Talk-to-Edit: Fine-Grained Facial Editing via Dialog

Yuming Jiang[1*] Ziqi Huang[1*] Xingang Pan[2] Chen Change Loy[1] Ziwei Liu[1✉]
[1]S-Lab, Nanyang Technological University [2]The Chinese University of Hong Kong
{yuming002, hu0007qi, ccloy, ziwei.liu}@ntu.edu.sg px117@ie.cuhk.edu.hk

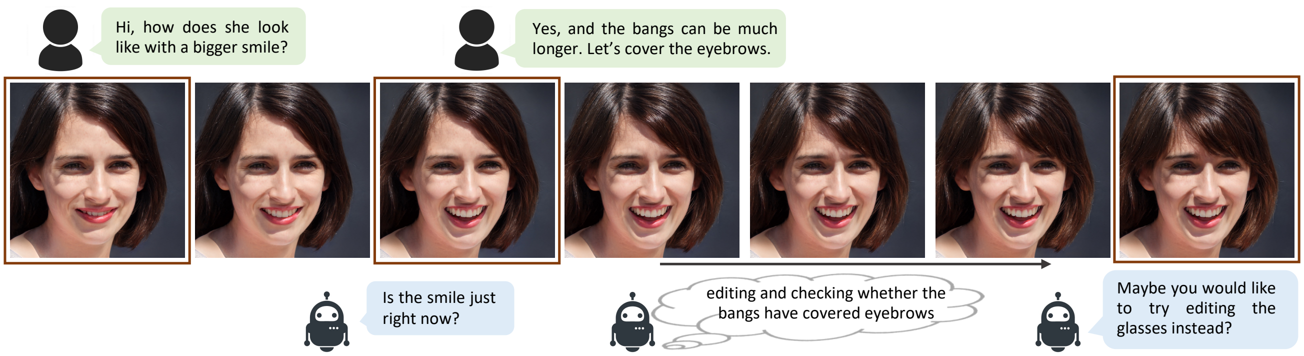


Figure 1: **An example of *Talk-to-Edit***. The user provides a facial image and an editing request. Our system then edits the image accordingly, and provides meaningful language feedback such as clarification or alternative editing suggestions. During editing, the system is able to control the extent of attribute change on a fine-grained scale, and iteratively checks whether the current editing step fulfills the user's request.

Abstract

*Facial editing is an important task in vision and graphics with numerous applications. However, existing works are incapable to deliver a continuous and fine-grained editing mode (e.g., editing a slightly smiling face to a big laughing one) with natural interactions with users. In this work, we propose **Talk-to-Edit**, an interactive facial editing framework that performs fine-grained attribute manipulation through dialog between the user and the system. Our key insight is to model a continual "semantic field" in the GAN latent space. **1)** Unlike previous works that regard the editing as traversing straight lines in the latent space, here the fine-grained editing is formulated as finding a curving trajectory that respects fine-grained attribute landscape on the semantic field. **2)** The curvature at each step is location-specific and determined by the input image as well as the users' language requests. **3)** To engage the users in a meaningful dialog, our system generates language feedback by considering both the user request and the current state of the semantic field.*

*We also contribute **CelebA-Dialog**, a visual-language facial editing dataset to facilitate large-scale study. Specifically, each image has manually annotated fine-grained attribute annotations as well as template-based textual descriptions in natural language. Extensive quantitative and qualitative experiments demonstrate the superiority of our framework in terms of **1)** the smoothness of fine-grained editing, **2)** the identity/attribute preservation, and **3)** the visual photorealism and dialog fluency. Notably, user study validates that our overall system is consistently favored by around 80% of the participants. Our project page is https://www.mmlab-ntu.com/project/talkedit/.*

1. Introduction

The goal of facial editing is to enable users to manipulate facial images in their desired ways. Thanks to the advance of deep generative models like GANs [10, 29, 3, 15, 16, 18], facial editing has witnessed rapid growth in recent years, especially in image fidelity. While there have been several attempts to improve facial editing quality, they often lack interactions with users or require users to follow some fixed control patterns. For instance, image-to-image translation models [53, 7, 12, 21, 26] only translate facial images between several discrete and fixed states, and users cannot give any subjective controls to the system. Other face editing methods offer users some controls, such as a semantic map indicating the image layout [22], a reference image demonstrating the target style [14, 25, 24], and a sentence describing a desired effect [5, 51, 30, 54, 46]. However, users have to follow the fixed patterns, which are too demanding and inflexible for most users. Besides, the only

*Equal contribution.

feedback provided by the system is the edited image itself.

In terms of the flexibility of interactions, we believe natural language is a good choice for users. Language is not only easy to express and rich in information, but also a natural form for the system to give feedback. Thus, in this work, we make the first attempt towards a dialog-based facial editing framework, namely **Talk-to-Edit**, where editing is performed round by round via request from the user and feedback from the system.

In such an interactive scenario, users might not have a clear target in their mind at the beginning of editing and thoughts might change during editing, like tuning an overly laughing face back to a moderate smile. Thus, the editing system is supposed to be capable of performing continuous and fine-grained attribute manipulations. While some approaches [37, 38, 42, 39, 11] could perform continuous editing to some extent by shifting the latent code of a pretrained GAN [16, 18, 15, 3], they typically make two assumptions: 1) the attribute change is achieved by traversing along a straight line in the latent space; 2) different identities share the same latent directions. However, these assumptions overlook the non-linear nature of the latent space of GAN, potentially leading to several shortcomings in practice: **1)** The identity would drift during editing; **2)** When editing an attribute of interest, other irrelevant attributes would be changed as well; **3)** Artifacts would appear if the latent code goes along the straight line too far.

