CLIPVG: Text-Guided Image Manipulation Using Differentiable Vector Graphics

Yiren Song*[1,2], Xuning Shao*[2], Kang Chen[2], Weidong Zhang[2], Zhongliang Jing†[1], Minzhe Li[1]

[1]Shanghai Jiao Tong University, Shanghai, China,
[2]Netease Games AI Lab, Hangzhou, China,
{songyiren, liminzhe, zljing}@sjtu.edu.cn, {shaoxuning, ckn6763, zhangweidong02}@corp.netease.com

Abstract

Considerable progress has recently been made in leveraging CLIP (Contrastive Language-Image Pre-Training) models for text-guided image manipulation. However, all existing works rely on additional generative models to ensure the quality of results, because CLIP alone cannot provide enough guidance information for fine-scale pixel-level changes. In this paper, we introduce CLIPVG, a text-guided image manipulation framework using differentiable vector graphics, which is also the first CLIP-based general image manipulation framework that does not require any additional generative models. We demonstrate that CLIPVG can not only achieve state-of-art performance in both semantic correctness and synthesis quality, but also is flexible enough to support various applications far beyond the capability of all existing methods.

Introduction

Large-scale vision-language pre-training models like CLIP (Contrastive Language-Image Pre-Training) significantly facilitate the task of text-guided image manipulation, whose goal is to automatically modify images based on given text prompts. In recent years, various studies (Patashnik et al. 2021; Gal et al. 2022; Kim, Kwon, and Ye 2022; Ramesh et al. 2022; Dis 2022) have been conducted on utilizing a pre-trained CLIP model for such purposes.

However, all existing CLIP-based works perform the manipulation on pixel-level, and thus share the same intrinsic limitation of raster image based methods, i.e., easily produce poor results. This is because CLIP cannot provide enough guidance for fine-scale pixel-level optimization since CLIP mainly focuses on high-level semantics of an image. As pointed out by (Gal et al. 2022; Nichol et al. 2022), the CLIP guided optimization process may be easily trapped in local optimum or impaired by adversarial solutions.

To mitigate such an issue, existing works typically incorporate additional generative models to ensure the synthesis quality (Patashnik et al. 2021; Gal et al. 2022; Kim, Kwon, and Ye 2022; Dis 2022; Ramesh et al. 2022). These models not only consume extra resources to train, but limit the domain of the input images and text prompts. Currently,

*These authors contributed equally.
†Corresponding author

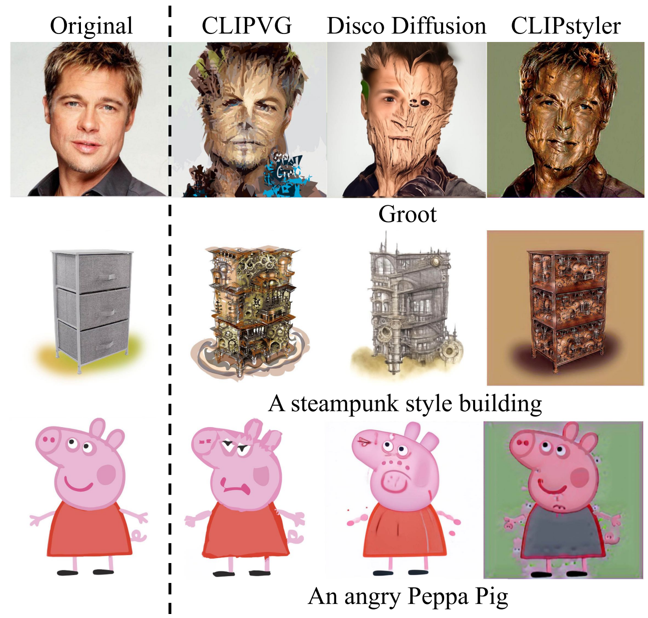


Figure 1: Text-guided manipulation results of CLIPVG and two baselines, i.e., Disco Diffusion (Dis 2022) and CLIPstyler (Kwon and Ye 2021).

the only solution that does not rely on additional generative models is CLIPstyler (Kwon and Ye 2021), which handles fine-scale features by additionally applying CLIP losses to a set of randomly sampled patches. However, this solution is only feasible for local texture style transfer, rather than general semantic manipulation (see Figure 1).

In this paper, we tackle CLIP-based image manipulation from a new perspective. Specifically, we vectorize the input raster image into vector graphics using a robust multi-round vectorization strategy and leverage a differentiable 2D vector graphics rasterizer (Li et al. 2020b) to optimize the color and shape of each geometric element (i.e., stroke or filled curve) so that the CLIP loss between the text prompt(s) and the corresponding rasterized image can be minimized. The major difference between this new framework, which we call CLIPVG, and exiting works is that CLIPVG performs image manipulation in the domain of vector graphics. Since the vector graphical elements naturally function as some kind of regularization for local shape and color, the optimization process is significantly more stable when performed on the parameters of vector graphical elements (e.g., color, line width, control points, etc.) than on pixels. We surprisingly find that the effectiveness of such regularization in CLIP-based image manipulation is almost comparable to additionally incorporating a large-scale pre-trained generative model. As illustrated in Figure 1, CLIPVG better conforms to text semantics and produces fewer visual artifacts than Disco Diffusion (Dis 2022) which relies on an additional diffusion model, i.e., (Dhariwal and Nichol 2021), trained on ImageNet (Deng et al. 2009).

