Advanced Computational Learning and Data Analysis (52025)
Ibrahim Bashir & Yoel Graumann 15.9.2024

Deep Learning Project

Introduction

In this project, we aimed to enhance the image diversity of the SDXS model, a highly efficient Latent Diffusion Model (LDM) architecture. While the SDXS model is optimized for speed and resource efficiency, it can sometimes produce outputs that lack artistic diversity and image accuracy (how well the generated image reflects the input text prompt). To address this, we integrated the DreamShaper model and IP-Adapter techniques, which allowed us to significantly improve the range and quality of generated images from both SDXS and DreamShaper. However, this improvement comes at the cost of increased latency, a trade-off we explored in our approach.

Section 1 will cover background information on diffusion models, the models used (SDXS and DreamShaper), and additional techniques utilized. **Section 2** will cover our modifications, experiment design, and background information on metrics used for assessing different metrics of success. Finally, **Section 3** will display the results and **Section 4** will include a discussion of the results and a brief conclusion.

---

1 Background Information

1.1 Stable Diffusion Models and DreamShaper

DreamShaper is a type of model used for text-to-image generation; it is often used for creating images with a particular artistic or dreamlike quality. The outputs tend to have a soft, surreal, and often painterly aesthetic, making it popular for generating art that has a fantasy or imaginative feel, but is also a general purpose model aimed at and known for versatility (e.g. photos, anime, etc, akin to Midjourney and DALL-E). It is a stable diffusion model, so it has three main components:

Text-encoder: usually a simple transformer-based encoder, e.g., CLIP that ‘tokenizes’ the text prompt, then maps the sequence of input tokens to a sequence of latent text-embeddings to serve as input to the U-Net.
U-Net model: takes the tokenized embeddings and a random noise or noisy latent array as input, and returns a noisy latent image ‘guided’ by the text embeddings to serve as input to the vae’s decoder.
VAE: variational autoencoder, responsible for converting an image into a latent space/compressed representation. It takes the text-informed latent image and decodes it into the final output image. The checkpoints can also sometimes include an encoder which can serve as the latent image input to the U-Net for image-to-image translation.

1.2 Stable Diffusion XS (SDXS) Model

The SDXS model optimizes Latent Diffusion Model (LDM) architectures for efficiency by employing various distillation (process of compressing a large complex “teacher” model into a smaller, more efficient “student” model) techniques [Yuda Song et al., 2024]. The model distills key components like the image decoder, U-Net, and ControlNet into more compact versions. In the image decoder, a VAE Distillation (VD) loss minimizes redundancy by downsampling images and applying a GAN loss. The U-Net is distilled by selectively removing blocks while using both output and feature knowledge distillation to reduce computational complexity. ControlNet, which embeds spatial guidance into diffusion models, is distilled through feature distillation focused on the U-Net’s decoder, ensuring the distilled ControlNet works seamlessly with the smaller U-Net. A one-step generation strategy leverages a warmup phase using feature matching to reduce trajectory crossings during sampling. Lastly, Segmented Score Distillation divides the training into phases, applying feature matching to low-frequency components and score distillation to high-frequency components, which reduces sampling blurriness and enhances image sharpness. In summary, the SDXS model reduces computational complexity while maintaining image quality by distilling its components.

1.3 IP-Adapter Models

Getting desired outputs from text-to-image models depicting complex and unorthodox scenes can be intensive, requiring time-consuming and complex prompt engineering or even fine-tuning of the models. IP-Adapters can be used in such contexts to achieve image prompt capability for the pretrained text-to-image diffusion models [Hu Ye et al., 2023]. That is, they enable a pre-trained text-to-image diffusion model to generate images with image prompt. The IP-Adapter is composed of an image encoder to extract image features from image prompt (we use a CLIP image encoder), and adapted modules with decoupled cross-attention to embed image features into the pretrained text-to-image diffusion model. Model Structure Illustration of IP-Adapter with SDXS and DreamShaper

To enhance image diversity, we integrated an IP-Adapter to extract features from the SDXS model and embed them into the DreamShaper model (which has its own U-Net, text encoder and VAE). In this way, we leverage the fast strong artistic style alongside the image diversity achieved by the DreamShaper model. A future development could be to use models that perform even faster and possibly also geared towards even more diverse image generation, but we used DreamShaper given its standard use in the community and the high quality images we observed in its use alongside the SDXS model.

