Final project report: Toward Better Feature Preserving in Text-to-Image Synthesis Based on DreamBooth

Zeyuan Feng[13873424]

Yinzhou Wang[38130742]

Kaiwen Hu[70127519]

ESE 5460 Deep Learning

Dec 18th, 2022

School of Engineering and Applied Science

Abstract

DreamBooth Ruiz et al. 2022 generates personalized images using few samples for tuning the original stable-diffusion model. However, the generated images might miss some features in the input image. To penalize such abnormality, we revised the DreamBooth algorithm by adding an extra loss on crucial features extracting using a categorical Variant Auto-Encoder. The new algorithm succeeded in preserving most of the features even with a weak prior prompt.

1 Introduction

Ruiz et al. 2022 proposed a diffusion-model-based approach, DreamBooth, to tackle a brand new problem, which is to synthesize personal subjects (animals, objects) in different contexts. DreamBooth aims to take in several (3-5) captured images of an instance and generate novel renditions of them in different contexts while maintaining their key features and achieving high fidelity. For example, the user could take 4 photos of a particular dog, and feed the photos to DreamBooth for training. Then DreamBooth would generate images of the dog swimming in a pool if the user inputs the text "a [v] dog swimming in a pool". Here, the "[v]" is an unique identifier either obtained from training process helps model to keep characteristics of the input dog or defined by the user.

Figure 1: Samples of the BreamBooth model.(We use this image from Ruiz et al. 2022 for demonstration purpose)

However, There are some limitations in this work. The most critical one is that DreamBooth may change some the appearance/features of the input object due to the prompted context (e.g., on top of the moon, the cat will tend change its color as shown in Figure 2).

Figure 2: Left: input image, right: DreamBooth Output

Our goal for this project is to address this specific problem for DreamBooth

1.1 Contribution

- We trained a variational autoencoder (VAE) for animal faces.
- We proposed to use the pretrained VAE to help preserve the features of the instants under the perturbation by the prompted text for a specific category of objects.
- The feature preservation is enhanced dramatically using our approach, while the output images do not suffer from overfitting to the input instants.

2 Background

This section introduces personalized text-to-image synthesis using diffusion models.

Diffusion models

The diffusion models are applied to recover images after they are corrupted by noises. The objective is to minimize

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

where $\mathbf{x}$ is the clean image, $\mathbf{c}$ is a conditioning vector (e.g., obtained from a text prompt), $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ is a noise term and α_t, σ_t, w_t are terms that control the noise schedule and sample quality. Eq. 1 basically says removing noise added to the images given the conditioning vector obtained from text prompt.

Text-to-image generation iteratively applies diffusion model with an initial image composed of random noise with smaller and smaller $\boldsymbol{\epsilon}$, and finally construct a "clean" image conditioned by the prompt text. Essentially it regards the initial random noise image as a fully corrupted image and tries to gradually remove the noise, thereby obtained a high-fidelity image. You can refer to Balaji et al. 2022 for more details.

Figure 3: Generate a clean image from random noise with text prompt (We use this image from Balaji et al. 2022 for demonstration purpose)

Conditioning vector generation

Obviously, a conditioning vector is needed for using the diffusion model. To get the conditioning vectors, the first step is to generate an instance-specific identifier $[V]$ from the input images of the instance (e.g.,

a particular dog). Then this identifier is used together with text prompt to generate the conditioning vector **c** using language models like T5-XXL.

DreamBooth

DreamBooth grabs a pretrained diffusion model and fine tunes such pretrained low-resolution model with the input images of the instant paired with the its class and the instance-specific identifier $[V]$ (e.g., a [V] dog). The tuned model should be able to recover both various dogs' images and the user-input images from noise under the corresponding prompts (i.e., a dog for various dog, and a [V] dog for the specific dog). After that, by blending the identifier [V] with prompt text, the tuned model should be able to generate the image of that instance in suggested context. For example, a image of the specific dog in the Acropolic as shown in Figure 1. In the final stage, DreamBooth converts the low-resolution images to high-resolution and high-fidelity images using its super- Resolution components.

4 Approach

The reason of the aforementioned limitations could be that the diffusion models are less effective for providing good representations of data in their latent space according to Yang et al. 2022. Hence, we extract features of the images using a pretrained variational autoencoder during the fine-turning phase and add a corresponding loss to penalize the feature difference between input instant and the output images.

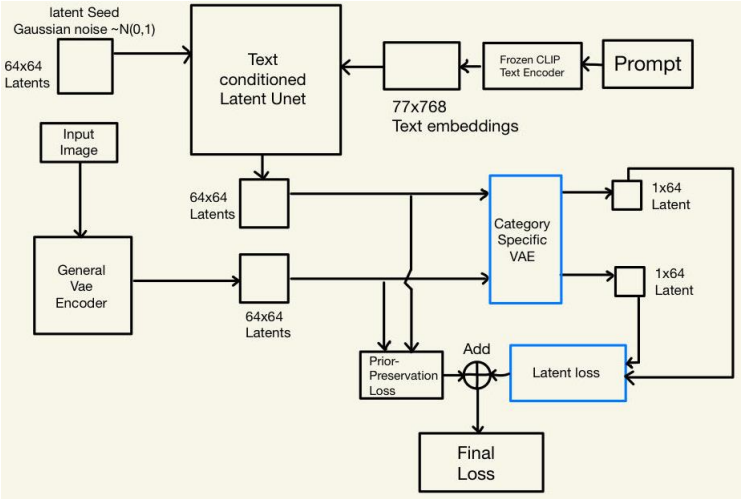

Figure 4: Loss flow we proposed

Figure 4 presents the fine-tuning (training) procedure in detail. A category-specific VAE is trained before the fine-tuning phase. In the fine-tuning phase, we first generate two 64*64 latents **z** from Gaussian noise and the input image (using a general VAE encoder), respectively. These latents should possess rough

