Sketch to Reality: Enhancing AI's Understanding and Creation of Art

Victor Barberteguy: victor.barberteguy@polytechnique.edu
Celia Nouri: celia.nouri@polytechnique.edu

***Abstract*—This project aims to bridge the gap between abstract sketches and realistic images through advanced AI techniques. Students will learn to generate descriptive captions for sketches, create realistic images from these sketches, fine-tune a generative model to improve its output, and apply conditional generation to control specific aspects of the generated images.**

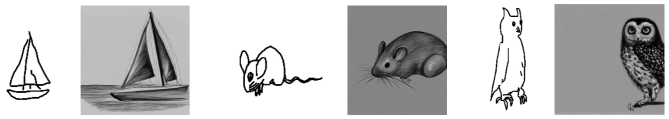

Fig. 1. Sketches from the sketchy dataset and corresponding sketches obtained by captioning the sketches and then using a conditioned latent diffusion model.

I. Introduction

In this project, we applied advanced AI models to assist artists' creative processes by bridging the gap between sketches and realistic images. This project is divided into three main tasks. First, we applied image captioning models to the sketchy database, which contains labeled sketches of different animals and objects. We combined a set of tricks to generate diverse and descriptive captions. Secondly, we used these captions to generate realistic images with Stable Diffusion. We are now able to generate realistic images from sketches using textual descriptions. Lastly, we fine-tuned Stable Diffusion for Sketch-Based Image Generation. In this last step, by integrating the concept of sketches to the diffusion model, we are able to generate sketches for any custom text prompt. The integrity of our codebase can be found here [1].

II. Image Captioning with Sketches

The first task was to generate image captions from sketches. We used and compared different image captioning models (LLaVA and BLIP-2) to generate textual descriptions for sketches from the Sketchy database. We tried different prompts and compared the descriptiveness and accuracy of the generated captions for the different models and prompting strategies.

A. LLaVA

LLaVA, that stands for Large Language and Vision Assistant, is a multi-modal model that bridges vision and language, and can be used on a wide range of tasks, among which classification, detection or captioning.

[1]The link to our code: https://github.com/VictorBbt/SketchToReality/

There are mainly three components in the model:

- A pretrained language model L, that takes as input an instruction in natural language and outputs an answer. The model used is Meta's opensource LLM LLaMA
- A pretrained vision encoder V, that computes vector embeddings from an input image. The model used Is CLIP ViT-L/14, that is left frozen during all the training
- A MLP projector W, which is used to map visual features to text embeddings

The training is divided in two steps. During the first step, the only updated weights are those of the projection matrix W, to align the visual and textual features. W takes as input an image embedding from V, and transforms it into a vector of the same size as the embeddings of L. This can be seen as a training "visual tokenizer" for the frozen LLM. Then, W and L are fine-tuned together on specific tasks. In [1], two specific tasks are investigated, namely Multimodal Chatbot and Science Question Answering.

In [2], the authors of LLaVA come with additional tricks to improve the model's performance. W is, in thre last version, a two-layer MLP and the training images are scaled up in order to deal better with the details in the images.

B. BLIP-2

1) Overview: Blip 2 [3] is a pre-training strategy to bridge the modality gap between state-of-the-art vision and language models efficiently. It enables zero-shot instructed image-to-text generation which, in our use case, enables us to generate captions and descriptions from sketches. The idea of BLIp-2 is to reduce pre-training costs by combining off-the-shelf frozen unimodal models: an image encoder to learn the vision-language representation and a large language model for text generation. The cross-modal alignment and communication between the image encoder and the LLM is done with the Query Transformer (Q-Former). The Q-Former is a lightweight transformer trained using a two stage training strategy. It is the only trainable component of the system, which makes BLIP-2 a generic and compute efficient visual-language pre-training method.

2) Model Architecture: The Q-Former is composed of two transformer submodules that share the same self-attention layers. The first module is an image transformer, which interacts with the frozen image encoder and the second module is a text transformer, which can be both a text encoder and a text decoder. A set of learnable query embeddings are created as additional model parameters and inputted to the image

transformer. The learnable queries interact with the image embeddings generated by the frozen image encoder through cross-attention layers. They also interact indirectly with the text input through sharing the same self attention layers. The Q-Former is initialized with the pre-trained weights of BERTbase.

