L2M-GAN: Learning to Manipulate Latent Space Semantics for Facial Attribute Editing

Guoxing Yang[1,2] Nanyi Fei[1] Mingyu Ding[3] Guangzhen Liu[1] Zhiwu Lu[1,2] Tao Xiang[4]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]The University of Hong Kong [4]University of Surrey, UK
luzhiwu@ruc.edu.cn

Figure 1. Attribute editing results by our L2M-GAN on CelebA-HQ. The first column shows the real source images, and each of the other columns shows the results of editing a specific attribute. Each edited image has an attribute value opposite to that of the source one.

Abstract

A deep facial attribute editing model strives to meet two requirements: (1) attribute correctness – the target attribute should correctly appear on the edited face image; (2) irrelevance preservation – any irrelevant information (e.g., identity) should not be changed after editing. Meeting both requirements challenges the state-of-the-art works which resort to either spatial attention or latent space factorization. Specifically, the former assume that each attribute has well-defined local support regions; they are often more effective for editing a local attribute than a global one. The latter factorize the latent space of a fixed pretrained GAN into different attribute-relevant parts, but they cannot be trained end-to-end with the GAN, leading to sub-optimal solutions. To overcome these limitations, we propose a novel latent space factorization model, called L2M-GAN, which is learned end-to-end and effective for editing both local and global attributes. The key novel components are: (1) A latent space vector of the GAN is factorized into an attribute-relevant and irrelevant codes with an orthogonality constraint imposed to ensure disentanglement. (2) An attribute-relevant code transformer is learned to manipulate the attribute value; crucially, the transformed code are subject to the same orthogonality constraint. By forcing both the original attribute-relevant latent code and the edited code to be disentangled from any attribute-irrelevant code, our model strikes the perfect balance between attribute correctness and irrelevance preservation. Extensive experiments on CelebA-HQ show that our L2M-GAN achieves significant improvements over the state-of-the-arts.

1. Introduction

Facial attribute editing [45, 48, 54, 3, 4, 29, 47], i.e., manipulating the semantic attributes of a real face image, has a wide range of real-world application scenarios

such as entertainment, auxiliary psychiatric treatment, and data augmentation for other facial tasks. With the tremendous success of deep generative models [12, 37, 26], facial attribute editing has become topical in recent works [33, 17, 52, 16, 27, 46, 58], most of which are based on generative adversarial networks (GANs) [12].

One of the main challenges for facial attribute editing is to meet two requirements simultaneously: (1) attribute correctness – the target attribute should correctly appear on the edited image; (2) irrelevance preservation – the irrelevant information (e.g., identity, or other attributes) should not be changed during attribute editing. However, meeting both requirements is hard because there often exist strong correlations between different attributes (e.g., moustache and gender) as well as between attributes and identity [16, 58]. As a result, editing one attribute may result in unintended altering of other characteristics of the face image.

To achieve attribute correctness whilst avoiding unintended altering, many recent methods [16, 27] resort to spatial attention. The assumption is that each attribute has local support regions which can be modeled using an attention module on feature maps of an encoder-decoder GAN framework. Once these support regions are identified, image manipulation can be restricted to those regions thus stopping unwanted changes in other regions. This assumption is valid for some local attributes such as bangs or glasses. It is however problematic when it comes to global attributes such as smiling/gender/age, for which support regions are global and overlapping between attributes is inevitable.

Another recent line of approach is to focus on the factorization of the latent space learned by a face synthesis GAN into attribute-relevant latent codes [46, 58]. Given a fixed pretrained GAN, the latent space vector is mapped to each attribute via subspace projection. However, there are two issues with this approach: (a) It relies on a fixed pretrained GAN to provide the latent space. Without end-to-end training with the factorization model, this latent space could be sub-optimal. (b) Guided by semantic labels, it can only disentangle different semantic attributes from each other. However, there are also other characteristics of a face image that are not described by a set of pre-defined attributes, e.g., identity and lighting condition.

To overcome the limitations of the current state-of-the-arts [16, 27, 46, 58], we propose a novel latent space factorization model, called learning-to-manipulate GAN or L2M-GAN, which is learned end-to-end and effective for editing both local and global attributes (see Figure 1). Similar to [46, 58], our L2M-GAN model is designed to factorize a GAN latent space into semantic codes guided by attribute annotations, without imposing any spatial constraints on feature maps as in [16, 27]. Differently, for each attribute, both attribute-relevant and -irrelevant codes are factorized explicitly. Moreover, the disentanglement between the two codes are enforced both before and after the editing. Concretely, inspired by the latest StarGAN v2 [6], we apply a style encoder on the input image to obtain the source style/latent space code. Further, we devise a new style transformer which is the key component for facial attribute editing. It is composed of two main modules: (1) a decomposer for disentangling the source style code into two orthogonal parts – an attribute-relevant code and everything else in an attribute-irrelevant code; (2) a domain transformer for transforming the attribute-relevant code from its original value/domain to a target one (e.g., unsmiling to smiling). Crucially, the transformed code is also subject to the orthogonality constraint w.r.t. the factorized attribute-irrelevant code. In this way, the attribute correctness and irrelevance preservation requirements are fulfilled explicitly in our L2M-GAN. Further, unlike the latest works [46, 58], our L2M-GAN can now be trained end-to-end.

