Diffusion Models

Stochastic Differential Equations

Variational Autoencoders (VAE)

Stable Diffusion

Summary

Plan for This Week

Progress Report

Baris Turan

Chair of Data-Driven Fluid Dynamics (ITLR-DDSim)
Institut für Thermodynamik der Luft- und Raumfahrt
Universität Stuttgart

20 May 2025

Outline I

1. Diffusion Models
2. Stochastic Differential Equations
3. Variational Autoencoders (VAE)
4. Stable Diffusion
5. Summary
6. Plan for This Week

Denoising Diffusion Probabilistic Models

- Forward Process:
  - Starting from input $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, successively add i.i.d. Gaussian noise according to:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_t, \ \ \epsilon_t \sim \mathcal{N}(0, \mathbf{I})$$

for $t = 1, 2, \ldots, T$

  - Train neural network (usually UNet) to predict ϵ_t at a given timestep t given noisy image $\mathbf{x}_t$.
  - Minimize

$$L = \left\|\epsilon - \epsilon_\theta(x_t, t)\right\|^2$$

Denoising Diffusion Probabilistic Models

- Backward Process:
  - Use the trained neural network to estimate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$
  - Starting from $x_T \sim \mathcal{N}(0, \mathbf{I})$, successively denoise

  the image as

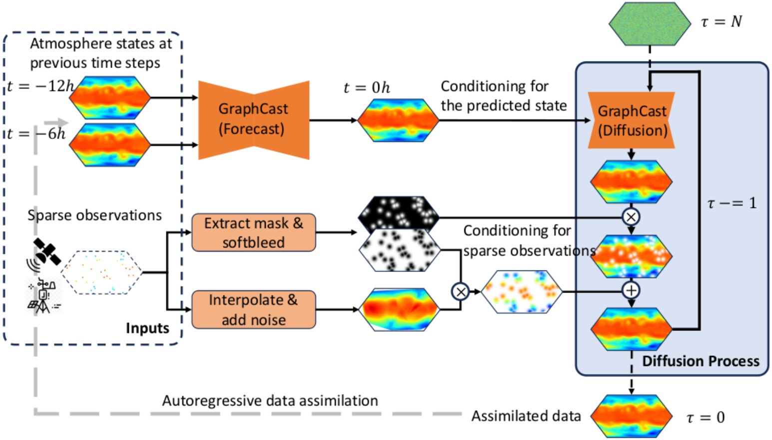
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\right) + \sqrt{\beta_t}\epsilon,\ \ \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

Conditional vs. Unconditional Models

- So far, we have been trying to estimate $q(\mathbf{x}_0)$
- For real life applications, we want $q(\mathbf{x}_0|y)$
- In image generation, y can be text, semantic map, a different image etc.
- For flow problems:
  - Boundary conditions
  - Sparse measurements
  - Information from governing equations
  - ...

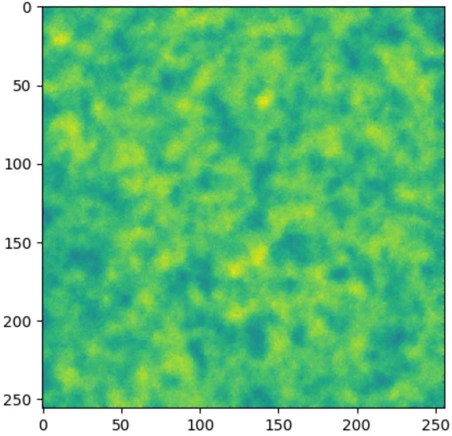
Example: DiffDA

- Weather-scale data assimilation framework
- Conditioned on sparse observations and predictions of a forecast model
- Conditioning for sparse observations implemented by soft-masking



