Homework Report

Filip Fedor

May 27, 2025

Introduction

I completed all the tasks. During finetuning I had issues because I initially didn't use 8-bit optimizers such as bnb.optim.AdamW8bit, as recommended. I hadn't used this before and didn't think it would make such a difference.

The problem was that, when using float16 prec on Colab with a T4 GPU, the loss became NaN after the first iteration. So switching to AdamW8bit helped with numerical stability.

1 Task 1

Show the images generated for the same initial sampled noise, for different step counts and different noise schedulers. Describe your findings and give an explanation for them.

Generated Images

Each column corresponds to a different random seed (42, 43, 44, 45), which determines the initial noise.

The rows show images generated with different configurations:

- The first 3 rows use the DDPM scheduler with 200, 50, and 10 steps in denoising (in this order)
- The next 3 rows use the DDIM scheduler, with 200, 50, and 10 steps in the same order.

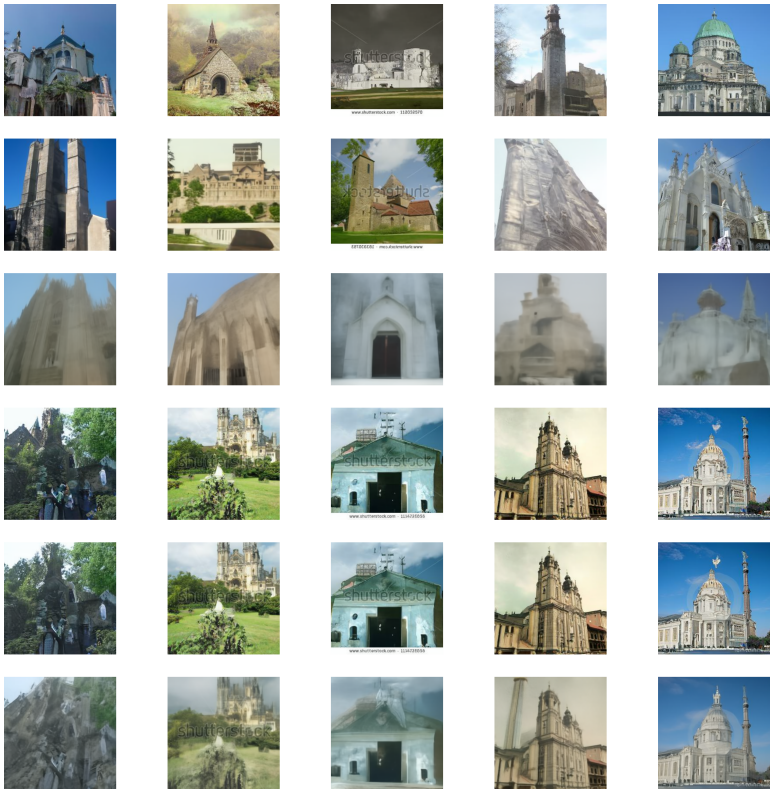


Figure 1: Caption describing the images for Task 1.

Findings

1. **DDPMScheduler**

   The reverse process samples from:

   $$x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$$

   where the distribution is:

   $$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

   with:

   $$\Sigma_\theta(x_t, t) = \beta_t \mathbf{I}$$

   and the mean function:

   $$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

   where:

   - $\alpha_t = 1 - \beta_t$

- $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$
- $\epsilon_\theta(x_t, t)$: neural network predicting the noise component

The denoising step samples from this distribution as:

$$x_{t-1} = \mu_\theta(x_t, t) + \sqrt{\beta_t} \cdot z, \quad z \sim \mathcal{N}(0, \mathbf{I})$$

Results:

In each denoising step we add random Gaussian noise: $\sqrt{\beta_t} \cdot z$, where $z \sim \mathcal{N}(0, \mathbf{I})$. So from that the sampling process is stochastic. So even with the same initial noise, different random z at each step lead to different outputs.

When we use many steps (in our case 200), each denoising step removes just a small amount of noise, and the model can correct small mistakes: this leads to high-quality, detailed images.

However, with fewer steps (e.g. 50 or 10), the steps are larger, and model has less time to clean up the noise. Since noise is added at every step, the model cannot fully denoise the sample, that's why we have blurry, distorted images.