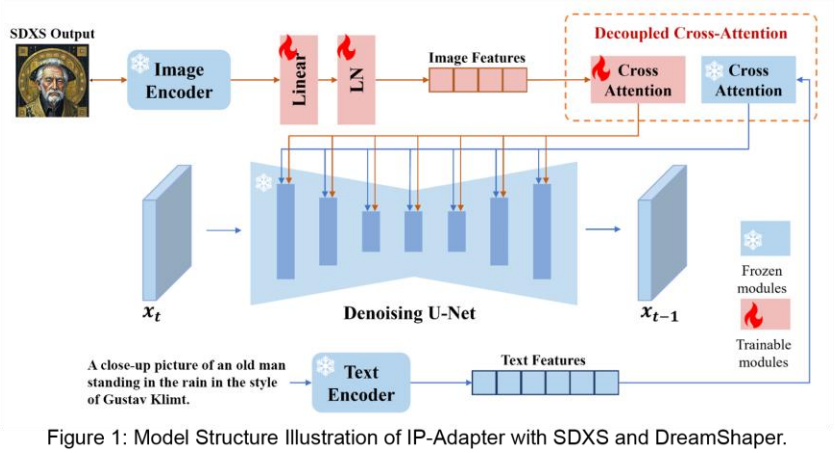


Figure 1: Model Structure Illustration of IP-Adapter with SDXS and DreamShaper.

2 Experiments and Metrics Used

2.1 Experimental Design

Our inquiry focused on the output of 4 model configurations: SDXS by itself, DreamShaper by itself, Fusion v.1 which is an IP-Adapted SDXS + DreamShaper model (but with more output weighting on the features of SDXS in the adaptation), Fusion v.2 (which is the same as v.1 but with more weighting on the features of DreamShaper rather than SDXS). We focused the design of the experiment on the models' ability to generate images according to the artistic style of the famous artist Gustav Klimt. To this end, we used the same text prompt for each of the models: 'a close up image of an old man standing in the rain in the style of Gustav Klimt'. We chose this particular prompt in order to quantify the success of the image generation on 3 fundamental aspects: how diverse are the generated images, how well the style of the images resembled Gustav Klimt's, and how well did the images generations reflect the prompt—e.g. besides the artistic style, does it really succeed at portraying an old man standing in the rain. We generated 20 images from each model for our evaluations.

2.2 ComfyUI, Integration of Models, Sampling and IP-Adaptation Process

The execution of the models themselves has been primarily done in ComfyUI, which is a node-based interface for stable diffusion that enables users to build and modify image generation pipelines by connecting various nodes that represent different components such as image inputs, diffusion models and various processing effects. Our workflows for SDXS and DreamShaper were fairly standard, with the U-Net and CLIP encoders attached to KSamplers, with an empty latent image, and their respective VAE's for the decoding in the final output.

KSamplers in diffusion models are used to iteratively refine noisy images into high-quality outputs. They work by progressively denoising images through several key parameters. **Steps** define the number of iterations in the process, with more steps typically yielding better quality but increasing computation time. The **Classifier-Free Guidance (CFG)** parameter controls how strongly the generated image adheres to the input prompt, with higher values ensuring more alignment but less creativity. **Samplers** such as Euler, DDIM, or Heun determine the method of noise reduction, each offering different trade-offs in speed and quality. **Schedulers** define how noise is introduced and removed across steps, using strategies like linear or exponential decay to control the image generation trajectory. Finally, the **denoise** strength parameter influences the intensity of

noise removal at each step, balancing subtle refinement or drastic changes. These elements collectively guide the image generation process, fine-tuning the output based on the selected parameters.

