Learning Style Subspaces for Controllable Unpaired Domain Translation

Gaurav Bhatt
University of British Columbia
Vancouver, Canada
gauravbhatt.cs.iitr@gmail.com

Vineeth N Balasubramanian
Indian Institute of Technology Hyderabad
Hyderabad, India
vineethnb@cse.iith.ac.in

Abstract

The unpaired domain-to-domain translation aims to learn inter-domain relationships between diverse modalities without relying on paired data, which can help complex structure prediction tasks such as age transformation where it is challenging to attain paired samples. A common approach used by most current methods is to factorize the data into a domain-invariant content space and a domain-specific style space. In this work, we argue that the style space can be further decomposed into smaller subspaces. Learning these style subspaces has two-fold advantages: (i) it allows more robustness and reliability in the generation of images in unpaired domain translation; and (ii) it allows better control and thereby interpolation of the latent space, which can be helpful in complex translation tasks involving multiple domains. To achieve this decomposition, we propose a novel scalable approach to partition the latent space into style subspaces. We also propose a new evaluation metric that quantifies the controllable generation capability of domain translation methods. We compare our proposed method with several strong baselines on standard domain translation tasks such as gender translation (male-to-female and female-to-male), age transformation, reference-guided image synthesis, multi-domain image translation and multi-attribute domain translation on celebA-HQ and AFHQ datasets. The proposed technique achieves state-of-the-art performance on various domain translation tasks while outperforming all the baselines on controllable generation tasks. Code - https://github.com/GauravBh1010tt/Controllable-Domain-Translation

1. Introduction

Translating data from one domain to another requires modeling complex inter-domain relationships. Recent efforts have shown the efficacy of deep generative models such as Generative Adversarial Networks (GANs) and other latent variable models for domain translation tasks [26, 18, 1, 16, 25, 20, 8, 6, 14, 5, 23]. A specific setting of domain translation that has received attention in recent years is the use of unpaired data. In many applications, access to paired data can be a bottleneck as it requires significant human effort for data capture. Unpaired domain-to-domain translation (UDT) leverages unmatched pairs from two different domains to learn the domains' structural and semantic relationships. The objective of UDT is to translate data from source to target domain with the constraint that the paired samples are not available, i.e., for any sample in the source domain, we do not have access to its counterpart in the target domain, which makes this problem challenging.

A common approach adopted by most recent UDT methods is to factorize the image distribution into a domain-invariant content space that helps capture information across domains and a domain-specific style space to capture the style variations [12, 25, 20, 16, 9]. However, using a single style space to capture different stylistic variations in the image may restrict reliable translation of domains as different stylistic variations may get mixed up at various levels. For example, consider two commonly used cases in gender translation: (i) generating female images corresponding to a bald male image; and (ii) generating a male image with a beard corresponding to a female image. In such cases, reliably translating images based on a particular style attribute could allow faithful translation. A possible remedy is to decompose the style space into smaller style subspaces where each subspace controls a unique stylistic variation in the data. This, in turn, allows controlled interpolation of the entire style space. In this work we focus on learning partitioned style space to achieve reliable UDT which helps us to mitigate the limitation of unreliable translations by existing UDT methods such as [12, 25, 20, 16, 9, 8].

Realizing such an idea of style decomposition may require several subspace encoders in existing methods that may not be scalable in practice. Another essential aspect to consider here is that though the style subspaces are not dependent on each other, they may not be mutually exclusive during generation. That is, an image may be generated

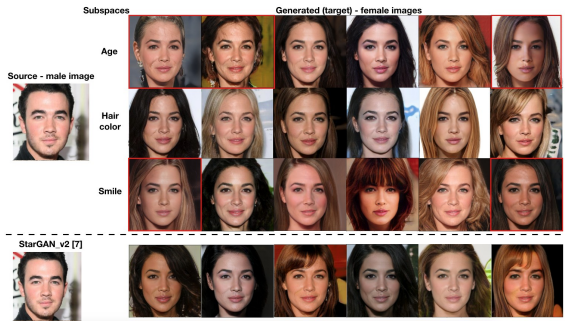

Figure 1. Controllable male-to-female translation. Top block captures our method's results with top-row representing the style-subspace '*age*' (older, younger), middle-row representing '*hair color*' (black, blonde, and brown), and last-row representing '*smile*' (smiling, non-smiling). Bottom block shows results obtained using StarGAN_v2 [8] for comparison. The proposed method generates samples from diverse subspaces such as old/young, smile/non-smile; whereas StarGAN_v2 lacks control over such diversity in its translations.

by combining inputs from multiple subspaces. Hence, using separate encoder networks may not be a feasible solution to incorporate such style subspaces. In this work, we propose a new methodology to learn a single latent space to represent style elements of domains but decompose this style space by partitioning it into smaller subspaces during the learning itself. This is achieved by associating a group of latent dimensions to a particular subspace using a learnable partition network. Interpolating across a particular latent group allows us to reliably translate samples focused on a specific attribute represented by a style subspace, as we show in our results. Learning the partitioned style space gives us the benefit of combining multiple subspaces during translation. For instance, using the proposed method, we can translate a given male image to a female image which is old, have a specific hair color *and* is smiling/non-smiling. Most existing UDT methods (such as ([17, 4, 12, 25, 20, 16, 9, 8])) fail to achieve such granularity in translation which is achieved by the proposed method. Moreover, we design our architecture and training procedure to incorporate all essential properties of an effective domain translation - high-quality image synthesis, handling multiple domains, reference-guided image synthesis, controllable generation, and scalable training. Figure 1 presents sample results that show the capabilities of our method, comparing them to StarGANv2 [8], a widely used UDT method. Our key contributions in this work are summarized below:

