Algerian Journal of Research and Technology
https://doi.org/10.58681/ajrt.24080203
A.J.R.T Volume 8 (N°2) (2024) 28 - 47

Infographics Assisted by Generative Models

N. Belemou, A.Ould Hocine

University of Science and Technology of Oran -Mohamed Boudiaf-, Dept. of Computer Science and Mathematics

**Corresponding author.*nedjwabel31@gmail.com

Received. July 07, 2024. Accepted. September 20, 2024. Published. December 20, 2024.

DOI: https://doi.org/10.58681/ajrt.24080203

Abstract. However, an extensive search for a suitable logo generator yielded limited results. To address this, a stable diffusion generative model was adapted using LoRA to generate logo images based on textual input, demonstrating successful outcomes. Multiple LoRAs were created and merged with the stable diffusion model to produce coherent and contextually relevant logos. This approach is accessible to users without coding experience through a web interface available on the Hugging Face platform.

Keywords. Artificial intelligence, Generative AI, Stable Diffusion, LoRA, Kohya_ss, Dream booth, Infographic, Logo.

INTRODUCTION

Generative Artificial Intelligence, known as Generative AI or GenAI for short, represents a notable development in the field of Artificial Intelligence (AI), especially in the area of machine learning.

Fundamentally, the goal of GenAI is to enable computers to produce novel, realistic or creative data instances that are identical to those in the training set. In contrast to traditional AI systems, which are mostly concerned with tasks related to recognition and classification, generative models gave the capacity to generate original material in a variety of media such as text, image, audio, 3D and more.

This ability to create new content has significant implications on various industries such as art, robotics, entertainment and healthcare.

Modern generative models have achieved unparalleled level in realism and diversity in their generated content, which helped in many domains after applying said models. For example, generative AI enables image generation, super-resolution, style transfer and image-to-image translation in the field of computer vision, and in natural language processing, it makes text generation, translation, summarization, and dialogue systems possible. Furthermore, generative AI can be even used in the creation of artistic and creative content, video games,

as well as in drug research where it helps in the generation of unique molecular structures with desired features.
One of the notable applications is in the creation of infographics, where generative AI can quickly produce visually compelling and informative graphics, such as logos, fliers, posters... Enhancing the way information is presented and understood.
With the current advancement in artificial intelligent, and how easily accessible it is, more and more contributions are made towards the goal of enhancing generative AI models performances in the infographic field. Where it plays a major factor in helping with the creativity and diversity of the designs.
Working on posters, logos, flyers… can prove to be a very hard task. It is a very time-consuming process as it is costly and demands creativity and variety. Knowing what the traditional way of making infographic content is without doubt very challenging, It was attempted to make it easier by using generative AI to assist in providing first drafts or even inspiration. To achieve this, a generative AI model was created using Stable Diffusion and LoRA.
This project represents a proposed contribution to the infographic field using generative AI. To ensure accessibility for anyone interested, the finalized work has been published on Hugging Face, which provides a platform called Model Hub for sharing pre-trained models.

GENERATIVE AI

Generative AI, short for generative artificial intelligence, is a field of AI that creates entirely new data, taking what it knows about a particular type of data and using that knowledge to make or generate new data. The data can vary in forms like text, images, audio, code, etc. It works in mainly three steps (Brynjolfsson et al., 2023): Data training which uses an extensive and large database. Rule learning, as it starts analyzing the data it begins understanding the rules and structures that characterize that type of data. Last but not least it generates new examples by applying the learned patterns and connections.
The most common generative AI techniques are Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Transformer networks.
Gen AI comes in many different types, such as text-to-text like the GPT series, T5, CTRL… Text-to-image like DALL-E, DRAW… ect. Image-to image such as DDColor, and image style transfer.Text-to-speech like SpeechT5.Text-to-video like VideoGPT, and Text-to-video-ms-1.7b.
Gen AI models can be accessed by any interested user without the need to have a coding background if the model is simple enough. Using gen AI to generate infographic content makes the outcome work (i.e. our model) easy for anyone to use and come up with decent results without the need to be good at coding or having a designing background.

STABLE DIFFUSION

Stable Diffusion is a deep learning, text-to image latent diffusion model capable of generating high quality photo-realistic images based on textual description that's considered as the text input (Rombach et al., 2021). Though its primary use is to generate images conditioned on textual prompts, it can also be used for in painting, out-painting, and generation image to-image translation aided by a text prompt. After its release in 2022, Stability AI continued its support by computational donations and training data to Comp Vis Group at Ludwig Maximilian University of Munich and Runway ML.
Stable Diffusion uses deep learning to achieve its effectiveness in text-to-image generation, thus it is based on deep neural network architecture, trained on a massive corpus of images linked with a textual description. This training dataset enables the model to establish

complex mappings between textual descriptions and visual attributes they represent. Stable Diffusion progressively refines its ability to translate text prompts into images by analyzing the relations between the image and its textual description in the training dataset.