Part of the innovation led by the SDXS model is its one-step process, so the steps are defined as just 1 with the denoising at 1.0 (since we are working with an empty latent image), and since it is a one-step process then the cfg, sampling and scheduling should be irrelevant too. For DreamShaper, we set the steps at 30 (and again, denoising at 1.0), cfg at 6.0, sampler is Euler and scheduling is normal. The Euler sampler uses the Euler method to iteratively reverse the diffusion process, where the Euler method approximates changes step-by-step by computing the next state based on the current state and a differential equation, efficiently generating images from noise. The normal scheduler is a linear schedule where noise is added and removed at a consistent rate during denoising (other schedulers for example may add or remove most of the noise at the beginning or end of the denoising for example).

In the IP-Adapter configuration though, we used a DPM++ 3M SDE sampler and a SGM Uniform scheduler for DreamShaper. The DPM++ 3M SDE sampler improves the reverse diffusion process by using a stochastic differential equation approach for more accurate and efficient denoising, while the SGM Uniform scheduler uniformly controls the noise schedule to maintain consistent progress throughout the denoising steps. The IP-Adapter has 2 main parameters in our context: **weight** (adjusts the influence of the IP-Adapter model on the final image, determining how strongly the adapter's guidance affects the generation, a value less than 1.0 reduces the adapters influence, while a value greater than 1.0 increases it), and **embedding combination type** (merges multiple embeddings according to some method). For both Fusion v.1 and v.2 we combined the embeddings of SDXS and DreamShaper by concatenating them, but the weight set for v.1 was 0.8 and the weight set for v.2 was 1.4—which means the IP-Adapter's contribution is 40% stronger than its default or baseline influence, weighing the SDXS embedding more than the DreamShaper, while the converse for the 0.8 weight.

2.3 Metrics

We used a variety of metrics, including a survey, to quantify the diversity of the generated images (how different are images generated by the same prompt different from each other), as well as the accuracy of the image generation (how well the generated image reflects the given prompt, and how well the images resemble the style of Gustav Klimt). Here is a breakdown of each metric used to quantify these features:

Diversity Score: we used a ResNet-50 neural network pre-trained on imagenet to embed the generated images into high-dimensional vectors that capture their essential features. The diversity score metric computes the average Euclidean distance between every pair of the feature vectors. The larger the average distance, the more diverse the generated images are, while a lower score indicates that the images are more similar to each other. We took notice to also normalize the figures so an appropriate distance comparison can be made across the images and models.

Inception Score: a metric used to assess the quality and diversity of images generated by models. It uses the predictions of a pre-trained Inception model to assess how well the generated images can be classified into different distinct categories (image diversity), and by checking if the generated images belong confidently to one specific class (image quality). A higher inception score is designed to represent higher image diversity and higher image quality.

Multi-Scale Structural Similarity Index Measure (MS-SSIM): evaluates the perceptual similarity between two images by comparing structural information like luminance, contrast, and texture at multiple resolutions. It extends the SSIM, which focuses on how well structures align between two images, by applying this comparison across different scales (resolutions) to capture both fine and broad details. A score close to 1 indicates high similarity, while lower scores closer to 0 suggest greater diversity.

Fréchet Inception Distance (FID): a score is a metric used to evaluate the quality of images generated by generative models by comparing the distribution of the generated images to that of real images. It calculates the distance between feature representations of the two image sets (real and generated) using the Inception V3 network. Specifically, the FID score computes the Fréchet distance between the multivariate Gaussian distributions of the two sets, based on their mean and covariance. Mathematically, it is given by the formula:

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu-\mu'\|_2^2 + \mathrm{tr}\left(\Sigma+\Sigma'-2(\Sigma\Sigma')^{\frac{1}{2}}\right)$$

μ, μ' are the means of the distributions of the feature vectors of the real and generated images respectively, with Σ, Σ being their covariances. Lower FID scores indicate that the generated images are closer to the real images in terms of visual quality. For our experiment, the real images used in the comparison were 21 paintings by Gustav Klimt, so in this context, a lower FID score indicates that the generated images more successfully resembled Klimt's works.