To address these challenges, we propose to learn a *vector field* that describes *location-specific* directions and magnitudes for attribute changes in the latent space of GAN, which we term as a "semantic field". Traversing along the curved trajectory takes into account the non-linearity of attribute transition in the latent space, thus achieving more fine-grained and accurate facial editing. Besides, the curves changing the attributes of different identities might be different, which can also be captured by our semantic field with the location-specific property. In this case, the identity of the edited facial image would be better preserved. In practice, the semantic field is implemented as a mapping network, and is trained with fine-grained labels to better leverage its location-specific property, which is more expressive than prior methods supervised by binary labels.

The above semantic field editing strategy is readily embedded into our dialog system to constitute the whole *Talk-to-Edit* framework. Specifically, a user's language request is encoded by a language encoder to guide the semantic field editing part to alter the facial attributes consistent with the language request. After editing, feedback would be given by the system conditioned on previous edits to check for further refinements or offer other editing suggestions. The user may respond to the system feedback for further editing actions, and this dialog-based editing iteration would continue until the user is satisfied with the edited results.

To facilitate the learning of semantic field and dialog-based editing, we contribute a large-scale visual-language dataset named **CelebA-Dialog**. Unlike prior datasets with only binary attribute labels, we annotate images in CelebA with attribute labels of fine granularity. Accompanied with each image, there is also a user request sample and several captions describing these fine-grained facial attributes.

In summary, our main contributions are: **1)** We propose to perform fine-grained facial editing via dialog, an easier interactive way for users. **2)** To achieve more continuous and fine-grained facial editing, we propose to model a location-specific semantic field. **3)** We achieve superior results with better identity preservation and smoother change compared to other counterparts. **4)** We contribute a large-scale visual-language dataset **CelebA-Dialog**, containing fine-grained attribute labels and textual descriptions.

2. Related Work

Semantic Facial Editing. Several methods have been proposed for editing specific attributes such as age progression [49, 44], hair synthesis [31, 47], and smile generation [43]. Unlike these attribute-specific methods relying on facial priors such as landmarks, our method is able to manipulate multiple semantic attributes without using facial priors. Image-to-image translation methods [53, 7, 12, 21, 26] have shown impressive results on facial editing. However, they are insufficient to perform continuous editing because images are translated between two discrete domains.

Recently, latent space based manipulation methods [52, 4] are drawing increasing attention due to the advancement of GAN models like StyleGAN [17, 19]. These approaches typically discover semantically meaningful directions in the latent space of a pretrained GAN so that moving the latent code along these directions could achieve desired editing in the image space. Supervised methods find directions to edit the attributes of interest using attribute labels [37, 38, 55], while unsupervised methods exploit semantics learned by the pretrained GAN to discover the most important and distinguishable directions [42, 11, 39]. InterFaceGAN [37, 38] finds a hyperplane in the latent space to separate semantics into a binary state and then uses the normal vector of the hyperplane as the editing direction. A recent work [55] learns a transformation supervised by binary attribute labels and directly adds the transformation direction to the latent code to achieve one-step editing. Some approaches [13, 1] consider the non-linear property of latent space. Different from existing methods, we learn a location-specific field in the latent space supervised by fine-grained labels to achieve precise fine-grained editing and to preserve facial identities.

Language-based Image Editing. The flexibility of natural language has attracted researchers to propose a number of text-to-image generation [50, 34, 48, 46] and manipulation [5, 51, 30, 54, 46] approaches. For example,

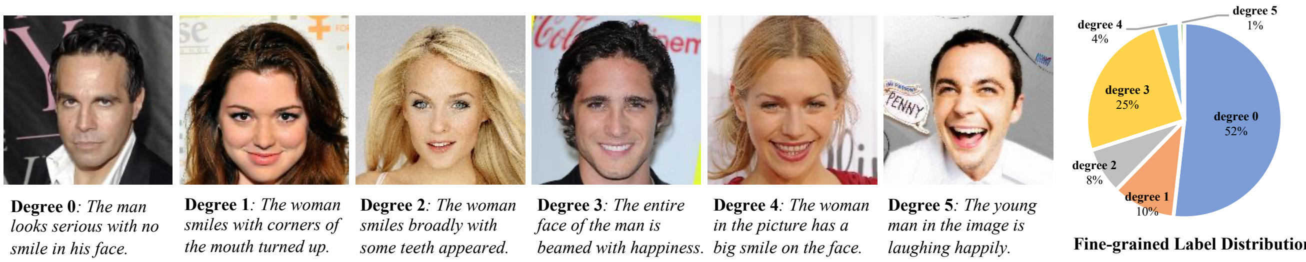


Figure 2: **Illustration of *CelebA-Dialog* dataset.** We show example images and annotations for the smiling attribute. Below the images are the attribute degrees and the corresponding textual descriptions. We also show the fine-grained label distribution of the smiling attribute.

given an input image, TediGAN [46] generates a new image conditioned on a text description. Some other approaches [40, 6, 20, 2, 9, 27, 23] allow users to give requests in the form of natural language but do not provide meaningful feedback, clarification, suggestion, or interaction. Chatpainter [36] synthesizes an image conditioned on a completed dialog, but could not talk to users round by round to edit images. Unlike existing systems that simply "listen" to users to edit, our dialog-based editing system is able to "talk" to users, edit the image according to user requests, clarify with users about their intention especially on fine-grained attribute details, and offer other editing options for users to explore.

