AI Illustrator: Translating Raw Descriptions into Images by Prompt-based Cross-Modal Generation

ACM MM 2022 (Oral)

Yiyang Ma, Huan Yang, Bei Liu, Jianlong Fu, Jiaying Liu

马逸扬

2022/09/04

Content

- Background
- Method
- Experiments

Content

- Background
- Method
- Experiments

Background

StyleGAN

StyleGAN v3?

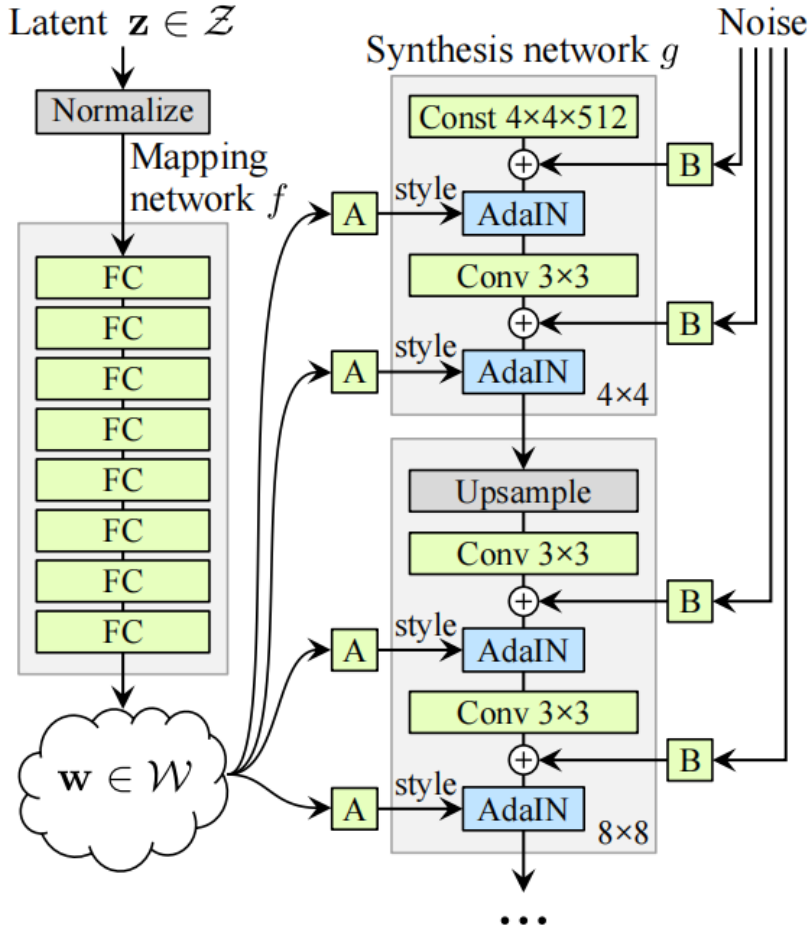

Figure 1 - Pipeline of StyleGAN v1

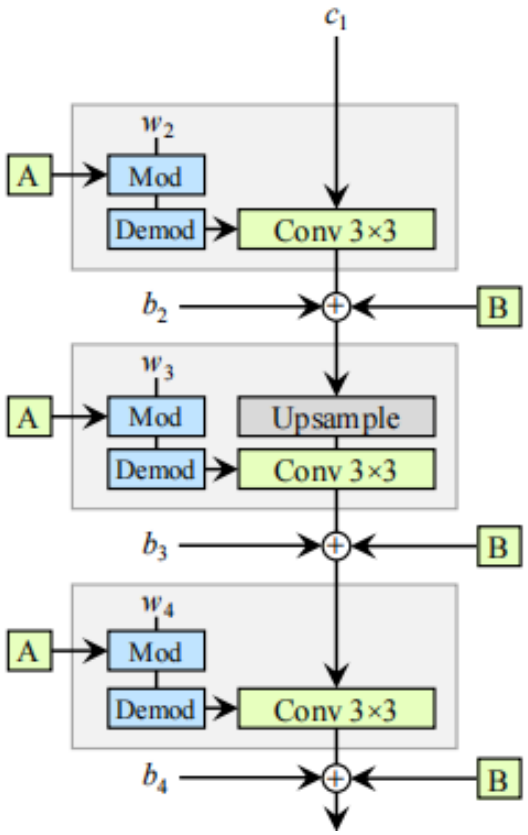

Figure 2 - Pipeline of StyleGAN v2

Background

CLIP

A pretrained model of mapping images and texts to embeddings. Matched image-text pairs will get close embedding pairs.