Moreover, as a text-guided image manipulation framework, CLIPVG is much more flexible than existing frameworks. Firstly, CLIPVG inherits some advantages from vector graphics, i.e., it, by nature, is resolution-independent and allows separate manipulations of color and shape of each vector graphical element. Secondly, CLIPVG supports a wider range of applications (e.g., face attribute editing, character design, font design, re-colorization, etc.) since it does not bound to the domain of any specific pre-trained generative model. In other words, CLIPVG can fully unleash the capability of CLIP for image manipulation. It even allows users to assign different text prompts for different regions of the image at the same time.

The main contributions of this paper are:

- We propose the first text-guided vector graphic manipulation framework which can achieve state-of-art performance without relying on any additional pre-trained models other than CLIP.
- We design a robust multi-round vectorization strategy which enables manipulation of raster images in the domain of vector graphics.
- We implement a flexible text-guided image manipulation system that supports a variety of controls far beyond the ability of all existing methods, and the source code of this system will be made publicly available.

Related Works

Text-guided Image Manipulation. Pioneering studies (Ramesh et al. 2021; Ding et al. 2021; Li et al. 2020a; Jiang et al. 2021) model the relationship between text and image as a part of the image generation framework. Recently, the standalone CLIP models (Radford et al. 2021), pre-trained on 400M text-image pairs, have shown a state-of-the-art performance in vision-language tasks. The latest methods (Patashnik et al. 2021; Gal et al. 2022; Kim, Kwon, and Ye 2022; Dis 2022; Ramesh et al. 2022; Sun et al. 2022) typically use a CLIP model for parsing text-based guidance, and an additional generative model to constrain the output images. The generative model can be either trained on a specific category of images (domain-specific) or on a large database containing diverse image categories (domain-agnostic).

CLIP is often combined with a domain-specific StyleGAN (Karras et al. 2020) model for a human face, a cat, a church, etc. Given a text prompt, StyleCLIP (Patashnik et al. 2021) uses CLIP to find the corresponding manipulation direction in the latent space. StyleGAN-NADA (Gal et al. 2022) adapts an existing StyleGAN model to a related domain defined by the prompt. It also proposes the directional CLIP loss to mitigate the mode collapse issue (Metz et al. 2017). DiffusionCLIP replaces the GAN model with a diffusion model for image generation. These methods are generally robust, but their capabilities are tied to the domain of these pre-trained generative model.

There are also some domain-agnostic methods, most of which are developed for general-purpose image synthesis. The text-guided image synthesis methods (Ramesh et al. 2021, 2022; Dis 2022; Nichol et al. 2022; Saharia et al. 2022; Ding et al. 2021, 2022) typically support generating an image from a text prompt and a random latent code. To further enable image manipulation, some of these methods also provide an encoder to convert an input image to a corresponding latent code. For example, DALL·E-2 (Ramesh et al. 2022) and Disco Diffusion (Dis 2022) employ the diffusion process (Dhariwal and Nichol 2021; Song, Meng, and Ermon 2021) as the encoder. These methods require massive data to train the general-purposed generative model. Moreover, additional upsampling models are often required to synthesize high resolution images (Ramesh et al. 2022; Saharia et al. 2022; Ding et al. 2022).

Different from the above solutions, CLIPVG does not depend on any additional model other than CLIP. The output image is constrained by the vector graphic specific regularization rather than a generative or upsampling model.

Recently, a domain-agnostic method CLIPstyler (Kwon and Ye 2021) is proposed to eliminate the dependency on the generative model. CLIPstyler applies the CLIP loss to a set of small randomly cropped patches to stabilize the optimization. The patch level CLIP loss helps to constrain the low level details of the image, and suppress the adversarial artifacts. However, since the text prompt is applied to each small patch, the use case is limited to the low level style transfer. In contrast, we relax the patch-wise constraints and support the general semantic manipulation.

Compared to all the raster image based solutions, our vector graphic based method also has other native benefits such as the infinite resolution.

Text-guided Vector Graphic Generation. Diffvg (Li et al. 2020b) proposes a differentiable rasterizer which supports the raster image based models for the vector graphics, including the CLIP models. The differentiable rasterizer backpropagates the gradients on pixels to the continuous vector graphic parameters, such as the control points and the color. The discrete topology, e.g., the number of vector graphical elements and the connection between the control points, are not changed. We use this topology preserving property of Diffvg to regularize our optimization process.

