SpaceEdit: Learning a Unified Editing Space for Open-Domain Image Color Editing

Jing Shi[1] Ning Xu[2] Haitian Zheng[1] Alex Smith[2] Jiebo Luo[1] Chenliang Xu[1]
[1]University of Rochester [2]Adobe Research

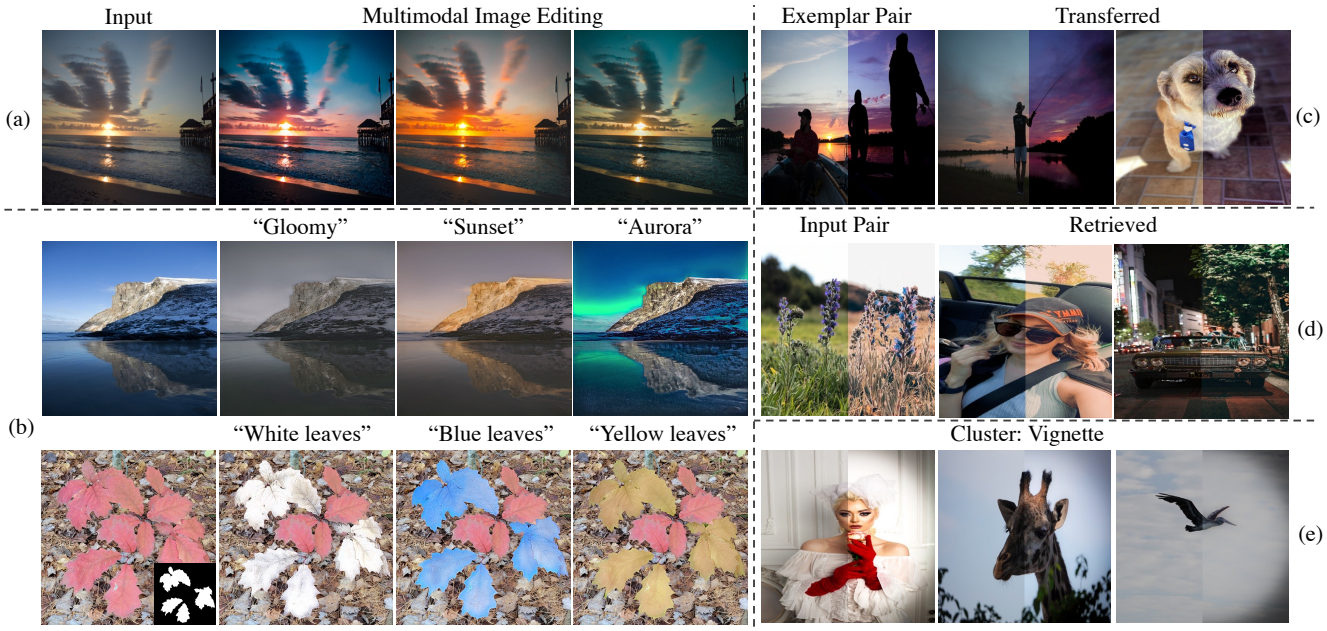

Figure 1. We propose a new image editing paradigm with a unified model that can handle various open-domain image editing tasks: (a) multimodal image editing, (b) language-guided image editing, (c) examplar-based image editing, (d) editing style retrieval, (e) editing style clustering. Images in (c)-(e) are visualized as half-before half-after edited.

Abstract

*Recently, large pretrained models (*e.g.*, BERT, StyleGAN, CLIP) show great knowledge transfer and generalization capability on various downstream tasks within their domains. Inspired by these efforts, in this paper we propose a unified model for open-domain image editing focusing on color and tone adjustment of open-domain images while keeping their original content and structure. Our model learns a unified editing space that is more semantic, intuitive, and easy to manipulate than the operation space (*e.g.*, contrast, brightness, color curve) used in many existing photo editing softwares. Our model belongs to the image-to-image translation framework which consists of an image encoder and decoder, and is trained on pairs of before-and-after edited images to produce multimodal outputs. We show that by inverting image pairs into latent codes of the learned editing space, our model can be leveraged for various downstream editing tasks such as language-guided image editing, personalized editing, editing-style clustering, retrieval,* etc. *We extensively study the unique properties of the editing space in experiments and demonstrate superior performance on the aforementioned tasks*[1].

1. Introduction

Image editing has shown wide spectrum of applications in various scenarios including image retouching [12, 40], style transfer [48, 49], language-guided image editing [18, 23, 26, 39], image harmonization [11], colorization [51], etc. However, the current research landscape independently studies these tasks on small and diverse datasets, underscoring the commonality of the image editing required for each

[1]Code and supplementary material can be found at the project page https://jshi31.github.io/SpaceEdit

task. As such, the customized approach for one specific task is cumbersome to extend to other related tasks, and the bespoke model trained on a particular dataset has difficulty generalizing to out-of-domain samples.

The recent surge of general pretrained architectures for vision [5, 8] and vision+language [27, 34] unifies different model structures for related tasks into common ones. These unified models are first trained on some pretraining datasets and then either fine-tuned on specific datasets or directly applied in a zero-shot manner for different downstream tasks. Numerous studies have demonstrated that the generalization and knowledge transfer capability of the pretrained models are key to their success. Here comes a natural question, *is there any unified pretraining task or network architecture that we can leverage for the scope of image editing?* One related work is StyleGAN [19], which is trained to generate realistic images for closed-domain categories such as faces, cats, and cars. Since then, a series of manipulation works [6, 35, 36, 42, 45, 46] have been built upon StyleGAN by inverting a given image to its latent space and then manipulating the latent code to generate a new image while keeping the generator intact.