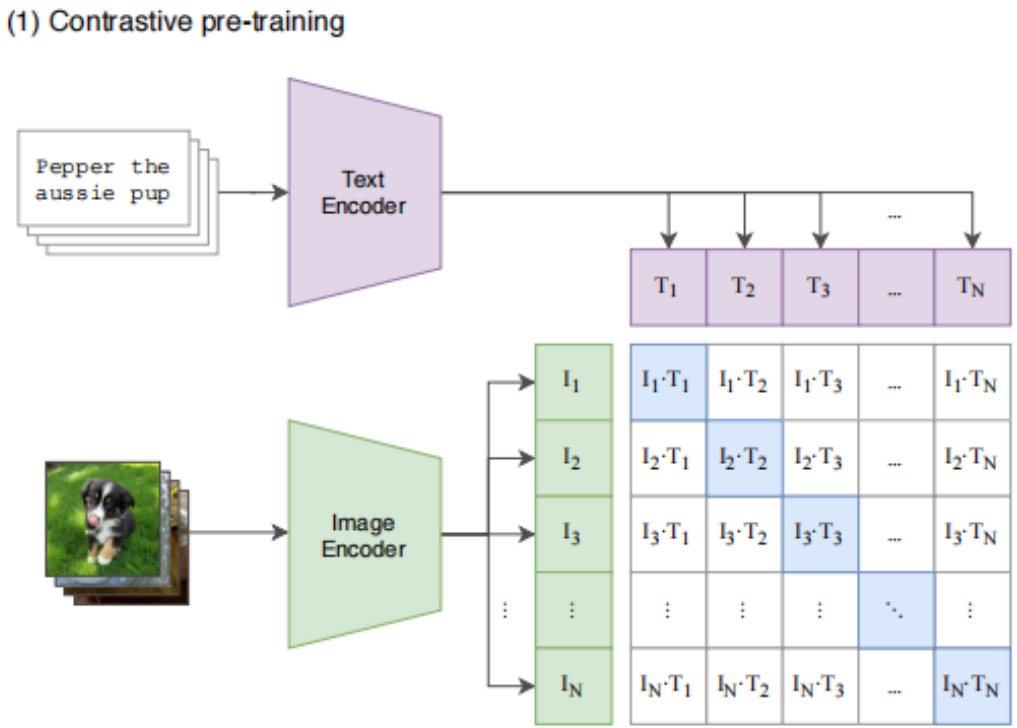

Figure 3 - Basic idea of CLIP

Background

Prompt

A method of using pretrained models to process inputs which are out of domain.

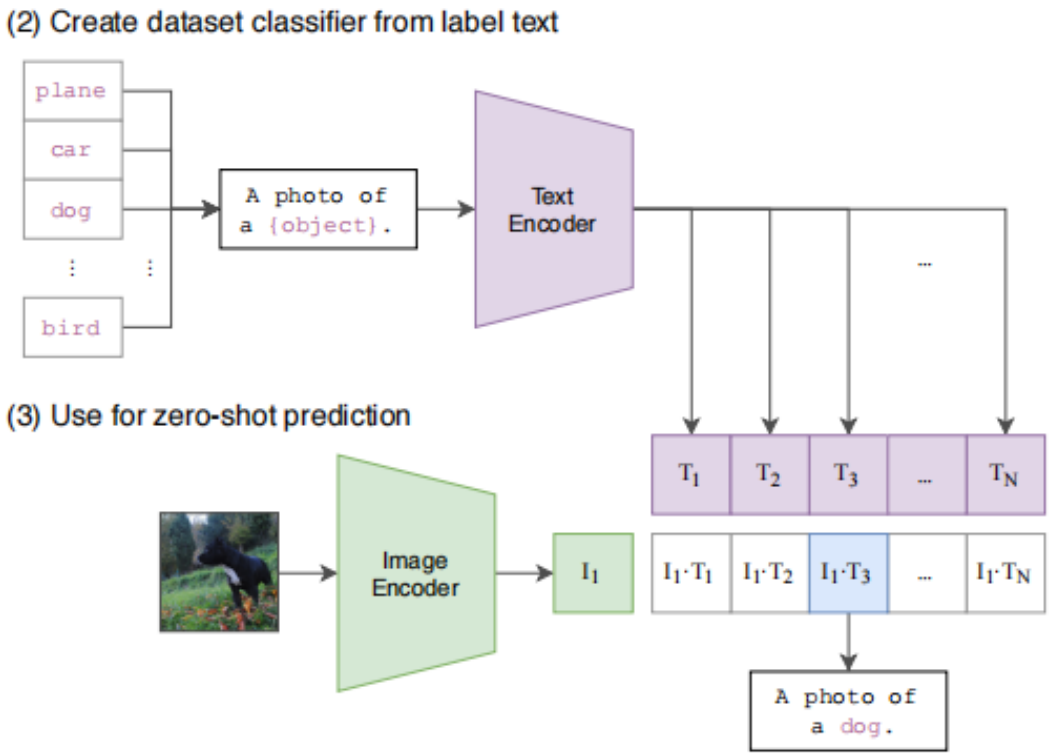

Figure 4 - Basic idea of Prompt

Content

- Background
- Method
- Experiments

Method

The total pipeline is shown below.