Survey: we constructed a survey asking respondents questions that relate to each feature we intended on evaluating. The survey starts by showing the participants four original artworks by Gustav Klimt. Then, we showed them 4 groups of pictures which were image generations from our 4 models in the experiment design. Subsequently, we asked the participants to rank the diversity of images for each of the four groups from 1 ("not at all diverse") to 5 ("very diverse"). Furthermore, we asked the participants to choose which of the 4 groups most closely matches or reflects the art style of Gustav Klimt. Finally, we revealed the original text prompt to the participants, and we asked them which of the four groups most accurately reflects the given prompt. A link to the survey can be found here.

Figure 2: four example artworks by Gustav Klimt.

Text Prompt: A close up picture of an old man standing in the rain in the style of Gustav Klimt.

Figure 3: image outputs of the four models. Group 1: SDXS, Group 2: DreamShaper, Group 3: Fusion v.1, Group 4: Fusion v.2.

3 Results

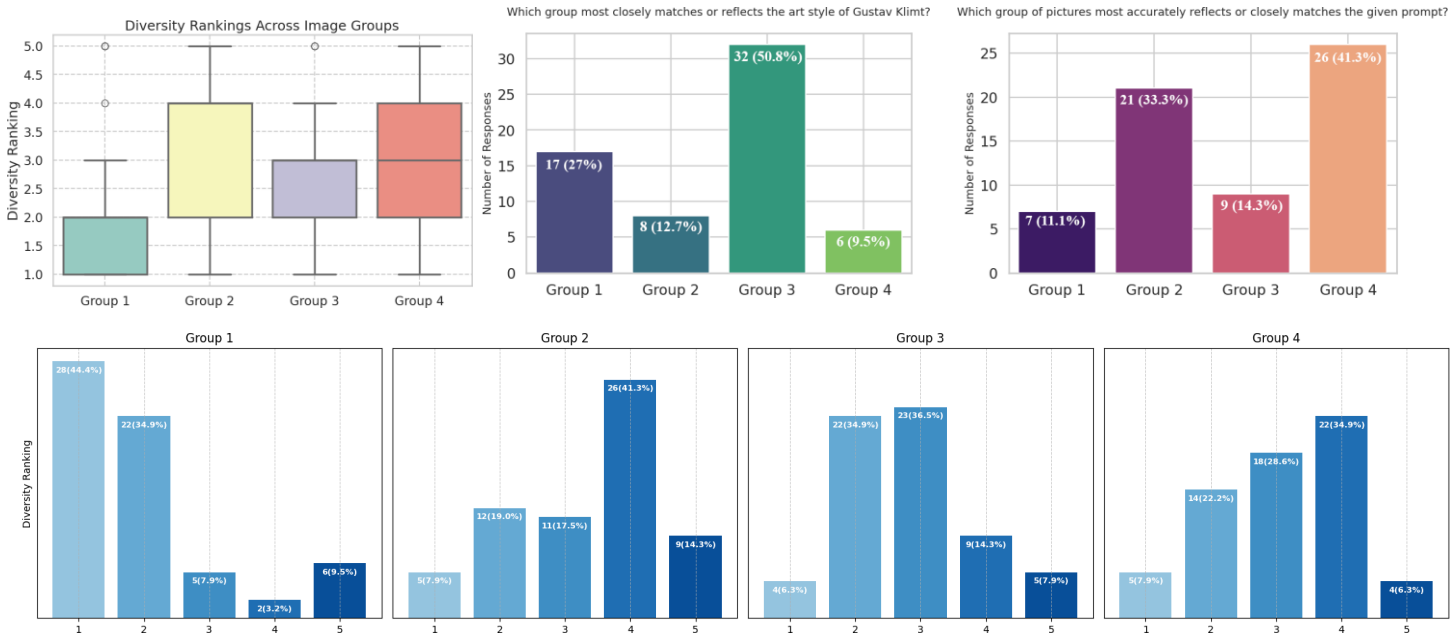
We can note some things regarding the image generations of the four models displayed in Figure 3. Firstly, it is clear that SDXS (Group 1) has the least image diversity, and lacks particular features specified by the prompt like rain, but nevertheless performs very well in replicating the art style of Gustav Klimt. DreamShaper (Group 2) achieves relatively much higher diversity in the images, showing seemingly different people in different attire and compositions. Additionally, DreamShaper is able to accurately involve elements related to rain, like rain itself, a blurred and wet background, winter attire, and umbrellas. However, it fails in replicating the art style of Gustav Klimt, despite incorporating gold elements, making the images look more like photographs than Gustav Klimt's artworks. Fusion v.1 (Group 3) does not have much diversity, but certainly more than SDXS, and incorporates different color and compositional elements in the style and background, despite the old man looking like the same person across all the images. This makes sense since despite being adapted alongside DreamShaper, SDXS is being weighted more heavily in the image generation. Finally, Fusion v.2 (Group 4) maintains a much higher diversity compared to SDXS as well as Fusion v.1, as well as incorporating rain-related elements that are missed in SDXS and Fusion v.1 too. Again, this makes sense since, contrarily, Fusion v.2 weighs the DreamShaper more than SDXS in the generation, so it is able to incorporate rainy elements, as well as Gustav Klimt's art style, and emphasize DreamShaper's image diversity. However, its reflection of Gustav Klimt's art style might be slightly behind Fusion v.1 and SDXS in the

