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

Zhaoyang Zhang, Junliang Chen, Hongbo Fu, Jianjun Zhao, Shu-Yu Chen, and Lin Gao

Abstract—Recent years have witnessed the emergence of various techniques proposed for text-based human face generation and manipulation. Such methods, targeting bridging the semantic gap between text and visual contents, provide users with a deft hand to turn ideas into visuals via text interface 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 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 highly 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.

Index Terms—Image Generation, Face Editing, Sketching Interface, Text-based User Interaction

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 in the content of a novel and wonder about more details behind the text. Can we reconstruct the faces in the novel simply from textual descriptions [1]? Although it sounds impossible in the past, this depiction-to-visualization procedure has the potential to become a reality now, enabled by the fascinating progress of human face generation and manipulation methods as well as natural language processing techniques. However, sometimes textual descriptions are incomplete for describing every detail of desired faces, *i.e.* one sentence may not cover detailed description of every facial part at once. This is due to the fact that the users may not want to specify every detail of the face at the very beginning. For example, a user may start with describing the desired face as "An oval face with blonde hair", then adding more facial descriptions such as "smiling mouth" and "big nose", *etc*. Thus how to generate satisfying human faces from partial descriptions becomes a challenging problem. In this work, we aim to visualize the text-to-face depiction process by building an interface for converting the textual descriptions to realistic human faces, where we introduce a recommendation mechanism with graph neural networks (GNNs) for proposing coherent faces given partial descriptions. Also, when detailed descriptions are provided, our model is able to generate facial images corresponding to the given texts, and our proposed GNNs will optimize over the generated faical parts and achieve higher fidelity and consistency.

Efforts have been devoted to 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 [2], a series of works (*e.g.*, [3], [4]) 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 [5], 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 [6] 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 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 each

- *Z. Zhang is with the Department of Computer Science, Yale University, CT 06520, USA. This work was done when he was an undergraduate at the University of Chinese Academy of Sciences, Beijing 100190, China. E-mail: zhaoyang.zhang@yale.edu*
- *L. Gao and S. Chen are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China. Email: {gaolin, chenshuyu}@ict.ac.cn*
- *J. Chen and J. Zhao are with the Department of Film and TV Technology, Beijing Film Academy. E-mail: juneleungchan@gmail.com, zhaojianjun@bfa.edu.cn*
- *H. Fu is with the School of Creative Media, City University of Hong Kong. E-mail: hongbofu@cityu.edu.hk*

Manuscript received April 19, 2005; revised August 26, 2015.

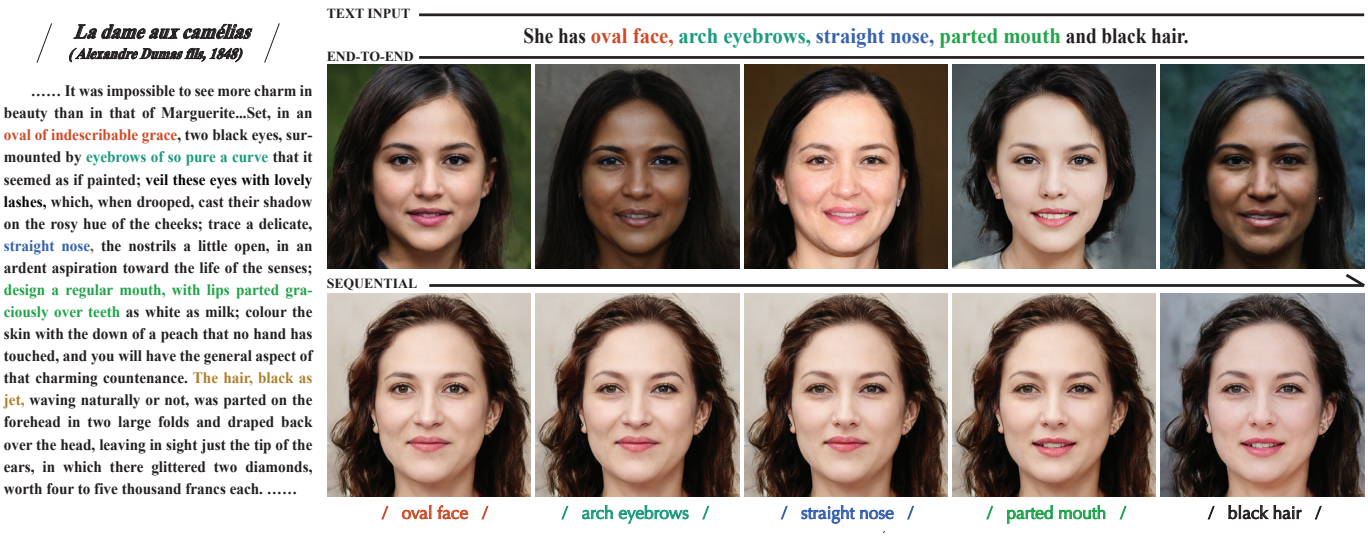


Fig. 1: We present a novel pipeline for text-driven face generation, supporting intuitive control over the part-level geometry and appearance of generated facial images using text as the only input (*Right-top*, manually simplified from a novel paragraph on the *Left*.). Our pipeline inherently supports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

facial component's geometry and appearance features, 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, which enables more controllable attributes than previous methods.
- We incorporate graph neural networks (GNN) into the generation process of face images, which enables geometry and appearance recommendation upon given conditions from the text.

