A Comparison of SoTA Style Transfer Methods

Shangjun Meng (shangjun), Yihang Bi (byihang), Zimo Si(zimosi), Jirong Yang(yjrcs)

Abstract

This paper investigates textual inversion in style transfer using Latent Diffusion Models (LDM) to integrate new concepts into generative processes with minimal examples. By embedding specific style and object placeholders trained separately, we employ the Stable Diffusion 2 tokenizer to refine the model's vocabulary and enhance image generation. Our results demonstrate the method's efficiency in preserving content semantics and improving style fidelity. This study highlights textual inversion's potential to advance generative modeling in artistic applications.

1. Introduction

Textual inversion offers a promising avenue for seamlessly integrating new concepts into the generation process of diffusion models. By leveraging the prior knowledge encoded in diffusion models, textual inversion enables the incorporation of a specific concept into generated text or images with minimal input requirements. This approach circumvents the challenges associated with introducing new concepts into large-scale language models, where fine-tuning the entire model for each new concept is both computationally expensive and susceptible to catastrophic forgetting, particularly when dealing with a limited number of examples.

Traditionally, adapting large models to accommodate new concepts has proven challenging. Fine-tuning the entire model or training separate adapters may mitigate catastrophic forgetting to some extent, but these methods fall short in simultaneously handling the coexistence of newly learned concepts and existing priors.

Textual inversion, on the other hand, offers a novel approach by learning a new embedding specifically tailored to the new concept. It is built upon Latent Diffusion Models (LDM), which consist of two key components: an autoencoder and a diffusion model [2]. The autoencoder's encoder maps input x to a latent space, while the decoder reconstructs it back to the original space. In the context of textual inversion, a placeholder is set for the new concept to be learned, and the encoder's computation process is intervened to learn an embedding vector to replace the vector associated with the tokenized string. This learned embedding captures the novelties of the style image(s) and projects it to the text space, thus generating a new, specific concept that conditions the LDM more precisely than a combination of known, conventional semantic concepts. Through this intervention, textual inversion effectively "injects" the new, case-specific concept into the model's existing vocabulary.

In the first part of our project, we set two placeholders in the prompt, one for the object and one for style. We provided 4 images for each placeholder and trained them separately. During the training process, we froze all parameters of other pertained models except the token embedding of the token encoder. For the pre-trained model, we used Stable Diffusion 2 tokenizer and pipeline [2], which has good State-of-the-Art performance and adaptability. We then compared the performances of a full-textual-inversion, txt2img paradigm and a style-inversion-only, img2img paradigm.

As an extended part of our project, we attempted to apply stochastic inversion to a content image to generate an embedded, semantically meaningful noise as a starting point, a method derived from a previous study that theoretically will allow us to achieve SoTA by only training on a single guidance image (zero-shot) and preserve key content semantics beyond superficial [3].

Textual inversion in style transfer is crucial because it allows for the customization and refinement of generative models to understand and replicate specific styles or concepts using a limited set of examples. So it enhances the flexibility and utility of generative models across various domains such as art, design, and communication.

2. Background

Textual inversion as a style transfer technique was first proposed by Gal et al. [1] The method was revolutionary for the task as previous methods mainly focused on certain aspects of the reference image (e.g. colors, texture/brushstroke) but the creative paradigms (e.g. perspective, expression of shapes) are often at loss. Attempts were made to account for these attributes in a predefined "white-list", or leveraging the attention mechanism of transformers to handle long-range dependencies. Textual inversion solved the problem of representation elegantly by in-

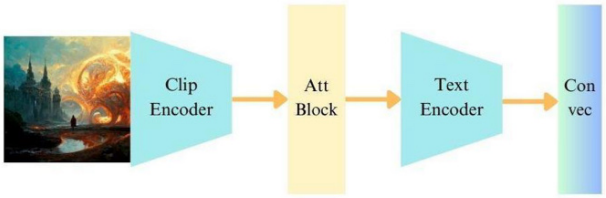

Figure 1. Textual Inversion process

versely constructing a new concept that best represents the style/object instance from existing ones, instead of attempting to decompose the image into existing concepts. Moreover, this method fully leverages the pre-trained text embeddings and significantly reduces training costs as it only tunes the feature vectors of the new placeholder tokens.

3. Methodology

3.1. Overview

In this work, we use inversion as the framework of our methods and test multiple modes of inversion-related techniques to improve transfer quality and reduce training costs.

3.2. Textual Inversion

We first pick two small sets of images for style and content reference, and assign a placeholder token for each "concept". To prevent the case where the token already exists, we wrapped our placeholders with "<>". The pre-trained Stable Diffusion v2-1 [2] pipeline components are then imported and the new tokens are inserted into the tokenizer. The text embedding tensor of the text encoder is reshaped to fit the expanded tokenizer vocabulary. We don't want the introduction of placeholders to affect the pretrained concepts and therefore froze all parameters of the variational autoencoder (VAE) and denoising UNet. With most parameters frozen, the training process becomes the simple optimization problem of grounding the text embeddings of the new concepts. For the loss function, we used mean sqaured error (MSE), which is the common practice for diffusion model training.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Once the training is completed, the resulting new embeddings are saved, and injected into the original SD pipeline [2] via the provided `add_textual_inversion` method. A prompt containing both style and content placeholders is then fed into the pipeline.