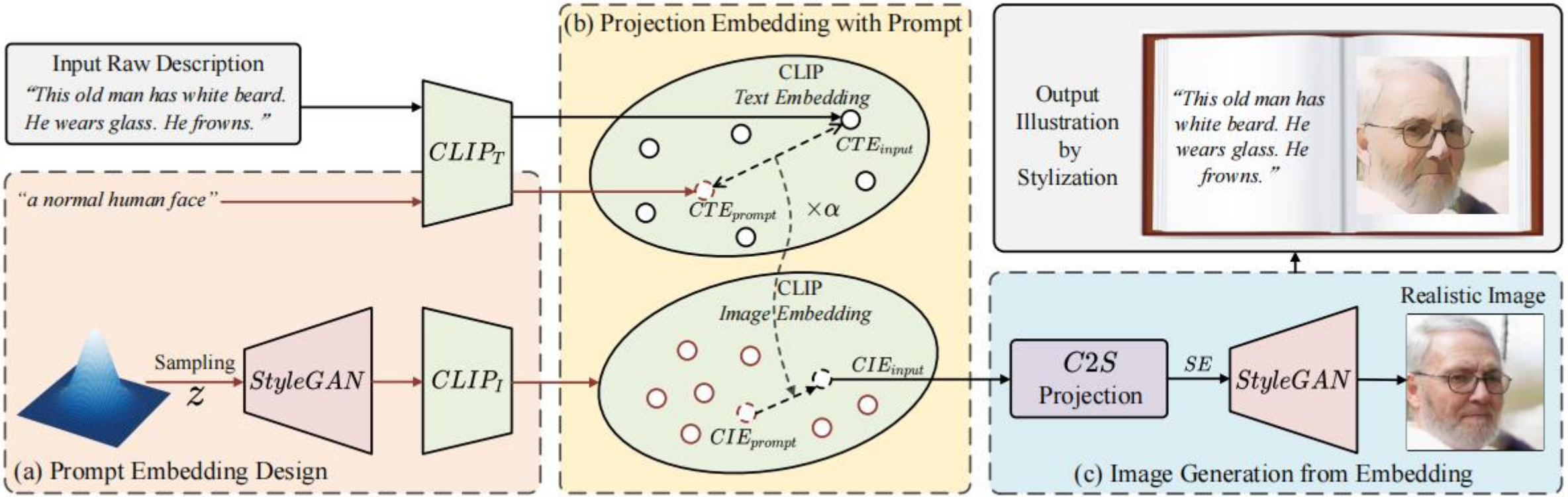

Figure 5 - Pipeline of AI Illustrator

The whole process of text-to-image translation concludes 2 projections and 1 generation.

Method

3 latent spaces:

- CLIP text latent space.

- CLIP image latent space.

- StyleGAN *Z* space.

So, the whole process is based on the projection among the 3 spaces above.

- Projection 1: input text embedding to image embedding.

- Projection 2: image embedding to StyleGAN *Z* space embedding.

- Generation: StyleGAN *Z* space embedding to final image.

Method

- Projection 1: input text embedding to image embedding.

Take CIE as the abbreviation of CLIP Image Embedding, CTE as the abbreviation of CLIP Text Embedding and SE as the StyleGAN *Z* Embedding.

It's hard to build a NN to achieve this projection because there is no paired data to train.

Due to the character of CLIP (semantically aligned text-image pair have aligned embedding pair), for two pairs of matched texts and images, we have:

$$CTE_1 - CTE_2 = CIE_1 - CIE_2 \tag{1}$$

Method

- Projection 1: input text embedding to image embedding.

If we can find a aligned embedding pair and this pair can generally represent all the texts and images within the domain, we can project the input *CTE* to *CIE* via equation 1. Name the “general embedding pair” as “prompt embedding pair”, we have:

$$CIE_{input} = CIE_{prompt} + (CTE_{input} - CTE_{prompt}) \quad (2)$$

In practice, we usually use

$$CIE_{input} = CIE_{prompt} + \alpha \cdot (CTE_{input} - CTE_{prompt}) \quad (3)$$

to control the distinctiveness of projection.

