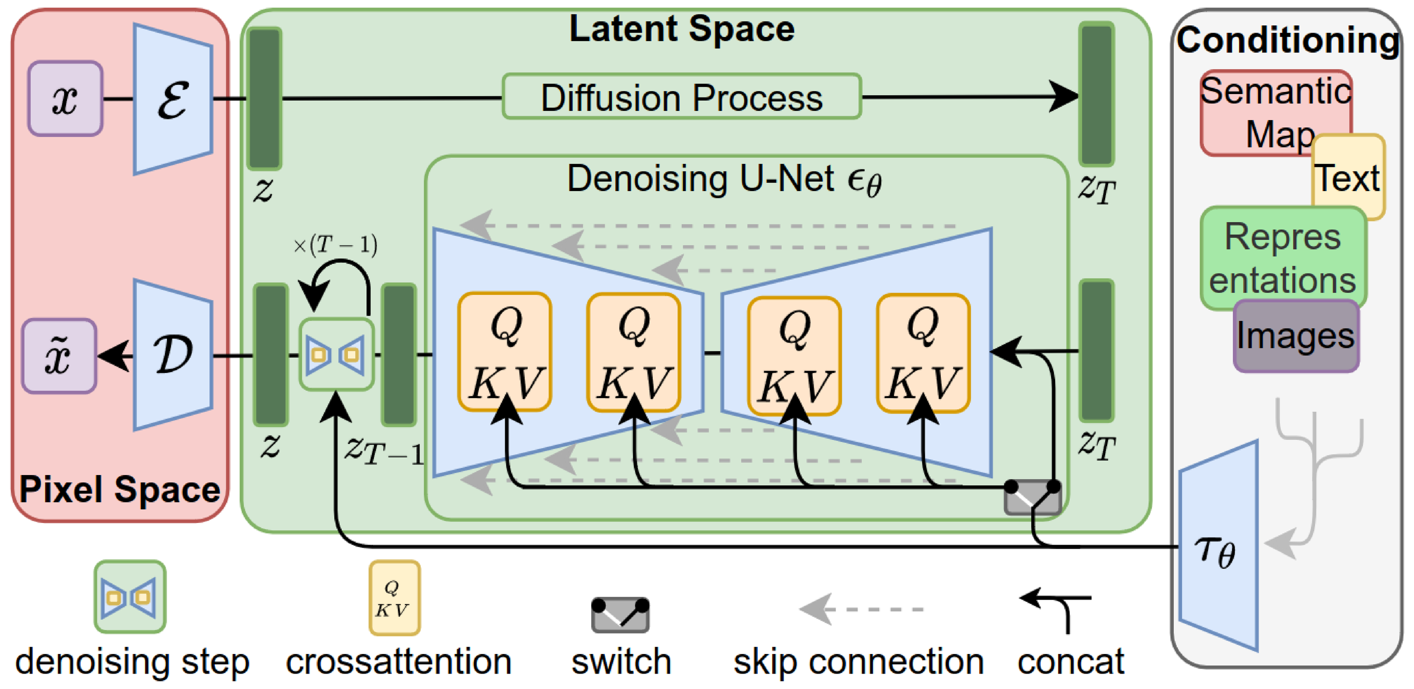
- Replacing the noising process is complex because of Gaussian properties like the reparameterization trick and stability
- Mixture of Gaussians and Gamma distributions are stable and can fit the diffusion framework with minimal technical adjustments [3]

Models	TKE ↓	DD ↓
Gaussian	3.87	**1.19**
Mix of Gaussians	**3.35**	1.20
Gamma	3.47	1.20

Latent Diffusion

Steps

1. Encode all datapoints (Training)
2. Learn the distribution of latent representations with denoising diffusion (Training)
3. Sample from the aforementioned distribution (Inference)
4. Decode the sample (Inference)



Autoencoder

Architecture

- Encoder-decoder with latent structure similar to input structure
- Each block consists of 3D convolutions + interpolation layers
- A transformer layer was applied to the latent space
- Conditioning added to both encoder and decoder

Loss

- Vital to accurately reconstruct the input data
- Patch-wise adversarial loss leads to more realistic samples [4]

$$Q^* = \arg\min_{AE,Z}\max_{D} \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\mathcal{L}_{\text{Rec}}(AE, Z) + \lambda_{\text{Adv}}\mathcal{L}_{\text{Adv}}(AE, Z, D)\right] \quad (2)$$

Autoencoder Results

Let $f \in \{2, 4, 8\}$ denote the downsampling factor, i.e the ratio of input to latent space dimensions per spatial dimension.

f	TKE ↓	DD ↓	Rec. Loss ↓
2	**1.68**	**1.10**	$\mathbf{6.1 \times 10^{-5}}$
4	1.86	1.17	5.7×10^{-4}
8	2.39	1.20	1.5×10^{-3}