An alternative generation method we conceived is to only inject the style concept, but adopting the StableDiffusionImg2ImgPipeline. We expected this method to demonstrate more semantic attributes from the original content pictures, resulting in a "filter-like" behavior.

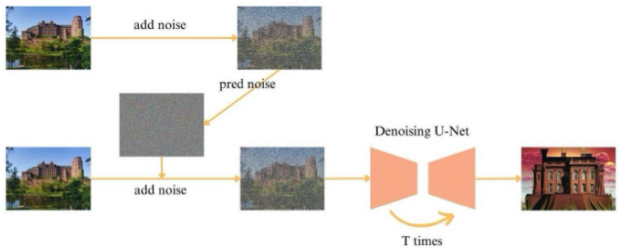

Figure 2. Stochastic Inversion&Denoising process

3.3. Stochastic Inversion

Proposed by Zhang et al. [3], an even more efficient and seamless style transfer method is to extract the notion of style by textual inversion and content by stochastic inversion. Instead of using two tokens or token-plus-content, we attach the features of the style image to the content image in latent space, maximizing the preservation of the features extracted during the style image learning process. However, the picture representation not only relies on the text description, but also on the random noise generated by the seed, and that randomness causes obvious changes in visual differences. In this case, we consider the process from image to noise as another inversion problem, attempting to add this inversion process to the original problem. Specifically, we apply random noise to the content image to obtain a blurred image(version 1), and then predict the original image and noise based on it. Subsequently, we added the predicted noise to the original image to obtain the blurred image(version 2), which was used to integrate content and style information and perform denoising together. This process can help the entire network better maintain the semantic content of the generated image and the original image.

3.4. Latent Space Diffusion

After obtaining the textual inversion, we use it as a conditional vector to control the subsequent Diffusion training process. In LDM latent space, we use the results of Stochastic Inversion as input, denoising and restoring through several Denosing U-Net Blocks under the condition vector, and get the result.

4. Results

Due to the nature of image synthesis, the qualitative data on the results are often not comparable and do not reflect the performance of the methods well. We did collect the best training loss for content and style for our two cases, shown in Tab. 1 and Fig. 5:

Content-Style	$\text{loss}_{content}$	loss_{style}
BMW-Comic	0.340	0.555
HuaweiBldg-Midjourney	0.443	0.505

Table 1. Training Losses

The statistics for both trials exhibit similar trends: higher loss for style training, which can be expected, as the content images are similar observations on one object while the similarity of style reference images resides in more abstract aspects and thus harder to fit.

Fulfillment of expectations, which is usually the critical metric for image synthesis, is most commonly measured by polling. Due to the scope of our project, we did not recruit participants to rate them. However, just by observation, we agreed that full-textual inversion generation with StableDiffusionPipeline yielded more intuitive results compared to the style-only inversion generation with Img2ImgPipeline. The desirable operation for BMW-Comic (please see Fig. 4) was a filter-like transformation that will redraw the car as if in comic book style. We initially expected the img2img model to do better, as it has a better representation of the car to start with. As it turned out, the output did learn some "comic-like" attributes, like flat color blocks and well-defined contours of shapes, but the actual gist of a car drawn in comic books is not well-represented.

In comparison, the full textual inversion trial where both concepts were injected and used a text prompt for a txt2img pipeline captured the idea better. The BMW M4 actually looks like it's from a frame of comic. Considering the modes of representation for the two processes, we believe the full textual inversion can better conjugate the style and content because they are both represented in the latent space as conditioning text vectors. The mixture of the concepts is therefore easier and the final result is more likely to land in the intuitive part of the original space.

The positioning and coloring of the car are also intriguing. We did not specify either in our prompt, and hand-picked pictures of silver-gray M4s, parked sideways to the camera. The resulting images generally include red, blue, and silver cars. The additional colors are likely from comic images, as most of them consist of large patches of red and blue. Zhang et al. also mentioned the tint of the output is best regulated by an additional learnable component [3].

For the challenging part however, we failed to get workable result. One reason is our general unfamiliarity with PyTorch Lightning, the library Stable Diffusion and InST were implemented with, and it took us a long time to understand the training and inference processes from the code. The other main reason is the lack of consistent environment on Google Colab. We were not able to control all the dependencies needed in due time for a clean run of the original code to understand what's going on, and our own implementation turned out to be ineffective ultimately (see Fig. 7). However, we are confident we have fully interpreted the implementation and given more time and local gpu RAM, can correctly train the model.

