Talk-to-Edit: Fine-Grained 2D and 3D Facial Editing via Dialog

Yuming Jiang, Ziqi Huang, Tianxing Wu, Xingang Pan, Chen Change Loy, Ziwei Liu

Abstract—Facial editing is to manipulate the facial attributes of a given face image. Nowadays, with the development of generative models, users can easily generate 2D and 3D facial images with high fidelity and 3D-aware consistency. However, existing works are incapable of delivering 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 demonstrate the effectiveness of our proposed framework on both 2D and 3D-aware generative models. We term the semantic field for the 3D-aware models as "tri-plane" flow, as it corresponds to the changes not only in the color space but also in the density space. 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, the 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/.

Index Terms—Facial Editing, Image Editing, Generative Models

1 Introduction

The aim of facial editing is to manipulate facial images in the ways specified by users. The recent advance in deep generative models like GANs [1], [2], [3], [4], [5], [6], [7] has promoted the rapid growth of facial editing in recent years, especially in image fidelity. Existing methods mainly focus on improving the quality of facial editing but neglect interactions with users or require users to follow some fixed control patterns. For example, image-to-image translation models [8], [9], [10], [11], [12] 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 [13], [14], a reference image demonstrating the target style [15], [16], [17], [18], and a sentence describing a desired effect [19], [20], [21], [22], [23]. However, users have to follow fixed patterns, which are too demanding and inflexible for most users. Besides, the only 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 [24], [25], [26], [27], [28] could perform continuous editing to some extent by shifting the latent code of a pre-trained GAN [3], [4], [5], [6], 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*

- *Yuming Jiang is with the S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: yuming002@e.ntu.edu.sg.*
- *Ziqi Huang is with the S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: ziqi002@e.ntu.edu.sg.*
- *Tianxing Wu is with the S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: tianxing.wu@ntu.edu.sg.*
- *Xingang Pan is with the S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: xingang.pan@ntu.edu.sg.*
- *Chen Change Loy is with S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: ccloy@ntu.edu.sg.*
- *Ziwei Liu is with S-Lab, Nanyang Technological University, Singapore, 639798. E-mail: ziwei.liu@ntu.edu.sg.*

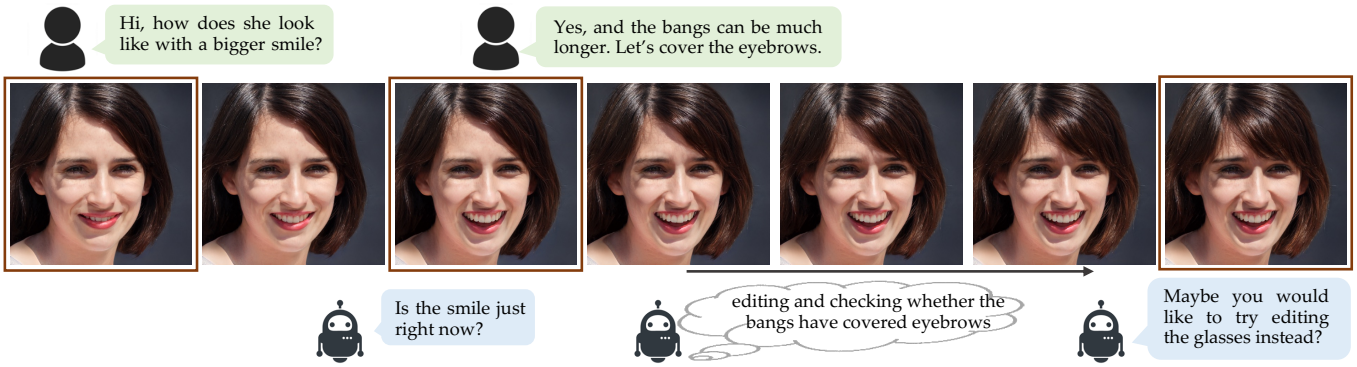


Fig. 1: **An example of *Talk-to-Edit***. The user provides a facial image and an editing request. Our system then edits the image accordingly and provides meaningful language feedback such as clarification or alternative editing suggestions. During editing, the system is able to control the extent of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

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 by each image, there is also a user request sample and several captions describing these fine-grained facial attributes.

