Text2Face: Text-based Face Generation with Geometry and Appearance Control

Zhaoyang Zhang[2, 3], Junliang Chen[4], Hongbo Fu[5], Jianjun Zhao[4], Shu-Yu Chen[1, 2], and Lin Gao[1, 2, *]

[1]*Institute of Computing Technology, Chinese Academy of Sciences*
[2]*University of Chinese Academy of Sciences*
[3]*Yale University*
[4]*Beijing Film Academy*
[5]*City University of Hong Kong*
[*]*Corresponding Author*

Abstract

Recent years have witnessed the emergence of various techniques proposed for text-based human face generation and manipulation. Such methods, targeting at bridging the semantic gap between text and visual contents, provide users with a deft hand to turn ideas into visuals via the interface of text and enable more diversified multimedia applications. However, due to the flexibility of linguistic expressiveness, the mapping from sentences to desired facial images is clearly many-to-many, causing ambiguities during text-to-face generation. To alleviate these ambiguities, we introduce a local-to-global framework with two graph neural networks (one for geometry and the other for appearance) embedded in to model the inter-dependency among facial parts. This is based upon our key observation that the geometry and appearance attributes among different facial components are not mutually independent, *i.e.*, the combinations of part-level facial features are not arbitrary and thus do not conform to a uniform distribution. By learning from the dataset distribution and enabling recommendations given partial descriptions of human faces, these networks are extremely suitable for our text-to-face task. Our method is capable of generating high-quality attribute-conditioned facial images from text. Extensive experiments have confirmed the superiority and usability of our method over the prior art.

Keywords: Image Generation, Text-based Interaction, Human Faces

1. Introduction

How does a certain character in a novel look like visually? This is a common question raised by readers when they are immersed into the content of a novel and wonder more details behind the text. Can we reconstruct the faces in the novel simply from textual descriptions [29] Although it sounds impossible in the past, this depiction-to-visualization procedure has the potential to become reality now, enabled by the fascinating progresses of human face generation and manipulation methods as well as natural language processing techniques. Inspired by this, in this work, we aim at visualizing such depiction by building an interface for converting the textual descriptions to human faces, and introduce a recommendation mechanism for proposing coherent faces given partial descriptions.

Efforts have been devoted to the field of text-based image generation in previous years, but not until recently do such methods begin to apply to facial images. Thanks to the visual-linguistic joint representation ability of CLIP [24], a series of works (*e.g.*, [21, 40]) derive in this domain. By attempting to bridge the semantic gap between the visual-linguistic joint latent space of CLIP and the latent space of the state-of-the-art face generation model, StyleGAN [12], such methods are capable of generating and editing face images with specific attributes that are semantically consistent with the given text prompts (*e.g.*, glasses, hairstyle, emotions and expressions), and have achieved impressive results. A concurrent study from [10] also provides a powerful tool for interactive editing of face images using text as hints. They model the mapping from the textual editing instructions to the editing directions in the StyleGAN latent space as a semantic field.

Different from previous works, our work sheds light on a text-guided face *generation* process rather than using

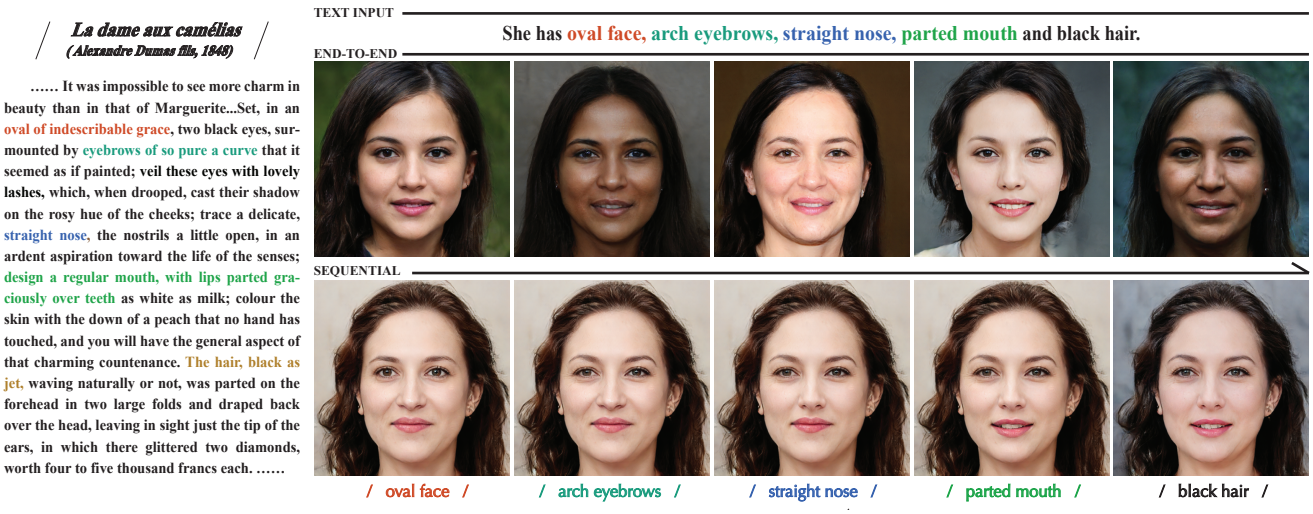

Figure 1. We present a novel pipeline for text-driven face generation, supporting intuitive control over the part-level geometry and appearance of generated facial images using text as the only input (*Right-top*, manually simplified from a novel paragraph on the *Left*.). Our pipeline inherently supports both end-to-end text-to-face generation (*Right-middle*) and sequential generation (*Right-bottom*), as illustrated here.

texts to guide the editing process of human faces, and we explicitly model the geometry and appearance features in the pipeline in a disentangled way, rather than an entangled representation as a StyleGAN latent feature, bringing more flexibility for part-level control. Moreover, we are enabling more attributes to be controlled via text, while previous methods only generate poor editing results on these attributes, as illustrated in our experiments. To this end, we propose a multi-stage framework comprising four parts, namely *Text Parsing Module*, *Feature Extraction Module*, *Graph Recommendation Module*, and *Global Generation Module*. The *Text Parsing Module* maps sentence inputs into attribute-value pairs, thus providing a simple yet accurate way of finding key textual hints. The *Feature Extraction Module* is responsible for disentangling the geometry and appearance features for each facial component, followed by a *Graph Recommendation Module*, which learns the inference relationship among facial components. Finally, the geometry and appearance features optimized by the *Graph Recommendation Module* are transformed into photo-realistic images by the *Global Generation Module*.

