Conceptual and Hierarchical Latent Space Decomposition for Face Editing

Savas Ozkan Mete Ozay Tom Robinson
Samsung Research UK
savas.ozkan@samsung.com

Abstract

Generative Adversarial Networks (GANs) can produce photo-realistic results using an unconditional image-generation pipeline. However, the images generated by GANs (e.g., StyleGAN) are entangled in feature spaces, which makes it difficult to interpret and control the contents of images. In this paper, we present an encoder-decoder model that decomposes the entangled GAN space into a conceptual and hierarchical latent space in a self-supervised manner. The outputs of 3D morphable face models are leveraged to independently control image synthesis parameters like pose, expression, and illumination. For this purpose, a novel latent space decomposition pipeline is introduced using transformer networks and generative models. Later, this new space is used to optimize a transformer-based GAN space controller for face editing. In this work, a StyleGAN2 model for faces is utilized. Since our method manipulates only GAN features, the photo-realism of StyleGAN2 is fully preserved. The results demonstrate that our method qualitatively and quantitatively outperforms baselines in terms of identity preservation and editing precision.

1. Introduction

Generative Adversarial Networks (GANs) [15, 20, 12] are formulated as a two-step learning procedure via generator and discriminator models. This pipeline can produce photo-realistic images that are hard to distinguish. Especially, generative models such as StyleGAN2 [20] have become one of the most effective image synthesis tools. They are capable of generating high-resolution images using nonlinear features learned from low-dimensional feature spaces. In the end, coarse and fine details of synthesized images are simply derived from these spaces. However, existing GAN models do not offer intuitive control, i.e., not human understandable parameterization, for image generation. Recent works [16, 39, 32, 36] have shown that the outputs of GAN models can be edited by disentangling their features spaces without the need of employing full supervision or updating pre-trained model parameters.

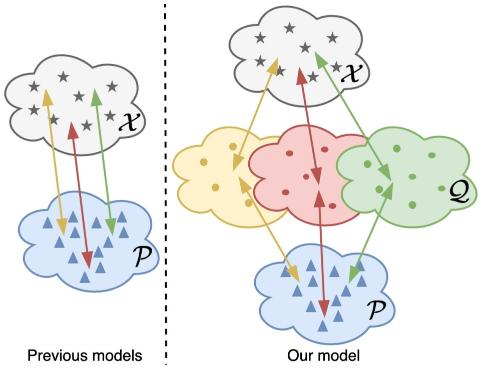

Figure 1: Rather than finding a direct mapping between a GAN space $\mathcal{X}$ of learned features (★), and a space $\mathcal{P}$ of parameters of face concepts (▲) [39], our method estimates an intermediate latent space $\mathcal{Q}$ whose latent codes (●) are conceptual and hierarchical. After all, a GAN space controller trained on this space can edit each concept independently while preserving the details at different abstraction levels of GANs for face editing.

There have been multiple efforts in the literature to achieve controllable content manipulation of features of GANs with supervised and unsupervised learning methods. In [17, 7], they use labelled data to expand user control onto a GAN space of features (depicted by $\mathcal{X}$ in Fig. 1). These labels are used to decouple the entangled representations with learnable projections. However, this control needs costly manual annotations that usually involve gathering many images and labels with clear definitions. Hence, performance is significantly impacted by the total number of annotations and the quality of labels. In contrast to the supervised methods, unsupervised approaches [34, 16, 28] achieve disentanglement by finding principal directions in feature spaces without relying on labeled data. In the end, each direction alters a different visual content so that a semi-automatic control is provided for the existing GAN models. Other studies [36, 22, 37] have suggested that manipulating the GAN spaces may be limited by biases present in pre-trained models. Hence, their goal is to train a new generator model from scratch using generative priors for controllable image synthesis. Although the models have robust control over the feature space, they are difficult and costly to train.

A setting [39] for face editing is to use concepts such as pose, light, expression, and likeness, derived from 3D Morphable face models [24, 30], as pseudo-labels. Each concept has a unique parameter set, and these parameters are used as pseudo-labels to disentangle a pre-trained GAN space. For this purpose, the authors propose a model (called Differentiable Face Reconstruction (DFR)) and fix the original GAN parameters. This model directly maps a GAN space $\mathcal{X}$ to a face parameter space $\mathcal{P}$ (i.e., morphable faces) as illustrated in Fig. 1. Later, with a differentiable face renderer, the DFR model is used to decouple the GAN space $\mathcal{X}$. To be specific, a face renderer is adapted to generate face images from single-dimensional parameters estimated by the DFR model. Thus, multi-resolution GAN features must be projected onto a concept parameter set. However, the drawback is that a face parameter space $\mathcal{P}$ with no hierarchical information is utilized to edit a GAN space $\mathcal{X}$. Thereby, the level of details in the GAN space is degraded and conceptual information in multi-resolution features may not be truly captured. Ultimately, the model fails to associate facial concepts with a multi-resolution GAN space.

Our paper proposes a novel method to manipulate GAN spaces for face editing. For this purpose, we propose an encoder-decoder model that estimates an intermediate latent space ($\mathcal{Q}$ in Fig. 1) while learning a mapping between a GAN space $\mathcal{X}$ and a face parameter space $\mathcal{P}$. Compared to baseline, employment of this new space is crucial, since it derives hierarchical and conceptual latent codes for further usages. In other words, face concepts are independently represented while hierarchical details at different abstraction levels of GANs are maintained. Later, these latent codes are used to learn controlling a GAN space for face editing. Pseudo-labels estimated by 3D morphable models are used, and our pipeline does not require to update the GAN models. Our approach has two main components:

(i) A transformer-based latent code decomposer that computes conceptual and hierarchical latent codes from a GAN space $\mathcal{X}$ of features, and the parameters of face concepts from $\mathcal{P}$. A novel encoder-decoder model is proposed to compute intermediate latent codes, with an encoder model based on transformer networks. These intermediate latent codes are then reprojected to the face parameter space $\mathcal{P}$ using a multi-resolution generative model. The proposed pipeline performs the decomposition in a self-supervised manner during the space mapping, relying solely on the reconstruction error at the output of our model.