We demonstrate the capability of our proposed *Talk-to-Edit* framework to manipulate 2D facial images and 3D-aware facial images. We experiment on two representative generative models, *i.e.,* StyleGAN [6] and EG3D [7], for 2D images and 3D images, respectively. Both of them utilize the latent space to map the latent code to the images. We perform the task of facial editing by manipulating the latent codes of the target images. On 2D images, the manipulation of latent codes corresponds to the editing of the color space. On 3D images, the manipulation of latent code yields deformations in both the color space and the density space, which we term as the "tri-plane flow". Our proposed *Talk-to-Edit* is a plug-and-play model.

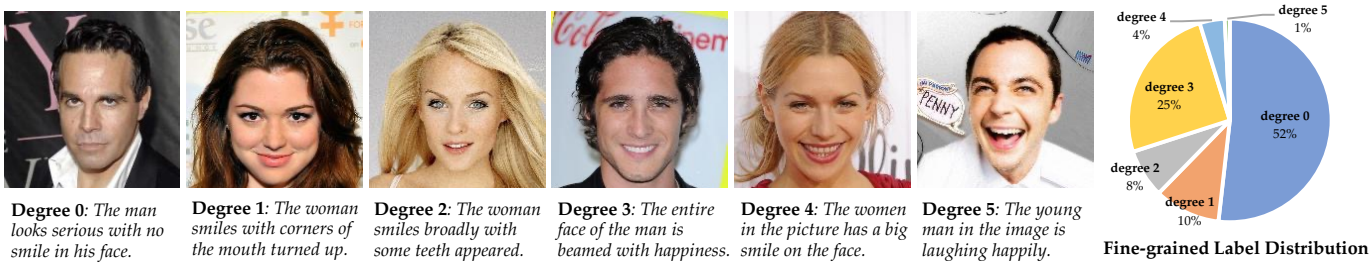
In summary, our main contributions are: **1)** We propose to perform fine-grained facial editing via dialog, an easier interactive way for users. **2)** To achieve more continuous and fine-grained facial editing, we propose to model a location-specific semantic field. **3)** We achieve superior results 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. **5)** Our proposed *Talk-to-Edit* can work on both 2D generative models and 3D-aware generative models.

Compared with the earlier version in ICCV 2021 [29], we demonstrate the potential of our proposed framework to work on 3D-aware generative models. Specifically, we train the 3D version of the semantic field, *i.e.,* "tri-plane flow", in the latent space of EG3D. To show superiority, we also adapt some 2D baselines to the 3D models for fair comparisons. In addition to the methodology, we provide more implementation details, *e.g.,* the editing on W+ space of StyleGAN, and explanations of the evaluation metrics. We also include failure case discussions in this version.

2 Related Work

Semantic Facial Editing. Several methods have been proposed for editing specific attributes such as age progression [30], [31], hair synthesis [32], [33], and smile generation [34]. 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 [8], [9], [10], [11], [12] 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 [35], [36] are drawing increasing attention due to the advancement of GAN models like StyleGAN [5], [6]. These approaches typically discover semantically meaningful direc-



Degree 0: *The man looks serious with no smile in his face.*

Degree 1: *The woman smiles with corners of the mouth turned up.*

Degree 2: *The woman smiles broadly with some teeth appeared.*

Degree 3: *The entire face of the man is beamed with happiness.*

Degree 4: *The women in the picture has a big smile on the face.*

Degree 5: *The young man in the image is laughing happily.*

Fig. 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.