3) Vision-Language Representation learning: In order to align visual representations coming from the frozen image encoder to the input text, the authors of BLIP-2 combined three training objectives. The objectives are using the same input formats and the same model parameters with different attention masking strategies, to control the interaction between the queries and the input text. First, the image-text contrastive loss learns to align the image and text representations by contrasting the image-text similarities of positive pairs those of negative pairs. This learning objective employs a unimodal self-attention mask to avoid information leaking by ensuring that queries and text cannot see each other. Second, the image-grounded text generation loss trains the Q-Former to generate texts based on the image representations transmitted through the learnt query embeddings. The multimodal causal self-attention mask is used here to control query-text interactions, such that queries can attend to each other but not to the text tokens, and text tokens can attend to all queries and all previous text tokens. Finally, the Image-Text matching objective uses a binary classification loss to learn a fine-grained alignment between image and text representation. A bi-directional self-attention mask is used to allow all queries and text to attend to each other. The output query embeddings thus carry multimodal information and are fed into a linear binary classifier that predicts if image-text pairs are matching or not matching pairs. Vision-to-Language Generation To connect the Q-Former to the LLM, the BLIP-2 authors first pass the output query embeddings through a fully-connected layer to project them to the LLM text embedding dimension. They are then prepended to the LLM text embedding as soft visual prompts conditions for the LLM text generation.

C. Prompt choice

First, we tried out for both studied models ten different prompts to get the most precise description of the provided sketches. For each prompt and each model, we generate captions for the same evaluation set composed of 100 sketches from the database. Indeed, as the sketches are very simple (see Appendix VI the prompt is even more important if one wants to have the LLM's provide a descriptive caption that goes beyond simple object recognition. We aimed to get the pose, view angle or sub-category (e.g, the type of dog). In other words, we want the captioning model to generate different captions for the hundreds of different dog sketches of the sketchy database.

To generate the prompts, we asked ChatGPT [2] to give us ideas of prompts of different types: from very factual (pose, sub-category) to narrative (invent a small narrative based on the sketch features).

[2]https://chat.openai.com

Among these sketches, we selected three that had the best performances in term of **length of the caption** and **accuracy**. By accuracy, we mean that the caption contains the label associated to a given sketch, that is if the model has corrected identified the main subject of the sketch. The three sketches and their scores can be seen in 2.

```
'prompt1':"USER: <image>\nWhat is drawn in this art sketch ? What are the specific features of this subject
(pose, where is it viewed from, sub-category, ...) ?"
'prompt2':"USER: <image>\nProvide a detailed description of the viewpoint and pose depicted in this sketch,
including any unique angles or perspectives"
'prompt3':"USER: <image>\nOffer a narrative interpretation of this sketch, considering its context, mood, and
potential symbolic meanings beyond simple object recognition."
```

Accuracy	Llava	**BLIP2**
Prompt 1	0.57	0.66
Prompt 2	0.59	0.61
Prompt 3	0.61	**0.69**

Length	**Llava**	BLIP2
Prompt 1	322.31	313.91
Prompt 2	402.77	256.18
Prompt 3	**437.3**	263.45

Fig. 2. The three prompts that gave the best results in terms of accuracy and answer length, with the corresponding values. We see that prompt 3 seems to be the best, providing best accuracy and length for both models

Now that we have referenced good prompts, we can generate images with the studied models and compare them. The table 2 seems to indicate that LLaVA is wordier than BLIP. The answers are longer, while having a comparable accuracy (although BLIP provides the best accuracy overall for prompt 3, 0.69 against 0.61). As we do not have reference captions (the model in the Sketchy paper is trained on sketch-image pairs), we cannot provide precise measures of descriptiveness, such as CIDEr [4].

