SMOOTH IMAGE-TO-IMAGE TRANSLATIONS WITH LATENT SPACE INTERPOLATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-domain image-to-image (I2I) translations can transform a source image according to the style of a target domain. One important, desired characteristic of these transformations, is their graduality, which corresponds to a smooth change between the source and the target image when their respective latent-space representations are linearly interpolated. However, state-of-the-art methods usually perform poorly when evaluated using inter-domain interpolations, often producing abrupt changes in the appearance or non-realistic intermediate images. In this paper, we argue that one of the main reasons behind this problem is the lack of sufficient inter-domain training data and we propose two different regularization methods to alleviate this issue: a new shrinkage loss, which compacts the latent space, and a Mixup data-augmentation strategy, which flattens the style representations between domains. We also propose a new metric to quantitatively evaluate the degree of the interpolation smoothness, an aspect which is not sufficiently covered by the existing I2I translation metrics. Using both our proposed metric and standard evaluation protocols, we show that our regularization techniques can improve the state-of-the-art multi-domain I2I translations by a large margin. Our code will be made publicly available upon the acceptance of this article.

1 INTRODUCTION

The growing interest in generative methods, specifically in image manipulation approaches, goes beyond academia and is also motivated by the enormous application potential, for instance, in the entertainment and fashion industry. Modern deep generative networks can artificially change a photo according to some desired "attribute" (e.g. changing people's age), and are already applied in leading image editing applications (Adobe, 2021). From a scientific point of view, these image transformations are usually called Image-to-Image (I2I) "translations", and the attributes are represented by "domains" (e.g., women pictures), where each domain shares some distinctive visual pattern called "style". In Multi-domain and Multi-modal Unsupervised Image-to-Image Translation (MMUIT), a single generator network maps images into multiple domains, and the process is conditioned by some random noise in order to generate diverse images for the same input image (multi-modal appearance). Moreover, the training dataset is "unsupervised", because no image-to-image correspondence is given across the domains.

In this paper, we focus on learning a semantically smooth latent style space, which can be used for continuous MMUIT translations. By linearly interpolating the style representations of the source and the target image, the intermediate generated images should correspond to a gradual transformation of the input image (see Fig 1 (a)). Interestingly, while state-of-the-art MMUIT approaches (Choi et al., 2020; Lee et al., 2020) can generate highly realistic translations, they usually struggle in interpolations across domains. For instance, the across-domain interpolations results of StarGAN-v2 (Choi et al., 2020) are often unrealistic, with abrupt changes between two close interpolation points (e.g., see Fig 1 (b)). This issue makes it hard to interpolate not to mention extrapolate images, or "animate" a translation, and limits the control on the desired *degree* of the transformation.

We argue that one of the main reasons for this problem is the low density of the true data distribution in the inter-domain regions of the latent representation space, which is caused by the lack of sufficient training data representing across-domain images. This concept is intuitively shown in Fig 1, where the latent-space regions of two domains in two different tasks (cats↔dogs and women↔men)

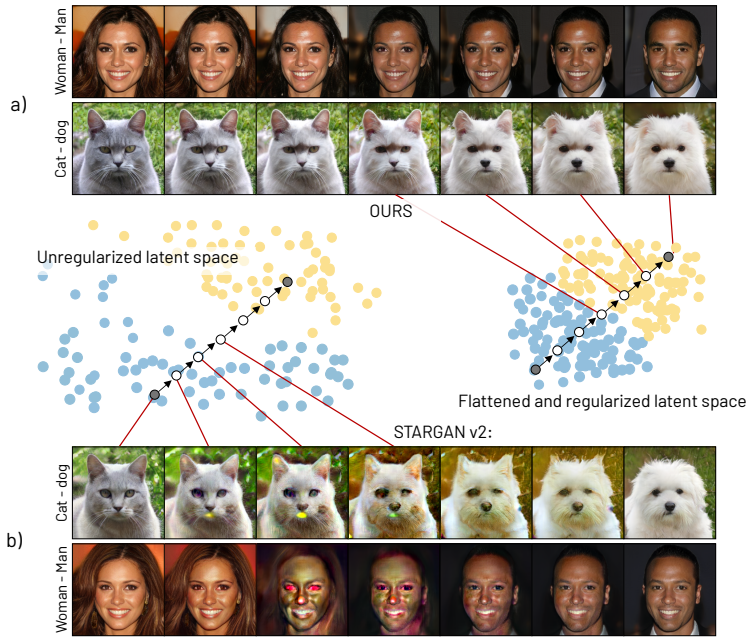

Figure 1: A schematic overview of our regularization approach. In the middle, blue and yellow points represent cat and dog (or woman and men) *training samples*, respectively. Linearly interpolating across the two domains leads to traverse a low density area of the true data probability distribution (middle left). Consequently, the corresponding generated images may look unrealistic (bottom). Conversely, our regularization compacts and flattens the latent space (middle right), resulting in a much smoother transition from one generated image to the next (top). Note that the middle figures are illustrative schemes, but the bottom and the top figures are images generated using StarGAN v2 (Choi et al., 2020) and on our regularized space, respectively.

are densely populated by training samples observed by the generator during training. However, since some real training photos are rare (e.g. people between two genders) or do not exist (e.g. half-cat and half-dog), the inter-domain region has not been sufficiently explored during training. Consequently, when we interpolate between two points belonging to these two domains, the inter-domain area may correspond to meaningless content once decoded by the generator. A similar phenomenon was studied by Tanielian et al. (2020), while Dai et al. (2017) exploit a *bad* generator to synthesize fake samples lying in the inter-class regions in a semi-supervised scenario. Finally, note that the same problem can affect the intra-domain areas: if the domain-specific training samples are too "scattered" in a large area, the generator may overfit the observed training points. To solve the overfitting problems related to non-compact representation spaces, Variational Auto Encoders (VAEs) (Kingma & Welling, 2013) regularize the latent space using a Kullback-Leibler divergence with respect to an a priori zero-centered Gaussian distribution. In this paper, we propose an alternative approach which can be used to regularize GAN-based MMUIT networks and produce higher-quality intra and inter-domain interpolations. Specifically, we propose two simple but effective regularization methods: (1) A new "shrinkage" loss for compacting the latent space, and (2) the use of Mixup (Verma et al., 2019) to generate inter-domain training samples.

The shrinkage loss is inspired by the *uniform* loss recently proposed by Wang & Isola (2020) to smooth the latent space of a discriminative network trained using self-supervision. The *uniform* loss shares the same goal of the variational regularization in VAEs, that is to make the distributions of the points in the representation space as uniform as possible. However, while the *uniform* loss has the effect of (uniformly) *spreading* the points on the surface of a unit sphere (using a Gaussian potential kernel), our shrinkage loss forces the points to (uniformly) come *closer* to each other. Moreover, we do not need to L_2-normalize our representations as in Wang & Isola (2020), an operation which is common in self-supervised learning (Chen et al., 2020; Grill et al., 2020; Caron et al., 2020) to increase the *invariance* of the representations (Wang & Isola, 2020) but which can lead to some information loss when the generation of the image details is important.

The second regularization method uses a Mixup strategy (Zhang et al., 2018a; Verma et al., 2019) in the latent space to populate the inter-domain regions with artificially generated training samples. Mixup-based methods are very popular data-augmentation techniques in discriminative networks, and, recently, they have also been used in a GAN scenario (Beckham et al., 2019). In our case, we mix the style representations of inter-domain pairs and we use these mixed samples at training time to generate e.g. people between different genders and "half-cat and half-dog" samples which are not included in the real training data. As far as we know, we are the first proposing a Mixup strategy in an MMUIT scenario.

Finally, we propose a new metric (Perceptual Proportionality, P^2) to evaluate the semantic smoothness of a latent space. As we show in §5, P^2 is simpler than the recently proposed Perceptual Path Length (PPL) (Karras et al., 2019) and it avoids different technical problems related to PPL.

Our contributions can be summarized as follows:

- We propose a new loss (the *shrinkage loss*) and a Mixup-based training strategy to smooth and regularize the latent space of GAN-based MMUIT and TUNIT networks. Both proposals are simple-to-reproduce and can be used in different MMUIT frameworks, jointly with standard losses and different architectural choices.
- We show that our approach, when plugged into two state-of-the-art MMUIT and TUNIT frameworks (StarGAN-v2 (Choi et al., 2020) and Baek et al. (2020)) leads to a large boost in the results and it is particularly effective when interpolations are used.
- We propose a new metric (P^2) that can be used to evaluate the smoothness of a semantic space.

2 Related Work

Image-to-image translation. The goal of I2I translation is to learn a mapping function which changes the domain-specific parts of the source image while keeping the domain-independent part. Early attempts are based on paired images (Isola et al., 2017; Siarohin et al., 2018; Zhu et al., 2017b), one-to-one domain mappings (Zhu et al., 2017a; Huang et al., 2018; Lee et al., 2018; Mao et al., 2019) and uni-modal deterministic translations (Choi et al., 2018; Liu et al., 2017; Pumarola et al., 2018), while recent models focus on MMUIT tasks. In the latter category, DRIT++ (Lee et al., 2020) separately models the domain-independent ("content") and the domain-specific ("style") image representations using a content encoder and a style encoder, while multi-modal translations are obtained by injecting random noise. DMIT (Yu et al., 2019) adds a domain-specific representation to the content and the style representations of DRIT++. StarGAN v2 (Choi et al., 2020), the current state-of-the-art method, can generate high-resolution and diverse images using a multi-domain discriminator, a style encoder and a noise-to-style mapping network (see §3).

Note that in I2I translation "unsupervised" means that images are not paired during training. However, they are tagged with domain labels. Baek et al. (2020) propose a "Truly UNsupervised" Image Translation (TUNIT) setting, where pseudo-labels are first mined through a clustering procedure and then used for MMUIT tasks.

Latent-space interpolations. Image interpolations in generative models are obtained using three main strategies. First, by interpolating the latent-space representations of two different images in VAEs and GANs. For example, in PGGAN (Karras et al., 2018) and StyleGAN (Karras et al., 2019; 2020), it is possible to interpolate two latent codes and generate smooth transitions (Abdal et al., 2019; Shen et al., 2020; Richardson et al., 2020; Zhu et al., 2020; Abdal et al., 2020). However, these networks are not designed to translate images in multiple domains. Moreover, linearly travelling a normally distributed VAE latent space can lead to sub-optimal results (Arvanitidis et al., 2018).

The second strategy is based on learning an interpolation function. In HomoGAN, Chen et al. (2019) train an interpolation network which interpolates two latent codes, at the expense of limited diversity. Shen et al. (2020) identify and exploit the emerging semantics in *pretrained* generative models (Karras et al., 2018; 2019) to linearly traverse the latent space without retraining the networks.

Finally, interpolations can be done using I2I translations networks (as we do in this paper). However, previous MMUIT works either focus on only interpolations within a domain (Huang et al., 2018; Lee et al., 2018), or they show only qualitative results (Lee et al., 2020; Choi et al., 2020). In contrast,

in this paper we show that our MMUIT style-space regularization method can generate realistic and smooth inter-domain interpolations, which we quantitatively analyze using both standard MMUIT evaluation protocols and our proposed P^2 metric.

Mixup-based regularization. Mixup (Zhang et al., 2018a) is a simple yet effective data-augmentation strategy, which is based on blending two input images at the pixel level, and, consequently, "blending" also the corresponding image labels. Verma et al. (2019) extend this idea by mixing the representations in the intermediate layers of the network. Importantly, they show that Mixup acts as a latent space regularizer, because it encourages the network to behave linearly between pairs of data points to create smoother class decision boundaries. This idea has been used in many discriminative networks (Yun et al., 2019; Zhang et al., 2018a; Sohn et al., 2020) and, recently, Beckham et al. (2019) proposed *adversarial Mixup* to regularize the latent space of an unsupervised auto-encoder. Our Mixup formulation is inspired by Beckham et al. (2019), which we extend to an MMUIT scenario and of which we propose a multi-domain adaptation. Note that Beckham et al. (2019) also propose a supervised version of their adversarial Mixup, which, differently from our proposal, is based on a more complex mixing strategy of the labels, obtained using an ad hoc label embedding function. Moreover, we do not use non-linear mixing strategies of the samples (e.g., by means of genetic algorithms) as our goal is to force the information organization in the semantic space to be as appropriate as possible under linear interpolations of its elements.