Tail Heaviness

- The number of standard deviations from the mean to the 0.01 and 99.99 percentiles indicates the heaviness of the distribution's left and right tails, respectively
- We define **Mean to Percentile** as $\frac{P_q - \mu}{\sigma}$ where q is either 0.01 or 99.99

	Mean to Percentile (σ)	
Distribution	0.01	99.99
gaussian	-3.72	3.72
laplace	-8.52	8.52
original-data	-0.89	7.12
embed-f2	-1.86	7.96
embed-f4	-1.82	4.83
embed-f8	-1.58	4.05

- To study the distribution of our embeddings, we transform the tensor by $\|\text{embed} - \mathbf{mean}(\text{embed})\|_2$

Latent Diffusion Results

- Large downsampling factors lead to higher compression error
- Small downsampling factors lead to harder denoising
- Latent Diffusion outperforms standard Diffusion

Models	TKE ↓	DD ↓
DDPM	3.87	**1.19**
LD-f2	3.73	1.21
LD-f4	**3.47**	1.21
LD-f8	4.90	1.41

Generated Sample

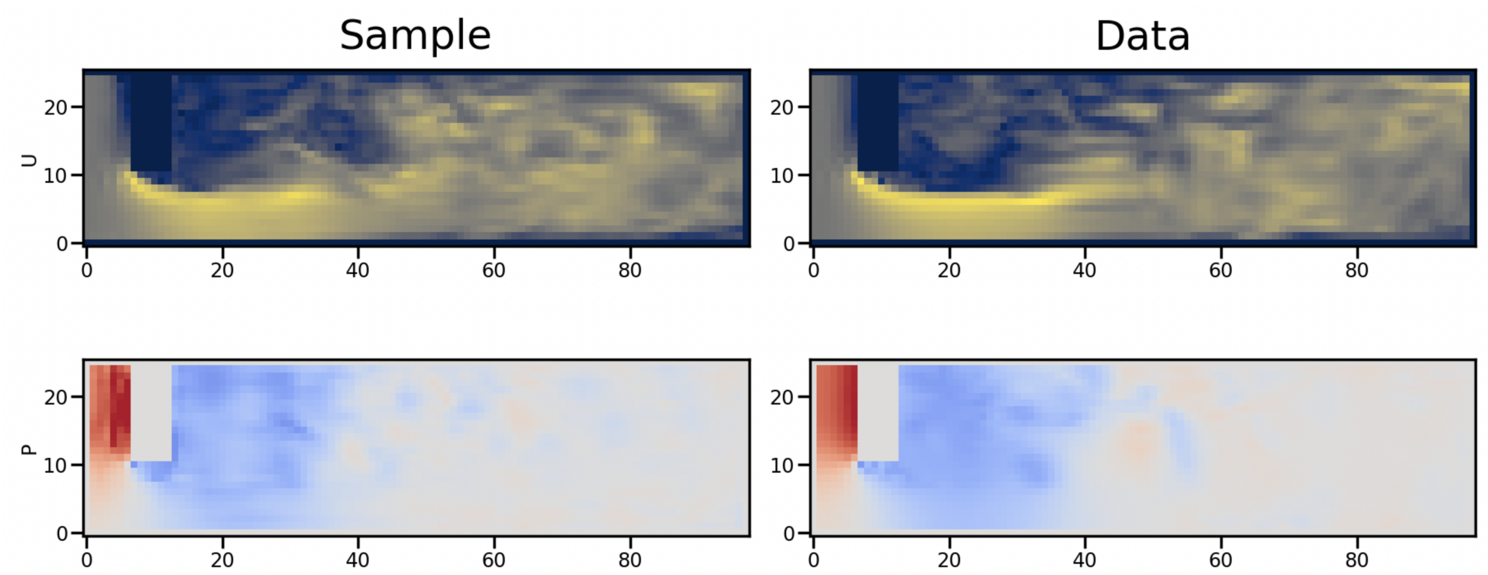


Figure 1. z-slice of a *wide elbow* geometry by LD-f4 model

References

Latent Diffusion scheme was presented in [5].

1. Lienen, M., Lüdke, D., Hansen-Palmus, J. & Günnemann, S. *From Zero to Turbulence: Generative Modeling for 3D Flow Simulation.* 2023.
2. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. (2020).
3. Nachmani, E., Roman, R. S. & Wolf, L. Non gaussian denoising diffusion models. (2021).
4. Esser, P., Rombach, R. & Ommer, B. *Taming transformers for high-resolution image synthesis.* in (2021).
5. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-resolution image synthesis with latent diffusion models.* in (2022).