Our main contributions are three-fold: (1) For the first time, we propose an end-to-end GAN model namely L2M-GAN for facial attribute editing by explicitly factorizing the latent space vector of a GAN into attribute-relevant and -irrelevant codes. (2) To facilitate the latent space factorization, we devise a novel style transformer by imposing the orthogonality constraint on the factorized attribute-relevant and -irrelevant codes both before and after the editing/transformation. (3) Extensive experiments on CelebA-HQ [23] show that our L2M-GAN achieves significant improvements over the state-of-the-arts. Importantly, once learned, our L2M-GAN has a wide use in other attribute manipulation tasks (e.g., attribute strength manipulation and manipulation with reference images) without re-training, and also generalizes well from photo to anime faces.

2. Related Work

Generative Adversarial Networks. Since its introduction in [12], GAN has attracted much attention [59, 42, 36, 8, 11, 21, 49] due to its powerful ability to generate photo-realistic outputs. Many variants of GANs were proposed to guarantee the training stability [1, 13, 43, 50] and improve the synthesis quality [40, 23, 55, 2, 24]. Apart from image synthesis with unconditional GANs, conditional methods [20, 39] were attempted for image-to-image translation. CycleGAN [60] proposed to utilize the cycle consistency loss to overcome the lack of paired training data. Instead of single domain transfer, StarGAN [5] and StarGAN v2 [6] coped with image translation among multiple domains. Due to these advances, GANs have recently been leveraged in a variety of real-world applications such as image inpainting [51, 53], image super-resolution [30, 56, 22], facial attribute editing [45, 48, 54], and medical image generation [10, 41].

Facial Attribute Editing. As a typical yet challenging generative task, facial attribute editing has been dominated by

GAN based methods [33, 31, 9, 16, 27, 28, 17, 57, 48, 46, 58]. Depending on what intermediate information is used for facial attribute editing, existing GAN based methods can be roughly divided into two groups: (1) **Editing over Feature Maps**: STGAN [33] employed an attribute difference indicator and then performed selective transfer over the feature maps of encoder-decoder for attribute editing. MaskGAN [31] exploited the semantic masks of the input image for flexible face manipulation with fidelity preservation. WrapGAN [9] learned the smooth wrap fields for photo-realistic attribute editing. PA-GAN [16] and CAFE-GAN [27] applied spatial attention to obtain local support regions pertinent to the attribute and then conduct the attribute editing inside these regions. As a result, they are often more suitable for a local attribute (e.g., bangs) than a global one (e.g., smiling). (2) **Editing over Latent Space**: Fader Network [28] leveraged adversarial training to disentangle attribute-related latent factors/codes from the latent space. AttGAN [17] modeled the relation between attributes and the latent space learned by a GAN. GeneGAN [57] and ELEGANT [48] exchanged attribute between two faces by swapping the attribute-related latent codes. InterfaceGAN [46] and In-Domain GAN Inversion [58] focused on the interpretation of the semantics of the latent space of GANs via subspace projection. In this work, our proposed L2M-GAN belongs to the second group, and relies on latent space factorization, similar to [46, 58]. However, different from [46, 58] that only factorize between different attribute-related codes, for each attribute, our factorization disentangles a latent code into an attribute-relevant and -irrelevant codes, both before and after attribute editing in the latent space. Crucially, this enables end-to-end training of both our factorization module and the GAN module, avoiding sub-optimal solutions suffered by [46, 58] which used a fixed pretrained GAN for factorization. Note that the end-to-end training of a latent space factorization module and an image generator has been attempted recently in [32]. However, the generator used in [32] is based on autoencoder and thus more limited in generated image quality compared to GAN; the factorization is again based on subspace projection and thus limited to between-attribute disentanglement as in [46, 58].

3. Methodology

3.1. Problem Formulation

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the set of real input images and the set of possible domains (values of multiple attributes), respectively. Since an attribute typically has multiple values (i.e. domains), the size of $\mathcal{Y}$ is bigger than the number of attributes. For example, given a set of attributes {smiling, gender}, the set of possible domains are {smiling and male, unsmiling and male, smiling and female, unsmiling and female}. For each input/source image $x \in \mathcal{X}$, its source domain (i.e., source attribute value) is denoted as $y \in \mathcal{Y}$. Given a target domain $\tilde{y} \in \mathcal{Y}$, the goal of a facial attribute editing model is to train an image style translation function denoted as T, to synthesize a new/target image $\tilde{x}$, changing only the domain from y to $\tilde{y}$ but preserving the domain-irrelevant information (e.g., identity or other attributes). Such a facial attribute editing model can be formally defined as $\tilde{x} = T(x, y, \tilde{y})$.

3.2. Our L2M-GAN Model

As illustrated in Figure 2, our L2M-GAN model consists of three modules: style encoder, style transformer, and generator. For easier understanding of our L2M-GAN model as well as notation simplicity, we set $|\mathcal{Y}| = 2$, i.e., only a single attribute (with two binary values) is considered. Adopting StarGAN v2 [6] as the backbone which is designed for multi-domain style transfer, our L2M-GAN model can be readily extended to multiple attribute manipulation with a single model (see results in Sec. 4.5).