Despite being successful for closed-domain image editing, StyleGAN has not been demonstrated to generate open-domain user photos which could contain various objects and complex scenes, therefore compromising its generalizability and application scenarios. In this paper, we are interested in one particular area of the open-domain image editing problem, *i.e.*, apply some artistic styles to a given photo to achieve a different look while keeping its original content, structure, and texture. Although not covering all editing scenarios, the applications of our problem are already quite useful and broad for many photo editors and photographers. Indeed many commercial photo editing softwares such as Adobe Lightroom provide some predefined global and local editing operations (*e.g.*, contrast, brightness, color curves) to solve this problem. However, their editing interfaces are not intuitive or convenient for many users, especially beginners, which we hope to mitigate with our newly proposed editing framework.

To achieve our goal, we propose a pretraining task that is useful for many editing downstream tasks. The pretraining task aims to transform a given before-edited image into an after-edited image with some artistic editing style controlled by some random noise vector. To learn the pretraining task, we first collect a new large-scale dataset with 60k pairs of before-and-after photos from the Lightroom Discover website[2]. Then we propose a new encoder-decoder network structure that appends the StyleGAN as a decoder to an image encoder. The modulation modules and the mapping network of StyleGAN are inherited; therefore sampling different latent codes can generate multimodal outputs.

[2]https://lightroom.adobe.com/learn/discover

Having trained the generator, we further analyze the properties of the new latent space $\mathcal{W}$, whose meaning is entirely different from StyleGAN's $\mathcal{W}$ space. Concretely, the $\mathcal{W}$ space of StyleGAN contains the complete content information of the generated images while our $\mathcal{W}$ space only captures various editing styles, which are independent of image content. Therefore, we use a recent method SeFa [37] to analyze the latent semantic directions and employ some GAN inversion method [20] to invert the latent code from a pair of before-and-after images. We find that our $\mathcal{W}$ space has similar controllability and semantic disentanglement as the original StyleGAN, and our $\mathcal{W}$ space emphasizes on the semantics of editing style. We also verify that our inverted latent code is useful for both generation and recognition (*e.g.* clustering, retrieval) tasks.

Given the unique properties of our editing space $\mathcal{W}$, we apply our pretrained generator to several open-domain image editing tasks. First, we explore the task of language-guided image editing (LGIE) [18, 39], which aims to edit an image to match a given editing request. Existing methods must train their full models with sophisticated pixel-level losses on the limited dataset, thus facing the overfitting issue given the enormous language and image space. In contrast, we propose a simple encoder which maps the input image and text features into the 512-dimensional editing space and then resorts to our pretrained generator to generate the output image. Experimental results verify the advantage of our pretrained model serving for this downstream task.

Second, inspired by recent styleCLIP [32], we further equip our generator with CLIP [34] for zero-shot free-form LGIE. Our method is able to not only generate semantic editing styles such as "sunset," "gloomy," but also change the color of an object to different colors as shown in Fig. 1.

Last but not least, since each latent code of a before-and-after pair in $\mathcal{W}$ space corresponds to some editing style, we can transfer the editing style of one image pair to the other images to achieve personalized editing. Besides, we can retrieve similar editing styles for personal style recommendation on a large database of user editing examples.

In summary, our contributions are three-fold. First, we propose a new pretraining task and a network architecture that is beneficial for various pertinent tasks for open-domain image color and tone editing. Second, we demonstrate that the $\mathcal{W}$ space of the pretrained model corresponds to various editing styles. Such embeddings are useful for both generative and recognition tasks. Finally, we demonstrate better performance of our pretrained model on various downstream tasks, including multimodal image editing and language-guided image editing benchmarks.

2. Related Work

Leveraging GAN latent space for image editing. Many works have been proposed to discover the semantics in

GAN's latent space for image editing in the supervised way [10, 24, 36], self-supervised way [17, 33], and unsupervised way [6, 37, 42, 43, 45]. However, all the above works focus on unconditional GANs while our method relies on conditional GAN. Although traversing the latent space of unconditional GANs can achieve image editing in closed-domain images such as faces, its incapability of generating real-world images (*e.g.*, multiple objects and complex scenes) limits their generalization and application. In addition, since their hidden spaces need to retain all the information of the generated outputs, the inversion [55] of an open-domain image is usually compromised for photo fidelity [1, 35]. In contrast, the editing space of our proposed model does not have such limitations. Moreover, since each inverted latent code in the editing space corresponds to some editing style, we can directly cluster them to find representative semantics, which is not investigated by previous methods.

Multimodal image editing. Our pretraining is a multimodal image editing task which requires diverse outputs controlled by some random vectors given an input image. A branch of works achieves the multimodal diversity by using an inverse mapping from the generated image to the input noise [56], disentangling of image content and style [15, 22], or explicitly enforcing the image diversity with distance-based loss term [25, 28]. However, the enforcement of diversity deteriorates the image quality. Inspired by the recent modulation approach [54] for multimodal image inpainting, we propose a similar network architecture specifically for open-domain image editing. The difference is that our modulation layer does not use the features of the input image, which leads to better fidelity and diversity.