We summarize our main contributions as follows:

- We enable detailed part-level attribute-conditioned face generation from textual descriptions, enabling more controllable attributes than previous methods.
- We incorporate graph neural networks (GNN) into the generation process of face images, enabling geometry and appearance recommendation upon given conditions from text.

2. Related works

2.1. Neural Face Generation and Editing.

The prosperity of deep neural networks has demonstrated their capability in the literature of human face generation and editing. To generate face images with high fidelity, Karras et al. [12] propose StyleGAN and a series of its variants [13, 11]. These models are capable of generating high-resolution photo-realistic faces by randomly sampling from a latent distribution $p_Z(\mathbf{z})$ and are robust to noisy inputs, thus inducing an abundance of follow-up works (e.g., [20, 1, 33]), which explore the properties of its intermediate latent space $\mathcal{W}$ to implement conditional face generation and editing. While StyleGAN-based methods could benefit from the unprecedented generation ability of StyleGAN and generate photo-realistic human faces, non-StyleGAN-based methods are deft in this domain as well. For example, Chen et al. [3] propose a structural framework to disentangle the geometry features from the appearance features, using sketch as intermediary. Lee et al. [15] adopt semantic masks as an intermediary for flexible face manipulation while preserving identity and fidelity.

Although these methods are promising in generating and/or manipulating human face images, they do not *implicitly* take into account the inherent coherence among the appearances and geometric features of facial components, thus being incapable of understanding high-level semantics and structures of human faces, let alone recommending and generating faces with geometrically coherent and appearance-consistent human faces. In contrast, our work *explicitly* models the relationship among facial part geome-

try and appearance (respectively) using graphs and achieves easier control over the geometry and appearance features.

2.2. Text-guided Graphics and Vision.

Text enjoys wide applications in human-computer interaction, with recent advances in vision and graphics having integrated text as an interface for image generation and manipulation. Previously, text-based image generation methods [31, 19, 23, 4] focus on generating simple-structured images like birds, using the CUB200 dataset [35], and flowers, using the Oxford-Flower-102 dataset [19], *etc*. These methods generally lack thorough analysis over the target data distribution (in their cases, birds and flowers, *etc*.; in our case, human faces), therefore being unable to improve the quality of the generated images. Based on large pretrained models, DALLE/DALLE2[25, 5] are able to generate complex and semantically abundant images from pure text inputs, achieving phenomenal effects on text-based image generation.

Recent progresses on text-guided graphics and vision are largely facilitated by the strong visual-linguistic representation ability of CLIP. CLIPasso [34] utilizes CLIP image encoder to measure the semantic and geometric similarity between input real images and abstracted sketches, benefiting from the rich semantics within the CLIP text-image joint latent space. CLIPstyler [14] incorporate CLIP for image style transfer, where the desired style is specified via text inputs. Sangkloy *et al*. [27] design a image retrieving system using both text and sketch as query. With the help of this system, users could conduct fine-grained retrieval which could not be achieved using any of the two modalities alone. 3D content creation field also benefits from CLIP, with Text2Mesh [17] being a representative work. The proposed method predicts per-vertex color and positional offsets from the input template mesh, and use a differentiable render to propagate the CLIP 2D semantic supervision to 3D.

Specifically within the face generation and manipulation community, Patashnik et al. [21] introduce three CLIP-based approaches under this direction, all targeting at manipulating inverted StyleGAN images. Xia et al. [39] yet map multi-modal inputs including text into the fixed $\mathcal{W}$ space of StyleGAN, forcing the embeddings of multiple modalities to be as close as possible to the inverted $w \in \mathcal{W}$ of their corresponding real face image. Jiang et al. [10] model the mapping between text features and StyleGAN latent editing directions using an MLP, by which they attempt to solicit the most salient editing direction corresponding to the textual hints. Their approach is deft at editing *global* attributes such as age, beard, smiling emotion, etc., instead of editing part-level geometry and appearance features as we do.

The above-mentioned methods, while having achieved impressive results in manipulating human faces, often rely too much on the representation ability of large pretrained models such as CLIP and StyleGAN. Thus they compromise detailed semantic control over each component of human faces, *i.e*., some attributes in the StyleGAN latent space are highly entangled (as mentioned in [39]). Our work, built upon a local-to-global framework, is able to translate semantic descriptions to part-level visuals with geometry and appearance compatibility, thus supporting disentangled control for each part while also taking the overall coherence into consideration. Also to note that most of the controllable facial attributes enabled by our method do not overlap with those enabled by previous works, and editing/generating these attributes using previous methods yields less satisfying results, as shown in Sec. 4.

3. Methodology

Given an input sentence s describing a human face, we aim to generate a photo-realistic facial image I^{final} with details in accordance with the descriptions in s. To eliminate potential abuse of our work, the input adjectives used to describe the face are restricted within a range (see more discussions the supplementary materials). Due to the diversity of linguistic descriptions, the mapping from sentences to faces is clearly many-to-many, bringing about more ambiguities when s contains fewer detailed specifications for each facial part. Therefore, we suggest a recommendation mechanism to infer the features of facial parts that are not specified in s from specified ones, aiming at a seamless combination of part features during global generation. Note that the input sentence s could also be several separate sentences, as long as they together describe the same face.

This requires us to learn the inter-dependency and intrinsic compatibility among facial parts, from both geometry and appearance perspectives. This requirement in turn leads us to design our whole pipeline in a local-to-global manner. Specifically, during training and inference, we divide a facial image into five parts, namely $part \in P := (leye,\ reye,\ nose,\ mouth,\ bg)$, where bg stands for *background*. See Fig. 2 for more details. Network details are included in the supplementary materials.