3. CelebA-Dialog Dataset

In the dialog-based facial editing scenarios, many rounds of edits are needed till users are satisfied with the edited images. To this end, the editing system should be able to generate continuous and fine-grained facial editing results, which contain intermediate states translating source images to target images. However, for most facial attributes, binary labels are not enough to precisely express the attribute degrees. Consequently, methods trained with only binary labels could not perform natural fine-grained facial editing. Specifically, they are not able to generate plausible results when attribute degrees become larger. Thus, fine-grained facial attribute labels are vital to providing supervision for fine-grained facial editing. Moreover, the system should also be aware of the attribute degrees of edited images so that it could provide precise feedback or suggestions to users, which also needs fine-grained labels for training.

Motivated by these, we contribute a large-scale visual-language face dataset named **CelebA-Dialog**. The **CelebA-Dialog** dataset has the following properties: **1)** Facial images are annotated with rich fine-grained labels, which classify one attribute into multiple degrees according to its semantic meaning; **2)** Accompanied with each image, there are captions describing the attributes and a user request sample. The **CelebA-Dialog** dataset is built as follows:

Data Source. CelebA dataset [28] is a well-known large-scale face attributes dataset, which contains 202,599 images. With each image, there are forty binary attribute annotations. Due to its large-scale property and diversity, we choose to annotate fine-grained labels for images in CelebA dataset. Among forty binary attributes, we select five attributes whose degrees cannot be exhaustively expressed by binary labels. The selected five attributes are Bangs, Eyeglasses, Beard, Smiling, and Young (Age).

Fine-grained Annotations. For Bangs, we classify the degrees according to the proportion of the exposed forehead. There are 6 fine-grained labels in total: 100%, 80%, 60%, 40%, 20%, and 0%. The fine-grained labels for eyeglasses are annotated according to the thickness of glasses frames and the type of glasses (ordinary / sunglasses). The annotations of beard are labeled according to the thickness of the beard. And the metrics for smiling are the ratio of exposed teeth and open mouth. As for the age, we roughly classify the age into six categories: below 15, 15-30, 30-40, 40-50, 50-60, and above 60. In Fig. 2, we provide examples on the fine-grained annotations of the smiling attribute. For more detailed definitions and examples of fine-grained labels for each attribute, please refer to the supplementary files.

Textual Descriptions. For every image, we provide fine-grained textual descriptions which are generated via a pool of templates. The captions for each image contain one caption describing all the five attributes and five individual captions for each attribute. Some caption examples are given in Fig. 2. Besides, for every image, we also provide an editing request sample conditioned on the captions. For example, a serious-looking face is likely to be requested to add a smile.

4. Our Approach

The pipeline of **Talk-to-Edit** system is depicted in Fig. 3. The whole system consists of three major parts: user request understanding, semantic field manipulation, and system feedback. The initial inputs to the whole system are an image $\mathbf{I}$ and a user's language request $\boldsymbol{r}$. A language encoder E is first employed to interpret the user request into the editing encoding $\boldsymbol{e}_r$, indicating the attribute of interest, changing directions, etc. Then the editing encoding $\boldsymbol{e}_r$ and the corresponding latent code $\boldsymbol{z}$ is fed into the "semantic field" F to find the corresponding vectors $\boldsymbol{f}_z$ to change the

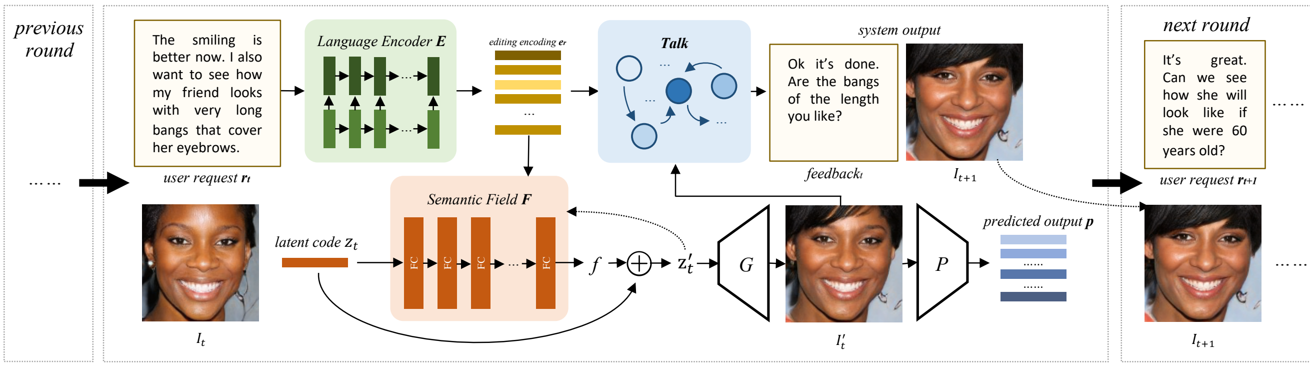


Figure 3: **Overview of *Talk-to-Edit* Pipeline.** In round t, we receive the input image $\mathbf{I}_t$ and its corresponding latent code $\boldsymbol{z}_t$ from the last round. Then the *Language Encoder* E extracts the editing encoding $\boldsymbol{e}_r$ from the user request $\boldsymbol{r}_t$, and feeds $\boldsymbol{e}_r$ to the *Semantic Field* F to guide the editing process. The latent code $\boldsymbol{z}_t$ is iteratively moved along field lines by adding the field vector $\boldsymbol{f} = F(\boldsymbol{z}_t)$ to $\boldsymbol{z}_t$, and a pretrained predictor is used to check whether the target degree is achieved. Finally, the edited image $\mathbf{I}_{t+1}$ will be output at the end of one round. Based on the editing encoding $\boldsymbol{e}_r$, the $Talk$ module gives language feedback such as clarification and alternative editing suggestions.

specific attribute degrees. After one round of editing, the system will return the edited image $\mathbf{I}'$ and provide reasonable feedback to the user. The editing will continue until the user is satisfied with the editing result.