2. **DDIMScheduler**

   The DDIM sampler is a non-Markovian deterministic process defined by:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon_\theta(x_t, t)$$

   where:

   - $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$,
   - $\epsilon_\theta(x_t, t)$: neural network predicting the noise component
   - in the formula there is no stochastic noise term

   Results:

   Even with only 10 steps, the generated images roughly preserve the object structure compared to 200 steps, although with lower quality.

   This is because DDIM removes the stochastic noise term $\sqrt{\beta_t} \cdot z$ present in DDPM and uses a deterministic update. This allow DDIM to take larger (fewer) steps without losing the overall structure.

   Because the sampling is deterministic, using the same initial noise always produces the same image.

2 Task 2

Show the images generated for your prompts and specify which prompts did you use. Make sure to span different categories - the instance prompt that you will use for finetuning in the future sections, something with dogs, something with other animal, something with humans and something like a landscape or inanimate object.

Generated Images

In each row, there are 3 images corresponding to a single prompt. The columns represent different seeds (in this case: 42, 43, 44). The prompts are the same as listed below:

1. a sailboat at sea;
2. a child playing;
3. a photo of a cat;
4. a photo of a dog;
5. a photo of sks dog;

The generated images follow the same order. Images were generated using the model: stable-diffusion-v1-5/stable-diffusion-v1-5.

We are mostly interested in the images for the prompt "a photo of a dog". These dog photos should not be tied to any specific breed or appearance.

The prompt "a photo of sks dog" in the fifth row currently shouldn't have any specific meaning. It should generate close images to the 4th prompt.

Figure 2: Generated images for each prompt (rows) and different seeds (columns: 42, 43, 44).

3 Task 3

Show the images generated for the same prompts and initial noise as in the previous section. Describe changes in different categories of objects. Generate images from additional prompts of your choice, showcasing your solution's strengths.

Generated Images

As in the previous section in each row, there are 3 images corresponding to a single prompt. The columns represent different seeds (in this case: 42, 43, 44). The prompts are the same as listed below:

1. a sailboat at sea;
2. a child playing;
3. a photo of a cat;
4. a photo of a dog;
5. a photo of sks dog;

Below images follow the same order. In this section, the images were generated using transfer learning. We finetuned the same model used previously: stable-diffusion-v1-5/stable-diffusion-v1-5 (using dreambooth). For the training we use 5 dog images of a specific breed along with the prompt: "a photo of sks dog". Here are images of this "sks dog".

Figure 3: Images used for finetuning

Figure 4: Generated images after finetuning for each prompt (rows) and different seeds (columns: 42, 43, 44).

And also images with different prompts such as:

1. a dog running on the beach
2. a photo of a dog in the park

Figure 5: Generated images after finetuning for additional prompts (rows) and different seeds (columns: 42, 43, 44).

Findings

As mentioned earlier, in the base model, the token "sks" has almost no meaning. So the goal after finetuing is for the model to associate "sks dog" with the same dog (breed) as in the photos we used in training. But it is not so ideal, after training the model also associate the "dog" with this specific breed.

Now we can clearly see that the token "sks" has meaning for the model by looking at the 5th row in Figure 4. But unfortunatelly also for the prompt "a photo of a dog" (4th row in Figure 4) the model now tends to generate images of a dog that closely resembles the one used in training with the "sks dog" prompt.

Also we can see some differences between the cat images in Figure 2 and Figure 4. After finetuning the cats on the images has more white fur around the neck and a white line over the nose. These are the features similiar to those in our images of a "sks dog" we used in training (Figure 3).

This suggests that the model's general representation of "dog" has been partially overwritten by features of the specific dog we used in finetuning.

Moreover, the finetuning process has also influenced the model's representations of other animals, making unintended feature transfer, that was not expected.

Also as it was mentioned in the notebook "The further away a concept from a dog, the less this change should be visible...". We can clearly see this from our results. The images for prompt "a sailboat at sea" are almost the same in Figure 2 and Figure 4 (after finetuning).