3 The Generative Framework

In MMUIT, the training set ($\boldsymbol{\mathcal{X}}$) of real images is supposed to be composed of m disjoint domains ($\boldsymbol{\mathcal{X}} = \bigcup_{k=1}^{m} \boldsymbol{\mathcal{X}}_k$, $\boldsymbol{\mathcal{X}}_i \cap \boldsymbol{\mathcal{X}}_j = \emptyset, i \neq j$), where each domain $\boldsymbol{\mathcal{X}}_k$ contains images with the same style. In "truly unsupervised" I2I translation (TUNIT) (Baek et al., 2020), the domain partition is not given but obtained using a clustering method to mine a domain (pseudo-)label for each training image. Thus, without loss of generality, we assume that each image $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ is associated with a label (or a pseudo-label) $y \in \mathcal{Y}$ denoting its domain (i.e., $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_y$).

Our regularization approach (§4) can be applied to both MMUIT and TUNIT scenarios. In this section, we show the framework for MMUITs, which is mainly inspired by StarGAN v2 (Choi et al., 2020), while in Appendix A we show the differences to apply the framework to TUNIT.

Following Choi et al. (2020), the style space $\boldsymbol{\mathcal{S}}$ is *explicitly* modeled through an encoder E and a noise-to-style mapping network F. $\boldsymbol{\mathcal{S}}$ follows the same partition of $\boldsymbol{\mathcal{X}}$: $\boldsymbol{\mathcal{S}} = \bigcup_{k=1}^{m} \boldsymbol{\mathcal{S}}_k$. The role of E is to extract the style code from an image: $\boldsymbol{s} = E(\boldsymbol{x})$ ($\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$). On the other hand, F (an MLP) is used to inject diversity (appearance "multi-modality) in the generation process by conditioning with respect to random input noise. We sample a random vector ($\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$) and we use F to transform $\boldsymbol{z}$ into a style code: $\boldsymbol{s} = F(\boldsymbol{z})$. The generator ($G$) translates a source image $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}}_i$ in a target domain $\boldsymbol{\mathcal{X}}_j$: $\hat{\boldsymbol{x}} = G(\boldsymbol{x}_i, \boldsymbol{s}_j)$, where $\boldsymbol{s}_j \in \boldsymbol{\mathcal{S}}_j$ represents the target style which may be either extracted from a reference image (e.g., $\boldsymbol{s}_j = E(\boldsymbol{x}_j)$) or randomly sampled (e.g., $\boldsymbol{s}_j = F(\boldsymbol{z})$). This generative framework is trained using different losses, which are briefly described below.

The *style reconstruction* loss (Huang et al., 2018; Zhu et al., 2017b; Choi et al., 2020) pushes the target style code and the code extracted from the generated image to be as close as possible:

$$\mathcal{L}_{sty} = \mathbb{E}_{\boldsymbol{x}_i \sim \boldsymbol{\mathcal{X}}_i, \boldsymbol{s}_j \sim \boldsymbol{\mathcal{S}}_j} \left[\| \boldsymbol{s}_j - E(G(\boldsymbol{x}_i, \boldsymbol{s}_j)) \|_1 \right]. \tag{1}$$

The *diversity sensitive* loss (Choi et al., 2020; Mao et al., 2019) is used to generate diverse images when G is conditioned on different styles in a same domain:

$$\mathcal{L}_{ds} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}_i, \boldsymbol{s}_1, \boldsymbol{s}_2 \sim \boldsymbol{\mathcal{S}}_j} \left[\| G(\boldsymbol{x}, \boldsymbol{s}_1) - G(\boldsymbol{x}, \boldsymbol{s}_2) \|_1 \right]. \tag{2}$$

The *cycle consistency* loss (Zhu et al., 2017a; Choi et al., 2018; 2020) is used to preserve the content of the source image $\boldsymbol{x}$:

$$\mathcal{L}_{cyc} = \mathbb{E}_{\boldsymbol{x}_i \sim \boldsymbol{\mathcal{X}}_i, \boldsymbol{s}_j \sim \boldsymbol{\mathcal{S}}_j} \left[\| \boldsymbol{x}_i - G(G(\boldsymbol{x}_i, \boldsymbol{s}_j), E(\boldsymbol{x}_i)) \|_1 \right] \tag{3}$$

Note that Eqs. (2) and (3) work in the pixel space while Eq. (1) is evaluated in the latent style space.

Finally, we adopt the multi-domain discriminator architecture proposed in StarGAN (Choi et al., 2018). While the discriminator used in Choi et al. (2020) requires multiple real/fake binary classification branches, the discriminator of Choi et al. (2018) is composed of only two branches, one

($D_{\text{r/f}}$) discriminates between real and fake images, and the other branch (D_{cls}) estimates a posterior probability over $\mathcal{Y}$ and classifies the domains. The reason behind this choice will be clarified in §4. The *adversarial* loss is then:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}}[\log D_{\text{r/f}}(\boldsymbol{x})\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}_i,\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}[\log(1 - D_{\text{r/f}}(G(\boldsymbol{x},\boldsymbol{s})))], \tag{4}$$

while the *domain classification* loss (Choi et al., 2018) is the cross-entropy loss, which, for the discriminator D and the generator G, can be formulated as:

$$\mathcal{L}^{D}_{\text{cls}} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}_i}[\log D_{\text{cls}}(y = i|\boldsymbol{x})], \tag{5}$$

$$\mathcal{L}^{G}_{\text{cls}} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}_i,\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}_j}[\log D_{\text{cls}}(y = j|G(\boldsymbol{x},\boldsymbol{s}))]. \tag{6}$$

We refer the reader to (Choi et al., 2018; 2020) and to Appendix B.1 for additional details.

4 Regularizing the Latent Style Space

In this section, we introduce our regularization approach, which is based on the *shrinkage loss* and on a Mixup-based sample generation. Specifically, as mentioned in §1, the goal of the proposed **shrinkage loss** is to compact the latent space in order to reduce the regions with a low true probability density. This is obtained by:

$$\mathcal{L}_{\text{shr}} = \mathbb{E}_{\boldsymbol{s}_1,\boldsymbol{s}_2\sim\boldsymbol{\mathcal{S}}}[||\boldsymbol{s}_1 - \boldsymbol{s}_2||_2^2]. \tag{7}$$

For each pair of points $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ in the latent space, Eq. (7) penalizes their squared Euclidean distance, in this way fighting against the tendency of G and D to increase the style-space support. In Eq. (7), the pair $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ is drawn from $\boldsymbol{\mathcal{S}}$ using a mixed strategy, including both style codes extracted from real images, and randomly generated codes. More in detail, with probability 0.5, we use two (randomly chosen) real samples $\boldsymbol{x}_1 \in \boldsymbol{\mathcal{X}}_i, \boldsymbol{x}_2 \in \boldsymbol{\mathcal{X}}_j$, and we extract the corresponding style codes: $\boldsymbol{s}_1 = E(\boldsymbol{x}_1)$, $\boldsymbol{s}_2 = E(\boldsymbol{x}_2)$. Moreover, with probability 0.5, we use $\boldsymbol{z}_1, \boldsymbol{z}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{s}_1 = F(\boldsymbol{z}_1)$, $\boldsymbol{s}_2 = F(\boldsymbol{z}_2)$. In practice, we alternate mini-batch iterations in which we use only real samples with iterations in which we use only generated samples. Note that, in both cases, we may have that both $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ belong to the same domain: Eq. (7) is applied to *all* the pairwise distances in $\boldsymbol{\mathcal{S}}$ (intra- and inter-domain). The gradient of $\mathcal{L}_{\text{shr}}$ is directly backpropagated through E and F. However, since G directly depends on $\boldsymbol{\mathcal{S}}$, and D indirectly depends on G, the effect of $\mathcal{L}_{\text{shr}}$ propagates to the whole generative framework.

In the **Mixup-based regularization**, inspired by Beckham et al. (2019), we use latent-space interpolations to generate "fake" samples to fool the discriminator of the adversarial loss (Eq. (4)). Moreover, we extend the domain classification loss (Eq. (6)) to classify mixed samples. Let:

$$\text{Mix}(\boldsymbol{s}_1, \boldsymbol{s}_2, \alpha) = (1 - \alpha)\boldsymbol{s}_1 + \alpha\boldsymbol{s}_2, \tag{8}$$

where $\boldsymbol{s}_1, \boldsymbol{s}_2 \sim \boldsymbol{\mathcal{S}}$ and α ($\alpha \in [0, 1]$), as suggested in Verma et al. (2019), is drawn from a Beta distribution: $\alpha \sim \texttt{Beta}(b, b)$. We use $b = 2$, which corresponds to a "hill" shape, with most of the mass in the center of the interpolation line. Thereby, most of the mixed samples are generated far from the real data points, i.e., in those areas of $\boldsymbol{\mathcal{S}}$ corresponding to a low-density of the true data distribution. The *adversarial mixup* loss is:

$$\mathcal{L}^{\text{mix}}_{\text{adv}} = \mathbb{E}_{\boldsymbol{x}_i\sim\boldsymbol{\mathcal{X}}_i,\boldsymbol{s}_j\sim\boldsymbol{\mathcal{S}},\alpha\sim\texttt{Beta}(b,b)}[\log(1 - D_{\text{r/f}}(G(\boldsymbol{x}_i, \boldsymbol{s}_{\text{mix}}))], \tag{9}$$

where: $\boldsymbol{s}_{\text{mix}} = \text{Mix}(\boldsymbol{s}_i, \boldsymbol{s}_j, \alpha)$ and $\boldsymbol{s}_i = E(\boldsymbol{x}_i)$. When sampling $\boldsymbol{s}_j$ ($\boldsymbol{s}_j \sim \boldsymbol{\mathcal{S}}$), similarly to Eq. (7), we use a mixed strategy, alternating: (1) $\boldsymbol{s}_j = F(\boldsymbol{z}), \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ with (2) $\boldsymbol{s}_j = E(\boldsymbol{x}_j)$, where $\boldsymbol{x}_j$ is a real image different from $\boldsymbol{x}_i$ and randomly sampled from the whole training set ($\boldsymbol{x}_j \sim \boldsymbol{\mathcal{X}}$). Note that we may have $\boldsymbol{s}_i, \boldsymbol{s}_j \in \boldsymbol{\mathcal{S}}_i$. In Eq. (9), G generates an image using the mixed style code ($G(\boldsymbol{x}_i, \boldsymbol{s}_{\text{mix}})$) that is used to "fool" $D_{\text{r/f}}$. Intuitively, this helps to disentangle $\boldsymbol{\mathcal{S}}$, because unrealistic images, lying in between $G(\boldsymbol{x}_i, \boldsymbol{s}_i)$ and $G(\boldsymbol{x}_i, \boldsymbol{s}_j)$, are "moved away" from the interpolation segment whose endpoints are $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

Analogously to Eq. (9), we extend Eq. (6) using our *domain-mixup classification* loss:

$$\mathcal{L}^{\text{mix}}_{\text{cls}} = \mathbb{E}_{\substack{\boldsymbol{x}_i\sim\boldsymbol{\mathcal{X}}_i,\boldsymbol{s}_j\sim\boldsymbol{\mathcal{S}},i\neq j,\\ \alpha\sim\texttt{Beta}(b,b)}}[(1 - \alpha)\log D_{\text{cls}}(y = i|G(\boldsymbol{x}_i, \boldsymbol{s}_{\text{mix}}) + \alpha\log D_{cls}(y = j|G(\boldsymbol{x}_i, \boldsymbol{s}_{\text{mix}}))], \tag{10}$$

where, similarly to Eq. (9), $\boldsymbol{s}_{\text{mix}} = \text{Mix}(E(\boldsymbol{x}_i), \boldsymbol{s}_j, \alpha)$ and either $\boldsymbol{s}_j = F(\boldsymbol{z}), \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, or $\boldsymbol{s}_j = E(\boldsymbol{x}_j)$. The constraint $i \neq j$ is used because we want to interpolate between samples of different domains (when $i = j$, then Eq. (10) corresponds to computing Eq. (6) twice). In Eq. (10), we use the binary cross-entropy and the mixing coefficient α is interpreted as a probability value. Specifically, we want that an image $\boldsymbol{x}_i$, when transformed using the mixed style code $\boldsymbol{s}_{\text{mix}}$ ($G(\boldsymbol{x}_i, \boldsymbol{s}_{\text{mix}})$), should belong to domain $\boldsymbol{\mathcal{X}}_i$ with probability $(1 - \alpha)$ and to domain $\boldsymbol{\mathcal{X}}_j$ with probability α.

