Homomorphic Latent Space Interpolation for Unpaired Image-to-image Translation

Ying-Cong Chen[1] Xiaogang Xu[1] Zhuotao Tian[1] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]Tencent Youtu Lab
{ycchen,xgxu,zttian,leojia}@cse.cuhk.edu.hk

Abstract

Generative adversarial networks have achieved great success in unpaired image-to-image translation. Cycle consistency allows modeling the relationship between two distinct domains without paired data. In this paper, we propose an alternative framework, as an extension of latent space interpolation, to consider the intermediate region between two domains during translation. It is based on the fact that in a flat and smooth latent space, there exist many paths that connect two sample points. Properly selecting paths makes it possible to change only certain image attributes, which is useful for generating intermediate images between the two domains. We also show that this framework can be applied to multi-domain and multi-modal translation. Extensive experiments manifest its generality and applicability to various tasks.

1. Introduction

Unpaired image-to-image translation and latent space interpolation were developed separately and serve different applications. Unpaired image-to-image translation [28, 9, 14, 4] aims to map images from one domain to another, e.g, translating a collection of neutral faces to smiling ones. Since no pair information is available, the connection of different domains is usually built upon the cycle-consistency constraint [28], which largely promotes the capacity of generative models and leads to many impressive results.

When the purpose is to generate a sequence of images between the input two domains, intermediate states should be considered, which is however beyond the capability of the cycle-consistency constraint. We show an example in Fig. 1 – directly using StarGAN [4] does not generate a natural sequence (or expression flow) to gradually close mouth. There exists a quick change between (c) and (d).

On the other hand, to generate smooth flow, latent space interpolation [11, 21, 22] focuses on intermediate states based on an assumption that deep neural networks can model natural images as flat and smooth distributions. Specifically, if x and y are sampled from two respective domains $\mathcal{X}$ and $\mathcal{Y}$, moving from x toward y in the latent space continuously produces realistic images from domain $\mathcal{X}$ to $\mathcal{Y}$. Albeit this nice property, this method cannot directly serve image-to-image translation because it does not distinguish among different attribute factors, and thus makes complicated expression transition tangled with identity or background changes. Also, the interpolation path ends at y instead of a translated version of x.

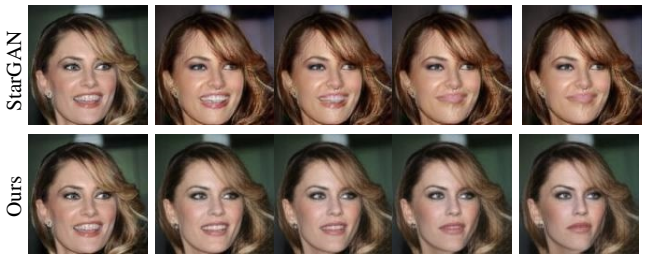


Figure 1. Rendering intermediate states between (a) "open-mouth" domain and (e) "close-mouth" domain. The first-row results are generated by StarGAN [4]. Rendering intermediate states is achieved by altering the input domain label continuously. (c) and (d) show that abrupt change of expression exists. Our results in the second row model intermediate regions and show smooth translation effect.

In this paper, we address latent space interpolation in unpaired image-to-image translation. This solution inherently allows modeling intermediate regions between different domains, with additional important and appealing capacity of multi-domain and multi-modal translation. Since in a flat and smooth latent space, many paths exist to connect two samples, interpolating along different paths leads to diverse intermediate results [24]. Our idea is to *choose the path that only corresponds to a certain attribute component* to make transition natural to human perception. Here the term *attribute* defines image domains. For example, *smiling* attribute divides facial images to *smiling* and *non-smiling* domains. Fig. 2 provides an example where translating be-