explicit sense (and this is obvious since Gustav Klimt doesn't have any artworks of closeups of old men standing in the rain, nevertheless Fusion v.2 performs well and applying this prompt). Fusion v.2 is also able to reflect Gustav Klimt's art style faithfully unlike DreamShaper.

Below is a table of the results for the metrics but for each of the four models specified by the design of the primary experiment. The best figures are highlighted in green, while the second-best in yellow.

Model	Diversity Score	Inception Score	MS-SSIM	FID
SDXS (Group 1)	0.102	1.048	0.231	355.488
DreamShaper (Group 2)	0.196	1.137	0.187	374.84
Fusion v.1 (Group 3)	0.187	1.145	0.135	325.392
Fusion v.2 (Group 4)	0.200	1.139	0.152	353.668

Below are some plots regarding the results of the survey, 63 respondents took part.



4 Discussion and Conclusion

4.1 Quantitative Metrics, Discussion

Diversity: we can see that the diversity score was by far the lowest for the SDXS model, while it was the highest for the Fusion v.2 model. DreamShaper interestingly comes in a close second, and Fusion v.1 in third.

Inception Score: we can see that Fusion v.1 model got the highest score, suggesting that it has the most diversity, with Fusion v.2 in a close second, suggesting that they're both high in diversity when compared to the other two models. DreamShaper comes in a very close third, and SDXS far behind.

MS-SSIM: a higher MS-SSIM score suggests that the images are very similar to each other, so again, SDXS performs most poorly in this regard. Fusion v.1 is again the best, followed by v.2, and DreamShaper in third.

FID: we can say the images generated by Fusion v.1 are most similar in terms of style to Gustav Klimt's artwork since it has the lowest score. Fusion v.2 in second, and SDXS in a close third. This makes sense since DreamShaper did not replicate the style of Gustav Klimt at all, and displayed more realistic, photograph-like images. However, if we were to use images of old

men standing in the rain (with any art style put aside) as the real distribution, DreamShaper would expectedly perform the best in terms of FID, since SDXS and Fusion v.1 have no rain-related attributes illustrated in the image generations.

Based on the scores, we could claim that Fusion v.1 was overall the most successful in generating the most diverse images and the best reflected Gustav Klimt's Art style. However, there is more to be said in note with the surveys.

4.2 Diversity - Survey, Discussion

For Group 1, 79.3% of the participants voted either 1 or 2 in terms of diversity. This means for the SDSX model, most people agree that the generated images displayed little to no diversity. In essence, the participants agree with all of our diversity metrics. On the other hand, 55.6% of the participants voted either 4 or 5 regarding image diversity for Group 2, so the majority of people agreed DreamShaper had a lot of diversity in the generated pictures. For the Fusion v.1 model (Group 3) the majority (71.4%) of participants voted either 2 or 3. In other words, most of the participants thought that the diversity was better than the SDXS model, but not as good as DreamShaper. Finally, for Group 4, 41.2% voted either 4 or 5, 28.6% voted 3, and 30.1% voted either 1 or 2. This suggests that according to the participants, the image diversity was higher for the Fusion v.2 model, essentially agreeing most with the diversity score (rather than the inception score and MS-SSIM).