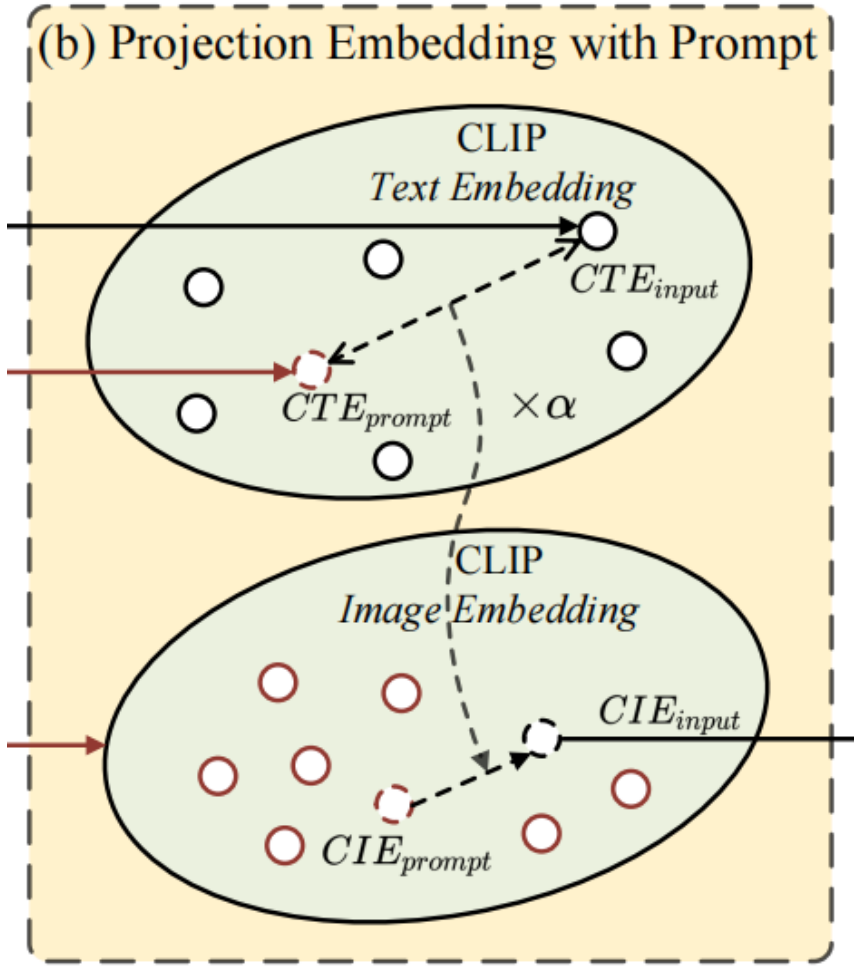

Figure 6 – Embedding projection

Method

- The method of finding the prompt embedding pair.

The prompt pair should be semantically aligned and has the ability to represent the whole domain. So, the pair should have unit length and the biggest average cosine similarity to all the embeddings within the domain. Take $\boldsymbol{y}$ as the embedding pair, we have:

$$\max_{\boldsymbol{y}} z = \frac{1}{n}\sum_{i=1}^{n}\frac{\boldsymbol{y}\cdot\boldsymbol{x_i}}{|\boldsymbol{y}|\cdot|\boldsymbol{x_i}|} \tag{4}$$

$$s.t.|\boldsymbol{y}| = 1 \tag{5}$$

We can simplify Eqn. 4 as:

$$\max_{\boldsymbol{y}} z = \boldsymbol{y}\cdot\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x_i} \tag{6}$$

which is the equation of a hyperplane.

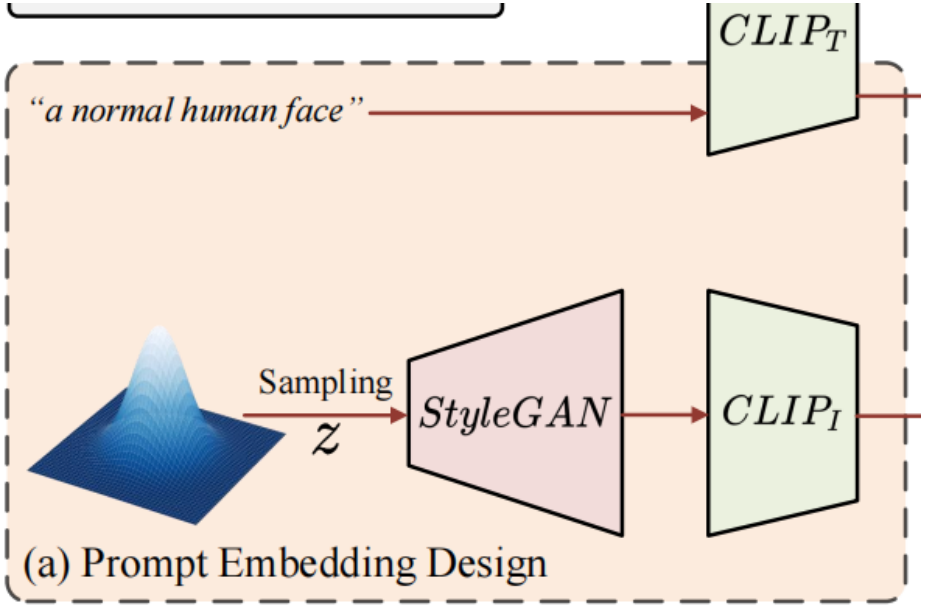

Figure 7 – Prompt embedding design

Method

- The method of finding the prompt embedding pair.

It's obvious that *z* will be biggest at the time of the hyperplane (Eqn. 6) and the hypersphere (Eqn. 5) are tangent. At this time, the vector $\boldsymbol{y}$ is the unit normal vector of the hyperplane:

$$\boldsymbol{y'} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i},\ \boldsymbol{y} = \frac{\boldsymbol{y'}}{|\boldsymbol{y'}|} \tag{7}$$

This is the prompt embedding we want.