(ii) A GAN space controller that manipulates GAN space $\mathcal{X}$ of features with face control parameters. To optimize the model, we enforce the consistency between the projected representations of the original and manipulated GAN features on the intermediate latent space $\mathcal{Q}$. The model is based on a transformer network that uses face control parameters to manipulate the GAN space $\mathcal{X}$.

Our contributions can be summarized as follows:

- We propose a novel encoder-decoder model that decomposes a multi-resolution GAN space $\mathcal{X}$ into an intermediate latent space $\mathcal{Q}$ in a self-supervised manner. The decomposition is performed by our novel NN-architecture, relying solely on the reconstruction error during the space mapping.
- The intermediate latent space $\mathcal{Q}$ is used to train a GAN space controller for face editing. To this end, we introduce a transformer-based network inspired by [18] where user-specific control inputs are first encoded to multi-resolution representations. These representations are then used to alter the GAN features.
- In the analyses, our method outperforms baselines in terms of identity preservation (achieving relative accuracy improvement by 33%) and editing precision under extreme pose, expression and illumination manipulation, both qualitatively and quantitatively.

2. Related Work

Image Synthesis with GANs: In GAN domain [15], recent advancements have led to significant improvements for image synthesis by focusing on designing architectures, loss functions, and training schemes [4, 19, 20, 12]. Among them, StyleGAN2 [20] is the most advanced GAN model that produces high-resolution images. However, GANs do not naturally promise explicit visual control over the image generation. Therefore, we will explore the techniques that manipulate image content particularly by focusing on faces.
Conditional GANs (CGANs): They [27, 6] have been widely employed for semantic face manipulation. Here, generator models are controlled by random noise and class labels/features that limit the content of synthesized images. Some studies focus on training auto-encoders using images as conditional information [17, 29, 23]. Later, this class of work is extended to train CGANs with unpaired data using cycle-consistency learning [25, 7]. However, these methods allow users to control a limited number of discrete classes/attributes [10]. Furthermore, the image resolution and quality are significantly limited compared to StyleGAN-based methods.
Embedding GANs: The aim is to project input images onto the feature space of a GAN before performing any content modifications. To be specific, encoder and generator models are trained together to map input images to a feature space and later synthesize manipulated images [8, 31, 47, 22, 37, 36]. The advantage is that images are directly embedded into a disentangled feature space. Other methods apply the inversion of a pre-trained generator for disentanglement in post-processing. These methods optimize GAN features with a different encoder by minimizing

the reconstruction error [1, 42, 44, 32, 40]. Style Transformer [18] employs a transformer network to update the GAN features by learning an encoder model.

Manipulating GAN Features: A distinct field of study for image manipulation is to project the features of a pretrained GAN into controllable sub-groups. These models are practical and easily to train. Note that our method must be considered under this category of image manipulation. Earlier works explore to obtain meaningful space directions with linear operations [34, 16, 35, 28]. However, linear techniques are not enough to disentangle the nonlinear components in feature spaces [43]. Later, other studies attempt to incorporate nonlinear models using facial attributes [2, 45, 44]. The drawback of these methods is that the contributions of each GAN feature at various abstraction levels for different face concepts are manually determined by the authors. StyleRig [39] utilizes 3D morphable face models to control face rigging information such as pose, expression, illumination, and likeness. Here, an important property of this method is that it aims to learn the multi-resolution characteristics automatically.

3. Proposed Method

In this section, we describe our full pipeline for face editing. In our setup, the parameters of original GAN models are fixed, and only their features are manipulated.

3.1. Overview

GAN space $\mathcal{X}$: In GANs, a feature $\mathbf{z}$ is sampled from a probability distribution $p(\mathbf{z})$. Later, a NN model $\sigma(\cdot)$ is utilized to synthesize an image $\mathbf{I_z} \in \mathbb{R}^{3\times h\times w}$ from the feature by $\mathbf{I_z} = \sigma(\mathbf{z})$. In several GANs, such as StyleGAN, a multi-resolution feature obtained from various abstraction levels $\mathbf{x} = [\mathbf{x}_i \in \mathbb{R}^d]_{i=1}^N$ is computed with nonlinear fully-connected networks by $\mathbf{x} = V(\mathbf{z})$. To this end, both theoretical and algorithmic improvements [19, 20] enable to generate high-quality and high-resolution images (configuration of StyleGAN is $w = h = 1024$, $N = 18$ and $d = 512$).

In StyleGAN [19], utilization of "style mixing" regularization during training aims to find a multi-resolution GAN space $\mathcal{X}$ where the feature $\mathbf{x}_i \in \mathcal{X}$ at the i^{th} abstraction level enforces to scatter its own details to the overall image content. In other words, image content can be manipulated by simply swapping or editing GAN features at different abstraction levels. However, since these features are entangled, this manipulation operation is not controllable.

Parameter space $\mathcal{P}$: In 3D morphable face models (e.g., FLAME [24], FaceWarehouse [5] or Basel [30]), an input face can be represented by a set of concepts $\mathbf{p} = (\beta, \psi, \theta, \gamma, \mathbf{R}, \mathbf{t}) \in \mathbb{R}^l$. Here, each term denotes a face concept such as the facial shape β, the skin texture/reflectance ψ, the facial expression θ, the scene illumination γ, the head pose rotation $\mathbf{R}$ and translation $\mathbf{t}$. In these models, each concept can be independently controllable. Thereby, various face combinations $\mathbf{I_p} = R(\mathbf{p})$ can be derived (with a differentiable renderer $R(\cdot)$) by sampling a parameter tuple $\mathbf{p}$ from a face parameter space $\mathcal{P} = \bigcup_{i=1}^M \mathbb{P}_i$. Here, M denotes the number of concepts and $\mathbb{P}_i$ denotes the i^{th} sub-concept space. Beyond this, efforts are also made towards estimating these parameters [9, 13] from real images $\mathbf{I_f}$ using an encoder model $E(\cdot)$ by $\mathbf{p_f} = E(\mathbf{I_f})$.