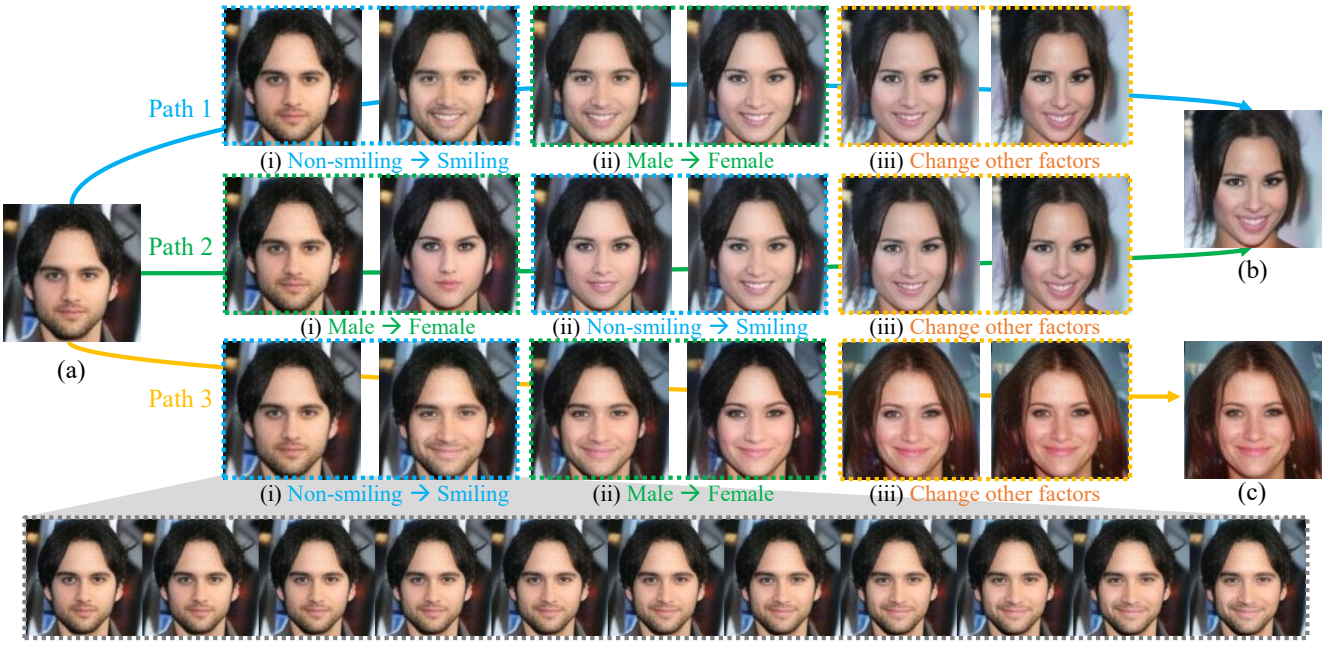


Figure 2. Illustration of latent space interpolation along different paths. Paths 1 and 2 connect (a) "non-smiling male" and (b) " smiling female". They change facial attributes in different orders – i.e., path 1 changes the smiling expression first while path 2 interpolates gender. They naturally serve the **multi-domain image-to-image translation** task where path 1(i) and 2(i) form translation between smiling and non-smiling domains, and male and female domains respectively. Path 3(i) synthesizes a smile different from path 1(i). Thus, using different target-domain samples, our method can produce output required for each domain, termed as **multi-modal image-to-image translation**. Image sequence of the last row illustrates the continuous change of path 3(i).

tween *Male* and *Female* (or *Smiling* and *Non-smiling*) can be achieved by interpolating along path 1(i) (or path 2(i)) respectively. Besides multi-domain and continuous translation capacity, as shown in paths 1(i) and 3(i), this model can also deal with multi-modal translation.

With this principle, the key to our method is a controllable *interpolator*, whose output is controlled by a vector $\boldsymbol{v}$. Each element of $\boldsymbol{v}$ corresponds to a *mixing* indicator for each attribute. We take path 3(i) of Fig. 2 for example. A proper $\boldsymbol{v}$ only deals with the smiling attribute between (a) and (c), while keeping other attributes untouched.

Although promising, this strategy requires conquering a few difficulties. First, interpolation is only allowed in a smooth and flat space. VAE [13] imposes Gaussian prior on the latent feature space so that interpolation is allowed. However, it could generate blurry results, as a Gaussian prior may be insufficient to model complicated natural images. Our solution is to directly minimize the Wasserstein distance between the interpolated and real samples of the latent space. This makes interpolated sample distribution as close as possible to the real ones. We also introduce a knowledge guidance loss that leverages a well-trained network to regularize the latent space, which further improves interpolation quality. Finally, a homomorphic loss is introduced to train the controllable interpolator. Our total contribution is manifold.

- We propose an interpolation-based framework for unpaired image-to-image translation, which is feasible for multi-domain, multi-modal and continuous translation tasks.
- We propose a few important strategies to train our model, leading to an interpolatable latent space and a controllable interpolator.
- Extensive experiments show that our model can generate high-quality results and is flexible to serve various applications.

2. Related Work

Latent Space Interpolation Latent space interpolation is widely used to visualize the manifold structure in a flat feature space [10, 1, 21, 22, 2]. Intuitively, semantical interpolation in the latent space indicates that the space captures certain high-level information, which is beneficial for both recognition [1] and generation tasks [10]. However, a vanilla interpolation between two images may not be that useful for creation, since all attributes would change together along the interpolation path, and users lose control of individual ones. One remedy is to interpolate along *attribute vectors* rather than between samples [23, 13, 10, 3]. For certain target attributes, average of positive and negative samples are computed, and the attribute vector is defined as

the difference of them. This cancels out the influence of non-target attributes and allows users to edit only the target one. Nevertheless, it ignores the fact that many attributes are intrinsically multi-modal. As illustrated in Fig. 2(b) and (c), smiling can be quite different. Interpolation with a universal *smiling* attribute vector can only generate the average smile. In contrast, our model can produce multi-modal results with different examples.

Unpaired Image-to-image Translation Unpaired image-to-image translation [28, 4, 9, 15] aims to map images of one domain to another. CycleGAN [28], DiscoGAN [9] and DualGAN [26] are three pioneering methods, which introduce the cycle-consistency constraint to build the connection. There are however a few remaining issues. The domain scalability issue refers to the incapability of handling more than two domains, which is addressed by StarGAN [4] and ModularGAN [27]. The multi-modality issue refers to incapability to produce multiple results, which is addressed by MUNIT [7] and DRIT [14]. The discreteness issue refers to the inability to continuously control the transformation strength between two domains, which is addressed by GANimation [17]. We note GANimation [17] requires continuous label annotation, which is costly and is limited in the field of facial expression.

Instead of relying on the cycle consistency constraint, our model seeks another way to tackle the unpaired image-to-image translation problem. Our model can be deemed as a general alternative that tackles the domain scalability, multi-modality and discreteness issues simultaneously.