2 Related Works

2.1 Neural Face Generation and Editing

The prosperity of deep neural networks has demonstrated their capability in human face generation and editing literature. To generate face images with high fidelity, Karras et al. [5] propose StyleGAN and a series of its variants [7], [8]. These models are capable of generating high-resolution photo-realistic faces by randomly sampling from a latent distribution $p_Z(\mathbf{z})$. They are robust to noisy inputs, thus inducing an abundance of follow-up works (e.g., [9], [10], [11]), 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 also deft in this domain. For example, Chen et al. [12] propose a structural framework to disentangle the geometry features from the appearance features, using the sketch as an intermediary. Lee et al. [13] 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 *explicitly* 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 geometry and appearance (respectively) using graphs and achieves easier control over the geometry and appearance features.

2.2 Text-based Multimodal Generation

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 [14], [15], [16], [17] focus on generating simple-structured images like birds, using the CUB200 dataset [18], and flowers, using the Oxford-Flower-102 dataset [15], *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 [19], [20] are able to generate complex and semantically abundant images from pure text inputs, achieving phenomenal effects on text-based image generation. Another track of works on text-based audio generation also grasps attention within the community [**?**], [21], [22], [23]. For example, MusicLM [21] models the music generation process in a hierarchical manner, which is proved efficient in previous arts. Our work also models the generation of facial images in a local-to-global manner.

Recent progresses in text-guided graphics and vision are largely facilitated by CLIP's strong visual-linguistic representation ability. CLIPasso [24] utilizes a 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 [25] incorpo-

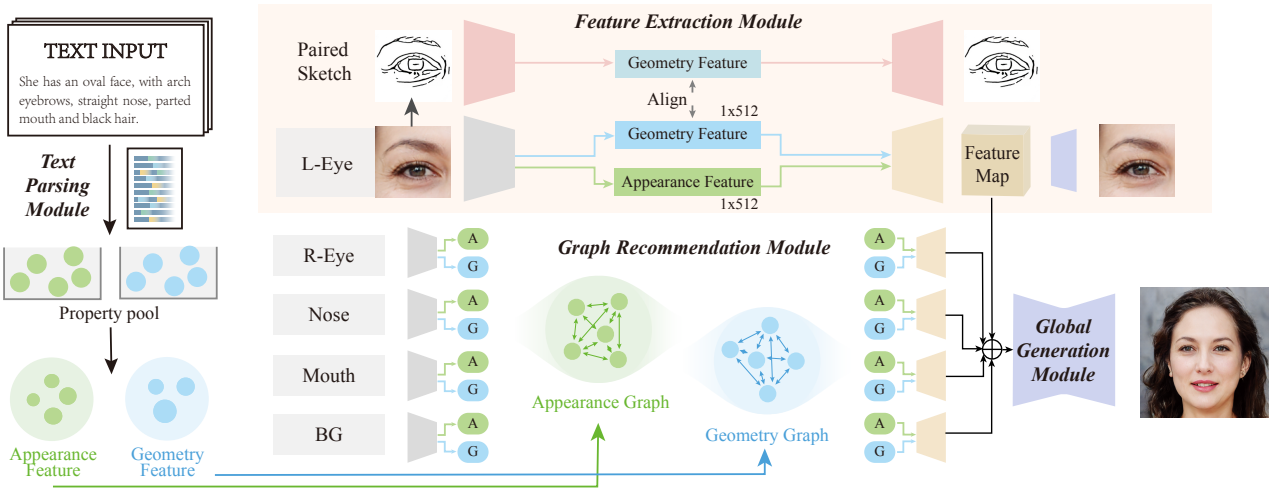


Fig. 2: **Overview of our pipeline.** Our pipeline follows a local-to-global manner. The *Text Parsing Module* parses one or multiple sentences s describing the same face into a set of keywords, which are used for conditionally sampling features for face generation from a property pool. The features 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 can propose recommendations for unspecified facial parts in s. Finally, the *Global Generation Module* fuses 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*.

rates CLIP for image style transfer, where the desired style is specified via text inputs. Sangkloy *et al.* [26] design an image retrieving system using both text and sketch as a 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 [27] being a representative work. The proposed method predicts per-vertex color and positional offsets from the input template mesh and uses a differentiable render to propagate the CLIP 2D semantic supervision to 3D.

Specifically within the face generation and manipulation community, Patashnik et al. [3] introduce three CLIP-based approaches under this direction, all targeting manipulating inverted StyleGAN images. Xia et al. [28] 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. [6] 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. AnyFace [29] first defines the problem of free-style text-to-image face as well as proposes a two-stream architecture that utilizes CLIP and StyleGAN to achieve open-world human face generation and manipulation. Different from their method which depends on StyleGAN to generate facial images as a whole, our method relies on partial generators to generate facial parts separately, together with two GNNs to ensure the fidelity and consistency of facial images. Very recently, diffusion models [30], [31], [32], [33], [34], [35] have elevated text-to-image generation to a new level, where text-to-face generation becomes a natural subtask. These approaches are 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 text-based facial image generation 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 [28]). 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 considering the overall coherence. 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 in 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 describe the same face together.