4.1. User Request Understanding

Given a user's language request $\boldsymbol{r}$, we use a language encoder E to extract the editing encoding $\boldsymbol{e}_r$ as follows:

$$\boldsymbol{e}_r = E(\boldsymbol{r}) \tag{1}$$

The editing encoding $\boldsymbol{e}_r$, together with the dialog and editing history, and the current state of the semantic field, will decide and instruct the semantic field whether to perform an edit in the current round of dialog. The editing encoding $\boldsymbol{e}_r$ contains the following information: **1)** request type, **2)** the attribute of interest, **3)** the editing direction, and **4)** the change of degree.

Users' editing requests are classified into three types: **1)** describe the attribute and specify the target degree, **2)** describe the attribute of interest and indicate the relative degree of change, **3)** describe the attribute and only the editing direction without specifying the degree of change. We use template-based method to generate the three types of user requests and then train the language encoder.

4.2. Semantic Field for Facial Editing

Given an input image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ and a pretrained GAN generator G, similar to previous latent space based manipulation methods [37, 38, 55, 32], we need to firstly inverse the corresponding latent code $\boldsymbol{z} \in \mathbb{R}^d$ such that $\mathbf{I} = G(\boldsymbol{z})$, and then find the certain vector $\boldsymbol{f}_z \in \mathbb{R}^d$ which can change the attribute degree. Note that adopting the same vector for all faces is vulnerable to identity change during editing, as different faces could have different $\boldsymbol{f}_z$. Thus, the vector should be *location-specific*, *i.e.*, the vector is not only unique to different identities but also varies during editing. Motivated by this, we propose to model the latent space as a continual "semantic field", *i.e.*, a vector field that assigns a vector to each latent code.

Definition of Continual Semantic Field. For a latent code $\boldsymbol{z}$ in the latent space, suppose its corresponding image $\mathbf{I}$ has a score s for a certain attribute. By finding a proper vector $\boldsymbol{f}_z$ and then adding the vector to $\boldsymbol{z}$, the attribute score s will be changed to s'. Intuitively, the vector $\boldsymbol{f}_z$ to increase the attribute score for the latent code $\boldsymbol{z}$ is the gradient of s with respect to $\boldsymbol{z}$.

Mathematically, the attribute score is a scalar field, denoted as $S : \mathbb{R}^d \mapsto \mathbb{R}$. The gradient of attribute score field S with respect to the latent code is a vector field, which we term as "semantic field". The semantic field $F : \mathbb{R}^d \mapsto \mathbb{R}^d$ can be defined as follows:

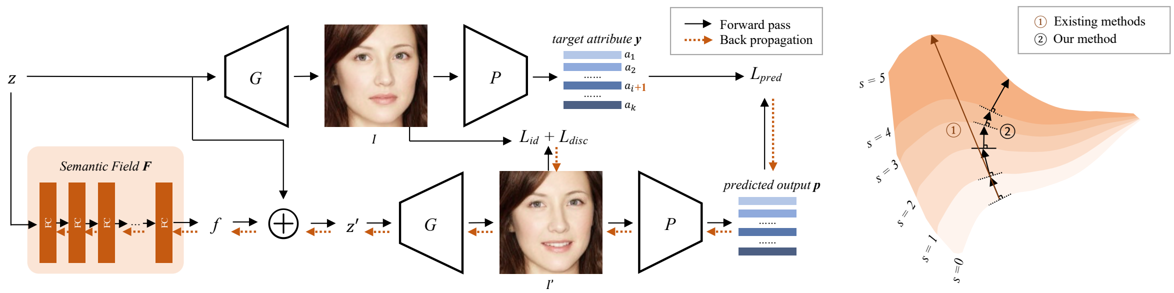
$$F = \nabla S. \tag{2}$$

For a specific latent code $\boldsymbol{z}$, the direction of its semantic field vector $\boldsymbol{f}_z$ is the direction in which the attribute score s increases the fastest.

In the latent space, if we want to change the attribute score s of a latent code $\boldsymbol{z}$, all we need is to move $\boldsymbol{z}$ along the latent direction in the semantic field. Due to the *location-specific* property of the semantic field, the trajectory of changing the attribute score from s_a to s_b is curved. The formula for changing attribute score is expressed as:

$$s_a + \int_{\boldsymbol{z}_a}^{\boldsymbol{z}_b} \boldsymbol{f}_z \cdot d\boldsymbol{z} = s_b, \tag{3}$$

where $\boldsymbol{z}_a$ is the initial latent code and $\boldsymbol{z}_b$ is the end point. As the semantic field is continuous and location-specific, continuous facial editing can be easily achieved by traversing the latent space along the semantic field line.



(a) Training Scheme of *Semantic Field* (b) *Semantic Field* in Latent Space

Figure 4: **(a) Training Scheme of *Semantic Field*.** Predictor loss, identity keeping loss and discriminator loss are adopted to ensure the location-specific property of semantic field. **(b) Illustration of *Semantic Field* in Latent Space.** Different colors represent latent space regions with different attribute scores. The boundary between two colored regions is an equipotential subspace. Existing methods are represented by the trajectory ①, where latent code is shifted along a fixed direction throughout editing. Our method is represented by trajectory ②, where latent code is moved along location-specific directions.