Stable Diffusion Architecture

Stable Diffusion can be thought of as a latent variable model which the word “latent” refers to hidden continuous feature space thus it employs a latent diffusion model (LDM) architecture (Zhang et al., 2023).

Diffusion models are trained with the goal of removing layers of Gaussian noise applied to training images. You can think of this as a series of denoising auto encoders. The primary architectural components of Stable Diffusion include a Variation auto encoder, a noise predictor (U-Net), and an optional text encoder.

How it works

1. User Input and Text Encoding

- The user provides a text prompt describing the desired image (for example: "an astronaut riding a horse").
- The text encoder transforms this prompt into a latent representation, a condensed version capturing the prompt's semantic meaning.
- The Frozen CLIP model, which is pre-trained to understand connections between text and images, plays a subtle but crucial role. It likely helps bridge the gap between the encoded text prompt and the latent image representation during the diffusion process.

2. Starting point: The latent Seed

The process starts with a Latent Seed, essentially a random noise vector in the latent space. This initial "canvas" provides the groundwork for the image to form.

3. Iterative diffusion with guidance

This step involves the Scheduler Algorithm and the Conditioned Latent U-Net working together to manipulate noise:

Scheduler: Controls how much noise is added or removed at each step during the diffusion process.

Diffusion Cycle:

- Noise Injection (Forward Diffusion): Forward diffusion gradually adds Gaussian noise to the compressed image using Markov chains, until all that remains is random noise so that the resulting image becomes unidentifiable. During training, all images go through this process one by one.
- Noise Prediction (Conditioned Latent U-Net): The U-Net takes the current noisy latent representation and the text embedding (from the Text Encoder) as input. It predicts the amount of noise present in that specific step.
- Noise Removal (Reversed Diffusion): Based on the scheduler's instructions and the U-Net's prediction, the model removes a calculated amount of noise from the latent representation.

4. Decoding the refined latent space

- After numerous diffusion steps, the Variation Auto encoder (VAE)’s Decoder takes the final denoised latent representation of the image.
- This decoder translates the refined latent space back into a high-resolution image, depending on the model.

5. The final image

- Once the diffusion cycle completes a set number of times, after its iterations of noise addition and removal determined by the scheduler, the model generates the final output image. This image visually represents the concept conveyed in your text prompt.

In summary, Stable Diffusion combines user provided text prompts, a controlled diffusion process with noise manipulation, and a pre-trained CLIP model to transform textual descriptions into captivating images.

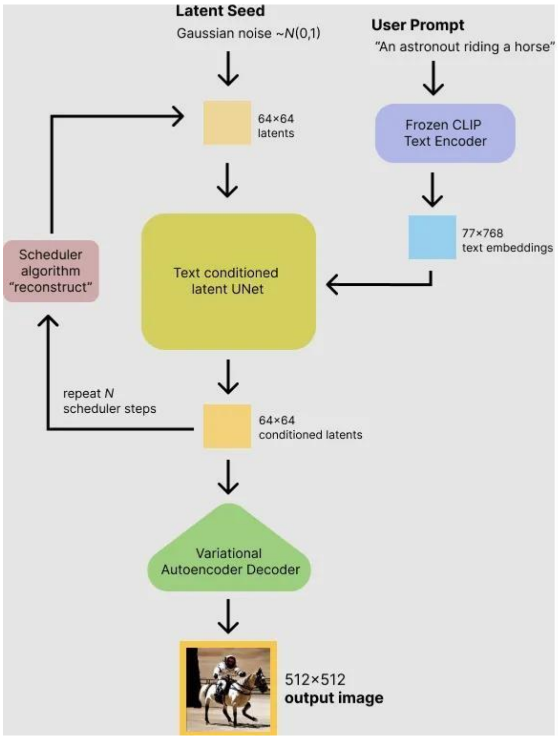


Fig. 1. Stable diffusion inference process (Mishra, 2023).

Stable Diffusion models

Stable Diffusion model has undergone continuous developments since it gained its recognition in 2022 for its ability to generate high-quality images from textual prompts. This resulted in various characteristics and uses for its variety of models.

Stable Diffusion 1.x series were the first generation of Stable Diffusion models developed by CompVis starting from Stable Diffusion v1.1 model till Stable Diffusion v1.5 model. All these models operate using 512x512 image resolutions.

The Stable-Diffusion-v1-5 checkpoint (CompVis, 2023), developed by runwayml, was initialized with the weights of the Stable-Diffusion-v1-2 checkpoint. It was then fine-tuned over 595,000 steps at a resolution of 512x512 on the "laion-aesthetics v2 5+" dataset. Additionally, 10% of the text conditioning was dropped to improve classifier-free guidance sampling.

Why Stable Diffusion

Stable Diffusion was chosen specifically because it is a versatile model with different dimensions and parameters that can be used for different purposes. It excels at generating high detailed images and even modifies existing images or enhance the quality all based on textual prompts. Stable Diffusion is also open source which made it easy to access to and implement our ideas on.

