TransEditor: Transformer-Based Dual-Space GAN for Highly Controllable Facial Editing

Yanbo Xu[1,4*]   Yueqin Yin[1*]   Liming Jiang[2]   Qianyi Wu[5]

Chengyao Zheng[3]   Chen Change Loy[2]   Bo Dai[2]   Wayne Wu[1,3✉]

[1]Shanghai AI Laboratory   [2]S-Lab, Nanyang Technological University   [3]SenseTime Research

[4]Hong Kong University of Science and Technology   [5]Monash University

yxubu@connect.ust.hk   yinyueqin0314@gmail.com   qianyi.wu@monash.edu

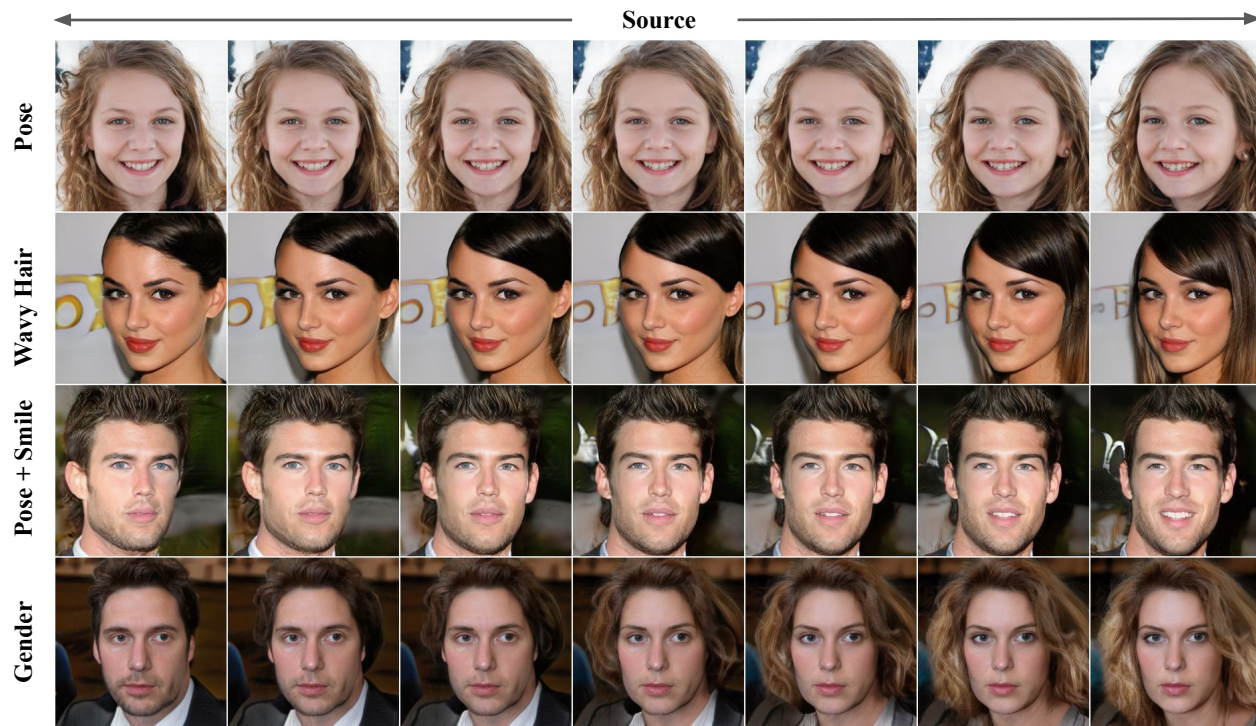{liming002, ccloy, bo.dai}@ntu.edu.sg   zhengchengyao@sensetime.com   wuwenyan0503@gmail.com

Figure 1. **Diverse editing examples by TransEditor.** Images on the fourth column are the sampled source images, which are semantically interpolated to the left and right sides. With our proposed dual latent spaces, namely $\mathcal{P}$-space and $\mathcal{Z}$-space, different attributes can be edited in one space alone or in dual spaces simultaneously, demonstrating the flexibility of our method. Head pose (the first row) and wavy hair (the second row) are edited via the $\mathcal{P}$-space alone. Disentangled dual spaces enable the editing of pose and smile simultaneously (the third row) by editing pose in the $\mathcal{P}$-space and smile in the $\mathcal{Z}$-space. The Cross-Space Interaction allows the cooperated editing of complex attributes like gender (the last row).

Abstract

*Recent advances like StyleGAN have promoted the growth of controllable facial editing. To address its core challenge of attribute decoupling in a single latent space, attempts have been made to adopt dual-space GAN for better disentanglement of style and content representations. Nonetheless, these methods are still incompetent to obtain plausible editing results with high controllability, especially for complicated attributes. In this study, we highlight the importance of interaction in a dual-space GAN for more controllable editing. We propose **TransEditor**, a novel Transformer-based framework to enhance such interaction. Besides, we develop a new dual-space editing and inversion strategy to provide additional editing flexibility. Extensive experiments demonstrate the superiority of the proposed framework in image quality and editing capability, suggesting the effectiveness of TransEditor for highly controllable facial editing. Code and models are publicly available at https://github.com/BillyXYB/TransEditor.*

---

1. Introduction

High-fidelity generative modeling has made a dramatic leap thanks to the progress of Generative Adversarial Networks (GANs) [13, 22, 23]. These efforts have expedited the advancement in facial editing [14, 16, 40], an important downstream task with many practical applications, circumventing the cumbersome manual editing processes. Nevertheless, *highly controllable* facial attribute editing remains challenging when applying current approaches [23–25].