CLIPdraw(Frans, Soros, and Witkowski 2021) combines CLIP and Diffvg for the first time, and uses CLIP to guide a set of randomly initialized strokes according to the text prompt. StyleCLIPdraw(Schaldenbrand, Liu, and Oh 2022) further controls the style of the generated vector graphic by a style image. ES-CLIP (Tian and Ha 2021) uses triangles instead of strokes as the vector graphical elements, and op-

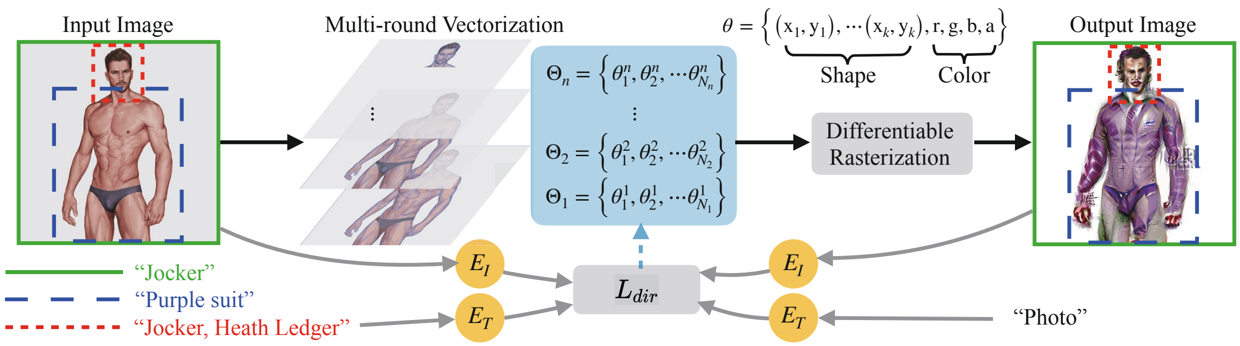


Figure 2: The overall schematics of CLIPVG. We vectorize the input image with a multi-round vectorization strategy. The optimization is guided by an ROI CLIP loss. The parameters are decoupled to enable fine-grained control.

timizes the triangles using evolution strategy.

The above methods generate the vector graphics from randomly placed vector graphical elements. In contrast, we focus on the manipulation of an existing image.

Image Vectorization. Raster image vectorization or image tracing is a well-studied problem in computer graphics. Adobe illustrator, the most advanced commercial vector graphic design tool, provides an image tracing tool (Ado 2022) with various control modes and options. By default, Adobe Image Trace (AIT) converts the raster image to a set of non-overlapping filled curves. The vectorization precision can be controlled by the number of target colors. The higher the number of target colors, the higher the precision.

Various other methods are also studied for image vectorization. Direct raster-to-vector conversion with neural networks are supported for the relatively simple images (Lopes et al. 2019; Carlier et al. 2020; Reddy et al. 2021). Stroke-based rendering can be used to fit a complex image with a sequence of vector strokes (Huang, Heng, and Zhou 2019; Zou et al. 2021), but the performance is limited by the predefined strokes. Diffvg can also be leveraged to fit an input image with a set of randomly initialized vector graphical elements. Based on Diffvg, CLIPasso (Vinker et al. 2022) controls the abstraction level or vectorization precision of the output by the number of strokes. LIVE (Ma et al. 2022) further proposes a coarse-to-fine vectorization strategy.

We adopt a decoupled approach that first vectorizes the input image, then manipulates the vectorized graphic according to the text prompts. Therefore, all of the above methods can be used in our framework. However, the further demand for image manipulation has not been considered by the existing vectorization methods. We introduce a multi-round vectorization strategy that specifically improves the robustness of image manipulation.

Method

The main schematic of CLIPVG is presented in Figure 2. We first vectorize the input raster image multiple times with different vectorization precision. All the vector graphical elements are jointly rasterized back to the pixel space by Diffvg. The rasterized image is the reconstruction of the input image at the beginning, and is iteratively optimized towards the direction of the text prompts. The gradients are derived from an ROI (Region Of Interest) CLIP loss and backpropagated to the shape and color parameters of each vector graphical elements. The optimization process is shown in Figure 3. More examples can be found in the supplementary material.

Figure 3: An example of the outputs during the iterative optimization process. the prompt is "Jocker, Heath Ledger".

Vectorization

A vector graphic is defined by a set of vector graphical elements. The parameters for each element depends on the type of the element, e.g., a filled curve can be represented as

$$\theta = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m), r, g, b, a\}, \quad (1)$$

where (x_i, y_i) are the coordinates of the i-th control point. m is the number of control points. (r, g, b, a) are RGB color and opacity values respectively. The optimization on the element is naturally regularized by the constant connection (sequence) of the control points, and the uniform texture within an element.

For an input image, the existing vectorization methods are able to generate a set of vector graphical elements which can be directly optimized by CLIPVG. However, we found several problems with this naive solution. First, although the topology within an element is preserved during optimization, there is no inter-element constraint. Two closely connected elements can be torn apart during the optimization, leaving a gap which is not always desirable. Second, the target image may require extra elements to represent the semantic of the text prompts. For example, the output of "a steampunk style building" should be much more complicated than the input of a cabinet in Figure 1. But the generation of new vector graphical elements is non-differentiable and is not supported by the optimization process.