MODEL ADAPTATION METHODS

Model adaptation methods are techniques used on a pre-trained model to modify and optimize them to enhance their performance on a specific task or on a new environment.

Fine-tuning is a method which requires the model to be trained on a new dataset with smaller learning rate, transfer learning.

Dream booth

Dream Booth is a training method that introduces the approach of "personalization" of text-to-image diffusion models, it specializes the images to the user's needs by fine-tuning them. It updates the entire diffusion model using only a handful of images depicting a particular subject or style (Ruiz et al., 2023). It operates by linking a unique keyword in the prompt to the example images, all while ensuring fidelity to the input images and providing new context.

LoRA

LoRA, an abbreviation for Low Rank Adaptation, is a lightweight training technique designed to significantly reduce the number of trainable parameters. It makes the training easier and less time and memory consuming by inserting a smaller number of weights into the model, later the model only trains these inserted weights. This results in producing smaller model weights of only a few hundred MBs, which makes it easier to share and store. Put in other words, simply LoRAs are smaller files (from 1MB to 200MB) that introduce new concepts to an existing Stable Diffusion checkpoint model that it gets combined with.
LoRA can be trained using less than 25 training images for subjects, and around 100 images for style, each image must be attached to a text file that describes the images thoroughly, and it's recommended to have fewer images of good quality than many images with low resolution. In Addition to training images, it is recommended to use regularization images to avoid over fitting. The regularization images need to be in the same class as the subject/object/style that the model is trained on. LoRA is supported for DreamBooth, Kendinsky2.2, Stable Diffusion XL, text-to-image, and Wuerstchen[1].

EXPERIMENTS

Stable diffusion model fine-tuned with Dream Booth

Using Stable Diffusion v1.5 model as a base model checkpoint, and Dream Booth technique on Automatic1111 and our training dataset, the creation of a logo AI generator model was achieved. With over 30 hours of training, with 35 images, and 150 steps for each image, over 5000 iterations, with an average speed of 25 s/it.
The model was finally tested on Automatic1111, and logos corresponding to textual prompts were generated. Below are some of the results obtained:
Attempt 1: The prompt used to generate these logo was "round logo of a hiking group, nature colors, hiker, trees, mountains, river", as can be seen in Figure 2, the general idea is conveyed that this logo is intended for a hiking group; however, it appears incoherent due to a messy background, and the shape of the logo is not solid, as some of the elements are completely out of proportion. The latent diffusion model v1-5 pretrained on an image-caption pair dataset was used, personalized with Dream Booth. These results represent logos of a hiking group.

---

[1] Hugging Face. (n.d.). *LoRA.* Hugging Face. https://huggingface.co/docs/diffusers/en/training/LoRA3

Fig. 2. Generating visual outcomes from textual input.

<u>Attempt 2:</u> The prompt used to generate these logos was "logo of a coffee shop, cup of coffee, hot, minimalist, unique background color", The coffee logo results are notably better than the hiking logo results. The shapes are visibly better, and the concept is easily perceived and understandable with a unique background and a round shape. Connections can easily be made between these logos and coffee elements such as coffee seeds, steam, coffee mugs, etc.

Fig. 3. Generating visual outcomes from textual prompts.

The latent diffusion model v1-5, pretrained on an image-caption pair dataset, was used, personalized with Dream Booth. These results are generated logos of a coffee shop.
<u>Attempt 3: After training the model for 10 extra hours :</u> A second attempt was made at generating a hiking group logo using the same prompt, with the addition of a 'round, unique background color.' Significant improvements were observed in terms of the logo's coherence.

Fig. 4. Generating visual outcomes from user prompts.

Again, the latent diffusion model v1-5, pretrained on an image-caption pair dataset, was used and personalized with Dream Booth. These results represent generated logos for a coffee shop, but the model was trained for an additional 10 hours.

Advantages

- Creative and diverse designs.
- The logos are easily conceived and adapted to human designs.
- The user can get personalized logos tailored according to their provided prompt.
- The model can provide various logos in a short period of time.
- The logos obtained are of high quality due to the high quality of the training dataset.

Limitations

- Occasional lack of the logo coherency making it less professional.
- Occasional background complexity.
- The logos sometimes turn out more complex than needed.
- Incoherent writings in some logos might appear.

