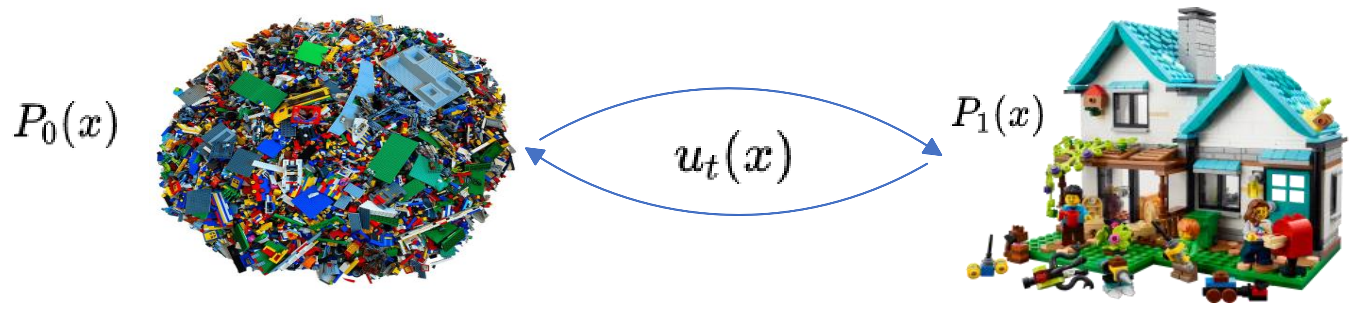
Optimal transport bigger, better but what about diffusion?

Students: Aksel Søren Beltoft (s194126), Christina Naja Tvilling Jensen (194123), Eline Dorothea Siegumfeldt (s183540) and Just Peter Taudorf Lorensen (194140)
Supervisor: Beatrix Miranda Ginn Nielsen (bmgi@dtu.dk)

What is flow matching?

Think of flow matching as giving a child a box of Lego blocks (representing, e.g., Gaussian noise) which are initially scattered randomly.



Flow matching is like creating an instruction manual that teaches a child how to assemble these blocks step-by-step into a specific shape, like a car, a castle or any desired target. The method plots a clear course from the jumble of blocks to the finished structure. We're essentially creating a 'manual' that describes how to transform this starting pattern, whatever it may be, into another specific pattern or distribution.

Given a target probability distribution path *pt(x)* and corresponding vector field *ut(x)* which generates *pt(x)*, the objective for flow matching can be defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \| v_t(x) - u_t(x) \|^2$$

Where θ denotes the learnable parameters of the CNF vector field *vt(x).*

Optimal transport and Diffusion

Optimal transport: The idea with OT is to find a function, T, that are able to transform one probability distribution into another probability distribution at the lowest possible effort. The lowest possible effort can also be calculated with the Wasserstein distance.

$$W(q_0, q_1)_2^2 = \inf_{\pi \in \Pi} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x,y)^2 d\pi(x,y)$$

Diffusion model:

Forward process:

The idea with the forward version of diffusion is to start with data points and slowly add noise until it approximates pure noise.

$$q(x_0, \ldots, x_T) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

Backward process:

The idea with the backward version of distribution is to turn a crappy distribution slightly better and then with a large amount of iterations get the ideal outcome we were looking for, which is similar to the distribution from the target.

General: $$p(x_t \mid x_{t-1}) = N(\mu(x,t;\theta), \beta(t))$$

Variance Preserving: $$p_t(x \mid x_1) = \mathcal{N}(x \mid \alpha_{1-t} x_1, (1-\alpha_{1-t}^2) I), \text{ where } \alpha_t = e^{-\frac{1}{2}T(t)}, T(t) = \int_0^t \beta(s) ds$$

Model objective/ loss function

Flow matching objective measures the discrepancy between the conditional target distribution *pt(x)* and the generated conditional distribution *vt(x)* by computing the squared norm of their difference.

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \left\| v_t(x) - u_t(x|x_1) \right\|^2$$

This loss function extends the Flow Matching Objective to a conditional setting, where the generated samples depend on conditioning variables *x1*.