To mitigate the above issues, we propose a multi-round vectorization strategy which takes into consideration the further need of image manipulation. We vectorize the input raster image multiple times with different vectorization precision, and derive a unique set of vector graphical elements from each round of vectorization,

$$\Theta_i = \{\theta^i_1, \theta^i_2, \ldots, \theta^i_{N_i}\}, \tag{2}$$

where N_i is the number of elements, and θ^i_j is the j-th element for the i-th round of vectorization. We increase the vectorization precision for each round, which usually results in more elements, i.e., $N_{i+1} > N_i$. We can further enhance a key region, e.g., the human face area in Figure 2, by another round of vectorization for the specific region.

We combine all the elements by placing the $(i + 1)$-th set of elements on top of the i-th set of elements. The full parameter set from n rounds of vectorization is

$$\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_n\}. \tag{3}$$

Our multi-round strategy can be used to enhance any existing vectorization method. The additional vector graphical elements allow CLIPVG to generate finer details according to the prompts. Moreover, the gap between the vector graphical elements can be filled by the redundant elements.

Loss Function

Similar to (Gal et al. 2022; Kim, Kwon, and Ye 2022; Kwon and Ye 2021), we adopt a directional CLIP loss which is defined to align the latent directions of the text and images,

$$\begin{aligned} \Delta T &= E_T\left(t_{pr}\right) - E_T\left(t_{ref}\right), \\ \Delta I &= E_I\left(I_{gen}\right) - E_I\left(I_{src}\right), \\ L_{dir}(t_{pr}, t_{ref}, I_{gen}, I_{src}) &= 1 - \tfrac{\Delta I \cdot \Delta T}{|\Delta I||\Delta T|}, \end{aligned} \tag{4}$$

where t_{pr} is the text prompt. t_{ref} is a reference text which is fixed to ”photo” in our implementation. I_{gen} is the generated image which is to be optimized. I_{src} is the source images. E_T and E_I are the text and image encoders of CLIP. ΔT and ΔI are the latent directions of the text and the images respectively. We will neglect the fixed t_{ref} and denote the loss as $L_{dir}(t_{pr}, I_{gen}, I_{src})$ in the following analysis.

We support multiple input text prompts, each associated with a specific ROI as shown in Figure 2. The ROI CLIP loss is

$$L_{roi}(\Theta, t^i_{pr}, A_i) = L_{dir}(t^i_{pr}, C_{A_i}(R(\Theta)), C_{A_i}(I_{init})), \tag{5}$$

where A_i is the area of the i-th ROI, and t^i_{pr} is the associated prompt. R is the differentiable rasterizer. $R(\Theta)$ is the rasterized image. I_{init} is the input raster image. $C_{A_i}(I)$ is an operation to crop the area A_i from the image I.

Our ROI CLIP loss in Eq. 5 is also generalized to support the random cropping enhancement in CLIPstyler (Kwon and Ye 2021). In this case, The CLIP loss is applied to a number of patches which are randomly cropped from the output image. This approach was adopted by CLIPstyler to enhance the local texture, and is further leveraged by our framework as an data augmentation method. For each ROI which is directly associated with an input text prompt, we apply the random cropping enhancement to derive a number of patches which are associated with the same prompt. We then calculate the ROI CLIP loss for each patch according to Eq. 5. The total loss of CLIPVG is

$$L_{total} = \sum_{i=1}^{h} w_i L_{roi}(\Theta, t^i_{pr}, A_i), \tag{6}$$

where w_i is the weight of CLIP loss for the i-th region. h is the total number of regions. A region can be either an ROI, or a patch which is randomly cropped from an ROI.

Optimization

The set of parameters Θ is optimized to minimize the total loss in Eq. 6 using Diffvg (Li et al. 2020b). The shape and color parameters are naturally decoupled in Eq. 1, where the shape is defined by the control points, and the color is defined by the RGB and opacity values. Therefore, we can optimize the shape and color parameters independently with two different learning rates. This is especially useful to keep either the shape or the color unchanged.

We can also edit only a subset of the vector graphical elements. The subset is usually defined by the elements which initially intersect with a certain subregion. Our framework generally allows the editable elements to move partially or fully outside the subregion during the iterative optimization process, leading to a seamless connection between the subregion and the rest of the image.

Experiments

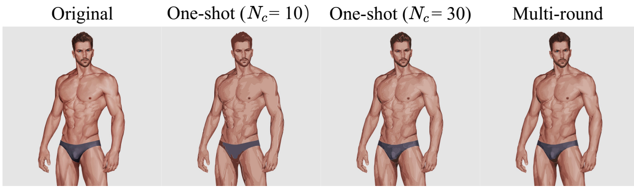
Experiment Setup

Implementation. The multi-round vectorization strategy of CLIPVG requires an arbitrary vectorization tool, e.g., AIT (Ado 2022), Diffvg (Li et al. 2020b), LIVE (Ma et al. 2022), etc. We use AIT (Ado 2022) as the default tool, since it gives the most accurate reconstruction results in our experiments. We adopt two rounds of vectorization by default. The first round is done by $N_c = 10$, and the second round is done by $N_c = 30$, where N_c is the number of target colors in AIT. We add another round of vectorization for the area of human face with $N_c = 30$.