tions in the latent space of a pretrained GAN so that moving the latent code along these directions could achieve the desired editing in the image space. Supervised methods find directions to edit the attributes of interest using attribute labels [24], [25], [37], while unsupervised methods exploit semantics learned by the pretrained GAN to discover the most important and distinguishable directions [26], [27], [28]. InterFaceGAN [24], [25] 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 [37] 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 [38], [39] consider the non-linear property of latent space. In recent works [40], [41], the transformer architecture is adopted to find the editing directions in the 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 preserve facial identities. Due to the emergence of diffusion models, there are some explorations [42], [43], [44] on diffusion-based facial editing. In DiffusionCLIP [42], facial images are edited through the reversed DDIM process guided by diffusion models, which are fine-tuned with CLIP loss. Asyrp [43] edits the images through the semantic latent space in pretrained diffusion models. DiffuseIT [44] proposes to use disentangled style and content representation to perform editing.

Language-based Image Editing. The flexibility of natural language has attracted researchers to propose a number of text-to-image generation [23], [45], [46], [47] and manipulation [19], [20], [21], [22], [23] approaches. For example, given an input image, TediGAN [23] generates a new image conditioned on a text description. Some other approaches [48], [49], [50], [51], [52], [53], [54] allow users to give requests in the form of natural language but do not provide meaningful feedback, clarification, suggestion, or interaction. Chatpainter [55] 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 fine-grained attribute details, and offer other editing options for users to explore.

3D-Aware Image Generation. Existing 3D-aware image generation works are mainly classified into three types: mesh-based, voxel-based, and implicit representation-based methods. Mesh-based methods [56], [57] have limited photorealism for high-quality image generation, while voxel-based methods [58], [59], [60], [61], [62], [63] are memory-inefficient as it requires to store voxel grids. Since the emergence of NeRF [64], the 3D representation is shifted to the implicit one [65], [66]. The implicit neural representation is implemented as fully-connected layers in accompany by positional encodings. Though it saves memories, it is inefficient to query and thus hampers high-quality image generation as well. The recently proposed EG3D [7] utilizes the advantages of implicit representation and explicit representation (*e.g.*, voxel grids) and proposes tri-plane representations. The StyleGAN architectures are employed to generate the intermediate tri-plane representations, which are then rendered to low-resolution features. The final images are generated by upsampling the low-resolution features. Recently, E3DGE [67] proposes an inversion framework for projecting the real image into EG3D for 3D-aware GAN editing. In this paper, we demonstrate the capability of our proposed framework to manipulate 3D-aware images in the latent space of EG3D.

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 [68] 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 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 [24], [25], [37], [69], 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.

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

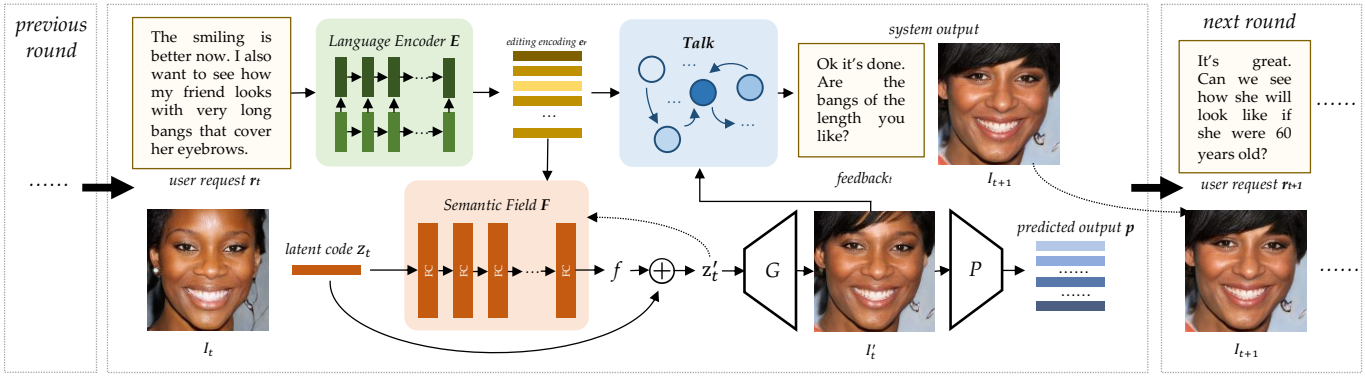
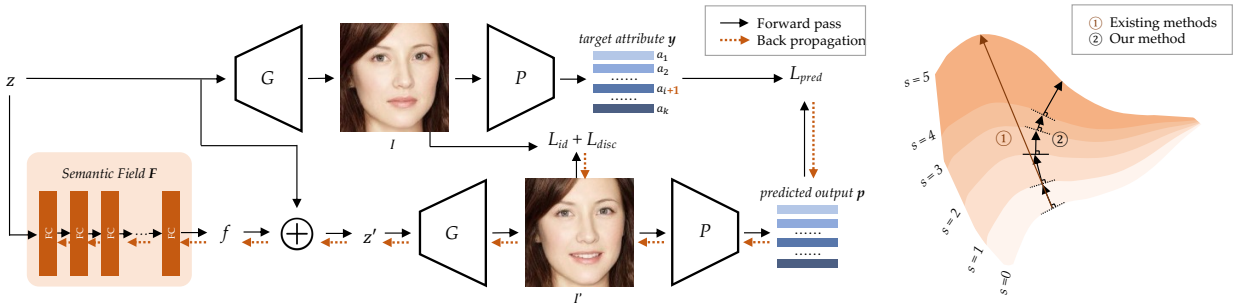