Discretization of Semantic Field. Though the attribute score field and semantic field in the real world are both continual, in practice, we need to discretize the continual field to approximate the real-world continual one. Thus, the discrete version of Eq. (3) can be expressed as:

$$s_a + \sum_{i=1}^{N} \boldsymbol{f}_{z_i} \cdot \Delta \boldsymbol{z}_i = s_b. \tag{4}$$

The semantic field F is implemented as a mapping network. For a latent code $\boldsymbol{z}$, we could obtain its corresponding semantic field vector via $\boldsymbol{f}_z = F(\boldsymbol{z})$. Then one step of latent code shifting is achieved by:

$$\begin{aligned} \boldsymbol{z}' &= \boldsymbol{z} + \alpha \boldsymbol{f}_z \\ &= \boldsymbol{z} + \alpha F(\boldsymbol{z}), \end{aligned} \tag{5}$$

where α is the step size, which is set to $\alpha = 1$ in this work. Since $\boldsymbol{f}_z$ is supposed to change the attribute degree, the edited image $\mathbf{I}' = G(\boldsymbol{z}')$ should have a different attribute score from the original image $\mathbf{I} = G(\boldsymbol{z})$. During editing, we repeat Eq. (5) until the desired attribute score is reached.

As illustrated in Fig. 4, to train the mapping network so that it has the property of a semantic field, a pretrained fine-grained attribute predictor P is employed to supervise the learning of semantic field. The predictor has two main functions: one is to push the output vector to change the attribute of interest in a correct direction, and the other is to keep the other irrelevant attributes unchanged. Suppose we have k attributes in total. The fine-grained attributes of the original image can be denoted as $(a_1, a_2, ..., a_i, ..., a_k)$, where $a_i \in \{0, 1, ..., C\}$ are the discrete class labels indicating the attribute degree. When we train the semantic field for the i-th attribute, the target attributes labels y of the edited image $\mathbf{I}'$ should be $(a_1, a_2, ..., a_i + 1, ..., a_k)$. With the target attribute labels, we can optimize the desired semantic field using the cross-entropy loss, then the predictor loss L_{pred} is expressed as follows:

$$L_{pred} = -\sum_{i=1}^{k} \sum_{c=0}^{C} y_{i,c} log(p_{i,c}), \tag{6}$$

where C denotes the number of fine-grained classes, $y_{i,c}$ is the binary indicator with respect to the target class, and $p_{i,c}$ is the softmax output of predictor P, *i.e.*, $p = P(\mathbf{I}')$.

As the *location-specific* property of the semantic field allows different identities to have different vectors, we further introduce an identity keeping loss [45, 41] to better preserve the face identity when shifting the latent codes along the semantic field. Specifically, we employ an off-the-shelf face recognition model to extract discriminative features, and the extracted features during editing should be as close as possible. The identity keeping loss L_{id} is defined as follows:

$$L_{id} = \left\| Face(\mathbf{I}') - Face(\mathbf{I}) \right\|_1, \tag{7}$$

where $Face(\cdot)$ is the pretrained face recognition model [8].

Moreover, to avoid unrealistic artifacts in edited images, we could further leverage the pretrained discriminator D coupled with the face generator as follows:

$$L_{disc} = -D(\mathbf{I}'). \tag{8}$$

To summarize, we use the following loss functions to supervise the learning of semantic field:

$$L_{total} = \lambda_{pred} L_{pred} + \lambda_{id} L_{id} + \lambda_{disc} L_{disc}, \tag{9}$$

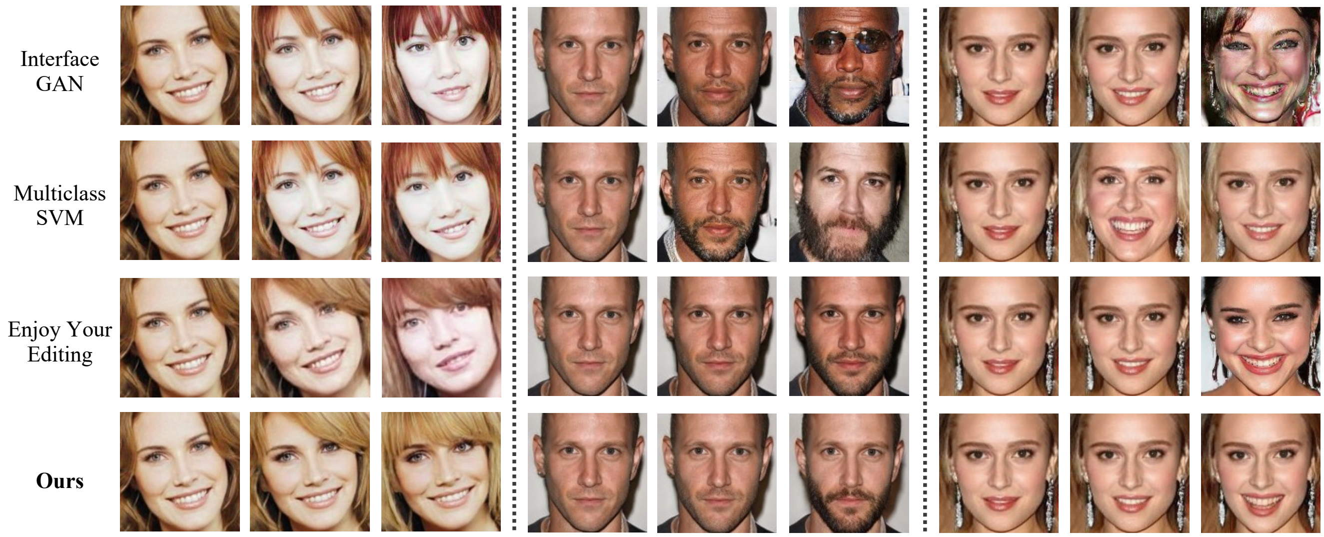
where λ_{pred}, λ_{id} and λ_{disc} are weights for predictor loss, identity keeping loss and discriminator loss respectively.