Finally, for the three prompts and the two models, we read the generated captions to see if their were similarities between the answers generated by a given model. The captions can be found on the code page. We found that BLIP's captions are less interesting, focusing more on a simple interpretation ('the black and white colours give a nostalgic feeling') rather than on an isightful narrative ('the duck is viewed from the side, it could be wandering around at the surface of a shallow lake'). BLIP's might be interpreting at a higher level and describes precisely what is a sketch, however the narratives offered by LLaVA are often more subject specific. As 'art sketch' is in all the prompts and answers, we ensure to have a generated image that will be in a sketch style. For an interpretation of the generated images, see V.

III. Generating Images from Sketch Descriptions with Stable Diffusion

After generating descriptive and accurate captions for the Sketchy database in the previous task, we used these generated sketch descriptions to generate images with Stable Diffusion. In this section, our goal is to utilize Stable Diffusion to generate realistic images based on textual descriptions obtained using our image captioning algorithm. We first introduce Stable Diffusion and discuss its capabilities and limitations. Then, we describe how we use the generated captions to synthesize realistic images that match the sketch descriptions with Stable Diffusion.Finally, we report and assess the quality and relevance of the generated images to the original sketches and descriptions.

A. Stable Diffusion

1) Deep Diffusion Probabilistic Models: Deep Diffusion Probabilistic Models (DDPMs) are a class of generative models thqt achieved state-of-the-art sample quality fir image generation tasks. The key idea behind DDPM is that, by repetitively adding small amounts of Gaussian noise to an image, one will end-up with pure random noise in T steps (typically 1000). Similarly to the noising step process, the denoising steps can be modeled as a normal distribution. We can train a neural network (a UNet in this case), to learn the parameters of the denoising step distribution, and repetitively denoise the noised image, starting from pure random noise, for T steps.

2) Latent Diffusion Models: Latent Diffusion Models are a class of Diffusion models that greatly reduce the high computational demands of diffusion models training and inference by applying diffusion steps in a low dimensional latent space. The learning process is divided into two stages. First, a perceptual compression stage in which an autoencoder is trained to downsample the image into a lower dimensional latent space and decode it back to the original image. The universal autoencoder needs to be trained once only and one can reuse or finetune an existing image autoencoder model such as the CLIP image encoder. The second stage consists of the diffusion model in which the denoising distribution is learnt directly in the latent space. Beyond unconditional generation, the ability to model conditional distributions is crucial to condition the image generation process on user specified inputs. In our use case, we want to condition the generation on the sketch descriptions that we generated for task 1. Stable Diffusion uses a innovative conditioning strategy by incorporating cross-attention layers to the diffusion process. Cross-attention has proven to be an effective way to learn attention-based models of various input modalities. The input to condition on (the sketch descriptions here) is first pre-processed by a domain-specific encoder. The encoder maps the input into an intermediate representation that is mapped to the intermediate layers of the U-Net via cross-attention. This method allows for flexible conditioning and guidance of the generation process by diverse input types such as as classes, texts prompts, bounding boxes or other images.

Put simply, a pre-trained text-to-image diffusion model $\hat{x}_\theta$ can generate images $x_{\text{gen}} = \hat{x}_\theta(\epsilon, c)$, given an initial noise map $\epsilon \sim \mathcal{N}(0, I)$ and a conditioning vector $c = \Gamma(P)$ generated using a text encoder Γ and a text prompt P. They are trained using the following loss to denoise a latent embedding $z_t := \alpha_t x + \sigma_t \epsilon$ as follows:

$$\mathbb{E}_{x,c,\epsilon,t} w_t \|\hat{x}_\theta(\alpha_t x + \sigma_t \epsilon, c) - x\|_2^2 \tag{1}$$

where x is the ground-truth image, c is a conditioning vector we obtained with Classifier Free Guidance (CFG) as a text prompt for instance, and α_t, σ_t, w_t are terms that control the noise schedule and sample quality, and are functions of the diffusion process time $t \sim \mathcal{U}([0, 1])$.