Finally, as mentioned in §3, the choice of the StarGAN-like multi-task discriminator (Choi et al., 2018) is related to the posterior probability over $\mathcal{Y}$ computed by D_{cls} and used in Eq. (10). Although Eq. (10) may be adapted to the multiple independent real/fake binary classification branches of StarGAN v2 (Choi et al., 2020), the above formulation is more natural.

5 Evaluation Protocols

Quality and diversity. We evaluate both the visual quality and the diversity of the generated images through the Fréchet Inception Distance (FID) (Heusel et al., 2017) and the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018b), respectively. FID is computed between domains using all images, including the interpolation images. LPIPS is computed between every pair of images in each domain. We average the results across the dataset. More details in Appendix D.

Semantic smoothness. Karras et al. (2019) recently proposed the Perceptual Path Length (PPL) to measure the smoothness of a semantic space. This metric is based on computing the perceptual variation between pairs of generated images under small perturbations (ϵ) in the latent space. The perceptual variation is estimated using an externally trained network, the same used in LPIPS. However, there are different problems with PPL. First, the value of ϵ should be manually estimated depending on the scale of the latent space, and PPL decreases quadratically with respect to ϵ. Then, PPL can be minimized by a "collapsed" generator with no diversity (e.g., constantly generating the same image, independently of the input style code). Alternative formulations, such as computing the standard deviation of perceptual distances over the interpolation line also suffer from similar problems (e.g., perceptual distances between adjacent interpolation points may be highly non-normally distributed).

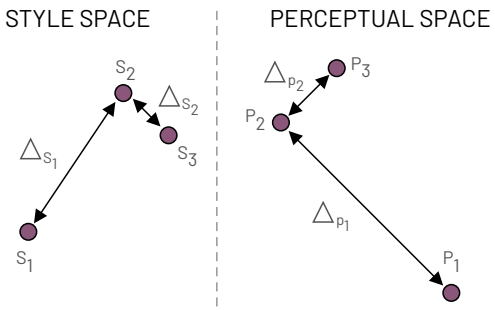

Figure 2: A schematic illustration of the P^2 metric.

To solve these issues, we propose a new smoothness metric called Perceptual Proportionality (P^2), whose intuitive idea is shown in Fig. 2. The right part of Fig. 2 shows a "perceptual" space ($\boldsymbol{\mathcal{P}}$), which in practice is the representation space of an externally pretrained network (ϕ). Specifically, we use the same network used by both the LPIPS and the PPL metric to compute their perceptual distances, which have been shown to be well aligned with the human perceptual similarity (Zhang et al., 2018b). On the left of the same figure, we have the style space ($\boldsymbol{\mathcal{S}}$) of the MMUIT framework we want to evaluate. Given 3 points on $\boldsymbol{\mathcal{S}}$ ($\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$), we can generate 3 corresponding images which are projected onto $\boldsymbol{\mathcal{P}}$ using ϕ ($\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$). While the absolute distances between these points in the two spaces are different, in an ideal situation, we would like to have the same ratio of their distances. For instance, assuming that: $\Delta_{s1} = ||\boldsymbol{s}_1 - \boldsymbol{s}_2||$, $\Delta_{s2} = ||\boldsymbol{s}_2 - \boldsymbol{s}_3||$ and $\Delta_{p1} = ||\boldsymbol{p}_1 - \boldsymbol{p}_2||$, $\Delta_{p2} = ||\boldsymbol{p}_2 - \boldsymbol{p}_3||$, then, ideally, we should have:

$$\Delta_{p1}/\Delta_{p2} = \Delta_{s1}/\Delta_{s2}. \tag{11}$$

Note that the ratios in Eq. (11) are unitless, so they can be compared to each other. In practice, however, the information organization in $\boldsymbol{\mathcal{P}}$ and in $\boldsymbol{\mathcal{S}}$ will not be exactly the same. Thus, our metric is based on averaging the total errors in Eq. (11) computed over a set of triplets of points. In more detail, we sample 3 points ($\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$) in $\boldsymbol{\mathcal{S}}$ and a source image $\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}$. Then we "translate" $\boldsymbol{x}$ using $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$, and we project the generated images onto $\boldsymbol{\mathcal{P}}$, obtaining: $\boldsymbol{p}_i = \phi(G(\boldsymbol{x}, \boldsymbol{s}_i))$ ($i \in \{1, 2, 3\}$). We compute Δ_{pj} and Δ_{sj} ($j \in \{1, 2\}$) as above, and finally we have:

$$P^2 = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3 \sim \boldsymbol{\mathcal{S}}} [| \frac{\Delta_{p1}}{\Delta_{p2} + \epsilon} - \frac{\Delta_{s1}}{\Delta_{s2} + \epsilon} |], \tag{12}$$

where ϵ is used for numerical stability. The lower the value of P^2, the more linear is $\boldsymbol{\mathcal{S}}$ with respect to $\boldsymbol{\mathcal{P}}$. P^2 has several advantages: (1) it is simple and relatively fast to compute; (2) it is parameter-free; (3) a "mono-modal" generator that generates always the same image or with a low diversity of outputs, results in a high value of P^2 (if $\boldsymbol{p}_i \sim \boldsymbol{p}_j$, $i \neq j$, then $\Delta_{p1}/\Delta_{p2} \sim 1$, while $\Delta_{s1}/\Delta_{s2} \neq 1$).

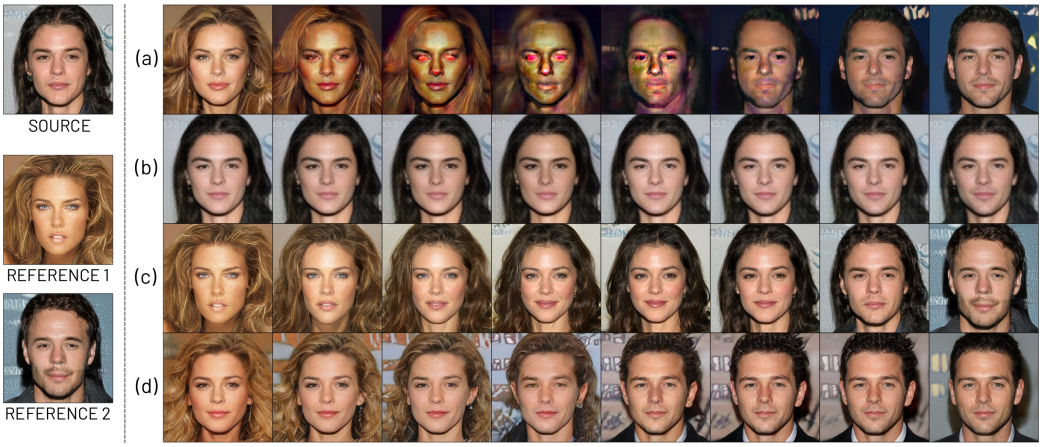

Figure 3: Inter-domain interpolations between genders using CelebA-HQ: (a) StarGAN v2, (b) HomoGAN, (c) InterFaceGAN, (d) our method. All the models use the same source and reference images. Our model generates smoother results while better preserving the source-person identity.

6 Experiments

Baselines. We compare our method with state-of-the-art MMUIT, TUNIT and interpolation function learning approaches (see §2). As a representative of the above categories, we use StarGAN v2 (Choi et al., 2020), TUNIT (Baek et al., 2020) and HomoGAN (Chen et al., 2019), respectively. Moreover, in the CelebA-HQ experiments, we use also InterFaceGAN (Shen et al., 2020) as a reference for a high-quality generation. Despite this model is not specifically designed for MMUIT and it is based on the *very training-intensive* model StyleGAN (Karras et al., 2018; 2019), it performs high-resolution linear interpolations images. All the models are tested using the official source codes.

Datasets and settings. Following the settings used in StarGAN v2 (Choi et al., 2020), we test our method with high-quality images of human and animal faces through CelebA-HQ (Karras et al., 2018) and AFHQ (Choi et al., 2020), respectively. We use CelebA-HQ with the gender (*male* and *female*) and the smile (*no smile*, *smile*) domains, while, in AFHQ, we use the *cat*, the *dog* and the *wildlife* domains. We do not use any additional information but the domain labels for the MMUIT setting, while no label is used in the TUNIT setting. All the images have a 256×256 resolution. For a fair comparison, we use the same training and testing images for all the models in each setting.

6.1 Ablation Study

In this section, we evaluate the impact of all the components of our method using FID, LPIPS and our proposed metric P^2. For completeness, we also show the PPL scores (§5). The results are shown in Tab. 1, where we separately analyse the contribution of the two proposed regularization methods, the shrinkage loss $\mathcal{L}_{\text{shr}}$ (Eq. (7)) and the sum of the two Mixup-based losses (Eq. (9) and Eq. (10)), cumulatively called $\mathcal{L}_{\text{mix}}$ for brevity. As the starting baseline we use StarGAN v2 (Choi et al., 2020), because our losses are added to this method using exactly its network architectural details and basic training losses (§3). Note that, as mentioned in §3, we use a differently branched discriminator with respect to StarGAN v2 which, according to Choi et al. (2020), leads to a slightly worse average performance.

Table 1: An ablation study of our regularization losses using CelebA-HQ with gender translations.

	Model	FID↓	LPIPS↑	PPL↓	$\mathbf{P^2}$↓
A:	StarGAN v2	42.32	.443	59.25	.213
B:	A + $\mathcal{L}_{\text{shr}}$	34.44	.448	**27.93**	.277
C:	A + $\mathcal{L}_{\text{mix}}$	26.44	.245	32.95	**.173**
D:	A + $\mathcal{L}_{\text{mix}}$ + $\mathcal{L}_{\text{shr}}$	**23.03**	**.511**	37.80	.181

Tab. 1 (B) shows a relative improvement in all the metrics except P^2 with respect to the base model, confirming the importance of a compact semantic space to improve the image quality and the diversity of the I2I translations. Comparing Tab. 1 (B) with Tab. 1 (C), we observe that $\mathcal{L}_{\text{mix}}$ obtains an even higher improvement on the image quality (FID: -37.52%) and the smoothness degree (P^2: -18.78%) with respect to StarGAN v2, at the expense, however, of diversity (LPIPS). Finally, Tab. 1 (D) shows that the combination of mixup and the shrinkage loss drastically improves both FID and LPIPS with respect to both the ablated methods. However, the latent-space smoothness degree of the full model is not the best over the tested combinations (e.g., it underperforms Tab. 1 (C) when measured with both P^2 and PPL). We speculate this result might be a consequence of a trade-off in MMUIT models between diversity and smoothness. The higher the diversity of the translations, the more challenging is to keep gradual the changes between neighbouring points in the latent space.