3.1. Pipeline

3.1.1 Text Parsing Module

By assumption, the input sentences contain certain patterns which are suitable for extracting attribute-value pairs directly using a regular parser [38]. The parser is used to acquire semantic descriptions for each facial part, including geometry descriptions and appearance descriptions, as previously mentioned. Specifically, given the input sentence(s) s, the parser $\mathcal{P}$ will produce a set of attributes $\mathcal{P}(s)$ that are used to index into the database for finding the corresponding geometry and appearance features for the subsequent gener-

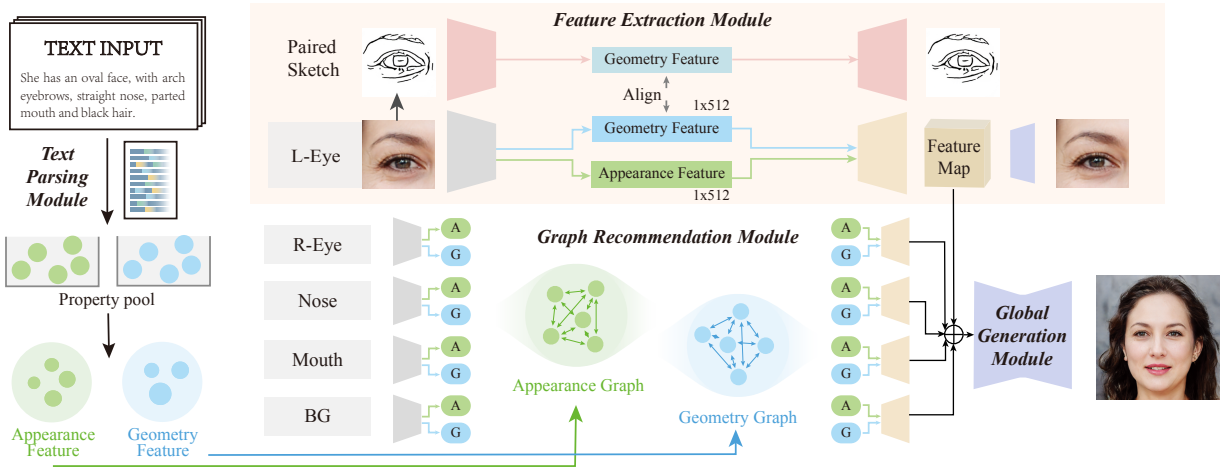

Figure 2. **Overview of our pipeline.** Our pipeline follows a local-to-global manner. The *Text Parsing Module* parses one or multiple sentences s describing the same face into a set of keywords, which are used for conditionally sampling features for face generation from a property pool. The features in the property pool are extracted in advance using the *Feature Extraction Module*, which is trained to disentangle geometry from appearance for each facial component. The *Graph Recommendation Module* contains two graphs, *Appearance Graph* and *Geometry Graph*. They learn the coherence among facial components from appearance and geometry perspectives, respectively, and thus are able to propose recommendation for unspecified facial parts in s. Finally, the *Global Generation Module* is used to fuse the part-level feature maps into a generated face image I^{final}. During inference, the input sentence s is parsed into keywords indexing into the property pool to get corresponding part features. The part features are optimized by the *Appearance Graph* and *Geometry Graph* , after which the optimized features are sent into the part-level decoders ($\{Dec^r\}$) in the *Feature Extraction Module* to get the feature maps. The feature maps are fused at fixed positions and translated into real image I^{real} by the *Global Generation Module*.

ation process. In our implementation, we parse the sentence s using the off-the-shelf spaCy [8] library by analyzing the dependency tree and part of speech of the words.

3.1.2 Feature Extraction Module

This module serves for local geometry and appearance disentanglement. It takes as input real images of facial components I^r_{part} (r standing for *real*) belonging to a whole image I, and outputs their corresponding geometry features f^{geo}_{part} and appearance features f^{app}_{part}. We omit all the subscript *part* in the rest of this section when there is no ambiguity. We propose our *Feature Extraction Module* for explicitly disentangling geometry and appearance features of facial images, using sketches as intermediary [3]. For each facial part, we first train an auto-encoder $AE^s := (Enc^s, Dec^s)$ (s standing for *sketch*) over the sketch domain using L1 reconstruction loss as supervision, after which we get the part-level sketch feature defined as $f^s := Enc^s(I^s) \in \mathbb{R}^{512}$. This geometry feature is further utilized to guide the disentanglement of the geometry and appearance features of real image I^r. This is done by another auto-encoder $AE^r := (Enc^r, Dec^r)$, whose architecture is inspired by [22]. AE^r extracts geometry and appearance features from I^r simultaneously, enabling us to formulate f^{app} and f^{geo} as two vectors, rather than the feature maps used in [3]. Using vectors rather than feature maps is a necessary formulation since the graph networks in *Graph Recommendation Module* could not take feature maps as input. The geometry feature of I^r is defined as the latent vector $f^{geo} \in \mathbb{R}^{512}$ acquired by the fully connected layer after the last encoding block, and the appearance feature of I^r is defined as the linear combination of IN parameters of encoding blocks. Formally, $f^{app} = \sum_i w_i(\mu_i \oplus \sigma_i)$, where $\oplus$ represents vector concatenation, μ_i and σ_i are the mean and standard deviation of the i-th layer's feature map, and $\{w_i\}$ are learnable weights. To achieve disentanglement, we force f^{geo} to be aligned with f^s, which is encoded by the pretrained sketch encoder Enc^s.

3.1.3 Graph Recommendation Module

With the disentangled geometry and appearance features, we propose two graph neural networks, one for recommending compatible geometry features for unspecified parts (*Geometry Graph*), and the other for unifying the appearance of generated face image from part-level (*Appearance Graph*). For the inference procedure, please refer to Sec. 3.2.