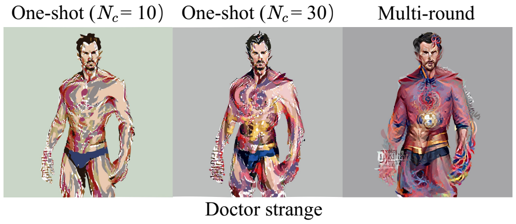
We apply random cropping to obtain $N_{patch} = 64$ patches from each ROI. The patches are randomly cropped in each iteration. The default CLIP loss weight is 30.0 for a text prompt associated ROI, and is $80.0/N_{patch}$ for each randomly cropped patch. The patch size is always set to 80% the longer edge of the ROI region, e.g., 400×400 for a 500×300 ROI, and zero-padding is adopted when necessary. Similar to CLIPstyler (Kwon and Ye 2021), we also apply the random perspective augmentation to the patches.

Similar to (Kwon and Ye 2021; Patashnik et al. 2021), we use the ViT-B/32 CLIP model (Radford et al. 2021). We employ the Adam (Kingma and Ba 2014) optimizer with a learning rate of 0.2 for the shape parameters, and 0.01 for the color parameters by default. The number of iterations is set to 150. The running time information is included in the supplementary material.

Baseline Methods. There is no existing text-guided manipulation method for the vector graphics. So we mainly

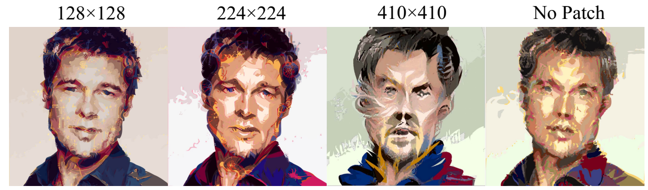


(a) The initialized vector graphics



(b) The manipulation results with different vectorization strategies.

Figure 4: Image vectorization and manipulation results with and without the multi-round strategy.



(a) CLIPVG results with different patch configurations.

(b) CLIPstyler results with different patch configurations.

Figure 5: CLIPVG and CLIPstyler results with different patch size configurations for the random cropping enhancement. The prompt is "Doctor Strange".

compare to the state-of-the-art raster image based methods. We consider two domain-agnostic baselines, Disco Diffusion v5.6 (Dis 2022) and CLIPstyler (Kwon and Ye 2021). Disco Diffusion is a popular open-source project based on a general diffusion model. CLIPstyler is a CLIP-guided style transfer method which does not rely on any generative model. We also compare CLIPVG to three domain specific methods, including StyleCLIP (Patashnik et al. 2021), StyleGAN-NADA (Gal et al. 2022) and Diffusion-CLIP (Kim, Kwon, and Ye 2022). The first two are based on the StyleGAN models, while the last one is diffusion model based. We run all the baseline methods with the official code base and the default configuration. We use images with a resolution of 512×512 as the inputs.

Ablation

Multi-Round Vectorization. We compare our multi-round vectorization strategy to the one-shot methods in Figure 4. The one-shot vectorization is done by AIT with $N_c = 10$ and $N_c = 30$ respectively. The multi-round vectorized image consists of all the elements from the one-shot cases, plus an additional set of elements for the face part of the image. It can be seen from Figure 4a that the multi-round vectorization strategy achieves the best reconstruction precision due to the availability of more vector graphical elements. After the image manipulation, some undesirable white spaces appear in the one-shot cases, since these areas are not covered by any elements. The white space issue is greatly alleviated by the multi-round strategy as shown in Figure 4b. Furthermore, the manipulated image of the multi-round case also has richer details than the one-shot cases.

Random Cropping Enhancement. We also evaluate the effect of different random cropping configurations in Figure 5. We try the patch size of 128×128, 224×224, 410×410, or no random cropping. Note that CILPstyler (Kwon and Ye 2021) has done a similar study and chosen 128×128 as the default patch size. We revisit the experiment here for two purposes. First, the text prompts used by CLIPstyler are generally related to the low level texture, e.g., "a cubism style painting". But we consider the prompts which require high-level semantic manipulation, e.g., "Doctor Strange" in Figure 5. Second, the low level texture can be constrained by the vector graphic specific regularization instead of the patch-wise loss in our case.

We show the results of CLIPVG in Figure 5a, and the results of CLIPstyler in Figure 5b as a reference. The original image is the same as the first row in Figure 1. When the patch size is relatively small, i.e., 128×128 or 224×224, We can see some obvious local artifacts with Doctor Strange's classic red-and-blue color scheme for both CLIPVG and CLIPstyler. The artifacts indicate that a small patch size is not feasible for the high-level semantic manipulation tasks.