3. Proposed Method

Without the loss of generality, we take the face attribute translation task as an example to introduce our method. Other tasks are also supported and are presented in the supplementary material. We define the dataset as $\mathcal{D} = \{(I_1, \boldsymbol{y}_1), (I_2, \boldsymbol{y}_2) \cdots (I_N, \boldsymbol{y}_N)\}$ of N samples, where $I_i \in \mathbb{R}^{H\times W\times 3}$ and $\boldsymbol{y}_i = [\boldsymbol{y}_i^1, \boldsymbol{y}_i^2, \cdots, \boldsymbol{y}_i^d]$ are the i-th face image and its corresponding attributes respectively. The subscript and superscript index samples and attributes respectively.

We further introduce the concept of *grouped attribute*. For example, we can group *angry, happy, sad, contemptuous, disguised, fear* and *surprise* – these attributes are provided in RaFD [12] dataset as binary attribute labels – to form the group *expression* attribute. Thus, the plain attributes $\boldsymbol{y}_i$ can be rearranged to $\boldsymbol{z}_i = \{\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, \cdots, \boldsymbol{z}_i^c\}$, where $\boldsymbol{z}_i^k \in \mathbb{R}^{c_i\times 1}$ denotes the k-th grouped attribute of the i-th sample. This makes it more intuitive to use our model. An instance is that paths 1(i), 2(ii) and 3(i) of Fig. 2, with the *expression* attribute, consider the 8 expressions rather than only *smiling*.

In the model level, we have an encoder $\boldsymbol{E}$, an interpolator $\boldsymbol{\mathcal{I}}$ and a decoder $\boldsymbol{D}$. The encoder $\boldsymbol{E}$ maps images I_i and I_j to feature $F_i = \boldsymbol{E}(I_i)$ and $F_j = \boldsymbol{E}(I_j)$, so that the interpolated feature $\boldsymbol{\mathcal{I}}(F_i, F_j)$ is indistinguishable from real samples. The interpolator $\boldsymbol{\mathcal{I}}$ produces interpolated results of two samples. The decoder $\boldsymbol{D}$ maps the latent features back to the image space. In the following, we elaborate on the design of each part.

3.1. Learning Encoder and Decoder

It is well known that natural images usually lie on a non-convex manifold, making interpolation usually difficult. We train an encoder to unfold the image manifold to a flattened latent space, such that the interpolated samples are in real-image space. This is achieved by applying GAN to make interpolated feature similar to that of real samples.

Specifically, we leverage WGAN-GP [5] to train our model. A critic $\boldsymbol{\mathcal{D}}$ is trained to maximize the Wasserstein distance between real samples and interpolated ones, and the encoder $\boldsymbol{E}$ and interpolator $\boldsymbol{\mathcal{I}}$ are trained to minimize the distance between them. It is formulated as

$$\min_{\boldsymbol{\mathcal{D}}} \mathcal{L}_{GAN_{\boldsymbol{\mathcal{D}}}} = \mathbb{E}_{\mathbb{P}_{\boldsymbol{\mathcal{I}}}}[\boldsymbol{\mathcal{D}}(\hat{F})] - \mathbb{E}_{\mathbb{P}_r}[\boldsymbol{\mathcal{D}}(F)] + \lambda_{gp}\mathcal{L}_{gp}, \tag{1}$$

$$\min_{\boldsymbol{E},\boldsymbol{\mathcal{I}}} \mathcal{L}_{GAN_{\boldsymbol{E},\boldsymbol{\mathcal{I}}}} = \mathbb{E}_{\mathbb{P}_r}[\boldsymbol{\mathcal{D}}(F)] - \mathbb{E}_{\mathbb{P}_{\boldsymbol{\mathcal{I}}}}[\boldsymbol{\mathcal{D}}(\hat{F})], \tag{2}$$

where $F = \boldsymbol{E}(I)$ is the feature extracted by the encoder, $\hat{F}$ is the interpolated feature generated by $\hat{F} = \boldsymbol{\mathcal{I}}(F_i, F_j)$, $\mathbb{P}_r$ and $\mathbb{P}_{\boldsymbol{\mathcal{I}}}$ are the distributions of real and interpolated samples respectively, and $\mathcal{L}_{gp}$ is the gradient penalty term defined in [5]. Here the interpolator $\boldsymbol{\mathcal{I}}$ works with encoder $\boldsymbol{E}$ cooperatively to generate reasonable images. More details of $\boldsymbol{\mathcal{I}}$ are provided in later sections.

Note that simply using Eqs. (1) and (2) may cause the encoder to map all images to a small feature space where interpolation becomes easy. To an extreme, if the encoder maps all images to a single point, the interpolated and real samples yield Wasserstein distance 0. But this trivial solution carries no information about the images. To avoid it, we additionally incorporate a decoder $\boldsymbol{D}$ to invert features back to images. The decoder is trained with perceptual loss [8] as Eq. (3). The reconstruction term for the encoder is defined as Eq. (4).

$$\min_{\boldsymbol{D}} \mathcal{L}_{\boldsymbol{D}} = \mathbb{E}(||\Phi_3(\boldsymbol{D}(F)) - \Phi_3(I)||^2), \tag{3}$$

$$\min_{\boldsymbol{E}} \mathcal{L}_{recon} = \mathbb{E}(||\Phi_3(\boldsymbol{D}(\boldsymbol{E}(I))) - \Phi_3(I)||^2), \tag{4}$$

where $\Phi_3(I)$ is the RELU3_1 feature of the VGG network.