Geometry Graph Our key observation here is that the geometry features of different facial parts should share an intrinsic coherence, *i.e.* not all the combinations of facial geometry form compatible faces [42]. For example, the eyes of the same face should be largely symmetric, while

the size of mouth and shape of jaw will both influence the contour of the whole face, *etc*. We formulate the recommendation problem as a conditional sampling and prediction of unspecified facial parts, and model the inter-dependency of geometry features among different facial parts as a 5-node (one node represents one facial part) bipartite graph $G^{geo} := (V^{geo}, E^{geo})$ during each step of inference, where V^{geo} contains the geometry features of 5 nodes and E^{geo} comprises the edges from every node of specified/predicted parts to every node of unspecified/unpredicted ones. Formally, let P_s denote the text-specified/predicted subset of P, we have

$$V^{geo} := \{f^{geo}_{part} \mid part \in P\} \tag{1}$$

$$E^{geo} := \{e^{geo}_{x\to y} : f^{geo}_x \mapsto f^{geo}_y \mid x \in P_s, y \in P \backslash P_s\} \tag{2}$$

where each edge e^{geo}_{xy} in E^{geo} is implemented as an MLP. We denote the output of *Geometry Graph* as $\{f'^{geo}\}$.

Appearance Graph With this appearance graph, we aim to achieve controllable style fusing for appearance features from different source images. We observe the fact that the appearance of one facial part may largely tell what other parts look like. That is, for example, if we know that the eyes of a face have a light/dark skin color, we will have enough confidence to reason that the whole face has a light/dark color. This inter-dependency of appearance features among different parts is modeled using a 5-node complete graph $G^{app} := (V^{app}, E^{app})$, formally,

$$V^{app} := \{f^{app}_{part} \mid part \in P\} \tag{3}$$

$$E^{app} := \{e^{app}_{x\to y} : f^{app}_x \mapsto f^{app}_y \mid x, y \in P, x \neq y\} \tag{4}$$

We model every edge $e^{app}_{xy} \in E^{app}$ as a unified EdgeConv [37] function which is shared across different edges to update the node features during every propagation. The outputs of *Appearance Graph* are denoted as $\{f'^{app}\}$.

3.1.4 Global Generation Module

We base our *Global Generation Module* on the commonly adopted image-to-image translation model pix2pixHD [36], which takes as input the optimized appearance feature $\{f'^{app}\}$ and the part-level geometry features $\{f'^{geo}\}$, and outputs the final synthesized image I^{final}. $\{f'^{app}\}$ and $\{f'^{geo}\}$ are first sent through the $\{Dec^r\}$ mentioned in Sec. 3.1.2, after which we spatially combine the feature map of the second-last layer of $\{Dec^r\}$ as indicated in Fig. 2. The combined feature map is then fused into a photo-realistic image I^{final} using Dec^{global} consisting of a sequence of ResBlocks [7].

3.2. Graph Recommendation Mechanism

We formalize the inference logic of *Graph Recommendation Module* in this subsection.

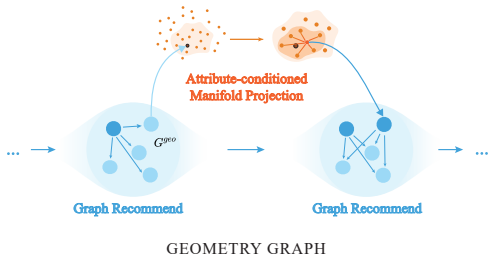

Figure 3. **Illustration of the graph recommendation for *Geometry Graph***. We iteratively perform attribute-conditioned manifold projection to generate compatible geometry features for the whole face.

3.2.1 Geometry Graph

The inference procedure of *Geometry Graph* follows a step-by-step manner, where we start from deciding the geometry feature for bg. If f^{geo}_{bg} is specified in the input sentence s, we conditionally sample a geometry feature from our property pool using the specified attributes as condition. If f^{geo}_{bg} cannot be directly inferred from the input sentence s, *i.e*. no key in $\mathcal{P}(s)$ is relevant with the face contour, we randomly sample a geometry feature for f^{geo}_{bg} from our property pool. Then f^{geo}_{bg} is used to predict compatible geometry features for all the other parts. Generally, the predicted feature for an unspecified part is forwarded as follows,

$$\hat{f}^{geo}_{part} = \frac{1}{|P_s|} \sum_{x \in V_s} e^{geo}_{x\to part}(f^{geo}_x), part \in P \backslash P_s \tag{5}$$

where P_s is the specified/predicted subset of P as mentioned in Sec. 3.1.3. When deciding the next part geometry feature, for example f^{geo}_{nose}, we already have a predicted one from f^{geo}_{bg}, which we denote as $\hat{f}^{geo}_{nose}$. Therefore, if $nose$ is not specified, we directly use $\hat{f}^{geo}_{nose}$ as f^{geo}_{nose}. Otherwise, we could sample from all the geometry features in our database which satisfy the specified attributes for $nose$, and apply manifold projection to $\hat{f}^{geo}_{nose}$ over the sampled subset of database. We call this process **attribute-conditioned manifold projection**, abbreviated as $\mathcal{A}$. Formally, the prediction logic for f^{geo}_{nose} can be formulated as follows,

$$f^{geo}_{nose} = \begin{cases} \hat{f}^{geo}_{nose}, & \text{if } nose \text{ is not specified} \\ \mathcal{A}(\hat{f}^{geo}_{nose}), & \text{if } nose \text{ is specified} \end{cases} \tag{6}$$

After the two iterations above, f^{geo}_{nose} and f^{geo}_{bg} have been decided, which will be fixed and used to predicted the rest undecided part geometry features like what has been done for predicting f^{geo}_{nose}. Iterations terminate until all the part-level geometry features have been decided. We denote the output of *Geometry Graph* as $\{f'^{geo}\}$.