B. Generating Realistic Images from Sketch Descriptions

Using the same test set of 100 sketches, we provide the Stable Diffusion model with the captions that were produced by the image-to-text models described above. All the generated images are available in Appendix VI, and Fig. 3 shows some examples of produced sketches. Although we provide and compare the results more thoroughly in V, we can see that when a black and white sketch is generated, it is always more complex and 'arty' than the very simple ones from the sketchy database. Rather than reproducing the style, we enhance the sketch and generate a drawing that would be drawn by an expert.

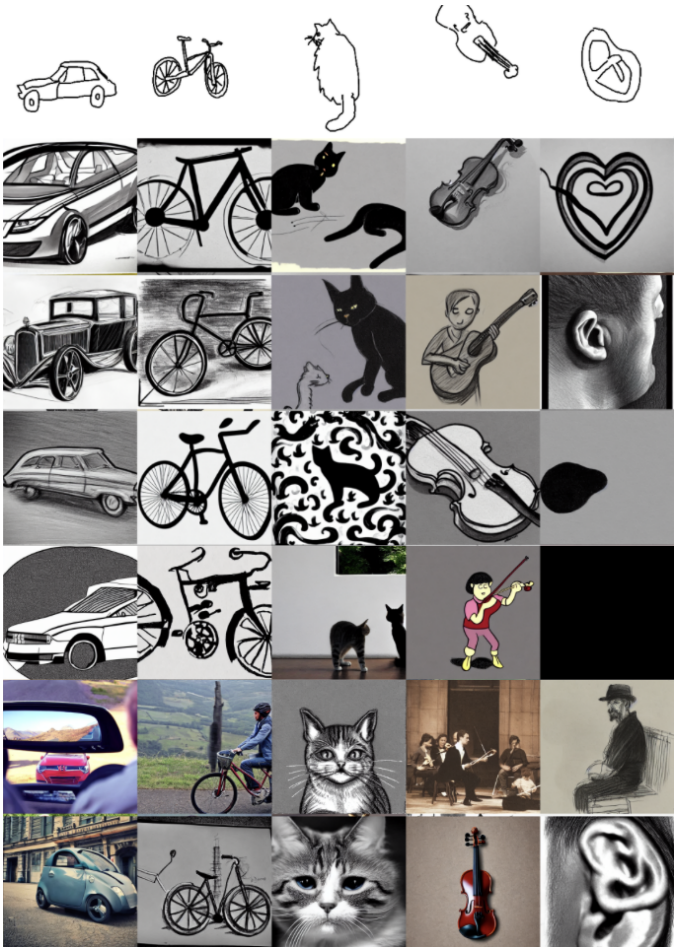

Fig. 3. Example of sketches in the sketchy database (first row), and output produced by LLaVA for prompt 1, 2 and 3 (respectively at row 2, 3, and 4) and BLIP2 for prompt 1, 2 and 3 (respectively at row 5, 6, and 7).

IV. Fine-Tuning Stable Diffusion for Sketch-Based Image Generation

In this last section, we explored the inverse Fine-tune Stable Diffusion on the Sketchy database to enhance its ability to generate high-quality images from sketches. One of the drawbacks of diffusion models is the impossibility to generate a specific subject in diverse contexts. When asking for a 'dog in Bora Bora' twice, the model won't output the same dog. To create visual stories, one would like to have a bunch of images of the same subjecct in different settings. In this work, we investigate two fine-tuning methods, namely Dreambooth [5] and Textual Inversion [6]. Both methods consists in assigning an unknown vocabulary token to a specific subject. For example, if I want

to generate images of my own dog, I can provide a few photos (usually 3 5) of my specific subject, and finetune the model to make it understand that a new word token like 'sks' corresponds to my specific subjects. This type of method has the advantages of being reasonable in terms of training time (compared to training from scratch a diffusion model), to not overfit to the data provided allthewhile integrating better the specific data distribution that we give. In other words, fine-tuning is a way to gear the diffusion model towards a specific task (in our case, integrating the style of sketch in our database).

A. DreamBooth

Dreambooth has been used successfully to integrate specific subjects in diffusion models. Here, we want to assess if Dreambooth is able to integrate a abstract concept, i.e a style rather than a specific subject. The goal is to assign a new vocabulary token to the idea of the style of our sketches in the database.