Language-guided image editing. Language is a flexible and user-friendly way to control image editing. [4, 9, 18, 38, 39] collect paired data (*i.e.* input image, language request, target image) for supervised training. However, the language annotation is expensive, and the limited data size would constrain their generalizability. Other works [7, 23, 29, 30, 50] are trained with only image caption pair but are restricted to domain-specific images such as birds and flowers. Recently, some attempts are made to achieve zero-shot open-vocabulary image editing [2, 32, 46] by modifying the latent space of a pretrained StyleGAN [19] via a state-of-the-art image-text matching model CLIP [34]. Hence, the data domain that the StyleGAN is pretrained on will limit the editing domain. Although [26] trains a generator by reconstruction and thus can work for any open image domain, the generation quality is not guaranteed. In contrast, the editing quality of our method is guaranteed by the unique properties of our learned editing space. We propose different approaches for both supervised and zero-shot language-guided image editing. Each of them achieves better editing results than other methods.

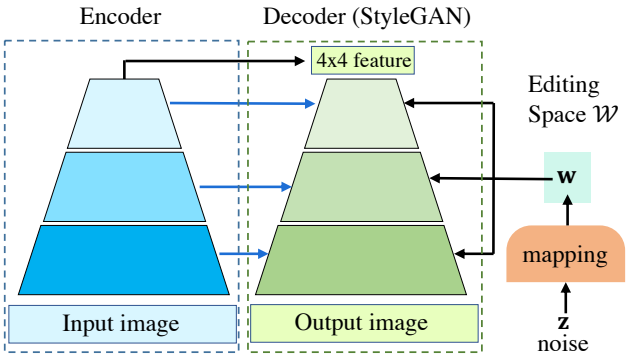

Figure 2. The structure of our generator for the pretraining task. The blue arrows represents skip connections.

3. Multimodal Image Editing as Pretraining

For the pretraining task, our goal is to learn an image-conditional generator with a latent space that can control various editing styles. The latent space should be semantic, disentangled as well as complete to be useful for various downstream editing tasks. We select multimodal image editing as our pretraining task as it encourages to produce diversified outputs with different editing styles.

We propose an image-to-image translation framework that consists of an image encoder and an image decoder with some random noise $\mathbf{z} \in \mathcal{Z}$ as additional inputs to control different editing styles. Since StyleGAN2 [1] has shown great disentanglement of its latent space for generative tasks, we adopt its architecture as our decoder where the noise input $\mathbf{z}$ is firstly mapped to an intermediate latent code $\mathbf{w} \in \mathcal{W}$, and then is further used to modulate the convolutional kernel at different layers, as depicted in Fig. 2. The role of the image encoder is to encode the input image into features of different levels, and the lowest 4x4 feature map is used to replace the original constant input of StyleGAN2. Apart from the straightforward docking of the encoder and decoder, we further stitch them via skip connection at different resolutions of the feature maps from the encoder to decoder, in view of preserving fine-grained details. Please refer to Appx. A for detailed structure.

More formally, let the source (before) image be I_{in}, the target (after) image I_{tgt}, the generator G, the discriminator D, the output image $I_{out} = G(I_{in}, \mathbf{w})$ where $\mathbf{w} = \text{Mapping}(\mathbf{z})$. Our generator is trained with the regular conditional discriminator loss $\mathcal{L}_{adv}$ as

$$\begin{aligned}\mathcal{L}_{adv} = &- \mathbb{E}_{I_{in}, I_{tgt}}[\log(D(I_{in}, I_{tgt}))] \\ &- \mathbb{E}_{I_{in}, I_{out}}[\log(1 - D(I_{in}, I_{out}))]. \end{aligned} \quad (1)$$

Note that we circumvent direct pixel supervision such as L1 loss [16] for the purpose of encouraging the generation diversity, as suggested in [54]. Some qualitative output results from our trained generator is visualized in Fig. 3. Our generator is able to not only generate diverse outputs

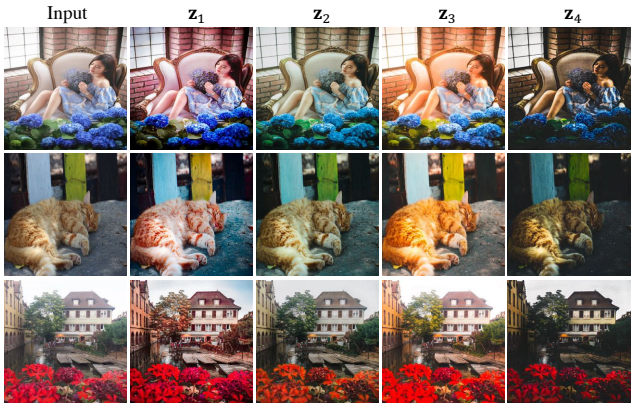

Figure 3. The multimodal image editing results controlled by different $\mathbf{z}$, each of which portrays one unique editing style.