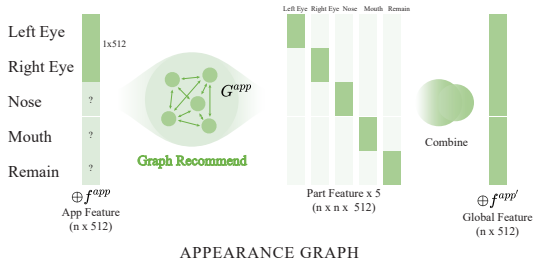

Figure 4. **Illustration of the graph recommendation for *Appearance Graph*.** Missing appearance features could be deduced from known ones. Known appearance features would unify with each other to achieve coherent facial appearances.

3.2.2 Appearance Graph

The *Appearance Graph* learns the relationship among the appearance features of different facial parts. Since the appearance features of different parts do not lie on the same manifold, we extend each $f^{app} \in \mathbb{R}^{512}$ to $\hat{f}^{app} \in \mathbb{R}^{2560}$ during both training and inference to expect $\{\hat{f}^{app}\}$ belong to the same space. Intuitively, one could interpret $\hat{f}^{app}$ as a vector belonging to the direct sum of five part-level appearance feature space . The extended dimensions and missing part-level appearance features are padded with zeros as default. $\{\hat{f}^{app}\}$ are used to perform message-passing updates, during which process the data flow between every pair of nodes unify the appearance features from different facial parts. Finally, after several rounds (5 in our implementation) of message-passing, we acquire the optimized appearance features for each part: $\{f'^{app}\}$, $f'^{app} \in \mathbb{R}^{512}$ by extracting the corresponding slices of $\{\hat{f}'^{app}\}$, which are optimized by *Appearance Graph* from $\{\hat{f}^{app}\}$. To be more specific, we have

$$\hat{f}^{app}_{part}[i \times 512 : (i+1) \times 512] = f^{app}_{part} \tag{7}$$

$$f'^{app}_{part} = \hat{f}'^{app}_{part}[i \times 512 : (i+1) \times 512], \tag{8}$$

where i is the index of $part$ in P.

3.3. Training Stages

The training process of the entire pipeline contains three stages. We introduce them respectively in this subsection. The training process is totally independent from any attribute label.

Stage I: Training the Feature Extraction Module. As described in Sec 3.1.2, $(f^{geo}, f^{app}) = Enc^r(I^r)$. During training, we force the decoder Dec^r to reconstruct the original image, *i.e.*, forcing $I^{recon} = Dec^r(f^{geo}, f^{app})$ to be as close to I^r as possible. Therefore we have the first loss term $\mathcal{L}_{recon}$ defined as follows,

$$\mathcal{L}^{local}_{recon} = \|I^r - I^{recon}\|_1. \tag{9}$$

To eliminate the interdependence of geometry and appearance features, we align the geometry feature space of real images ($\{f^{geo}\}$) with that of sketches ($\{f^s\}$), where f^s is extracted via the pre-trained Enc^s. Thus, the second loss term $\mathcal{L}_{align}$ comes as follows,

$$\mathcal{L}_{align} = \|f^{geo} - f^s\|_2. \tag{10}$$

Further, we utilize the third loss term – adversarial loss $\mathcal{L}_{adv}$, in a similar way as [16] do, by employing a discriminator Dis^r,

$$\mathcal{L}^{Enc,Dec}_{adv} = \mathbb{E}[(Dis^r(I^{recon}) - 1)^2], \tag{11}$$

$$\mathcal{L}^{Dis}_{adv} = \mathbb{E}[(Dis^r(I^r) - 1))^2] + \mathbb{E}[(Dis^r(I^{recon})^2]. \tag{12}$$

In summary, the training objective for *Stage I* is formulated as a minimax game as follows,

$$\min_{Enc,Dec} \mathcal{L}^{local}_{recon} + \lambda_{align}\mathcal{L}_{align} + \lambda_{adv}\mathcal{L}^{Enc,Dec}_{adv}, \tag{13}$$

$$\min_{Dis} \mathcal{L}^{Dis}_{adv}. \tag{14}$$

In our implementation, we set $\lambda_{align} = 0.01$, and $\lambda_{adv} = 0.005$.

Stage II: Training the Geometry Graph in the Graph Recommendation Module. The *Geometry Graph* models the geometric coherence among facial parts. This is enabled by learning a set of MLP-based mappings between the latent spaces of every pair of facial components. For each pair of facial components $x, y \in V^{geo}, x \neq y$, we force the MLP e_{xy} to map f^{geo}_x to f^{geo}_y. Therefore the loss is simply defined as an L2 loss between the predicted y geometry feature $f'^{geo}_y := e_{xy}(f^{geo}_x)$ and the f^{geo}_y:

$$\min_{e_{xy}} \|f^{geo}_y - f'^{geo}_y\|_2. \tag{15}$$

Stage III: Joint Training of the Global Generation Module and the *Appearance Graph* in the Graph Recommendation Module. The *Appearance Graph* learns the style inter-dependency among facial components, with which we want to achieve appearance reasoning when observing partial appearance of a face, and appearance fusing when combining facial components from different sources. Therefore, we train our *Appearance Graph* together with the *Global Generation Module* using the reconstruction loss as main supervision. Given the original geometry features $\{f^{geo}\}$ and partial appearance features $\{Dropout(f^{app}, p)\}$, where *Dropout* represents *Dropout* function operating on every part-level appearance feature and p is the *Dropout* probability ($p = 0.1$ in our implementation), we first compute the optimized appearance features $\{f'^{app}\}$ by calling G^{app}. Then $\{f^{geo}\}$ and $\{f'^{app}\}$ are used to compute the local feature maps for each part, which are further combined into $F \in \mathbb{R}^{32\times512\times512}$. Finally, we have