Stable Diffusion model fine-tuned with Kohya_ss (LoRA)

Configuring Training Parameters and Hyper parameter Insights

```
!accelerate launch --num_cpu_threads_per_process=2 "./train_network.py"    \
                   --enable_bucket --min_bucket_reso=256 --max_bucket_reso=2048         \
                   --pretrained_model_name_or_path={pretained_model_name}         \
                   --train_data_dir={projectPath}         \
                   --resolution="512,512" --output_dir={loraPath}  \
                   --network_alpha="64" --save_model_as=safetensors              \
                   --network_module=networks.lora --text_encoder_lr=5e-05 --unet_lr={learning_rate}
                   --network_dim=64 --output_name={lora_name} --lr_scheduler_num_cycles="1"      \
                   --no_half_vae --learning_rate={learning_rate} --lr_scheduler="constant"
                   --train_batch_size="3" --max_train_steps="100000" --save_every_n_epochs="99999"
                   --mixed_precision="fp16" --save_precision="fp16" --seed="1234"        \
                   --caption_extension=".txt" --cache_latents --optimizer_type="AdamW"       \
                   --max_data_loader_n_workers="1" --clip_skip=2 --bucket_reso_steps=64       \
                   --max_train_epochs={Number_of_epoches}\
                   --mem_eff_attn --xformers --bucket_no_upscale --noise_offset=0.05
```

Fig. 5. A snippet from the LoRA code indicating the configuration of the training parameters.

- **!accelerate launch -num_cpu_threads_per_process=2 "./train_network.py" :** This launches the training script using accelerate, specifying that each process can use up to 2 CPU threads.
  - **--enable_bucket:** Enables bucket resolution, a technique to handle images of different sizes efficiently.
  - **--min_bucket_reso**=256
  - **-max_bucket_reso**=2048**:** Specifies the minimum and maximum resolutions for the bucket.
  - **--resolution="512,512":** Specifies the resolution of the training images.
  - **--network_alpha="64":** Alpha value for the LoRA, hyper parameter that controls the scaling factor of the low-rank updates applied to the model's weights. A higher value of alpha means that the updates have a stronger impact, potentially leading to faster adaptation but also possibly causing more significant changes to the model's learned features. Conversely, a lower alpha reduces the impact of the updates, leading to more conservative adjustments.
  - **--network_module=networks.LoRA:** Specifies that the LoRA network module is used.
  - **--text_encoder_lr=5e-05:** Learning rate for the text encoder.
  - **--unet_lr=1e-06:** Learning rate for the U-Net.
  - **--network_dim=64:** specifies the rank of the low-rank matrices used to adapt the pre-trained model. it is a hyperparameter that defines the dimensionality of the low-rank updates and directly impacts the number of parameters that will be fine-tuned.
  - **--lr_scheduler_num_cycles=1** implicitly means the learning rate will go through one complete cycle of rising and falling over 100 epochs.
  - **--no_half_vae:** Do not use half-precision for the VAE.
  - **--lr_scheduler="constant":** Learning rate scheduler type.
  - **--train_batch_size="3":** the batch size is a critical hyperparameter that affects how the model is trained, When training our LoRA, instead of updating the model weights after every single training example (which would be very slow), They are updated after computing gradients based on 3 examples, so 1 iteration = 3 examples.
  - **--cache_latents:** Cache latent representations of the input data.

- **--optimizer_type="AdamW":** Type of optimizer to use.
- **--max_train_epochs=1:** parameter specifies the maximum number of epochs the training process will run. An epoch represents one complete pass through the entire training dataset.:
- **--xformers:** Use xformers library for efficient training.
- **--noise_offset=0.05:** Offset for noise to avoid overfitting.

LoRA_detailed_logo

The Stable Diffusion v1.5 checkpoint model was fine-tuned using Kohya_ss's scripts to create the LoRA model. With over 180 high-quality logos and very detailed prompts, 150 steps each, 27,150 iterations in total, and over 4 hours of training, on 3 A100 40GB VRAM GPUs provided by Google Colab Pro, the following results were obtained from this model:

Prompt 1: “Logo of a coffee shop, coffee mug, steam, detailed, cozy colors, and unique background color”

Fig. 6. Generating visual outcomes from textual prompts: a creative process.

The latent diffusion model v1-5, pretrained on an image-caption pair dataset, was used and fine-tuned using LoRA with the Kohya_ss technique.

Prompt 2: “Logo of a hiking group, river, mountain, sun, trees, hiker, sunset, round, simple, unique background color”.

Fig. 7. Text-to-visual transformation: four visual outcomes generated from text.

Prompt 3: “Logo of a library, set of books, dark blue background, minimalist, cozy and fun colors” .

Fig. 8. Transforming text into visuals: generating 4 images from a single prompt.

Prompt 4: “Logo of ice cream, ice cream cone, fun colors, unique background, simple, round” .

Fig. 9. Visualizing prompts: generating 4 unique outcomes from user text.

Advantages
- Coherent and understandable designs.
- Creative and logical results.
- The logos are easily conceived and adapted by humans
- The generated images are loyal to the prompt
- The logos obtained are of high quality due to the high quality of the training dataset.
- The user can get personalized logos tailored according to their provided prompt.

Limitations
- Some of the concepts may have repetitive results.
- The designs are too simple and basic.
- Complicated prompts risk incoherent designs or out of subject results.