On the other hand, for prompts such as "a dog running on the beach" or "a photo of a dog in the park", we can see that while the dog now resembles the finetuned "dog" (the "sks dog"), the scenarios and environments remain diverse.

4 Task 4

Show the images generated for the same prompts and initial noise as in the previous sections. Describe changes between current images and those from section 2

Generated Images

The generated images are the same as in the previous sections. So once again for each row, there are 3 images corresponding to a single prompt. The columns represent different seeds (in this case: 42, 43, 44). The prompts are the same as listed below:

1. a sailboat at sea
2. a child playing
3. a photo of a cat
4. a photo of a dog
5. a photo of sks dog

Below images follow the same order. Here in this section we use prior preservation method. As before we finetune the model on a prompt "a photo of sks dog", but this time we also include a class prompt "a photo of dog" along with 50 images of generic dogs. This is done to help keeping the general concept of a dog while still learning the specific features of the "sks" dog. Below we can see generated images after such a training.

Figure 6: Generated images after finetuning with class image preservation for each prompt (rows) and different seeds (columns: 42, 43, 44).

And also images with different prompts such as:

- a dog running on the beach
- a photo of a dog in the park

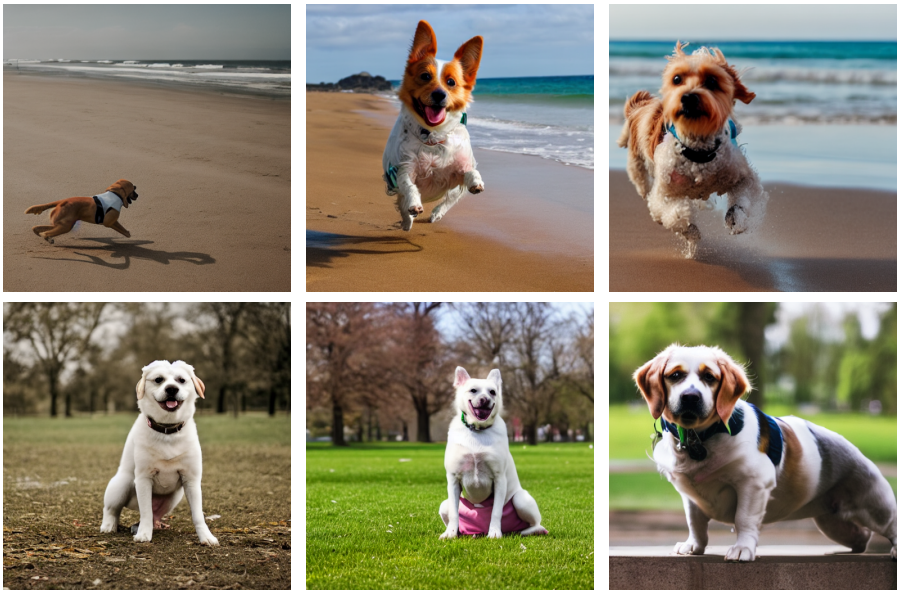


Figure 7: Generated images after finetuning with class image preservation for each additional prompt (rows) and different seeds (columns: 42, 43, 44)

Findings

Now, what do we expect from this setup? After applying prior preservation during training, we expect that the model will keep the specific features of the "sks" dog for the prompt with "sks", but will no longer apply its features as strongly to unrelated animals or generic prompts as it was observed in Section 3.

The results quite confirm our expectations. For example let's take the cat prompt (third row): in Figure 6 the generated cats are much closer in appearance to those in Figure 2, meaning with prior preservation the cats no longer have such a strong features of our "sks" dog, unlike in Figure 4, where the cats showed characteristics such as white fur around the neck and a white line over the nose.

Next, let's look at the prompt "a photo of a dog" (fourth row). In Figure 4, the images are almost identical to those generated for the prompt "a photo of sks dog", indicating that the finetuning without prior preservation (section 3) caused the model to strongly associate the general concept of "dog" with the specific appearance of the "sks dog".

However, in Figure 6 (with prior preservation), the images for the prompt "a photo of a dog" shows fewer "sks dog" features. They are more similar to the outputs from the base model in Figure 2, suggesting that prior preservation helps keeping a more general concept of "dog". Of course, there are still some "sks dog" features present: such as similar coloring, which shows that prior preservation reduces overfitting, but it does not eliminate it completely.