- We hypothesize and demonstrate that decomposing the domain-specific style space in unpaired domain-to-domain (UDT) translation tasks into smaller style subspaces allow for reliable controllable translation.
- We present a new method to learn the style space via partitioning into subspaces that captures the requirements of controllable UDT translation. Our training procedure is scalable in practice and can be used to synthesize high-quality images (256×256) when trained on a single Tesla P100 GPU.
- To evaluate the performance of domain translation methods on the task of controllable generation, we introduce a new evaluation metric that quantifies the extent of reliable generation by any domain translation method.
- We present comprehensive results, including high-resolution generation, that show the usefulness and control demonstrated by our approach over state-of-the-art baselines on benchmark datasets, CelebA-HQ and AFHQ.

2. Related Work

Most of the work in unpaired domain translation can be broadly categorized into five categories: high-quality image synthesis, handling multiple domains, reference-guided image synthesis, controllable generation, and scalable training. We describe related efforts from each of these perspectives below.

High-quality image synthesis - Existing work in domain-to-domain translation is dominated by the use of Generative Adversarial Networks (GANs) with cyclic consistency [26, 27, 18, 1, 12, 7, 15, 22, 2, 3, 16]. Cycle-GAN [26] was the first work to introduce a cyclic consistency idea that was effective when pairing information between domains was not available. Cycle-GAN inherently models an image-to-image mapping, which also results in mode collapse and limits high-quality image generation. Later works showed that such mode collapse could be resolved by embedding the latent spaces [1, 12, 16, 25, 20]. Embedding latent spaces with GANs results in diverse image synthesis as the model learns multi-modal mappings.

Handling multiple domains - Most existing UDT methods are restricted to two domains [26, 12, 16, 25, 20]. StarGAN [6] was the first method that efficiently handled multiple domains simultaneously. However, StarGAN does not employ embeddings of a latent space and thus learns a deterministic mapping. More recently, StarGAN_v2 [8] proposed to handle multiple domains where the mappings are multi-modal.

Reference-guided image synthesis - The idea of dividing the data into a content and style space gives access to reference-guided image synthesis, where an image can be translated using the style information of a given reference image. MUNIT-GAN [12] was among the first methods that introduced this idea. Quite recently, several other works have followed suit and allow reference-guided synthesis by using content and a style space [16, 25, 20, 8].

Controllable generation - While most of the focus in UDT translation has been on high-quality image synthesis or han-

Method	High-quality	multi-domain	Ref-guided	controllable	scalable
UNIT-GAN [18]	-	-	-	-	-
CycleGAN [26]	-	-	-	-	-
MUNIT-GAN [12]	✓	-	✓	-	-
DRIT++ [16]	✓	✓	✓	-	-
Council-GAN [20]	✓	-	✓	-	-
ACL-GAN [25]	✓	-	✓	-	-
StarGAN_v2 [8]	✓	✓	✓	-	✓
DosGAN [17]	-	-	-	✓	-
Homographic [4]	-	✓	-	✓	-
Proposed	✓	✓	✓	✓	✓

Table 1. Categorization of UDT translation methods.Here, ✓denotes if a particular property is satisfied.

dling multiple domains, the idea of controllable generation (where the generated image can be controlled by factors that can be specified) is addressed only by a few [4, 17, 14]. DosGAN [17] uses an attribute pre-trained classifier during domain translation, which helped to learn disentanglement among the domains. In another work, [4] introduced homographic interpolation of latent space to achieve controllable generation for UDT translation.

Scalable training - Most domain translation methods are based on generative models that are computationally expensive to train. Hence, scalability is one of the important factors to consider while proposing any new architecture. Most UDT translation methods are comprised of a pair of generators and discriminators (from source-to-target, and vice versa) [12, 16, 20, 25], which makes it infeasible for them to handle multiple domains simultaneously. Some new methods have adopted different training paradigms, such as learning a council of generator-discriminator networks [20], or leveraging adversarial consistency [25]; nonetheless, their training procedure is not scalable to incorporate multiple domains. Scalability issue is better handled by StarGAN [6] and its successor StarGAN_v2 [8]. StartGAN_v2 introduced the idea of using a single generator framework that reduces computational burden and allows scalable training.

In this work, we partition the style space into smaller subspaces by introducing a partition network. The idea of learning style-subspace specific latents gives us an advantage of high-quality reliable UD2D translation over existing non-controllable methods such as [12, 25, 20, 16, 9, 8]. When compared to controllable generation methods such as [17, 4], the proposed methods achieves granular translations such as combining multiple style subspaces (to be discussed in Section 4.2), which is not achieved by existing controllable generation methods. We further discuss the relationship of proposed method to existing methods in Section 3.6.