outlines of the original object images. Then the category-specific VAE uses the latents to generate two 1*64 feature vectors $f(\mathbf{z})$. A mean squared error loss of these two features with a gain of β is added to the original Prior-Preservation Loss proposed by Ruiz et al. 2022. The new loss is given as follows

$$\mathbb{E}_{\mathbf{z},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}\left[w_t\left\|\hat{\mathbf{z}}_\theta^f\left(\alpha_t\mathbf{z}+\sigma_t\boldsymbol{\epsilon},\mathbf{c}\right)-\mathbf{z}^f\right\|_2^2+\lambda w_{t'}\left\|\hat{\mathbf{z}}_\theta^f\left(\alpha_{t'}\mathbf{z}_{\mathrm{pr}}+\sigma_{t'}\boldsymbol{\epsilon}',\mathbf{c}_{\mathrm{pr}}\right)-\mathbf{z}_{\mathrm{pr}}^f\right\|_2^2\right] \tag{2}$$

where $\hat{\mathbf{z}}_\theta^f\left(\alpha_t\mathbf{z}+\sigma_t\boldsymbol{\epsilon},\mathbf{c}\right)=\left[\hat{\mathbf{z}}_\theta\left(\alpha_t\mathbf{z}+\sigma_t\boldsymbol{\epsilon},\mathbf{c}\right)\quad\beta f(\hat{\mathbf{z}}_\theta\left(\alpha_t\mathbf{z}+\sigma_t\boldsymbol{\epsilon},\mathbf{c}\right))\right]^T$ merely augments the original predictive latent variables $\hat{\mathbf{z}}_\theta\left(\alpha_t\mathbf{z}+\sigma_t\boldsymbol{\epsilon},\mathbf{c}\right)$ with its features extracted by the autoencoder. We augments the original latent variables in the same way $\mathbf{z}^f=[\mathbf{z}\quad\beta f(\mathbf{z})]^T$.

5 Experimental Results

We coded a VAE from scratch and trained it on the Kaggle Animal Face Dataset (Choi et al. 2020). We sampled 16130 images from the dataset and converted them into 1*64*64 gray-scale images for training.

On the other hand, four cat images (shown in Figure 5) were used for the fine-tuning phase. We used the pretrained stable diffusion model v1-5 as the start point and then tuned the model with the input cat images using DreamBooth with or without proposed VAE implementation while keeping all other conditions unchanged. After obtaining the trained models, we tested different input prompts, especially those which bring perturbation on the instant's features, to evaluate the performance of our modified loss.

The quality of output images varies a lot depending on the different backgrounds. Both methods have a relatively good performance on some common backgrounds such as grassland. However, the difference became enormous when some unusual backgrounds were used. The prompt *A cat on the moon* was used and the output images of DreamBooth algorithm are shown in Figure 6. Noticeably, the features of cat did not preserved as expected. Not only the color of the output cat images abnormally diverged from the original color, but the overall physique of the cat varied a lot. Especially, when the color of the object and background become similar, the DreamBooth method can hardly distinguish them in order to generate a clear output. On the contrary, after implementing VAE in the DreamBooth algorithm, the quality of output images increased dramatically. In Figure 6, all four cat images are more clear and vivid, and are also much closer to the input cat instant. However, the modified DreamBooth method, because of the exist of VAE of the input image, shifts more weights into the objective from the input image and has relatively less weights on the background. Thus, its background is less diverge than its from original DreamBooth method. Note that β in the equation 2 can be used to tradeoff the weights distribution of objective and background.

Figure 5: Input Cat Images from *https://www.kaggle.com/datasets/andrewmvd/animal-faces*

Figure 6: The first row shows generated cat images using DreamBooth whereas the second row shows generated cat image using DreamBooth with VAE Implementation

6 Discussion

We proposed a new loss function to allow more features being preserved for DreamBooth. Weak Prior (like "on the moon") will have a dramatically smaller impact on the characteristics of objects in our new generated images. Due to the limited time, we only investigate our new method on one category and did not explore its effect on more general settings, namely all categories.

In terms of future work, the problem of inconsistent background is still exist especially when using prompt that has weak prior. If we have more time, we may explore a better loss function or more reasonable weights that will punish the wrong background more.

References

Balaji, Yogesh, Nah, Seungjun, Huang, Xun, Vahdat, Arash, Song, Jiaming, Kreis, Karsten, Aittala, Miika, Aila, Timo, Laine, Samuli, Catanzaro, Bryan, et al. (2022). "eDiffi: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers". In: *arXiv preprint arXiv:2211.01324*.

Choi, Yunjey, Uh, Youngjung, Yoo, Jaejun, and Ha, Jung-Woo (2020). "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ruiz, Nataniel, Li, Yuanzhen, Jampani, Varun, Pritch, Yael, Rubinstein, Michael, and Aberman, Kfir (2022). "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation". In: *arXiv preprint arXiv:2208.12242*.

Yang, Ling, Zhang, Zhilong, Song, Yang, Hong, Shenda, Xu, Runsheng, Zhao, Yue, Shao, Yingxia, Zhang, Wentao, Cui, Bin, and Yang, Ming-Hsuan (2022). "Diffusion models: A comprehensive survey of methods and applications". In: *arXiv preprint arXiv:2209.00796*.