1) Model Overview: To use Dreambooth, one must provide a few images of a specific subject as well as a class name **[C]** ('cat', 'dog'), and then the model returns a fine-tuned text-to-image encoder with an identifier **[V]** ('sks') that encodes a precise concept in the diffusion model's "vocabulary". The tokens must associated with a weak prior in the model's dictionary, and must not be split during tokenization. To this end, trigrams are found to be the best tokens to encode a new concept. The model learns that the specific images are associated to 'a [V] [C]'.

The main challenge with this kind of fine-tuning, is to maintain the class specific prior. We don't want that the model overfits to our dog image (like mode collapse in Generative Adversarial Networks); we would like to keep the same intra-class diversity while being able to generate images of the specific subjects in diverse contexts. This issue is similar to that to language drift, in which a finetuned language model loses syntactic meaning as it overfits the finetuning data.

To prevent this drift, the authors came up with a **class specific prior preservation loss** to foster intra-class diversity. Once the finetuning starts, they supervise the model with its own generated samples. Using the forzen pre-trained diffusion model, samples $x_{\text{pr}} = \hat{x}(z_{t1}, c_{\text{pr}})$ are created using random noise and conditioning vector $c_{\text{pr}} := \Gamma(f(\text{"a [C]"}))$. A new term is added to the loss of 1 :

$$\mathbb{E}_{x,c,\epsilon,\epsilon',t}[w_t \|\hat{x}_\theta(\alpha_t x + \sigma_t \epsilon, c) - x\|_2^2 + \lambda w'_{t'} \|\hat{x}_\theta(\alpha_{t'} x_{\text{pr}} + \sigma_{t'} \epsilon', c_{\text{pr}}) - x_{\text{pr}}\|_2^2], \quad (2)$$

where the new term is the **prior-preservation loss** that supervises the model with its own generated images, and λ controls for the relative weight of this term. The training process takes up to 15 20min on Colab's [3] GPU. The process is summed up in Figure 4.

[3]https://colab.research.google.com

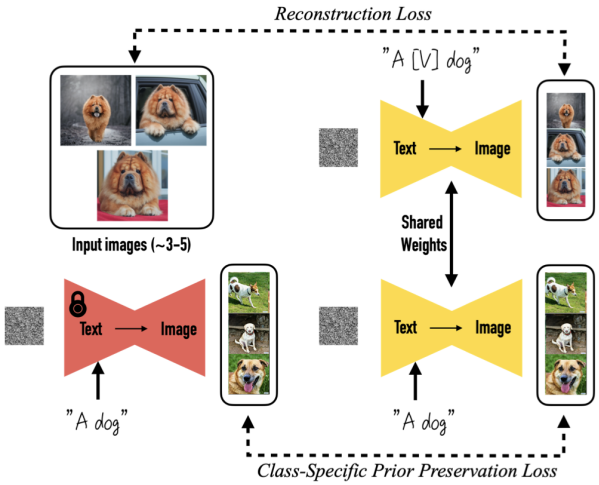

Fig. 4. Architecture and training of Dreambooth. We see that we encode the specific subject to a new word while maintaining diversity with the prior preservation loss.

2) Experiments: Successfully finetuning Dreambooth can be tricky, as there are a lot of hyperparameters to be chosen (number of epochs,. We decided to keep the recommended parameters, but to try different dine-tuning prompts ad sketches in order to make the model better understand the sketch style. The three processes we followed are shown in Fig. 5.

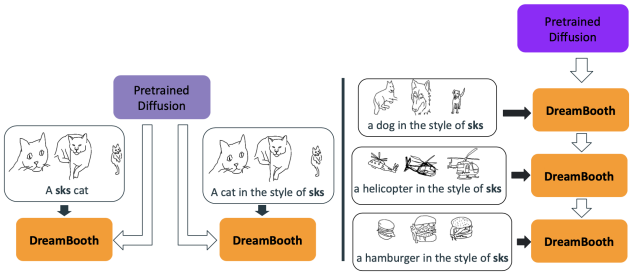

Fig. 5. Different processes to finetune our model using dreambooth. On the right, training on one class (cats) with different prompts. On the left, sequentially train the same model on sketches of different classes with the same prompt.