The main challenge of highly controllable facial editing lies in the clean disentanglement of attributes in the latent space. For instance, it is expected to maintain the consistent face characteristic when manipulating the head pose of a portrait. Representative GAN-based methods [23, 24] explored the single latent space representation and the style modulation technique of generators for better image generation and semantic editing. On another note, some studies [6, 39] focus on the local-region editing for specific facial parts in a fine-grained manner. However, under the single latent space design, these works still suffer from the latent entanglement of certain complicated attributes, such as facial pose.

Working towards these challenges, recent studies [4, 25] have presented preliminary attempts that leverage the idea of dual latent spaces for style and content disentanglement within the StyleGAN [22] architecture, achieving plausible semantic separation in each space. However, these attempts may not be adequate to take full advantage of the potential of a dual-space GAN. We empirically observed that these methods fail to obtain decent facial editing results, especially in complicated attributes, as shown in Fig. 6. Altering the pose of a face may produce dissatisfactory artifacts, and a drastic change in hue easily occurs when interpolating the style code of DAT [25] by fixing the content code. Accordingly, we conjecture that the underlying cause is the lack of *interaction* between the two latent spaces, and thus editing one space may distort the other, making the controllable facial editing infeasible.

In this paper, we show that the latent space *interaction* in a dual-space GAN plays a significant role in facial editing. Inspired by recent immense success of Transformers [5, 9, 20, 27, 38] in vision tasks, we propose a novel Transformer-based dual-space GAN, named **TransEditor**, to strengthen the dual latent space interaction. The two latent spaces in our generator, *i.e.*, $\mathcal{P}$-space and $\mathcal{Z}$-space, serve as the initial input feature map of the generator and the style modulation [23] for all layers, respectively. Specifically, we propose a *Transformer-Based Cross-Space Interaction* module, where we incorporate a cross-space attention mechanism based on the Transformer to enhance the interaction between these two spaces, facilitating highly controllable facial editing. The design of interaction is *nontrivial* since potential entanglement might be aggravated via the interaction. We propose to use the $\mathcal{P}$-space as the query and the $\mathcal{Z}$-space as the key and value. Therefore, $\mathcal{P}$-space is only exploited to re-weight the value matrix from $\mathcal{Z}$-

space through the Cross-Space Interaction module, hence still being disentangled with the refined code from $\mathcal{Z}$-space. Thanks to the design of the interaction module, our dual spaces allow flexible editing while maintaining disentanglement semantically in an expressive manner, with $\mathcal{P}$-space mainly controlling the structural information and $\mathcal{Z}$-space determining the texture representation. Then, different from previous methods [32–34] that perform editing in a single space, we devise a novel *Dual-Space Editing* strategy to leverage the controllability brought by TransEditor (Fig. 1). Further, to enable the real image editing, we also extend existing inversion techniques [31] to fit the proposed dual-space design.

The contributions of this work can be summarized as follows. We propose TransEditor, a novel Transformer-based dual-space GAN for highly controllable facial attribute editing. Through the introduced cross-space attention mechanism, the two latent spaces can establish meaningful interaction in a disentangled manner. Additionally, we develop a novel flexible dual-space image editing and inversion strategy to leverage the improved controllability provided by TransEditor. Extensive experiments demonstrate the effectiveness of TransEditor in highly controllable and stable facial attribute editing, outperforming state-of-the-art approaches especially for complicated attributes.

2. Related Work

2.1. Structured Latent Space for Disentanglement

Single Latent Space. ProgressiveGAN [21] uses a latent code as the input feature map to control the generation of the whole image. StyleGAN [22, 23] improves the disentanglement using a mapping network that maps the initial distribution to an intermediate latent space w, thus the distribution of latent codes can be reformed to the given real image distribution. Besides, the layer-wise Adaptive Normal Instance (AdaIN) [11, 18] is used to improve the disentanglement further. StyleMapGAN [24] reshapes the latent code into a tensor with spatial structure. The method achieves good reconstruction results but poor semantic editing performance, known as the Editability-Distortion trade-off [2, 36, 44]. Its encoder also needs to be trained together, the training will fail otherwise. The reason is that the network relies on the encoder to make the projected style map of the real images achieve local correspondence [24].

Dual Latent Spaces. To achieve better disentanglement or spatial awareness, some studies that explore dual latent spaces design have been proposed. SNI [4] separates the latent space into spatially-variable and spatially-invariable parts. The separation allows a certain degree of local editing. However, the limited capacity of the spatial code (4×4 or 8×8) may cause failure when performing content control [25]. DAT [25] improves the structure by introducing a symmetric space similar to the origin latent space and utilizing separate operations during the generating process for disentanglement. However, the content space pro-

posed by DAT [25] operates on each pixel individually, so it lacks a global structure connection and cannot perform well on some attributes with large structural changes like pose. There are other flavors of dual latent spaces for various tasks. Zhu *et al*. [43] propose $\mathcal{F}$-space and $\mathcal{S}$-space for image compositing. Park *et al*. [29] use a style code vector and a structure code with spatial dimension. [3, 26] disentangle the pose and style for virtual try-on.

2.2. GAN-Based Image Editing

Latent Space Manipulation. Since the latent space of GAN is semantic-aware, it is possible to manipulate the attributes of the generated image through latent space navigation [42]. Previous works aiming at finding semantic directions can typically be divided into the supervised ones [12, 32], the self-supervised ones [19, 35], and the unsupervised ones [15, 34]. InterfaceGAN [32] finds hyperplanes in the latent space that corresponds to particular attributes by using attribute classifiers and SVM. Jahanian *et al*. [19] need a large set of paired examples to fit the image transformations. SeFa [34] performs decomposition on eigenvectors and finds interpretable directions without extra labels. Meanwhile, there are some studies [6, 39] that focus on local editing for specific facial parts. StyleSpace [39] proposes a space of channel-wise style parameters to perform highly localized and disentangled attribute editing.