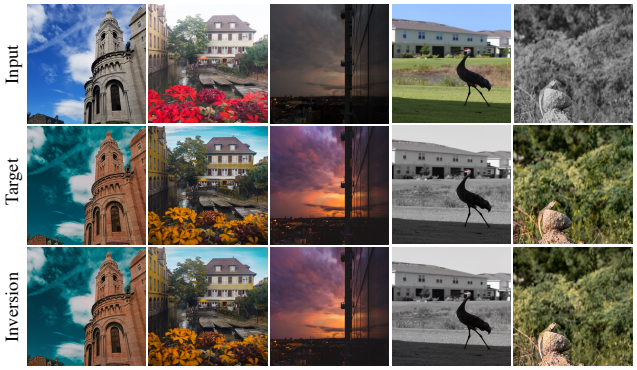

Figure 4. The visualization of conditional GAN inversion.

given different noise inputs on a single image, but also produce consistent editing styles given the same noise input on different images, indicating the independence between the learned editing space and image content.

4. Editing Space Analysis

4.1. Editing Space Inversion

Similar to StyleGAN, the $\mathcal{W}$ space of our generator is more disentangled than the input $\mathcal{Z}$ space. Therefore we rely on the $\mathcal{W}$ space as the editing space for our editing tasks. The first question is whether the style embedding for any source and target image pair can be inverted into editing space, which measures the completeness and upper-bound editing ability of the $\mathcal{W}$ space. To answer this question, we propose a *conditional GAN inversion* problem: finding a $\mathbf{w}$ that can transfer the source image I_{in} to the target I_{tgt}. We adapt an existing unconditioned GAN inversion method [55] to solve this problem, as formulated in Eq. (2)

$$\mathbf{w}, \mathbf{n} = \arg\min_{\mathbf{w},\mathbf{n}} \mathcal{L}_{\text{LPIPS}}(I_{tgt}, G(I_{in}, \mathbf{w}, \mathbf{n})) + \lambda_n \mathcal{L}_n(\mathbf{n}), \tag{2}$$

where $\mathbf{w}$ and $\mathbf{n}$ are the inverted latent code and stochastic noise inputs to different layers of the decoder, respectively. $\mathcal{L}_{\text{LPIPS}}$ is the LPIPS perceptual loss [52] and $\mathcal{L}_n$ denotes the noise regularization term [20] with λ_n as a balance weight. We show some randomly picked inversion results in Fig. 4. It is clear that our editing space $\mathcal{W}$ can represent diverse editing styles such as drastic color manipulation, colorization, and local editing, which are useful for various downstream tasks. Besides qualitative results, we also show the quantitative result of reconstruction errors on both training and testing datasets in Tab. 1. With inverted w, the outputs from our generator can almost reconstruct the target images perfectly with negligible ∼4 pixel errors, indicating the completeness of our learned editing space.

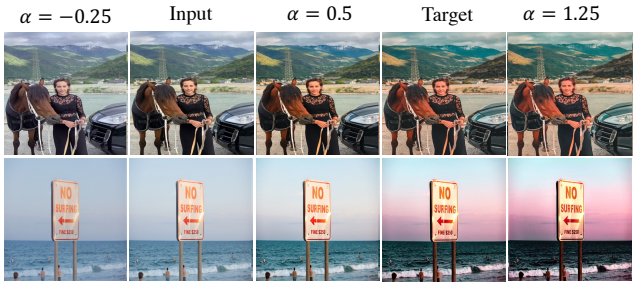

Figure 5. From the left to right, the strength the editing style increases.

Inversion	Train	Test
Init	24.88	24.93
$\mathbf{w}$	4.43	4.43
$\mathbf{w}_0$	1.86	1.86

Table 1. Init, $\mathbf{w}$, $\mathbf{w}_0$ measure the *mean pixel absolute error (maximum 255)* between source and target image, inverted and target image, source and reconstructed source image, respectively.

4.2. Interpolation

A special case of the conditional GAN inversion, which has not been investigated in the previous literature, is to find a latent code $\mathbf{w}_0$ that can reconstruct the source image itself. Such latent code has some semantic meaning in terms of editing as it represents the unchanged status of the source image. We can find its embedding by simply replacing the I_{tgt} term with I_{in} in Eq. (2). The reconstruction error on the testing dataset is less than 2 pixel difference as shown in Tab. 1.

With the help of $\mathbf{w}_0$, we can control the strength of an arbitrary editing style $\mathbf{w}$ by using their linear interpolations as $\mathbf{w}' = (1-\alpha)\mathbf{w}_0 + \alpha\mathbf{w}$, where α is a factor to control the strength of editing. Some examples are shown in Fig. 5.

4.3. Other Properties

We further demonstrate the editing capability and recognition capability of $\mathcal{W}$ space. For editing capability, as Fig. 3 reveals that each $\mathbf{w}$ shows a consistent style for different images, enabling the transfer of $\mathbf{w}$ inverted from one image pair to other images to achieve similar editing style, indicating its *transferability* property as shown in Fig. 12, detailed in Sec. 6.3.2. For recognition capability, we demonstrate that the latent codes representing similar editing styles

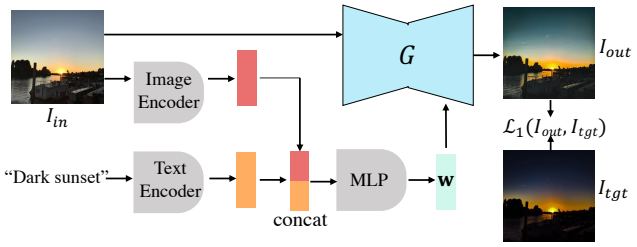

Figure 6. The structure for supervised LGIE. Only gray shaded module are trained while the generator is frozen.

are distributed closely in $\mathcal{W}$ space by studying the retrieval and cluster performance in $\mathcal{W}$ space (see Sec. 6.2), showing that the latent code has the intrinsic capability to be used for recognize the editing style.