Optimal transport objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\| v_t(\psi_t(x_0)) - \left(x_1 - (1-\sigma_{\min})x_0\right) \right\|^2$$

Where the conditional flow can be defined as:

$$\psi_t(x) = (1-(1-\sigma_{\min})t)x + tx_1$$

As it can be seen OT a more natural choice for the conditional propability path by defining the mean and std changing linearly in time.

$$\mu_t(x) = tx_1, \text{ and } \sigma_t(x) = 1-(1-\sigma_{\min})t$$

OT Samples from probability path trained from CIFAR-10

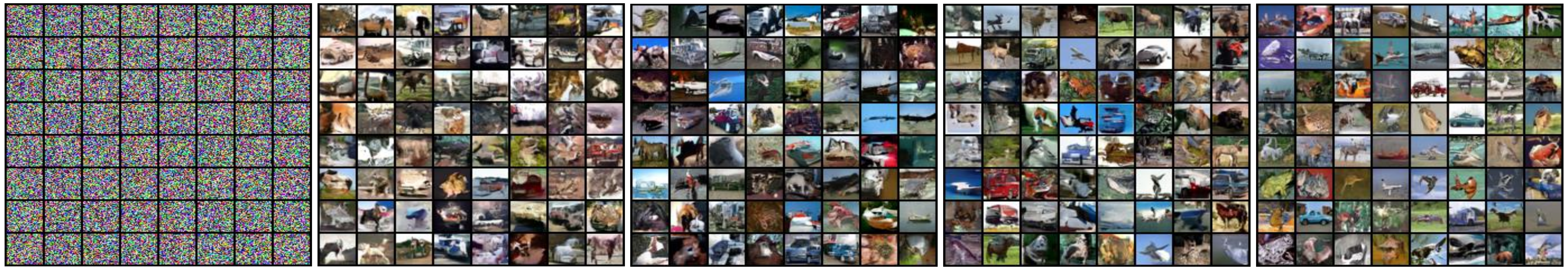
FID: 47.270

CPU running time: 4h 6m

VP Diffusion Samples from probability path trained from CIFAR-10

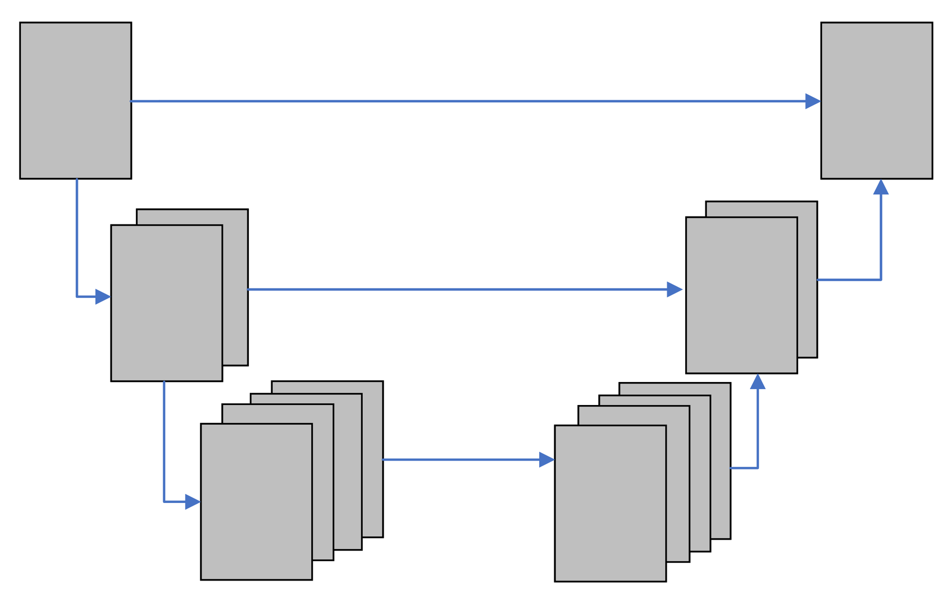
FID: 85.143

CPU running time: 4h 19m

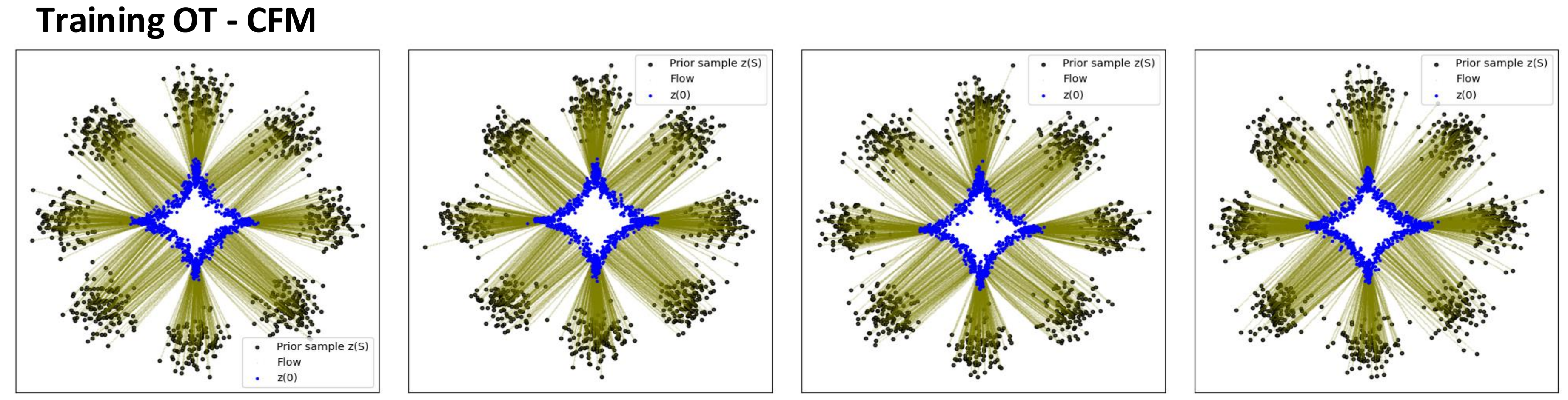


U-NET

U-Net model is a type of neural network architecture commonly used for image segmentation tasks. It is called U-Net, since it has a u shaped architecture. U-Net models consist