Style Encoder. Under the multi-task learning setup [5, 6], our style encoder *SE* is composed of multiple output/domain branches for $\mathcal{Y}$. Again, for clarity of presentation, in Figure 2, only one domain/branch is shown for our style encoder. Formally, given an input image $x \in \mathcal{X}$ with its source domain $y \in \mathcal{Y}$, the style code extracted by our style encoder *SE* is denoted as $s = SE_y(x) \in \mathbb{R}^d$, where d is the dimension of the style code. In this paper, our style encoder is the same as in StarGAN v2 [6].

Style Transformer. As stated in [46, 7], the style code $s \in \mathbb{R}^d$ extracted by the style encoder *SE* contains rich semantic information of the input image x. Some information is relevant to the attributes of interest, while the other is not. It is thus necessary to conduct factorization on the d-dimensional latent space to disentangle it into two parts. Concretely, the style code s can be decomposed into $s_{re} \in \mathbb{R}^d$ and $s_{un} \in \mathbb{R}^d$, where s_{re} denotes the style code relevant to y and s_{un} denotes the style code unrelated/irrelevant to y. Since only the domain-related style code s_{re} will be transformed during editing, we must produce an output style code $\tilde{s}$ for the target domain $\tilde{y}$ without changing the other characteristics of the input image. This is possible only if s_{re} and s_{un} are disentangled. To this end, we introduce an orthogonality loss on these two code vectors (to be formulated in Sec. 3.3). Note that orthogonality is also exploited in previous latent space factorization methods based on subspace projection [46, 58, 32]. However, they focus on *between-attribute disentanglement*, while we separate attribute-relevant style code from the irrelevant code (i.e., everything else including other attributes as well as identity information, lighting condition etc). This is vital for irrelevance preservation as other attributes do not cover all irrelevant information of the input image.

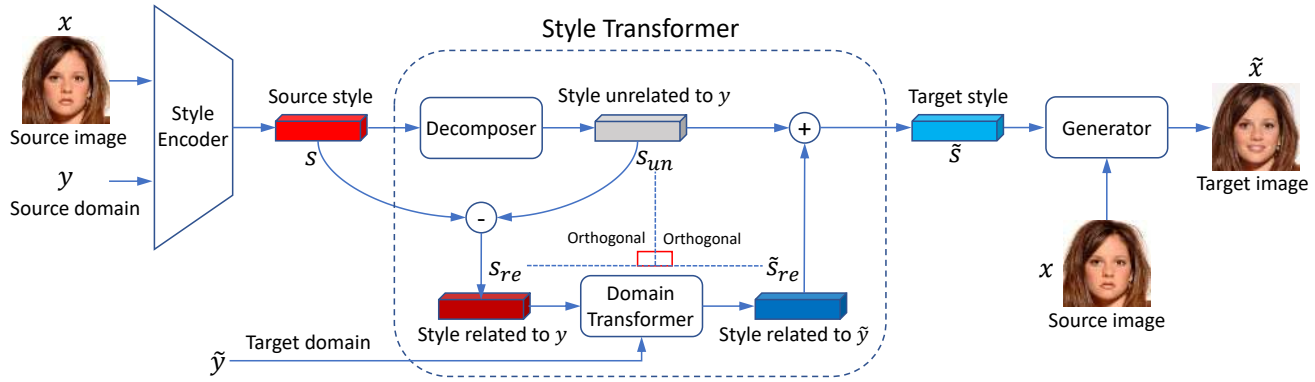

Figure 2. A schematic illustration of the proposed L2M-GAN model. Our novel Style Transformer is the key component for facial attribute editing, which is composed of a decomposer and a domain transformer. Each cuboid denotes a style code: the red/blue ones are relevant to the source/target domain of a given attribute (smiling here), and the grey one is attribute-irrelevant code. The whole attribute editing model is trained in a GAN framework with a discriminator (not shown here) on the generated target images.

Formally, given an input/source style code s, our style transformer *ST* transforms it into an output/target style code $\tilde{s} = ST(s, \tilde{y})$, where $\tilde{y} \in \mathcal{Y}$ denotes the target domain. Firstly, a decomposer P is applied to the style code s to extract the domain-irrelevant style code $s_{un} = P(s)$, resulting in $s_{re} = s - s_{un}$. Secondly, the style code s_{re} related to the source domain y is transformed into $\tilde{s}_{re} = DT(s_{re}, \tilde{y})$ by a domain transformer *DT*, where $\tilde{s}_{re}$ is the style code related to the target domain $\tilde{y}$. Since we have excluded the attribute-irrelevant style information from s_{re}, $\tilde{s}_{re}$ can be obtained without changing any information unrelated to the source/target domain. To further make sure that, the same orthogonal loss is added between $\tilde{s}_{re}$ and s_{un} (to be formulated in Sec. 3.3). The style transformer *ST* is defined as:

$$\tilde{s} = ST(s, \tilde{y}) = DT(s - P(s), \tilde{y}) + P(s). \quad (1)$$

Generator. Our generator G takes an image x and the transformed style code $\tilde{s}$ as input, and generates the output image $\tilde{x} = G(x, \tilde{s})$ which reflects the information of the style code $\tilde{s}$. Similar to StarGAN v2 [6], we adopt adaptive instance normalization (AdaIN) [19, 24] to transfer the information contained in the style code to the output image.