Semantic Knowledge Guidance Previous work has observed that a pretrained VGG network [20] can be utilized for latent space interpolation [3, 23, 1]. We leverage this property to guide the training of our encoder. Inspired by [18, 6], we treat a pretrained VGG network as a *teacher*,

and use its intermediate layer to guide the training of our encoder, formulated as

$$\min_{\boldsymbol{E},\boldsymbol{P}} \mathcal{L}_{KG} = \mathbb{E}_{\mathbb{P}_r} ||\boldsymbol{P}[\boldsymbol{E}(I)] - \Phi_5(I)||^2, \quad (5)$$

where $\boldsymbol{P}$ is a 1×1 convolutional layer that adapts the feature space defined by $\boldsymbol{E}(I)$ to the space of $\Phi_5(I)$. Φ_5 denotes the ReLU5_1 layer of the VGG network [20]. As the VGG network is trained with millions of images, $\Phi_5(I)$ contains rich semantic information and provides extra guidance for the encoder. Generally, this term works as regularization and helps the encoder converge to a good result.

By combining Eqs. (2), (4) and (5), the final objective function of the encoder $\boldsymbol{E}$ is

$$\mathcal{L}_{\boldsymbol{E}} = \lambda_{GAN_{\boldsymbol{E}}}\mathcal{L}_{GAN_{\boldsymbol{E},\boldsymbol{\mathcal{I}}}} + \lambda_{recon}\mathcal{L}_{recon} + \lambda_{KG}\mathcal{L}_{KG}, \quad (6)$$

where $\lambda_{GAN_{\boldsymbol{E}}}$, λ_{recon} and λ_{KG} are scalars to balance terms. We set them as 1s in our experiments.

3.2. Learning Interpolator

With a well-learned encoder that maps images to a flat space, interpolation can be done linearly as

$$\boldsymbol{\mathcal{I}}(F_i, F_j) = F_i + \alpha(F_j - F_i), \quad (7)$$

where F_i and F_j are two real samples, and $\alpha \in [0, 1]$ is a parameter that controls the level of mixing of two samples. The second term $\alpha(F_j - F_i)$ can also be viewed as a *shifting vector* that points from F_i towards F_j.

Note that Eq. (7) only defines one possible path that connects samples i and j. Other interpolation methods like Slerp [19] can also connect them and produces different intermediate results. Nevertheless, all these handcrafted methods do not allow adjusting how attributes are mixed. So they are not usable for our task. To accommodate image-to-image translation, we extend $\boldsymbol{\mathcal{I}}(F_i, F_j)$ to a more flexible $\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)$, where $\boldsymbol{v} \in [0, 1]^{c\times1}$ is a *control vector*. Each dimension of $\boldsymbol{v}$ sets the interpolation strength of each grouped attribute between two samples. More specifically, the linear interpolation defined in Eq. (7) is extended to a piecewise one of

$$\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j) = F_i + \sum_{k=1}^{c} \boldsymbol{v}^k \boldsymbol{\mathcal{T}}^k(F_j - F_i), \quad (8)$$

where $\boldsymbol{v}^k$ is the k^{th} dimension of $\boldsymbol{v}$, and $\boldsymbol{\mathcal{T}}^k(\cdot)$ is a learnable mapping function represented by CNN.

Minimizing Homomorphic Gap It is expected that $\boldsymbol{\mathcal{T}}^k(F_j - F_i)$ and $\boldsymbol{v}^k$ correspond to the interpolation direction and strength of the k^{th} grouped attribute $\boldsymbol{z}^k$ respectively. As $\boldsymbol{v}^k$ varies from 0 to 1, the k^{th} grouped attribute changes from sample i to j accordingly. If all possible values of $\boldsymbol{z}$ form an *attribute space*, interpolation in the latent feature space should correspond to interpolation in the attribute space. Let $\mathcal{A}(\cdot)$ be a function that maps latent feature to an attribute vector, i.e., $\mathcal{A}(F_i) = \boldsymbol{z}_i$, we define the relation between the latent space and the attribute space as

$$\mathcal{A}(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)) = \boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\mathcal{A}(F_i), \mathcal{A}(F_j)), \forall \boldsymbol{v} \in [0, 1]^{c\times1} \quad (9)$$

where $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ can be viewed as an interpolation function defined in the attribute space. Further, $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is defined as $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = [\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)^1 \cdots, \boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)^c]$, where $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)^k = \boldsymbol{z}_i^k + \boldsymbol{v}^k(\boldsymbol{z}_j^k - \boldsymbol{z}_i^k)$. So the left hand side of Eq. (9) denotes the attribute values of interpolated samples $\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)$, and the right hand side contains the corresponding attribute values of the two samples. As both sides are conditioned on the same control vector $\boldsymbol{v}$, they are expected to be equal. In this regard, Eq. (9) describes an ideal case that the interpolation operations $\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}$ and $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}$ share the same structure in the latent feature and attribute space. This property is analogous to *homomorphism* in algebra. In practice, there inevitably exists a gap between two sides in Eq. (9), which we call the *homomorphic gap*.

With Eq. (9) introduced, our objective turns to minimizing the homomorphic gap. Recall that $\mathcal{A}(\cdot)$ maps latent feature to attribute values, which is not defined for interpolated features. We choose to train a network $\mathcal{A}'(\cdot)$ to approximate $\mathcal{A}(\cdot)$ and replace $\mathcal{A}(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j))$ with $\mathcal{A}'(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j))$ in Eq. (9). Then we reduce the homomorphic gap by minimizing the cross-entropy of $\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ and $\mathcal{A}'(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j))$, as shown in Eq. (10). We call it the *Homomorphic loss*:

$$\min_{\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}} \mathcal{L}_{\boldsymbol{\mathcal{I}}_{hom}} = \mathbb{E}[-\boldsymbol{\mathcal{I}}'_{\boldsymbol{v}}(\boldsymbol{z}_i, \boldsymbol{z}_j) \log(\mathcal{A}'(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)))]. \quad (10)$$

Also, $\boldsymbol{v}$ is defined everywhere in the c-dimensional unit hypercube. During training, we assign uniformly random values to $\boldsymbol{v}$ to cover the whole feasible set.