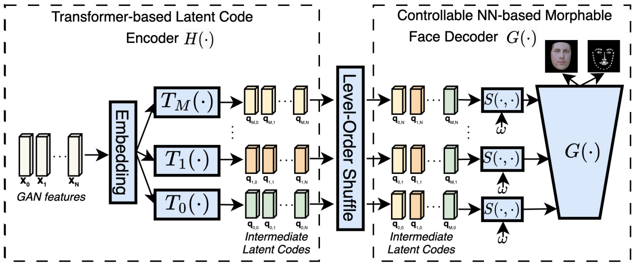

Figure 2: The overall architecture of Transformer-based Latent Space Decomposer. Transformer-based Latent Code Encoder $H(\cdot)$ (left) and Controllable NN-based Morphable Face Decoder $G(\cdot)$ (right) are used to estimate the intermediate latent codes $\mathbf{q}$ in a self-supervised manner. Details are given in the Subsection 3.2

In a self-supervised learning setup, our method first learns an intermediate latent space $\mathcal{Q}$ that represents both hierarchical and conceptual information existing in $\mathcal{X}$ and $\mathcal{P}$, respectively. Then, this space is used to obtain an optimum GAN space controller for face editing.

3.2. Transformer-Based Latent Space Decomposer

Our baseline method [39] computes a model f that maps a GAN space $\mathcal{X}$ onto a face parameter space $\mathcal{P}$ by $f : \mathbf{x} \in \mathcal{X} \mapsto \mathbf{p} \in \mathcal{P}$. Later, the space $\mathcal{P}$ is used to optimize a GAN space controller (i.e., an encoder-decoder model) for face editing. However, the space $\mathcal{P}$ encapsulates only the conceptual information so that the hierarchical information in $\mathcal{X}$ is eventually degraded during manipulation.

Therefore, our objective is first to find an intermediate latent space $\mathcal{Q}$ in the course of transforming the space $\mathcal{X}$ to the space $\mathcal{P}$ with a set of models by $\phi : \mathcal{X} \to \mathcal{Q} \to \mathcal{P}$. After all, instead of the space $\mathcal{P}$, intermediate latent space $\mathcal{Q}$ is employed for the optimization of the controller model. This space essentially must comprise two properties to overcome the weakness of our baseline:

(I) *Employment of hierarchical representations:* At each abstraction level, representations with different levels of detail are learned and encoded in GAN spaces. In order to preserve these varying details and information granularity for face editing, they must be projected to a hierarchical intermediate latent space in training.

(II) *Mutually exclusive and conceptually consistent representations:* In order to control generating faces for different concepts, each concept must be represented by a unique sub-space with distinct and diverse characteristics. Therefore, the representations of concepts must be mutually independent.

To achieve these properties while learning the intermediate latent space $\mathcal{Q}$, we propose a latent space decomposition method that is formulated by encoder-decoder mapping functions $h : \mathcal{X} \rightarrow \mathcal{Q}$ and $g : \mathcal{Q} \rightarrow \mathcal{P}$ which decompose $\phi(\mathbf{x}) = g \circ h(\mathbf{x})$. Briefly, $h(\cdot)$ denotes a transformer-based latent space encoder, and $g(\cdot)$ represents a controllable NN-based morphable face decoder. Fig. 2 visualizes these two models and the input-output relations.

Transformer-based Latent Space Encoder (TLSE): Given a multi-resolution feature $\mathbf{x}$, a function is learned to decompose it into conceptual (M dimensional) and hierarchical (N dimensional) codes by $\mathbf{Q} = H(\mathbf{x})$ where $\mathbf{Q} = [\mathbf{Q}_{i,j} \in \mathbb{R}^k]_{i,j=1}^{M,N}$. These decomposed codes should capture information from both face parameter and GAN spaces. In other words, they must be conceptual and hierarchical. These codes are then used to train a GAN space controller for face editing by measuring the hierarchical and conceptual consistencies between original and manipulated latent codes. Details are explained in the next subsection.

In practice, our model first projects each GAN feature obtained at the i^{th} abstraction level $\mathbf{x}_i$ to M different codes $[\mathbf{e}_{j,i}]_{j=1}^{M}$ using multiple linear layers by $\mathbf{e}_{j,i} = \mathbf{W}_j \mathbf{x}_i$. As a result, each code $\mathbf{e}_{j,i}$ represents an embedding of one of the face concepts. Later, we arrange these embeddings to create a code sequence $[\mathbf{e}_{j,i}]_{i=1}^{N}$ for each concept (i.e., Level-Order Shuffle in Fig. 2) and feed them to multiple transformer networks. To this end, we train M different transformer blocks for each concept $[T_i(.)]_{i=1}^{M}$. Fig. 2 illustrates the encoder. In our model, transformers are particularly selected for their ability to unveil the hierarchical dependencies among inputs (property (I)). Furthermore, the codes are deliberately separated for each concept to achieve independent control for the GAN space (property (II)). The distribution of representations belonging to the intermediate latent space $\mathcal{Q}$ is visualized with t-SNE [41] in Fig. 3. It indicates that the space $\mathcal{Q}$ contains disentangled representations of different concepts.

Controllable NN-based Morphable Face Decoder (CMFD): Unsupervised learning of the intermediate latent space $\mathcal{Q}$ is a challenging task. To be specific, there is no labeled data obtained from the intermediate latent space (neither supervised nor pseudo labels). Therefore, one of the property of 3D morphable face models must be leveraged with multi-resolution generative models where the generator produces a single image output using a multi-resolution representation. Here, we use 3D morphable face models in a flexible data generation pipeline for our generative model, since multiple face renderings can be generated for the same person by preserving or discarding some of the facial concepts.