3. Proposed Method

3.1. Problem Formulation

We begin by introducing the data from a source domain $X_{src} \sim \mathcal{P}_{src}$, and a target domain $X_{trg} \sim \mathcal{P}_{trg}$ (P_{src} and P_{trg} denote the probability distribution over the domains X_{src} and X_{trg}). Our broad aim is to learn the translation, $g(\cdot)$ from X_{src} to X_{trg}. We operate in the challenging setting where samples available from the two domains during training are unpaired , i.e., $\forall x_s \in X_{src}, g(x_s) \notin X_{trg}$. Our specific objective is to learn $g(\cdot)$ through a latent space which can be decomposed into subspaces that allow us to control the generation of samples in $\mathcal{P}_{trg}$ given $x_s \in X_{src}$. For example, in human facial images, the subspaces could correspond to attributes such as hair color, age, or smile. We refer to these as *style subspaces* for this reason.

Issue of Non-identifiability. Due to the unpaired setting, it may not be possible to identify the style subspace relevant for a data pair $\{x_s, x_t\}$ given during training. For instance, in the case of male-to-female translation, sample x_s could be a male with black hair and a beard, while sample x_t may belong to a female with blonde hair and no facial hair. Considering there is more than one attribute changing, associating x_s and x_t to a common subspace (say, hair color) is not possible. We refer to this issue as *non-identifiability*.

To solve this problem, we hypothesize a joint distribution of both domains conditioned on a latent distribution that can be factorized based on the style subspaces. We describe our framework to achieve this below.

3.2. Proposed Framework

Our proposed architecture, shown in Figure 1, has four basic modules: the style encoder, mapping network, partition network, and the generator (with a style mixing network). Since it is possible to train in both directions in an unpaired domain translation task, we refer to the two domains as $\mathcal{D}_1$ and $\mathcal{D}_2$ going forward (instead of source and target), for convenience and clarity of presentation. We assume that there are K style subspaces that the latent space is partitioned into.

Style Encoder. Our style encoder E takes in an image $x_i; i \in \{\mathcal{D}_1, \mathcal{D}_2\}$ and extracts a style vector $f_x = E_2^i(E_1(x_i)), i \in \{\mathcal{D}_1, \mathcal{D}_2\}$, where $f_x \in \mathbb{R}^l$ is an l-dimensional vector, E_1 corresponds to shared layers of the encoder and E_2^i corresponds to domain-specific layers. The style encoder is designed such that only the last block of neural network layers is specific to domain i (rest of the layers are shared across domains). This architecture design allows our training to be efficient.

Mapping Network. To induce diversity among the translated images, we introduce a mapping network M which takes a noise vector from the standard normal Gaussian, $z \sim \mathcal{N}(0, 1)$, and generates a random style vector $f_z = M_2^i(M_1(z)), i \in \{s, t\}$, where $f_z \in \mathbb{R}^l$, M_1 corresponds to shared layers and M_2^i corresponds to domain-specific layers (similar to the encoder). The mapping network's role is to transform a random Gaussian noise vector to a meaningful style vector that can provide diversity of generation in the target domain.

Partition Network. This component of our architecture is

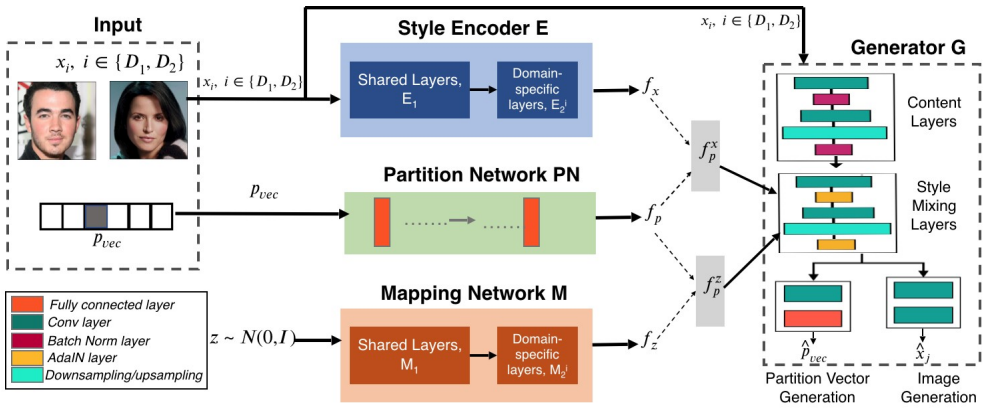

Figure 2. Architecture of our proposed model for controllable unpaired domain translation.

important to address the non-identifiability issue. To this end, we introduce a partition vector $p_{vec} \in \{0,1\}^K$ (where K is the number of subspaces) that stores the style subspace information for each training sample. In particular, p_{vec} is designed as a one-hot vector where the k^{th} position refers to the k^{th} subspace and is set to 1 when that style is present in a given image. (For e.g., the face image of a female with black hair and a smile would be associated with two p_{vec} vectors - one for hair color and another for the smile). Given input images from $\mathcal{D}_1$ and $\mathcal{D}_2$ with a common trait (say, black hair), we pass the corresponding one-hot partition vector p_{vec} to a partition network PN which transforms the partition vector to a l-dimensional style vector $f_p = PN(p_{vec})$. This network behaves like an additional input to guide the styled generation of the translated image. To achieve this, we concatenate f_p with f_x and f_z (defined above) to obtain two vectors:

$$f_p^x = [f_p; f_x] \text{ and } f_p^z = [f_p; f_z] \tag{1}$$

Learning f_p, f_p^x and f_p^z facilitates the identification of style dimensions that are responsible for generating samples corresponding to a specific subspace.