3.3. Learning Objectives

Adversarial Loss. To encourage the generator to synthesize indistinguishable images from real images, we adopt an adversarial loss. We utilize a multi-task discriminator [34, 38, 6] for making the discrimination between generated images and real images. Our multi-task discriminator D has multiple output branches, each of which learns a binary classifier to determine whether an image is real or fake w.r.t. a domain. The adversarial loss is given by:

$$\mathcal{L}_{adv} = \mathbb{E}_{x,y,\tilde{y}}\big[\log D_y(x) + \log(1 - D_{\tilde{y}}(G(x, \tilde{s})))\big], \quad (2)$$

where $D_y(\cdot)$ denotes the output branch of D w.r.t. the domain y. The style encoder *SE* and style transformer *ST* are learned to provide the style code $\tilde{s}$ w.r.t. the target domain $\tilde{y}$. With the image x and the style code $\tilde{s}$, G is learned to synthesize an output image $\tilde{x} = G(x, \tilde{s})$ that is indistinguishable from real images in the target domain $\tilde{y}$.

Cycle-Consistency Loss. Due to the lack of paired reference images (e.g., same person both smiling and not smiling) as supervision, we choose to utilize the cycle-consistency loss [60] defined as:

$$\mathcal{L}_{cyc} = \mathbb{E}_{x,y,\tilde{y}}\|G(G(x, \tilde{s}), s) - x\|_1. \quad (3)$$

Style Reconstruction Loss. In order to better learn both the style encoder *SE* and the generator G for image synthesis w.r.t. the style code $\tilde{s}$, we impose a style reconstruction constraint [6] on the style code extracted from $\tilde{x}$:

$$\mathcal{L}_{sty} = \mathbb{E}_{x,y,\tilde{y}}\|\tilde{s} - SE_{\tilde{y}}(G(x, \tilde{s}))\|_1. \quad (4)$$

Perceptual Loss. To enforce G to preserve the domain-irrelevant information like personal identity when generating $\tilde{x}$, we take a perceptual loss on board:

$$\mathcal{L}_{per} = \mathbb{E}_{x,y,\tilde{y}}\|F(x) - F(G(x, \tilde{s}))\|_1, \quad (5)$$

where $F(\cdot)$ denotes the feature vector outputted by a pre-trained ResNet-18 model [15].

Orthogonality Loss. To guarantee that the style code s_{un} is orthogonal to s_{re} (or $\tilde{s}_{re}$), we define the orthogonality loss to directly measure the dependence between them. The orthogonality loss is formally defined as:

$$\mathcal{L}_{ort} = \mathbb{E}_{x,y,\tilde{y}}\big[\|s_{re} \odot s_{un}\|_1 + \|\tilde{s}_{re} \odot s_{un}\|_1\big], \quad (6)$$

where $\odot$ denotes the element-wise product. Instead of simply computing the inner product, our L1-norm orthogonality loss provides a stronger constraint, which enables us to obtain a better disentangled latent space.

Overall Loss. Our overall loss can be summarized as:

$$\min_{G,ST,SE} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{sty}\mathcal{L}_{sty} + \lambda_{per}\mathcal{L}_{per} + \lambda_{ort}\mathcal{L}_{ort}, \tag{7}$$

where λ_{cyc}, λ_{sty}, λ_{per} and λ_{ort} denote the hyperparameters for balancing the above five losses.

3.4. Model Applications

In addition to face synthesis with the domain label (i.e., attribute value) as supervision, our trained L2M-GAN can be directly exploited for more facial attribute manipulation tasks (e.g., attribute strength manipulation and manipulation with reference images) in the learned disentangled latent space, without the need of re-training the whole model. Moreover, although trained on real photo images, our L2M-GAN model can also generalize to other image domains such as anime faces (see Sec. 4.5). Here, we only present the formulations of attribute strength manipulation and manipulation with reference images.

Manipulation with Different Strengths. As noticed in [44, 2, 24, 46], the synthesized images by the well-trained GANs change their appearance continuously with linear interpolations of two latent codes in the learned disentangled latent space. This suggests that the semantic information contained in the latent code can be changed gradually but without influence on the other information. Therefore, when we move the attribute-relevant style code s_{re} in the direction of $\tilde{s}_{re} - s_{re}$, the synthesized image changes only the attribute-related information but with irrelevance preservation, due to the orthogonality constraint between $\tilde{s}_{re} - s_{re}$ and s_{un} implied in the disentangled latent space. Formally, we define a new style code s_m as:

$$s_m = s_{re} + \lambda_s(\tilde{s}_{re} - s_{re}), \tag{8}$$

where λ_s denotes the strength of attribute manipulation. We then adopt λ_s as the controlling factor to synthesize a series of images with different attribute strengths:

$$x_n = G(x, s_m + s_{un}). \tag{9}$$

Manipulation with Reference Images. Given a source image x_s (with the source domain y) and a reference image x_r (with the target domain $\tilde{y}$), we extract two style codes s_s and s_r from them, respectively. After decomposing s_s and s_r independently, we merge the domain-related style code s_r^{re} of x_r and the domain-unrelated style code s_s^{un} of x_s into a new style code s_n for face synthesis:

$$\begin{aligned} s_n &= s_s^{un} + s_r^{re} \\ &= P(SE_y(x_s)) + (SE_{\tilde{y}}(x_r) - P(SE_{\tilde{y}}(x_r))). \end{aligned} \tag{10}$$

In the disentangled latent space, with the new style code s_n, we can translate the source image into the target domain of the reference image, but without changing the other information of the the source image.

4. Experiments

4.1. Dataset and Settings

Dataset. Experiments are conducted on the widely-used CelebA-HQ [23] dataset. It consists of 30,000 high quality facial images, which are picked from the original CelebA dataset [35] and processed to the size of 1024×1024 with higher quality. Each image has 40 attributes annotations inherited from the original CelebA. To obtain the training/test split, we re-index each image in CelebA-HQ back to the original CelebA and classify it into the training or test set by following the standard split of CelebA, which results in a training/test split of 27,176/2,824.

Implementation Details. For fair comparison, we resize all images to 256×256, which is the resolution used by most previous works. We set the batch size to 8 and the number of total iterations to 100K during training our L2M-GAN in an *unsupervised* way. All modules are initialized using the He initialization [14] and then trained using Adam [25] with the learning rate 1e-4, $\beta_1 = 0$ and $\beta_2 = 0.99$. The hyperparameters are empirically set as $\lambda_{cyc} = 1$, $\lambda_{sty} = 2$, $\lambda_{per} = 2$ and $\lambda_{ort} = 1$. To improve the training stability, we adopt the EMA strategy [23, 50], following StarGan v2 [6]. Our L2M-GAN is trained on PyTorch with a single TITAN RTX GPU, which takes about 40 hours to train. More details are given in the supplementary material.

4.2. Qualitative Results

Due to space constraint, we compare our L2M-GAN with the state-of-the-art methods (i.e., StarGAN [5], CycleGAN [60], ELEGANT [48], PA-GAN [16], and InterfaceGAN [46]) for a specific attribute: **Smiling**. More results on other attributes can be found in the supplementary material. Note that this attribute is one of the most challenging among the 40 facial attributes because adding/removing a smile requires high-level understanding of the input face image for modifying multiple facial components simultaneously. In our experiments, each test image is adopted as the input image to generate an output image with a domain label (i.e., attribute value) opposite to that of the input image. For StarGAN, CycleGAN, PA-GAN, and our L2M-GAN, the domain label of the input image is directly used as supervision for face synthesis. For ELEGANT, a reference image with the target domain label is randomly selected from the test set, which can then be used as supervision for face synthesis. For InterfaceGAN, we adopt the In-Domain GAN Inversion [58] as the encoder and StyleGAN [24] as the backbone, which is marked as InterfaceGAN*. As in [46], for InterfaceGAN*, the latent codes are moved to another side of the found hyperplane for face synthesis.

The qualitative results are shown in Figure 3. We have the following observations: (1) StarGAN and CycleGAN tend to generate blurs and artifacts around mouth, and thus fail to edit the smiling attribute in most cases. (2) ELE-

Figure 3. Qualitative results for facial attribute editing on the specific attribute: **Smiling**. The first column shows the real source/input images. The other columns from left to right are the editing results of StarGAN [5], CycleGAN [60], ELEGANT [48], PA-GAN [16], InterfaceGAN* [46], and our L2M-GAN. InterFaceGAN* denotes the InterfaceGAN [46] with StyleGAN [24] as the backbone and In-Domain GAN Inversion [58] as the encoder. Better viewed on-line in color and zoomed in for details.

GANT often transfers the unexpected irrelevant information from reference images because it exchanges attributes in the latent space that may be not well disentangled. (3) Based on region attention, PA-GAN preserves the irrelevant regions well but does not change the attribute value correctly due to the insufficient modification. It also tends to generate blurs and artifacts around mouth. (4) InterfaceGAN* generates high-quality images but fails in some details, e.g., eyes and teeth. It sometimes even changes the identity information of the input image due to not considering identity during factorization. (5) Our L2M-GAN manipulates the attribute correctly/naturally and produces high-quality images with sharper details, which demonstrates that it can change the attribute-relevant information correctly whilst preserving attribute-irrelevant information.

4.3. Quantitative Results

For quantitative evaluation, two evaluation metrics are adopted: attribute manipulation accuracy, and quality of generated images. In addition, user study results are also provided for subjective evaluation.