The LoRA model adaptation method was continued using the Kohya_ss technique to enhance the model's diversity and creativity. To achieve this, separate datasets were required, containing images paired with captions, with the images corresponding to the specific objectives. Multiple LoRAs were created using the same approach. The first LoRA, named 'LoRA_cartoon_logo,' assists in generating creative, modern, cartoon-style logo designs. Additionally, a black and white LoRA, named 'LoRA_blacknwhite_logo,' was created to generate simple and minimalist logos in black and white with a unique gray background, aligning with current trends.

LoRA_cartoon_logo
The Stable Diffusion model v1-5 was fine-tuned using the open-source Kohya_ss code, following the same approach as previously, but with a different dataset to achieve the desired results. Over 200 high-quality logos and very detailed prompts were used, with 100 steps each, totaling 20,000 iterations and more than 3 hours of training on three A100 40GB VRAM GPUs provided by Google Colab Pro. The following are some of the results obtained from this model:
Prompt 1: "logo, simple, modern, logo of a bakery, chef, bread, wheat".

Fig. 10. Generating images from text prompts: our approach.

The latent diffusion model v1-5 is used, and the LoRA_cartoon_logo is applied.
Prompt 2: "logo, simple, modern, logo of a clothing store, suits, classic, dark background".

Fig. 11. Generating an image from textual input: our approach.

Prompt 3: "logo, simple, modern, confined, logo of a sport shop, man, athlete".

Fig. 12. Text-to-image generation: transforming prompts into visuals.

LoRA_blacknwhite_logo

The Stable Diffusion model v1-5 was fine-tuned using the Kohya_ss open-source code, similar to the previous approach, but a different dataset was used to achieve the desired results. With over 102 high-quality logos and very detailed prompts, 100 steps each, a total of 10,100 iterations, and over 3 hours of training on 3 A100 40GB VRAM GPUs provided by Google Colab Pro, the following results were obtained from this model:

Prompt 1: "logo, black and white, simple, large logo of a gym, man, and bodybuilding"

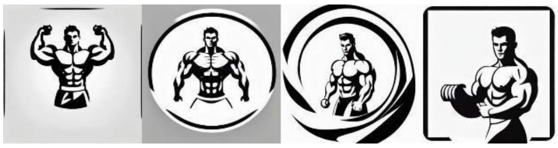


Fig. 13. Generating images from text prompts: a creative approach.

The latent diffusion model v1-5 is used, and the blacknwhite LoRA is applied.

Prompt 2: "logo, black and white, simple, round, large logo of a pastry, cake"

Fig. 14. Generating images from textual prompts: a user-driven approach.

Prompt 3: "logo, black and white, simple, round, large logo of a pet shop, dog".

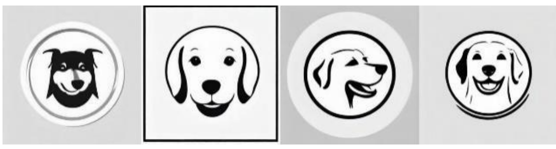


Fig. 15. Transforming text into visual art: generating images from user prompts.

ANALYZING RESULTS

The Impact of our LoRA on the Stable Diffusion model

Some results obtained from the Stable Diffusion v1-5 model with textual prompts were used, and then, using the seed of the generated images, our LoRA was applied to generate the final results. Below are some of the results obtained after applying our LoRA_detailed_logo as examples: This approach takes the user's textual input (prompt) and generates images using the latent stable diffusion model v1-5 and saves the resulting seed. It shows the generated outcome without using the LoRA. Then we take the seed and generate new images using our LoRA.

It can be observed how the images gain more coherence and correspondence to the logo concept, and how the visuals can be easily grasped by the user after the application of the LoRA.

Table 1. Generating images with Latent Stable Diffusion and LoRA Enhancement.

Prompt	**Results after using Stable Diffusion v1-5**	**Results after applying our LoRA using the same seed**
"red logo of a lion and a soccer ball".		
"A large round logo of a spaceship business, rocket, stars, planets, universe, natural colors".		
"A large logo of a coffee shop, coffee, coffee mug, steam, round, detailed".		
"A large logo of a fancy bakery, chef holding bread, close up, detailed, cozy colors, brown shades".		

Comparison between the models (DreamBooth and LoRAs)

The checkpoint model trained with Dream Booth and the LoRAs (LoRA_detailed_logo, LoRA_blacknwhite_logo, LoRA_cartoon_logo) trained with Kohya_ss will be compared based on several metrics.

Table 2. Comparison between our two generative models.