Generator (with Style Mixing Network). Our generator module G, as shown in Figure 1, first extracts content information from a given image $x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\}$. This content block consists of convolution, batch normalization and downsampling layers. The output of the content block is fed to a *style mixing network* which also takes in the outputs of the partition network, f_p^x and f_p^z, as input. The style mixing network consists of convolution blocks and adaptive instance normalization (AdaIN) layers [11] to facilitate this mixing. The parameters for AdaIN are obtained by passing f_p^x and f_p^z (separately) through a few neural network layers. The last part of the generator consists of an image generation block and a partition vector generation module. Both these blocks take the output of the style mixing network as input. The image generation block synthesizes a translated image in the target domain while the partition vector generation computes a partition vector corresponding to the given source-target pair, i.e. we have:

$$\begin{aligned}[\hat{x}_j^x,\ \hat{p}_{vec}^x] &= G(x_i, f_p^x) \\ [\hat{x}_j^z,\ \hat{p}_{vec}^z] &= G(x_i, f_p^z)\end{aligned} \tag{2}$$

where $i, j \in \{\mathcal{D}_1, \mathcal{D}_2\}$ such that $i \neq j$ (i.e. the input and output are from two different given domains). The generator hence outputs $[\hat{x}_j^x,\ \hat{p}_{vec}^x]$ corresponding to the real image input, and $[\hat{x}_j^z,\ \hat{p}_{vec}^z]$ corresponding to the noise input. We show in the next section how these are used to train the overall framework.

3.3. Training Objectives

We now describe the objectives used to train our framework.

Adversarial training. Given a source image sample $x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\}$ and its partitioned style vector f_p^z sampled from the mapping network M, the generator synthesizes a translated image in the target domain, $\hat{x}_j, j \neq i$. We use a discriminator D^i for each domain (the architecture is similar to the style encoder) to compute an adversarial loss:

$$\begin{aligned}\mathcal{L}_{Adv}^z = &\mathop{\mathbb{E}}_{x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\}} \Big[\log D^i(x_i)\Big] + \\ &\mathop{\mathbb{E}}_{\substack{x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\} \\ z \sim \mathcal{N}(0,1)}} \Big[\log(1 - D^i(G(x_i, f_p^z)))\Big]\end{aligned} \tag{3}$$

We compute the above loss given input samples from both $\mathcal{D}_1$ and $\mathcal{D}_2$. A similar loss is also computed for the other output obtained through the generator as below:

$$\begin{aligned}\mathcal{L}_{Adv}^x = &\mathop{\mathbb{E}}_{x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\}} \Big[\log D^i(x_i)\Big] + \\ &\mathop{\mathbb{E}}_{x_i, i \in \{\mathcal{D}_1, \mathcal{D}_2\}} \Big[\log(1 - D^i(G(x_i, f_p^x)))\Big]\end{aligned} \tag{4}$$

Eqn 3 promotes generation of diverse samples from the target distribution, while Eqn 4 promotes generation of samples similar in style to the given input, x_i, from the target distribution.

Partition vector loss. Given the partition vector $\hat{p}_{vec}$ generated by the generator (as described above), we compute the partition loss by minimizing the squared error of the generated partition vector with the original input partition vector p_{vec}:

$$\mathcal{L}_{par} = \frac{1}{2}\Big(||\hat{p}^x_{vec} - p_{vec}||_2^2 + ||\hat{p}^z_{vec} - p_{vec}||_2^2\Big) \quad (5)$$

The partition loss encourages the generator not only to generate a relevant image from the target domain but also to be aware of the style under consideration (this allows us to change the input p_{vec} to other styles at inference to controllably generate the same content with different styles).

Partition consistency loss. To ensure that the translated image preserves the properties of the source domain, we introduce a variant of cyclic consistency loss. We compute the partition style vector corresponding to the original sample x_i and the generated sample $\hat{x}_i$, and compute the L_2-loss between these two vectors:

$$\mathcal{L}_{pc} = |||[f_p; E(x_i)] - [f_p; E(\hat{x}_i)]||_2^2 \quad (6)$$

where $\hat{x}_i$ is generated using the generator $G(x_i, f_p^x)$ (see Eqn 2), $E(x_i)$ denotes the output of the style encoder for x_i, $E(\hat{x}_i)$ denotes the output of the style encoder for $\hat{x}_i$, and each $[f_p; E(x)]$ term is as defined in Eqn 1. Note that while L_2 norm on the image space may not reflect human visual perception, L_2 norm in the latent space is generally more meaningful. Using the partitioned latent space allows us to use simple L_2 or L_1 norms effectively for comparisons.