Attribute Manipulation Accuracy. Attribute manipulation accuracy is used to evaluate whether the specific attribute correctly appears on generated images after manipulation. To obtain this accuracy, we train a binary classifier for the smiling attribute on the training set using ResNet-18, which can achieve over 95% prediction accuracy on the test set. Table 1 shows our L2M-GAN outperforms all the competitors on this attribution correctness measure, despite not using an attribute classifier when training our model. StarGAN achieves relative high attribute manipulation accuracy but at the cost of image quality degradation (see the next paragraph). PA-GAN leads to the lowest accuracy (only 48.2%), showing that PA-GAN modifies images insufficiently when editing such a challenging attribute.

Image Quality. We adopt the Frechét Inception Distance (FID) [18] to evaluate the quality of generated images. FID is computed between the distribution of real images in the training set and that of the generated images (which are synthesized from all test images). Table 1 shows the FID scores of five compared methods and our L2M-GAN. We can observe that our L2M-GAN has the best average FID

Method	FID (+)	FID (-)	FID (avg)	Acc (att)
StarGAN [5]	32.6	38.6	35.6	91.6%
CycleGAN [60]	22.5	24.4	23.5	78.8%
ELEGANT [48]	39.7	42.9	41.3	74.1%
PA-GAN [16]	20.5	**21.4**	21.0	48.2%
InterFaceGAN* [46]	22.8	23.9	23.4	92.1%
L2M-GAN (ours)	**17.9**	23.3	**20.6**	**93.1%**

Table 1. Quantitative results for facial attribute editing on the specific attribute: **Smiling**. FID (+) (or FID (-)) denotes the FID score for adding (or removing) a smile, and FID (avg) denotes the simple average of FID (+) and FID (-). Acc (att) denotes the attribute manipulation accuracy.

Method	Smiling (+)	Smiling (-)	Smiling (avg)
StarGAN [5]	4.0%	6.2%	5.1%
CycleGAN [60]	7.6%	9.4%	8.5%
ELEGANT [48]	1.8%	2.6%	2.2%
PA-GAN [16]	19.4%	3.4%	11.4%
InterFaceGAN* [46]	8.2%	27.8%	18.0%
None	5.6%	10.6%	8.1%
L2M-GAN (ours)	**53.4%**	**40.0%**	**46.7%**

Table 2. User study results of facial attribute editing on the specific attribute: **Smiling**. Smiling (+) (or Smiling (-)) denotes the results of adding (or removing) a smile, and Smiling (avg) denotes the average of Smiling (+) and Smiling (-).

score, indicating that it can generate images with the highest quality. Particularly, our L2M-GAN leads to significant improvements over all the competitors for adding a smile. More quantitative results on other attributes can be found in the supplementary material.

User Study Results. We also conduct user study to evaluate the attribute editing results under human perception. Concretely, we consider adding a smile and removing a smile as two editing tasks, and randomly choose 50 test images for each task (the same setting is used for all methods). For each test image, 10 volunteers are asked to select the best generated image among those obtained by all methods, according to: 1) whether the attribute is correctly manipulated; 2) whether the irrelevant region is well preserved; 3) whether the generated image is natural and realistic. Because manipulating the smiling attribute is very challenging, we also provide a choice of "none of these methods performs well" to avoid the case where none can well manipulate the attribute but the volunteers have to choose one as the best. The images generated by different methods are shown in a random order for a fair comparison. Table 2 shows the user study results averaged over all 10 volunteers. It can be clearly seen that our L2M-GAN significantly outperforms the competitors in the user study.

4.4. Ablation Study Results

We conduct ablation study to show the contribution of our perceptual loss and orthogonality loss. We use StarGAN v2 [6] as our baseline. With a source image and a reference image as inputs, StarGAN v2 can generate a target/output image by mixing their latent codes. As a result, it often also transfers attribute-irrelevant information from the reference image, with the unintended altering of the identity of the source image. On top of StarGAN v2, we thus add various components including the orthogonal loss and perceptual loss, resulting in our full L2M-GAN. Its five simplified versions are considered: (1) BL (Baseline) – StarGAN v2; (2) BL+PL – StarGAN v2 with the perceptual loss (PL); (3) BL+PL+OL (Single) – StarGAN v2 with the perceptual loss and orthogonality loss having only the first term of Eq. (6) but defined by inner product instead; (4) BL+PL+OL (Inner) – StarGAN v2 with the perceptual loss and orthogonality loss having two terms of Eq. (6) but defined by inner product instead; (5) BL+OL (L1) – StarGAN v2 with only the orthogonality loss having two L1-norm terms of Eq. (6). Our full L2M-GAN can be denoted as BL+PL+OL (L1), i.e., StarGAN v2 with the perceptual loss and orthogonality loss having two L1-norm terms of Eq. (6). The results in Table 3 show that: (1) The perceptual loss is important for facial attribute edit-

Method	FID (+)	FID (-)	FID (avg)	Acc (att)
BL (Baseline)	19.8	30.8	25.3	77.2%
BL+PL	19.8	25.1	22.5	78.7%
BL+PL+OL (Single)	18.9	24.4	21.7	88.3%
BL+PL+OL (Inner)	18.3	23.8	21.1	92.0%
BL+OL (L1)	18.3	23.9	21.1	91.8%
BL+PL+OL (L1)	**17.9**	**23.3**	**20.6**	**93.1%**

Table 3. Ablation study results for our full L2M-GAN model on the specific attribute: **Smiling**. BL (Baseline) – our baseline based on StarGAN v2 [6]; PL – perceptual loss; OL (Single) – orthogonality loss having only the first term of Eq. (6) but defined by inner product; OL (Inner) – orthogonality loss having two terms of Eq. (6) but defined by inner product; OL (L1) – orthogonality loss having two L1-norm terms of Eq. (6).

