Appendix

A. Network Structure

We take the structure of the generator for 256x256 image as an example and illustrate the model structure in detail in Fig. 13. Note that the mapping from $\mathbf{z}$ to $\mathbf{w}$ are an MLP which is the same as StyleGAN2 [20], and both $\mathbf{z}$ and $\mathbf{w}$ are 512-dimensional. The whole structure of the decoder is the same as StyleGAN2 except that the skip connection is added, which is denoted as blue arrow. Since the $\mathbf{w}$ is operated actually at the conv-layer after each feature map, so the position of the arrow from $\mathbf{w}$ is shown at the end of each feature map in the decoder. The skip connection will add the feature of the encoder to the corresponding feature of the decoder. The encoder is basically the inverse structure of the decoder, and the feature size and dimension are shown in Fig. 13, which are obtained through convolutional layers. The omitted layers in the encoder follow the same rule that the resolution will reduce twice and the dimension will increase twice at maximum 512 dimension every two conv-layers.

B. Implementation Details

The whole project is implemented with Pytorch [30].

B.1. Multimodal Image Editing

The model for visualization and the comparison methods are for 256x256 images. The input image is normalized to $(-1, 1)$ The model is optimized by Adam [21] with learning rate 0.0025, $\beta_1 = 0$, $\beta_2 = 0.99$. We totally train the model for 5 million images, with batch size 32 on 8 V100 GPUs for 1 day. We also trained the generator for 512x512 images for language-guided image editing.

B.2. Language-Guided Image Editing

We use the pretrained generator for 512x512 images.

Supervised LGIE. The image encoder is ResNet50 [13], text encoder is the text transformer from the CLIP model [33]. The image resolution is 512x512 and the input image is normalized to $(0, 1)$ The model is trained with Adam [21] with learning rate 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Zero-shot LGIE. The editing process is optimized with the same optimizer and hyperparameter as the GAN inversion process in StyleGAN2 [20]. Moreover, the balance weight λ is flexible. We will output all the edited results given different λ ranging from 0.1 to 0.5 and then we select the best one.

B.3. Retrieval and Clustering

We need to conduct the conditional GAN inversion for all the dataset to obtain the $\mathbf{w}$ to support the style retrieval and clustering. Therefore, to accelerate the speed, we inverse the $\mathbf{w}$ corresponding to 128x128 resolutional images. We follow the same training setting for the inversion as StyleGAN2 [20]. The average time for such inversion is 30s per sample. Then we use KNN with cosine distance on the $\mathcal{W}$ space for retrieval and k-means with cosine distance on the $\mathcal{W}$ for clustering. Even though the $\mathbf{w}$ is only for 128x128 generator, it will not harm the output performance because in this stage we do not use $\mathbf{w}$ to generate images.

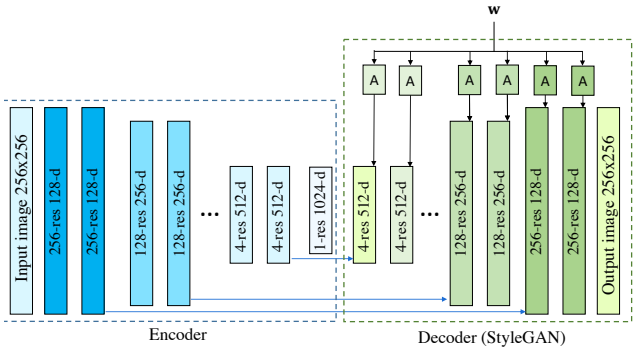

Figure 13. The details of our encoder and the skip connection.

B.4. Examplar-Based Image Editing

We conduct the conditional GAN inversion for 512x512 resolutional generator, and transfer the inverted $\mathbf{w}$ to new 512x512 resolutional images.

C. Effect of w at Different Layers

We analyze the effect of $\mathbf{w}$ at different layers using SeFa [36]. We compute the principle directions of the $\mathcal{W}$ space from the parameters of the affine matrix in the designated layers. For a given principle direction $\mathbf{n}$ and the $\mathbf{w}_0$ of the input image, we traverse the $\mathcal{W}$ space using a scalar α as

$$\mathbf{w} = \mathbf{w}_0 + \alpha \mathbf{n}. \tag{4}$$

Here we analyze the 256x256-resolutional generator with 14-layer decoder, where 12-14 conv-layers have output feature map size of 256x256, 10-11 conv-layers 128x128, 8-9 conv-layers 64x64, and 1-7 conv-layers have the size from 4x4 to 32x32. Fig. 14 and 15 show two examples for the traversing at these layers. We can see that in the high-level layers, the traverse of $\mathbf{w}$ exhibits salient color change, while for the low-level layers (layer 1-7), it does not. This means that the $\mathbf{w}$ in low-level layers will be ignored by the generator. And at different high-level layers, they seems to be able to achieve the similar effect, such as the green effect can be achieved by all the 12-14, 10-11, 8-9 layers in Fig. 14, so it is still not quite clear to us what editing styles different layers emphasize. And this could be further studied for future work. Moreover, for different images, the same direction generally has the same semantic according to the comparison of Fig. 14 and 15. However, the same style effect will overlay the original color style of the image, which explains why the final styles of the two examples have little different.