GAN Inversion. To edit a real image using Latent Space Manipulation, the image should be projected back into the latent space of the generator [42]. Optimization-based methods [1, 28] directly optimize the latent code to minimize the pixel-wise reconstruction loss. However, the optimization process is slow and the inverted code may land out of the original semantic manifold, making the editing process suffer from meaningful semantic manipulation. This problem is resolved by learning-based methods [31, 36], which use an extra encoder trained and guided by the generator to directly embed the image into a latent code, offering benefit during real image editing.

2.3. Transformer-Based Interaction

Multi-head attention module is often used in some multi-modality tasks to establish the inter-modal interaction between text modalities and visual modalities, such as image captioning [7, 17] and text-to-image translation [8, 30]. TransStyleGAN [27] introduces a Transformer structure to model the correlation between different layers of style codes within a single latent space based on the StyleGAN2 architecture. In this work, we leverage the multi-head attention module in the Transformer [37] to establish the interaction between the proposed dual latent spaces (*i.e.*, $\mathcal{Z}$-space and $\mathcal{P}$-space), facilitating more flexible and controllable attribute editing.

3. Methodology

Our goal is to achieve more controllable facial attribute editing through the disentangled but collaborative dual spaces. Fig. 2 (a) shows the structure of our model. We propose two latent spaces $\mathcal{P}$ and $\mathcal{Z}$ with separate mappings (Sec. 3.1), which are used as the initial input feature map of the generator and the layer-wise style modulation, respectively. Then, an interaction module based on the Transformer (Sec. 3.2) is proposed to model the interaction between these two spaces, making them more balanced during editing. Further, we develop a new dual-space image editing and inversion strategy (Sec. 3.3) for real image editing.

3.1. Dual Latent Spaces

Instead of learning a generator $\mathbf{G}$ that maps a single Gaussian distribution to an image $\boldsymbol{x}$, *i.e.*, $\boldsymbol{x} = \mathbf{G}(\boldsymbol{z}), \boldsymbol{z} \in \mathcal{N}(0, \mathbf{I})$, two separated latent spaces are used in our method, denoted as $\mathcal{Z}$ and $\mathcal{P}$. Thus, our generation process can be reformulated as:

$$\boldsymbol{x} = \mathbf{TransEditor}(\boldsymbol{z}, \boldsymbol{p}), \quad (1)$$

where $(\boldsymbol{z}, \boldsymbol{p}) \in \mathcal{Z} \times \mathcal{P}$.

Note that it is *non-trivial* to determine how the two spaces should be integrated with the generator. For the StyleGAN2-based architecture [23], at each layer, the generation process can be expressed as:

$$\boldsymbol{F}_{i+1} = \mathbf{ModuConv}(\boldsymbol{T}_{\boldsymbol{w}_i}, \boldsymbol{F}_i), \quad (2)$$

where $\boldsymbol{F}_i$ denotes the feature map produced from the previous $i - 1$ layer, and $\boldsymbol{T}_{\boldsymbol{w}_i}$ is the modulation and demodulation process determined by the layer style code $\boldsymbol{w}_i$. Although the modulation and demodulation are performed at every layer, each feature map is the convoluted result from the previous layer. Thus, the initial feature map $\boldsymbol{F}_0$ is the foundation that the whole generation process is based on. Therefore, compared with StyleGAN2 in Fig. 2(b), to provide more controllability, we replace the learned constant input with the latent input from $\mathcal{P}$-space. Besides, the additional latent capacity provided by $\mathcal{P}$-space allows us to remove the noise inputs.

In addition, we consider that reshaping a single sampled vector for the entire latent input is inherently entangled [4], hence our two spaces $\mathcal{Z}$ and $\mathcal{P}$ consist of separate sub-vectors, *i.e.*, $\boldsymbol{z} \in \mathbb{R}^{n \times 512}, \boldsymbol{p} \in \mathbb{R}^{n \times 512}$. To further encourage a desirable disentanglement property, we exploit separate mappings for the dual spaces: $(\boldsymbol{z}^+, \boldsymbol{p}^+) \in M_{\boldsymbol{z}}(\mathcal{Z}) \times M_p(\mathcal{P})$, which can be written as:

$$\boldsymbol{z}^+ = \begin{bmatrix} \boldsymbol{z}_1^+ \\ \boldsymbol{z}_2^+ \\ \ldots \\ \boldsymbol{z}_n^+ \end{bmatrix} = \begin{bmatrix} M_{\boldsymbol{z}_1} & 0 & \ldots & 0 \\ 0 & M_{\boldsymbol{z}_2} & \ldots & 0 \\ \ldots & & & \\ 0 & 0 & \ldots & M_{\boldsymbol{z}_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \ldots \\ \boldsymbol{z}_n \end{bmatrix} \quad (3)$$