For image prompt embedding, we can just sample a lot of images, calculate their *CIE* and get the prompt *CIE* via Eqn. 7.

For text prompt embedding, because text itself is the carrier of semantics, we can simply use a general sentence as prompt, like "A normal human face.".

Method

- Projection 2: image embedding to StyleGAN *Z* embedding.

We can build a NN to achieve this because it's easy to sample a lot of images and extract their *CIE*s and *SE*s as training pairs. The architecture of the NN is shown below.

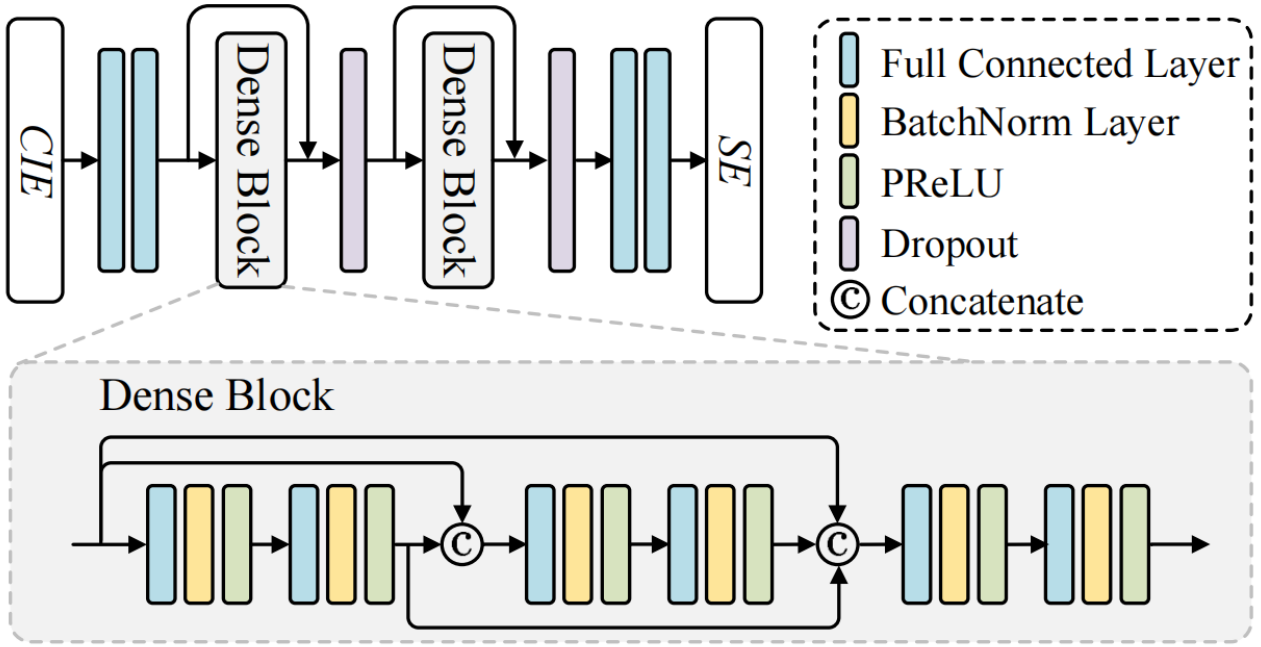

Figure 8 – Architecture of C2S projector

Method

- Projection 2: image embedding to StyleGAN *Z* embedding.

The training loss consists of 3 parts.

Basic constraint of the network:

$$\mathcal{L}_{l1} = ||SE_{pred} - SE_{true}||_1 \tag{8}$$

Semantic consistency loss:

$$\mathcal{L}_{sem\_cons} = CosDis(CIE_{input}, CLIP_I(G(SE_{pred}))) \tag{9}$$

The regularization loss which ensures the predicted *SE* is in the StyleGAN *Z* space:

$$\mathcal{L}_{reg} = ||mean(SE_{pred})||_1 + ||std(SE_{pred}) - 1||_1 \tag{10}$$

The total loss is the combination of the three losses:

$$\mathcal{L} = \lambda_{sem\_cons} \cdot \mathcal{L}_{sem\_cons} + \lambda_{l1} \cdot \mathcal{L}_{l1} + \lambda_{reg} \cdot \mathcal{L}_{reg} \tag{11}$$

Method

- Generation and stylization.

After getting the *SE*, we can generate the image we want by pretrained StyleGAN.

If we want to use the images as illustrations, we can cartoonize the realistic images.