Rigorous Training According to Eq. (8), optimizing Eq. (10) needs to optimize $\boldsymbol{\mathcal{T}}^k(\cdot)$, where $k = 1, \cdots, c$. In experiments, when complicated attributes exist, the corresponding $\boldsymbol{\mathcal{T}}^k(\cdot)$ tends to be lazy – that is, it may update F_i slightly to fool the attribute classification network $\mathcal{A}'(\cdot)$. To alleviate this problem, we turn $\mathcal{A}'(\cdot)$ to a rigorous classifier: instead of mapping F_i to $\boldsymbol{z}_i$, $\mathcal{A}'(\cdot)$ is trained to map the interpolated feature $F_i + \sum_{k=1}^{c} \boldsymbol{v}^k \boldsymbol{\mathcal{T}}^k(F_j - F_i)$ to attribute $\boldsymbol{z}_i$, expressed as

$$\min_{\mathcal{A}'} \mathcal{L}_{\mathcal{A}'} = \mathbb{E}[-\boldsymbol{z}_i \log(\mathcal{A}'(\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)))]. \quad (11)$$

From Eqs. (10) and (11), we note that $\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(\cdot)$ and $\mathcal{A}'(\cdot)$ are mutually dependent. Therefore, they are iteratively updated during training. In this way, $\mathcal{A}'(\cdot)$ keeps checking unchanged parts, making it harder for $\boldsymbol{\mathcal{T}}^k(\cdot)$ to fool.

Handling Residual Components When $\boldsymbol{v} = \boldsymbol{1}$ where $\boldsymbol{1} = [1, 1, \cdots, 1] \in \mathbb{R}^{c\times1}$, $\boldsymbol{\mathcal{I}}_{\boldsymbol{v}}(F_i, F_j)$ is expected to reach

sample j. However, this is not guaranteed with solely the homomorphic loss, because the provided attributes may not explain everything. Therefore, we extend Eq. (8) to

$$\mathcal{I}_{\boldsymbol{v}}(F_i, F_j) = F_i + \sum_{k=1}^{c+1} \boldsymbol{v}^k \mathcal{T}^k(F_j - F_i), \quad (12)$$

where the additional mapping function $\mathcal{T}^{c+1}(F_j - F_i)$ models the residual components that are not explained by the given attributes. Accordingly, we extend the c-dimension control vector $\boldsymbol{v}$ to $c+1$ dimensions, where the last dimension is the edit strength of the residual mapping function. Now we can safely impose the terminal of the interpolation curve as F_j, which is formulated as

$$\mathcal{L}_{\mathcal{I}_t} = ||\mathcal{I}_{\boldsymbol{v}}(F_i, F_j) - F_j||^2, \text{where } \boldsymbol{v} = \mathbf{1}. \quad (13)$$

To summarize this part, the overall loss function of $\mathcal{I}_{\boldsymbol{v}}$ is

$$\mathcal{L}_{\mathcal{I}} = \lambda_{GAN_{\mathcal{I}}} \mathcal{L}_{GAN_{\boldsymbol{E},\mathcal{I}}} + \lambda_{\mathcal{I}_{hom}} \mathcal{L}_{\mathcal{I}_{hom}} + \lambda_{\mathcal{I}_t} \mathcal{L}_{\mathcal{I}_t}, \quad (14)$$

where $\mathcal{L}_{GAN_{\boldsymbol{E},\mathcal{I}}}$, $\mathcal{L}_{\mathcal{I}_{hom}}$ and $\mathcal{L}_{\mathcal{I}_t}$ are defined in Eqs. (2), (10) and (13) respectively. $\lambda_{GAN_{\mathcal{I}}}$, $\lambda_{\mathcal{I}_{hom}}$ and $\lambda_{\mathcal{I}_{hom}}$ are set to 1 in our experiments.

The training procedure is outlined in Algorithm 1. More training details are contained in the supplementary material.

Algorithm 1 Training Our Model

Input: I_i and $\boldsymbol{z}_i$, where $i = 1, 2, \cdots, N$
Output: encoder $\boldsymbol{E}$, interpolator $\mathcal{I}_{\boldsymbol{v}}$ and decoder $\boldsymbol{D}$

```
while not converged do
    sample v from c-dimensional uniform distribution;
    t ← 0;
    while t < 5 do
        update the critic D based on Eq. (1);
        update the decoder D based on Eq. (3);
        update the P in Eq. (5);
        update the attribute classifier A' based on Eq. (11);
    end while
    update the encoder E based on Eq. (6);
    update the interpolator I based on Eq. (14).
end while
```

3.3. Applications

We describe how our model can be applied to multi-domain, multi-modal and continuous translation as follows.

Multi-domain Translation For each target domain t, we preselect a sample I_t. Given a query sample I_q, domain translation is conducted as

$$I_{out} = \boldsymbol{D}(\mathcal{I}_{\boldsymbol{v}_t}(\boldsymbol{E}(I_q), \boldsymbol{E}(I_t))), \quad (15)$$

where $\boldsymbol{v}_t$ is the vector corresponding to the target domain.