of a contracting path and an expansive path. Where the contracting path is a convolution network that consist of a repeated application of convolution followed by a ReLu and a max pooling operation. During this step, the spatial information is reduced while increasing the feature information. The expansive path is responsible for restoring the spatial resolution of the image while combining it with features learned from the contracting path, this is achieved through a sequence of up-convolutions. To preserve the details during the upsampling process, U-Net uses skip connections. Which skip one or more layers in the contracting path and concatenate the feature maps with the corresponding layers in the expansive path to preserve the details.
The model used for this poster/paper is 4 steps deep with the following multiplication (1,2,4,8)



Training OT - CFM



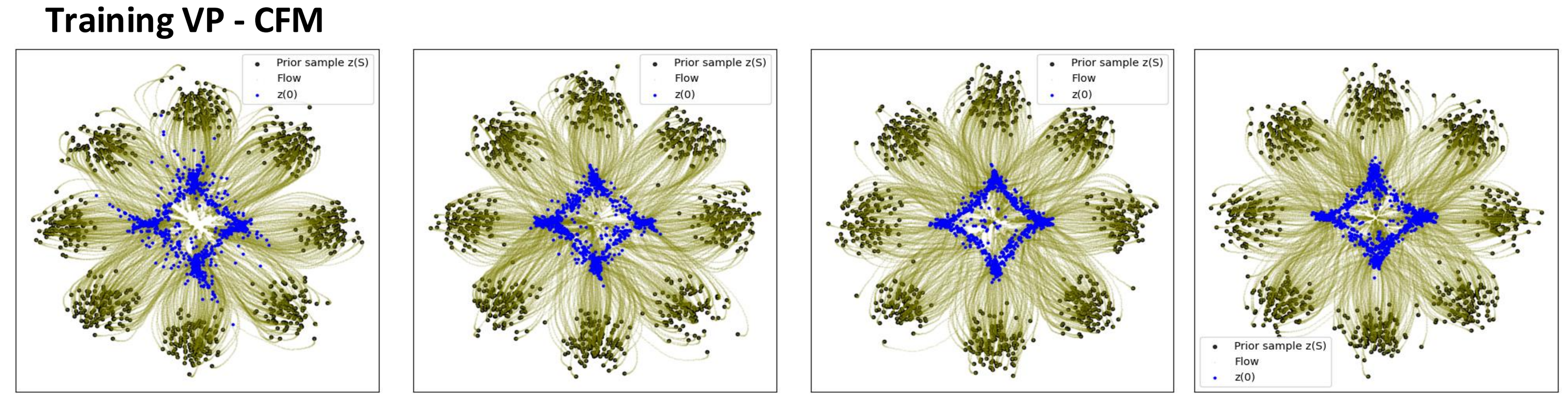
loss: 0.104 time: 146 s
CPU time total: 59 min