Fig. 3: **Overview of *Talk-to-Edit* Pipeline.** In round t, we receive the input image $\mathbf{I}_t$ and its corresponding latent code $\boldsymbol{z}_t$ from the last round. Then the *Language Encoder* E extracts the editing encoding $\boldsymbol{e}_r$ from the user request $\boldsymbol{r}_t$, and 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.



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

Fig. 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.

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 the 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 [70], [71] 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:

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

where $Face(\cdot)$ is the pretrained face recognition model [72].

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 the 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.

The feedback provided by the system comes from one of 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. The rules for the Talk module are illustrated in Figure 7. We will provide more details in Section 4.6.

4.4 2D Facial Image Editing

StyleGAN [6] achieves state-of-the-art performance in generating facial images of high fidelity. StyleGAN maps an input latent code z sampled from a fixed distribution (*e.g.*, Gaussian distribution) to a facial image. The latent space of the fixed distribution is called Z space. The editing on the Z space follows the Eq. (5).

In StyleGAN, a mapping network $M(\cdot)$, composed of 8 fully-connected layers, is employed to map the latent code from Z space to the W space. On top of these two spaces, $W+$ space is an extended space of W space. The difference is that W space shares the same latent code at different layers while $W+$ space accepts different latent codes at different layers. Compared to the Z space, the $W+$ space is better disentangled, and thus editing on this space can provide better results.

We propose a variant of the semantic field to support the editing on $W+$ space. In the $W+$ space, deep layers of latent codes of $W+$ space control the low-level features

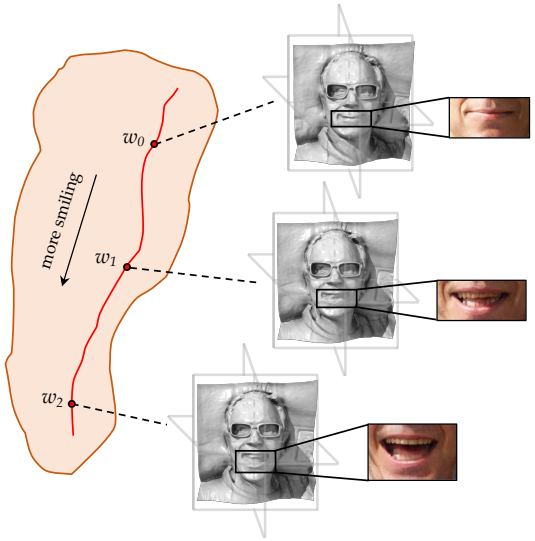


Fig. 5: **Tri-plane Flow in 3D Facial Image Editing.** Traversing through the latent space of 3D generative models, we can observe changes in color space (RGB images) and density space (meshes).

of facial images, such as color, brightness, illuminations, and etc. During facial editing, keeping these layers fixed helps to maintain the low-level features of facial images. Therefore, when updating latent codes, we only update the first k layers of latent codes. We empirically set k as 8 for 128×128 images and 10 for 1024×1024 images.

