Project Report: Text to Movie Poster Generation

Luis Henrique Simplicio Ribeiro, Alison Yao, Isidora Diaz

May 8, 2023

1 Method

We have a conditional generative modelling task, where we want to have access to a model $p_\theta(x|\hat{y})$, which is able to create a rendition of a realistic movie poster x, whose content is given by a textual description encoded as $\hat{y}$. In order to solve this, we will use Stable diffusion [1], a latent diffusion model that was trained on millions of image-caption pairs (stemming from a subset of LAION-5B [2]). The model is composed of CLIP ViT-L/14 [3], a 123 million parameters text encoder that maps a text prompt to its vector representation. This vector representation is then combined with a sampled noise, and given as input to a 860-million-parameters UNet [4], which sequentially predict and remove the added noise for each time step $t \in \{T, T-1, \cdots, 1\}$, allowing the generation of an image x_0. In our study, we use this framework to find the prompt structure p, with, $\hat{y} = \tau_\phi(p)$, which combined with a series of hyperparameters chosen during inference time, including the number of denoising steps and image dimensionality, lead the model to generate a realistic movie poster that matches the prompt. We used the diffusers library [5] to interface with the model, and we performed a careful study to arrive in a good combination of hyperparameters.

2 Empirical Strategy

Since movie posters usually have an aspect ratio of 2:3, our initial goal was to generate images of size $768 \times 512 \times 3$. However, usually when the height is bigger than the width, we observed that the generated images would lose spatial coherence, and start to repeat elements over the vertical axis. Therefore, we ended up generating movie posters of dimension $512 \times 512 \times 3$. In order to generate images conditioned on an input prompt Stable Diffusion uses a technique called classifier-free guidance [6], by including the conditional information $\hat{y}$ as input for the network during the prediction of the added noise and controlling the guidance with the so-called guidance scale w. Increasing w has the effect of emphasizing the difference between the conditional prediction $\epsilon_\theta(x_t, \hat{y}, t)$ and the unconditional prediction $\epsilon_\theta(x_t, \varnothing, t)$, leading the generated image to move away from the unconditional image, and towards an image which matches $\hat{y}$. We tested many different values for w, and 7.5 led to a good trade-off between image quality and diversity.

Another important hyperparameter is the definition of how the variance schedule changes over time. For instance, DDPM [7] employs a simple linear variance schedule, iDDPM [8] on the other hand uses a cosine schedule that improves over DDPM by not destroying the structure in the input data too fast. After the exploration of different schedulers [9, 10] we decided to use DPM. We also tested various values for the number of denoising steps T, and due to a good trade-off between image quality and the time taken to generate an image, we decided to use $T = 50$. Finally, regarding the input text prompt, we specified that we want a movie poster. Then we describe the poster content instead of the movie plot. The input has the structure of who or what is doing something. We

can also specify the style and the background. An example is "a comic book style movie poster of the mandalorian". Since we were unable to directly specify that we wanted posters without text on them. We used the negative prompt "text on the movie poster" to remove the text. Negative prompts allow the specification of elements that we do not want present in the image. This is possible by exchanging the unconditional prediction for a prediction conditioned on the negative prompt in the classifier-free guidance setting. In this manner, when we increase w, we force our generated image to move away from the image conditioned on the negative prompt. We will have $\hat{y}_P$, the same as $\hat{y}$ before, and $\hat{y}_N = \tau_\phi(p_N)$, where p_N is the negative prompt, and $\hat{y}_N$ encodes the elements we want to remove:

$$\tilde{\epsilon}_\theta(x_t, \hat{y}_P, \hat{y}_N, t) = \epsilon_\theta(x_t, \hat{y}_P, t) + w(\epsilon_\theta(x_t, \hat{y}_P, t) - \epsilon_\theta(x_t, \hat{y}_N, t))$$

3 Results

Figure 1 shows some of the success cases of the applied strategy. In general, we are able to generate movie posters not relying on any existing franchise or character, and also for known franchises (like John Wick and Frozen). We can also specify the style of the movie poster and remove the text using a negative prompt. For instance, for the first image we used the positive prompt "a manga style movie poster of John Wick", and the negative prompt "text on the movie poster".

Figure 1: Samples of the generated images using our strategy

4 Analysis

The strategy applied is very promising, but in many cases, the model still generates low-quality images, in the sense of having blurred characters or images that do not match the prompt. Therefore, it is often needed to use different seeds to generate good images. Using a negative prompt was shown to be surprisingly effective, even though it leads to changes in the generated image beyond the removal of the text. Trying to directly specify on the positive prompt that we wanted a movie poster without text on it, did not work very well, we theorize that it might be difficult for the model to keep track of many different elements when generating an image, so it focuses on specific parts of the prompt. We also identify that the model is usually good to realize movie posters of famous franchises, but it might struggle with the depiction of unusual styles or backgrounds for those franchises. For instance, we used a prompt p to generate a Batman movie poster, to p we added " in a sunny day" and generated another image using the same sampled noise, and the model completely ignored the additional information, since it probably associates Batman with night.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[2] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[5] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[6] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[8] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[9] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[10] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

Appendices

A Image size

Figure 2: We fix the size and generate five images with the prompt "a movie poster about a werewolf trying to destroy the world in disney style". From top to bottom, we have images generated with sizes $1024 \times 512 \times 3$, $768 \times 512 \times 3$ and $512 \times 512 \times 3$ respectively. We can observe that the higher the difference between the height and the width, the more repeated elements we can see in the generated movie poster.