The local artifacts are suppressed by using a large patch size or disabling the random cropping. CLIPVG with a patch size of 410×410 can effectively change the hairstyle and the identity of the face according to the prompt. CLIPVG without random cropping suffers from less accurate semantic change and blurry details, indicating that the random cropping enhancement is still beneficial for the overall quality. On the other hand, there is little meaningful semantic change for CLIPstyler even if the patch is enlarged or disabled. It only achieves limited skin color or local texture change, implying that the optimization is stuck in a local optimum due to the lack of low level constraints.

In conclusion, a large patch size can be adopted by our vector graphic based optimization process to achieve robust high-level semantic manipulation. As a result, we relax the constraints on the small patches, and select a larger default patch size than CLIPstyler, i.e., 410×410 for the region of 512×512.

Comparisons

Domain-Agnostic Methods. We compare CLIPVG to Disco Diffusion and CLIPstyler in Figure 6. CLIPVG gen-

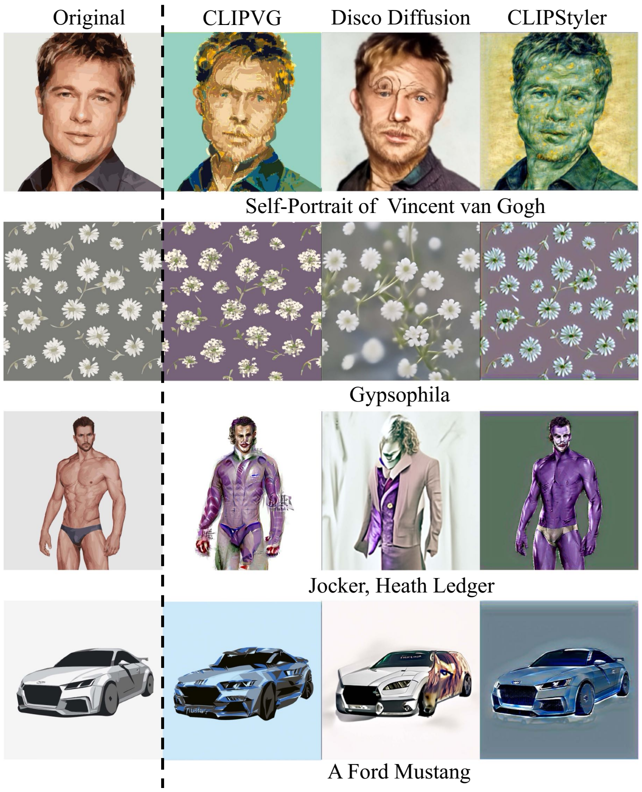


Figure 6: Comparison with the domain-agnostic methods.

erally delivers the desired semantic transfer. It is the only method which achieves both identity and texture change in the "Self-Portrait of Vincent van Gogh" case, while the other methods only address one aspect. It also manages to modify the number and shape of petals according to "Gypsophila". The results of Disco Diffusion are relatively unstable, e.g., the position of head is incorrect in the "Joker, Heath Ledger" case. It also generates a head of the Mustang Horse in the "A Ford Mustang" case, which can be taken as a local optimal solution from the pixel-level manipulation. The results indicate that domain agnostic image manipulation is a very challenging problem even with the help of a large-scale pretrained generative model. CLIPstyler is always limited to the local texture and color change due to the strict patch-wise constraints. Compared to the raster image based methods, CLIPVG can focus on the global semantic and suppress the over-editing of the local area by leveraging the vector graphic specific regularization.

Domain-Specific Methods. We compare CLIPVG to StyleCLIP, StyleGAN-NADA and DiffusionCLIP in Figure 7. Similar to the domain-agnostic case, the results of CLIPVG correctly reflect the semantics of the text prompts. StyleCLIP generally fails to match the text prompt when the desired change is out of the domain of the original StyleGAN model. StyleGAN-NADA and DiffusionCLIP can better change the semantics by finetuning the generative model according to the prompts. The semantic manipulation of CLIPVG is sometimes more thorough than StyleGAN-

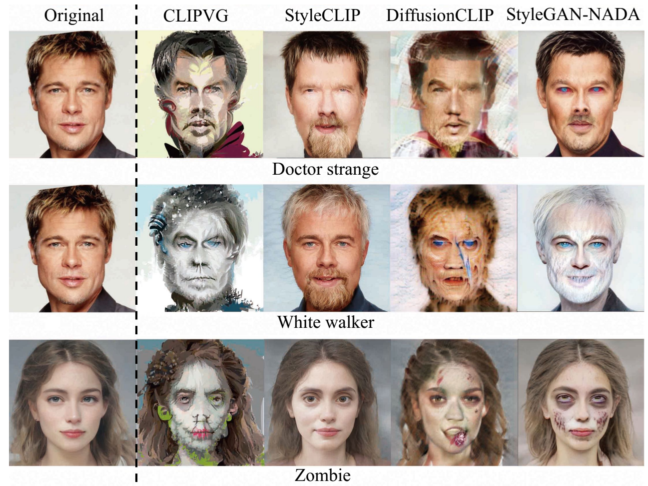


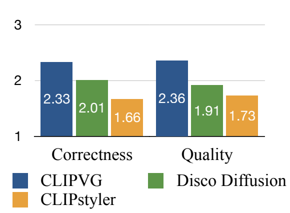
Figure 7: Comparison with the domain-specific Methods.