Other Losses. In addition to the above loss terms, we also use style reconstruction loss as in [8][12], and the $R1$-regularizer loss as in [19], considering their effectiveness for domain translation. These loss terms help achieve good quality translations and better training convergence. We denote these together as $\mathcal{L}_o$.

Final training objective. Our final training objective is given as:

$$\min_{E,M,PN,G} \max_{D} \mathcal{L}^x_{Adv} + \mathcal{L}^z_{Adv} + \lambda_{par}\mathcal{L}_{par} + \lambda_{pc}\mathcal{L}_{pc} + \lambda_o\mathcal{L}_o \quad (7)$$

where λ_{par}, λ_{pc}, and λ_o controls the weights given to the partition loss, consistency loss, and other losses respectively.

3.4. Training and Inference

Training. During training, we switch between providing images from both $\mathcal{D}_1$ and $\mathcal{D}_2$ as input, so that the model learns the bi-directional mapping. In each case, the corresponding domain-specific layers are activated while the rest of the network is shared (see Figure 1). Our framework also allows translation from a domain to itself, where the generator generates images from the source domain itself (we call this *self-modal* translation). In other words, given domains $male$ and $female$, we learn *cross-modal translations*: $\{male \rightarrow female\}$ and $\{female \rightarrow male\}$; as well as *self-modal translations*: $\{male \rightarrow male\}$ and $\{female \rightarrow female\}$. In our implementation, we do a coin flip to choose between source and target domain for each batch. This simple strategy is scalable and works well in all our experiments.

Inference. During inference, our model has two options to translate an image: we can extract the style information from a reference image using the style encoder E, and then using the appropriate domain information, generate the target domain image using the generator G. Alternatively, we can sample a random style vector from the mapping network M and synthesize the image using the generator G. Interestingly, we demonstrate that we can achieve controllable generation at inference by simply changing the partition vector value p_{vec}. Since each dimension of p_{vec} is now associated with a particular style subspace, we can interpolate across this k^{th} subspace by varying the k^{th}-dimension of p_{vec}. Our model also allows combining multiple subspaces (where p_{vec} is passed as a binary vector instead of a one-hot vector), which provides image generation with multiple desired attributes.

3.5. Evaluation of Controllable Generation

Existing methods that show controllable UD2D translation [17][4] broadly use qualitative results to demonstrate the control. To improve this, we introduce a new quantitative metric to study average controllable generation (ACG) by a domain translation method. ACG is defined as the average classification accuracy of attributes in a dataset obtained by training a classifier on the generated images for each attribute's presence and absence.

For example, we consider images on attributes provided in the widely used CelebA dataset – smile, age, hair color (black, brown, blonde), lips, and nose size. We also use a combination of attributes such as smile+age, smile+black hair, and so on, providing us a total of 16 style attributes (which define our style subspaces). Next, we train a multi-label classifier (C_{ACG}) with the output layer neurons as 16. For a given domain translation model $\mathcal{M}$, we generate images conditioned on the above 16 subspaces. Let the source image passed to $\mathcal{M}$ be x_s and the translated image for a subspace k be $\hat{x}^k = \mathcal{M}\ (x_s, k)$. We compute the classification accuracy of the classifier trained on the above data. We expect that for a given subspace, higher classification performance should be achieved if the quality of translations are good. ACG is hence computed as:

$$ACG = \frac{1}{N}\frac{1}{15}\sum_{k=1}^{15}\sum_{n=1}^{N} Acc(C_{ACG}(\hat{x}^k_n); c^k_n) \quad (8)$$

where N is the number of samples generated for each subspace, and c^k_n is the ground truth subspace for the given

sample. (We use $N = 500$ in our experiments). The higher value of ACG corresponds to a better controlled generation. We also compute $ACG@r$, where the model $\mathcal{M}$ is given r chances to predict the subspace label. For all controllable generation methods – DosGAN [17], homographic interpolation [4], and ours – we keep the value of $r = 1$, while for other methods that don't offer explicit control, we vary r between 1 to 5 to give them a better chance.

3.6. Relationship to Existing UD2D Methods

Our idea of introducing a mapping network is also explored in some styleGAN-based works [21, 8]. The mapping network has an advantage over VAE-based methods that try to achieve control in the generation, as shown in these efforts. The proposed method shares some similarities with StarGAN-v2 [8], such as using domain-specific layers to handle multiple domains and the mapping network. However, StarGAN-v2 had a different objective, and the addition of partition network and partition loss allows us to address the non-identifiability issue and provide us with the ability to achieve controllable generation over StarGAN-v2. We also factor the image into a content space, and a style space, similar to MUNIT-GAN [12] and its most recent variant ACL-GAN [25]. Nonetheless, none of the MUNIT-GAN variants (Council-GAN[20], ACL-GAN[25]) achieve controllable generation. Moreover, these methods use multiple generators-discriminators for bi-directional translation. We use a simpler architecture to achieve our objectives herein.