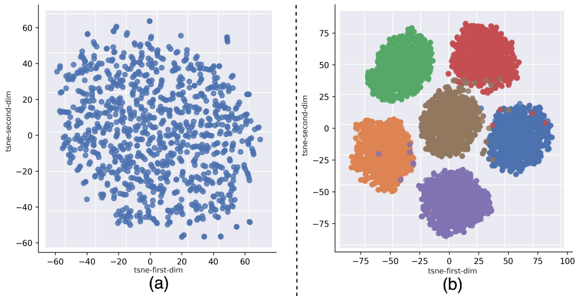

Figure 3: (a) GAN space $\mathcal{X}$ and (b) intermediate latent space $\mathcal{Q}$ are visualized with t-SNE. Intermediate latent space $\mathcal{Q}$ contains disentangled representations that cover multiple and independent sub-spaces (each member of each space is depicted with a different color) by $\mathcal{Q} = \bigcup_{i=1}^{M} \mathbb{Q}_i$ ($M = 6$ in (b)), while the GAN space $\mathcal{X}$ is entangled.

For this purpose, we propose a controllable NN-based morphable face encoder model and a scheme to train the model. This model aims to find a projection between the *hierarchical* intermediate latent space $\mathcal{Q}$ and *conceptual* face parameter space $\mathcal{P}$, and train the parameters of the TLSE. Hence, it is a function that synthesizes a rendered morphable face $\mathbf{I}_\mathbf{Q} \in \mathbb{R}^{3\times h\times w}$ and facial landmarks $\mathbf{L}_\mathbf{Q} \in \mathbb{R}^{1\times h\times w}$ projected onto a 2D binary image using intermediate latent codes $\mathbf{Q}$ by $[\mathbf{I}_\mathbf{Q}, \mathbf{L}_\mathbf{Q}] = G(\mathbf{Q})$. In essence, this model serves as a differentiable renderer, but it receives multi-resolution latent codes. Here, the rendered images also do not need to be high-resolution (i.e., $h << 1024$, $w << 1024$, and we use $h = w = 64$ in our experiments). The architecture of our generative model is based on [3], where pixel coordinates are encoded with latent Fourier blocks. This provides two advantages: 1) it encodes the geometric properties better, and 2) it represents the higher frequency content better, as discussed in [38]. Indeed, we require a differentiable renderer model (that is, the renderer in 3D morphable face models is used) $R : \mathbb{R}^l \rightarrow \mathbb{R}^{3\times h\times w}$ to generate ground-truth renders and optimize our models. Note that its parameters are also fixed.

Furthermore, we introduce a stochastic sampler $S(\cdot,\cdot)$ to be used for both intermediate latent codes $\tilde{\mathbf{Q}} = S(\mathbf{Q}, \omega)$ and face concept parameters $\tilde{\mathbf{p}} = S(\mathbf{p}, \omega)$. Here, this function stochastically activates (preserves codes) or deactivates (codes are set to zero) some of the concepts with a random binary mask $\omega \in \mathbb{Z}^M$ at each training iteration. Specifically, this regularization method provides a better conceptual decomposition for intermediate latent codes by randomly combining/dropping some of face concepts at each iteration. For the sake of simplicity of the notation, we will denote $\mathbf{p}$ and $\mathbf{Q}$ instead of $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{Q}}$ from now on. The decoder model is visually summarized in Fig. 2. We then train both H and G models minimizing the following loss $\mathcal{L}_{all}$;

$$\mathcal{L}_{all} = \mathcal{L}_{photo}(\mathbf{I}_\mathbf{Q}, \mathbf{p}) + \mathcal{L}_{land}(\mathbf{L}_\mathbf{Q}, \mathbf{p})$$
$$+\lambda_{gan}\mathcal{L}_{gan}(\mathbf{I}_\mathbf{Q}, \mathbf{p}) + \lambda_{orth}\mathcal{L}_{orth}(\mathbf{W}),$$

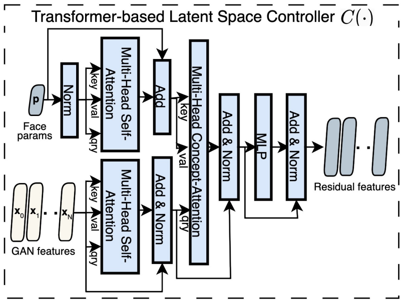
Transformer-based Latent Space Controller $C(\cdot)$
p
Face params
Norm
key
val
qry
Multi-Head Self-Attention
Add
key
val
Multi-Head Concept-Attention
qry
Add & Norm
MLP
Add & Norm
Residual features
x_0 x_1 x_N
GAN features
key
val
qry
Multi-Head Self-Attention
Add & Norm

Figure 4: The overall architecture of Transformer-based Latent Space Controller $C(\cdot)$. It is used for manipulating input GAN features $\mathbf{x}$ with face parameters $\mathbf{p}$. Details are given in the Subsection 3.3.

where λ_{gan} and λ_{orth} are regularization parameters. The pixel-wise photometric loss is defined as the Euclidean distance ($||\cdot||_2$) between face images rendered by our generative model and differentiable renderer model by

$$\mathcal{L}_{photo}(\mathbf{I_Q}, \mathbf{p}) = ||\mathbf{I_Q} - R(\mathbf{p})||_2, \quad (1)$$

Landmark loss is also adapted to enhance face renders for the concepts related to face meshes like expression and head pose using a pixel-wise binary-cross entropy loss by

$$\mathcal{L}_{land}(\mathbf{L_Q}, \mathbf{p}) = \mathbf{L_r}\log(\mathbf{L_Q}) + (1 - \mathbf{L_r})\log(1 - \mathbf{L_Q}), \quad (2)$$

where $\log(\cdot)$ is the element-wise logarithm. $\mathbf{L_r} \in \mathbb{R}^{1\times h\times w}$ denotes the landmark positions that are computed using the concept parameters $\mathbf{p}$ and then projected onto a 2D binary image with a differentiable face renderer. To find the association of the rendered images and landmark positions in the intermediate latent space, we introduce a GAN-based penalty term inspired by [26]:

$$\mathcal{L}_{gan}(\mathbf{I_Q}, \mathbf{p}) = \log(1 - D(\mathbf{I_Q}, \mathbf{L_r})), \quad (3)$$

where, $D(\cdot, \cdot)$ denotes a conditional discriminative network. To ensure the conceptual independence for the intermediate codes, an orthogonal regularization term is incorporated by

$$\mathcal{L}_{orth}(\mathbf{W}) = ||\mathbf{W}\mathbf{W}^{\mathrm{T}} - \mathbb{I}||_1. \quad (4)$$

where $\mathbb{I}$ denotes the identity matrix and $||\cdot||_1$ is the ℓ_1 norm.