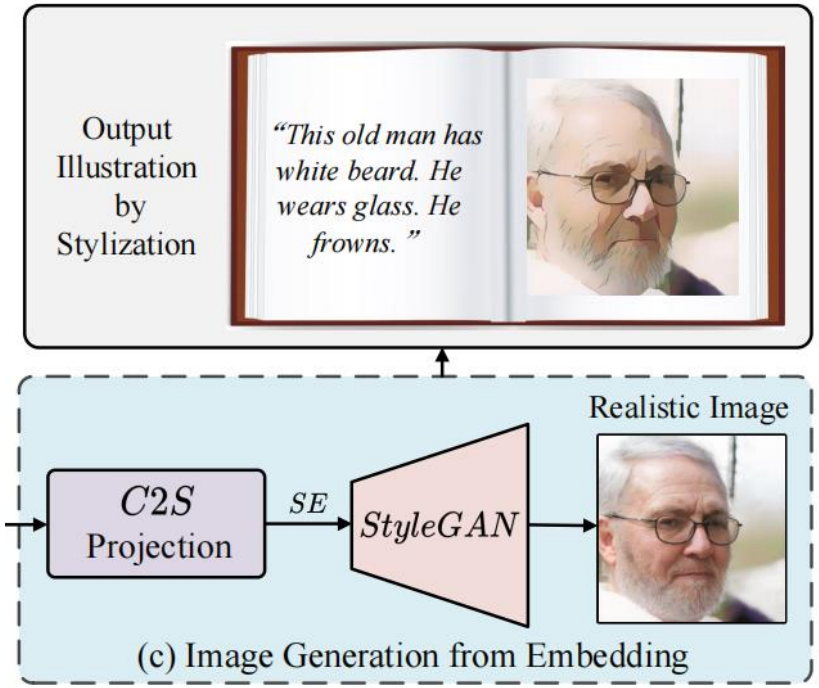

Figure 9 – Generation and stylization

Content

- Background
- Method
- Experiments

Experiments

Our method is based on CLIP which can deal with open-world words.

But in order to compare with the method which cannot process open-world words, we first show the translation results containing only the words of Multi-Modal CelebA dataset.

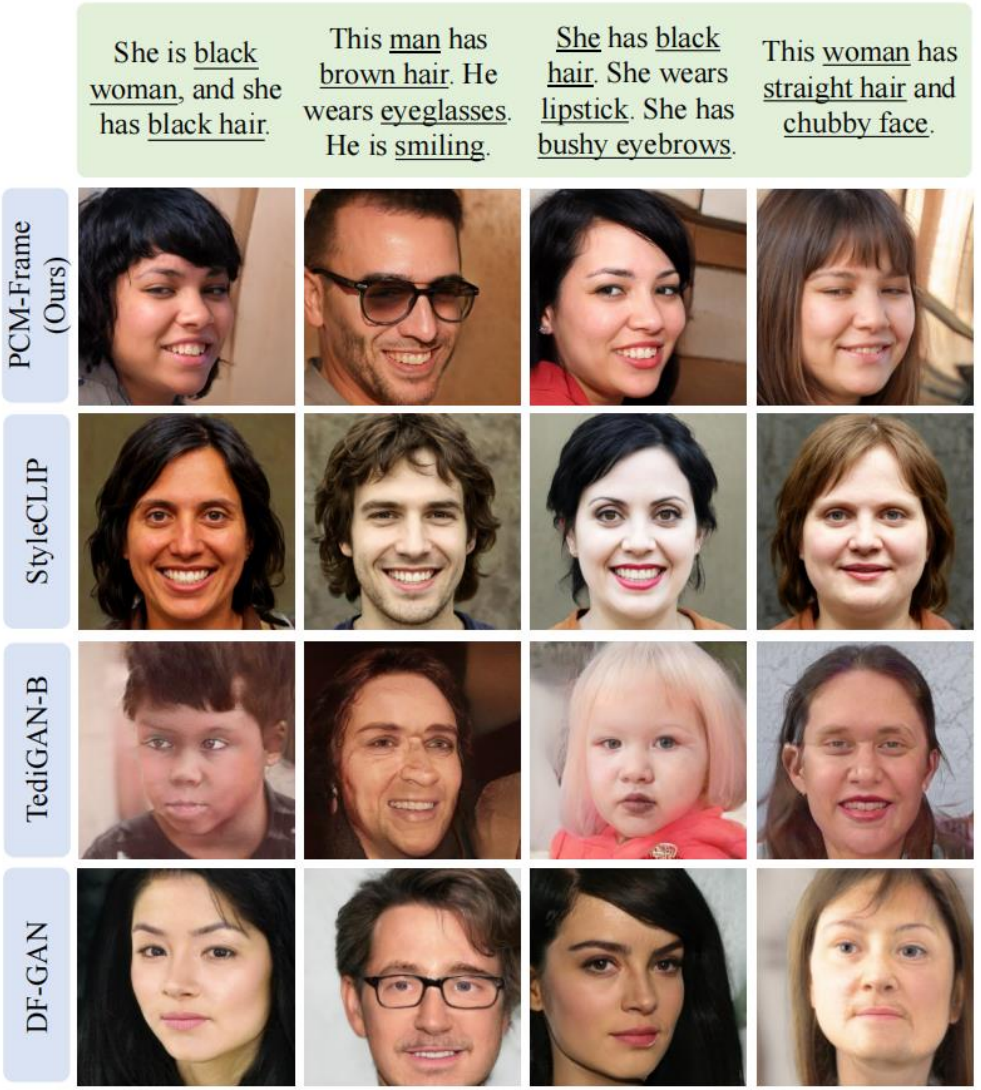

Figure 10 – Translation results with limited words.