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 $P := (leye,\ reye,\ nose,\ mouth,\ bg)$, 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 suitable for extracting attribute-value pairs directly using a regular parser [36]. As previously mentioned, the parser is used to acquire semantic descriptions for each facial part, including geometry and appearance descriptions. 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 generation process. In our implementation, we parse the sentence s using the off-the-shelf spaCy [37] 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_p^r (r standing for *real*) belonging to a whole image I, and outputs their corresponding geometry features f_p^{geo} and appearance features f_p^{app}, where $p \in P$ is in short of *part*. We omit all the subscript *p* 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 [12]. For each facial part, we first train an auto-encoder consisting of $\mathcal{E}^s, \mathcal{R}^s$ (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 $f^s := \mathcal{E}^s(I^s) \in \mathbb{R}^{512}$. Serving as the geometry features, these part-level sketch features are further utilized to guide the disentanglement of the geometry and appearance features of real image I^r. Such disentanglement is done by another auto-encoder $\mathcal{E}^r,\ \mathcal{R}^r$. This auto-encoder learns to extract 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 [12]. 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 $\mathcal{E}^s$.

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*). Please refer to Sec. 3.2 for the inference procedure.

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 [38]. For example, the eyes of the same face should be largely symmetric, while the size of the mouth and the shape of the 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

$$V^{geo} := \{f_p^{geo} \mid p \in P\} \tag{1}$$

$$E^{geo} := \{e_{x\to y}^{geo} : f_x^{geo} \mapsto f_y^{geo} \mid x \in P_s, y \in P \backslash P_s\} \tag{2}$$

where each edge e_{xy}^{geo} 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 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,

$$V^{app} := \{f_p^{app} \mid p \in P\} \tag{3}$$

$$E^{app} := \{e_{x\to y}^{app} : f_x^{app} \mapsto f_y^{app} \mid x, y \in P, x \neq y\} \tag{4}$$

We model every edge $e_{xy}^{app} \in E^{app}$ as a unified EdgeConv [39] 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 [40], 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 $\{\mathcal{R}^r\}$ mentioned in Sec. 3.1.2, after which we spatially combine the feature map of the second-last layer of $\{\mathcal{R}^r\}$ as indicated in Fig. 2. The combined feature map is then fused into a photo-realistic image I^{final} using $\mathcal{R}^{global}$ consisting of a sequence of ResBlocks [41].

3.2 Graph Recommendation Mechanism

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

3.2.1 Geometry Graph

The inference procedure of *Geometry Graph* follows a step-by-step manner, where we start by deciding the geometry feature for

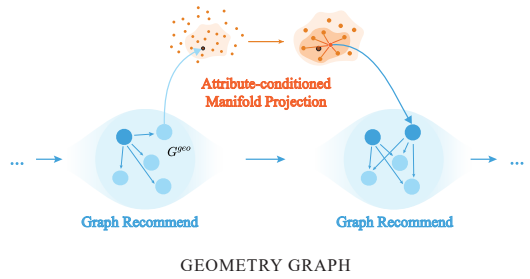


Fig. 3: **Illustration of the graph recommendation for *Geometry Graph***. We iteratively perform attribute-conditioned manifold projection to generate compatible geometry features for the whole face.

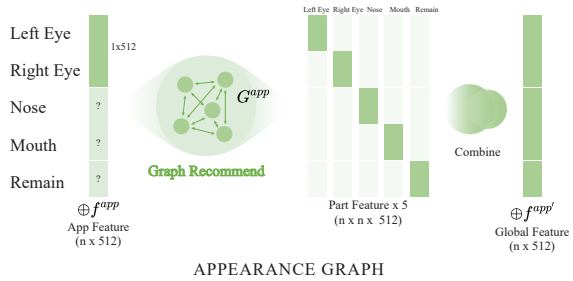


Fig. 4: **Illustration of the graph recommendation for *Appearance Graph*.** Missing appearance features could be deduced from known ones. Known appearance features would unify with each other to achieve coherent facial appearances.

bg. If f_{bg}^{geo} is specified in the input sentence s, we conditionally sample a geometry feature from our property pool using the specified attributes as the condition. If f_{bg}^{geo} 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_{bg}^{geo} from our property pool. Then f_{bg}^{geo} is used to predict compatible geometry features for all the other parts. Generally, the predicted feature for an unspecified part is forwarded as follows,

$$\hat{f}_p^{geo} = \frac{1}{|P_s|} \sum_{x \in V_s} e_{x \to p}^{geo}(f_x^{geo}), p \in P \backslash P_s \tag{5}$$

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

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

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

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 unifies the appearance features from different facial parts. For each round of message passing, we perform

$$\hat{f}_p^{app} = \frac{1}{|P| - 1} \sum_{x \in P \backslash \{p\}} e_{x \to p}^{app}(\hat{f}_x^{app}, \hat{f}_p^{app}) \tag{7}$$

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

$$\hat{f}_p^{app}[i \times 512 : (i+1) \times 512] = f_p^{app} \tag{8}$$

$$f_p'^{app} = \hat{f}_p'^{app}[i \times 512 : (i+1) \times 512], \tag{9}$$

where i is the index of part p 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 independent of any attribute label.