DosGAN [17] introduces a classification loss using a pre-trained classifier in terms of controllable generation. Their architecture is not trained end-to-end and hence falls short in smoothly interpolating the latent space. A more recent method [4] uses an interpolation network to learn various paths for UD2D translation. Their method is restricted to a single path at a time and thus falls short of learning complex granular translations that are achieved by combining multiple subspaces (see Sec 4 for our study). For example, their method does not support male-to-female translation when conditioned on multiple attributes. Our idea and approach to partitioning the attribute space give us more flexibility and robustness to control the style space while maintaining quality and diversity.

4. Experiments and Results

We evaluate the performance of our model on a variety of UDT tasks on celebA-HQ [13] (celebrity with attributes) and AFHQ [8] (animal faces HQ) datasets: gender transformation (male-to-female and female-to-male), age transformation (young-to-old, old-to-young), reference-guided image synthesis, multi-attribute domain translation, and multi-domain image translation. We compare our work with several strong baselines for UDT such as CycleGAN [26], MUNIT-GAN [12], DRIT++ [16], Council-GAN [20], ACL-GAN [25], StarGAN_v2 [8], DosGAN [17], and homographic interpolation [4]. In all our experiments, we set λ_{par} and λ_{pc} to 1.

To evaluate the diversity and quality of translated images, we use LPIPS score (Learned Perceptual Image Patch Similarity) [24] and FID metric (Frechet Inception Distance) [10]. For LPIPS score, we generate 10 samples for each testing data, while for FID, we generate 5000 fake samples used to compute the generated distribution statistics. (Due to space constraints, please refer to our Supplementary section for implementation details).

4.1. Dataset Preparation

We use 10 attributes provided in the CelebA-HQ dataset as style-subspaces: 'Bags Under Eyes', 'Big Lips', 'Blond Hair', 'Big Nose', 'Black Hair', 'Double Chin', 'Oval Face', 'Smiling', 'Brown Hair', 'Young'. Before training, we divide the data into its respective subspace, where we have 5000 pairs (male and female) per subspace, amounting to a total training set of 50000 pairs. For testing on celebA-HQ, we use the same split provided by [8] for fair comparison; this consists of 1000 male-female pairs in the test set.

To evaluate the performance of the proposed method on reference-guided image translation, we use the animal translation task using the AFHQ dataset [8]. Specifically, we perform cat-to-dog and dog-to-cat translation. The AFHQ dataset consists of 5000 training samples of dogs and cats, while 1000 samples from each domain are used for testing.

4.2. Results and Discussion

Gender translation. The results on gender translation tasks is shown in Table 2. Here, we evaluate the proposed model on male-to-female and female-to-male translation tasks. We use the mapping network M to translate the given source sample using a Gaussian noise $z = \mathbb{N}(0, 1)$. The proposed method achieves the highest LPIPS value and lowest FID score, suggesting that we can maintain diversity and quality among the generated images. Other methods that attempt controllable generation - DosGAN [17] and homomorphic interpolation [4] - do not produce high-quality samples, as reflected in their higher FID scores.

Controllable generation. We use the proposed ACG metric to evaluate the proposed method's performance on controllable generation (Table 3). We use ACG@1 for all controllable UDT methods such as DosGAN [17] and homomorphic interpolation [4]. For other methods such as StarGAN_v2 [8], MUNIT-GAN [12], ACL-GAN [25] which do not attempt controllable generation, we evaluate the performance on ACG@1 and ACG@5 to give them their best chance (please refer to supplementary section for more experiments). The proposed method outperforms all baselines

Method	male-to-female LPIPS ↑	male-to-female FID ↓	female-to-male LPIPS	female-to-male FID
MUNIT-GAN [12]	0.36	19.02	0.35	23.42
DRIT++ [16]	0.37	24.61	0.35	25.12
Council-GAN [20]	0.42	18.10	0.41	21.16
Homomorphic [4]	0.40	21.42	0.39	23.12
ACL-GAN [25]	0.43	16.63	0.43	18.31
DosGAN [17]	0.38	22.15	0.37	24.23
StarGAN_v2 [8]	0.45	13.92	0.44	16.78
Proposed	**0.46**	**11.79**	**0.45**	**16.42**

Table 2. Results on gender translation task on celebA-HQ (male-to-female and female-to-male). We evaluate the performance based on quality and diversity of generated samples. ↑ denotes higher the better, while ↓ means a lower value is desirable.

Method	male-to-female ACG@1	male-to-female ACG@5	female-to-male ACG@1	female-to-male ACG@5
MUNIT-GAN [12]	0.40	0.41	0.37	0.39
ACL-GAN [25]	0.41	0.42	0.38	0.40
StarGAN_v2 [8]	0.43	0.45	0.40	0.42
DosGAN [17]	0.45	-	0.45	-
Homomorphic [4]	0.47	-	0.47	-
Proposed	**0.57**	-	**0.59**	-

Table 3. Results on Average Controllable Generation on CelebA-HQ dataset

on the controllable generation task. The ACG metric varies a lot between controllable and non-controllable methods which justifies their (non-controllable methods) inability to learn reliable domain translation. There are many cases where non-controllable methods consistently fail, such as shown in Figure 1, where for the given male image, the StarGAN_v2 is unable to generate an old female image or a no-smiling female image. In contrast, the proposed method can reliably translate images using different style-subspaces, achieving better control over the generations.