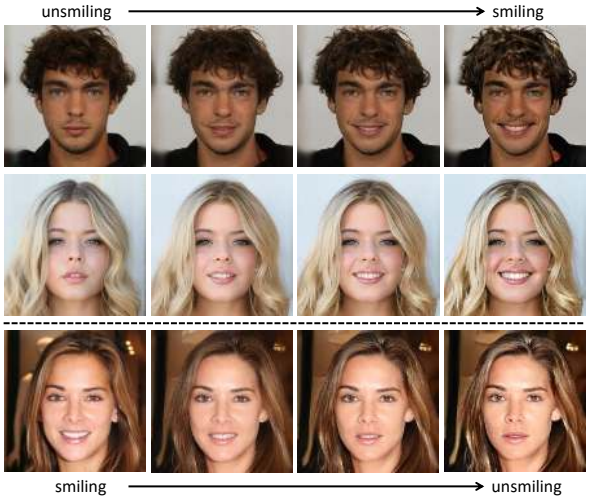

Figure 4. Examples of attribute strength manipulation by our L2M-GAN. The first column shows the real input/source images, and the next three columns show the same subject with a gradually changed attribute (smiling or unsmiling).

Figure 5. Examples of face synthesis by our L2M-GAN with reference images as inputs. For each reference image, the attribute **Smiling** (with binary attribute values) is added to the source/input image and thus transferred to the target/output image.

ing. (2) The orthogonality loss brings in a big boost in accuracy (see BL+PL+OL (Single) vs. BL+PL), since better latent space disentanglement can be obtained. (3) Enforcing orthogonality both before and after the style transformation further boosts the latent space disentanglement (see BL+PL+OL (Inner) vs. BL+PL+OL (Single)). (4) A more strict orthogonality loss defined with L1-norm leads to improvements over the inner product based one. (5) The perceptual loss is indeed complementary to the orthogonality loss (see BL+OL (L1) vs. BL+PL+OL (L1)).

4.5. Results on Other Attribute Manipulation Tasks

Attribute Strength Manipulation. We synthesize a series of images with different attribute strengths w.r.t. the smiling attribute by varying λ_s according to Eq. (8). The visual results are shown in Figure 4. We can see that the attribute strengths of the images along each row are changed gradually (but without changing any irrelevant information) as λ_s increases, indicating that the learned latent space is well disentangled and the domain-related style code only contains the information related to the domain.

Manipulation with Reference Images. We also synthesize images using reference images as supervision. As shown in Figure 5, our L2M-GAN can transfer the smiling attribute precisely from the reference images but without any smiling-irrelevant information, providing further evidence that the latent space is well disentangled by our model.

Multiple Attribute Manipulation. With StarGAN v2 as the backbone, our L2M-GAN can also be applied to multiple attribute manipulation. Figure 6 shows the results of manipulating two attributes by our L2M-GAN. The set of two attributes {smiling, gender} are considered in our L2M-GAN in the mean time. We can observe that our L2M-GAN can precisely manipulate the two attributes in all cases due to the well-learned semantics of the latent space.

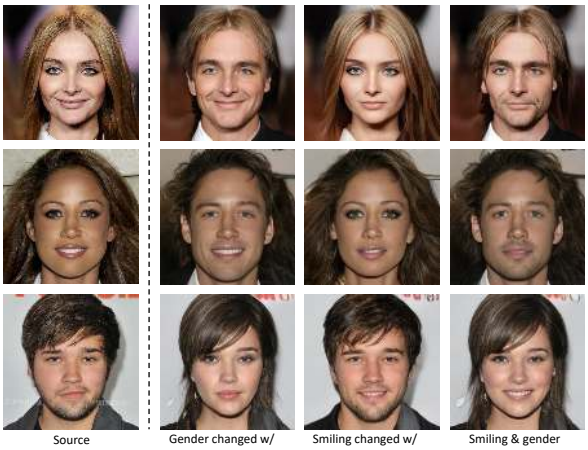

Figure 6. Examples of multiple attribute manipulation by our L2M-GAN. The first column shows the real input/source images, and the next three columns show the results of changing only a single attribute or two attributes simultaneously.

Figure 7. Examples of anime face manipulation by our L2M-GAN. The first row shows the input images, and the second row shows the results of manipulating the smiling attribute of anime faces.

Generalization to Anime Face Manipulation. Our L2M-GAN can be directly applied to anime face manipulation, without re-training the whole model. The results in Figure 7 show that our L2M-GAN has a good generalization ability for cross-dataset facial attribute editing.