5. Language-Guided Image Editing

To show the advantage of our pretrained network on the downstream tasks, we firstly show the language-guided image editing (LGIE) by leveraging our pretrained model. Other downstream tasks are illustrate in Sec. 6.3. Given an image I, and a language editing request r, LGIE aims to generate a new image following the editing request. Language is a convenient way to incorporate user's editing intention, which is a more intuitive and convenient interface than existing operation-based editing interfaces. Given our pretrained generator, we solve the LGIE tasks by finding a mapping between the text input and our low-dimensional editing space, which is a different framework compared to previous works [2, 4, 7, 9, 18, 23, 26, 29, 30, 38, 39, 50]. Next, we describe our approaches for both supervised LGIE as well as zero-shot LGIE.

Supervised LGIE. The supervised LGIE directly learns the mapping from language to the $\mathcal{W}$ space from the data triplet consisting of the input image, target image, and language request. The structure of the model is shown in Fig. 6, where the image and text feature are merged by concatenation, followed by a Multilayer Perception (MLP) to predict a latent code $\mathbf{w}$. Given $\mathbf{w}$, the generator serves as a render to generate the output image with the designated style. The training is driven by the L1 loss between the output image and target image, written as $\mathcal{L}_1(I_{out}, I_{tgt})$. The generator G is frozen while the other parameters are trained. Our novel learning framework could be potentially useful for other image editing tasks with paired supervision, such as supervised image harmonization, which will be left for future study.

Zero-Shot LGIE. Inspired by StyleCLIP [32], we propose to use the pretrained image-text CLIP model [34] to directly find a latent code $\mathbf{w}$ given an editing request r through optimization. Specifically, given the CLIP visual encoder f_v and textual encoder f_t, the latent code $\mathbf{w}$ is optimized by

$$\arg\min_{\mathbf{w}} - \langle f_v(G(I, \mathbf{w})), f_t(r) \rangle - \lambda \langle f_v(G(I, \mathbf{w})), f_v(I) \rangle , \quad (3)$$

Figure 7. The multimodal image editing performance compared with other methods

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity and λ a balance weight. Its first term enforces the CLIP similarity between the generated image and the request. The second term drives the similarity of the generated image to the original image. Since the CLIP model is trained on billions of image-text pairs and thus understands free-form language, this approach is generic for open-vocabulary requests.

Moreover, to achieve precise local editing, our approach can accepts as input an additional binary mask M to indicate the editing foreground and background. Given an editing request, we can simply replace the term $G(I, \mathbf{w})$ in Eq. (3) with $M \odot G(I, \mathbf{w}) + (1 - M) \odot I$, where $\odot$ is Hadamard product.

6. Experiments

We evaluate the pretraining task, $\mathcal{W}$ properties, and downstream tasks in this section. Due to space limitation, we put the implementation details in Appx. B.

6.1. Multimodal Image Editing

Dataset. We use the Adobe Discover dataset collected from the Adobe Discover website, where Lightroom users upload their edited images along with editing operations. This paired dataset contains open-domain images with various editing styles, focusing on color and tone retouching while not changing image content, geometry, or texture. Given the large number of active users, totally 62416 before-and-after image pairs are collected with the split of 49932/6242/6242 for train/val/test.

Metrics. *Fréchet Inception Distance* (FID) [14] measures

	FID↓	LPIPS ↑
BiCycleGAN [56]	12.2837	0.0857
DivCo [25]	9.9586	0.1705
Ours	**5.1755**	**0.1945**
Ours shallow	6.0958	0.1581
Ours comod [54]	5.6355	0.1479

Table 2. Quantitative results of multimodal image editing on Discover dataset.

the quality and diversity of a set of generated images compared to the set of real images through the feature computed from an Inception network [41]. LPIPS [53] measures the diversity of an image set by computing the average feature distance of all pairs of images, following [55]. We generate 10 random outputs for one input to compute LPIPS.

Comparison methods. *BiCycleGAN* [56] learns the mapping from the output image to the input noise to encourage diversity. *DivCo* [25] follows the structure of BiCycleGAN but adds the contrastive loss to encourage better diversity.

Result analysis. Our algorithm surpasses BiCycleGAN and DivCo by a large margin according to FID, mainly due to the benefit of the StyleGAN-like structure. And as indicated in [54], the modulation-based conditional generator is intrinsically stochastic w.r.t. the input noise even without explicit diversity constraint used in [25, 56]. The qualitative comparison in Fig. 7 shows that our model can create more diversified editing styles, while the BicycleGAN and DivCo will only generate images in a single editing style with different degrees. Moreover, we sample the same $\mathbf{z}$ for different images in Fig. 3, showing that the same $\mathbf{z}$ ($\mathbf{w}$) has global consistency for all the images.