D. More visual results

D.1. Retrieval

Examples are shown in Fig 16.

D.2. Cluster

More examples are shown in Fig. 17.

D.3. Multimodal Image Editing

More examples are shown in Fig. 18.

D.4. Exemplar-Based Image Editing

More examples are shown in Fig. 19.

D.5. Language-Guided Image Editing

More examples for zero-shot LGIE are shown in Fig. 20. Note that the flower example does not receive mask input, but it still can handel local editing. This verifies that our model can understand the language semantic and the generator has good ability for local editing.

E. Language-Guided Image Editing

E.1. Supervised LGIE

We further provide more detailed introduction of the dataset, metrics, and more comprehensive comparison with other methods collected from [38].

Dataset. *MA5k-Req.* MA5k-Req [38] augments the language request to the image pairs in the MIT-Adobe FiveK dataset [3]. It contains 24,750 image pairs with one language annotation each and is divided into 17,325/2,475 /4,950 for train/val/test split.

Metrics. We follow the metrics in [38].

- *L1* distance directly measures the averaged pixel absolute difference between the generated image and ground truth image with pixel normalized to 0-1.
- *SSIM* measures image similarity through luminance, contrast, and structure.
- *FID* measures the Fréchet distance between two Gaussians fitted to feature representations of the Inception network over the generated image set and ground truth image set.
- *Image variance* σ measures the language controllability by computing the pixel variance of 10 output of the same input image controlled by different languages.

Comparison methods.

- *Input*: the evaluation between input and target image.
- *Bilinear GAN* [28], *SISGAN* [7], *TAGAN* [29]: these three methods are trained by learning the mapping between the caption and image without image pairs. Since there is not image caption in our task but the paired image and request, we drop the procedure of image-caption matching learning but adapt them with the L1 loss between input and target images.
- *Pix2pixAug* [43]: the pix2pix model [16] augmented with language used in [43].
- *GeNeVa* [9]: a GAN-based dialogue guided image editing method. We use it for single-step generation.
- *RL*: an RL approach introduced in [38].
- *T2ONet* [38]: T2ONet map the language request to a series of editing operations using weak supervision.
- *EDNet* [18]: EDNet enforce the language controllability using cyclic loss.

F. Data Collection

We collect the dataset called Discover-Req, where we augment the language request that describes what are edited for the before-and-after images. The whole process obtains the permit and the Discover images are allowed for research use. Totally we collected the language annotation for 4423 pairs of images with one sentence from Photoshop expert and three sentences from amateurs for each pair. The expert are hired from Upwork[3] and the amateurs from ScaleAI[4]. The annotation quality of the expert is trustable. To control the quality of amateurs, we only hire those who pass the annotation test, and the annotation result must be approved by another worker.

	L1 ↓	SSIM↑	FID↓	$\sigma_{\times 10^2}$ ↑
Target	-	-	-	-
Input	0.1190	0.7992	12.3714	-
Bilinear GAN [28]	0.1559	0.4988	102.1330	0.8031
Pix2pixAug [43]	0.0928	0.7938	14.5538	0.5401
SISGAN [7]	0.0979	0.7938	30.9877	0.1659
TAGAN [29]	0.1335	0.5429	43.9463	1.5552
GeNeVa [9]	0.0933	0.7772	33.7366	0.6091
RL [38]	0.1007	0.8283	7.4896	**1.6175**
T2ONet [38]	0.0784	0.8459	6.7571	0.7190
EDNet [18]	-	-	9.9500	-
Ours	**0.0731**	**0.8721**	**5.9791**	0.6809

Table 5. Quantitative results on MA5k-Req test sets. $\sigma_{\times 10^2}$ means that the image variance has been scaled up 100 times.

G. Tag List creating

The final tag list is: *dark, blue, red, white, vivid, vintage, warm, brown, clear, clarity, green, natural, yellow, orange, retro, cool, black, vignette, vibrant.*

The steps for creating this tag list is as follows. We firstly create a prior tag list based on the Adobe Photoshop commonly used style effect. Next, we tokenize all the annotated sentences in Discover-Req dataset, stemitize all the tokens, and manually select the style-like tokens and merge them with the prior tag list. Then we remove the tag that occurs to most of the image such as *bright* "contrast". Finally, we filter out those tags that occur less than 5 times among all sentences.