NADA and DiffusionCLIP, e.g., the color of cloth is changed more accurately according to the "Doctor Strange" prompt. This can be explained as StyleGAN-NADA and DiffusionCLIP tend to produce results based on some general domain knowledge learned from a set of images, not respecting each individual input image as mush as CLIPVG does.

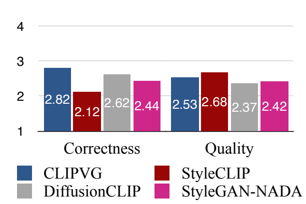
Quantitative Results. We conduct a pilot study for quantitative comparison. The users are asked to compare different methods from two perspectives, the semantic correctness, i.e., how well do the output image and the prompt fit together, and the image quality. The details can be found in the supplementary document.

The domain-agnostic and domain-specific results are shown in Figure 8a and 8b respectively. CLIPVG outperforms Disco Diffusion and CLIPstyler for both the semantic correctness and the image quality in Figure 8a, and achieves comparable performance as the state-of-the-art domain-specific methods in Figure 8b. The pilot study confirms the strong semantic manipulation capability and the robustness of CLIPVG, which come from the vector graphic specific regularization and the multi-round vectorization.

Notice that, compared with other domain-specific methods, StyleCLIP tends to produce results of higher quality but of weaker semantic connections with the given text prompts. This is because StyleCLIP strictly constrains the output to be

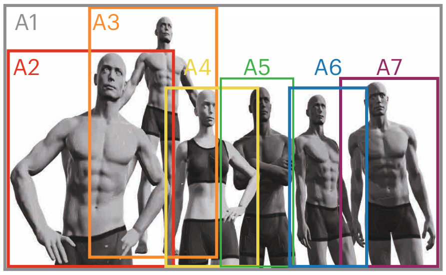


(a) Domain-agnostic.



(b) Domain-specific.

Figure 8: Average user ratings of different methods in the pilot study.

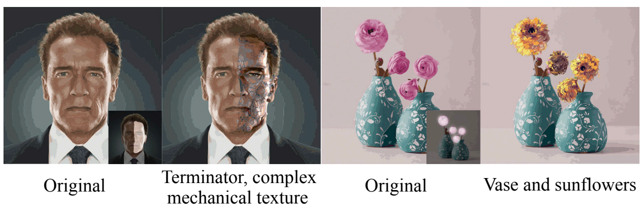


(a) The original image and the ROIs.

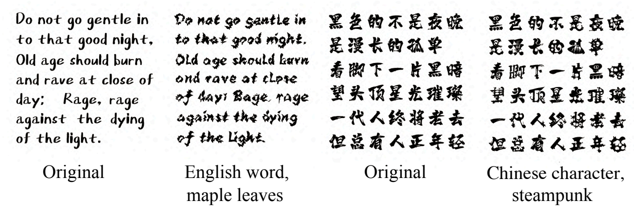
(b) Output with all the ROI prompts.

(c) Output with the global prompt only.

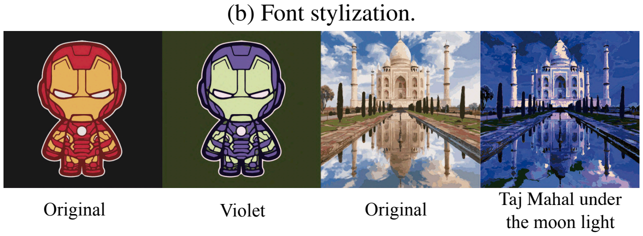
Figure 9: Text-guided manipulation results with and without the ROI prompts. The prompts are: 1. "Justice League Six", 2. "Aquaman", 3. "Superman", 4. "Wonder Woman", 5. "Cyborg", 6. "Flash, DC Superhero" and 7. "Batman". The areas of each ROI are shown as A1 to A7 in (a). The global prompt of (c) is the concatenation of all the prompts.



(a) Subregion editing.



(b) Font stylization.



(c) Re-colorization.

Figure 10: Separate control of parameters. (a) optimizes the parameters of a specific subregion. The target subregion is shown in the bottom right corner of the input. (b) optimizes the shape parameters. (c) optimizes the color parameters.

within the domain of the pre-trained StyleGAN model. The performances of StyleGAN-NADA and DiffusionCLIP are more balanced because they finetune the models based on input text prompts, while CLIPVG naturally achieves such balance without additional efforts.

In addition to the pilot study, we also present the quantitative results of CLIP score (Nichol et al. 2022; Kim, Kwon, and Ye 2022; Kwon and Ye 2021) in the supplementary material for the further evaluation of the semantic correctness.

Fine-Grained Control

Our flexible framework supports a set of fine-grained control methods, including the ROI specific prompts and the separate control of parameters. These methods can be applied to various applications on a needed basis.