5 Task 5

Show the images generated for a generic dog prompt, as well as your special finetuned dog. Demostrate different ways of mixing the two models.

Generated Images

For each prompt, we examine how the generated image changes when switching between the original model and the finetuned model at different points during the denoising process (total of 50 steps). We consider:

1. Original → Finetuned

   We start with the original model and then change to finetuned one after:

   - 0 steps
   - 15 steps
   - 40 steps

2. Finetuned → Original

   We start with the finetuned model and then changed to the base one after:

   - 10 steps
   - 35 steps
   - 50 steps

The structure of the images is the same as described above. For each prompt, we first present results for the Original → Finetuned setup, with model switching occurring at three different denoising steps and then the same for Finetuned → Original. So each row corresponds to different denoising steps and columns represent different seeds. Used DDIMSchedulers.

a sailboat at sea – Original → Finetuned

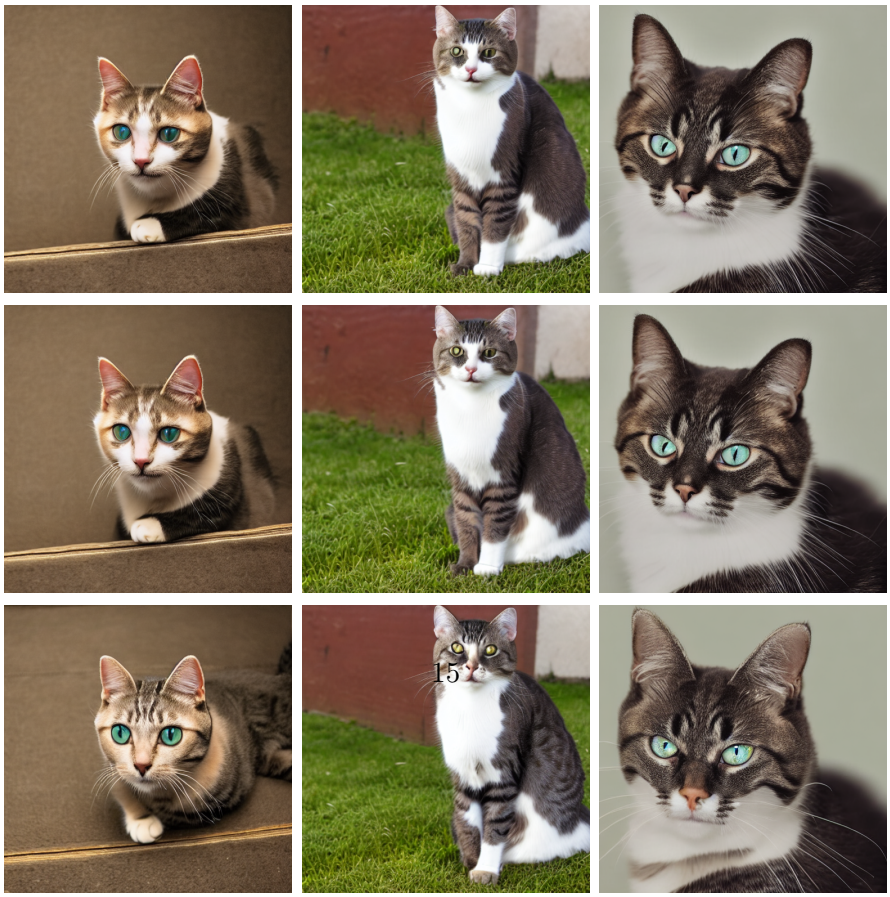
a sailboat at sea – Finetuned → Original