3.3. Latent Space Face Editing with Transformers

For each multi-resolution GAN feature $\mathbf{x}$, a synthetic image can be generated using a pre-trained GAN model by $\mathbf{I_x} = \sigma(\mathbf{x})$ (we implement σ using StyleGAN2 [20] in the analyses). Our goal is to edit a GAN feature $\mathbf{x}$ using a control parameter $\mathbf{p_e}$ and a trainable controller $C(\cdot, \cdot)$ by

$$\mathbf{x_e} = \mathbf{x} + \Delta\mathbf{x_e}, \quad (5)$$

where $\mathbf{x_e}$ and $\Delta\mathbf{x_e} = C(\mathbf{x}, \mathbf{p_e})$ denote the edited GAN feature and residual feature, respectively. An image for the edited feature can be also generated by $\mathbf{I_{x_e}} = \sigma(\mathbf{x_e})$.

For this purpose, we propose a transformer-based GAN space controller $C(\mathbf{x}, \mathbf{p_e})$ inspired by [18]. Since the control face parameter $\mathbf{p_e}$ is a single variable, we first decompose the vector representation into level-wise representations with a multi-head self-attention. Next, we design a multi-head concept attention that intuitively manipulates the GAN features with the face control parameters. Compared to self-attention models, the query is projected from GAN features $\mathbf{x}$, while key and value are computed from control face parameters $\mathbf{p_e}$. The transformer architecture is illustrated in Fig. 4. Compared to baseline [39] (i.e., fully-connected encoder-decoder), this model can capture hierarchical dependencies at multiple abstraction levels during the manipulation step. As a result, the model can learn these dependencies and can edit the coarse, medium, and fine details in the GAN space using control parameters.

Optimization: Given the GAN feature and face parameter pairs $(\mathbf{x_1}, \mathbf{p_{x_1}})$ and $(\mathbf{x_2}, \mathbf{p_{x_2}})$, we optimize the controller without using ground-truth labels. During training, we sample a random binary mask $\omega \in \{0, 1\}^M$ at each iteration. A new control parameter $\mathbf{p_e}$ is calculated as a linear mixture of $\mathbf{p_{x_1}}$ and $\mathbf{p_{x_2}}$ by $\mathbf{p_e} = \omega\mathbf{p_{x_1}} + (1 - \omega)\mathbf{p_{x_2}}$. Next, the feature $\mathbf{x_1}$ is edited by the parameters $\mathbf{p_e}$ to estimate $\mathbf{x_{1\to e}}$. We train the controller $C(\cdot, \cdot)$ by minimizing the conceptual dissimilarity between the original and manipulated GAN features with the loss

$$\begin{aligned}\mathcal{L}_{consist} = &||S(H(\mathbf{x_1}), \omega) - S(H(\mathbf{x_{1\to e}}), \omega)||_1\\ &+||S(H(\mathbf{x_2}), 1 - \omega) - S(H(\mathbf{x_{1\to e}}), 1 - \omega)||_1\\ &+\lambda_{edit}||\mathbf{x_1} - \mathbf{x_{1\to e\to 1}}||_2 + \lambda_{sparse}||\Delta\mathbf{x_e}||_1,\end{aligned}$$

where, $S(\cdot, \cdot)$ and $H(\cdot)$ denote the stochastic sampler and a pre-trained TLSE model presented previously. λ_{edit} is the coefficient that scales effects of cycle-consistency loss, and λ_{sparse} is used to control the sparsity of disentanglement.

Multi-Task Learning: The consistency loss used in our model can be separately calculated for each concept. To be specific, the mean absolute error between the outputs of the pre-trained TLSE $H(\cdot)$ employed using the original and edited codes can be computed for each concept. Thereby, we formulate the optimization step in a multi-task learning setting to prevent overfitting to a particular concept during training. An uncertainty-based multi-task learning method [21] is utilized to better learn shared representations by scaling the loss objectives for each concept, and by mitigating the sensitivity for weight selection.

4. Experimental Analyses

Implementation Details: We train our models using the LAMB optimizer [46] with a learning rate of 0.0032, a batch size of 1024 and an iteration of 40K. For the transformers implementing the model $H(\cdot)$, we use the architecture presented in [11] (2 layers with 8 multi-heads). For the transformer implementing $C(\cdot)$, we used 4 layers with 8 multi-heads. Training time for our pipeline takes approximately 1 hour with multiple GPUs. Empirically, the dimension of intermediate codes k is set to 16. In addition, λ_{gan}, λ_{orth}, λ_{edit} and λ_{sparse} coefficients are set to 0.01, $1e-4$, 0.01 and $1e-4$, respectively.

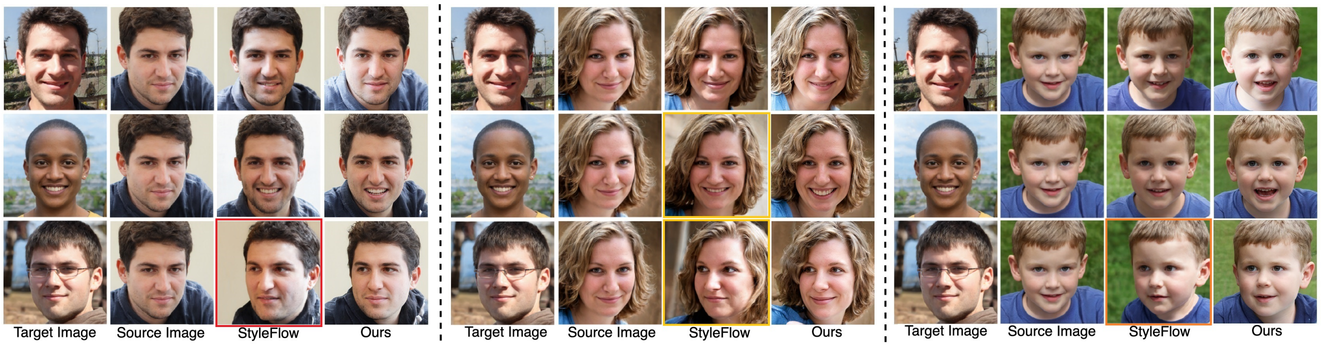

Figure 5: Samples for transferring pose, expression, and illumination simultaneously from target to source images.