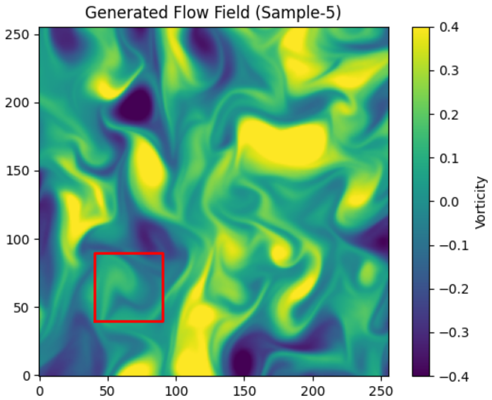
Übung on Unconditional Diffusion Model

- 2D homogeneous isotropic turbulence on 256x256 grid.
- UNet architecture
- Trained it on Google Colab for around 300 epochs



Übung on Conditional Diffusion Model

- Uses measurements in a random 50x50 region as condition.
- Condition, timestep and noisy data are concatenated.
- I used the pretrained model this time.



Outline I

1. Diffusion Models
2. Stochastic Differential Equations
3. Variational Autoencoders (VAE)
4. Stable Diffusion
5. Summary
6. Plan for This Week

SDEs

- Describe the time evolution of stochastic processes
- General form is given by

$$dx_t = \underbrace{\mu(\mathbf{x}_t, t)dt}_{\text{Deterministic Part}} + \underbrace{\sigma(\mathbf{x}_t, t)d\mathbf{w}_t}_{\text{Probabilistic Part}} .$$

- $\mu(\mathbf{x}_t, t)dt$: drift term
- $\sigma(\mathbf{x}_t, t)$: diffusion term
- Equivalent to PDE for the probability distribution (Fokker-Planck-Kolmogorov Equation):

$$\frac{\partial}{\partial t}p(\mathbf{x}, t) = -\frac{\partial}{\partial x}\left[\mu(\mathbf{x}, t)p(\mathbf{x}, t)\right] + \frac{\partial^2}{\partial \mathbf{x}^2}\left[\frac{1}{2}\sigma\sigma^T p(\mathbf{x}, t)\right]$$

Diffusion Models as SDEs

- SDE for forward diffusion:

$$d\mathbf{x} = -\frac{1}{2}\beta_t \mathbf{x} dt + \sqrt{\beta_t} d\mathbf{w}$$

- SDE for backward diffusion:

$$d\mathbf{x} = \left[-\frac{\beta_t}{2} - \beta_t \nabla_{\mathbf{x}} \log p(\mathbf{x})\right] + \sqrt{\beta_t} d\mathbf{w}$$

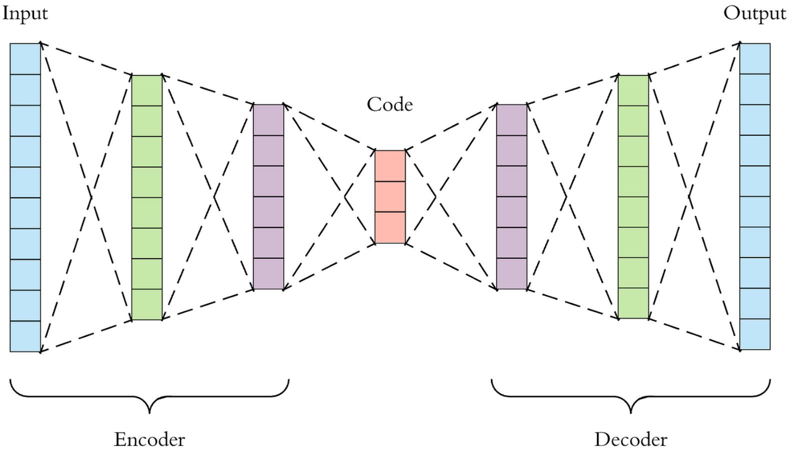
- $s = \log p(\mathbf{x})$ is the score function.
- Score based diffusion models estimate the score function.
- The SDE is then solved numerically using Euler-Maruyama integration