6.2 Comparison with the state of the art

Qualitative comparison. We first compare our method with state-of-the-art approaches on CelebA-HQ. As shown in Fig. 3, the images are obtained by linearly interpolating the style codes between $\boldsymbol{s}_1 = E(\boldsymbol{x}_1)$ and $\boldsymbol{s}_T = E(\boldsymbol{x}_2)$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two *reference* images belonging to two different domains. The intermediate style codes ($\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_T\}$) are used to transform a common *source* image $\boldsymbol{x}$, leading to a set of images $\{G(\boldsymbol{x}, \boldsymbol{s}_1), \dots, G(\boldsymbol{x}, \boldsymbol{s}_T)\}$ for each compared generation method, which are shown in the corresponding rows of Fig. 3. Specifically, Fig. 3 (a) shows that StarGAN v2 generates artifacts and unrealistic results, especially in the center of the interpolation line between the two domains. On the other hand, the interpolation results of HomoGAN are very smooth, since neighbouring images are almost indistinguishable the one from the other (Fig. 3 (b)). However, the HomoGAN translations endpoints ($G(\boldsymbol{x}, \boldsymbol{s}_1)$ and $G(\boldsymbol{x}, \boldsymbol{s}_T)$) change very little the one from the other, calling into question whether the model can do image-to-image translations. Conversely, our method (Fig. 3 (d)), successfully translates the source image into the target domains and the intermediate translation results are both highly realistic and gradually changing. The quality of our results is comparable with the reference model InterFaceGAN (Fig. 3 (c)), which is based on the computationally very intensive training of StyleGAN with high-resolution images (Karras et al., 2019; 2020; 2018). Fig. 4 we show an example where we perform inter-domain interpolations between multiple domains at the same time. We show additional qualitative results in Appendix E.

Fig. 5 shows the results on AFHQ. We observe that our model interpolates animal images very smoothly, generating inter-species animals. Conversely, StarGAN v2 interpolations contain abrupt changes, artifacts and unrealistic results, similarly to those generated with CelebA-HQ images. Note that we cannot use InterFaceGAN on this dataset due of the lack of a publicly available pretrained StyleGAN model on AFHQ. In the Appendix E we show additional comparative results on this dataset.

Quantitative comparison. In Tab. 2, we use the CelebA-HQ dataset and we quantitatively compare our method with the other approaches with respect to the image quality (FID) and diversity (LPIPS). As expected, InterFaceGAN achieves the best FID, but with a very low diversity degree (LPIPS scores). In fact, style and content are not disentangled in the StyleGAN latent space, and this prevents the use of a noise-to-style mapping network (similar to our F, see §3) to inject style-specific diversity in the image translations. The quantitative results of HomoGAN confirm its qualitative evaluation, with a diversity degree even lower than InterFaceGAN. StarGAN v2 clearly outperforms HomoGAN, and our method largely outperforms StarGAN v2 with respect to all the metrics.

Figure 4: Inter-domain interpolations between multiple domains (gender and expression).

We also quantitatively measure the latent-space smoothness using PPL and our proposed P^2 (Tab. 2). In Tab. 2, HomoGAN gets the best PPL, which is significantly lower than all other models. However, as previously seen in the qualitative results, this model generates images with very little changes along the interpolation lines. Interestingly, our proposed P^2 metrics is more aligned with the qualitative results, since assigns to HomoGAN the lowest-ranking value over the three compared meth-

Figure 5: Inter-domain interpolations between StarGAN v2 and our model in on AFHQ.

Table 2: Image quality (FID) and translation diversity (LPIPS) measured on the CelebA-HQ dataset. §Reference: StyleGAN-based model with 1024×1024 images.†: always generates the same image.

Model	**FID↓**		**LPIPS↑**		**PPL↓**		**P^2↓**	
	Gender	Smile	Gender	Smile	Gender	Smile	Gender	Smile
HomoGAN (Chen et al., 2019)	55.23	58.02	.001	< .001	**5.42**†	**1.17**†	.250	.220
StarGAN v2 (Choi et al., 2020)	42.32	28.16	.443	.413	59.25	40.79	.213	.178
Ours	**23.03**	**22.62**	**.511**	**.480**	37.80	35.04	**.181**	**.167**
InterFaceGAN (Shen et al., 2020)§	13.75	12.81	.067	.027	51.73	24.24	.157	.123

ods (see §5). Compared to StarGAN v2, our approach drastically improves both the PPL and the P^2 scores, quantitatively showing that our regularization methods can smooth the semantic-space representations. We note that InterfaceGAN gets better P^2 than our model. However, StyleGAN (the model on which InterfaceGAN is based) is massively trained to disentangle the factors of variation of its semantic space (Karras et al., 2019; 2020). In *Appendix C.2* we show the evaluation of P^2.

Tab. 3 shows the quantitative results for the more challenging AFHQ dataset, where there is a more significant inter-domain difference than in CelebA-HQ. Our method outperforms with a significant margin all the other tested approaches in all the settings and with all the metrics, except Baek et al. (2020) with respect to the LPIPS metric. Due to the lack of space, we show the qualitative results of the TUNIT setting in Appendix E. In the AFHQ dataset, we do not include HomoGAN because that model requires well-aligned training images having the same orientation (Hom, 2021) and this makes it hard to train HomoGAN on the animal face images of AFHQ.

Table 3: Quantitative evaluation on AFHQ.

Model	**Setting**	**FID↓**	**LPIPS↑**	**PPL↓**	**P^2↓**
StarGAN v2	MMUIT	15.64	.435	79.62	.226
Ours	MMUIT	**11.56**	**.454**	**19.49**	**.211**
Baek et al. (2020)	TUNIT	19.67	**.442**	22.80	.173
Ours	TUNIT	**17.23**	.307	**17.94**	**.148**

Overall, the quantitative and the qualitative analysis show that our regularization method drastically improves the state-of-the-art multi-domain translations in both the MMUIT and TUNIT settings.

7 Conclusion

In this paper, we presented a regularization approach for MMUIT networks which is based on the hypothesis that the true data distribution in the inter-domain regions of the representation space is not well modeled because of the inherent scarcity of inter-domain training data. To solve this problem, we propose two simple, yet very effective regularization approaches, respectively based on the shrinkage loss (which compacts the latent space) and on a Mixup data augmentation strategy (which populates the regions across two domains). Moreover, we propose a new metric to explicitly evaluate the semantic smoothness of a style space.

Using both our P^2 metric and common MMUIT evaluation protocols, we showed that the proposed regularization losses can be plugged in existing MMUIT frameworks, leading to a significant quality improvement of the results in all the tested MMUIT settings.

REFERENCES

HomoGAN: issue on unaligned images. https://github.com/yingcong/HomoInterpGAN/issues/3, 2021. Accessed: 2021-03-15.

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

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

Adobe. Taking It to the MAX: Adobe Photoshop Gets New NVIDIA AI-Powered Neural Filters. https://blogs.nvidia.com/blog/2020/10/20/adobe-max-ai/, 2021. Accessed: 2021-03-15.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *ICLR*, 2018.

Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. *arXiv preprint arXiv:2006.06500*, 2020.

Christopher Beckham, Sina Honari, Vikas Verma, Alex M Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, and Chris Pal. On adversarial mixup resynthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *NeurIPS*, volume 32. Curran Associates, Inc., 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *NeurIPS*. Curran Associates, Inc., 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *CVPR*, 2019.

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 *CVPR*, 2018.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *NeurIPS*, 2017.

Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *NeurIPS*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

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 *NeurIPS*, 2017.

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

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

Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

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

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

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

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. *IJCV*, 2020. ISSN 1573-1405. doi: 10.1007/s11263-019-01284-z. URL https://doi.org/10.1007/s11263-019-01284-z.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017.

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*, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

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

Qi Mao, Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, Siwei Ma, and Ming-Hsuan Yang. Continuous and diverse image-to-image translation via signed attribute vectors. *International Journal of Computer Vision*, 130(2):517–549, 2022.

Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 2018.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020.

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

Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable GANs for pose-based human image generation. In *CVPR*, 2018.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jérémie Mary. Learning disconnected manifolds: a no gan's land. In *ICML*, 2020.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

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

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: beyond empirical risk minimization. In *ICLR*, 2018a.

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

Jiapeng Zhu, Deli Zhao, Bolei Zhou, and Bo Zhang. Lia: Latently invertible autoencoder with adversarial learning. 2019.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017b.

A Our framework in the TUNIT Setting

In this section, we describe the generative framework we adopted for the TUNIT setting, which is based on the method presented in Baek et al. (2020). Similarly to the MMUIT setting described in the main paper, in which we modify StarGAN v2 adding our losses to the StarGAN v2 native losses and changing the discriminator, also for the TUNIT setting we modify the approach proposed in Baek et al. (2020) by:

1. adding our losses to the losses used in Baek et al. (2020) and
2. replacing the discriminator of Baek et al. (2020) with our discriminator (the latter being described in Sec. 3 of the main paper and in more detail in Sec. B.2).

For completeness, we briefly describe below the approach proposed in Baek et al. (2020), emphasizing that this is not our contribution. We believe that the main interest in describing the details of the method proposed by Baek et al. (2020) is that their losses are drastically different from those used in StarGAN v2 (e.g., see below the Mutual Information maximization or the contrastive loss). Despite that, as shown in Sec. 6.2 of the main paper and in Sec. E, our regularization methods can successfully be used jointly with the losses and the architecture proposed in Baek et al. (2020), showing the generality of our regularization proposal.

A.1 THE ADOPTED TUNIT FRAMEWORK

The architecture proposed by Baek et al. (2020) is composed of an encoder network E, which has two branches for pseudo-label classification E_C and style extraction E_S, a generator G and a multi-task discriminator D, which has as many output branches as the number of domains m. Since in the TUNIT setting the domain partition is not available, the model jointly learns to cluster the real images and to translate them into different domains.

Computing the pseudo-labels. Baek et al. (2020) use IIC (Ji et al., 2019) to cluster the real images in multiple domains and extract the corresponding pseudo-labels. The main idea in IIC is that two augmented versions of the same image (e.g. obtained using horizontal flipping) should be similarly classified. For this reason, they define the joint probability matrix $\mathbf{P} \in \mathbb{R}^{m \times m}$:

$$\mathbf{P} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}}[E_C(\boldsymbol{x}) \cdot E_C(f(\boldsymbol{x}))^T], \tag{13}$$

where f is the data augmentation function. Then, they maximize the Mutual Information (MI) computed as:

$$\mathcal{L}_{MI} = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{P}_{ij} \ln \frac{\mathbf{P}_{ij}}{\mathbf{P}_i \mathbf{P}_j}, \tag{14}$$

where $\mathbf{P}_i$ denotes the m-dimensional marginal probability vector, and $\mathbf{P}_{ij}$ denotes the joint probability of domain i *and* domain j. For more details, we refer to Ji et al. (2019); Baek et al. (2020).

To further help domain classification, Baek et al. (2020) use also a contrastive loss (Hadsell et al., 2006):

$$\mathcal{L}^E_{style} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}} \left[-\log \frac{\exp(E_S(\boldsymbol{x}) \cdot E_S(f(\boldsymbol{x}))/\tau)}{\sum_{i=0}^{N} \exp(E_S(\boldsymbol{x}) \cdot E_S(\boldsymbol{x}_i^-)/\tau)} \right], \tag{15}$$

where the sum in the denominator is over N *negative* samples $\boldsymbol{x}^-$ ($\boldsymbol{x}^- \neq \boldsymbol{x}$) contained in a queue Q (we refer to He et al. (2020) for more details).

Learning to translate images. Baek et al. (2020) force the target style code and the code extracted from the generated image to be as close as possible:

$$\begin{aligned} \mathcal{L}^G_{style} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}, \boldsymbol{s} \sim \boldsymbol{\mathcal{S}}}[\\ -\log \frac{\exp(E_S(G(\boldsymbol{x}, \boldsymbol{s})) \cdot \boldsymbol{s})}{\sum_{i=0}^{N} \exp(E_S(G(\boldsymbol{x}, \boldsymbol{s})) \cdot E_S(\boldsymbol{x}_i^-)/\tau)}], \end{aligned} \tag{16}$$

where $\boldsymbol{s} = E_S(\tilde{\boldsymbol{x}})$ is extracted from a randomly sampled reference image $\tilde{\boldsymbol{x}} \sim \boldsymbol{\mathcal{X}}$, and $\boldsymbol{x}_i^-$ denotes the negative samples as described in Eq. (15).

Then, the *image reconstruction loss* is used to reconstruct the source image. It is defined as:

$$\mathcal{L}_{rec} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}}[\|\boldsymbol{x} - G(\boldsymbol{x}, E_S(\boldsymbol{x}))\|_1]. \tag{17}$$

Finally, Baek et al. (2020) use an adversarial loss based on a multi-task discriminator which is similar to the StarGAN v2 discriminator (Choi et al., 2020) (see Sec. 3 of the main paper). Conversely, we adopt the multi-domain discriminator architecture proposed in StarGAN (Choi et al., 2018), analogously to what we used for the MMUIT setting (see Sec. 3 of the main paper).