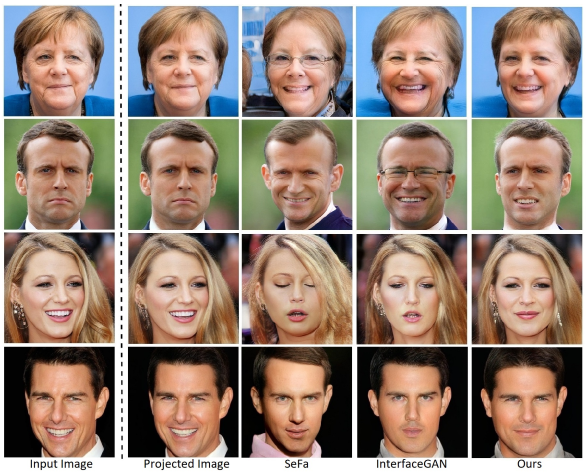

Figure 6: Results on real faces.

Dataset: We use a pre-trained StyleGAN2 model [20] trained on FFHQ dataset in all our experiments. For this purpose, our pipeline requires StyleGAN2 features and corresponding 3D morphable face parameters for training. Therefore, we randomly sample 200K features from StyleGAN2 space where 190K parameters are reserved for training and the rest is used for testing. To increase the diversity of facial patterns, up to 5 separate features are combined to produce one feature, leveraging the idea of style mixing. To implement a 3D morphable model and estimate the face parameters, a NN-method [9] based on the Basel Face model [30] is used. Note that we tested our approach with FLAME face model [24] computed by DECA [13], and still, equivalent results are obtained. To compare our method with baselines, we also use a synthetic dataset released by StyleFlow [2] for qualitative and quantitative evaluations.

Baselines: As baselines, we compare our method with five works: InterfaceGAN (IG) [34], GANSpace (GS) [16], SeFa [35], StyleRig (SR) [39] and StyleFlow (SF) [2]. For GS, SeFa and SF, we used the code provided by the authors. For IG, we used the previously reported results and the code provided in [33] for limited concepts. Lastly, we implemented SR model from scratch based on their paper.

Metrics: We evaluate our results using face identity, edit consistency and edit precision scores on the StyleFlow dataset. To reproduce the same results [2], the embeddings of a face model [14] on the original and edited images are used to calculate the cosine similarity for face identity scores. Furthermore, we report the edit permutation consistency proposed in [2] only for the light parameter with DPR model [48] (for other concepts, attribute model is not open-sourced, so we couldn't report any score). Lastly, we randomly swap the facial concepts between source and target image parameters to transfer expression, pose and illumination concepts. We calculate the mean error between the edited and target images using Basel outputs for edit scores.

4.1. Qualitative and Quantitative Comparisons

To demonstrate our ability to produce high-quality disentangled image edits, we conduct tests by transferring face parameters (i.e., pose, expression, and illumination) from target images to source images. Results are illustrated in Fig. 5. The results indicate that our method can successfully handle extreme changes in pose, expression, and illumination. Notably, our approach preserves other attributes such as background, clothing, and hair color from the source images. This is achieved by utilizing mutually independent latent codes during optimization, which ensures that other concepts remain unaffected in GAN space manipulation.

We also compare our results with the reported SF results. However, the baseline model cannot adequately preserve face identity for extreme pose and illumination changes (as seen in the left-column face set - last row - red box). Furthermore, there are significant pose and expression misalignments between the target and SF-edited images (yellow boxes). Lastly, altering face concepts with SF leads to undesired changes on face attributes such as age (as seen in the right-column face set, last row - orange box). Note that SR results are not included in this discussion since previous work [2] has already indicated that the model does not perform well when all face concepts are simultaneously applied. However, our supplementary material provides additional analyses for IG, SeFa, SR and SF.

In practice, editing real faces is essential for the final application. Hence, we report visual results on real faces in Fig. 6. Real faces are first projected to StyleGAN2 space

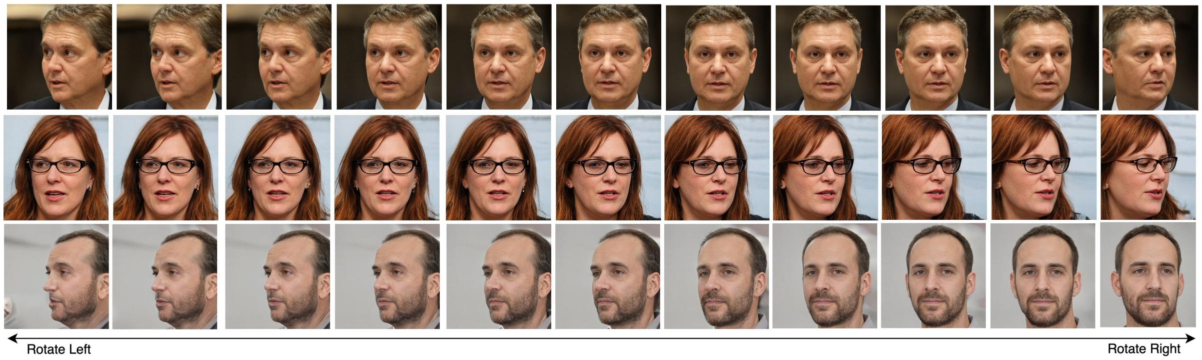

Figure 7: Only pose manipulation. Face identity and other features like background and facial attributes are largely preserved.