Model	**DreamBooth**	**LoRAs**
Creativity	Moderate	High
Diversity	Generic	very diverse
Coherence	Moderate	Very high
Accuracy	Moderate	High
Output Quality	High	Very high
Output Originality	Moderate	Depends on the 'LoRA_scale' parameter
Text appearance in logos	Occasional	Low
GPU	GTX1050 4GB VRAM	3 A100 40GB VRAM
Training time	40 hours (30 hours + extra 10 hours)	Average of 4 hours

The influence of descriptive keywords on AIgenerated logo designs

The text-to-image generation models rely heavily on the textual description provided by the user to generate the desired outcome. In our work, a text-to-image AI-generated logo model, by using extra keywords to describe the complexity or the shape of the generated logos such

as “minimalist/detailed” for complexity, and “round/square/ etc.” for the shape, result in major changes.
These experiments were run through using the ‘LoRA_detailed_logo’ LoRA.

Minimalist vs Detailed
After using the same prompt 'a large logo of a burger, close up, unique color in the background, realistic colors,' one version includes 'minimalist' as an additional keyword, while another includes 'detailed.' The results obtained are as follows:

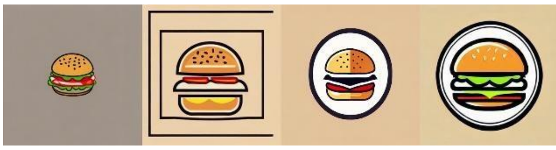


Fig. 16. Impact of 'minimalist' keyword on generated results using 'LoRA_detailed_logo' LoRA.

Fig. 17. Impact of 'detailed' keyword in prompt on generated results using ' LoRA _detailed_logo' LoRA.

It can be observed that using 'minimalist' allows the model to prioritize simplicity and abstraction, focusing on capturing the core essence or concept conveyed by the prompt while disregarding extraneous details. This approach encourages creativity and permits more open-ended interpretations, leading to outputs that reflect minimalist principles and may exhibit abstract or conceptual qualities. Conversely, using 'detailed' directs the model to pay close attention to specific nuances, intricacies, and elements within the prompt, resulting in outputs that closely align with the detailed specifications, leaving little room for ambiguity or interpretation.

Using a shape keyword vs. not specifying
After using the same prompt, 'A large round logo of a spaceship business, rocket, stars, planets, universe, and natural colors,' the first one includes 'round' as an additional keyword, while the other does not. The following are the results obtained:

Fig. 18. Exploring the impact of 'round' keyword on shape generation with ' LoRA _detailed_logo' LoRA.

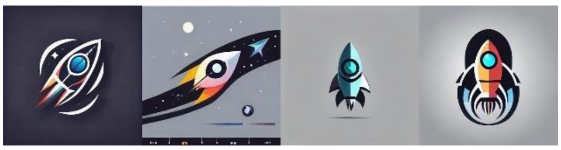


Fig. 19. Exploring the impact of prompt words on shape generation without shape keywords using ' LoRA _detailed_logo' LoRA.

Exploring the effects of parameter adjustment on our LoRA model outcomes

The ‘cfg’ (Classifier-Free Guidance) parameter

The guidance scale parameter (cfg) is a technique employed to enhance the quality and relevance of generated outputs by utilizing two modes: conditioned (with a prompt) and unconditioned (without a prompt). The following formula illustrates its effect:

$$\boxed{\text{Output}=\text{Unconditioned Output} + \text{cfg} \times (\text{Conditioned Output}-\text{Unconditioned Output})}$$

Mainly It adjusts the strength of the model's adherence to the input prompt.

```
lora_scale=0.95
cfg=7
images = model(
    prompt, num_inference_steps=ninf,
    cross_attention_kwargs={"scale": lora_scale},
    generator=torch.manual_seed(seed), guidance_scale=cfg,
    height=h, width=w, num_images_per_prompt=numimg,
).images
```

Fig.20. Screenshot of program showing the placement of 'cfg'.

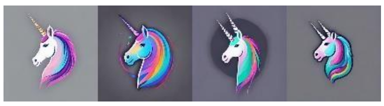
A higher guidance scale intensifies the model's focus on the prompt, leading to more precise and relevant outputs, whereas a lower scale results in more generic outputs with greater diversity. This balance allows for fine-tuning the model's performance to achieve the desired level of prompt fidelity and image quality.

Experiments with the ‘cfg’ value

Some examples of the impact of the 'cfg' parameter are presented below:

Prompt 1: “A large logo of a unicorn, close up, detailed, vibrant and fun colors”.

Table 3. Generated images of a unicorn logo using a latent stable diffusion model fine-tuned with LoRA with the ‘cfg’ parameter with different values.

“cfg” value	**The generated images**
cfg = 3	
cfg=8	
cfg=13	
cfg=20	

Prompt2: “A large round logo of a surfing shop, surfer on blue big waves, circular sunset, palms, close up, detailed, precise, natural colors” .

Table 4. Generated images of a unicorn logo using a latent stable diffusion model fine-tuned with LoRA with the ‘cfg’ parameter with different values.