H. Customized Purity

Standard purity is computed for single labeled sample. However, each of our image pair has been labeled with multiple tags (the tokenized sentence may contain multiple valid tags). Therefore, we will extend the computation for purity to support multi-label situation. Specifically, for each cluster C_i, we firstly construct its corresponding tag pool T_i by collecting all the tag labels of all the samples in this cluster (the tag pool allows the same tag to occur many times). Next, for each tag t_j in the tag list of length L, we count t_j in each T_i and find the cluster with the maximum count of t_j as C_j. So now we have assigned the tag t_j to C_j. Note

[3] https://www.upwork.com/

[4] https://scale.com/

that in this way, one cluster might be assigned by multiple tags, but it does not matter. Then, we count the number of t_j in C_j as N_j and let $|\cdot|$ denote the total number of the elements of a set, the purity is defined as

$$\text{purity} = \frac{\sum_j^L N_j}{\sum_j^L |C_j|}. \tag{5}$$

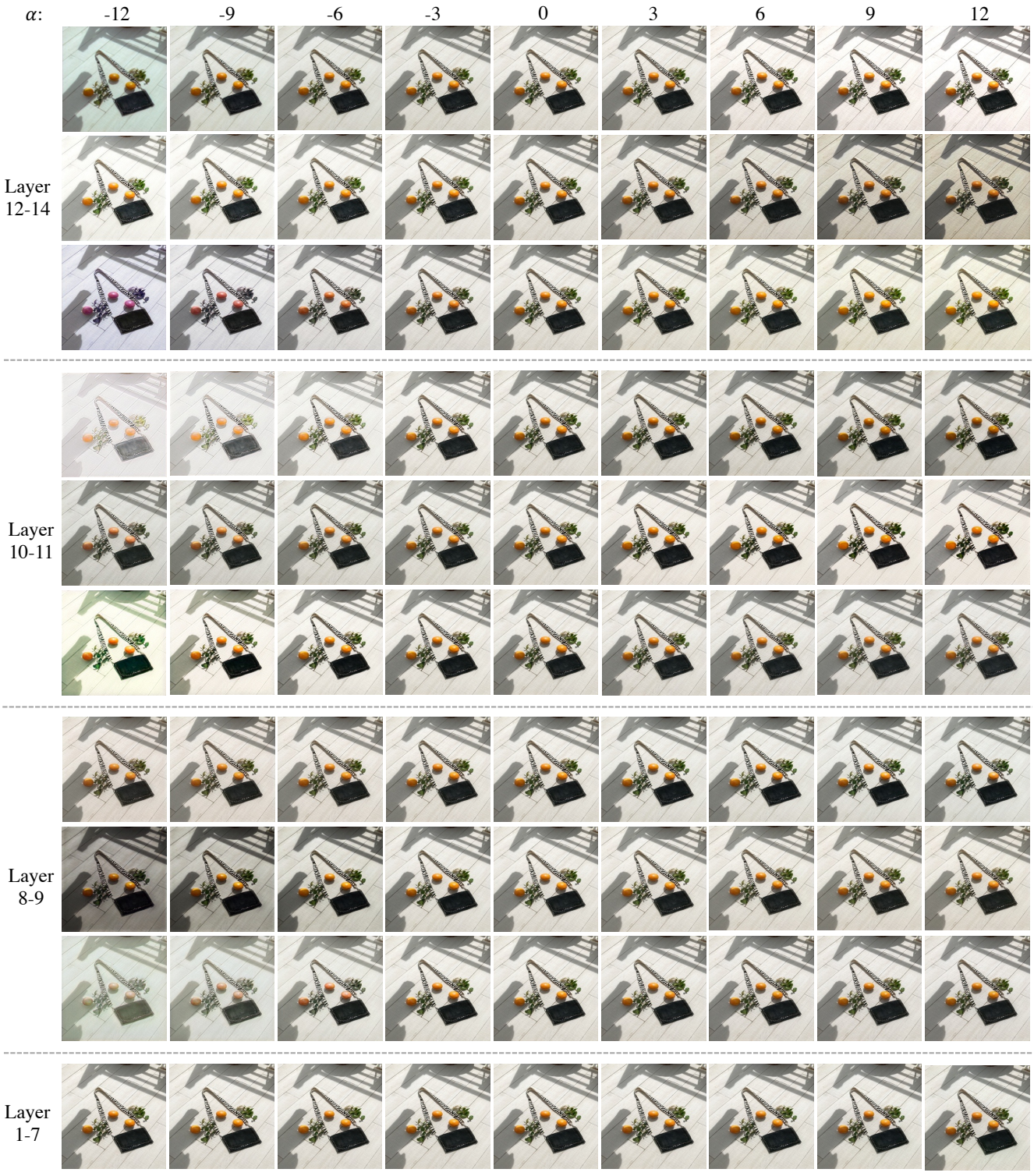

Figure 14. The visualization of the SeFa disentanglement on different layers. We select top-3 principle directions in layer 12-14, 10-11, 8-9 and top-1 direction for layer 1-7.