Dim	Attribute	Labels
1	Age	Young
2	Expression	Mouth_Slightly_Open, Smiling
3	Hair Color	Black_Hair, Blond_Hair Brown_Hair, Gray_Hair
4	Hair Style	Receding_Hairline, Bangs
5	Gender Trait	Male, No_Beard, Mustache, Goatee, Sideburns

Table 1. Grouped Attributes of CelebA [16]. The 1st-3rd columns: dimension index in the control vector $\boldsymbol{v}$, name of grouped attributes, corresponding attribute labels.

Dim	Attribute	Labels
1	Expression	happy, angry, contemptuous, sad, disgusted, neutral, fearful, surprised
2	Gaze	look left, look front, look right
3	Others	is_Caucasian, is_male, is_kid

Table 2. Grouped Attributes of RaFD [12].

Multi-Modal Translation By using different exemplars in Eq. (15), we can generate results like MUNIT [7].

Continuous Translation By changing $\boldsymbol{v}_t$ in Eq. (15) smoothly, our model allows changing attributes continuously. This controls the edit strength or generates animation along the translation process.

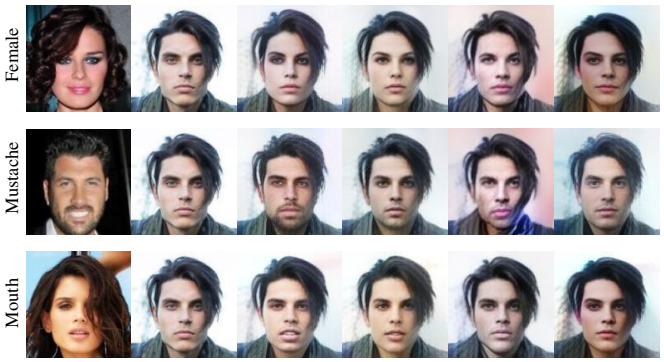
4. Experiments

Datasets Our experiments are conducted on CelebA [16] and RaFD [12]. CelebA contains 200K celebrity images, each with 40 attribute labels. We define grouped attributes based on these labels as shown in Table 1. Separation of training and testing sets follows that of [16]. RaFD [12] is a smaller dataset that contains 67 identities, each displays 8 emotional expressions, 3 eye locations, and 3 other attributes about the identities. Similarly, we group these labels into 3 higher-level attributes as shown in Table 2. In our experiments, we use 65 identities for training and the other two for testing. All images are center cropped, resized to 128×128.

4.1. Analysis

Pivotal Parts in Training It is noted that the knowledge guidance loss $\mathcal{L}_{KG}$ and the homomorphic loss $\mathcal{L}_{\mathcal{I}_{hom}}$ with rigorous training play a key role in our model. Without either of them, the training may converge poorly, leading to unsatisfactory results. To illustrate this, we disable each part and compare results with our final one in Fig. 3. The homomorphic loss Eq. (10) allows controlling the interpolated attribute with control vector $\boldsymbol{v}$. As shown in Fig. 3(f), without this term, the generated image cannot transfer the target attribute from the reference image.

When we disable the rigorous training, the interpolator may produce small change to just deceive the discriminator, leading to very mild update of results. This is shown in Fig. 3(d). Compared with our final model in Fig. 3(c), the effect



(a) Ref (b) Original (c) Final (d) no RT (e) no KG (f) no Hom

Figure 3. Effectiveness of rigorous training (RT), knowledge guidance (KG) and homomorphic loss (Hom). Each row edits one attribute. (a) and (b) are the input reference and original images respectively. (c) is our final result. (d-f) are the results without using one component each.

(a) Ref (b) Original (c) Gender (d) Smile (e) Hair Color

Figure 4. Illustration of the role of control vector and exemplar. (a) and (b) are the reference and original images respectively. (c)-(e) are the results conditioned by different $\boldsymbol{v}$. Rows 1 and 2 are of different reference images, and thus the results vary accordingly.

is not desirable. The knowledge guidance loss utilizes a well-trained network as a teacher to guide training of the encoder. As the teacher network is trained on many images, it effectively extracts semantic features and seldom suffer from overfitting. As shown in Fig. 3(e), without this term, the encoder does not learn a smooth and flat latent space. This makes the generated image look unrealistic.

Pivotal Parts in Testing The control vector $\boldsymbol{v}$ and the reference exemplars are also important to apply our model to image-to-image translation tasks. The control vector determines which attribute to alter, while the exemplars determine how attribute translation is instantiated. By jointly using both of them, we flexibly control the interpolation results. This is illustrated in Fig. 4. Each row shows how the result changes with the same exemplar and yet a different control vector. Each column shows how it changes with the same control vector and yet a different exemplar. As shown in Fig. 4(c)-(e), by setting $\boldsymbol{v}$ to one-hot vectors presenting the gender, expression, and hair color respectively, we successfully and effectively vary corresponding attributes. The

(a) (b) (c) (d) (e)

Figure 5. Multi-modal image-to-image translation on bang attribute. (a) is the input image. (b-d) present four different output (the 1st-3rd rows) and the corresponding exemplars (the 4th row).

exemplars also affect the final results. For example, results in the 1st and 2nd rows of Fig. 4 have quite different gender, expression, and hair color change.