As above mentioned, our TUNIT model differs from Baek et al. (2020) because of the discriminator and the addition of our regularization losses.

B IMPLEMENTATION DETAILS

B.1 THE OVERALL ARCHITECTURE

Fig. 6 shows the architecture of our framework in the MMUIT setting.

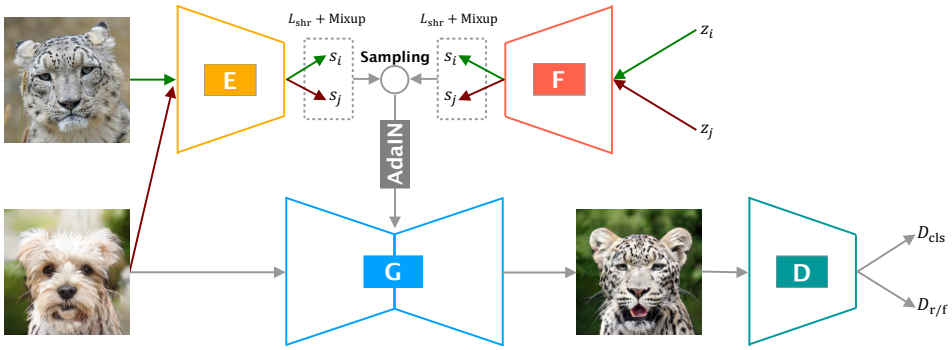

Figure 6: In the MMUIT setting, the generator G takes an image $\boldsymbol{x}$ and a style code $\boldsymbol{s}$ as input and generates an image that is fed to the discriminator D. D learns to classify the images into their own domain (D_{cls}) and to discriminate between real and fake images ($D_{\text{r/f}}$). The encoder E and the noise-to-style mapping network F are two instruments to get a specific style code $\boldsymbol{s}$. Our shrinkage loss $\mathcal{L}_{\text{shr}}$ and the mixup-based losses regularize the style space.

	Layer	Resample	Output shape
	Image $\boldsymbol{x}$	-	$256 \times 256 \times 3$
	Conv3×3	-	$256 \times 256 \times 64$
	ResBlk	AvgPool	$128 \times 128 \times 128$
	ResBlk	AvgPool	$64 \times 64 \times 256$
	ResBlk	AvgPool	$32 \times 32 \times 512$
	ResBlk	AvgPool	$16 \times 16 \times 512$
	ResBlk	AvgPool	$8 \times 8 \times 512$
	ResBlk	AvgPool	$4 \times 4 \times 512$
$D_{\text{r/f}}$	LReLU	-	$4 \times 4 \times 512$
	Conv4×4	-	$1 \times 1 \times 512$
	LReLU	-	$1 \times 1 \times 512$
	Conv1×1	-	$1 \times 1 \times 1$
D_{cls}	LReLU	-	$4 \times 4 \times 512$
	Conv4×4	-	$1 \times 1 \times 512$
	LReLU	-	$1 \times 1 \times 512$
	Conv1×1	-	$1 \times 1 \times m$

Table 4: The discriminator architecture. m is the number of domains.

B.2 The Discriminator

Table 4 shows the details of the discriminator we used in both the MMUIT and the TUNIT setting.

C P^2 Metric

C.1 Implementation Details

Perceptual distances. As mentioned in the main paper, the proposed P^2 metric is based on distances computed over the space $\boldsymbol{\mathcal{P}}$. In practice, we compute the involved perceptual distances using an externally pre-trained network (ϕ), the same network used by both the LPIPS and the PPL metric, which has been shown to be well aligned with the human perceptual similarity (Zhang et al., 2018b). However, although Zhang et al. (2018b) (who first proposed LPIPS) claim that their distance is a metric, their formulation is based on the squared Euclidean distance between the features of different layers of ϕ:

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} w_l \|\phi_l(\boldsymbol{x}_1(h,w)) - \phi_l(\boldsymbol{x}_2(h,w))\|_2^2, \tag{18}$$

where $\phi_l(\boldsymbol{x}(h,w))$ is the feature at position (h,w) in the convolutional feature map of layer l, and w_l is a learned layer-specific weight. Thus, Eq.(18) does not satisfy the triangle inequality, which is necessary for a distance to be a proper metric. For this reason, we use a slightly different formula:

$$d'(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} w_l \|\phi_l(\boldsymbol{x}_1(h,w)) - \phi_l(\boldsymbol{x}_2(h,w))\|_2. \tag{19}$$

In Zhang et al. (2018b), ϕ is an AlexNet (Krizhevsky et al., 2017) pre-trainted on ImageNet, while the weights $\{w_l\}$ are trained in order to mimic the human perceptual distance. Accordingly, we have re-trained the weights $\{w_l\}$ following the protocol and the dataset used in Zhang et al. (2018b) (which is *different* from the I2I translation datasets used in the main paper), but replacing Eq.equation 18 with Eq.equation 19.

Finally, Δ_{p1} in Eq. (12) of the main paper is computed using: $\Delta_{p1} = d'(\boldsymbol{x}_1, \boldsymbol{x}_2)$ (and similarly for Δ_{p2}).

Computing P^2 without an explicit style space $\mathcal{S}$. In InterFaceGAN (Shen et al., 2020), there is no separation between the "content" and the "style" representations, thus we cannot sample three arbitrary points $\boldsymbol{s}_i$, in $\boldsymbol{\mathcal{S}}$ and then generate $G(\boldsymbol{x}, \boldsymbol{s}_i)$ ($i \in \{1,2,3\}$) as in Sec. 5 of the main paper. For this reason, we approximate the sampling procedure as follows. We ask InterFaceGAN to interpolate between two reference latent codes using T equally spaced interpolation points. In this way we get a sequence of T generated images $I = (\hat{\boldsymbol{x}}_1, ..., \hat{\boldsymbol{x}}_T)$. Then we randomly choose $i, k \in \{1, ..., T\}$ and we select $\hat{\boldsymbol{x}}_i$, $\hat{\boldsymbol{x}}_{i+k}$ and $\hat{\boldsymbol{x}}_{i+2k}$ in I. In this way the three chosen images are selected using a constant step (k). As a consequence, $\Delta_{s1} = \Delta_{s2}$ and $\Delta_{s1}/\Delta_{s2} = 1$. Hence, Eq. (12) in the main paper can be rewritten as:

$$P^2 = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}, i, k \sim \{1,...,T\}}[|\frac{\Delta_{p1}}{\Delta_{p2} + \epsilon} - 1|], \tag{20}$$

where $\Delta_{p1} = d'(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_{i+k})$ and $\Delta_{p2} = d'(\hat{\boldsymbol{x}}_{i+k}, \hat{\boldsymbol{x}}_{i+2k})$. For a fair comparison, we adopt this procedure for all the tested methods (including ours).

C.2 Evaluation

To evaluate , we use the PPL evaluation protocol adopted in StyleGAN (Karras et al., 2020), we used P^2 to rank the of the per-image P^2 score interpolations. Fig. 7 shows the top most row shows the interpolation with the highest P^2 value, while the bottom row corresponds to the lowest score.

Figure 7: Random examples with low P^2 ($\leq$ 10th percentile) in the first row, while in the second row we show some examples with high P^2 ($\geq$ 90th percentile). There is a clear correlation between P^2 scores and the smoothness of interpolations.

D EVALUATION PROTOCOL

D.1 FID-COMPUTATION DETAILS

The FID scores are computed using the interpolation results as follows. For each $\boldsymbol{\mathcal{X}}_i \rightarrow \boldsymbol{\mathcal{X}}_j$ domain translation, we use 1,000 test source images. For each source image ($\boldsymbol{x}$), we separately randomly select two different reference images ($\boldsymbol{x}_1 \in \boldsymbol{\mathcal{X}}_i$ and $\boldsymbol{x}_2 \in \boldsymbol{\mathcal{X}}_j$), which are used to extract the start and the end style codes ($\boldsymbol{s}_1 = E(\boldsymbol{x}_1) \in \boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{s}_T = E(\boldsymbol{x}_2) \in \boldsymbol{\mathcal{S}}_j$). $\boldsymbol{s}_1$ and $\boldsymbol{s}_T$ are linearly interpolated ($\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_T\}$) and the intermediate points are used to generate the new images $\{G(\boldsymbol{x}, \boldsymbol{s}_1), \dots, G(\boldsymbol{x}, \boldsymbol{s}_T)\}$, with $T = 20$. The FID scores are computed by averaging over all the $T \times 1,000$ generated images. Concerning LPIPS, for each source image we sample 10 style codes in each target domain, we generate the corresponding images (without interpolations), and then we compute the LPIPS distances between every pair of images in the same domain, averaging the results across the dataset.

Since this evaluation method is based on interpolations, for fair comparison we also check the quality of images with FID computed only on some random points in the latent space, as done in Choi et al. (2018; 2020). On CelebA-HQ, (Gender translations), we have: StarGAN-v2 Choi et al. (2020), 23.9 and ours: 24.8 (StarGAN-v2 is slightly better than ours). On AFHQ, TUNIT (Baek et al., 2020), 17.13; ours, 16.65 (ours is slightly better than Baek et al. (2020)). These results show that, overall, our method does not reduce the image quality of the original translation task. Note that the LIPIPS scores reported in all the tables (of the main manuscript) were computed without interpolations, and they show that our method, in most of the cases, can significantly increase the diversity of the original translation task.

In the **CelebA-HQ dataset**, we compare also with InterFaceGAN (Shen et al., 2020), based on StyleGAN (Karras et al., 2019; 2020) and trained with high-resolution images. However, InterFaceGAN is not designed for MMUIT tasks, and does not have an image encoder. InterFaceGAN performs face editing by "moving" a latent code on the pretrained StyleGAN face representation space along a given direction (e.g. more smile - less smile). Thus, given a generated image $\boldsymbol{x} = G(\boldsymbol{z})$, where $\boldsymbol{z}$ is a StyleGAN latent code, InterFaceGAN edits $\boldsymbol{z}$ through:

$$\boldsymbol{z}' = \boldsymbol{z} + \alpha \boldsymbol{n},$$

where $\boldsymbol{n}$ is the unit normal vector defining a domain-separation hyperplane (e.g. smile vs non-smile) and α controls how much positive (or negative) the editing should be (e.g. more smile or less smile). We refer to Shen et al. (2020) for additional details.

For I2I translations with InterFaceGAN, we need to use an encoder from images to the StyleGAN face representation space (e.g. Richardson et al. (2020); Zhu et al. (2019)). However, the chosen encoder may influence the translation performance. To have a fair comparison between InterFaceGAN and MMUIT models in CelebA-HQ, we instead choose the two reference images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (see the main paper, sec. 6.2), *obtained using InterFaceGAN* as follows. Following Shen et al. (2020), for each StyleGAN generated image $\boldsymbol{x}$, we generate $\boldsymbol{x}_1 = G(\boldsymbol{z} + \alpha_1 \boldsymbol{n}))$ and $\boldsymbol{x}_2 = G(\boldsymbol{z} + \alpha_2 \boldsymbol{n})$ with: $\alpha_1 = -3$ (e.g. no smile) and $\alpha_2 = 3$ (e.g. big smile). These two reference images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are used for computing the interpolations as described in Sec. 6.2 of the main paper, and we emphasize that they are used for all the methods, including HomoGAN (Chen et al., 2019), StarGAN v2 (Choi et al., 2020) and ours. For the quantitative analysis, we repeat this process 1,000 times, using a different pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ at each iteration.

Note that this evaluation protocol does not use any image that is present in the training set of CelebA-HQ. Note also that the selection of the reference images using InterFaceGAN most likely helps to increase the InterFaceGAN performance being biased on the StyleGAN representation space.

In the **AFHQ dataset**, we do not compare with InterFaceGAN, being InterFaceGAN and StyleGAN not trained on AFHQ. For this reason, both in the MMUIT and the TUNIT settings, the two reference images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are simply randomly selected among the real images of the testing AFHQ split.