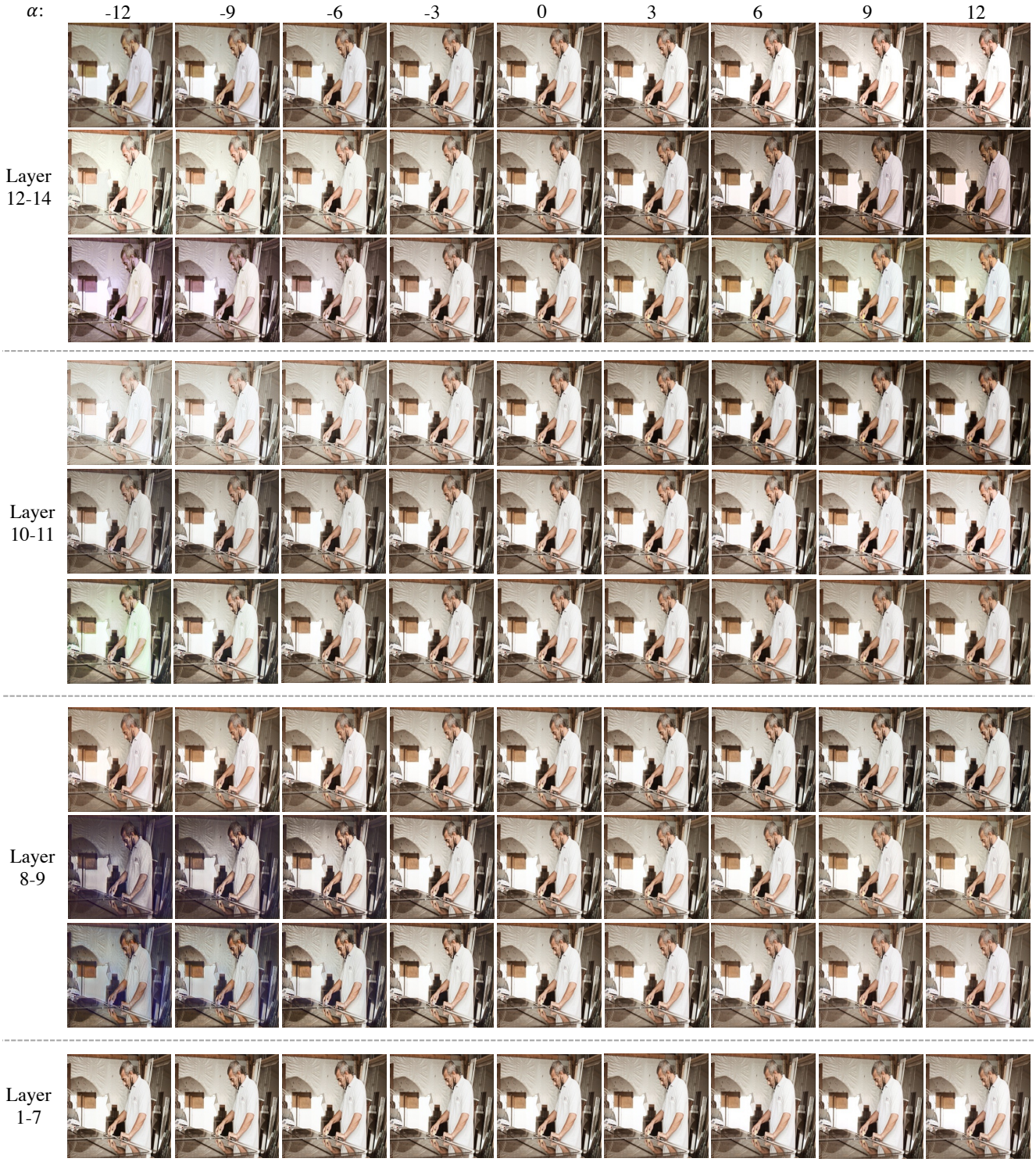

Figure 15. The visualization of the SeFa disentanglement on different layers. We select top-3 principle directions in layer 12-14, 10-11, 8-9 and top-1 direction for layer 1-7.