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

$$\mathcal{L}_{recon}^{local} = \|I^r - I^{recon}\|_1. \tag{10}$$

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 $\mathcal{E}^s$. Thus, the second loss term $\mathcal{L}_{align}$ comes as follows,

$$\mathcal{L}_{align} = \|f^{geo} - f^s\|_2. \tag{11}$$

Further, we utilize the third loss term – adversarial loss $\mathcal{L}_{adv}$, in a similar way as [42] do, by employing a discriminator $\mathcal{D}^r$, x

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

$$\mathcal{L}_{adv}^{\mathcal{D}} = \mathbb{E}[(\mathcal{D}^r(I^r) - 1))^2] + \mathbb{E}[(\mathcal{D}^r(I^{recon})^2]. \tag{13}$$

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

minimax game as follows,

$$\min_{\mathcal{E},\mathcal{R}} \mathcal{L}_{recon}^{local} + \lambda_{align}\mathcal{L}_{align} + \lambda_{adv}\mathcal{L}_{adv}^{\mathcal{E},\mathcal{R}}, \tag{14}$$

$$\min_{\mathcal{D}} \mathcal{L}_{adv}^{\mathcal{D}}. \tag{15}$$

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

Stage II: Training the *Geometry Graph*. 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_x^{geo} to f_y^{geo}. Therefore the loss is simply defined as an L2 loss between the predicted y geometry feature $f_y'^{geo} := e_{xy}(f_x^{geo})$ and the f_y^{geo}:

$$\min_{e_{xy}} \|f_y^{geo} - f_y'^{geo}\|_2. \tag{16}$$

Stage III: Joint Training of the *Global Generation Module* and the *Appearance Graph*. The Appearance Graph learns the style inter-dependency among facial components, with which we want to achieve appearance reasoning when observing the 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 $I^{final} = \mathcal{R}^{global}(F)$. The first loss is L1 reconstruction loss,

$$\mathcal{L}_{recon}^{global} = \|I^{final} - I\|_1. \tag{17}$$

We further employ VGG loss [45] and Lab loss [46] to constrain on the visual accuracy of generated images. Therefore, the training objective for this stage is as follows,

$$\min_{G^{app},\mathcal{R}^{global}} \mathcal{L}_{recon}^{global} + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_{Lab}\mathcal{L}_{Lab}. \tag{18}$$

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

4 EXPERIMENTS

4.1 Data Preparation

Using text as the interface of interaction requires us to prepare a dataset with high-quality facial images and accurate semantic annotations for each of them. For facial images, we generate a dataset with a capacity of 40K images using the attribute-conditioned sampling method provided by StyleFlow [10], where we impose constraints on the yaw and pitch during the generation process but randomize other attributes. This eliminates the impact of extreme poses and background since we desire the facial images to be as frontal as possible, enabling us to capture and model the geometry patterns of faces more accurately. Also, we use the StyleGAN generators provided by [47], which are trained on various other datasets. For training the *Feature Extraction Module*, we prepare the corresponding sketches for these images used during feature disentanglement following the method in [12].

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 the evaluation, we change and set the type for each attribute and calculate the accuracy of this attribute after generation. Results show that our method generates face images satisfying the semantic designations of the input text and surpasses the accuracy of previous arts [14], [51].

For facial attribute annotations, we use APIs from Face++ [48], Microsoft Azure [49], and Alibaba [50] since the detection of some desired attributes is only provided by one of these APIs. Unless otherwise specified, we set the resolution of generated images to 512×512 in all our experiments.

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 briefly explain why we only use frontal and occlusion-free faces:

- 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 projected to 2D.
- Non-frontal faces and occlusions would make it difficult for the detection API to make accurate judgments. Intuitively, for example, if the face has a big yaw/pitch, the arched eyebrow may look like a straight eyebrow, leading to misjudgment of the API.

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. 4.2.3), and perceptual study (Sec. 4.2.6). We also present more generated results in Fig. 6.

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 images 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 [48], [49], [50] 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 [14], [16], [51], and text-guided face manipulation [3], [28]. Since our work can be adapted to support face manipulation, we make comparisons for the two tasks.

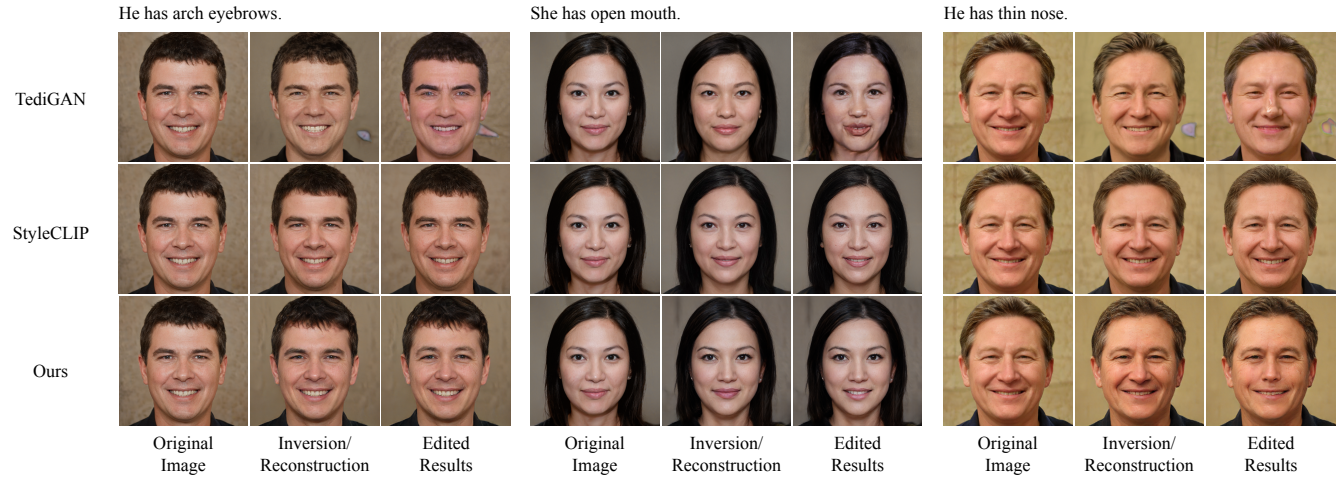


Fig. 5: **Editing comparisons with state-of-the-art methods.** We perform single-attribute editing for each example. In all three examples, TediGAN [28] fails to produce changes corresponding to the text-specified facial attributes. For StyleCLIP [3], it succeeds in turning a closed mouth into an opened one, while it also fails in 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 [6], [10], [43], [44] 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.