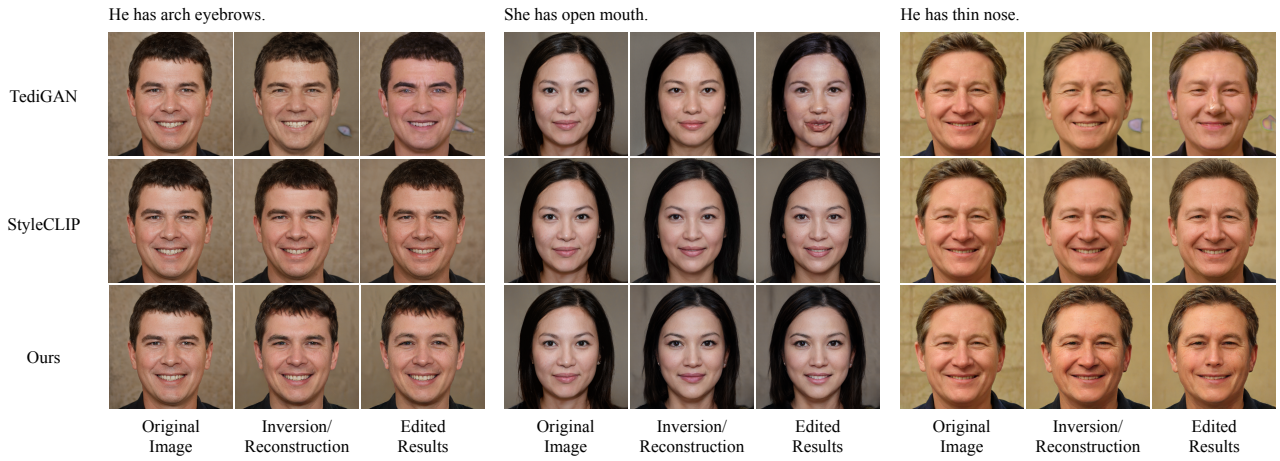

Figure 5. **Editing comparisons with state-of-the-art methods.** We perform single-attribute editing for each example. In all three examples, TediGAN [39] fails to produce changes corresponding to the text-specified facial attributes. For StyleCLIP [21], it succeeds in turning a closed mouth into an opened one, while it also fails on the other two cases. We speculate from an empirical perspective that the success of editing an opened mouth and the failure of editing eyebrows/nose shape may both ascribe to the entangled nature of the StyleGAN latent space, as prior arts [1, 10, 9, 26] have already managed to change the mouth openness via StyleGAN latent manipulation but none (to our knowledge) have succeeded in editing eyebrows/nose in the same way. Overall, our method yields the most satisfying results from both reconstruction quality and editing effectiveness.

$I^{final} = Dec^{global}(F)$. The first loss is L1 reconstruction loss,

$$\mathcal{L}_{recon}^{global} = \|I^{final} - I\|_1. \tag{16}$$

We further employ VGG loss [28] and Lab loss [30] to constrain on the visual accuracy of generated images. Therefore, the training objective for this stage is as follows,

$$\min_{G^{app}, Dec^{global}} \mathcal{L}_{recon}^{global} + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_{Lab}\mathcal{L}_{Lab}. \tag{17}$$

We set $\lambda_{vgg} = 0.2, \lambda_{Lab} = 0.001$ in our experiments.

4. Experiments

4.1. Data Preparation

When generating the training dataset (as well as our database), we only generate frontal faces to eliminate the negative impacts of occlusion and pose, i.e. we reasonably use the a priori of face layout and pose. Here, we explain for short why we only use frontal and occlusion-free faces. The reason is two-fold:

- Non-frontal faces bring about difficulties for the graph recommendation module to infer the accurate geometry/appearance correlation. For example, an apparent geometry relation within the human face is the symmetry of two eyes. If a face has a big yaw, the symmetry would not exist in the image space because this 3D symmetry is not preserved when being projected to 2D.
- Non-frontal faces and occlusions would make it difficult for the detection API to make accurate judgements. Intuitively, for example, if the face has a big yaw/pitch, the arched eyebrow may look like a straight eyebrow, which would lead to misjudgement of the API.

Attr		Face	Brows	Eyes	Nose	Mouth
Acc	Ours	0.84	0.92	0.87	0.86	0.76
	AttnGAN	0.16	0.16	0.24	0.30	0.20
	DM-GAN	0.17	0.14	0.22	0.28	.0.22

Table 1. **Text-image correspondence accuracy.** During evaluation, we change and set the type for each attribute and calculate the accuracy of this attribute after generation. Results show that our method succeeds in generating face images satisfying the semantic designations of the input text and surpasses the accuracy of previous arts [31, 43].

4.2. Results and Evaluations

We conduct extensive experiments to demonstrate the effectiveness and usability of our system. We evaluate our method from four aspects: attribute accuracy of the generated images (Sec. 4.2.1), comparison with the state-of-the-art text-based image generation techniques on human faces (Sec. 4.2.2), ablation study (Sec. 4.3), and perceptual study (see supplementary materials).

4.2.1 Attribute Accuracy of the Generated Faces

To test the accuracy of text-image correspondence of the generated images (*i.e.* do the attributes in the generated im-

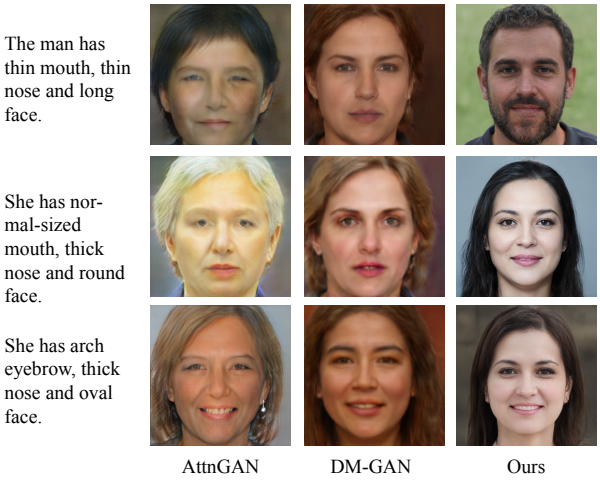

Figure 6. **Generation comparisons with state-of-the-art methods.** Given the same input sentence (leftmost in each example), our result is significantly better than the other two methods, in terms of both image quality and attribute accuracy.

ages match the descriptions?), for each attribute, we generate a batch of 100 images by specifying only one attribute in the input sentence. Then, these generated images are sent to the facial attribute detection APIs [2, 6, 18] for re-detection. We calculate the accuracy for each attribute, as shown in Table 1.