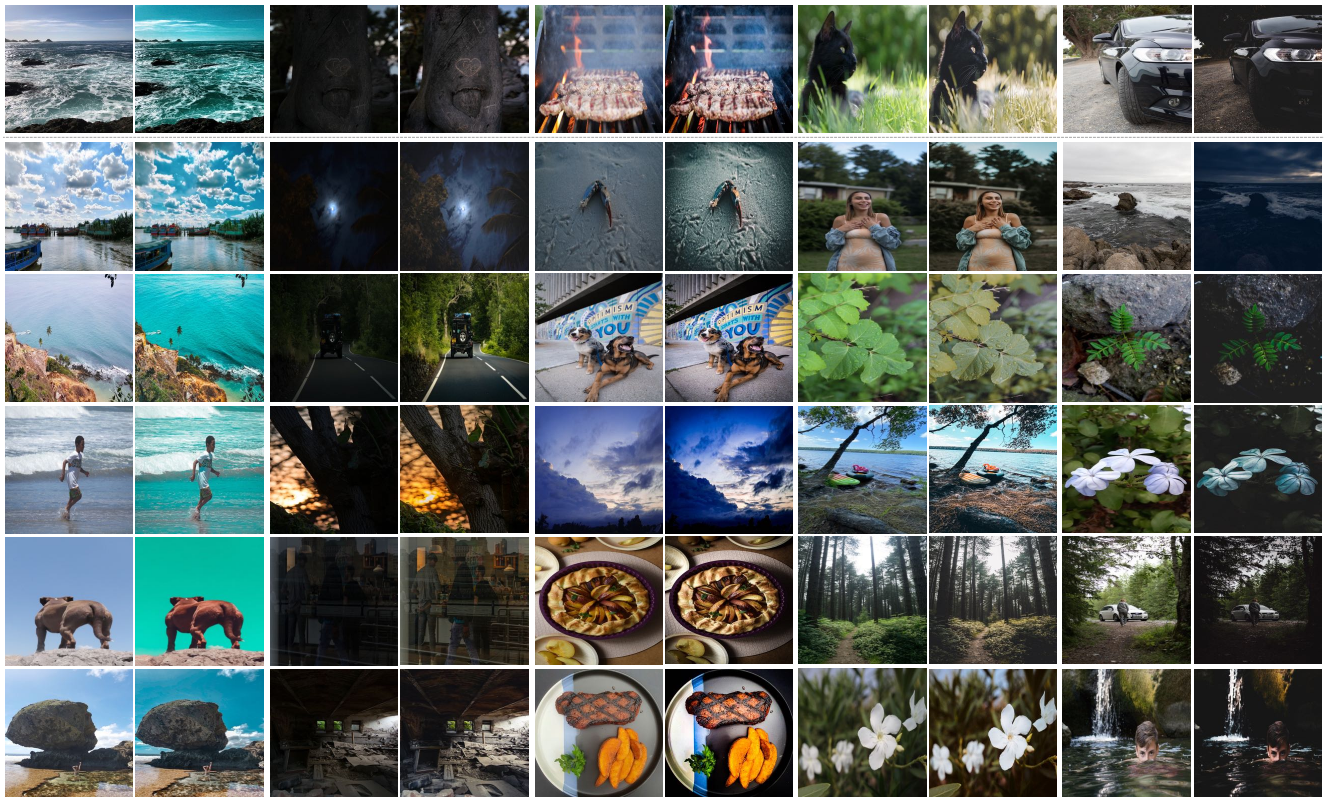

Figure 16. The visualization of the image pair retrieval results. The first row is the query pair, and the second to the last row are five retrieved pairs. For each pair, the left is source and the right is target.