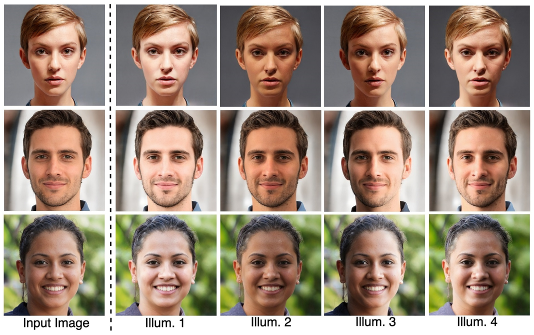

Figure 8: Only illumination is manipulated.

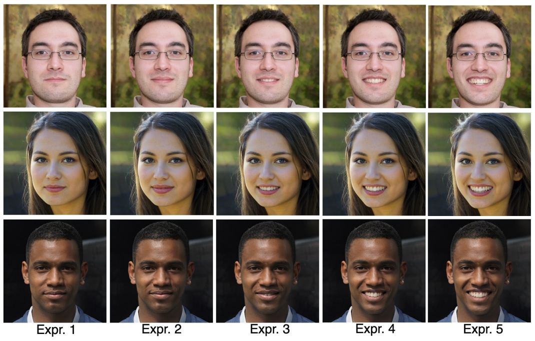

Figure 9: Only expression is manipulated.

Edit	IG [34]	GS [16]	SR [39]	SF [2]	Ours
illum	0.945	0.942	0.954	0.963	**0.981**
pose	0.940	0.939	0.959	0.966	**0.986**
expr	0.946	0.973	0.975	0.967	**0.983**
all	0.895	0.902	0.923	0.941	**0.963**

Table 1: Face identity evaluation (cosine similarity) to compare different methods.

Edit	IG [34]	GS [16]	SF [2]	Ours
illum	0.66	0.58	0.53	**0.35**

Table 2: Edit consistency evaluation (mean absolute error) to compare different methods.

Edit	GS [16]	SR [39]	Ours
illum	0.48	0.46	**0.41**
pose	0.20	0.17	**0.11**
expression	3.74	3.67	**3.49**

Table 3: Face concept edit precision evaluation (mean absolute error) to compare different methods.

using [33]. We compare our method with IG model, whose implementation is shared with the same model that only allows to edit pose and expression concepts, and SeFa model. The results demonstrate that our method gives significantly better results compared to IG and SeFa in terms of reducing distortion (first row), preserving attributes and identities (first, second and fourth rows), and producing realistic expressions (third row).

To support our claims and visual results, we evaluate our method with the quantitative analyses. Tab. 1 presents face identity similarity scores (higher is better) calculated by modifying illumination (illum), pose, expression (expr) and all three concepts simultaneously (all). Our method yields significantly better results in terms of identity scores compared to baselines. The results show that our method only manipulates the targeted part(s) without changing identities. This property is further confirmed in Tab. 2, where we measure the cycle edit consistency between sequentially transferred light-expression and pose-light concepts for the same person, as proposed in [2]. The error is significantly reduced due to our conceptually independent latent space. Lastly, we evaluate our method using Basel model outputs in Tab. 3 which reports the mean absolute error. Our method again demonstrates superior performance for the manipulation of illumination, expression and pose parameters.

4.2. An Ablation Study for the Individual Concepts

In this section, we qualitatively demonstrate the superiority of our method for handling individual pose, illumination, and expression changes. The overall results indicate that each of these concepts can be manipulated without conceptually interfering with the others. Fig. 7 illustrates the edited faces by rotating them in both left and right directions. Our method preserves face identity and attributes, even with extreme pose variations. However, our method is constrained by the biases of StyleGAN2 model. As seen in the last row of the figure, the rotation is restricted by the

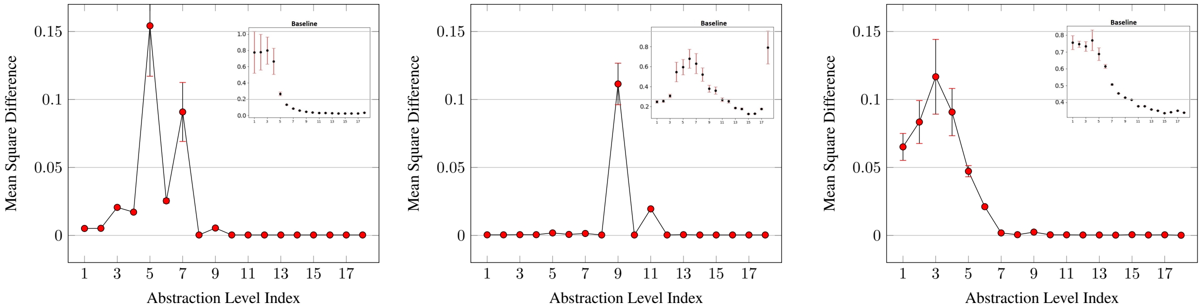

Figure 10: Change of StyleGAN2 features $\mathbf{x}$ at different abstraction levels after expression (left), illumination (middle) and pose (right) manipulation. Baseline results [39] are also added for each concept.

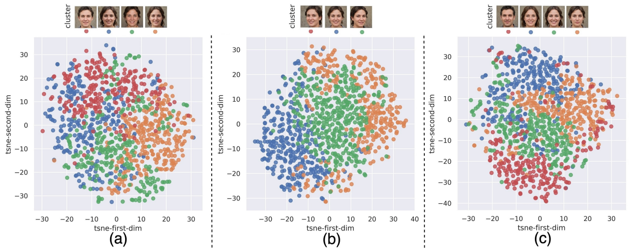

Figure 11: In plots (a) (illumination), (b) (pose) and (c) (expression), each sub-space $\mathbb{Q}_i$ of intermediate latent space $\mathcal{Q}$ is individually visualized to demonstrate that the distribution of codes in each sub-space is conceptually consistent with that of the sub-spaces in the face parameter space P.