As for the updates of the latent code for the first k layers, we adopt a regularization method [69]. To enforce the field vector $\boldsymbol{f}_w$ to be a valid latent direction that would not make the edited latent code fall into the outlier region of pretrained StyleGAN latent space, we feed the output of the semantic field $F(\boldsymbol{w})$ into the mapping network of the StyleGAN. Also, we use the $M(\mathbf{0})$ as the mean point of the latent direction. With $M(\mathbf{0})$, the field vector $\boldsymbol{f}_w$ can be viewed as an offset to the mean direction. To summarize, the latent code is updated as follows:

$$\boldsymbol{w}' = \boldsymbol{w} + \alpha \cdot \boldsymbol{f}_w = \boldsymbol{w} + \alpha(M(F(\boldsymbol{w})) - M(\mathbf{0})), \tag{11}$$

where α is the step size, $F(\cdot)$ denotes the semantic field network and the $M(\cdot)$ is the mapping network of the StyleGAN.

4.5 3D Facial Image Editing

For 3D-aware facial image editing, we adopt the EG3D model [7] as the generative model for its good performance on 3D-aware image generation. The whole architecture takes the latent code z and camera parameters P as inputs. A tri-plane representation is first generated through a StyleGAN2-based feature generator. Then neural renderer is employed to render the generated tri-plane representation to a 128×128 image, which is then upsampled by a super-resolution module to generate the final image.

In EG3D, the latent code z is also translated to an intermediate W space. The W space also considers the camera parameters. To avoid the influence of camera parameters,

we keep the camera parameters unchanged during the editing. The latent code is updated as follows:

$$\begin{aligned} \boldsymbol{w}' &= \boldsymbol{w} + \alpha(M(\boldsymbol{f}_w, P) - M(\mathbf{0}, P)) \\ &= \boldsymbol{w} + \alpha(M(F(\boldsymbol{w}), P) - M(\mathbf{0}, P)). \end{aligned} \tag{12}$$

Similar to the observations in StyleGAN, in EG3D, we also find that fixing the last few layers helps the maintenance of the low-level features of the original facial images. Different from the editing in the latent space of StyleGAN, where the manipulation of latent code only reflects the changes in the color space, the editing on EG3D corresponds to the changes in the color space and the density space. As shown in Fig. 5, by moving the latent code in the latent space, we can observe smooth changes in the images and meshes. The changes in images and meshes are caused by the deformations in tri-plane representations. The editing of the latent code manipulates the tri-plane representation and thus changes the color and density in the rendering process. We can term the editing in EG3D as the "tri-plane flow".

4.6 Implementation Details

User Request Understanding. The language encoder E has three components: 1) a learnable 300-D word embedding; 2) a two-layer LSTM with cell size of 1024; 3) fully-connected layers following the LSTM to generate the editing encoding $\boldsymbol{e}_r$.

As shown in Fig. 6, commonly, users' editing requests could be roughly classified into three major types: **1)** Describe the attribute and specify the target degree, *e.g., Let's try extremely long bangs that cover the entire forehead.* **2)** Describe the attribute of interest and indicate the relative degree of change, *e.g., The bangs can be slightly longer.* **3)** Describe the attribute and only the editing direction without specifying the degree of change, *e.g., Let's make the bangs longer.* Since the types of facial editing requests are relatively fixed, we use template-based text generation methods to form a pool of editing requests. The request pool is used to train the language encoder. We prepare more than 300 request templates with diverse sentence patterns. A pool of synonymous words is used to enrich the user request templates. We use 10,000 user requests in total. For each generated request, we provide their corresponding hard labels to train the language encoder E. The language encoder is optimized as a classification model, which uses the cross-entropy loss as the loss function. The learning rate is set as 10^{-3}, the batch size is 2048, and the Adam optimizer [73] is adopted.