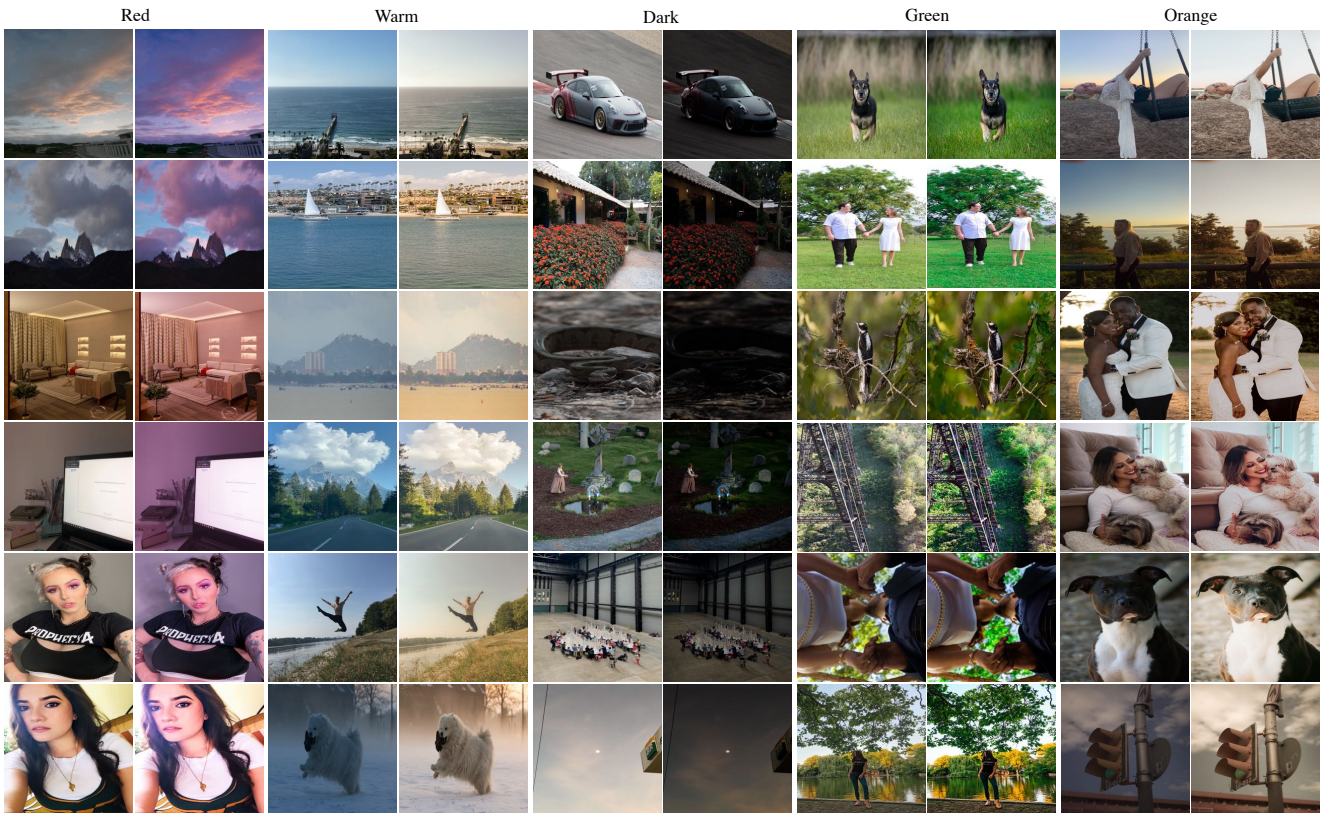

Figure 17. The visualization of the cluster results. For each image pair, the left is source and the right is target.