D.2 DATASETS

We follow the setting in Choi et al. (2020) when evaluating the performances on the CelebA-HQ (Karras et al., 2018) and the AFHQ dataset (Choi et al., 2020). CelebA-HQ is a High-Quality

version of the CelebA (Liu et al., 2015) dataset, consisting of 30,000 images with a 1024×1024 resolution. We use the training and the testing lists provided in Choi et al. (2020). Differently from Choi et al. (2020), we also use the smile attribute for testing. The AFHQ dataset consists of 15,000 high-quality images at 512×512 resolution. It includes three domains "cat", "dog", and "wildlife", each composed of 5,000 images. For each domain, we use the the training and testing lists in Choi et al. (2020). In the MMUIT setting, the CelebA-HQ and the AFHQ datasets are tested with a 256×256 resolution. In the TUNIT setting, following Baek et al. (2020), we used test images at a 128×128 resolution.

D.3 Baselines

We use the official and public source codes for all the compared methods, namely StarGAN v2 (Choi et al., 2020)[1], HomoGAN (Chen et al., 2019)[2], InterFaceGAN (Shen et al., 2020)[3] and TUNIT (Baek et al., 2020)[4]. Each model is trained using its own best hyperparameter values, as selected by the respective authors and provided jointly with the public code.

E Additional Results

Additional comparisons with sota. The smoothness problem in MMUIT methods is an issue attracting a growing interest in the community, as witnessed, e.g., by Mao et al. (2022), which treats the same problem addressed in our paper. Fig. 8 shows three interpolation results taken from Fig. 5 and 6 of Mao et al. (2022), obtained with three different MMUIT methods. This figure shows

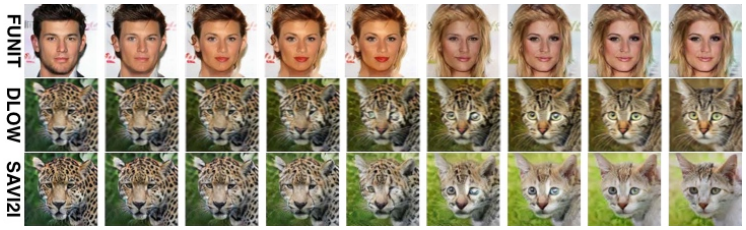

Figure 8: Qualitative comparison with DLOW (Gong et al., 2019), FUNIT (Liu et al., 2019) and SAVI2I (Mao et al., 2022). This figure shows that obtaining smooth interpolations is a widespread issue.

that the non-smoothness problem is shared by other MMUIT models, including SAVI2I, the solution proposed in Mao et al. (2022) (which is, by the way, much more complex than our regularization method). Note also that FUNIT (Liu et al., 2019), despite not producing inter-domain artifacts, generates abrupt changes.

Inter-domain Interpolations. We show additional qualitative comparisons between different MMUIT state-of-the-art methods and our proposal in Figure 9 and Figure 10 for the CelebA-HQ and the AFHQ dataset, respectively. In Fig. 11, we show qualitative comparisons in the TUNIT setting.

Similarly to the results showed in the main paper, we observe that our method generates very smooth inter-domain interpolations, while StarGAN v2 generates artifacts along the interpolation line, and HomoGAN produces very little changes between domains. In CelebA-HQ, our visual results are very similar to InterFaceGAN, which is based on the training-expensive model StyleGAN (Karras et al., 2019; 2020).

Figure 12 and Figure 13 show additional qualitative examples on the CelebA-HQ dataset, while Figure 14 and Figure 15 show additional examples on the AFHQ dataset.

[1] https://github.com/clovaai/stargan-v2
[2] https://github.com/yingcong/HomoInterpGAN
[3] https://github.com/genforce/interfacegan
[4] https://github.com/clovaai/tunit

Intra-domain Interpolations Fig. 16 and Fig. 17 show intra-domain interpolation examples of our model in the MMUIT setting.

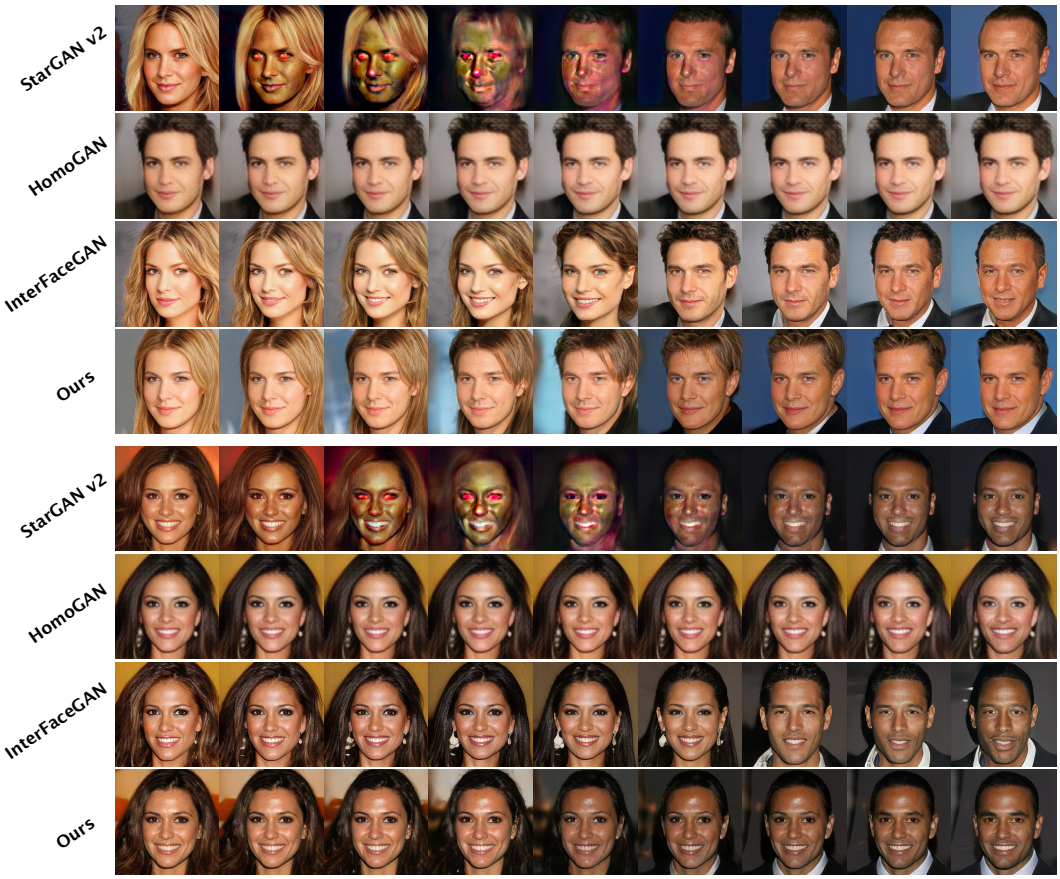

Figure 9: CelebA-HQ dataset: qualitative comparisons between StarGAN v2 (Choi et al., 2020), HomoGAN (Chen et al., 2019), InterFaceGAN (Shen et al., 2020) and our proposed method on gender translation.

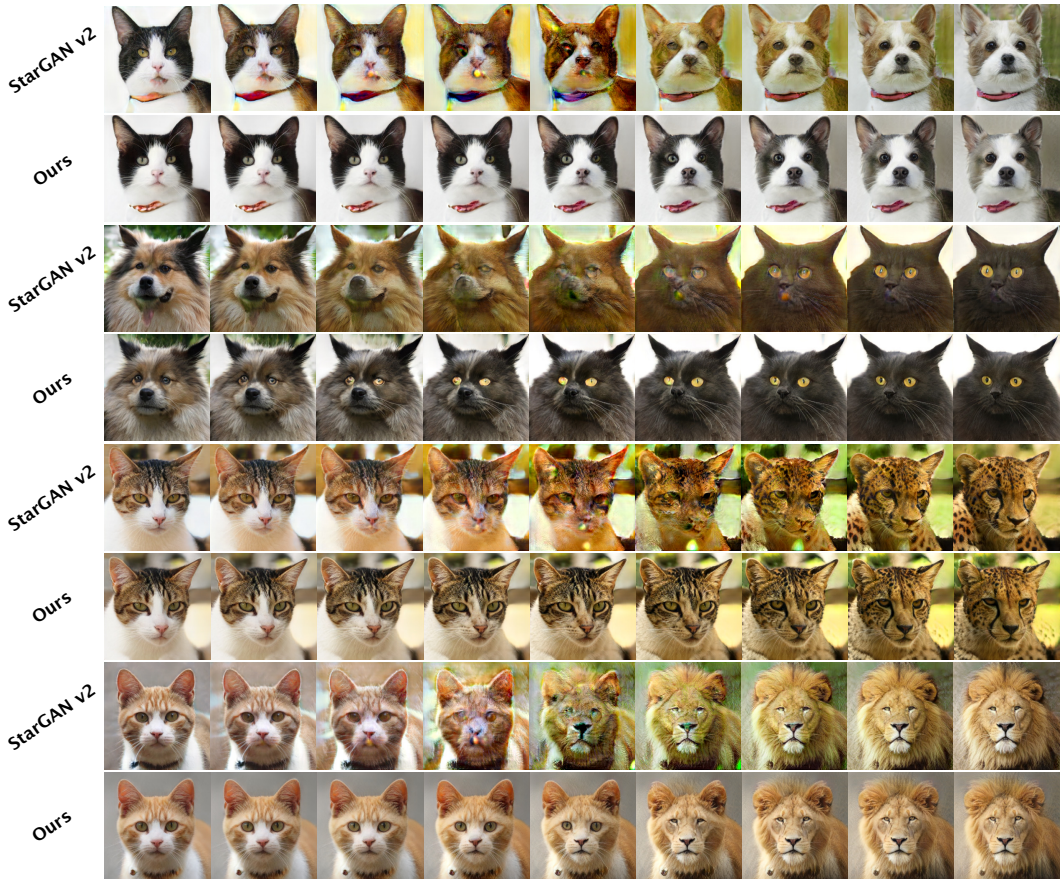

Figure 10: AFHQ dataset: qualitative comparisons between StarGAN v2 (Choi et al., 2020) and our proposed method on animal translation.

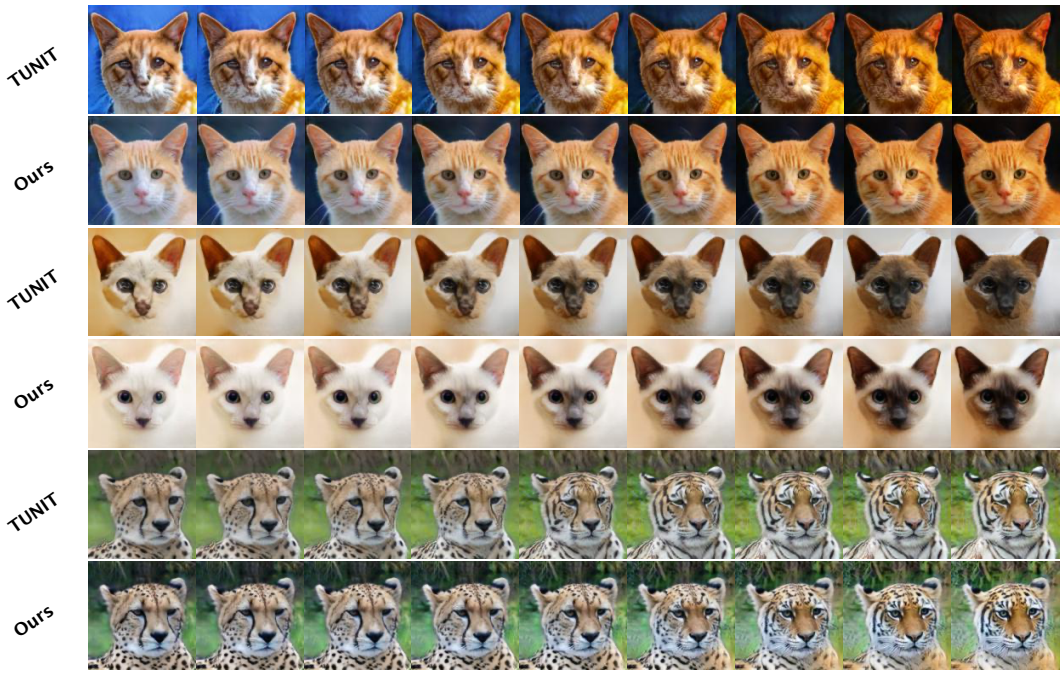

Figure 11: AFHQ dataset: qualitative comparisons between TUNIT (Baek et al., 2020) and our proposed method on animal translation.