GAN model. Despite this limitation, our results are photo-realistic, and the identity and attributes are still preserved. Fig. 8 and 9 show the results for the exclusive illumination and expression edits. The results demonstrate that only the contents related to targeted concepts are edited. Furthermore, the results remain conceptually consistent for the same face parameters (i.e., each column uses the same control parameters). The results reported for illumination also show that our model can implicitly unveil the facial structure in the latent space.

4.3. Analyses of Intermediate Latent Spaces

In this section, we analyze the intermediate latent space that is learned by our method. First, we inspect the property of *employment of hierarchical representations* (I). For this purpose, we visualize how our method edits the StyleGAN features $\mathbf{x}$ for various face concepts in Fig. 10. The plots illustrate the mean square difference and variance between the input and edited StyleGAN features at different abstraction levels. We use 10000 StyleGAN2-generated features, and the face parameters randomly swap for the manipulation. Notably, these changes for each concept are sparse compared to our baseline method [39] (Their plots are also added.). This means that our approach selectively alters only the targeted parts of the multi-resolution input features. In other words, hierarchical learning is achieved by not disturbing unrelated details in multi-resolution GAN space. We also compare the indices for each concept which are manually determined by [2]. These indices overlap with our results as well.

Second, we analyze the property of *mutually exclusive and conceptually consistent representations* (II). Fig. 3 visualizes t-SNE representations computed on intermediate latent codes $\mathbf{Q}$. Plots show the distribution of the latent codes. We can see that these latent codes are well-separated (i.e., mutually exclusive). This indicates that the intermediate latent space $\mathcal{Q} = \bigcup_{i=1}^{M} \mathbb{Q}_i$ is spanned by multiple and independent sub-spaces. Next, we inspect each conceptual sub-space $\mathbb{Q}_i$ individually. For this purpose, we use the corresponding face parameter labels (either γ, $\mathbf{R}$ or θ of $\mathbf{p}$) for intermediate latent codes. Since these parameters are continuous, we first cluster each sub-space of face parameters $\mathbb{P}_i$ and use the representative cluster centers of each concept as its sub-classes in our plots. To enhance the interpretibility, we also visualize these cluster centers for each concept using StyleGAN2 model. Plots for illumination (a), pose (b) and expression (c) illustrate the space distribution of each conceptual sub-space $\mathbb{Q}_i$ by coloring them with the corresponding sub-classes of each face concept. The results show that our method obtains a conceptually consistent latent space during self-supervised learning.

5. Conclusion

We propose a method to edit a pretrained GAN space for face editing. Our model differs from previous techniques that an intermediate latent space is estimated by an encoder-decoder model whose latent codes can control manipulation of conceptual and hierarchical information. The proposed pipeline performs this decomposition by relying solely on the reconstruction error during the mapping between the GAN space and face parameter space. Later, this intermediate space is used to optimize a GAN space controller for face editing. As a result, conceptual controllability of our method for illumination, pose and expression is enhanced while the photo-realism of StyleGAN2 is preserved. Both qualitative and quantitative results indicate the superiority of our method over baselines.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8296–8305, 2020. 3

[2] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics (ToG), 40(3):1–21, 2021. 3, 6, 7, 8

[3] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14278–14287, 2021. 4

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018. 2

[5] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics, 20(3):413–425, 2013. 3

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems, 29, 2016. 2

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8188–8197, 2020. 1, 2

[8] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5154–5163, 2020. 2

[9] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 0–0, 2019. 3, 6

[10] Xin Ding, Yongwei Wang, Zuheng Xu, William J Welch, and Z Jane Wang. Ccgan: Continuous conditional generative adversarial networks for image generation. In International conference on learning representations, 2021. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 5

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12873–12883, 2021. 1, 2

[13] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. ACM Transactions on Graphics (ToG), 40(4):1–13, 2021. 3, 6

[14] Adam Geitgey. Github-face recognition, 2020. 6

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, 63(11):139–144, 2020. 1, 2

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. Advances in Neural Information Processing Systems, 33:9841–9850, 2020. 1, 3, 6, 7

[17] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. IEEE transactions on image processing, 28(11):5464–5478, 2019. 1, 2

[18] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11337–11346, 2022. 2, 3, 5

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. 2, 3

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8110–8119, 2020. 1, 2, 3, 5, 6

[21] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7482–7491, 2018. 5

[22] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 852–861, 2021. 1, 2

[23] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5549–5558, 2020. 2

[24] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. ACM Trans. Graph., 36(6):194–1, 2017. 2, 3, 6

[25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. Advances in neural information processing systems, 30, 2017. 2

[26] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408, 2016. 5

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

[28] Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. Advances in Neural Information Processing Systems, 34:13497–13510, 2021. 1, 3

[29] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In ACM SIGGRAPH 2019 Real-Time Live!, pages 1–1. 2019. 2

[30] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In 2009 sixth IEEE international conference on advanced video and signal based surveillance, pages 296–301. Ieee, 2009. 2, 3, 6

[31] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14104–14113, 2020. 2

[32] 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 Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2287–2296, 2021. 1, 3

[33] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. ACM Transactions on Graphics (TOG), 42(1):1–13, 2022. 6, 7

[34] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9243–9252, 2020. 1, 3, 6, 7

[35] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1532–1540, 2021. 3, 6

[36] Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11254–11264, 2022. 1, 2

[37] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. Gan-control: Explicitly controllable gans. In Proceedings of the IEEE/CVF international conference on computer vision, pages 14083–14093, 2021. 1, 2

[38] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems, 33:7537–7547, 2020. 4

[39] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6142–6151, 2020. 1, 2, 3, 5, 6, 7, 8

[40] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG), 40(4):1–14, 2021. 3

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008. 4

[42] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11379–11388, 2022. 3

[43] Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6721–6730, 2021. 3

[44] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In Proceedings of the IEEE/CVF international conference on computer vision, pages 13789–13798, 2021. 3

[45] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. Learning non-linear disentangled editing for stylegan. In 2021 IEEE International Conference on Image Processing (ICIP), pages 2418–2422. IEEE, 2021. 3

[46] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962, 2019. 5

[47] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14263–14272, 2021. 2

[48] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7194–7202, 2019. 6