Experiments

Then, we show the translation results containing open-world words. This task is more challenging.

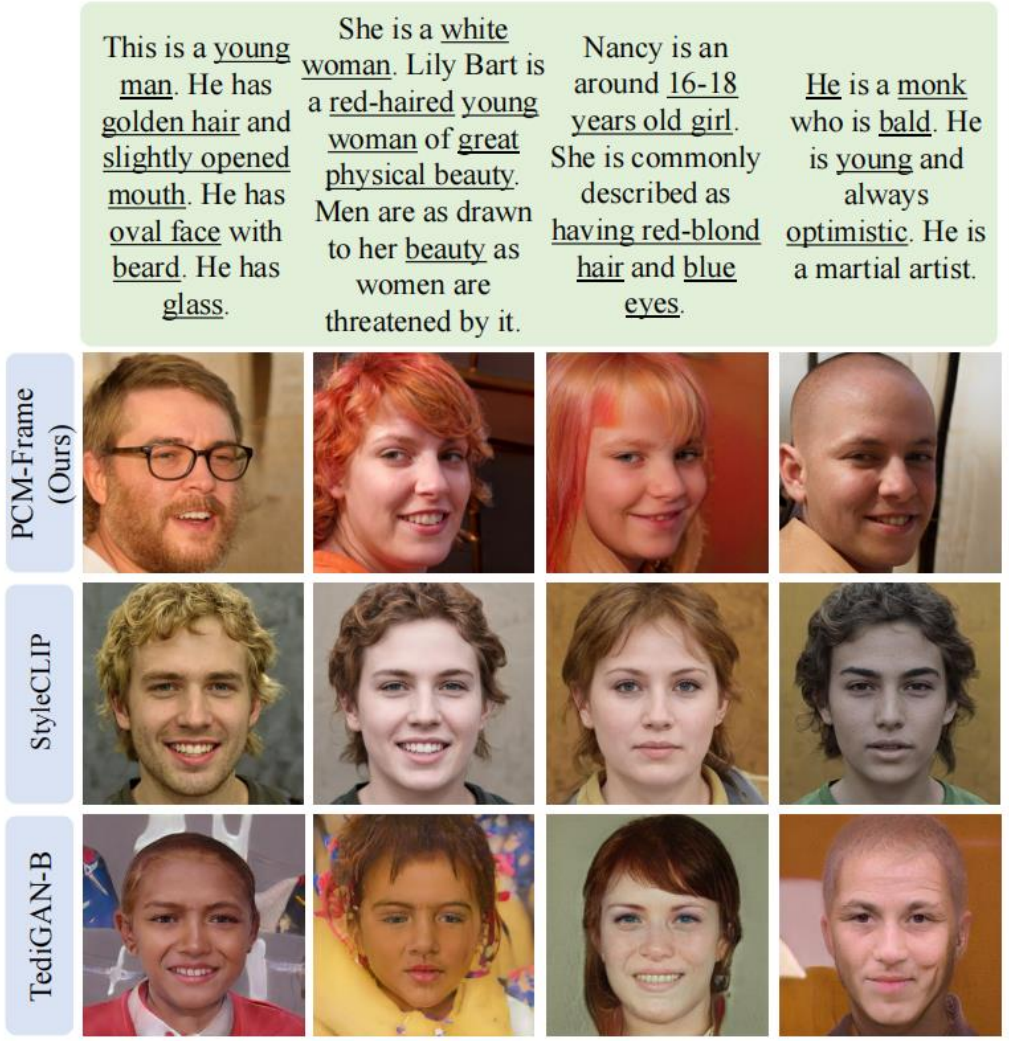

Figure 11 – Translation results with open-world words.

Experiments

Our method can generate diverse results with one input by taking random *SE*s in certain layers of StyleGAN. The results are shown.

Figure 12 – Diverse translation results.

Experiments

Our method can also translating non-face images as long as we have the corresponding pretrained generative model.

