# Diffusion models

Exercise 2.1

Advanced Deep Learning in Computer Vision

February 23, 2026

In this exercise, you are asked to build a diffusion model that generates $16 \times 16$ sprites:

Figure 1: Example sprites

**Your joint pdf report for tasks 2.1, 2.2 and 2.3 should include the tasks below:**

1. Calculate $\beta_t$, $\alpha_t$, and $\bar{\alpha}_t$ in the *__init__* function of the Diffusion class. (See files *ddpm.py* and *playground.py*)

2. Complete the implementation of forward process in the function *q_sample*. (See files *ddpm.py* and *playground.py*)

3. Complete the implementation of the reverse process in the function *p_sample*. (See files *ddpm.py* and *playground.py*)

4. Implement the training function. (See file *ddpm_train.py*). Training takes around 1 hour on a CPU with

a reduced dataset size of 40K images. You should be able to see reasonable image generations between epochs 20-30 (tested with SEED=1). Set DATASET_SIZE to None if you want to train on the full dataset.

5. (OPTIONAL) Read about Denoising Diffusion Implicit Models (DDIMs) [1] and adapt your exercise to include also a DDIM version. Do you observe any differences?

**Notation:** In the lecture, we follow the notation of the ddpm paper, while in the code, we follow the notation from the OpenAI code repository. Here we provide a mapping between the two.

- $T$ is the total number of diffusion steps
- $x_t$ = image at timestep t
- $x_T \sim \mathcal{N}(0, \mathbf{I})$
- $\beta_t$ = betas[t]
- $\alpha_t$ = alphas[t]
- $\bar{\alpha}_t$ = alphas_bar[t]
- $q(x_t|x_0)$ = q_sample
- $p_\theta(x_{t-1}|x_t)$ = p_sample

# References

[1] Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." ICLR 2021.