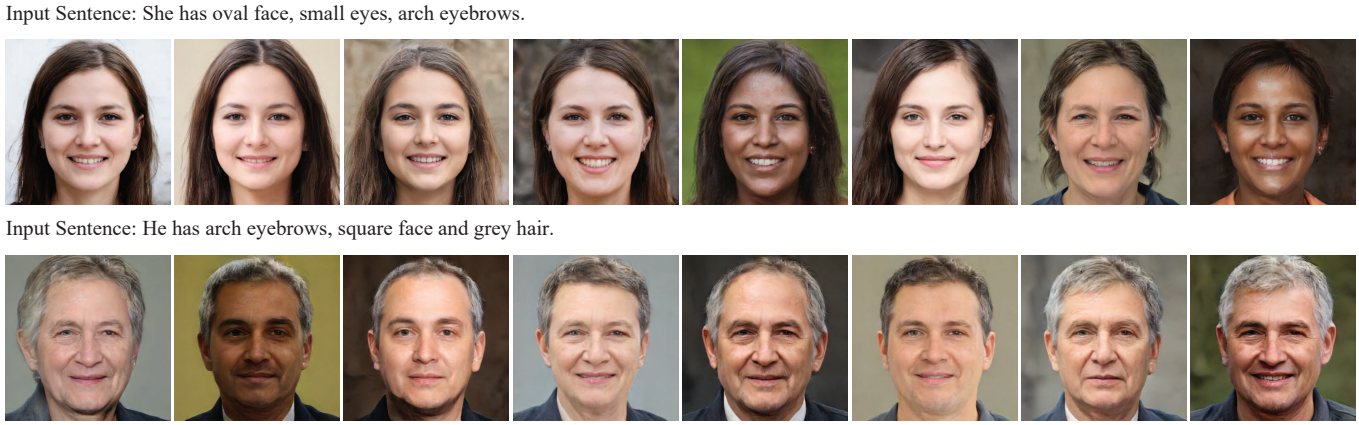


Fig. 6: **More end-to-end generated results.** Given the input sentences, our method can generate diverse faces conditioned on the prompts in the text.

For the generation task, we compare our method with AttnGAN [14] and DM-GAN [51] by retraining their models using the official implementations but with our dataset and setting the same sentence as the input to all three works. Since the original implementations of AttnGAN and DM-GAN 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. Specifically, for each image in our generated dataset, we randomly generate 10 sentences describing each face using the detected facial attributes, and retrain their model under the original resolution with our generated sentences. This is deemed as a fair comparison by us since generating images with a 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. In contrast, 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 this comparison is better than nothing since there do not exist relevant works under the same settings as ours: text-to-face generation with disentangled feature control. As shown in Fig. 9, our method's generation results are visually high-quality and semantically accurate. In contrast, results from previous text-to-image generation methods could not reach such a high resolution while also being deficient in satisfying the conditions in the input texts.

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 $\mathcal{E}^r$, and then substitute the features of specified editing attributes and perform graph recommendation upon the modified features. Here we compare our method with the two existing open-world-text-based editing methods [28], [52] for editing functionality and only compare the results of editing single attribute, because it is intuitive to perform multi-attribute editing

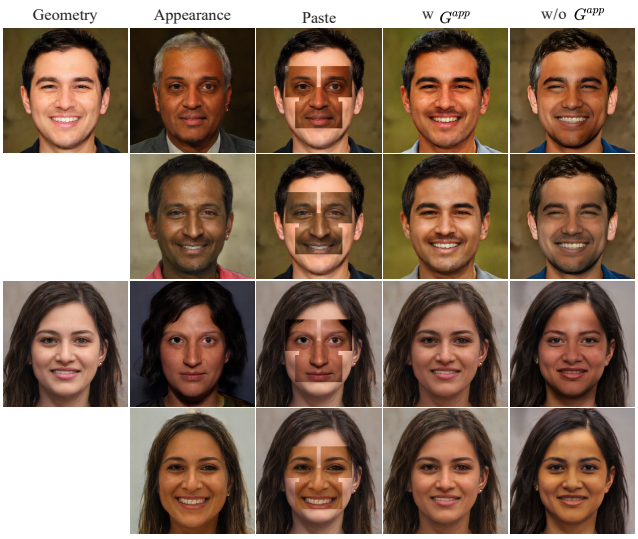


Fig. 7: **Ablation study of the *Appearance Graph*.** Editing the appearance of the source image (Geometry) using part-level reference images (Appearance). The Paste column shows the pasted appearance reference over the source image. As shown in the rightmost two columns, the edited results with *Appearance Graph*are much more color-consistent than the rightmost column where the results are generated without incorporating the *Appearance Graph*.

by serializing the editing processes of single-attribute editing. We use the standard optimization-based method in [28] and the Global Direction method in [3] for comparison. As shown in Fig. 5, editing results from TediGAN [28] often fail to convey the semantics indicated in the input texts. Obvious artifacts exist in those results as well. StyleCLIP [3] synthesizes more meaningful editing results, as shown in the middle example in Fig. 5. However, in the left and right examples, it fails to generate apparent editing effects, *i.e.*the eyebrows in the left example are not "arch" and the nose in the right example is not "thin". Our method, on the other hand, generates semantically consistent and visually meaningful results for all examples shown in Fig. 5.