4.3. System Feedback

The system *Talk* module provides natural language feedback as follows:

$$feedback_t = Talk(feedback_{t-1}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{e}_r, \boldsymbol{h}), \tag{10}$$

where $\boldsymbol{r}$ is the user request, $\boldsymbol{s}$ is the current system state, $\boldsymbol{e}_r$ is the editing encoding, and $\boldsymbol{h}$ is the editing history.



(a) Bangs (b) Beard (c) Smiling

Figure 5: **Qualitative Comparison.** We compare our approach with InterfaceGAN, Multiclass SVM and Enjoy Your Editing. Our editing results are more realistic. Besides, our method is less likely to change the identity and other attributes.

The feedback provided by the system comes from one of the three categories: **1)** checking if the attribute degree of the edited image meets users' expectations, **2)** providing alternative editing suggestions or options, and **3)** asking for further user instructions.

5. Experiments

Evaluation Datasets. We synthesize the evaluation dataset by sampling latent codes from the StyleGAN pretrained on CelebA dataset [28]. Using latent codes, we then generate corresponding images. When comparing with other latent space based manipulation methods, we use the latent code for editing directly to avoid the error introduced by GAN-inversion methods. Considering computation resources, we compare our method with baselines on 128×128 images.

Evaluation Metrics. We evaluate the performance of facial editing methods in terms of identity and attribute preservation as well as the photorealism of edited images. To evaluate the identity preservation, we extract the features of the images before and after editing with FaceNet [35], and compute their euclidean distance. As for the irrelevant attribute preservation, we use a retrained attribute predictor to output a cross-entropy score indicating whether the predicted attribute is consistent with its ground-truth label.

Apart from the aforementioned metrics, we also conduct a user study. Two groups of editing results (one is our result, the other is another method) are provided to participants. The participants are supposed to compare two groups of editing images and then choose the more suitable group for each of the following questions: 1) *Which group of images is more visually realistic?* 2) *Which group of images has more continuous changes?* 3) *After editing, which group of images better preserves the identity?*

5.1. Comparison Methods

InterfaceGAN. InterfaceGAN [38] uses a single direction to perform continuous editing. The direction is obtained by computing the normal vector of the binary SVM boundary.

Multiclass SVM. We further propose an extended version of InterfaceGAN, named Multiclass SVM, where fine-grained labels are used to get multiple SVM boundaries. During the editing, directions will be constantly switched.

Enjoy Your Editing. Enjoy your editing [55] learns a mapping network to generate an identity-specific direction, and it keeps fixed during editing for one identity.

5.2. Quantitative Evaluation

Identity/Attribute Preservation. To fairly compare the continuous editing results with existing methods, we produce our results purely based on semantic field manipulation and language is not involved. We compute the identity preservation and attribute preservation scores for the editing results of baseline methods. Table 1 shows the quantitative comparison results. Our method achieves the best identity and attribute preservation scores.

Ablation Study. The *location-specific* property of semantic field has the following two indications: 1) the trajectory to edit one identity might be a curve instead of a straight line; 2) the editing trajectories are unique to individual identities. The superiority over InterfaceGAN and Enjoy Your Editing validates that the curved trajectory is vital for continuous editing and we will provide further analysis in Section 5.4. Compared to Multiclass SVM, our results confirm the necessity of different directions for different identities.

Table 1: **Quantitative Comparisons.** We report Identity / Attribute preservation metrics. A lower identity score (smaller feature distance) means the identity is better preserved, and a lower attribute score (smaller cross-entropy) means the irrelevant attributes are less changed. Our method has a superior performance in terms of identity and attribute preservation.

Methods	Bangs	Eyeglasses	Beard	Smiling	Young
InterfaceGAN	0.7621 / 0.7491	0.7831 / 1.1904	1.0213 / 1.6458	0.9158 / 0.9030	0.7850 / 1.4169
Multiclass SVM	0.7262 / 0.5387	0.6967 / 0.9046	1.1098 / 1.7361	0.7959 / 0.8676	0.7610 / 1.3866
Enjoy Your Editing	0.6693 / 0.4967	0.7341 / 0.9813	0.8696 / 0.7906	0.6639 / 0.5092	0.7089 / 0.5734
Talk-to-Edit (Ours)	0.6047 / 0.3660	**0.6229** / 0.7720	0.8324 / 0.6891	0.6434 / 0.5028	0.6309 / 0.4814
Talk-to-Edit (Ours) *	**0.5276 / 0.2902**	0.6670 / **0.6345**	**0.7634 / 0.5425**	**0.4580 / 0.3573**	**0.6234 / 0.2731**

