1. **Understanding Diffusion**

    a. **Explain what happens during the forward diffusion process, using your own words and referencing the visualization examples from your notebook.**

    The forward diffusion process gradually adds noise to an image over many steps. Each step slightly degrades the image by blending in Gaussian noise. Over time, the image becomes more and more random until it turns into complete noise. This process is what the model learns to reverse. When visualized in the notebook, I could clearly see how the original digit image slowly got fuzzier with each step until nothing recognizable remained. This helped me understand how the model has to learn to "denoise" from any point in that chain.

    b. **Why do we add noise gradually instead of all at once? How does this affect the learning process?**

    Adding noise step-by-step allows the model to learn how to reverse small degradations. If noise were added all at once, the learning task would be too hard. This gradual approach teaches the model to reconstruct clean images in a controlled way and makes the learning process more stable. Each timestep teaches it how to go from a slightly noisy version to a slightly cleaner one.

    c. **Look at the step-by-step visualization - at what point (approximately what percentage through the denoising process) can you first recognize the image? Does this vary by image?**

    In my results, images started looking somewhat recognizable around step 200 (out of 300), and became more defined closer to step 100. At the final few steps, they looked very close to the clean training images. This matches the idea that earlier denoising steps deal with random noise, while later ones handle finer details.

2. **Model Architecture**

    a. **Why is the U-Net architecture particularly well-suited for diffusion models? What advantages does it provide over simpler architectures?**

    U-Net is a powerful architecture for image-to-image tasks because of its symmetric encoder-decoder shape and skip connections. For diffusion, U-Net allows the model to handle both the large-scale and fine-scale details of the noisy input, which is essential for reversing the noise. It has enough depth to capture complex patterns and a wide enough structure to maintain context.

    b. **What are skip connections and why are they important? Explain them in relations to our model**

Skip connections pass feature maps from earlier layers directly to deeper layers, which helps preserve image details that might otherwise get lost. Without skip connections, the decoder part of the network might struggle to reconstruct the image, especially at high resolutions. They also help with training stability and gradient flow.

c. **Describe in detail how our model is conditioned to generate specific images. How does the class conditioning mechanism work?**

In this project, class conditioning was done using one-hot encoded vectors for each digit (0 to 9). These vectors were embedded and passed into the model, influencing the generation process. Essentially, the model learned to generate images not just based on noise and timestep, but also based on the intended class. This helped it produce more accurate samples per class during generation.

3. **Training Analysis**

a. **What does the loss value tell of your model tell us?**

The training loss represents how closely the model's predicted noise matched the actual noise added to the images. A lower loss means the model is doing a better job at predicting the noise, which in turn means it's learning how to reverse the diffusion process effectively. In my case, the loss steadily dropped during training, showing progress.

b. **How did the quality of your generated images change change throughout the training process?**

Early in training, generated images were blurry or completely wrong. But after a few epochs, digits started to form. Midway through training, I could already tell which digit the model was trying to create. By the final epochs, the images looked sharp and were correctly shaped. The improvement was gradual but very noticeable.

c. **Why do we need the time embedding in diffusion models? How does it help the model understand where it is in the denoising process?**

Time embedding tells the model which step of the diffusion process it's working on. Since the model needs to denoise differently depending on how noisy the image is, giving it a sense of "time" helps guide that process. These embeddings let the model learn what to do at each specific step.

4. **Practical Applications**

a. **How could this type of model be useful in the real world?**

Diffusion models can be used for generating art, creating realistic images from

text prompts, medical imaging (like reconstructing MRIs), data augmentation, and even restoring old or damaged photos. They're also being explored for video generation and 3D modeling.

b. What are the limitations of our current model?

One big limitation is the time it takes to generate images, the step-by-step process is slow. They also need a lot of training data and GPU resources. Sometimes, the outputs look realistic but may not be logically correct (like wrong digit shapes). Plus, small changes in input can lead to very different results.

c. If you were to continue developing this project, what three specific improvements would you make and why?

Three improvements I'd propose:

- Speed up generation using fewer denoising steps with similar quality (some papers are already doing this).
- Use CLIP or another guidance method to improve semantic accuracy.
- Add better training regularization like dropout or noise scheduling to improve stability, especially with fewer samples.