Mini-project 1

Advanced Machine Learning (02460)
Technical University of Denmark
Jes Frellsen and Mikkel Niklas Rasmussen

February 2026
(Version 3.0)

1 Formalities

This is the project description for the first mini project in *Advanced Machine Learning* (02460). The project is part of the course exam and will count towards your final grade.

Deadline You must submit your report as a group electronically via DTU Learn by the 5th of March 2026 at 12:00 (noon).

Groups You must do the project in groups of 3–4 people. You need an exception to deviate from this group size. You do not need to document individual contributions.

What should be handed in? You must hand in a single report in PDF format and your code in a single file (either a zip or tar archive).

Length The report must consist of:

1. A single page with the main text, including figures and tables. This page must include names, student numbers, course number and the title "Mini-project 1" (so do not include a front page).
2. A single page of well-formatted code snippets.
3. A single page containing the declaration of use of generative AI.
4. Unlimited pages of references.

You must use the provided template for the project (`template.tex`), without changing the font size and margins. Any content violating the space limitation will not be evaluated. No additional appendices are allowed.

Code You may use all code you were given during weeks 1–3 in the course and any code given as part of the project.If you use code from elsewhere, it must be documented in the report.

2 Project description

In this project, you will train and evaluate various deep generative models on binarized MNIST and standard MNIST. The project is divided into three parts.

Part A: Priors for variational autoencoders (VAEs)

We will consider different priors for VAEs with a product of Bernoulli likelihood, $p(\mathbf{x}|\mathbf{z})$, trained on binarized MNIST, where pixels with values over 0.5 are set to 1, and pixels with values less than 0.5 are set to 0. You need to train VAEs with all the following priors:

- A standard Gaussian prior.
- A mixture of Gaussian (MoG) prior.
- A Flow-based prior.

For each of the three priors, you have to:

- Show a plot of the prior and the aggregate posterior.
- Discuss and compare across priors the match between the priors and the aggregate posteriors.
- Evaluate the *test set* log-likelihood as approximated by the ELBO.
- Discuss and compare the test set log-likelihood across priors.

You must report the mean and standard derivation over multiple training runs when reporting the test set log-likelihood. You also need to briefly describe the architectures used for the encoder and decoder.

Part B: Sampling quality of generative models

In the second part, we consider generative models trained on standard MNIST (i.e., non-binarized). You need to train:

- A Denoising Diffusion Probabilistic Model (DDPM) using a U-Net.
- A DDPM trained in latent space of a β-VAE that has a Guassian likelihood, $p(\mathbf{x}|\mathbf{z})$.

In the report, you must:

- Show four representative samples from each of the two diffusion models (DDPM and latent DDPM), and four samples from a VAE of your choice (trained on either binarized MNIST or standard MNIST).
- Compute and report the Fréchet Inception Distance (FID) for samples generated from each of the three generative models (DDPM, latent DDPM, and the chosen VAE) using the provided code (the function `compute_fid` in `fid.py`). For the latent DDPM, you must additionally report FID scores for different values of β, including $\beta = 10^{-6}$.
- Measure and report the wall-clock sampling time (e.g., samples per second) for the chosen VAE, DDPM, and latent DDPM.
- Discuss and compare the sampling quality and FID scores across the three models in relation to their sampling times.
- Plot, discuss, and compare the prior from the β-VAE and the learned latent DDPM distribution against the aggregate posterior.

You must also briefly describe the neural network architectures used in the DDPM and latent DDPM.

Part C: Code snippets

In the code part of the report, you must include code snippets from your implementation showing the following:

- How you evaluate the VAE ELBO for a non-Gaussian prior, e.g., the MoG prior or the Flow-based prior.
- The central parts of your Flow implementation.
- Your implementation of the DDPM ELBO.
- Your implementation of the DDPM sampling algorithm.
- The central parts of your latent DDPM training and sampling steps.