$$\boldsymbol{p}^+ = \begin{bmatrix} \boldsymbol{p}_1^+ \\ \boldsymbol{p}_2^+ \\ \ldots \\ \boldsymbol{p}_n^+ \end{bmatrix} = \begin{bmatrix} M_{\boldsymbol{p}_1} & 0 & \ldots & 0 \\ 0 & M_{\boldsymbol{p}_2} & \ldots & 0 \\ \ldots & & & \\ 0 & 0 & \ldots & M_{\boldsymbol{p}_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \ldots \\ \boldsymbol{p}_n \end{bmatrix} \quad (4)$$
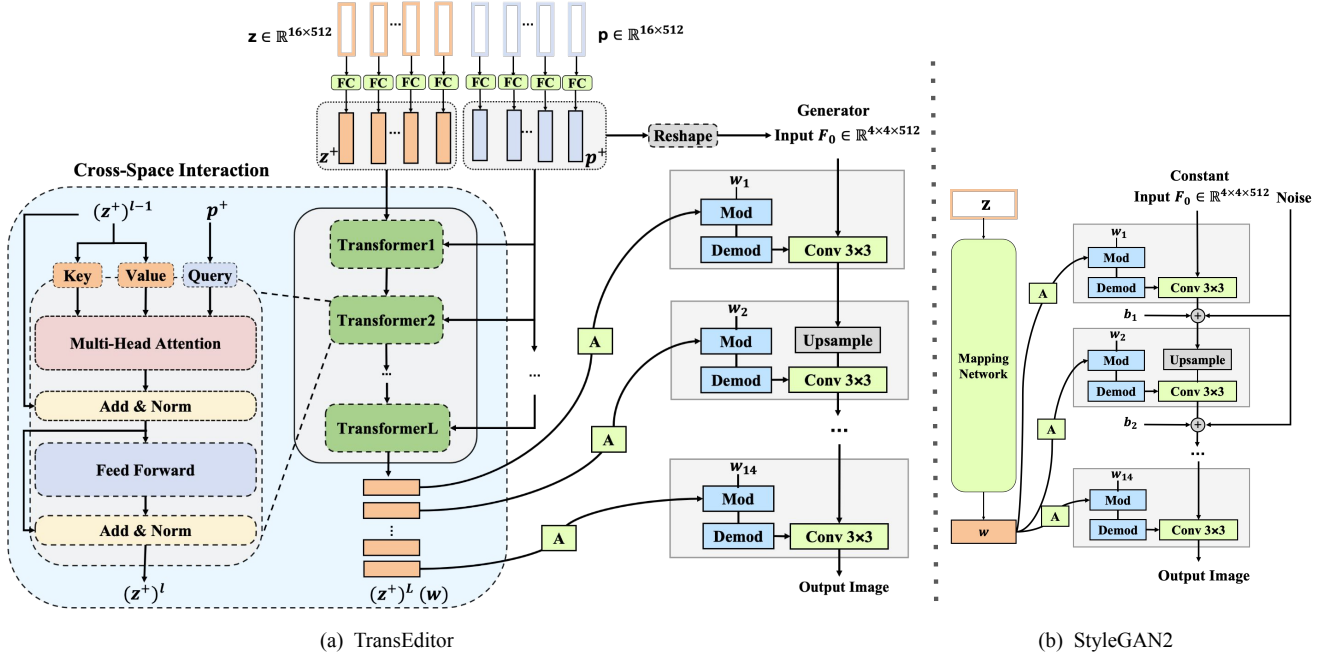
Figure 2. **Network architecture of TransEditor.** (a) shows the structure of our model, which contains two separate latent spaces $\mathcal{Z}$ and $\mathcal{P}$, a Cross-Space Interaction module based on the Transformer, and a generator. Compared to (b) StyleGAN2 [23] that leans a constant input, our generator uses the $\boldsymbol{p}^+$ code as the input and the interaction result $(\boldsymbol{z}^+)^L(\boldsymbol{w})$ for style modulation.

Note that each $M_{\boldsymbol{z}_i}$ or $M_{\boldsymbol{p}_i}$ is a MLP [10] module, and the mapped space $\boldsymbol{z}^+ \times \boldsymbol{p}^+ \in \mathbb{R}^{n \times 512} \times \mathbb{R}^{n \times 512}$. In our experiments, we set $n = 16$.

3.2. Transformer-Based Cross-Space Interaction

In a dual-space GAN, naive generation using two separated latent codes might be problematic. SNI [4] shows that adding the style code at all layers influences the disentanglement performance, *i.e.*, changing the style code at early layers of the generator affects the structural information. Although DAT [25] achieves better disentanglement than SNI, we find that when fixing DAT's content code and interpolating its style code, drastic changes in hue could easily appear and cause artifacts. We attribute this phenomenon to the lack of interaction in the dual latent spaces, as they are not correlated by any means.

Inspired by the cross-domain Transformer model [30], we correlate the two separated spaces via a cross-attention-based interaction module. The mapped latent code $\boldsymbol{z}^+$ is used as the key (K) and value (V), and the latent code $\boldsymbol{p}^+$ as the query (Q). The interaction in the l-th layer transformer can be written as:

$$Q = \boldsymbol{p}^+ W^Q, K = (\boldsymbol{z}^+)^l W^K, V = (\boldsymbol{z}^+)^l W^V, \quad (5)$$

$$(\boldsymbol{z}^+)^{l+1} = \mathbf{softmax}(\frac{QK^\mathsf{T}}{\sqrt{d_k}})V + (\boldsymbol{z}^+)^l, \quad (6)$$

where W^Q, W^K, W^V are linear projection matrices, and d_k is the common dimensionality of the latent code. In other words, an attention process queried by $\boldsymbol{p}^+$ will be operated on $\boldsymbol{z}^+$. Since $\boldsymbol{p}^+$ acts as the query only, the two spaces are still separated, yet $\boldsymbol{z}^+$ has adopted to the query $\boldsymbol{p}^+$. This design enables better disentanglement of the dual spaces while maintaining global consistency during editing (see Sec. 4.3).

The refined latent code $(\boldsymbol{z}^+)^L$, that is $\boldsymbol{w}$, serves as the style modulation parameters of the generator $\mathbf{G}$. The generation process of image $\boldsymbol{x}$ is thus formulated as:

$$\boldsymbol{x} = \mathbf{G}(\boldsymbol{w}, \boldsymbol{F}_0), \quad (7)$$

where $\boldsymbol{w}$ is the modulation input produced from Eq. (6), and the Reshape module (see Fig. 2(a)) reshapes the mapped $\boldsymbol{p}^+$ code into the initial feature map $\boldsymbol{F}_0$ whose spatial dimension is suitable as the input to the generator. The training of our generator is entirely unsupervised, which only applies the adversarial loss and path length regularization following StyleGAN2 [23].