loss: 0.096 time: 133 s

loss: 0.108 time: 132 s

loss: 0.089 time: 133 s

Training VP - CFM

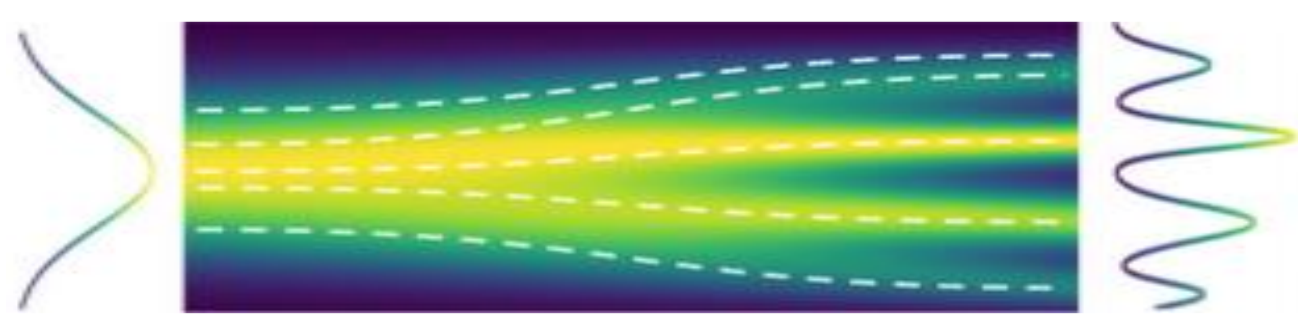


loss: 7.7, time: 33 s
CPU time total: 13 min

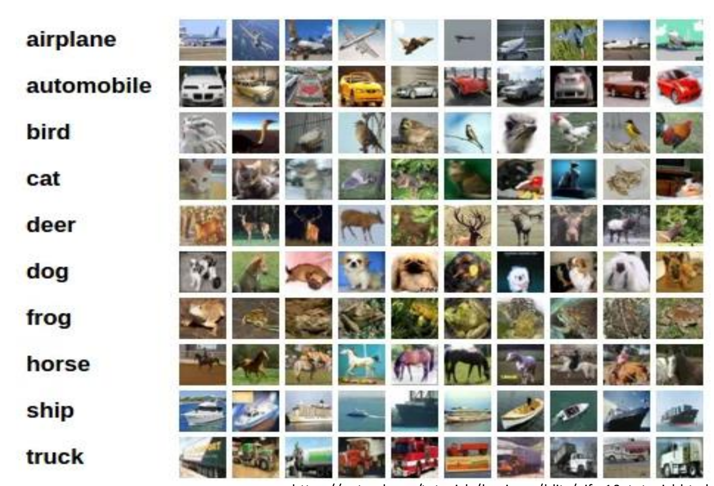
loss: 7.0 time: 36 s

loss: 7.5 time: 35 s

loss: 6.35 time: 37 s



OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport
Derek Onken, Samy Wu Fung, Xingjian Li, Lars Ruthotto



https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html