a child playing – Original → Finetuned

a child playing – Finetuned → Original

a photo of a cat – Original → Finetuned

a photo of a cat – Finetuned → Original

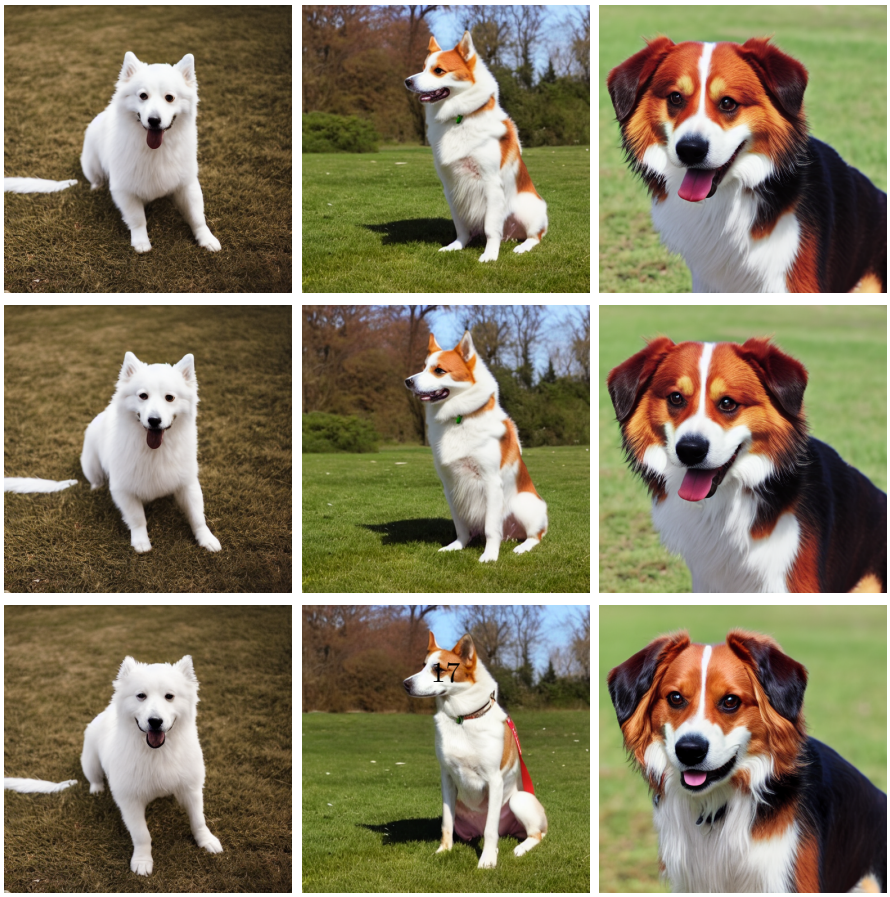


a photo of a dog – Original → Finetuned

a photo of a dog – Finetuned → Original

a photo of sks dog – Original → Finetuned

a photo of sks dog – Finetuned → Original



a dog running on the beach – Original → Finetuned

a dog running on the beach – Finetuned → Original

a photo of a dog in the park – Original → Finetuned

a photo of a dog in the park – Finetuned → Original

Findings

What we observe for all prompts is that when we start with the finetuned model and switch to the original model later in the denoising process, the generated images retain strong features of the "sks" dog, regardless of the step at which we switch the models.

This suggests that the early denoising steps performed by the finetuned model already encode key identity features of the "sks" dog, which the original model does not overwrite in the later steps.

A similar pattern holds when we start with the original model and then switch to the finetuned model. Except for the case when the we switch at step 0 (first row), which of course results in images almost the same as for finetuned model.

Looking at the results the best configuration (for my eyes) is: Original $\rightarrow$ Fituned with switching at step 15. This setup seems to preserve much of the base model's general structure and also introduce features of the "sks" dog in a controlled style.

Conclusion

Summarize the key takeaways from all tasks. Mention any insights, potential improvements, or open questions.

Key takeaways

- Dreambooth finetuning teaches the model to associate the specific token "sks" with a specific dog appearance. However, the model not only learns this new concept but also propagate this to the class of dog and even to the other animal classes.
- Prior preservation with class images (e.g. dog" photos) during training helps model to keep its general understanding of the dog class while learning new concept of the "sks" dog.
- Model switching during the denoising process showed that most of the information is encoded in the early steps.

Potential improvements

- In model switching we could use more different switching points (e.g. 0, 5, 10, 15, 20, ...). Potentially we could see when specific features emerge during generation.
- Use a better dataset for class images in prior preservation.