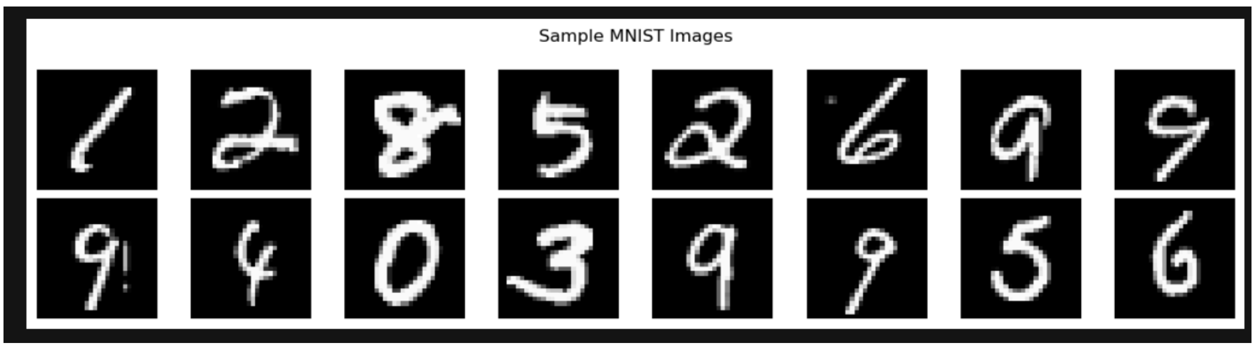
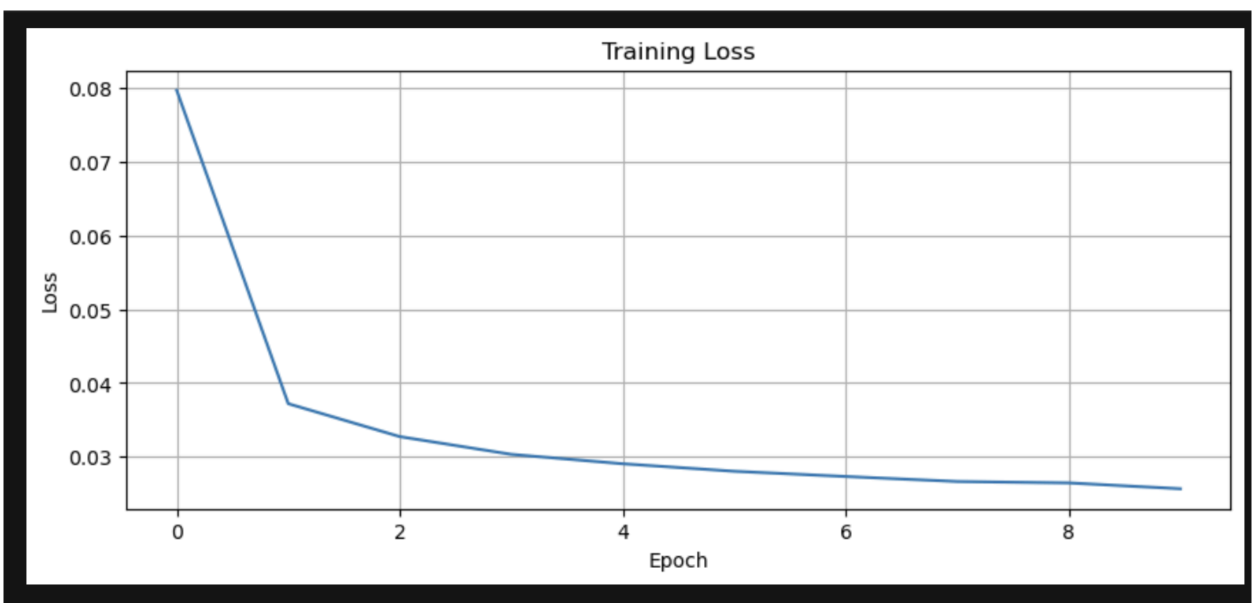
(a) Brief description of the diffusion model and its key components

In this lab we implemented a Denoising Diffusion Probabilistic Model for generating MNIST digits. The model has two phases: a forward process, which gradually adds Gaussian noise to data until it becomes pure noise, and a reverse process, which learns to denoise step by step so we can generate new images from random noise. Key components include the noise scheduler, which defines a linear schedule of betas and precomputes alpha_t, alpha_bar_t, and their square roots so we can add noise in one step via x_t = sqrt(alpha_bar_t) x_0 + sqrt(1 - alpha_bar_t) epsilon. The U-Net backbone takes a noisy image and a sinusoidal timestep embedding and predicts the added noise epsilon_theta(x_t, t). Training minimizes the MSE between the true noise and the predicted noise over random timesteps. Sampling starts from x_T ~ N(0, I) and repeatedly applies the learned reverse step to obtain x_{t-1} from x_t until we get a clean image at t = 0.

(b) Screenshots of generated images and training loss curve





(c) Reflection

Implementing the DDPM from scratch clarified how the forward process is fixed (just adding noise with a schedule) while the model only learns the reverse process by predicting noise. Seeing the reverse diffusion visualization made the iterative denoising idea concrete. The same core ideas underlie large-scale models like Stable Diffusion, so this lab was a solid foundation for understanding modern diffusion-based image generation.