4.2.3 Ablation Study

Graph Recommendation Module is an essential part of 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 the 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. 10, we fix f_{bg}^{geo} and keep changing f_{eye1}^{geo} and f_{eye2}^{geo}. In this way, the *Geometry Graph* is expected to predict f_{nose}^{geo} and f_{mouth}^{geo} 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. 7. While without *Appearance Graph*, the swapping operation produces images with inconsistent color.

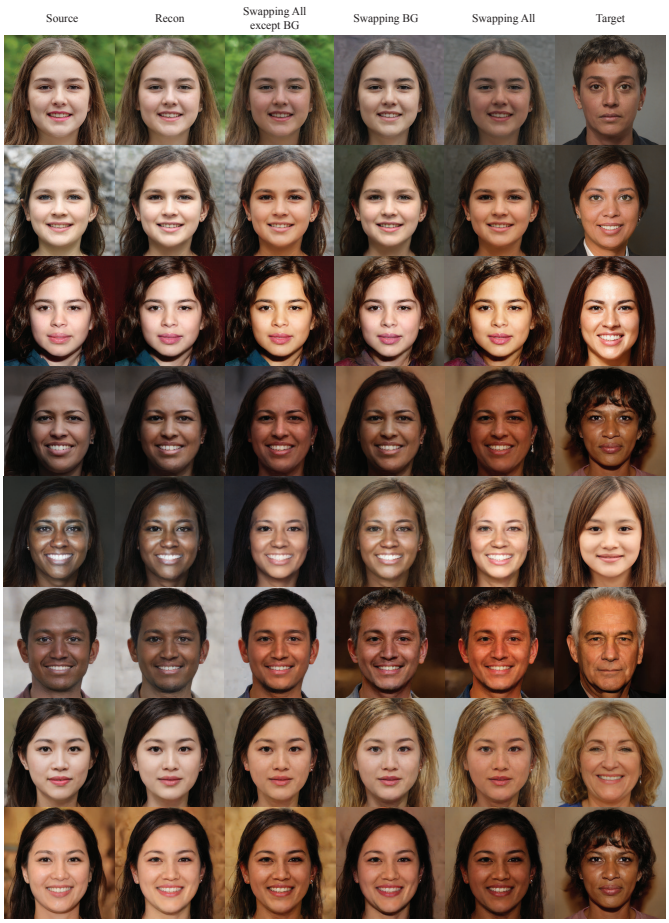


Fig. 8: **Results of partial appearance morphing with *Appearance Graph*.** The third column shows results generated from the geometry features from the source image (shown on the leftmost column), *bg* appearance feature from the source image, and other partial appearance features from the target image on the rightmost column. It keeps the background appearance nearly intact while shifting the facial color toward that of the target image as much as possible. The fourth column is the opposite, where the *bg* appearance feature comes from the target image while the rest appearance features inherit from the source image. The corresponding effect is the maintenance of facial appearance and swapping of the background appearance. Note that the background appearance here includes the hair color feature.

4.2.4 Geometry and Appearance Morphing

The encoder network $\mathcal{E}^r$ 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. 11, 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.

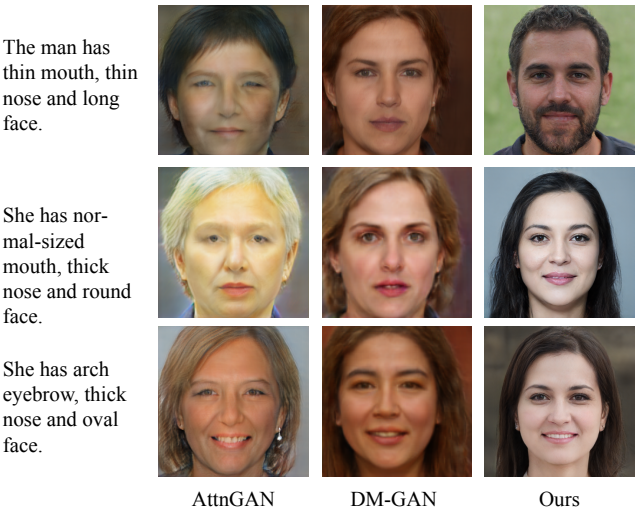


Fig. 9: **Generation comparisons with state-of-the-art methods.** Given the same input sentence (leftmost in each example), our result is significantly better than the other two methods in terms of both image quality and attribute accuracy.

	[14]	[51]	[28]	[3]	Ours
mean	4.22	4.48	6.22	7.28	7.84
p	1.81e-14	1.46e-13	8.06e-9	1.70e-3	

TABLE 2: **T-test results for text-image correspondence perceptual study.** The results further prove the superiority of our method on text-image correspondence over various alternatives, with a higher mean score in the perceptual study as well as a significantly small p-value in paired t-tests.