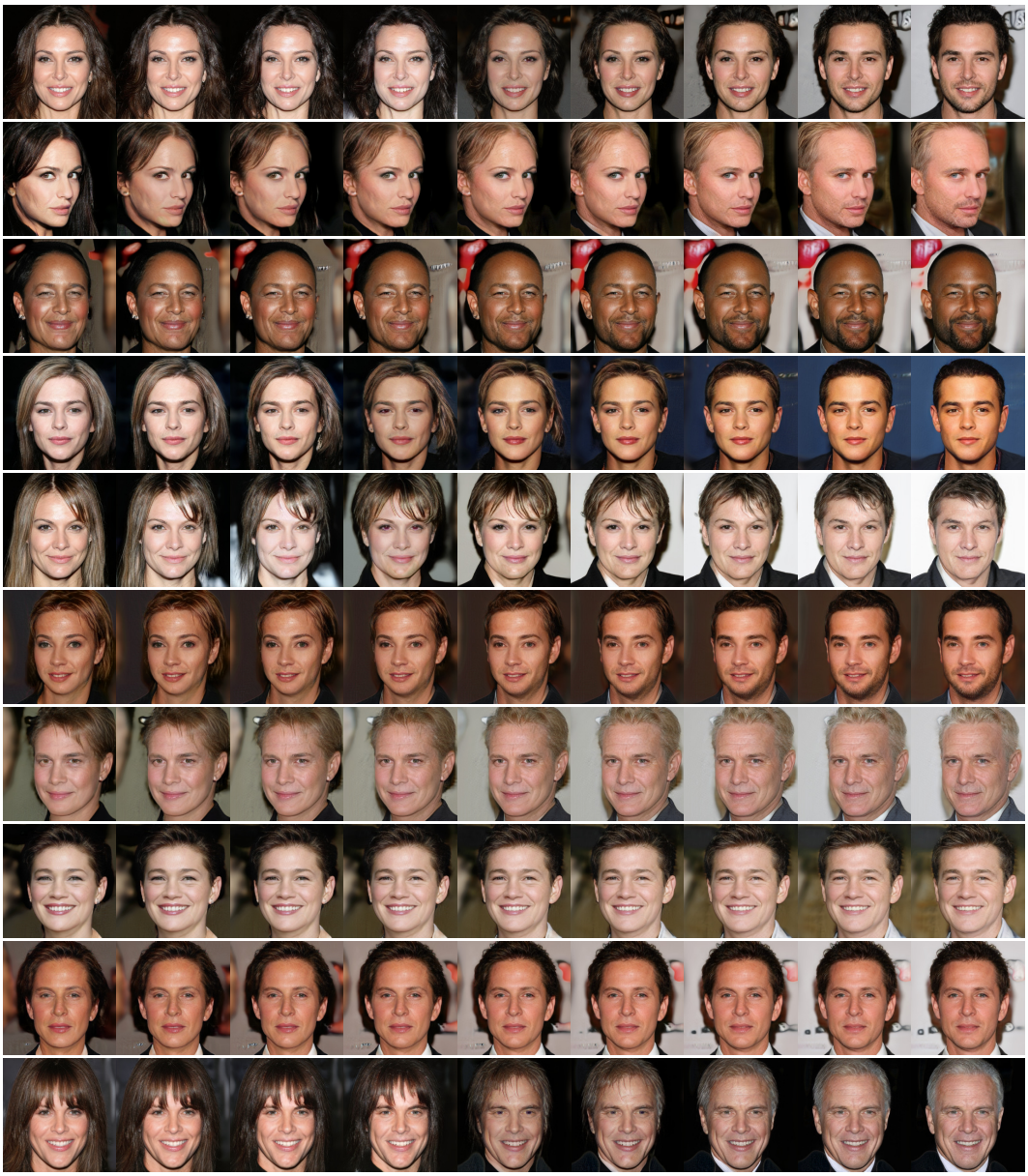

Figure 12: More examples of gender translation on the CelebA-HQ dataset (Karras et al., 2018).

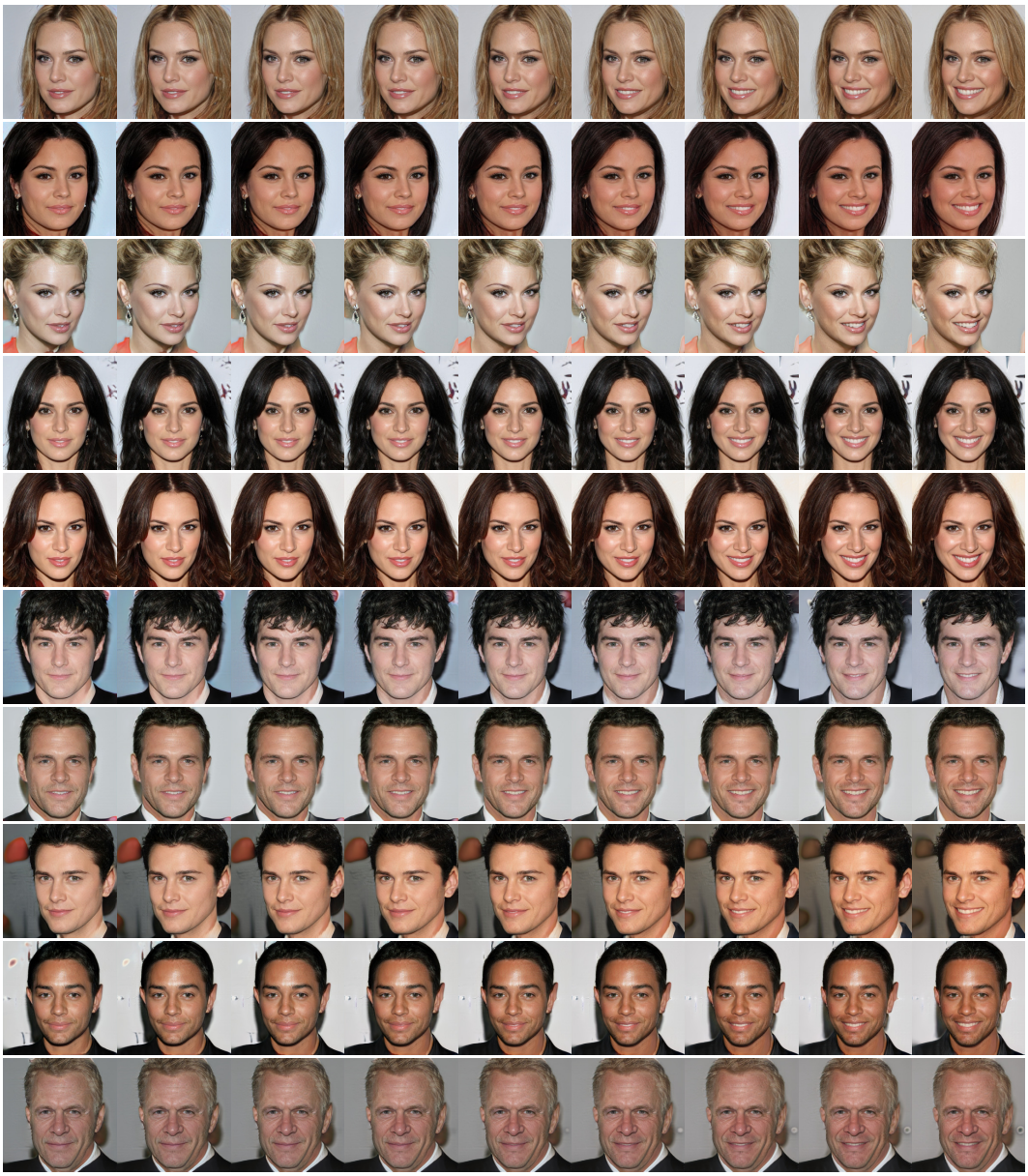

Figure 13: More examples of smile translation on the CelebA-HQ dataset (Karras et al., 2018).

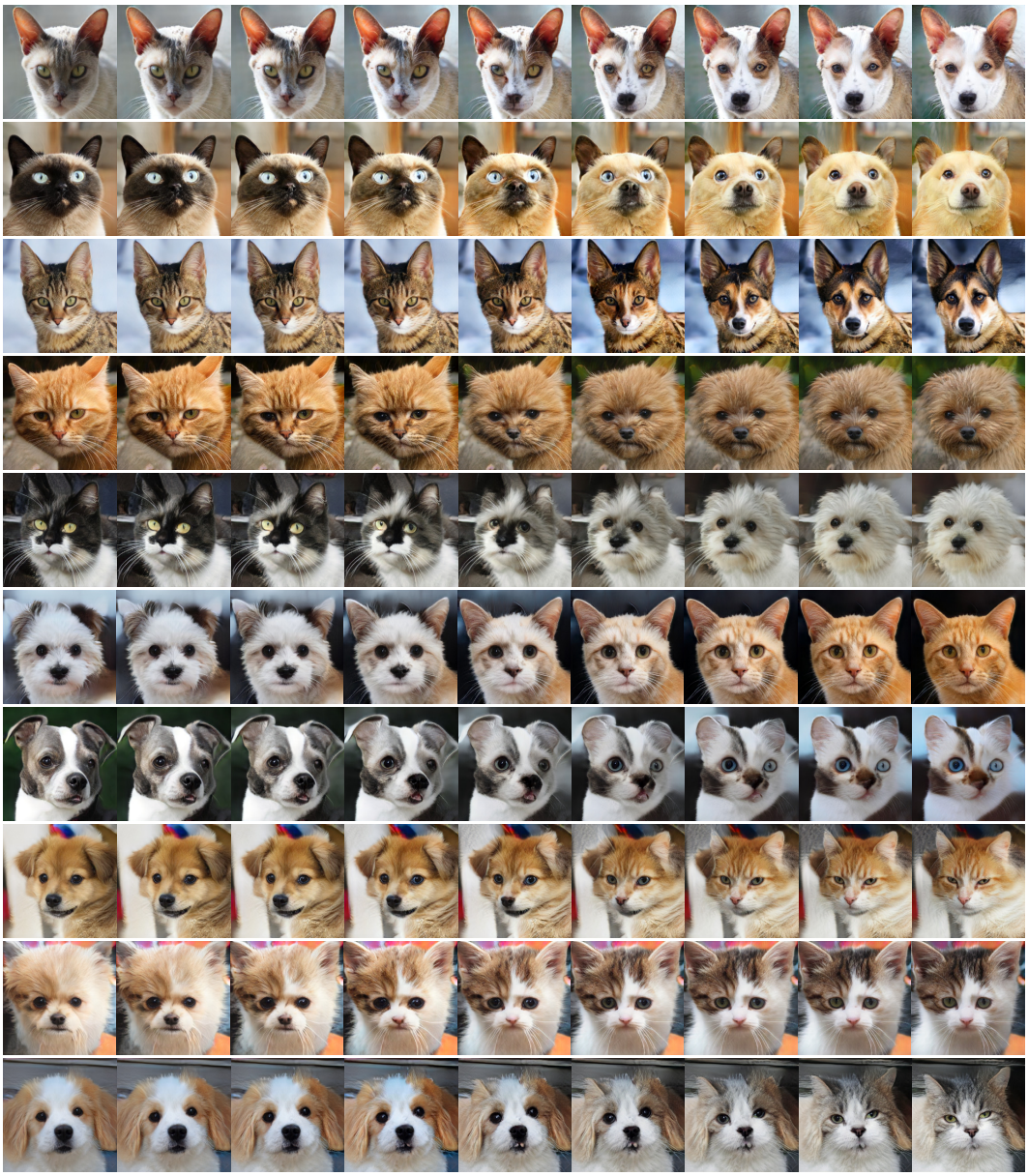

Figure 14: More examples of animal face translation on the AFHQ dataset (Choi et al., 2020).