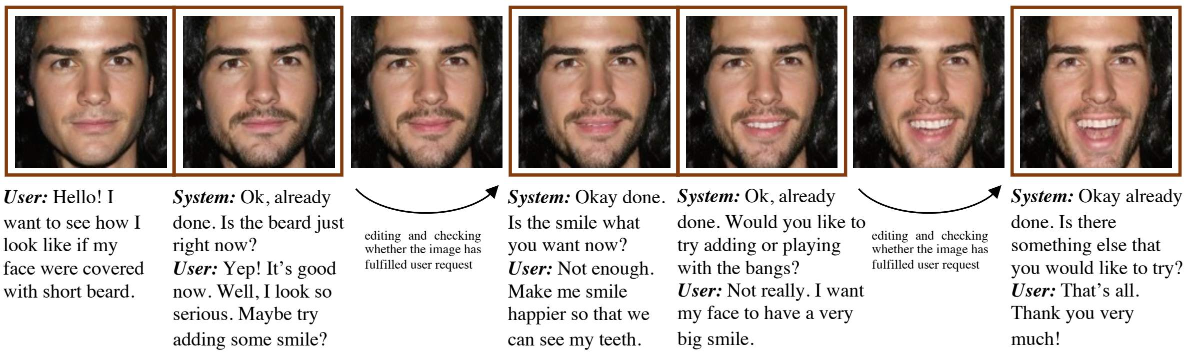


Figure 6: **Results of dialog-based facial editing.** The whole process is driven by the dialog between the user and the system.

5.3. Qualitative Evaluation

Visual Photorealism. Qualitative comparisons are shown in Fig. 5. The results of our method displayed are edited on W+ space. Our proposed method is less likely to generate artifacts compared to previous methods. Besides, when the edited attribute comes to higher degrees, our method can still generate plausible editing results while keeping the identity unchanged.

User Study. We conduct a user study, where users are asked the aforementioned questions and they need to choose the better images. A total number of 27 participants are involved and they are required to compare 25 groups of images. We mix the editing results of different attributes together in the user study. The results of user study are shown in Fig. 9 (a). The results indicate that the majority of users prefer our proposed method in terms of image photorealism, editing smoothness, and identity preservation.

Dialog Fluency. In Fig. 6, we show a dialog example, where the system is asked to add beard for the young guy in the picture. After adding the beard into a desired one, the system then continues to edit the smile as required by the user. The system could talk to the user smoothly in the whole dialog. To further evaluate the fluency of dialog, we invite seven participants to compare six pairs of dialog. In each pair of dialog, one is generated by the system, and the other is revised by a human. Participants need to decide which one is more natural or if they are indistinguishable. The results are shown in Fig. 9 (b). Over half of the participants think the system feedback is natural and fluent.

* edits on W+ space. Others edit on Z space.

5.4. Further Analysis

High-Resolution Facial Editing. Since our editing method is a latent space manipulation based method, it can be extended to images with any resolutions as long as the pretrained GAN is available. Apart from editing results on 128×128 images shown in previous parts, we also provide some 1024×1024 resolution editing results in Fig. 7.

Location-specific Property of Semantic Field. When traversing the semantic field, the trajectory to change the attribute degree is determined by the curvature at each step, and thus it is curved. To further verify this hypothesis, we randomly sample 100 latent codes and then continuously add eyeglasses for the corresponding 1024×1024 images. For every editing direction, we compute its cosine similarity with the initial direction. The average cosine similarity against the attribute class change is plotted in Fig. 10. We observe that the cosine similarity tends to decrease as the attribute class change increases. It confirms that the editing direction could constantly change according to its current location, and thus the location-specific property is vital for continuous editing and identity preservation.

Real Image Editing. In Fig. 8, we show an example of real image editing results. The image is firstly inversed by the inversion method proposed by Pan *et al.* [33]. The inversion process would finetune the weight of StyleGAN, and we observe that the trained semantic field still works.

(a) Bangs (b) Eyeglasses

(c) Beard (d) Smiling

Figure 7: **High-Resolution Image Editing.** Our method can be generalized to 1024×1024 images.

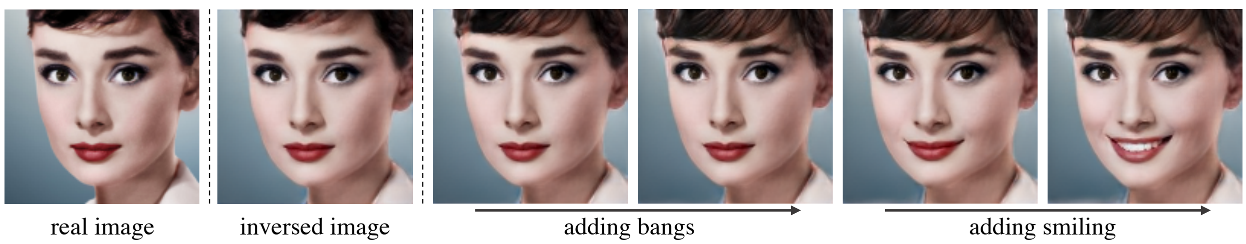


Figure 8: **Real Image Editing.** Given a real image, we first inverse the image and find its corresponding latent code in latent space. We firstly add bangs and then add smiling.

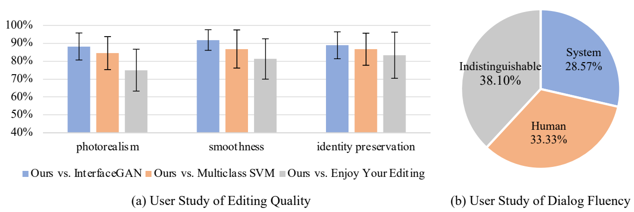


Figure 9: **User Study.** (a) The percentage of participants favoring our results against existing methods. Our results are preferred by the majority of participants. (b) Over half of the participants think the system feedback is natural.