Forward-backward interpolation. During training, the partition vector (p_{vec}) learns latent-to-subspace partition by associating k^{th} dimension of p_{vec} to a group of latents that controls a particular subspace k. During inference, p_{vec} allows us to linearly interpolate across a particular style-subspace by varying the k^{th} dimension of p_{vec} and then generating target sample $\hat{x}^t$.

$$p_{vec}[k] = \delta; \delta \in \{-\infty, +\infty\} \tag{9}$$

$$f_p, f_z = PN(p_{vec}), M(z); z \in \mathcal{N}(0,1) \tag{10}$$

$$f_p^z = [f_p; f_j]; \hat{x}^t = G(x_s, f_p^z) \tag{11}$$

where $\delta \in \mathbb{R}$ is the weight given to the k^{th} subspace. Interestingly, we found out that it is possible to interpolate a subspace in the backward direction by using negative values for δ. The backward interpolation has significance in cases such as the 'age' subspace, where forward interpolation refers to making the translated image younger while backward interpolation refers to making it older. Similarly, for the 'smile' subspace, a positive value means a more smiling face while a negative value means lesser or no-smile. The result of forward-backward interpolation is shown in Figure 3 where we achieve controllable gender translation by interpolating across style-subspaces.

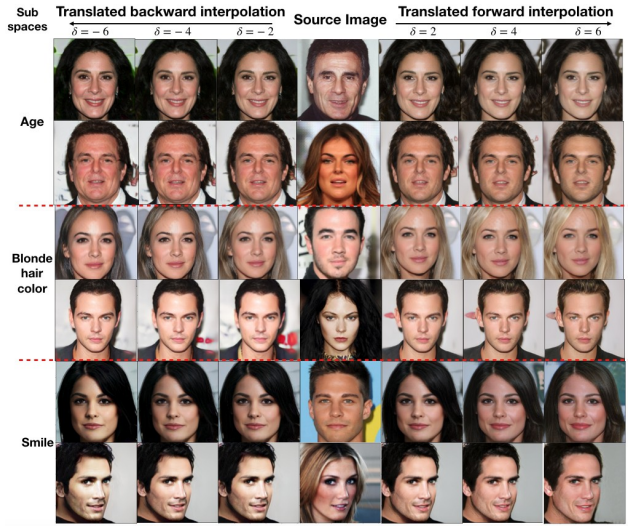

Figure 3. Forward-backward interpolation for gender translation. The top block represents interpolating across 'age' subspace, while the middle shows 'blonde hair color', and the bottom shows 'smile' subspace. In each of the block, the top row shows male-to-female translation and the bottom row shows female-to-male translation.

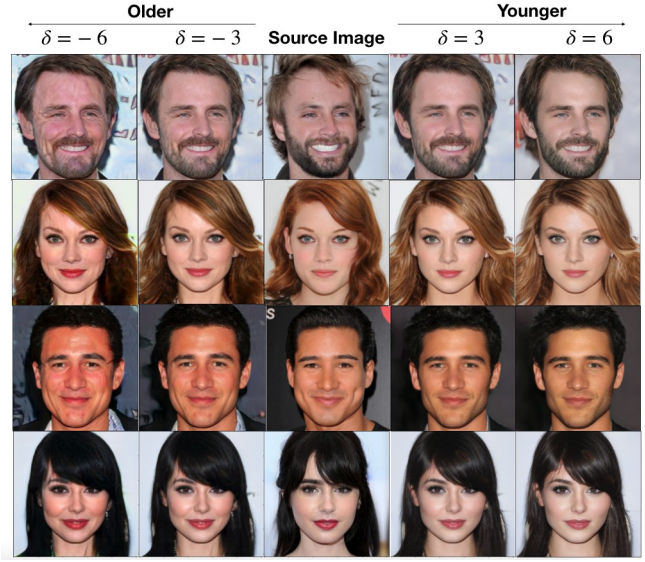

Figure 4. Results on age transformation. Here, forward translation corresponds to older-to-younger, while backward translation refer to younger-to-older translation.

Age transformation. The design of our architecture and training procedure allows us to also perform domain-specific transformation (self-modal translation, as noted in Section 3.4). At inference time, using domain-specific layers, we achieve age transformation (young-to-old and old-