5. Conclusion

We have proposed a novel facial attribute editing model based on latent space factorization in GAN. The proposed L2M-GAN is the first end-to-end GAN model for facial attribute editing based on latent space factorization and is effective for both local and global attribute editing. This is due to a novel style transformer that factorizes the latent code into attribute-relevant and -irrelevant parts by enforcing orthogonality both before and after the transformation. Extensive experiments show that our L2M-GAN achieves significant improvements over the state-of-the-arts.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (61976220 and 61832017), and Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098). Zhiwu Lu is the corresponding author.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017. 2

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 2, 5

[3] Ying-Cong Chen, Huaijia Lin, Michelle Shu, Ruiyu Li, Xin Tao, Xiaoyong Shen, Yangang Ye, and Jiaya Jia. Facelet-bank for fast portrait manipulation. In *CVPR*, pages 3541–3549, 2018. 1

[4] Ying-Cong Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I Pao, Jiaya Jia, et al. Semantic component decomposition for face attribute manipulation. In *CVPR*, pages 9859–9867, 2019. 1

[5] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 2, 3, 5, 6, 7

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8185–8194, 2020. 2, 3, 4, 5, 7

[7] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, pages 5771–5780, 2020. 3

[8] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Trans. Neural Networks and Learning Systems*, 30(7):1967–1974, 2018. 2

[9] Garoe Dorta, Sara Vicente, Neill DF Campbell, and Ivor JA Simpson. The gan that warped: Semantic attribute editing with unpaired data. In *CVPR*, pages 5356–5365, 2020. 3

[10] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018. 2

[11] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, pages 5744–5753, 2019. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[16] Zhenliang He, Meina Kan, Jichao Zhang, and Shiguang Shan. PA-GAN: Progressive attention generative adversarial network for facial attribute editing. *arXiv preprint arXiv:2007.05892*, 2020. 2, 3, 5, 6, 7

[17] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial attribute editing by only changing what you want. *IEEE Trans. Image Processing*, 28(11):5464–5478, 2019. 2, 3

[18] 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. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[19] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017. 4

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 2

[21] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2019. 2

[22] Kui Jiang, Zhongyuan Wang, Peng Yi, Guangcheng Wang, Tao Lu, and Junjun Jiang. Edge-enhanced GAN for remote sensing image superresolution. *IEEE Trans. Geoscience and Remote Sensing*, 57(8):5799–5812, 2019. 2

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 2, 5

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 2, 4, 5, 6

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 5

[26] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 2

[27] Jeong Kwak, David K. Han, and Hanseok Ko. CAFE-GAN: Arbitrary face attribute editing with complementary attention feature. In *ECCV*, 2020. 2, 3

[28] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017. 3

[29] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, pages 1558–1566, 2016. 1

[30] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017. 2

[31] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards diverse and interactive facial image manipulation. In *CVPR*, pages 5549–5558, 2020. 3

[32] Xiao Li, Chenghua Lin, Chaozheng Wang, and Frank Guerin. Latent space factorisation and manipulation via matrix subspace projection. In *ICML*, 2020. 3

[33] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*, pages 3673–3682, 2019. 2, 3

[34] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, pages 10550–10559, 2019. 4

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 5

[36] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *Advances in Neural Information Processing Systems*, pages 9628–9637, 2018. 2

[37] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 2

[38] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In Jennifer G. Dy and Andreas Krause, editors, *ICML*, pages 3478–3487, 2018. 4

[39] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[40] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 2

[41] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks. In *MICCAI*, pages 417–425, 2017. 2

[42] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional GANs for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 2

[43] Henning Petzka, Asja Fischer, and Denis Lukovnikov. On the regularization of wasserstein gans. In *ICLR*, 2018. 2

[44] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 5

[45] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *CVPR*, pages 4030–4038, 2017. 1, 2

[46] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *CVPR*, pages 9240–9249, 2020. 2, 3, 5, 6, 7

[47] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *CVPR*, pages 7064–7073, 2017. 1

[48] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. ELEGANT: Exchanging latent encodings with GAN for transferring multiple face attributes. In *ECCV*, pages 172–187, 2018. 1, 2, 3, 5, 6, 7

[49] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in the deep generative representations for scene synthesis. *arXiv preprint arXiv:1911.09267*, 2019. 2

[50] Yasin Yazici, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in GAN training. In *ICLR*, 2019. 2, 5

[51] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017. 2

[52] Weidong Yin, Ziwei Liu, and Chen Change Loy. Instance-level facial attributes transfer with geometry-aware flow. In *AAAI*, pages 9111–9118, 2019. 2

[53] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018. 2

[54] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *ECCV*, pages 417–432, 2018. 1, 2

[55] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, pages 7354–7363, 2019. 2

[56] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. RankSRGAN: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, pages 3096–3105, 2019. 2

[57] Shuchang Zhou, Taihong Xiao12, Yi Yang, Dieqiao Feng, Qinyao He, and Weiran He. GeneGAN: Learning object transfiguration and attribute subspace from unpaired data. In *BMVC*, 2017. 3

[58] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *ECCV*, 2020. 2, 3, 5, 6

[59] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, pages 597–613, 2016. 2

[60] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2, 4, 5, 6, 7