The editing encoding $\boldsymbol{e}_r$ generated by the language encoder E is implemented as hard labels containing the following information: (1) request type, (2) the attribute of interest, (3) the editing direction, and (4) the change of degree. In practice, the same user request could be interpreted differently depending on the dialog context. For example, simply saying *"Yes"* has different meanings under different scenarios. If the system makes a suggestion *"Do you want to make the bangs longer?"*, by replying *"Yes"*, the user means to make the bangs longer. However, if the system asks if the desired effect is achieved in the previous round, *"Yes"* means the editing is satisfactory in this context. Therefore, multiple language encoders are needed to parse the user request under different dialog contexts. During training, the weights of word embedding and LSTM are shared across different language encoders. The current system feedback decides which language encoder would be used.

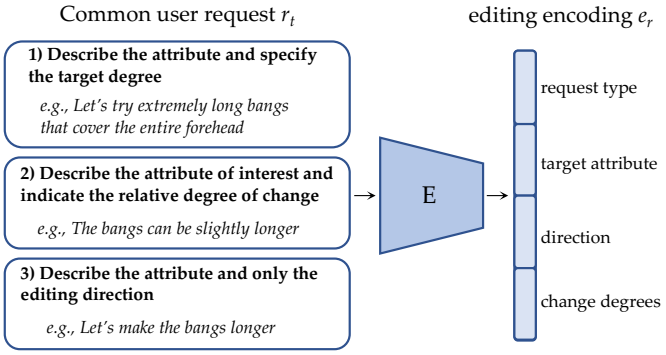


Fig. 6: **Illustration of User Request Understanding Module.** The language encoder is trained to translate the user request into editing encodings.

Semantic Field. The training of the semantic field requires the following pretrained models: fine-grained attribute predictor P, face recognition model $Face$, StyleGAN generator G, and discriminator D. The fine-grained attribute predictor P is pretrained on CelebA-Dialog dataset using our fine-grained attribute labels with a multi-class cross-entropy loss. StyleGAN G and its corresponding discriminator D are trained on the CelebA dataset [68] and FFHQ dataset [5] for 128×128 and 1024×1024 facial images respectively. As for the $Face$ Model, we use the off-the-shelf ArcFace model [72] trained on LFW dataset [74], [75].

Since the pretrained StyleGAN has the mode collapse problem, during the training of the semantic field, we need to sample the training latent codes such that all fine-grained attribute classes are more balancedly distributed. The mapping network of semantic field F is composed of 8 fully-connected (FC) layers with dimension 512. Except for the last FC layer, each FC layer is followed by a leaky ReLU with a slope of 0.2. The learning rate for training the semantic field is 10^{-4}, batch size is set as 32, and Adam optimizer [73] is adopted.

For the tri-plane flow for 3D-aware images, we use the same pretrained fine-grained attribute predictor P, face recognition model $Face$, and discriminator D. As for the generator, we use the pretrained EG3D model. For the viewpoint sampling, the EG3D uses fixed camera parameters, to generate the latent code in the W space. We follow this strategy, during the training, the latent code in the W space is obtained using the same fixed camera parameters as those used in EG3D.

System Feedback (Talk Module). The aim of our *Talk* module is to provide natural language feedback according to the user's editing request and the information on the editing history. Specifically, we track the dialog-based editing system using a finite-state machine as shown in Figure 7. At each system state, a few possible types of system feedback are defined, and the actual type of system feedback is selected based on pre-defined probabilities. The actual content of the system feedback is then instantiated based on the feedback type and system state.