Outline I

1. Diffusion Models
2. Stochastic Differential Equations
3. Variational Autoencoders (VAE)
4. Stable Diffusion
5. Summary
6. Plan for This Week

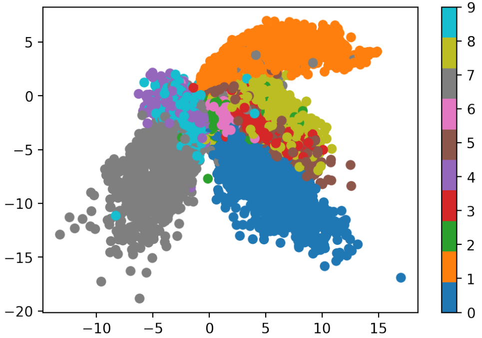
Autoencoder

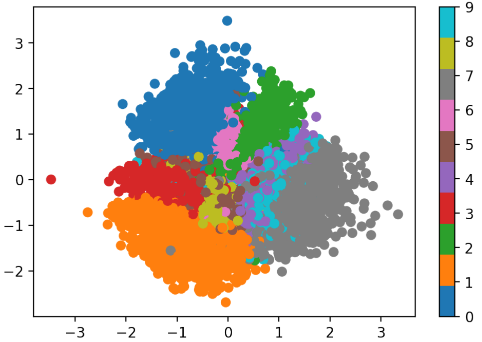
- A network aiming to reconstruct the input
  - Enconder: Compresses the input to the latent space
  - Decoder: Reconstructs input from the latent space representation.
- Can be used for feature extraction
- Decoder can potentially act as a generative model.



VAE

- Encoder maps input $\mathbf{x}$ to a probability distribution $p(x|\mathbf{z})$
- Decoder samples from this distribution and reconstructs the input
- In addition to the usual L1 or L2 reconstruction loss, the loss function includes a KL divergenece term to make the distribution similar to $\mathcal{N}(0, \mathbf{I})$
- Latent space is more centered and regularized, so better for generative tasks than vanilla AE.



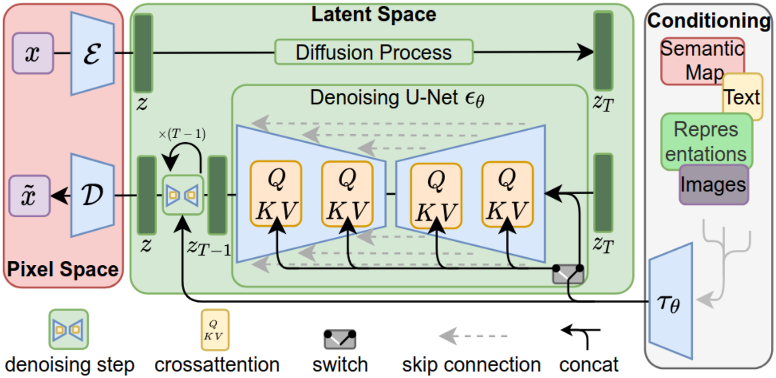


Outline I

1. Diffusion Models
2. Stochastic Differential Equations
3. Variational Autoencoders (VAE)
4. Stable Diffusion
5. Summary
6. Plan for This Week

Stable Diffusion

- Uses a VAE to compress the data into latent space
- Diffusion is carried out in latent space to reduce computational costs
- Uses attention for conditionin.



Outline I

1. Diffusion Models

2. Stochastic Differential Equations

3. Variational Autoencoders (VAE)

4. Stable Diffusion

5. Summary

6. Plan for This Week

Summary

- I have read about diffusion models, SDEs, VAEs
- I have done the Übungs on diffusion models

Outline I

1. Diffusion Models
2. Stochastic Differential Equations
3. Variational Autoencoders (VAE)
4. Stable Diffusion
5. Summary
6. Plan for This Week

Plan for This Week

- Read Hao's windfarm proposal
- Work on conditioning based on energy spectrum, two-point correlation
- Help Fabian with the urban heat island problem