3.3. Dual-Space Image Editing and Inversion

Intuitively, some complicated attributes (*e.g.*, age) may involve changes in both the facial structure and the texture. With our disentangled dual latent spaces, we propose to edit such complex attributes using $\mathcal{Z}$ and $\mathcal{P}$ together. To our knowledge, this is the first study that performs attribute editing through the two latent spaces simultaneously.

As enforcing the semantic distribution to fit the original Gaussian distribution is undesirable [22], the editing procedure of our model via the latent manipulation is operated on the product space $\mathcal{Z}^+ \times \mathcal{P}^+$. Hence, our dual-space manipulation procedure can be written as:

$$I_{(\mathcal{Z}^+ \times P^+)}(\boldsymbol{z}^+, \boldsymbol{p}^+) = (I_{\mathcal{Z}^+}(\boldsymbol{z}^+), I_{\mathcal{P}^+}(\boldsymbol{p}^+)), \quad (8)$$
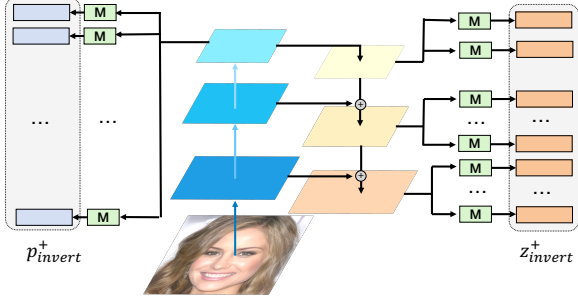
Figure 3. **Dual-Space Inversion Architecture.** The inverted z^+ code comes from different layers of the extracted feature map, while p^+ code comes from the highest feature map.

where $z^+ \in \mathcal{Z}^+, p^+ \in \mathcal{P}^+$, and I_i represents the manipulation operated on space i.

It is noteworthy that the two operations $I_{\mathcal{Z}^+}$ and $I_{\mathcal{P}^+}$ could be different. In our case, we utilize the linear latent manipulation method by InterFaceGAN [33] for both spaces. Specifically, for each attribute, we use an SVM classifier to train two hyper-planes represented by the normal vectors in the two separated latent spaces $\mathcal{Z}^+$ and $\mathcal{P}^+$. Thereafter, for known latent codes z^+ and p^+, we can move λ_z steps along n_z and λ_p steps along n_p to get the new latent codes $(z^+ + \lambda_z * n_z, p^+ + \lambda_p * n_p)$. For the attributes that are fully contained in one space only, we edit them by operating on that space solely(*e.g.*, $\lambda_p = 0$). More complicated attributes like gender will be better manipulated if the two spaces are used simultaneously. The manipulated codes will then be used for generating the edited image.

To edit real images, it is necessary to invert the images back into the dual latent spaces. We adopt a pSp [31] encoder architecture for our dual-space image inversion. As shown in Fig. 3, three levels of feature maps of the input real image are first extracted using a feature pyramid. Since our z^+ space has a layer-wise structure, different features are used to produce each z_i^+. The p^+ latent code is only mapped from the highest-level feature in the encoder and injected as the initial feature map input of the generator.

The aforementioned inversion strategy maps real images into our trained dual latent spaces, and thereby we can apply linear latent manipulation to perform dual-space editing.

4. Experiments

In our experiments, we first evaluate the effectiveness of our method in highly controllable facial editing (Sec. 4.1). Then, we compare our mothod with three representative state-of-the-art methods (Sec. 4.2), *i.e.*, the single-space methods (StyleGAN2 [23], StyleMapGAN [24]), and the dual-space method (DAT [25]), on both qualitative (Sec. 4.2.1) and quantitative (Sec. 4.2.2) aspects. In addition, we perform an ablation study (Sec. 4.3) to isolate each pivotal components of our method. We trained TransEditor on CelebA-HQ [21] and FFHQ [22] at a resolution of 256×256. Details of experiment settings and implementation can be found in our supplementary material. The quan-

Table 1. The **LPIPS** score comparison on FFHQ-256 [22]. A higher **LPIPS** represents a more diverse result. $\textbf{LPIPS}_z$ is obtained by fixing p code and randomly sampling z code, similar for $\textbf{LPIPS}_p$. Δ_{LPIPS} is the difference between $\textbf{LPIPS}_z$ and $\textbf{LPIPS}_p$, and a lower value shows more balanced dual spaces.

Method	$\textbf{LPIPS}_{all} \uparrow$	$\textbf{LPIPS}_z$	$\textbf{LPIPS}_p$	$\Delta_{LPIPS}\downarrow$
DAT [25]	0.5596	0.5221	0.4071	0.1150
Ours	**0.5618**	0.5118	0.4456	**0.0662**

titative metrics used in this section are shown as follows.
Re-scoring Calculation. We designed this metric given by C_e/C_i to quantitatively evaluate the editing performance, where C_e and C_i denote the cumulative change of attribute score of the edited and the influenced attributes, respectively. It effectively measures how the editing of one attribute affects other attributes. Details of this metric are included in the supplementary material. A lower value represents a less entangled (better) editing result.
Learned Perceptual Image Patch Similarity (LPIPS). LPIPS [41] measures the diversity of a latent space. A larger LPIPS score indicates a more diverse space.