Ablation Study of the network structure. Firstly, since the study of Sec. 6.2 suggests that our editing space takes most effect at high-resolution layers of the decoder, we remove the deeper layers of both encoder and decoder and only keep the layers sensitive to $\mathbf{w}$, so as to reduce the model size. We denote such setting as *Ours shallow*, whose performance in Tab. 2 is worse than the standard setting. Therefore, it proves that the depth of the network is still critical for editing performance.

Moreover, our standard network is only modulated by the noise input, while it also can be co-modulated by the feature extracted from the input image, similar to the structure of [54]. We therefore compare this setting as *Ours comod* in Tab. 2. However, the performance for co-modulation drops. One possible reason is that the image modulation features bring some input-constrained information which impairs the editing quality and stochasticity.

6.2. Latent Space Analysis

We analyze the semantics of the editing space $\mathcal{W}$ with the following experiments.

Semantic disentanglement. Given the line of works [6, 37, 42, 43, 45] tackling unsupervised GAN latent semantic discovery, we adopt *Semantic Factorization* (SeFa) [37] for the sake of simplicity. Some discovered principal semantic direction is visualized in Fig 8, showing that the editing space $\mathcal{W}$ can be disentangled.

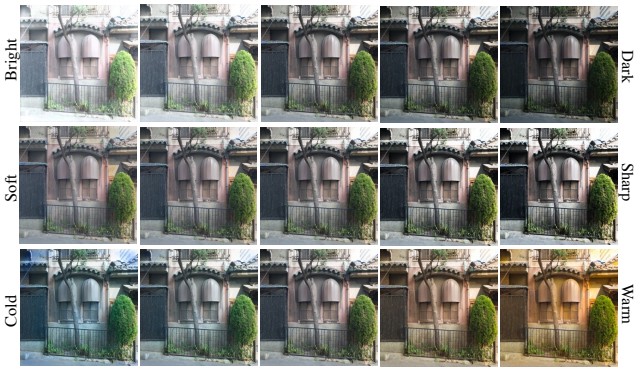

Figure 8. The visualization for unsupervised latent direction discovery using SeFa. The center column is the input image, and each row is the traverse through one SeFa principle direction across w_0.

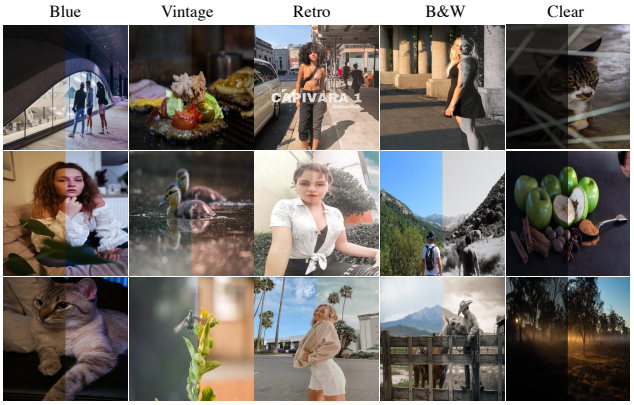

Figure 9. The clustering of the dataset using w. For each image, the left half is the before-image, and the right half is the after-image.

Layerwise effect of $\mathbf{w}$. Similar to StyleGAN, our $\mathbf{w}$ applies to different layers of the decoder. So we further analyze its layerwise semantics using SeFa. We find that the editing is only caused by the $\mathbf{w}$ on high-resolution layers, while the effect of $\mathbf{w}$ in low-resolution layers is not obvious. Concretely, $\mathbf{w}$ is most effective for the top 6 out of 14 layers in the decoder for 256x256 resolution input. This is reasonable since our model focus on color manipulation which is typically controlled via the top layers of the StyleGAN [47]. However, we cannot tell obvious semantic differences among the top layers, as shown in Appx. C, which might be because the color adjustment is already located in a fine-grained subspace.

Retrieval capability. Next, we assess the distribution of different editing styles in the editing space $\mathcal{W}$. We conduct k-nearest neighbor (KNN) search in the database using

	Lr Operation	$\mathcal{W}$ (ours)	$\mathcal{W}$ (euc)
Purity ↑	4.25	**12.76**	11.30

Table 3. Quantitative clustering results on Discover dataset. Euc denotes cluster using euclidean distance.

	L1 ↓	SSIM↑	FID↓	$\sigma_{\times 10^2}$ ↑
Input	0.1190	0.7992	12.3714	-
T2ONet [39]	0.0784	0.8459	6.7571	**0.7190**
EDNet [18]	-	-	9.9500	-
Ours	**0.0731**	**0.8721**	**5.9791**	0.6809
Ours w/o vis	0.0795	0.8596	6.9757	0.6281

Table 4. Quantitative results on MA5k-Req test sets. $\sigma_{\times 10^2}$ denotes the image variance scaled by 100 times.

inverted $\mathbf{w}$ with cosine distance. Given a pair of before-and-after images as query, the retrieved KNN image pairs carry the similar editing style, shown in Fig. 1 (see more in Appx. D.1) The retrieval result illustrates that the similarity in the $\mathcal{W}$ space measures the similarity of editing style.