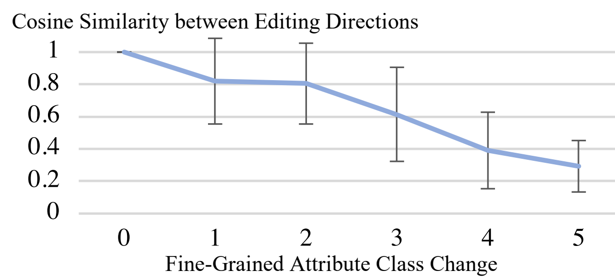


Figure 10: **Cosine Similarity.** We compute the average cosine similarity between the initial direction and directions of later steps. As the attribute class changes, the cosine similarity decreases, indicating that the editing trajectories for most facial images are curved.

6. Conclusion

In this paper, we present a dialog-based fine-grained facial editing system named **Talk-to-Edit**. The desired facial editing is driven by users' language requests and the system is able to provide feedback to users to make the facial editing more feasible. By modeling the non-linearity property of the GAN latent space using semantic field, our proposed method is able to deliver more continuous and fine-grained editing results. We also contribute a large-scale visual-language facial attribute dataset named **CelebA-Dialog**, which we believe would be beneficial to fine-grained and language driven facial editing tasks. In future work, the performance of real facial image editing can be further improved by incorporating more robust GAN-inversion methods and adding stronger identity keeping regularization. We also hope to deal with more complex text requests by leveraging advanced pretrained language models.

Acknowledgement. This study is supported by NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

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

[2] Ryan Y Benmalek, Claire Cardie, Serge Belongie, Xiadong He, and Jianfeng Gao. The neural painter: Multi-turn image generation. *arXiv preprint arXiv:1806.06183*, 2018. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 2

[4] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 2

[5] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *CVPR*, pages 8721–8729, 2018. 1, 2

[6] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention gan for interactive image editing. In *ACM MM*, pages 4383–4391, 2020. 3

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 1, 2

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 5

[9] Tsu-Jui Fu, Xin Eric Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. Sscr: Iterative language-based image editing via self-supervised counterfactual reasoning. *arXiv preprint arXiv:2009.09566*, 2020. 3

[10] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 1

[11] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 1, 2

[13] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 2

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 1

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1, 2

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 2

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 1, 2

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 2

[20] Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. Codraw: Collaborative drawing as a testbed for grounded goal-driven communication. *arXiv preprint arXiv:1712.05558*, 2017. 3

[21] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, pages 1857–1865. PMLR, 2017. 1, 2

[22] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, pages 5549–5558, 2020. 1

[23] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *CVPR*, pages 7880–7889, 2020. 3

[24] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. *arXiv preprint arXiv:2103.01456*, 2021. 1

[25] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *arXiv preprint arXiv:1705.08086*, 2017. 1

[26] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 1, 2

[27] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe what to change: A text-guided unsupervised image-to-image translation approach. In *ACM MM*, pages 1357–1365, 2020. 3

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 3, 6

[29] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[30] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. *arXiv preprint arXiv:1810.11919*, 2018. 1, 2

[31] Kyle Olszewski, Duygu Ceylan, Jun Xing, Jose Echevarria, Zhili Chen, Weikai Chen, and Hao Li. Intuitive, interactive beard and hair synthesis with generative models. In *CVPR*, pages 7446–7456, 2020. 2

[32] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. In *ICLR*, 2021. 4

[33] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep genera-

tive prior for versatile image restoration and manipulation. In *ECCV*, pages 262–277. Springer, 2020. 7

[34] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069. PMLR, 2016. 2

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 6

[36] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018. 3

[37] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, pages 9243–9252, 2020. 2, 4

[38] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 4, 6

[39] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 2

[40] Jing Shi, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu. A benchmark and baseline for language-driven image editing. In *ACCV*, 2020. 3

[41] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 5

[42] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, pages 9786–9796. PMLR, 2020. 2

[43] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *CVPR*, pages 7083–7092, 2018. 2

[44] Wei Wang, Zhen Cui, Yan Yan, Jiashi Feng, Shuicheng Yan, Xiangbo Shu, and Nicu Sebe. Recurrent face aging. In *CVPR*, pages 2378–2386, 2016. 2

[45] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, 2021. 5

[46] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse image generation and manipulation. *arXiv preprint arXiv:2012.03308*, 2020. 1, 2, 3

[47] Jun Xing, Koki Nagano, Weikai Chen, Haotian Xu, Li-yi Wei, Yajie Zhao, Jingwan Lu, Byungmoon Kim, and Hao Li. Hairbrush for immersive data-driven hair modeling. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, pages 263–279, 2019. 2

[48] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, pages 1316–1324, 2018. 2

[49] Hongyu Yang, Di Huang, Yunhong Wang, and Anil K Jain. Learning face age progression: A pyramid architecture of gans. In *CVPR*, pages 31–39, 2018. 2

[50] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5907–5915, 2017. 2

[51] Tianhao Zhang, Hung-Yu Tseng, Lu Jiang, Weilong Yang, Honglak Lee, and Irfan Essa. Text as neural operator: Image manipulation by text instruction. *arXiv preprint arXiv:2008.04556*, 2020. 1, 2

[52] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, pages 597–613. Springer, 2016. 2

[53] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 1, 2

[54] Shizhan Zhu, Raquel Urtasun, Sanja Fidler, Dahua Lin, and Chen Change Loy. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*, pages 1680–1688, 2017. 1, 2

[55] Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. In *ICLR*, 2021. 2, 4, 6