	[14]	[51]	[28]	[3]	Ours
mean	3.98	4.35	6.79	7.46	7.46
p	1.65e-15	9.45e-17	7.26e-7	0.958	

TABLE 3: **T-test results for image quality perceptual study.** The results show that the generated image quality of our method is comparable to that of StyleGAN generators [3], while significantly better than that of [14], [51]. Note that although [28] also uses StyleGAN generators, their score is significantly lower than ours. This may be attributed to the fact that the optimized latent codes from their methods sometimes generate out-of-distribution images which exhibit obvious artifacts, as shown in Fig. 5.

.

4.2.5 Partial Appearance Morphing with Appearance Graph

In Fig. 8, we demonstrate the potential capability of the *Appearance Graph* in the *Graph Recommendation Module*, where by substituting some of the appearance features of the input image with those of the target image, we could get face images with blending appearances. Specifically, if we swap the part-level appearance features of $\{nose, mouth, leye, reye\}$, the *Appearance Graph* could propagate the appearance features it infers from these nodes to the whole face while maintaining the appearance of bg intact. On the other hand, by swapping the appearance feature of only bg, we get the skin color maintained while the hair color changed.

4.2.6 Perceptual Study

To evaluate the faithfulness of the synthesized results with respect to the given sentence, we prepared two user studies and asked human users to judge the effectiveness of our method in comparison with the existing solutions from two aspects: text-image correspondence and quality of generated images. The box plots of these two user study statistics are illustrated in Fig. 12.

The evaluation was done via an online questionnaire to evaluate the correspondence between the given sentence and the synthesized results according to user perception. For each sample, we show a sentence depicting a portrait and 5 images synthesized by the compared methods, including AttnGAN [14], DM-GAN [51], TediGAN [28] and StyleCLIP [3], and ours. To avoid bias and ensure the score fairness, we place those images randomly. Each participant is asked to evaluate 10 examples according to 2 criteria: the text-image correspondence and the quality of the generated images, each of them on a 10-point Likert scale (1 = strongly negative to 10 = strong positive). We invite 20 participants to participate in this study and get 20 (participants) $\times$ 10 (questions) = 200 subjective evaluations for each method. The statistics are shown in the supplementary materials. We perform one-way ANOVA tests on the 5 methods mentioned above and find significant effects for text-image correspondence ($F_{(4,45)} = 229.48, p < 0.001$) and for image quality ($F_{(4,45)} = 360.23, p < 0.001$). The results from paired t-tests further testify the effectiveness of our method over the other methods, as illustrated in Tables 2 and 3.

4.3 Failure Modes

Since our dataset only contains frontal faces, the failure modes of our pipeline mainly center around the non-frontal issue and occlusion issue. Another failure mode is the issue of generating/reconstructing complicated hair styles such as wavy hair, plate hair, bangs, etc.. We initially test our model on the CelebA [13] and FFHQ [5] datasets but achieve under-expected results, and we believe that this could attribute to the three failure modes mentioned above. More failure examples are illustrated in Fig. 13. As shown in Fig. 13, the left image contains unclear boundaries between ears and hair, the middle image and the right image both show artifacts around the ear which make it look like wearing ear rings.

5 Limitations

The motivation for 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 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 hairstyles such as wavy hair, plate hair, bangs, etc. Thus it could not generate faces with such hairstyles. We refer to the readers to [46], [53] 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 work to generate and check their validity. Limited by the diversity encoded in the StyleGAN generator, our database

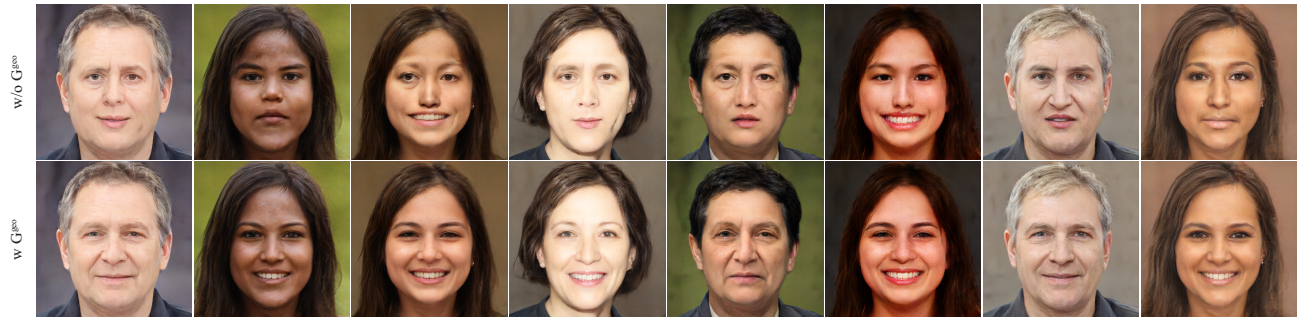


Fig. 10: **Ablation study of the *Geometry Graph*.** We randomly sample facial geometry features to generate face images. The upper row shows the results generated from geometry features without being optimized by the *Geometry Graph*, and the lower row shows the results generated using *Geometry Graph*. Obviously, there exist artifacts on the borders 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, producing more realistic results.

Fig. 11: **Interpolation via Geometry and Appearance Axes.** The appearance gradually changes along each column, while the geometry changes along each row.

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 have presented 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 demonstrated the effectiveness of our method by comparing it 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-world scenario. However, our current pipeline may not apply to complex sentences. Generation from sentences with more fuzzy descriptions is to be adapted in the future.

References

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

[2] 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.