4.1. Latent Space Interpolation and Editing

Our dual latent spaces achieve a certain degree of semantic separation, with $\mathcal{P}$-space controlling structural information like pose and $\mathcal{Z}$-space controlling texture information.
Disentangled and Balanced Dual Spaces. The two latent spaces in our architecture are both semantically meaningful while achieving desirable disentanglement. Specifically, the head pose is entirely controlled by the $\mathcal{P}$-space. As shown in Fig. 4a, when re-sampling the z code, all the generated images share the same head pose. On the other hand, when z code is fixed, a similar texture, *i.e.*, color, makeup, race, will appear on all results (Fig. 4b). Besides, under the dual space setting, it is often desirable to have more balanced spaces rather than being dominant by a single space. DAT [25] uses a diversity loss to encourage the diversity of its content space. In Tab. 1, our dual spaces achieve a more balanced space separation than DAT while obtaining a higher overall diversity. The diversity difference between our dual spaces (Δ_{LPIPS}) is 0.0662, which is half of that of DAT [25]. This is consistent with our qualitative observation that both spaces of TransEditor shows higher controllability since they are more balanced.
Latent Space Interpolation. The two spaces are both smooth and semantic-aware, thus it is possible to change certain face attributes by interpolating towards a specific direction. Fig. 4c shows a fixed head pose at each row and a smooth change in texture when z code is being interpolated. Similarly, Fig. 4d shows small changes in texture information at each row with fixed z code, and the consistency in the head pose of each column with the same p code. The smoothness and semantic-aware property of the latent space allows better attribute editing.
Dual Latent Space Editing. Using the dual-space latent manipulation method (Sec. 3.3), attribute editing can be per-

(a) Fix the p code and sample z code

(b) Fix the z code and sample p code

(c) Interpolate z code

(d) Interpolate p code

Figure 4. **Dual Latent Spaces of TransEditor**. Each row in (a) is generated by a fixed p code and a randomly sampled z code. Similarly, each row in (b) is generated by a fixed z code and a randomly sampled p code. In (c), each column starts from the same z code and interpolates towards the same direction. Each row has the same sampled p code. Similarly, each column in (d) starts from the same p code then interpolates in the same way. Each row in (d) shares the same z code.
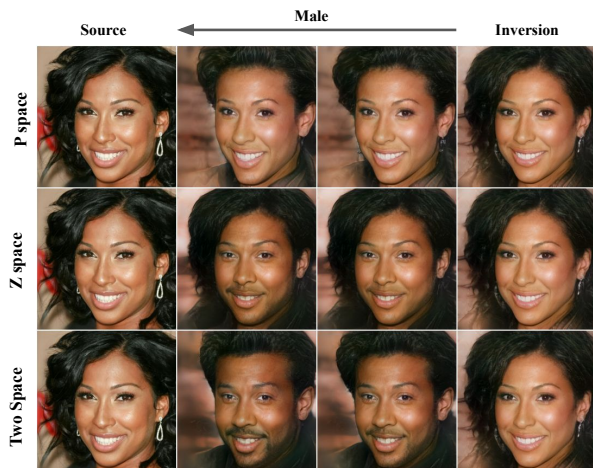

Figure 5. **Dual-Space Editing of attribute Male.** When editing using the p code (first row), the hair volume will decrease. And the face will gradually grow a small beard if using the z code (second row). The third row shows the result of editing the male attribute jointly via both spaces.

formed through linear interpolation towards a normal direction of the trained hyper-planes. The result in Fig. 5 indicates that p code controls the structural information such as hair volume, and z code controls the texture information like the beard, and the joint editing results in the third row demonstrate the cooperation between the two latent spaces can achieve better editing performance. More editing results in Fig. 1 further demonstrate the flexibility of our dual-space editing strategy.

4.2. Comparison to State-of-the-art Methods

4.2.1 Qualitative Evaluation

Sampled Image Editing. Fig. 6 shows the editing results of some sampled images. Since the change in gender may involve variation in both texture and structure information, the editing of gender is accomplished by editing on both $\mathcal{Z}$-space and $\mathcal{P}$-space (Content and Style space for DAT [25]). For pose editing, only content space of DAT [25] and $\mathcal{P}$-space of ours are used. Compared with other methods, our method achieves better editing results. As mentioned in Sec. 2, when the editing involves the style space of DAT [25], the image tone could be easily changed (the third row of Fig. 6a). The manipulation of the head pose is even a more challenging task, as the texture and structure information need to stay aligned to preserve the identity. Fig. 6b shows the advantage of TransEditor. Dur-
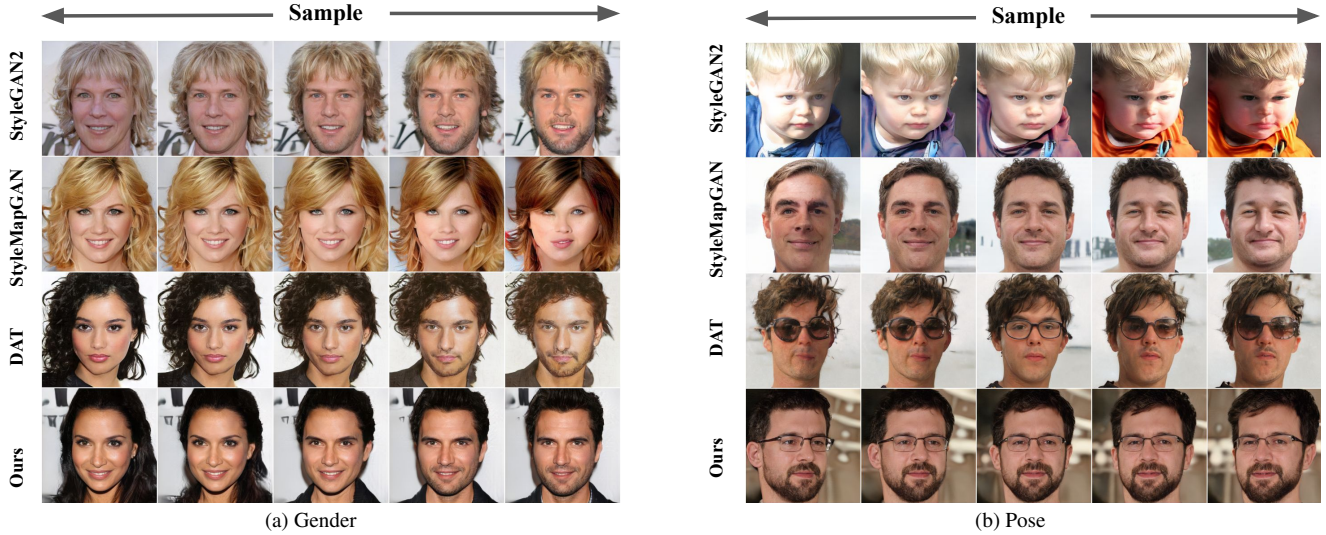
(a) Gender   (b) Pose