a) On a single class object: We use dreambooth on a single class object. We perform the finetuning once on a single class image like cat, but try whether the classical prompt provided in the article 'a sks cat', or a prompt that more reflects the context in which we want to use the token 'sks'. As we want the model to get the idea of a artistic style, we provide the prompt 'a cat in the style of sks'.

b) Sequential Training over multiple class objects: Next, we think that the the artistic style should be better integrated of the model is provided sketches of different objects. That way, it would be easier to differentiate between the specific object at hand (a cat, a helicopter) and the sketch style, that is the only common point between the data presented. However, doing this in a single training can be clumsy ('a cat, helicopter, and a hamburger in the style of sks'). Therefore we decided to sequentially train the same model three times on very different objects. We hope that the newly trained model has

not forgotten the prior over 'sks' it had before, but has rather updated it to correspond better to the artistic style.

The figure 6 shocases some generated images for the different finetuning methods desrcibed above.

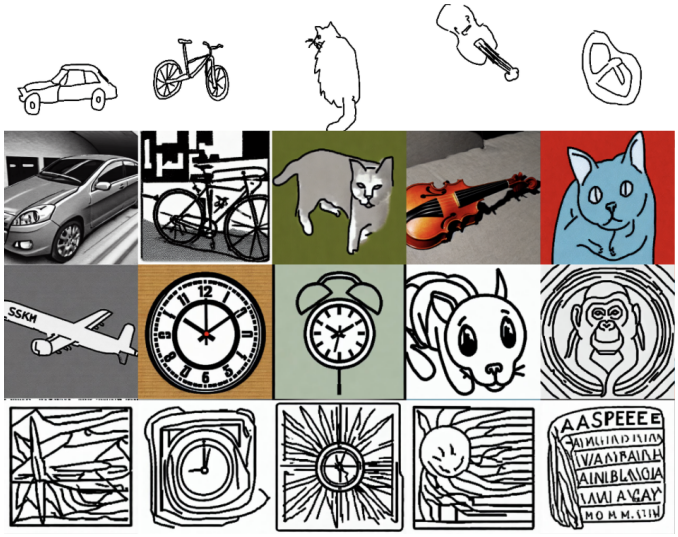

Fig. 6. Example of sketches in the sketchy database (first row), and output produced by Dreambooth for the *'sks cat'* (row 2), *'cat in the style of sks'* (row 3) and sequential training (row 4).

B. Textual Inversion

Textual Inversion [6] is an paper that was released before Dreambooth [5], however there is no need to specify a prompt like 'a [V] [C]'. The task is to infer a new word encoding only by providing pictures. Moreover, the user can specify whether the subject is an object or a style, which seemed to fit better to our precise task.

To do so, they directly optimize the loss from 1 over the of the specific set of input images. To condition these generations, they use a set of generic prompts like 'An image of [V]', 'A photo of [V]'. More precisely, [V] corresponds to a certain embedding of the text encoder v, and they replace the embedding of the tokenized string of the generic prompt 'An image of [V]' with v, thus injecting the concept in the vocabulary:

$$\underset{v}{\operatorname{argmin}}\, \mathbb{E}_{x,c,\epsilon,t} w_t \|\hat{x}_\theta(\alpha_t x + \sigma_t \epsilon, c) - x\|_2^2 \qquad (1)$$

The overall process of Textual Inversion can be seen in Fig. 7.

We use Textual Inversion in two ways, in order to compare it with Dreambooth IV-A. In both settings,we specify that we want to inject a new style. In the first configuration, we train it on the same three cat sketches as Dreambooth. Then, as we can provide as many images we want, we gave the 9 sketches that we used for the sequential training of Dreambooth. That way, we give the same information as Dreambooth in the sequential setting. Some sketches generated with these methods are in Fig. 8.