Clustering capability. Inspired by the retrieval result, it uncovers another simple way for latent semantic discovery – cluster in the $\mathcal{W}$ space and regard each cluster center as an editing style. We employ K-means algorithm with cosine distance for clustering. To evaluate the cluster performance, ideally we need to annotate the style class for each editing pair. However, as the editing styles in the dataset are diversified and compositional, a predefined list of style tags might be short-sighted. So we instead annotate a complete sentence that describes the edit, allowing novel styles to be included. Then we create a style tag list including both common styles and the novel styles mentioned in the labeled sentences. Next, we evaluate the clustering performance by purity which is a measure of the extent to which clusters contain a single class. As the standard purity only considers the data sample with the single-class label, while our sample (image pair) bears multiple style tags. Hence we customize the computation of purity in Appx. H.

For comparison, as the Adobe discover dataset also contains the ground-truth Lr operation parameter, we compare our editing space with the Lr operation space. The result shown in Tab. 3 indicates that our editing space has better semantics to represent styles than the Lr operation space. Moreover, we compare the default cosine distance with the euclidean distance and find that the cosine distance is better. Fig. 9 shows the representative tag for some clusters. Due to space limit, the details for the tag list and annotation process are in Appx. G and F.

6.3. Downstream Tasks

6.3.1 Language-guided image editing

Experimental settings. For supervised LGIE, we follow the experiment setting of [39] on the MA5K-Req [39] dataset. The evaluation metrics are L1, SSIM, FID, and image variance σ. Due to the space limit, we put the detailed description and more comparison methods in Appx E.1. We show two SOTA comparison methods *T2ONet* [39] and *EDNet* [18] that both designed for global image editing, as well as a base evaluation between the input and output images denoted as *Input*.

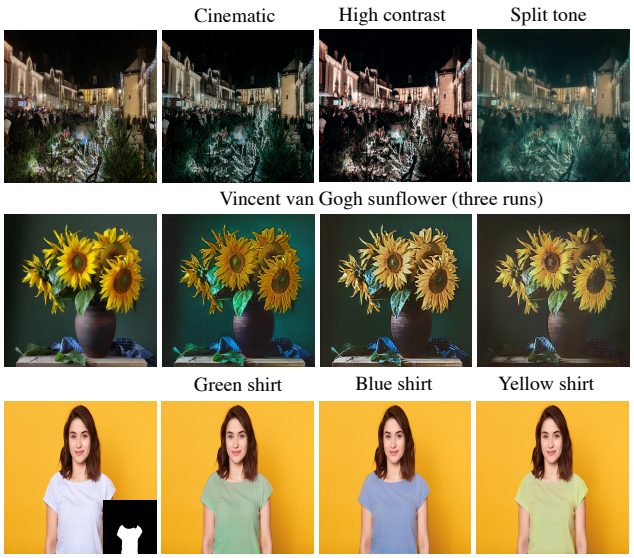

Figure 10. The open-vocabulary, open-image, language-guided image editing samples optimized by CLIP. The last row show the local editing with mask input.

For zero-shot LGIE, as it works for open-domain image and open-vocabulary requests, we compare the qualitative performance on given examples with two other SOTA methods – OpenEdit [26] and StyleCLIP [32]. OpenEdit has no constraint for both image and request, while StyleCLIP can only work for close-domain images.

Result analysis. For the supervised LGIE, the performance is shown in Tab. 5, showing that our method achieves the best editing quality and comparable variance as T2ONet. Given the strong editing ability of the pretrained generator, the LGIE task becomes easier because the model only needs to predict a latent code of 512 dimensions instead of the entire image space. Moreover, we study whether the language input alone is sufficient to predict the latent code. We denote the setting without image input as *ours w/o viz* shown in Tab. 5, which shows inferior results to the standard setting, thus suggesting the importance of the visual input.

For the zero-shot LGIE, we firstly show our result in Fig. 1 and 10, indicating that our model can achieve the editing with the diversified directive of high-level semantic (aurora), editing terminology (split tone), color manipulation (green shirt), or even some texture change (Van Gogh painting). Furthermore, the comparison with the SOTA is drawn in Fig. 11. StyleCLIP completely fails in these cases because it does not work for open-domain images. The face

Figure 11. The open-vocabulary, open-image, language-guided image editing samples optimized by CLIP with comparison to other methods.

will pop up due to the memory of its generator pretrained on face dataset. Despite that OpenEdit can accept open-domain images, its editing does not follow the request well, and the output image contains obvious artifacts. In contrast, our method can handle these cases well. Despite imperfect, our model has the potential to achieve gray image colorization while other methods cannot.

6.3.2 Personalized Editing and Recommendation

Given a user-edited before-and-after image pair as an exemplar, our model can achieve both personalized editing and editing style recommendation. For personalized editing, we study exemplar-based image editing (EBIE), which is to edit the input image following the editing style of the user preferred exemplar. This task can be naturally tackled by the transferability property (Sec. 4.3) of the $\mathcal{W}$ space without training. When there are multiple exemplars with consistent styles, we can find a common editing direction by averaging the latent code of all the exemplars. We compare our approach with the Lr preset, which is a set of Lr operations that can also be applied to other images to achieve a similar editing effect. The visualization of the comparison is shown in Fig. 12, indicating our transfer result is reasonable and visually comparable with the Lightroom preset. However, the preset approach must know the exact preset parameters of the exemplar images, while our method is free from such constraint and thus is more general. Moreover, different from the photorealistic style transfer [49] where the color and texture of the reference image

Figure 12. The visualization of the exemplar-based image editing. The left of the dash line are exemplars and the right is the transferred editing.

is directly transferred to the source image, our EBIE tries to transfer the relative editing style. Taking the first row of Fig. 12 as an example, our method transfer the "brighten" effect instead of the green color to the other image.