Figure 3. Top to bottom: Content, Style, generated with SDPipeline

5. Conclusion

Our results substantiate the efficacy of the artistic image-to-image and text-to-image generative model and inversion algorithms delineated in the paper [1–3] as well as in associated studies. The comparative analysis of the sets of images from distinct models can underscore the potential improvements that can be integrated into future iterations. The techniques and methodologies employed in our research hold significant promise for advancing studies.

Due to time constraints, besides the challenging part, a comprehensive reimplementation of the entire model and fine-tuning of all parameters were not feasible within this study. Future research endeavors will aim to conduct an in-depth analysis of the model, with a particular focus on investigating the specific roles and interactions of each module within the image-to-image generation framework.

References

[1] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *CVPR*, 2022. 1, 3

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image syn-

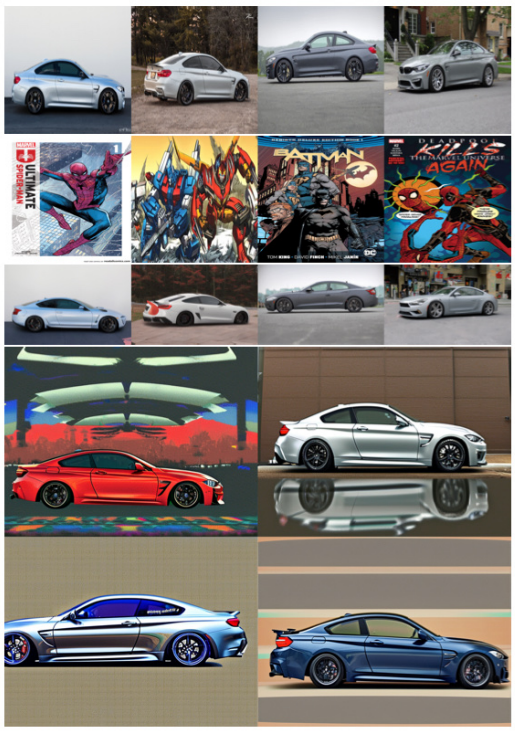

Figure 4. Top to bottom: Content, Style, generated with SD-Pipeline, generated from content image with img2imgPipeline

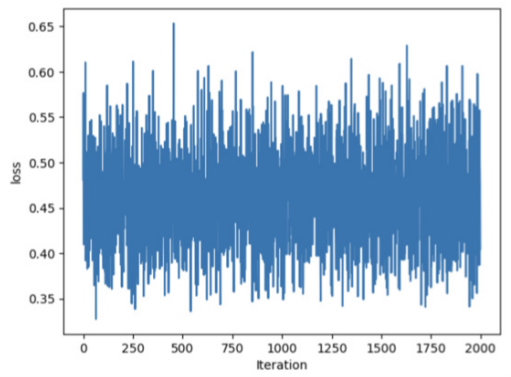

Figure 5. Loss curve of training style

thesis with latent diffusion models. pages 10684–10695, June 2022. 1, 2, 3

[3] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. *CVPR*, 2023. 1, 2, 3

6. Appendix

Code and Models are available at: https://github.com/PUzzir-Byh/styletransfer

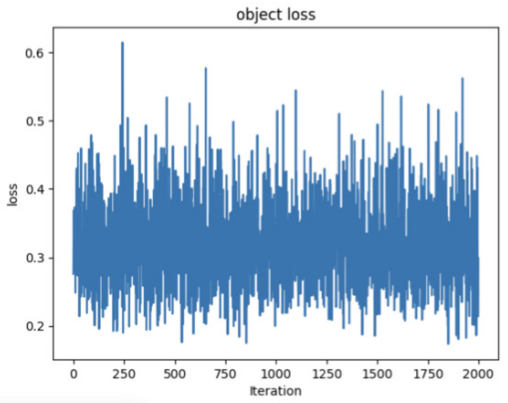

Figure 6. Loss curve of training object

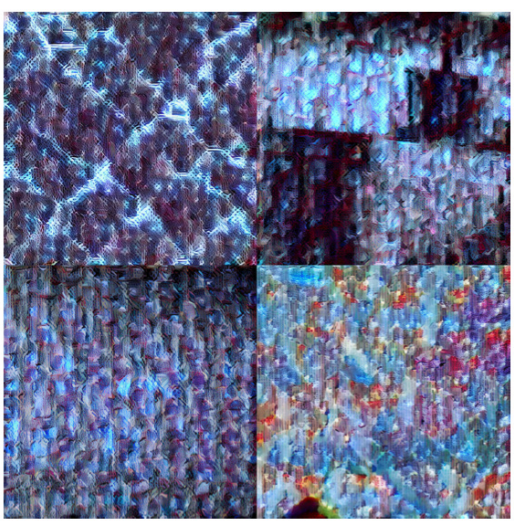

Figure 7. Wrong Results