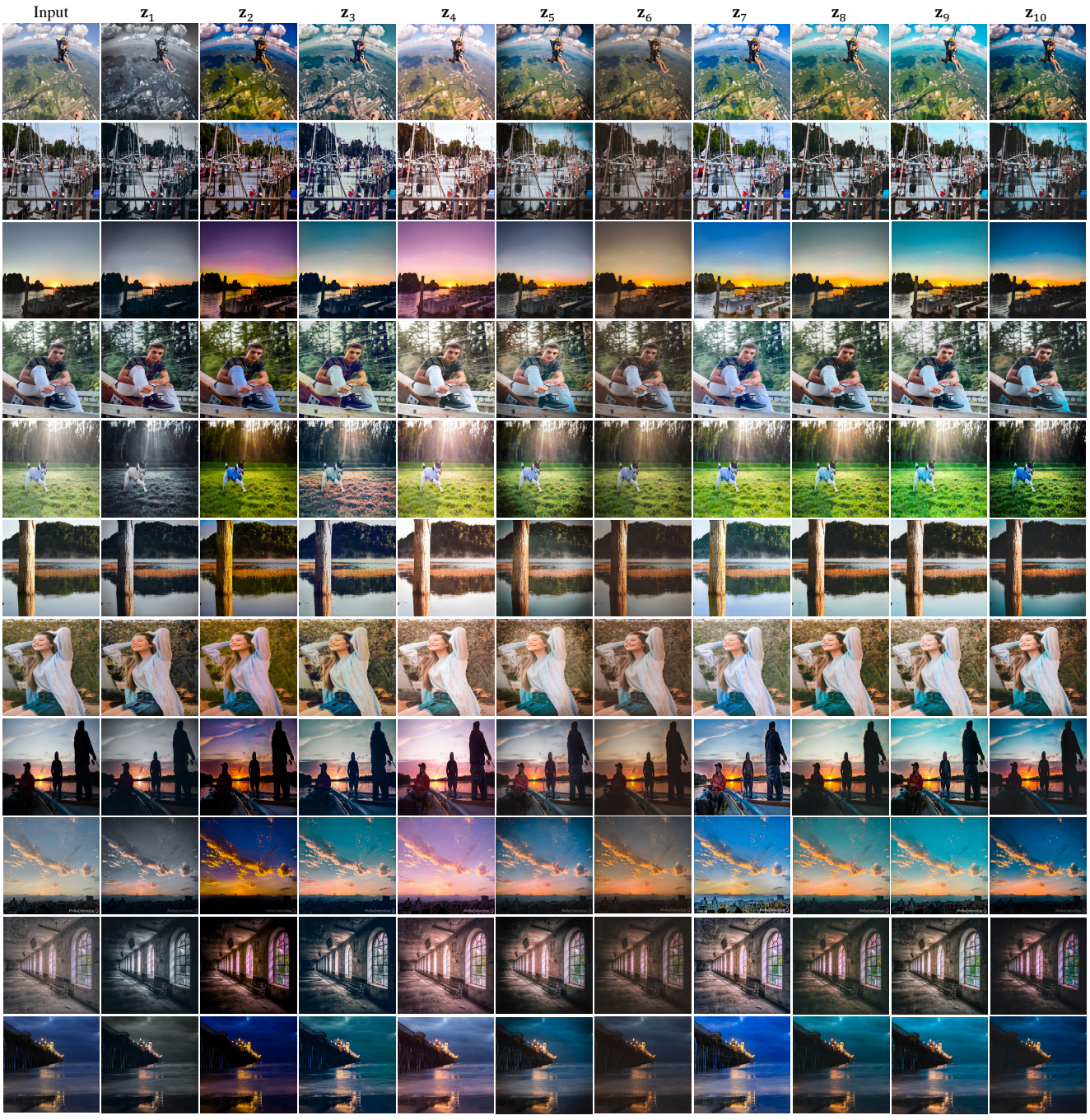

Figure 18. The visualization of the multimodal image editing result. Each column corresponds to the same $\mathbf{z}$, indicating one $\mathbf{w}$ has globally the same editing effect for all the images.