Figure 6. **Sample Image Editing Comparison**. Images on the third column are the sampled source images, which are semantically interpolated to the left and right sides. Our method preserves the identity better than other methods when editing gender (a) and pose (b).

ing pose editing, the p code is being manipulated when fixing the z code. However, the interaction process ensures that the style modulation parameter received by the generator has been aligned with the p code (Eq. (6)), thus producing a consistent texture throughout manipulation.

Real Image Editing. To enable real image editing, we use the dual-space inversion method mentioned in Sec. 3.3. The latent manipulation method is identical to the sampled image editing. Fig. 7 shows the comparative results on real image editing with state-of-the-art methods. In Fig. 7a, StyleMapGAN [24] suffers from the global semantic editing, in which the face turns to male with long hair. The distortion of DAT [25] and attribute entanglement of StyleGAN2 [23] can be clearly observed. As shown in Fig. 7b, all the baselines fail to edit the head pose while our method obtains plausible results.

4.2.2 Quantitative Evaluation

For editing performance comparison, we chose three attributes: smile, head pose, and gender, which represent editing in $\mathcal{Z}$-space, $\mathcal{P}$-space, and both spaces, respectively. We use them to calculate our re-scoring metric. The comparative results of our method with StyleGAN2 [23], StyleMap-GAN [24], and DAT [25] are presented in Tab. 2. We observe that when editing each specific attribute, the influence of our model on other attributes is minimal, indicating that our method is the least entangled when performing editing.

4.3. Ablation Study

Space Interaction via Transformer. The space interaction is crucial in our architecture for the semantic balance between the two spaces. Removing the interaction process (*i.e.*, the generator receives two completely independent codes) produces an unbalanced result. Under this setting, the $\mathcal{P}$-space controls the majority of information, leading to

more entangled results during editing and joint changes in color and shape (the first row of Fig. 8).

In contrast, with the cross-space interaction mechanism, the tones of the images keep consistent in a successful pose editing (the third row of Fig. 8). Therefore, establishing a connection between the two spaces enables a more balanced setting, which benefits facial attribute editing.

Dual-Space Design v.s. Single-Space Design. We then evaluate the role of the dual-space design. TransStyle-GAN [37] employs the self-attention mechanism to establish connections between different style codes within the single-space design. However, as can be seen in the second row of Fig. 8, when the step size of pose-editing gets larger, the face orientation is still difficult to change, while the distortion of the face and the change of tone could be serious. This indicates the difficulty of editing pose in a single space, demonstrating the advantage of our dual latent spaces.

Selection of K, Q, V Matrices. For the Cross-space Interaction module, we need to consider the choice of the $\mathbf{K}, \mathbf{Q}, \mathbf{V}$ matrices for the cross-space attention. Some studies in the multi-modality field use the single-modality feature that needs to be refined as the query, and the other modality feature as the key and value. In our case, this is equivalent to using the z^+ space as the query matrix and the p^+ space as the key and value matrix. However, the output of the cross-space attention module, as the refined feature of the z^+ space, is a weighting of the value matrix (p^+ space), which may produce certain entanglement of the two spaces. This setting (Fig. 9a) shows a severe entanglement between $\mathcal{P}$-space and $\mathcal{Z}$-space that swapping p codes result in different textures, which is not desirable for editing. On the other hand, our setting shown in Fig. 9b is much better disentangled, since the p^+ space is only adopted to as query to help update the z^+ space. Therefore, this interaction method in Sec. 3.2 is more desirable for our design.
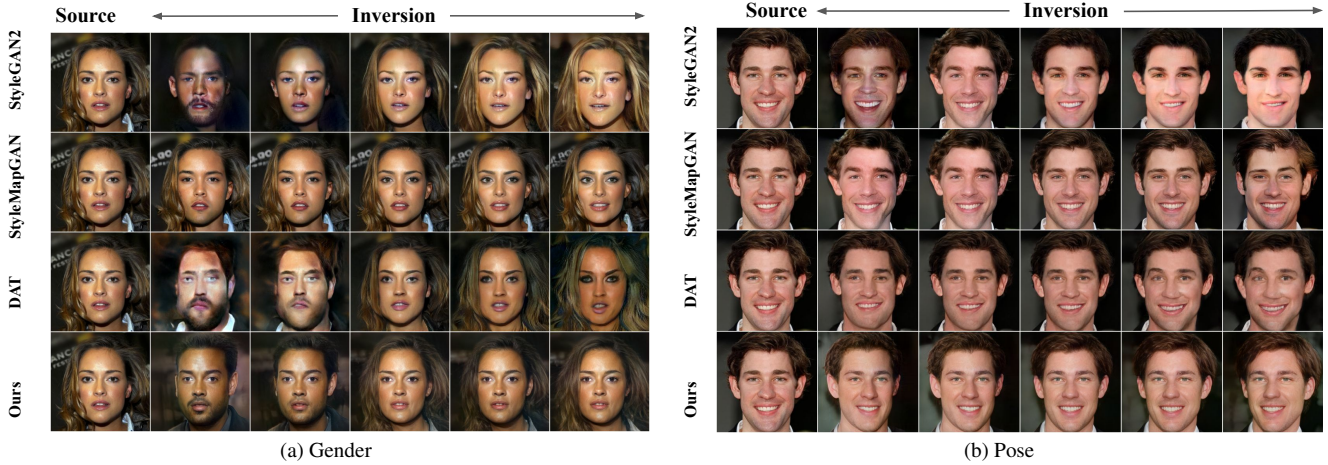
(a) Gender