“cfg” value	The generated images
cfg=3	
cfg=8	
cfg=13	
cfg=20	

Analysis of CFG Value Adjustments

It can be concluded that adjusting the CFG scale allows for fine-tuning the balance between adhering closely to the prompt and maintaining creative flexibility. High CFG values enhance adherence and detail capture but potentially reduce diversity in the generated outputs, making them less natural, while low CFG values promote creativity and diversity in the generated outputs but may deviate from the intended context. These ranges may vary based on the specific task, training dataset, and desired outcomes.

Table 5 explains the differences between high CFG and low CFG values:

Table 5. Comparison table between low and high CFG values.

Aspect	High CFG Value	Low CFG Value
Prompt Adherence	Strong adherence, captures detailed elements accurately	Looser adherence, may miss some details
Fidelity / Creativity	High fidelity, less creative variability	More creative, less tied to prompt specifics
Handling Complexity	Incorporates full range of specified elements	Captures essence, may overlook complex details
Risk of Overfitting	Risk of forced or unnatural outputs	Balanced, natural but may miss important elements

The ‘LoRA_scale’ parameter

The LoRA _scale parameter controls the degree to which the LoRA parameters influence the final output. It acts as a multiplier for the contributions of the LoRA model during the cross-attention operations in the Stable Diffusion pipeline.

```
lora_scale=0.95
cfg=7
images = model(
    prompt, num_inference_steps=ninf,
    cross_attention_kwargs={"scale": lora_scale},
    generator=torch.manual_seed(seed), guidance_scale=cfg,
    height=h, width=w, num_images_per_prompt=numimg,
).images
```

Fig. 21. Screenshot of program showing placement of ' LoRA _scale' in cross-attention mechanism.

A higher LoRA _scale increases the impact of the LoRA parameters on the image generation process. This means the images will more closely reflect the adaptations learned by the LoRA model, and the opposite is true.

Experiments ‘LoRA_scale’ parameter value

Some examples of the impact of the 'LoRA_scale' parameter are presented below.

Prompt 1: "a logo of a supermarket, shopping cart, vegetables, minimalist, unique color in the background”.

Table 6. Exploring supermarket logo variations with latent stable diffusion and LoRA tuning.

LoRA _scale value	The generated images
LoRA _scale = 0.1	
LoRA_scale =1	
LoRA_scale =1.9	

Prompt 2: "A large logo of a wedding dresses shop, detailed, close up, cozy and viding colors, soft design” .

Table 7. Supermarket logo variations based on ' LoRA _Scale' parameter in latent stable diffusion model.

LoRA_scale value	The generated images
LoRA_scale = 0.1	
LoRA_scale =1	

LoRA_scale =1.9 

Analysis of 'LoRA_scale' parameter value adjustments

It can be concluded that at lower values of the 'LoRA_scale' parameter, the LoRA weights have no significant effect, meaning the model behaves as if it had not been fine-tuned with LoRA at all. At values much higher than 1, overfitting or excessive influence from the LoRA weights can occur, potentially distorting the outputs in undesirable ways. There is no strict upper limit, but practical values typically range from 0.7 to 1.1. These ranges may vary depending on the specific task, training dataset, and desired outcomes.

WEB APPLICATION INTERFACE

A user-friendly web application for the model has been created using Gradio[2], allowing users to easily select the desired logo style from the available options. Currently, three LoRAs are available for use, trained with the kohya_ss open-source code, each utilizing different datasets to generate varied visual outcomes (images).

The LoRA developed is detailed_LoRA, which, when applied to the latent diffusion model v1-5, generates images corresponding to the user's textual prompt. Secondly, blacknwhite_LoRA was created to generate minimalist logos in black and white only, employing the same approach. Finally, cartoon_LoRA was developed, a LoRA that generates logos in a fun, modern cartoonish style, also using the same method. This web application is available via a space in Hugging Face Spaces, easily accessible and navigable to anyone who needs our work[3].

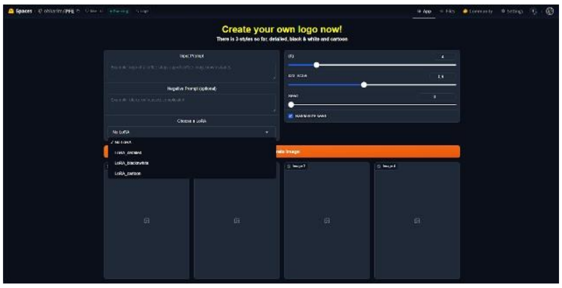


Fig.22. User-friendly interface for hugging face space built with gradio.

RELATED WORKS

Logo-LoRA

Published by: "prushton" on' hugging face' platform (artificialguybr, 2023). This LoRA has been rained using "myradeng/random100logos" dataset and the base model "stable-diffusion-v-1-5". Below are several examples for the prompt 'University logo".

This model provides detailed designs with strong symbolism, such as buildings and trees. But there are readability issues and cluttered, disjointed designs, Lack of coherent identity.