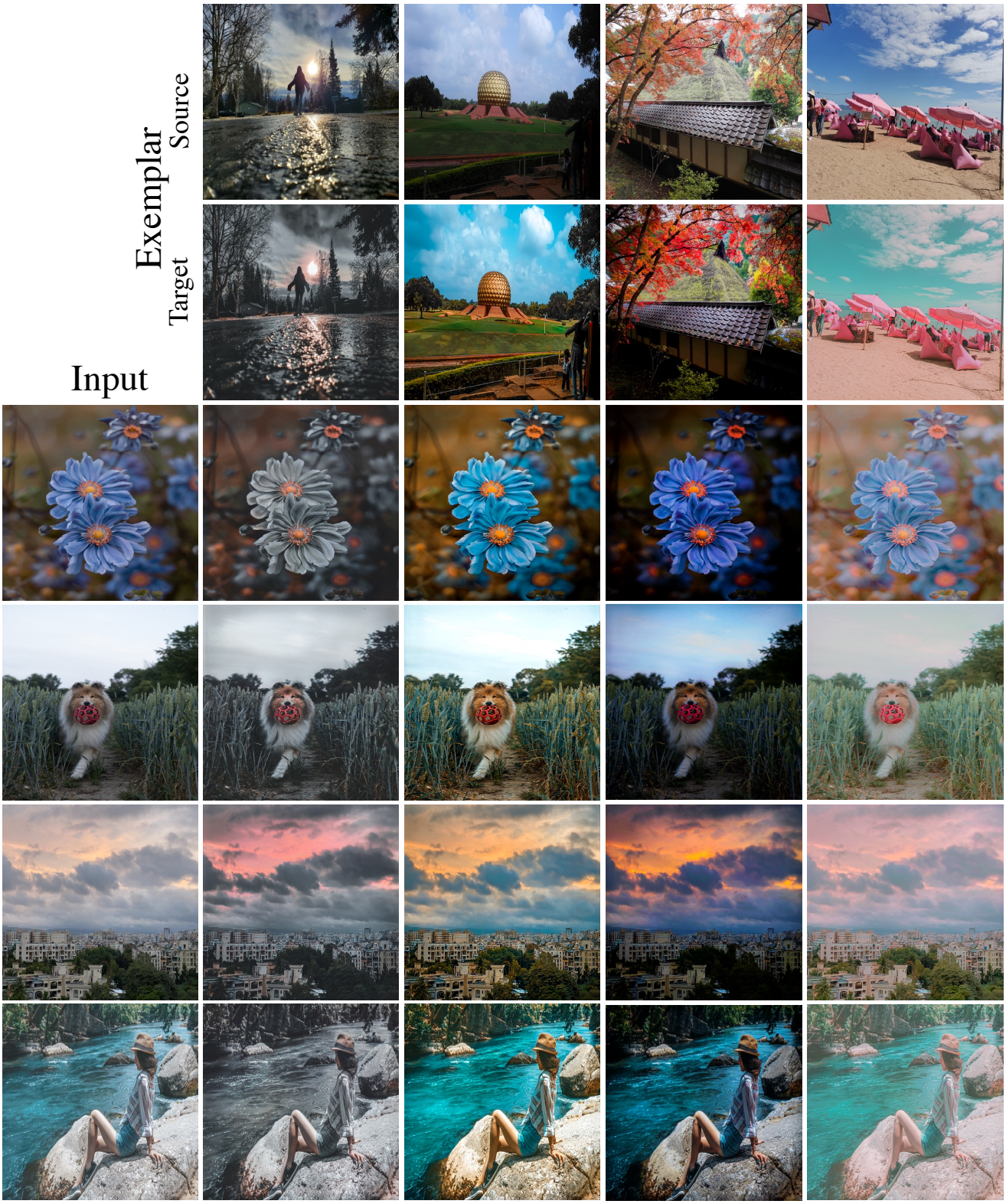

Figure 19. The visualization of exemplar-based image editing.

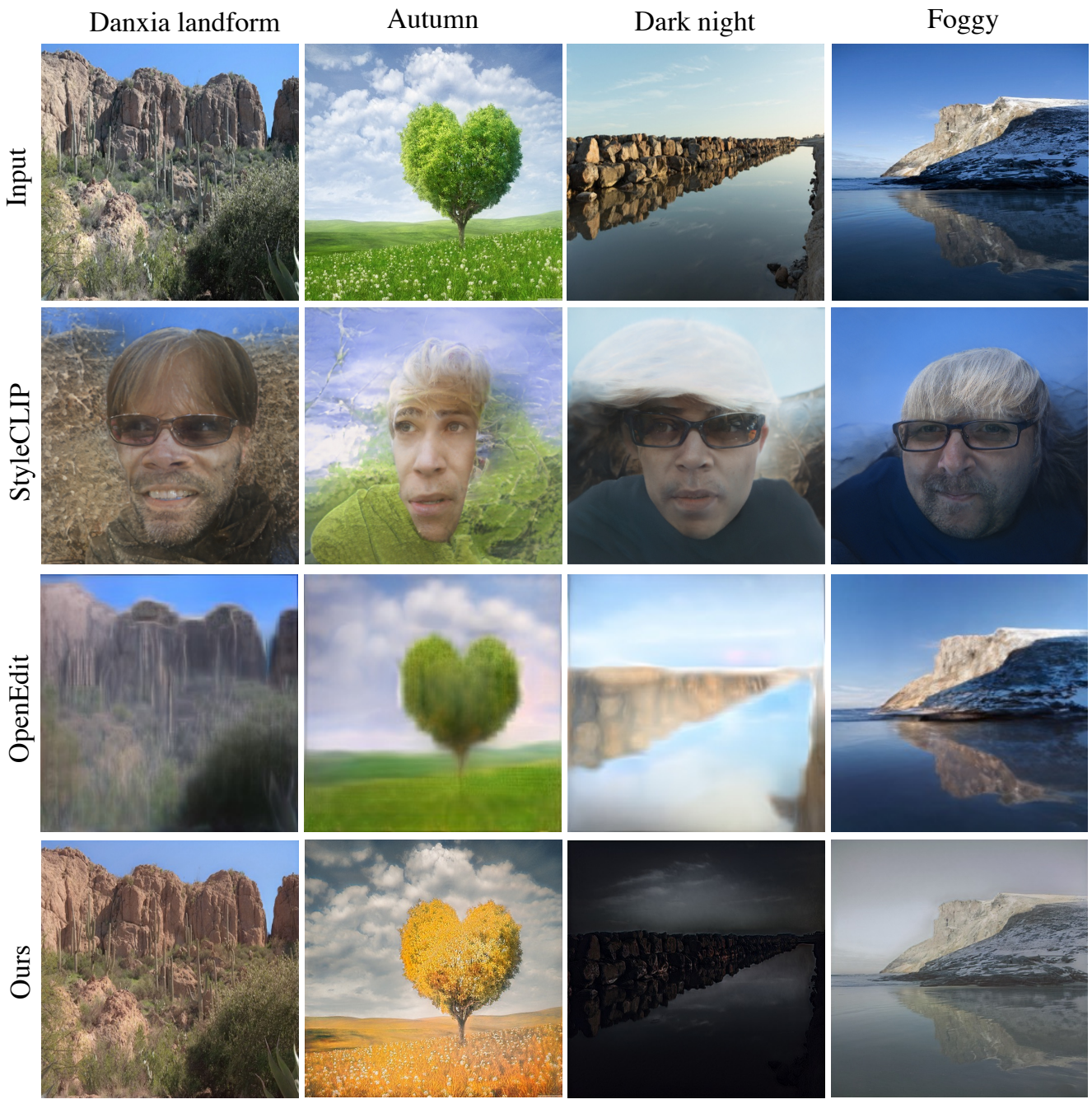

Figure 20. The visualization of the zero-shot LGIE.