Overall, it seems like the human perception of diversity is most like the diversity score metric. and that Fusion v.1 and Fusion v.2 both achieved higher diversity than the SDXS, with Fusion v.2 coming close but nevertheless slightly behind on the image diversity achieved by DreamShaper.

4.3 Accuracy - Survey, Discussion

Most people (41.3%) thought that the Fusion v.2 model (Group 4) was most accurate in representing the prompt, in second place was the DreamShaper (33.3%). This can be understood since we can see rain, umbrellas, winter attire and features in the generated images of DreamShaper and Fusion v.2 which is all lacking in the images of Fusion v.1 (despite it achieving Gustav Klimt's art style, presumably since SDXS doesn't incorporate any rain-themed features). More people voted for the Fusion v.2 model because it also had the style of Gustav Klimt well-represented unlike that of DreamShaper.

In Summary, it seems like our updated models have done well in integrating the style of Gustav Klimt and accuracy in depicting the scene described by the given text prompt (a close up picture of an old man standing in the rain) according to both the Survey and the calculated metrics. Perhaps doing a more moderate weighting between SDXS and DreamShaper may yield even more successful generations in capturing both image diversity, style, and prompt-accuracy given the trade-off illustrated between Fusion v.1 and Fusion v.2's incorporation of SDXS and DreamShaper.

Conclusion

An alternative in our use-case would be to use images of paintings themselves to be used for the IP-Adapter as the image-prompt, replacing the need for the SDXS, which will also probably yield image generation that captures artistic styles even more accurately. However, using SDXS in this way enables us to involve such painting styles automatically without the need of additional data—i.e. the images of the paintings—and the cost of the time needed to acquire that additional data. The metrics and survey conducted also illustrate how our approach not only increased image diversity of SDXS, but also the accuracy (how closely the generated image reflects the prompt) and the quality of the images with respect to both SDXS and DreamShaper. Naturally, this approach comes at the cost of an increase in the latency (time taken to generate the image), but this is inevitable if image quality and diversity are to be improved and considering the improvement in the performance, this trade-off proves to be beneficial. However, further steps can be taken to decrease the current latency by choosing models that perform better than DreamShaper in this sense, and in ways which nevertheless improves or maintains the image diversity and accuracy of its image generation.

References

1. Song, Yuda, Zehao Sun, and Xuanwu Yin. *SDXS: Real-Time One-Step Latent Diffusion Models with Image Conditions*. 17 Apr. 2024.
2. Ye, Hu, Jun Zhang*, Sibo Liu, Xiao Han, and Wei Yang. *IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models*. 13 Aug. 2023.
3. Alayrac, Jean-Baptiste, et al. *Flamingo: A Visual Language Model for Few-Shot Learning*. 15 Nov. 2022.
4. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z: *Rethinking the Inception architecture for computer vision*. 25 Mar. 2015.

Supplementary Materials and Code

Google Drive: https://drive.google.com/drive/u/0/folders/1DRs_rl40iD2U_YDawFTRFLYaHWOs_tmL
GitHub: https://github.com/YoelGraumann/GustavKlimtFinalProject

Resources

DreamShaper Checkpoints: https://civitai.com/models/112902/dreamshaper-xl
SDXS Checkpoints: https://huggingface.co/IDKiro/sdxs-512-dreamshaper/tree/main
IP-Adapter Nodes and Checkpoints: https://github.com/cubiq/ComfyUI_IPAdapter_plus
Pre-trained ResNet-50: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html
Inception V3: https://pytorch.org/hub/pytorch_vision_inception_v3/

Survey Link

https://docs.google.com/forms/d/e/1FAIpQLSdphtVQbrmLRO9IZ30d0nQG37jmioSNWQk-PEXtvbg9N5EFYw/viewform?usp=sf_link