B Choice of scheduler and number of denoising steps

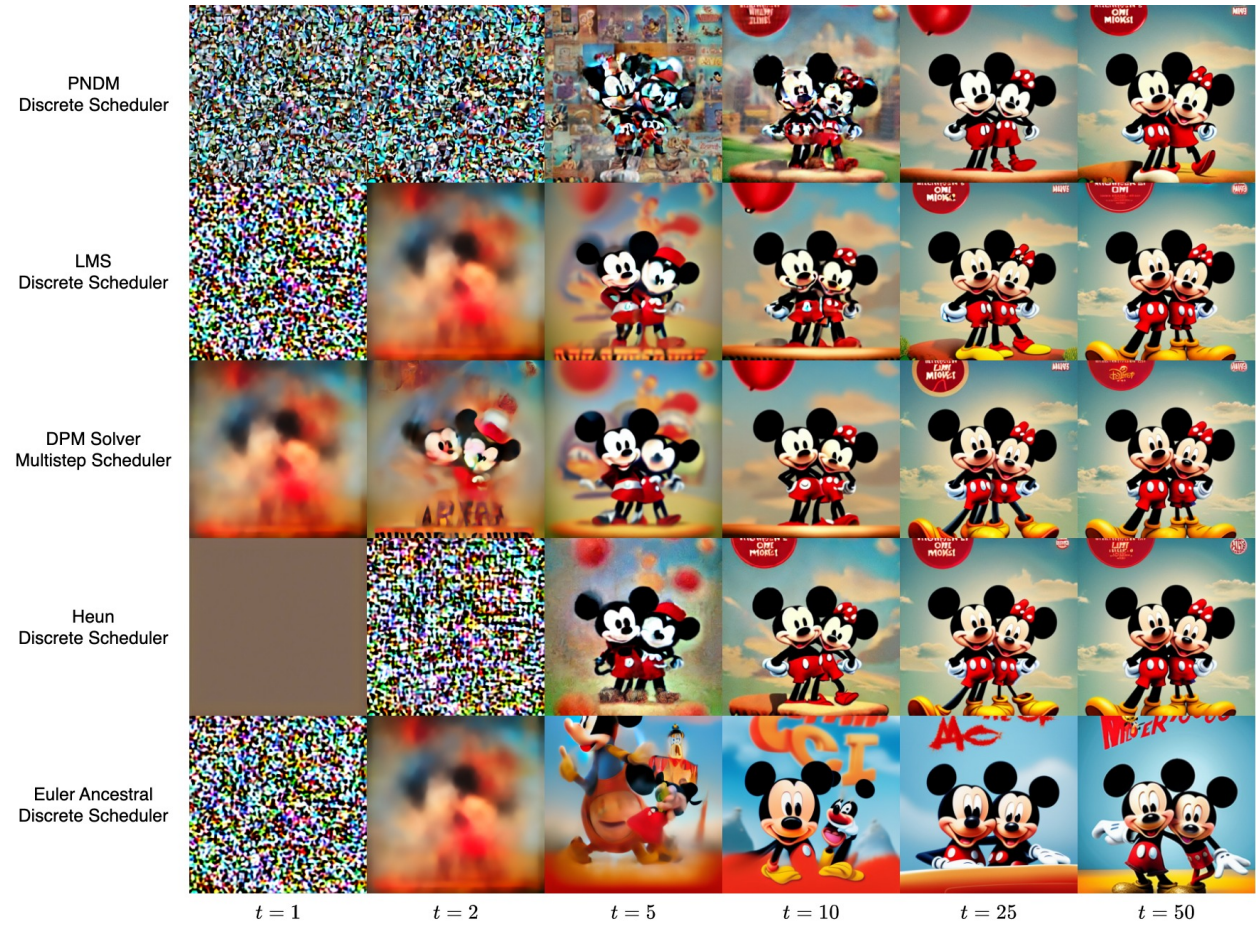

Figure 3: We tried 5 different variance schedulers. LMS, DPM and Heun usually converge to a similar image with the same number of time steps. PNDM usually needs additional time steps in order to have the image quality as the others, and the Euler Ancestral scheduler might lead to slightly different images, making it tricky to compare with the other schedulers.

C Effect of guidance scale

Figure 4: We also tested how the guidance scale w affects the generated movie posters. We fixed 6 different latent vectors and used $w \in \{0, 2, 7.5, 15, 30, 50\}$, with the prompt "a comic book style movie poster of Star Wars with Darth Vader on it". Higher values of w lead to better-looking images, but values that are too high might also lead to deformations in the final image.

D Model's ability to generalize a known concept

Figure 5: We test the model's ability to re-imagine a known concept. For instance, in this example, we check whether the model can create a movie poster with the characters from the famous franchise Shrek, with a slightly different style. We fix 5 different seeds for each of the rows. For the first row, we used the prompt "A movie poster of a blue Shrek", for the second row we used the same prompt, but we also added the negative prompt "text on the movie poster". For the third row, we used the prompt "a movie poster of Shrek" and finally, for the last row we used the same prompt as before, adding the negative prompt "text on the movie poster". As we can see the model sometimes struggle to change the color of Shrek to blue, since the character is strongly associated with the green color. We can also see how using negative prompts removes the text from the image but also change its style.

E Latent space smoothness

Figure 6: In order to understand the effect of the latent vector in the generated image we conducted latent space interpolation. We selected two noise vectors z_{start} and z_{end}, and generated the images on the top-left corner and bottom-right corner respectively, using the same prompt. After that we linearly interpolated between these two vectors, generating 4998 images. We picked 25 images of the corresponding equally spaced latent vectors, and observed that although the latent space is smooth, there are some regions where the transitions are more drastic than others. For instance, the transitions on the top are smoother than at the bottom, where even a male character is introduced. We also see regions that correspond to abstract concepts, such as the rows in the middle.