(b) Pose

Figure 7. **Real Image Editing Comparison**. Images on the first column are the real source images. The fourth column shows the reconstruction results, which are semantically interpolated to the left and right sides. Our method preserves the facial identity better than other methods when editing gender (a) and pose (b).

Table 2. **Quantitative editing comparison between StyleGAN2 [23], StyleMapGAN [24], DAT [25] and Ours.** The row-column entry represents the degree of change of column attribute while editing the row attribute. For example, the pose row and gender column shows the change of gender when editing the head pose. The comparison shows our method has the least effect on other attributes during editing.

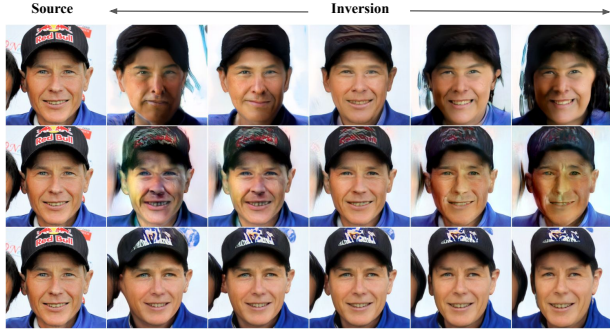|	Pose↓				Gender↓				Smile↓			
Method	StyleGAN2	StyleMap	DAT	Ours	StyleGAN2	StyleMap	DAT	Ours	StyleGAN2	StyleMap	DAT	Ours
Pose	-	-	-	-	0.683	3.418	4.869	**0.231**	0.577	0.947	0.625	**0.153**
Gender	0.738	0.340	0.386	**0.313**	-	-	-	-	0.189	1.804	0.514	**0.035**
Smile	0.0712	0.181	0.034	**0.031**	0.146	0.146	0.111	**0.040**	-	-	-	-

Figure 8. **Ablation Study on the Transformer and the dual-space design**. Rows from top to bottom showcase editing results of head pose using TransEditor w/o Transformer, TransEditor w/o dual spaces, and TransEditor itself, respectively.

(a) Z as Q matrix, P as K,V matrix (b) P as Q matrix, Z as K,V matrix

Figure 9. **Ablation study on different interaction methods using Transformer.** Both images are generated by fixing z code and sampling p code at each column. Figure(b) is our current setting, which shows a more disentangled result than figure(a).

5. Conclusion and Discussion

This paper introduces TransEditor, a novel dual-space GAN architecture with a Cross-Space Interaction mechanism based on the Transformer. Besides, we propose a new dual-space image editing and inversion strategy for highly controllable facial editing. Extensive experiments show the effectiveness of TransEditor in attribute disentanglement and controllability, surpassing state-of-the-art baselines in complicated attribute editing. The proposed TransEditor is readily applicable to many real-world applications such as photo retouching and face manipulation, which however, might be used unethically. Devising better media forensics approaches could be countermeasures. As for limitations, the editing process relies on auxiliary classifiers (for semantic boundaries), whose quality and diversity may limit editable attributes. In addition, the improvement on the cross-space interaction of a dual-space GAN for editing tasks can be interesting future work.

References

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

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, 2021.

[3] Badour Albahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *TOG*, 40, 2021.

[4] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *CVPR*, 2020.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

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

[7] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *CVPR*, 2020.

[8] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *arXiv preprint arXiv:2105.13290*, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[10] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32:2627–2636, 1998.

[11] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017.

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

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014.

[14] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020.

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *NeurIPS*, 2020.

[16] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *TIP*, 28:5464–5478, 2019.

[17] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019.

[18] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

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

[20] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021.

[21] 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.

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

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

[24] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *CVPR*, 2021.

[25] Gihyun Kwon and Jong Chul Ye. Diagonal attention and style-based gan for content-style disentanglement in image generation and translation. In *ICCV*, 2021.

[26] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Tryongan: Body-aware try-on via layered interpolation. *TOG*, 40, 2021.

[27] Heyi Li, Jinlong Liu, Yunzhi Bai, Huayan Wang, and Klaus Mueller. Transforming the latent space of stylegan for real face editing. *arXiv preprint arXiv:2105.14230*, 2021.

[28] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. *NeurIPS*, 2019.

[29] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *NeurIPS*, 33, 2020.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[31] 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. In *CVPR*, 2021.

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

[33] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *PAMI*, 2020.

[34] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021.

[35] Nurit Spingarn, Ron Banner, and Tomer Michaeli. Gan "steerability" without optimization. In *ICLR*, 2020.

[36] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM TOG*, 40:1–14, 2021.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[38] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021.

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

[40] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *ECCV*, 2018.

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

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

[43] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *TOG*, 2021.

[44] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*, 2020.