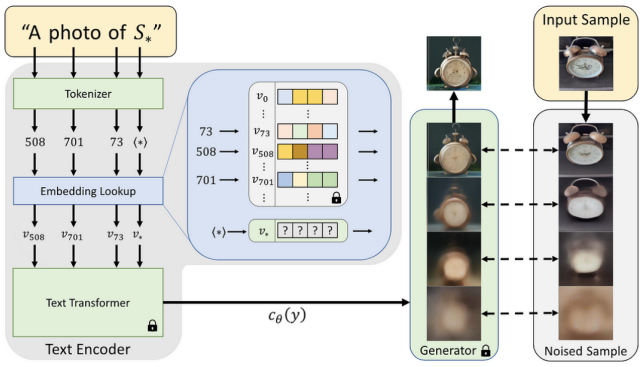

Fig. 7. Process of Textual Inversion. We directly optimize the model to find the correspoding embedding of the new word.

Fig. 8. Some images generated by Textual Inversion with the cat data only (row 1), and trained on all the data used for the sequential training of DreamBooth (row 2).

V. Results

Throughout the article, we generated images with the described methods on our test set of 100 sketches. All the mosaics produced are in the Appendix VI, and fewer examples were spread in the different parts of the previous parts 368.

A. Quality of the generation

To measure the quality of the generation, we compute the **Fréchet Inception Distance** [7] between the target distribution (the set of sketches from the sketchy database) and the generated images. We report the results in the Table 9.

Model used for generation	LLaVA Captioning prompt 1	LLaVA Captioning prompt 2	LLaVA Captioning prompt 3	BLIP2 Captioning prompt 1	BLIP2 Captioning prompt 2	BLIP2 Captioning prompt 3
FID	271	273	272	**270**	283	277

Model used for generation	DreamBooth sks cat	DreamBooth style of sks	Dreambooth sequential	Textual Inversion with cats	Textual Inversion with 4 classes
FID	232	**229**	253	273	276

Fig. 9. Frechet Inception Distances for the different captions generated by LLaVA and BLIP (see 2 to see which prompt corresponds to which number) and the different models finetuned with Dreambooth and Textual Inversion. The best FID is achieved by DreamBooth with *'a cat in the style of sks'*.

One striking remark overall is that the range of FID (above 200) means that the generated sketches are quite dissimilar to the original one. A general takeaway should be to finetune

the model on more data, or to choose carefully the hyperparameters in order to get more close-looking sketches. Then, we looking closer to Table 9, we see that within one type of model, the FIDs are quite alike. Indeed, for captioning models (LLaVA and BLIP), the different prompts do not make a huge difference in the FIDs, and neither do the different finetuning procedures for the Dreambooth and Textual Inversion Methods. However, there is a huge difference between the captioning methods and the direct finetuning models. The sketches generated by the captioning models, when they are effectively sketches, are very aesthetic, as if they were made by an expert. It enhances the sketches, but not does not integrate the style of the sketchy database. This is not the objective pursued in this work, but is a interesting idea for further investigation. The direct methods more generally generate sketches that are more alike the target, thus resulting in better average FID. However, textual inversion method rarely outputs real sketches, but more likely coloured images or real images, eventually leading to a performance comparable to that of the captioning models (in terms of FID).

Overall, Dreambooth seems to yield better results, as all the FIDs are below those of the other models. Qualitatively, we see on the mosaics that the sketches produced by the model really have the stroke that resembles the original sketches. Moreover, we find less coloured sketch tha in any of the other methods. The concept of the type of sketch we wanted to convey has been better integrated within the diffusion model. What is more surprising is that the sequential method (training the same model three times) has the worst FID, yet it seems that this is the model which is visually the best. This could be due to the presence of a lot of strokes in the background, as if 'sks' has the principal component of what produces the sketch, that is a background of strokes. This biases the pixel values taken into account, and then yields a higher FID. On top of that, the sequential model seems to have overfitted the face of the husky dog sketch, and we find some exact strokes of this sketch in several generated others.

B. Discussion

It is worth noting that the results discussed above are only calculated on the test dataset of 100 sketches, and thus might not reflect exactly the statistics of the models (the sketchy database contains thousands of sketches). We would need to generate more images to provide a more meaningful comparison.