ROI Prompts. CLIPVG is domain-agnostic and is not restricted by any generative model. It can be potentially leveraged to manipulate a complicated image containing multiple objects. The ROI CLIP guidance allows us to define different targets for the different objects. An example is shown in Figure 9, where each character is assigned a different prompt. It can be seen from Figure 9b that when ROI prompts are enabled, each character in the output image has a clear correspondence with its associated prompt. In contrast, the identity of each character becomes very ambiguous if a global prompt is used instead of the ROI prompts, as shown in Figure 9c. It is worth noting that transforming each character separately by a domain-specific method is not practical, since the ROIs are overlapping in this example.

Separate Control. CLIPVG optimizes the shape and color of all the vector graphical elements simultaneously by default. But it is sometimes desirable to edit a certain subregion or a certain aspect of the image. We can define a target subregion by a mask, and optimize the vector graphical elements within the subregion as shown in Figure 10a. We stylize the fonts in Figure 10b by editing the shape of the elements and keeping the color unchanged. Re-colorization is done in Figure 10c by optimizing only the color parameters.

Conclusion

We introduce CLIPVG, the first vector graphic based solution for text-guided image manipulation. The optimization process is greatly stabilized by the vector graphic specific regularization. We eliminate the dependency on additional pre-trained models, and support domain-agnostic image manipulation. We develop a robust multi-round vectorization strategy, and a set of fine-grained control methods which enables a wide range of applications. Extensive experiments and human evaluation confirm the superior semantic transfer performance and robustness of our method over the existing baselines.

References

2022. Adobe Image Trace. https://helpx.adobe.com/illustrator/using/image-trace.html. Accessed: 2022-08-03.

2022. Disco Diffusion. https://github.com/alembics/disco-diffusion. Accessed: 2022-07-28.

Carlier, A.; Danelljan, M.; Alahi, A.; and Timofte, R. 2020. Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems*, 33: 16351–16361.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 248–255. IEEE Computer Society.

Dhariwal, P.; and Nichol, A. Q. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 8780–8794.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; and Tang, J. 2021. CogView: Mastering Text-to-Image Generation via Transformers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 19822–19835.

Ding, M.; Zheng, W.; Hong, W.; and Tang, J. 2022. CogView2: Faster and Better Text-to-Image Generation via Hierarchical Transformers. *arXiv preprint arXiv:2204.14217*.

Frans, K.; Soros, L.; and Witkowski, O. 2021. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*.

Gal, R.; Patashnik, O.; Maron, H.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *ACM Trans. Graph.*, 41(4).

Huang, Z.; Heng, W.; and Zhou, S. 2019. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8709–8718.

Jiang, Y.; Huang, Z.; Pan, X.; Loy, C. C.; and Liu, Z. 2021. Talk-to-Edit: Fine-Grained Facial Editing via Dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13799–13808.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Kim, G.; Kwon, T.; and Ye, J. C. 2022. DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2426–2435.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kwon, G.; and Ye, J. C. 2021. Clipstyler: Image style transfer with a single text condition. *arXiv preprint arXiv:2112.00374*.

Li, B.; Qi, X.; Lukasiewicz, T.; and Torr, P. H. S. 2020a. ManiGAN: Text-Guided Image Manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 7877–7886. Computer Vision Foundation / IEEE.

Li, T.-M.; Lukáč, M.; Gharbi, M.; and Ragan-Kelley, J. 2020b. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6): 1–15.

Lopes, R. G.; Ha, D.; Eck, D.; and Shlens, J. 2019. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7930–7939.

Ma, X.; Zhou, Y.; Xu, X.; Sun, B.; Filev, V.; Orlov, N.; Fu, Y.; and Shi, H. 2022. Towards Layer-wise Image Vectorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Metz, L.; Poole, B.; Pfau, D.; and Sohl-Dickstein, J. 2017. Unrolled Generative Adversarial Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16784–16804. PMLR.

Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2085–2094.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR*, abs/2204.06125.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8821–8831. PMLR.

Reddy, P.; Gharbi, M.; Lukac, M.; and Mitra, N. J. 2021. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7342–7351.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi,

S. S.; Lopes, R. G.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR*, abs/2205.11487.

Schaldenbrand, P.; Liu, Z.; and Oh, J. 2022. StyleCLIPDraw: Coupling Content and Style in Text-to-Drawing Translation. *arXiv preprint arXiv:2202.12362*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Sun, J.; Deng, Q.; Li, Q.; Sun, M.; Ren, M.; and Sun, Z. 2022. AnyFace: Free-style Text-to-Face Synthesis and Manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 18666–18675.

Tian, Y.; and Ha, D. 2021. Modern Evolution Strategies for Creativity: Fitting Concrete Images and Abstract Concepts. *arXiv preprint arXiv:2109.08857*.

Vinker, Y.; Pajouheshgar, E.; Bo, J. Y.; Bachmann, R. C.; Bermano, A. H.; Cohen-Or, D.; Zamir, A.; and Shamir, A. 2022. CLIPasso: Semantically-Aware Object Sketching. *arXiv preprint arXiv:2202.05822*.

Zou, Z.; Shi, T.; Qiu, S.; Yuan, Y.; and Shi, Z. 2021. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15689–15698.