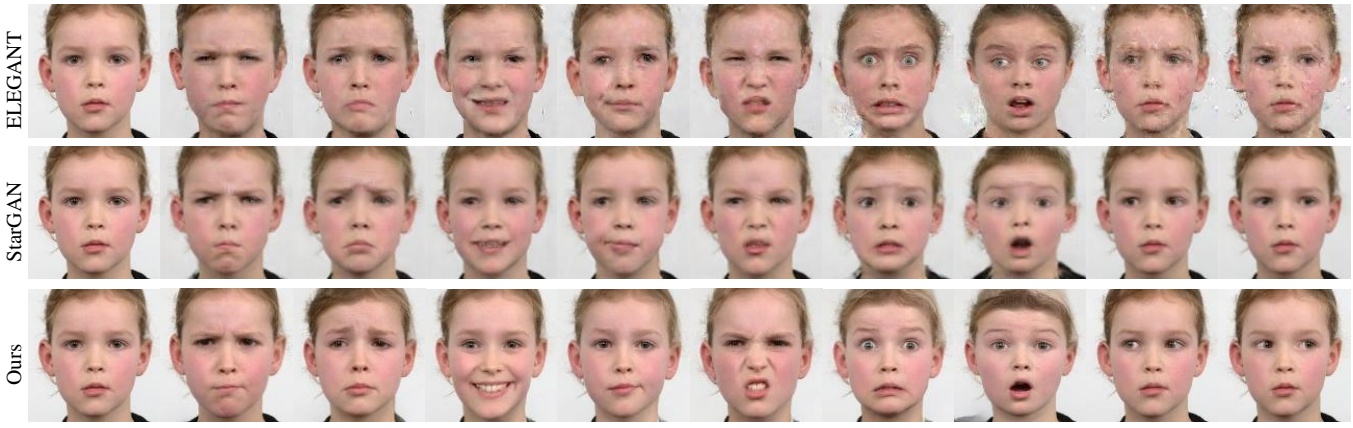
4.2. Comparison with Other Methods

One of the largest advantages of our model is the ability to handle multi-modal, multi-domain, and continuous image-to-image translation. In this section, we provide both qualitative and quantitative comparison with other methods.

4.2.1 Qualitative Evaluation

Multi-Modal Translation Using different exemplars, our model can produce multiple outputs for image-to-image translation. Fig. 5 compares our approach with two multi-modal translation methods, i.e., MUNIT [7] and ELEGANT [25]. For ELEGANT [25], the assumption of attribute disentangled to different latent codes, again, could be hard to achieve. As shown in Fig. 5(d), the image does not change much. Compared with our method, MUNIT [7] does not leverage information of multiple domains. When skin, hair color and background are wrongly updated, as shown in Fig. 5, the result quality decreases.

Multi-Domain Translation Our model deals with multi-domain image-to-image translation with Eq. (15). Figs. 6 and 7 compare our results with two related methods, i.e., StarGAN [4] and ELEGANT [25]. StarGAN [4] takes domain labels as input to generator, and produces target domain results. ELEGANT [25] divides the latent code into different parts. Each part encodes information of one attribute. Visually, our model accomplishes more natural

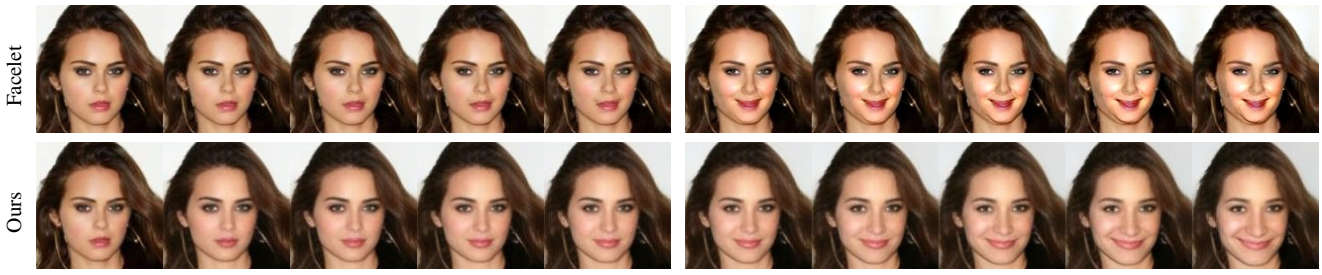


(a) Original (b) Angry (c) Sad (d) Happy (e) Contemptuous (f) Disgusted (g) Fear (h) Surprise (i) Look Left (j) Look Right

Figure 6. Multi-domain image-to-image translation on RaFD [12].

Original Young Mustache Not Smile Close Mouth Black Hair Bangs Hairline Female

Figure 7. Multi-domain image-to-image translation on CelebA [16].



(a) Interpolation (b) Extrapolation

Figure 8. Illustration of attribute interpolation and extrapolation. (a) is the result of interpolation. (b) further increases the edit strength to perform exaggeration.

– and with significant changes – results than ELEGANT [25] and StarGAN [4]. ELEGANT [25] assumes each attribute can be well disentangled into different parts of the latent code. This is not easily achieved because several attributes are intrinsically correlated. As a result, the training is not stable, causing sometimes noisy results. StarGAN [4] works well, and yet still occasionally produces strong edit, leading to visual artifacts.

Continuous Translation With the well learned latent space, our model allows synthesizing images across different domains. This has already been shown in Figs. 1 and 2. We also note that a good latent space should uncover the structure of natural image manifold [2]. To an extreme, it should even gain the capacity of *extrapolation*. This allows

	Young	Male	Smiling	Black_Hair	Bangs	Mustache	Hairline	Mouth_Open	Total
ELEGANT [25]	20%	27%	22%	36%	48%	41%	23%	35%	31%
StarGAN [4]	28%	24%	23%	47%	42%	47%	21%	34%	33%
Facelet [3]	25%	30%	35%	24%	25%	49%	10%	16%	27%
Ours	41%	18%	43%	49%	43%	48%	33%	45%	40%

Table 3. Turing Test on CelebA dataset. Each entry reports the percentage of taking the edited image as real. Higher is better.