4.2.2 Comparison with State-of-the-Arts

Existing text-based works that are relevant to our work can be categorized into two tracks: text-based image generation [31, 43, 23], and text-guided face manipulation [21, 39]. Since our work can be adapted to support face manipulation, we make comparisons for the two tasks.

For the generation task, we compare with [31, 43] by retraining their models using the official implementations but with our own dataset, and setting the same sentence as the input to all three works. Since the original implementations of [31, 43] both set the maximum resolution to 256×256, we directly use their results under this resolution for comparison with our results which have a resolution of 512×512. This is deemed as a fair comparison by us due to the fact that generating images with higher resolution is often considered to be more difficult. Please note that their models are not specifically designed for text-to-face generation but rather for a more general text-to-image generation task, while our model is specifically designed for generating human faces. Although we explicitly take into account the prior of human face layout into our model architecture, we argue that our comparison is better than nothing, since there does not exist relevant works under the exactly same settings as ours: text-to-face generation with disentangled feature control.

For the manipulation tasks, we adapt our pipeline as follows to support manipulation given an input image I: We encode I to get $\{f^{geo}\}$ and $\{f^{app}\}$ using Enc^r, and then substitute the features of specified editing attributes and perform graph recommendation upon the modified features. Here we compare with the two existing open-world-text-based editing methods [39, 32] for editing functionality and only compare the results of editing single attribute, because it is intuitive to perform multi-attribute editing by serializing the editing processes of single-attribute editing. We use the standard optimization-based method in [39] and the Global Direction method in [21] for comparison.

4.3. Ablation Study

Graph Recommendation Module is an essential part in our framework to ensure the quality and realism of the generated results. To demonstrate its validity for geometry or appearance recommendation, we conduct an ablation study with/without graph. Since the *Appearance Graph* and *Geometry Graph* operate separately, we perform the ablation study in two ways. First, we randomly edit one part of the face and observe the generated images with/without the *Geometry Graph*. Specifically, as illustrated in Fig. 7, we fix f^{geo}_{bg} and keep changing f^{geo}_{eye1} and f^{geo}_{eye2}. In this way, the *Geometry Graph* is expected to predict f^{geo}_{nose} and f^{geo}_{mouth} to form a compatible face. Second, we testify the effectiveness of our *Appearance Graph* by swapping the appearance features of several facial parts from two faces. We replace the appearance features of the source person with those of the target person. With *Appearance Graph*, such a swapping operation is expected not to produce any sharp boundaries on the faces, as shown in Fig. 8. While without *Appearance Graph*, the swapping operation produces images with inconsistent color.

4.4. Geometry and Appearance Morphing

The encoder network Enc^r of our framework could extract the geometry and appearance respectively from a real image. The representations of those two features are both 1×512 latent vectors. Our method could do interpolation in each feature domain. As shown in Fig. 9, the upper left and the lower right are the given images. Along the vertical axis is to interpolate the appearance, while along the horizontal axis is to interpolate the geometry. The intermediate images between the two corners are the interpolation results, where the geometry and appearance features smoothly change.

5. Limitations

The motivation of our work originates from an entertainment and interaction setting. Therefore, directly using our model for applications such as criminal investigation is improper and should involve more dedicated considerations

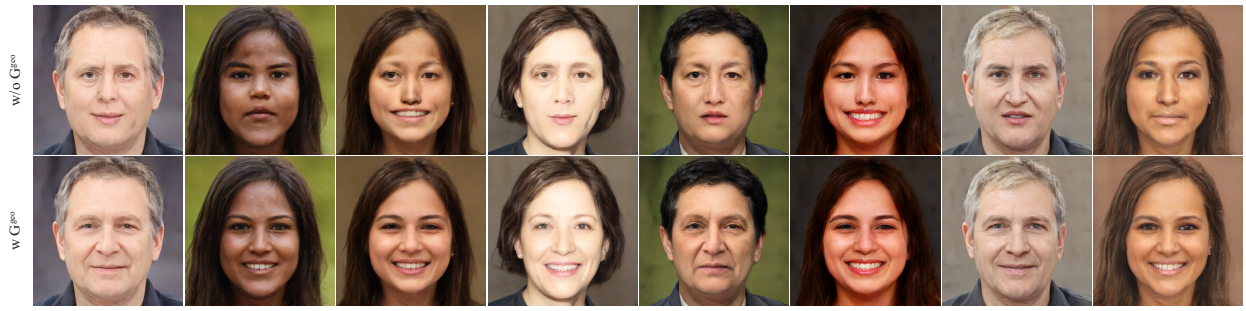

Figure 7. **Ablation study of the *Geometry Graph*.** We randomly sample facial geometry features to generate face images. The upper row shows the results generated from geometry features without being optimized by the *Geometry Graph*, the lower row shows the results generated using *Geometry Graph*. Obviously, there exist artifacts on the boarders of different facial parts in the generated faces when the geometry features are not being optimized by *Geometry Graph*. On the other hand, when optimized by *Geometry Graph*, the geometry features of different facial parts are more consistent with each other and thereby producing more realistic results.

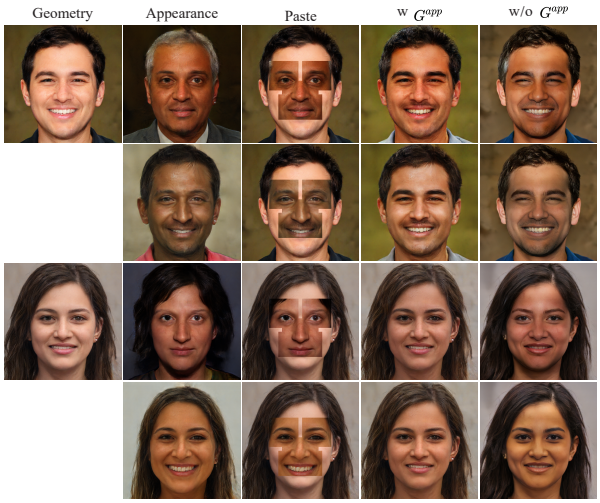

Figure 8. **Ablation study of the *Appearance Graph*.** Editing the appearance of the source image (Geometry) using part-level reference images (Appearance). The Paste column shows the pasted appearance reference over the source image. As shown in the rightmost two columns, the edited results with *Appearance Graph*are much more color-consistent compared to the rightmost column where the results are generated without incorporating the *Appearance Graph*.