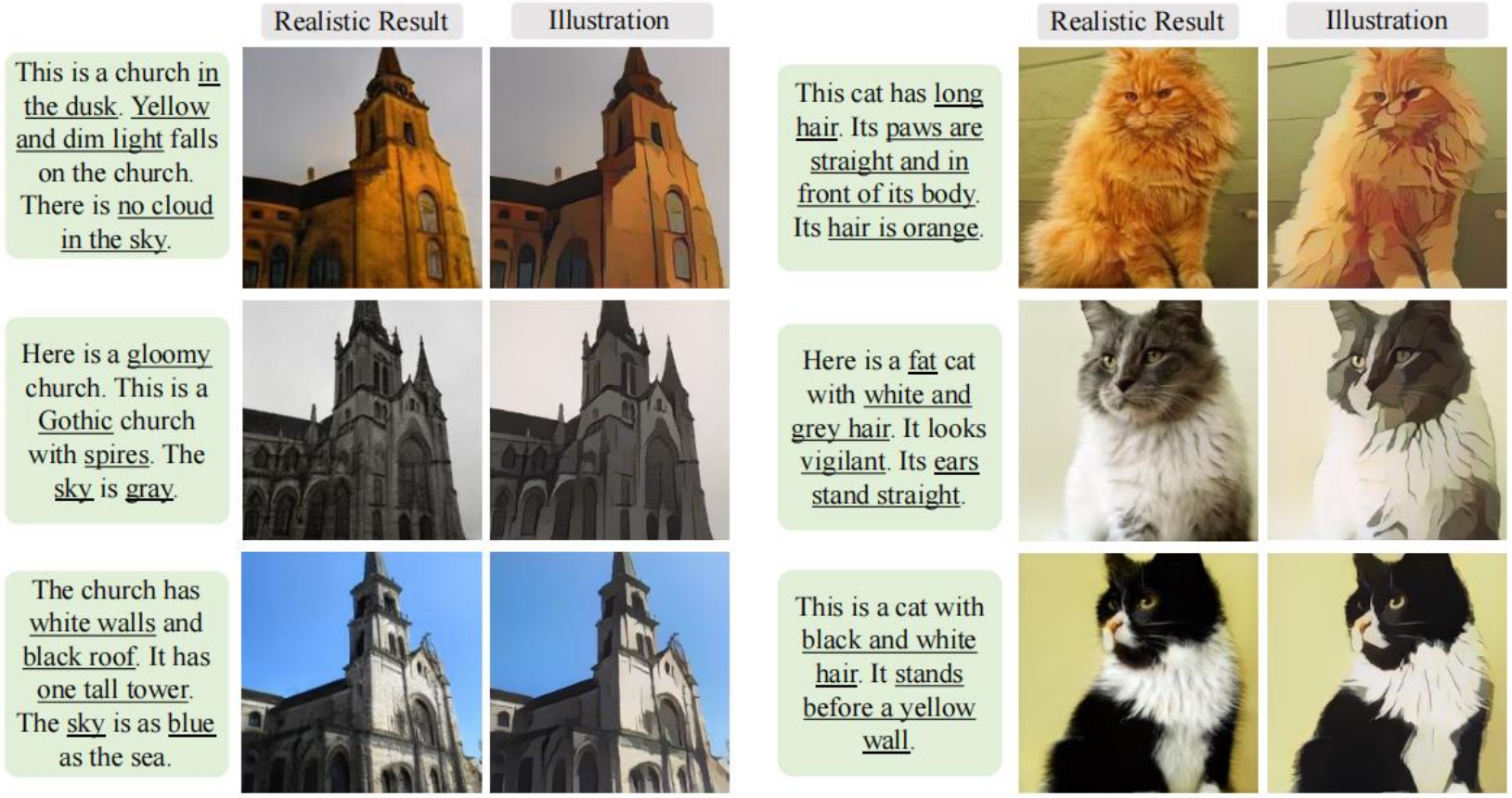

Figure 13 – Non-face results.

Experiments

Our method can also be used to manipulate the generated images via the equation below:

$$CIE_{target} = CIE_{origin} + \alpha \cdot (CTE_{target} - CTE_{origin}) \quad (12)$$

Figure 14 – Manipulation results.

Experiments

The ablation consists of 2 parts.

First, we demonstrate the efficiency of the proposed loss functions.

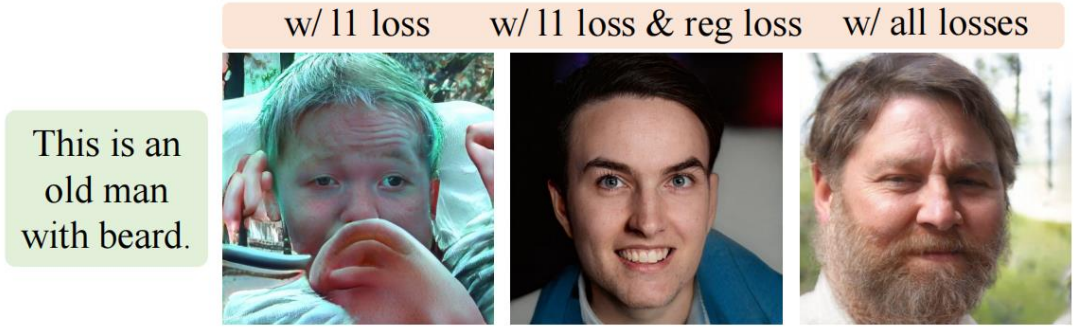

Figure 15 – Ablation on losses.

Second, we demonstrate the efficiency of the proposed prompts.

Figure 16 – Ablation on prompts.

Thanks for watching.

马逸扬
myy12769@pku.edu.cn