	Young	Male	Smiling	Bangs	Black_Hair	Mustache	Hairline	Mouth_Open	Total
Ours > StarGAN	64%	51%	83%	50%	72%	49%	46%	74%	61%
Ours > Facelet	76%	72%	67%	57%	83%	49%	90%	80%	72%
Ours > ELEGANT	89%	88%	50%	59%	65%	51%	76%	71%	69%

Table 4. A/B Test on CelebA dataset. Each entry reports the percentage that our results are preferred. Larger than 50% indicates that our method is statically more preferred by the subjects.

exaggerating the difference between two domains. Fig. 8 compares the interpolation/extrapolation capacity between our model and Facelet [3].

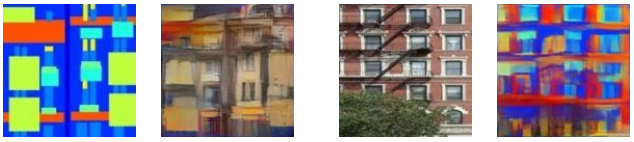
Facelet [3] is a feature interpolation approach whose latent feature is defined by a pretrained VGG network. Similar to ours, it requires only discrete attribute labels and has the capability to translate between different domains smoothly. However, when applying very strong edit strength, the result quality could drop. In contrast, our model works consistently well in both situations of interpolation and extrapolation. This indicates that the encoder trained by Eq. (6) actually unfolds the natural image manifold, leading to a flat and smooth latent space that allows interpolation and even extrapolation.

4.2.2 User Study

We also conduct user study on the Amazon Mechanical Turk platform to compare our performance with others. Turing Test and A/B Test are conducted.

Turing Test Each time subjects are presented with an arbitrary real image and the other that is edited by one method. Both images are normalized to 128×128. Subjects are requested to pick the real one. Table 3 shows the percentage that an edited image is regarded as real. Note that different attributes are counted separately, each includes 2,500 comparisons. Higher value means that human is harder to distinguish between the real image and the edited one. The final statistics show that our model has 40% chance to fool human eyes, which outperforms StarGAN [4] (33%), ELEGANT [25] (31%) and Facelet [3] (27%). We also note for *Male* attribute, people are easier to identify the edited image. The reason might be that our model only changes gender traits on faces, while the hairstyle or clothes are also highly correlated with gender. Therefore, subjects can recognize the edited image based on the incompatibility of faces and other cues.

A/B Test A/B Test refers to the pair-wise comparison of our model and another baseline model. Each time subjects are given an original image and two edited ones (our method vs. another), and are asked to pick one with higher edit quality. All three images are scaled to 128×128 and placed in one row. Similar to the Turing Test, different attributes are separately counted, and each one includes 2,500 comparisons. Table 4 presents the percentage that images generated by our method are chosen. Overall, our method outperforms StarGAN [4], ELEGANT [25], and Facelet [3] by 61%, 72% and 69% respectively.



Label ⇒ Facade Facade ⇒ Label

Figure 9. A failure case. Our method does not perfectly handle the situation when two domains are essentially different.

4.3. Limitations

Our model relies on the assumption that images of different domains can be embedded in a smooth and flat space. This is hardly achieved when these domains are very different. Fig. 9 illustrates a case that performs translation between facade images and semantic labels. Our model does not perform well in this case, since it is very difficult to find intermediate regions in between.

5. Concluding Remarks

We have proposed a framework for unpaired image-to-image translation focusing on generating natural and gradually changing intermediate results. Our method is based on latent space interpolation, which intrinsically allows continuous translation. In addition, by learning a controllable interpolator, we flexibly select the interpolation path, which alters the target attribute while keeping others almost intact. We have also shown that our method can serve multi-domain and multi-modal image-to-image translation.

References

[1] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai. Better mixing via deep representations. In *ICML*, 2013. 2, 3

[2] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv*, 2018. 2, 7

[3] Y.-C. Chen, H. Lin, M. Shu, R. Li, X. Tao, Y. Ye, X. Shen, and J. Jia. Facelet-bank for fast portrait manipulation. In *CVPR*, 2018. 2, 3, 8

[4] Y. Choi, M. Choi, and M. Kim. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 1, 3, 6, 7, 8

[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 3

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 3

[7] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 3, 5, 6

[8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[9] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 1, 3

[10] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv*, 2018. 2

[11] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 1

[12] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 2010. 3, 5, 7

[13] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv*, 2015. 2

[14] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 1, 3

[15] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 3

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 5, 7

[17] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 2018. 3

[18] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2014. 3

[19] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH*, 1985. 4

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014. 3, 4

[21] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. *arXiv*, 2017. 1, 2

[22] D. Ulyanov, A. Vedaldi, and V. Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *AAAI*, 2018. 1, 2

[23] P. Upchurch, J. R. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Q. Weinberger. Deep feature interpolation for image content changes. In *CVPR*, 2017. 2, 3

[24] T. White. Sampling generative networks. *arXiv*, 2016. 1

[25] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *ECCV*, 2018. 6, 7, 8

[26] Z. Yi, H. R. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. 3

[27] B. Zhao, B. Chang, Z. Jie, and L. Sigal. Modular generative adversarial networks. In *ECCV*, 2018. 3

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1, 3