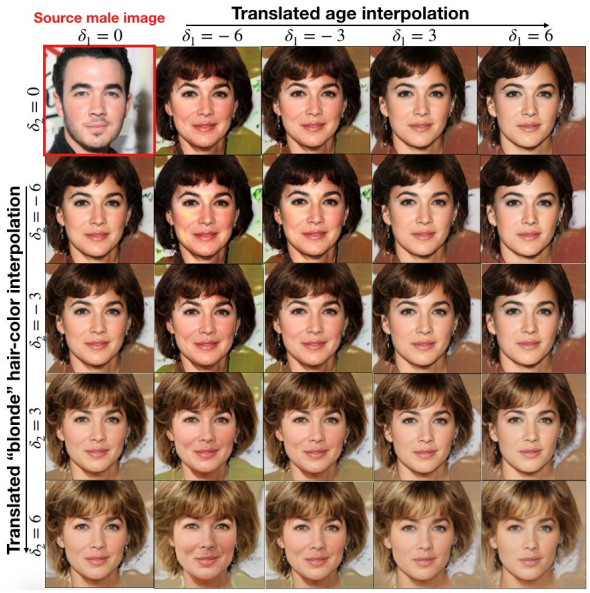

Figure 5. Combining 'age' and 'blonde hair color' style-subspace while performing male-to-female translation. Here, the horizontal axis is the 'age' style-subspace, and the vertical axis is the 'blonde hair-color' style-subspace.

to-young) on the same domain. The images are synthesized using Eqns 9 - 11, where the δ corresponding to age subspace is varied. The old-to-young translation is achieved by forward interpolation, while the young-to-old translation is achieved by backward interpolation. The result of age transformation is shown in Figure 4. Our method provides smooth interpolation of the style space. With negative values of δ, the generated image is older, while with positive values of δ, the generated image is younger.

Combining multiple style-subspaces. Our idea of learning style-subspaces gives us the capability of combining multiple subspaces, making it possible to achieve multi-attribute domain translation. As shown in Figure 5, we achieve male-to-female translation by interpolating across 'age' (k_1) and 'blonde hair color' (k_2) subspaces simultaneously. This is achieved by associating δ_1 to the 'age' subspace while δ_2 to 'blonde hair color' subspace. The partition vector in this case is a vector with weights δ_1 and δ_2. Figure 5 shows how the proposed method allows controllable generation across multiple subspaces by smoothly interpolating across the style-subspaces. These results demonstrate our effectiveness over controllable methods such as [4] and [17].

Reference-guided image synthesis. In reference-guided image synthesis, at inference time, we extract the content information from a given source image while style information is extracted from another reference image. Given two source images (x^c and x^s), we use the style encoder E to compute the feature vector f_x^s for image x^s. The generator G takes x^c as input while the style information is provided in the form of f_x^s. The results on dog-to-cat and cat-to-dog translation tasks are reported in Table 4. We use the FID and LPIPS scores to validate the quality and diversity of translations. The proposed method receives highest LPIPS score and lowest FID values when compared to other baselines. We also use the AFHQ dataset to evaluate the reference-guided image translation task. These results are shown in Figure 6.

Method	cat-to-dog		dog-to-cat	
	LPIPS ↑	FID ↓	LPIPS	FID
MUNIT-GAN [12]	0.29	21.32	0.30	45.23
DRIT++ [16]	0.30	20.15	0.33	44.87
Council-GAN [20]	0.32	16.23	0.39	40.98
Homomorphic [4]	0.30	19.25	0.35	43.68
ACL-GAN [25]	0.33	15.05	0.40	41.07
DosGAN [17]	0.30	19.64	0.34	47.34
StarGAN_v2 [8]	0.35	10.06	0.41	**39.05**
Proposed	**0.37**	**7.50**	**0.42**	39.31

Table 4. Results on animal translation task on AFHQ dataset (cat-to-dog and dog-to-cat).

Figure 6. Reference guided images translation on AFHQ dataset. Here, the content information (x^c) is extracted from top row, while the first column is used to get the reference style (x_s).

5. Conclusion

In this work, we proposed a novel architecture to achieve controllable generation in unpaired domain translation, based on learning style subspaces. To evaluate the model's capability of controllable generation, we introduced the ACG metric. The introduction of partition network and partition loss outperformed various UDT methods [8, 25, 17, 4] on the task of controllable domain translation on different datasets, showing the usefulness of the proposed method.

References

[1] Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 195–204, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[2] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *arXiv preprint arXiv:1802.10151*, 2018.

[3] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018.

[4] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2416, 2019.

[5] Eleni Chiou, Eleftheria Panagiotaki, and Iasonas Kokkinos. Beyond deterministic translation for unsupervised domain adaptation. *arXiv preprint arXiv:2202.07778*, 2022.

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[9] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in neural information processing systems*, pages 1287–1298, 2018.

[10] 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.

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[12] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[14] Kunhee Kim, Sanghun Park, Eunyeong Jeon, Taehun Kim, and Daijin Kim. A style-aware discriminator for controllable image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18239–18248, 2022.

[15] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[16] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, pages 1–16, 2020.

[17] Jianxin Lin, Zhibo Chen, Yingce Xia, Sen Liu, Tao Qin, and Jiebo Luo. Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[18] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.

[19] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[20] Ori Nizan and Ayellet Tal. Breaking the cycle-colleagues are all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7860–7869, 2020.

[21] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020.

[22] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[23] Dvir Yerushalmi, Dov Danon, and Amit H Bermano. Leveraging in-domain supervision for unsupervised image-to-image translation tasks via multi-stream generators. *arXiv preprint arXiv:2112.15091*, 2021.

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[25] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. *arXiv preprint arXiv:2003.04858*, 2020.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE*

international conference on computer vision, pages 2223–2232, 2017.

[27] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017.