Figure 9. **Interpolation via Geometry and Appearance Axes.** The appearance gradually changes along each column, while the geometry changes along each row.

beforehand. In other words, one of our model's limitations, from the application perspective, is that the accuracy and experimental settings restrict it from being used as a way to facilitate applications requiring extra accuracy.

Another limitation of our work from the technical perspective, is that our model does not perform well on complicated hair styles such as wavy hair, plate hair, bangs, etc. Thus it could not generate faces with such hair styles. We refer to the readers to [30, 41] about how to manipulate complex hairstyles. More details about failure modes are appended in the supplementary materials.

Last but not least, the generation results of our model rely a lot on the dataset/database. The frontal faces used in our work require extensive works to generate and check their validity. Limited by the diversity encoded in the StyleGAN generator, our database inherits such bias. The bias could be reduced as we are continuing enlarging our dataset. We will release the code and provide an online system when the dataset is diverse enough.

6. Conclusion

In this work, we present a local-to-global framework for generating realistic facial images from pure textual inputs, enabling linguistic control over the geometry and appearance features of every facial part. We demonstrate the ef-

fectiveness of our method by comparing with the state-of-the-art text-based editing and text-to-image models as well as conducting a convincing user study under a real-word scenario. However, currently our pipeline may not apply to complex sentences. Generation from sentences with more fuzzy descriptions is to be adapted in the future.

Acknowledgement

This work was supported by grants from the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the National Natural Science Foundation of China (No. 62061136007 and No. 62102403), Science and Technology Service Network Initiative of the Chinese Academy of Sciences (No. KFJ-STS-QYZD-2021-11-001) and the Youth Innovation Promotion Association CAS.

References

[1] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *CoRR*, abs/2008.02401, 2020. 2, 7

[2] Alibaba. https://help.aliyun.com/document_detail/130846.html, 2020. 8

[3] S.-Y. Chen, F.-L. Liu, Y.-K. Lai, P. L. Rosin, C. Li, H. Fu, and L. Gao. Deepfaceediting: Deep face generation and editing with disentangled geometry and appearance control, 2021. 2, 4

[4] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao. Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[5] A. R. et al. Hierarchical text-conditional image generation with clip latents, 2022. 3

[6] Face++. https://www.faceplusplus.com.cn/face-detection/, 2020. 8

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. 5

[8] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. 4

[9] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020. 7

[10] Y. Jiang, Z. Huang, X. Pan, C. C. Loy, and Z. Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 3, 7

[11] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 2

[12] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 1, 2

[13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. 2

[14] G. Kwon and J. C. Ye. Clipstyler: Image style transfer with a single text condition. *arXiv preprint arXiv:2112.00374*, 2021. 3

[15] C.-H. Lee, Z. Liu, L. Wu, and P. Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[16] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks, 2017. 6

[17] O. Michel, R. Bar-On, R. Liu, S. Benaim, and R. Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13492–13502, June 2022. 3

[18] Microsoft. https://docs.microsoft.com/en-in/azure/cognitive-services/face/, 2020. 8

[19] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 3

[20] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Transactions on Graphics (TOG)*, 39:1 – 14, 2020. 2

[21] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021. 1, 3, 7, 8

[22] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [to appear]. 4

[23] T. Qiao, J. Zhang, D. Xu, and D. Tao. Mirrorgan: Learning text-to-image generation by redescription. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3, 8

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[25] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation, 2021. 3

[26] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 7

[27] P. Sangkloy, W. Jitkrittum, D. Yang, and J. Hays. A sketch is worth a thousand words: Image retrieval with text and sketch. *European Conference on Computer Vision, ECCV*, 2022. 3

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 7

[29] SmartClick. https://smartclick.ai/articles/how-artificial-intelligence-is-used-in-the-film-industry/, 2021. 1

[30] Z. Tan, M. Chai, D. Chen, J. Liao, Q. Chu, L. Yuan, S. Tulyakov, and N. Yu. Michigan: Multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)*, 39(4):1–13, 2020. 7, 9

[31] X. Tao, Z. Pengchuan, H. Qiuyuan, Z. Han, G. Zhe, H. Xiaolei, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 7, 8

[32] A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H.-P. Seidel, P. Pérez, M. Zollhöfer, and C. Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, 2020. 8

[33] A. Tewari, M. Elgharib, M. B. R., F. Bernard, H.-P. Seidel, P. Pérez, M. Zollhöfer, and C. Theobalt. Pie: Portrait image embedding for semantic control, 2020. 2

[34] Y. Vinker, E. Pajouheshgar, J. Y. Bo, R. C. Bachmann, A. H. Bermano, D. Cohen-Or, A. Zamir, and A. Shamir. Clipasso: Semantically-aware object sketching, 2022. 3

[35] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[36] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[37] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 5

[38] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, and W.-Y. Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6609–6618, 2019. 3

[39] W. Xia, Y. Yang, J.-H. Xue, and B. Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 7, 8

[40] W. Xia, Y. Yang, J.-H. Xue, and B. Wu. Towards open-world text-guided face image generation and manipulation, 2021. 1

[41] C. Xiao, D. Yu, X. Han, Y. Zheng, and H. Fu. Sketchhairsalon: Deep sketch-based hair image synthesis, 2021. 9

[42] B. Zhu, C. Lin, Q. Wang, R. Liao, and C. Qian. Fast and accurate: Structure coherence component for face alignment, 2020. 4

[43] M. Zhu, P. Pan, W. Chen, and Y. Yang. DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5802–5810. Computer Vision Foundation / IEEE, 2019. 7, 8