Another idea would be to use more measures to assess the similarity of the generations and the target distribution, for example the CLIP-I measure, that consists in computing the cosine similarity between CLIP [8] embeddings of a pair of images, or the DINO measure [9] .

VI. Conclusion

Throughout this work, we tried to assess different methods to enhance the integration of a new art style in diffusion models. We investigated two approaches, by directly finetuning the diffusion models (with DreamBooth or Textual Inversion) or by interleaving an image captioning model (LLaVA or BLIP2). Furthermore, we tried different prompts or training configurations for each of these models before comparing quantitatively the quality of the generation with FID. A key difference between these two approaches is that text-in-the-middle approach could be more successfully to augment the capability of a unskilled user, i.e make their drawing more 'arty' and technically complex, rather than bein used to reproduce a style.

We found that DreamBooth yields the best results, and we are confident that if it is finetuned on more data and with better hyperparameters, a artistic style could be successfully encoded by a new word. This is what could be done in future work, as well as providing more quantitative and thorough evaluation of the quality and diversity of the generated images.

From an ethical point of view, it would mean that the model understands well the technicity of an artistic style (the stroke, the colours) rather than fully mastering the creative process of an artist. We do not declare that we have given a idea of creativity to our model, yet combining and deforming some style with such approaches can be part of a AI artist creative process.

References

[1] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[2] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved Baselines with Visual Instruction Tuning," Oct. 2023. arXiv:2310.03744 [cs].

[3] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," June 2023. arXiv:2301.12597 [cs].

[4] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," June 2015. arXiv:1411.5726 [cs].

[5] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," 2023.

[6] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion," 2022.

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," Jan. 2018. arXiv:1706.08500 [cs, stat].

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[9] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Montreal, QC, Canada), pp. 9630–9640, IEEE, Oct. 2021.

[10] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: learning to retrieve badly drawn bunnies," *ACM Transactions on Graphics*, vol. 35, pp. 1–12, July 2016.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," Dec. 2020. arXiv:2006.11239 [cs, stat].

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022. arXiv:2112.10752 [cs].

APPENDIX

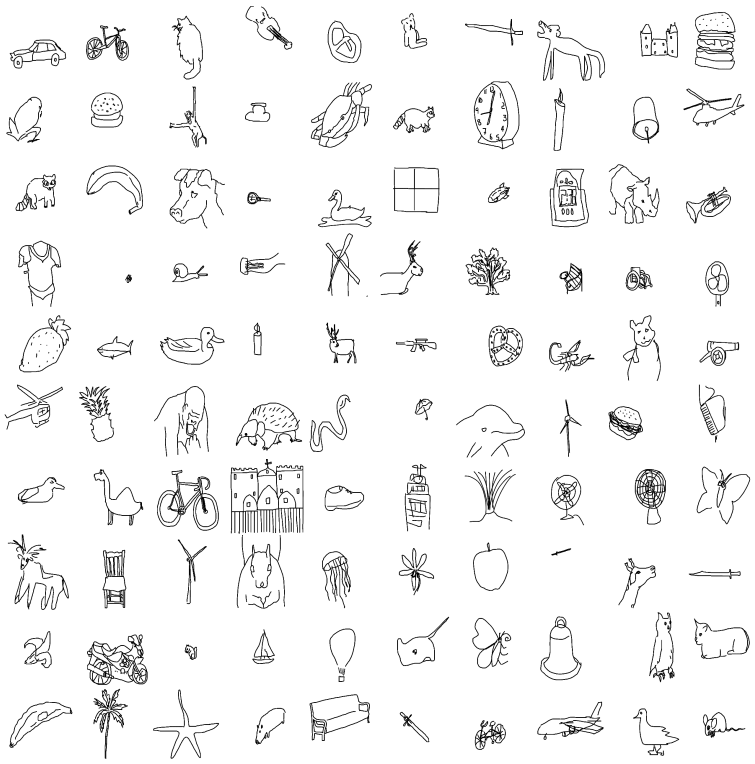

Fig. 10. Test set we used to generate new images. These 100 sketches are take from the sletchy database and are to be compared with those we generate with the introduced methods