Editing style recommendation is to recommend the image pairs with similar editing styles to a given image pair. This task is beneficial for the photography pedagogy if a user wants to see multiple photo examples of the same editing style for specialized learning. Such task can be handled via the retrieval capability in the $\mathcal{W}$ space, as illustrated in Sec. 6.2. The visualization is shown in Appx. D.1.

7. Conclusion and Discussion

This paper introduces a new image editing paradigm: learn a pretrained I2I generator with an editing space that can work as a unified interface to bridge multiple downstream tasks. We find the editing space is well disentangled and complete for color editing, which can be used for both editing and recognition. Experiments on the downstream tasks prove the advantages of our pretrained model.

Limitation. Our method relies on the Adobe Discover dataset and thus cannot be expected to manipulate image content (*e.g.* geometric change) or texture (though we have shown some particular texture changes in painting style, they are not general). For LGIE, a faithful image manipulation is not guaranteed if the text requests are mapped to the CLIP space where images are not well populated.

Potential Negative Impact. Our model might be maliciously used to generate fake photos to forge criminal evidence, *e.g.*, daytime to night. Therefore we keep the user's identity and editing history to monitor misuse.

Acknowledgement. This work has been partially supported by the National Science Foundation (NSF) under Grant 1909912 and by an Adobe research gift. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020. 3

[2] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 3, 5

[3] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 2011. 12

[4] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8721–8729, 2018. 3, 5

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[6] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, 2020. 2, 3, 6

[7] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *ICCV*, 2017. 3, 5, 12

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[9] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *ICCV*, 2019. 3, 5, 12

[10] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019. 3

[11] Zonghui Guo, Dongsheng Guo, Haiyong Zheng, Zhaorui Gu, Bing Zheng, and Junyu Dong. Image harmonization with transformer. In *ICCV*, 2021. 1

[12] Jingwen He, Yihao Liu, Yu Qiao, and Chao Dong. Conditional sequential modulation for efficient global image retouching. In *ECCV*, 2020. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 11

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 5

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 3

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 3, 12

[17] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *ICLR*, 2020. 3

[18] Wentao Jiang, Ning Xu, Jiayun Wang, Chen Gao, Jing Shi, Zhe Lin, and Si Liu. Language-guided global image editing via cross-modal cyclic mechanism. In *ICCV*, 2021. 1, 2, 3, 5, 7, 12

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 3

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2, 4, 11

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 11

[22] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 3

[23] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *CVPR*, 2020. 1, 3, 5

[24] Zhiheng Li, Martin Renqiang Min, Kai Li, and Chenliang Xu. Stylet2i: Toward compositional and high-fidelity text-to-image synthesis. In *CVPR*, 2022. 3

[25] Rui Liu, Yixiao Ge, Ching Lam Choi, Xiaogang Wang, and Hongsheng Li. Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In *CVPR*, 2021. 3, 6

[26] Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In *ECCV*, 2020. 1, 3, 5, 7

[27] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2

[28] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019. 3

[29] Xiaofeng Mao, Yuefeng Chen, Yuhong Li, Tao Xiong, Yuan He, and Hui Xue. Bilinear representation for language-based image editing using conditional generative adversarial networks. In *ICASSP*, 2019. 3, 5, 12

[30] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. *arXiv preprint arXiv:1810.11919*, 2018. 3, 5, 12

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 11

[32] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 2, 3, 5, 7

[33] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020. 3

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2, 3, 5, 11

[35] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 3

[36] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2, 3

[37] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, pages 1532–1540, 2021. 2, 3, 6, 11

[38] Jing Shi, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu. A benchmark and baseline for language-driven image editing. In *ACCV*, 2020. 3, 5

[39] Jing Shi, Ning Xu, Yihang Xu, Trung Bui, Franck Dernoncourt, and Chenliang Xu. Learning by planning: Language-guided global image editing. In *CVPR*, pages 13590–13599, 2021. 1, 2, 3, 5, 7, 12

[40] Yuda Song, Hui Qian, and Xin Du. Starenhancer: Learning real-time and style-aware image enhancement. In *ICCV*, 2021. 1

[41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[42] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, 2020. 2, 3, 6

[43] Binxu Wang and Carlos R Ponce. The geometry of deep generative image models and its applications. In *ICLR*, 2021. 3, 6

[44] Hai Wang, Jason D Williams, and SingBing Kang. Learning to globally edit images with textual description. *arXiv preprint arXiv:1810.05786*, 2018. 12

[45] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021. 2, 3, 6

[46] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *CVPR*, 2021. 2, 3

[47] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *IJCV*, 2021. 6

[48] Jonghwa Yim, Jisung Yoo, Won-joon Do, Beomsu Kim, and Jihwan Choe. Filter style transfer between photos. In *ECCV*, 2020. 1

[49] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *ICCV*, 2019. 1, 8

[50] Xiaoming Yu, Yuanqi Chen, Thomas Li, Shan Liu, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. *arXiv preprint arXiv:1909.07877*, 2019. 3, 5

[51] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 1

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[54] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 3, 6

[55] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 3, 4, 6

[56] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Multimodal image-to-image translation by enforcing bi-cycle consistency. In *NeurIPS*, 2017. 3, 6