---

[2] Gradio. (n.d.). *Building interfaces*. Gradio documentation. https://www.gradio.app/guides/quickstart

[3] https://huggingface.co/spaces/ohkarim/PFE/tree/ma in

Fig.23. AI-generated logos from text input using logo- LoRA.

LoRA-logo-simple

Published by: “dbert123” on the “hugging face” platform (Huggingface, dber123, 2023). This LoRA has been trained based on StabilityAI’s stable-diffusion-xl-base-1.0” model. The following presents the results of the analysis:

Fig.23. Logo generation from user input using LoRA -logo simple.

The logos obtained by this model are incomprehensive and incoherent; all of the logos generated contain illegible text which makes the overall result unflattering.

LoRA-logo-redmond-1-5v

Published by “artificialguybr” on the “hugging face” platform (Huggingface, artificialguybr, 2023). This LoRA was fine-tuned on “Liberte Redmond SD 1.5” Model, that itself was fine-tuned on “stablediffusion-v1-5” Model. Below are the results obtained from the analysis:

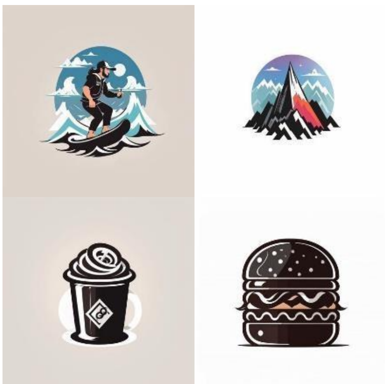


Fig. 24. Generating logos from textual input using LoRA logo-redmond-1-5v0.

The results obtained after using this model are logos of high quality, comprehensive, and free from text illegibility. However, there is an issue with the color choice. The similarities and differences between this work and other existing works in the field are analyzed.

Table 8. A Comparative Analysis of Our Work vs. Others.

Model	Our work	Logo LoRA	Logo LoRAsimple	LoRAlogo-redmon d-1-5v
Accuracy	High	Moderate	Very low	high
Understandable visuals	Yes	Moderate	No	Yes
Text appearance rate	Low	High	High	Low
Quality	High	Average	Average	High
Creativity	High	High	Low	High

CONCLUSION

Artificial intelligence is experiencing rapid development in the field of generative AI, with applications emerging in nearly every domain, such as healthcare, entertainment, and, more recently, in infographics.

By utilizing Stable Diffusion models and fine-tuning techniques, the performance of generative AI in the infographic domain can be enhanced, making tasks more creative, faster, and easier.

A web application has been developed to generate logos using textual prompts. This application incorporates a Stable Diffusion model, which generates images with three different LoRAs, each representing a distinct style. The generated images are of high quality, with the context aligning with the user's textual prompt to a certain extent, while still allowing room for AI creativity.

While the work is not yet perfect, future improvements are planned, including the addition of input formats such as photos and expanded functionalities like inpainting, image style transfer, and more. The aim is to extend the model's applicability beyond logos, making it suitable for all infographic-related tasks, including posters, flyers, web interfaces, and more.

REFERENCES

artificialguybr. (2023). *Logo.Redmond V2*. Huggingface. Retrieved from https://huggingface.co/artificialguybr/LogoRedmond-LogoLoRAForSDXL-V2

Brynjolfsson, E., Li, D., & Raymond, L. R. (2023). *Generative AI at Work* (Working Paper No. 31161). National Bureau of Economic Research. https://www.nber.org/system/files/working_papers/w31161/w31161.pdf

CompVis. (2023). *Stable Diffusion v1-5 Model*. Hugging Face. https://huggingface.co/runwayml/stablediffusion-v1-5

Gradio. (n.d.). *Building interfaces*. Gradio documentation. https://www.gradio.app/guides/quickstart

Hugging Face. (n.d.). *LoRA*. Hugging Face. https://huggingface.co/docs/diffusers/en/training/LoRA3

Huggingface, dber123. (2023). *LoRA-logo-simple*. Huggingface. https://huggingface.co/dber123/LoRA-logosimple

Huggingface, artificialguybr. (2023). *logo-redmond-1-5v-logo-LoRA-for-liberteredmond-sd-1-5*. Huggingface. https://huggingface.co/artificialguybr/logoredmond-1-5v-logo-LoRA-for-liberteredmond-sd-1-5

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. *ArXiv*, *2112.10752*. https://github.com/CompVis/stable-diffusion

Mishra, O. (2023). *Stable Diffusion Explained*. Medium. Retrieved from https://medium.com/@onkarmishra/stablediffusion-explained-1f101284484d

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2023). *DreamBooth: Fine-tuning text-to-image diffusion models for subject-driven generation*. arXiv. https://arxiv.org/abs/2208.12242

Zhang, T., Wang, Z., Huang, J., Tasnim, M. M., & Shi, W. (2023). *A survey of diffusion based image generation models: Issues and their solutions*. arXiv preprint arXiv:2308.13142. https://arxiv.org/abs/2308.13142