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

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

[5] 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)*, 2019, pp. 4396–4405.

[6] 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.

[7] 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)*, 2020, pp. 8107–8116.

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

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

[10] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows," *CoRR*, vol. abs/2008.02401, 2020. [Online]. Available: https://arxiv.org/abs/2008.02401

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

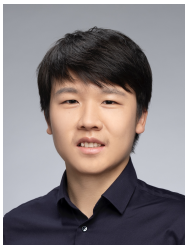
[16] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Fig. 12: **Two box plots for illustrating the statistics from the perceptual studies.** As has been discussed in Section 4.2.6, the face images generated by our model are rated the highest score in the user study of text-image correspondence. Such a result may benefit from the part-level control of each facial attribute. In terms of the quality of the generated images, our model generates high-fidelity images whose quality could be compared with that of StyleGAN [5]. The reason why TediGAN [28]'s image quality is rated lower than StyleCLIP [3] may attribute to the edited latent codes, which lead to generating visual artifacts in the synthetic images.

Fig. 13: **Less successful cases.** The boundary between the ears and the hair is not clear in these images. Moreover, the curly hair styles in the middle image and the right image are not perfectly handled. We believe that these two artifacts are due to the limited capability of *Global Generation Module* of understanding such detailed spatial information from the input feature maps. As discussed in Sec. 4.3, .

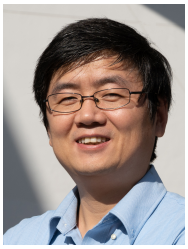
[17] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao, "Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.
[19] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.
[20] A. R. et al, "Hierarchical text-conditional image generation with clip latents," 2022.
[21] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," 2023.
[22] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," 2023.
[23] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," 2023.
[24] 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.
[25] G. Kwon and J. C. Ye, "Clipstyler: Image style transfer with a single text condition," *arXiv preprint arXiv:2112.00374*, 2021.
[26] 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.
[27] 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)*, June 2022, pp. 13 492–13 502.
[28] 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.
[29] J. Sun, Q. Deng, Q. Li, M. Sun, M. Ren, and Z. Sun, "Anyface: Free-style text-to-face synthesis and manipulation," 2022.
[30] Z. Huang, K. C. Chan, Y. Jiang, and Z. Liu, "Collaborative diffusion for multi-modal face generation and editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
[31] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
[32] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 2426–2435.
[33] Y. Ma, H. Yang, W. Wang, J. Fu, and J. Liu, "Unified multi-modal latent diffusion for joint subject and text conditional image generation," 2023.
[34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.
[35] J. N. M. Pinkney and C. Li, "clip2latent: Text driven sampling of a pre-trained stylegan using denoising diffusion and clip," 2022.
[36] 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*, 2019, pp. 6609–6618.
[37] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.
[38] B. Zhu, C. Lin, Q. Wang, R. Liao, and C. Qian, "Fast and accurate: Structure coherence component for face alignment," 2020.
[39] 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.
[40] 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.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
[42] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," 2017.
[43] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," in *Proc. NeurIPS*, 2020.
[44] 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.
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
[46] 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)*, vol. 39, no. 4, pp. 1–13, 2020.
[47] seeprettyface, "seeprettyface.com," 2020. [Online]. Available: seeprettyface.com
[48] Face++, "https://www.faceplusplus.com.cn/face-detection/," 2020. [Online]. Available: https://www.faceplusplus.com.cn/face-detection/
[49] Microsoft, "https://docs.microsoft.com/en-in/azure/cognitive-services/face/," 2020. [Online]. Available: https://docs.microsoft.com/en-in/azure/cognitive-services/face/
[50] Alibaba, "https://help.aliyun.com/document_detail/130846.html," 2020. [Online]. Available: https://help.aliyun.com/document_detail/130846.html
[51] 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*. Computer Vision Foundation / IEEE, 2019, pp. 5802–5810. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Zhu_DM-GAN_Dynamic_Memory_Generative_Adversarial_Networks_for_Text-To-Image_Synthesis_CVPR_2019_paper.html
[52] 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.
[53] C. Xiao, D. Yu, X. Han, Y. Zheng, and H. Fu, "Sketchhairsalon: Deep sketch-based hair image synthesis," 2021.



Zhaoyang Zhang is a Ph.D. student at Yale Computer Graphics Group. His research interests lie in 2D and 3D digital content creation. Prior to joining Yale, he obtained his B.Eng. (summa cum laude) in Computer Science and Technology from the University of Chinese Academy of Sciences (UCAS) in June 2022.

Junliang Chen is currently a master candidate in the Department of Film and TV Technology, Beijing Film Academy. His research interests include digital film technology.



Hongbo Fu received a BS degree in information sciences from Peking University, China, in 2002 and a PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. He is a Full Professor at the School of Creative Media, City University of Hong Kong. His primary research interests fall in the fields of computer graphics and human computer interaction. He has served as an Associate Editor of The Visual Computer, Computers & Graphics, and Computer Graphics Forum.

Jianjun Zhao is an associate professor in the Department of Film and TV Technology, Beijing Film Academy. He received his Ph.D degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences. His research focuses on film virtual production and physics-based character animation.

Shu-Yu Chen received her PHD degree in Computer Science and Technology from the University of Chinese Academy of Sciences. She is currently working as an Assistant Professor at the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics.

Lin Gao received his PhD degree in computer science from Tsinghua University. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young researcher award. His research interests include computer graphics and geometric processing.