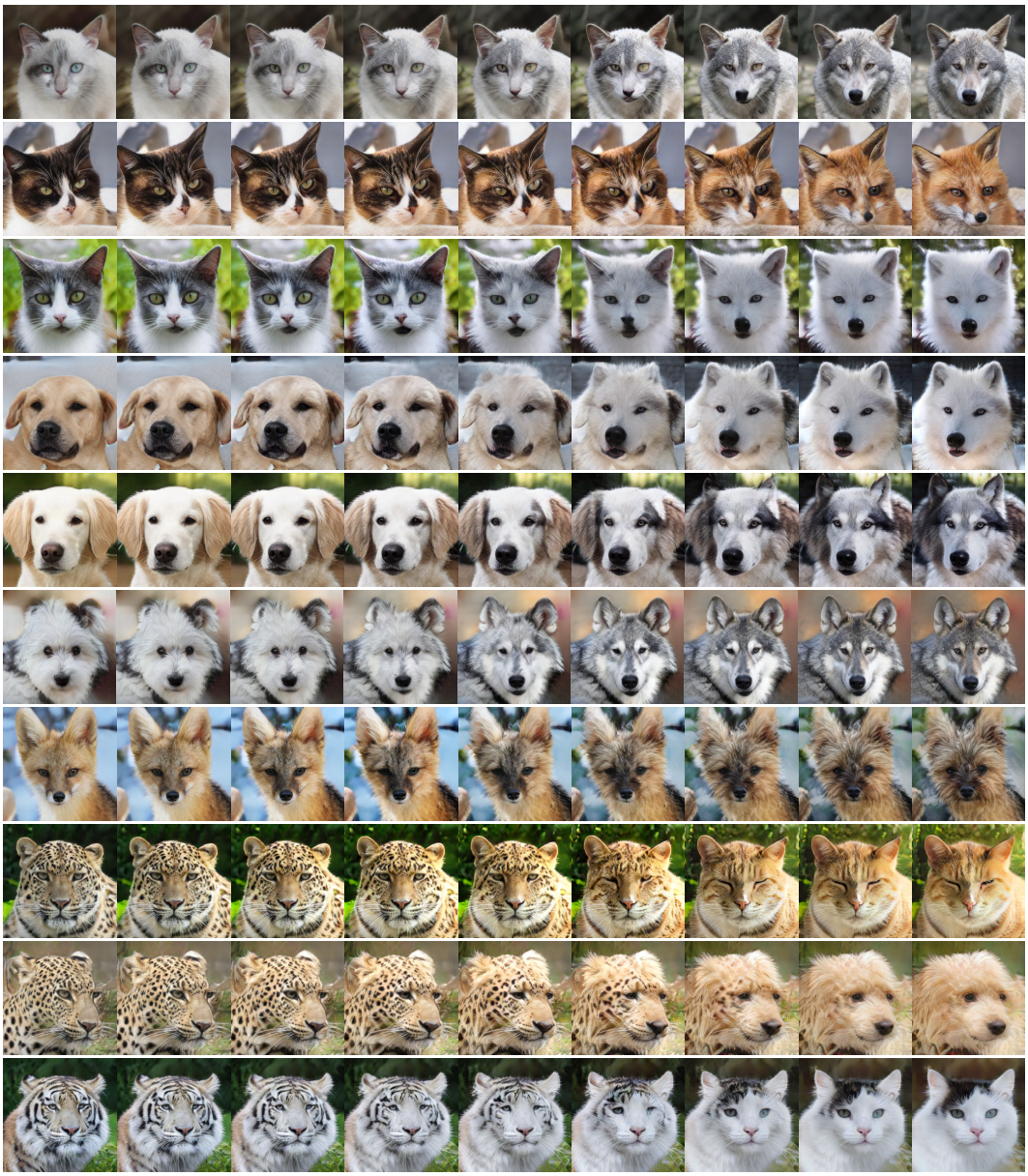

Figure 15: More examples of animal face translation on the AFHQ dataset (Choi et al., 2020).

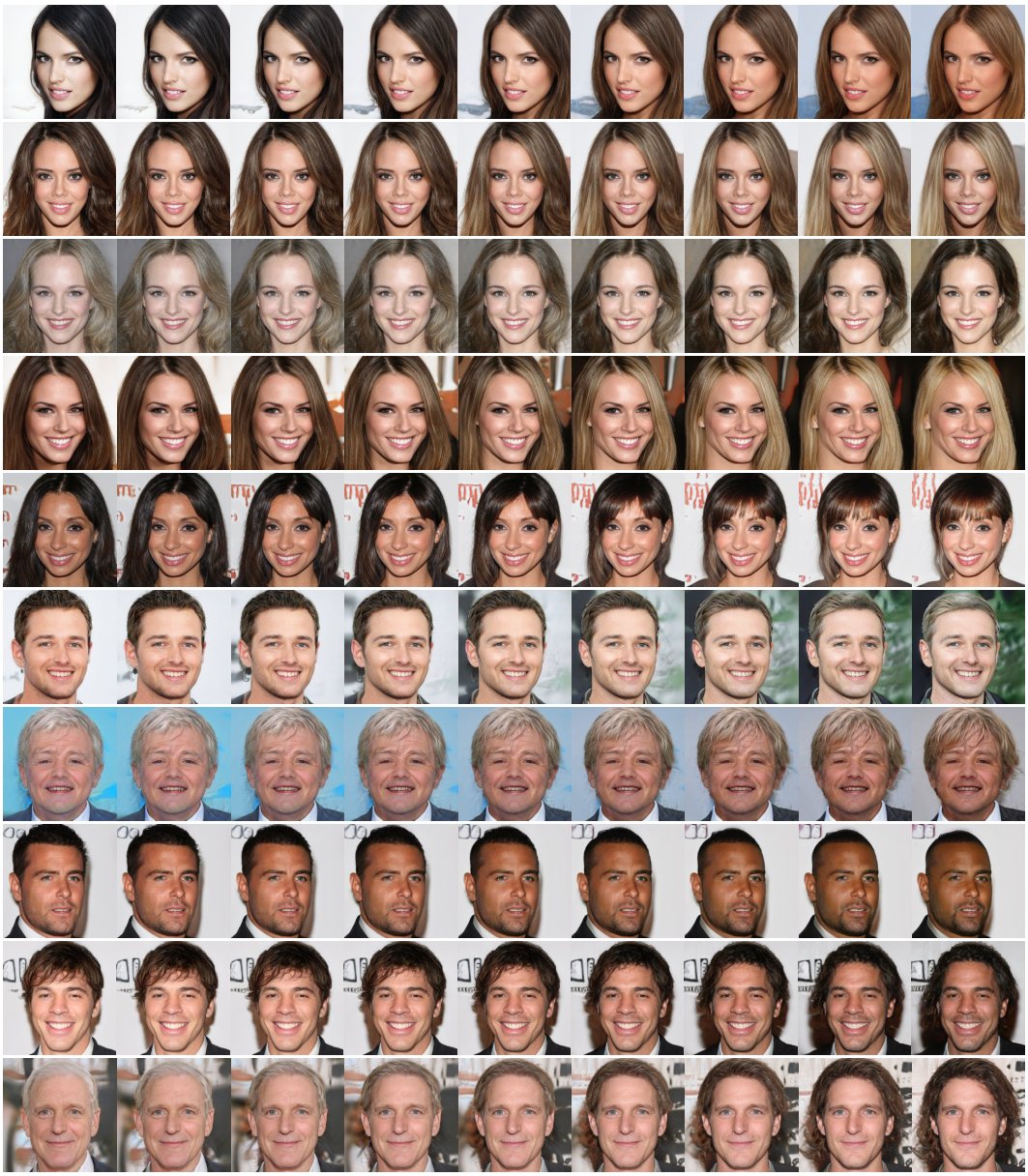

Figure 16: Intra-domain interpolation examples of our model on the CelebA-HQ dataset.

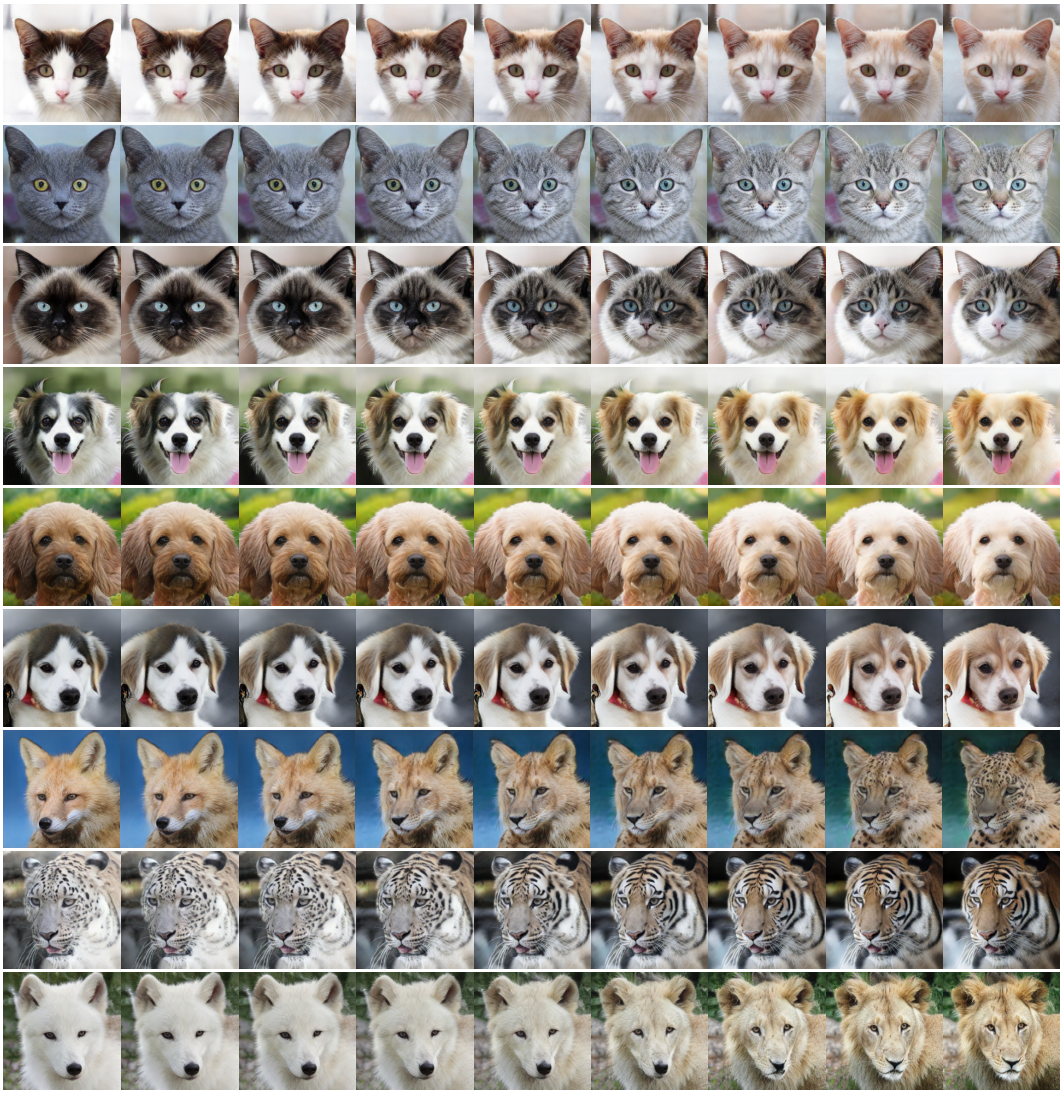

Figure 17: Intra-domain interpolation examples of our model on the AFHQ dataset. Note that the "wildlife" domain in AFHQ contains different animal species, and this is why, e.g., in the last row, a wolf is transformed into a lion.