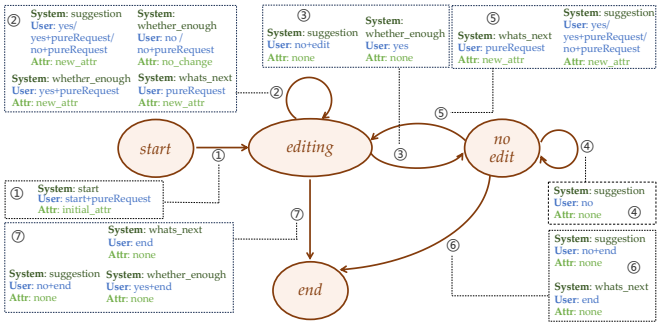


Fig. 7: **Illustration of the Talk Module.** The Talk module follows the rules of our pre-defined state machine.

We design four different states to describe the status of the entire editing system. **1)** ***start***, that is, the first round of dialog, **2)** ***editing***, where the system performs editing in the current round of dialog. Transition to this "editing" state requires specifying the attribute of interest, which is either the currently edited attributes (*i.e.*, no_change) or the new target attribute (*i.e.*, new_attr or initial_attr). As for the transition from this "editing" state to other states, the attribute value is set to none, **3)** ***no edit***, where the system does not edit the image and waits for further instructions from the user. and **3)** ***end***, where the system ends the conversation upon the user's request.

After one round of editing, the *Talk* module will provide a natural language feedback, which belongs to one of the following categories: 1) **whether_enough** – in the *edit* state, the system might check whether the current attribute degree is satisfying, in order to achieve fine-grained editing desired by the user. For example, after the user requests to make the bangs longer, the system could give the following feedback, *e.g.*,, *"Are the bangs now of the length you like?"*. 2) **suggestion** – in both the *edit* state or the *no edit* state, the system could provide editing suggestions, *e.g.*, *"Do you want to try manipulating the age?"* In order to let the user fully explore possible manipulation options, the system tends not to suggest editing an attribute that has been edited before. If there exists a larger number of attributes not edited by the user yet, then there is a higher probability for the system to make a suggestion, 3) **whats_next** – simply asks the user what other attributes he or she would like to edit, *e.g.*, *"Ok, what's next?"*.

We sample a sentence from a pool of templates of the chosen feedback category and randomly replace phrases using a predefined pool of synonyms to extend the language richness. We observe that this simple design can provide meaningful feedback to some extent.

5 Experiments

5.1 Evaluation Datasets

We synthesize the evaluation dataset by sampling latent codes from the StyleGAN pretrained on CelebA dataset [68] (for 2D images) and EG3D pretrained on FFHQ datset [5] (for 3D images). Using latent codes, we then generate corresponding images. When comparing with baseline methods (explained in Sec. 5.2), we use the latent code for editing directly to avoid the error introduced by GAN-inversion methods.

5.2 Comparison Methods

InterfaceGAN. InterfaceGAN [24] is a latent space based method. The continuous editing is achieved by moving the latent code along a straight line, *i.e.*, adding the a same vector to the original latent code. The direction used for changing the attribute degree is obtained by computing the normal vector of the binary classification SVM boundary. This direction is fixed throughout the editing. We first train binary attribute predictors to classify the generated images. Then the corresponding latent codes are used to train the binary SVM.

Multiclass SVM. We further propose an extended version of InterfaceGAN as one of the baseline methods, named Multiclass SVM. Instead of the binary classification SVM, we train multiple SVM boundaries for fine-grained labels. More specifically, for each pair of neighbouring classes, a classification SVM would be trained. Thus, for one attribute, there are five SVM boundaries in total. During the editing, directions will be switched according to current states. The attribute predictor used for the classification of generated images is the same as the one we use for predictor loss.

Enjoy Your Editing. Enjoy your editing [37] learns a mapping network to generate identity-specific directions for each initial latent codes. The identity-specific directions keep same during editing for one image. We reimplement the method, train the mapping network with the original design and same hyper-parameters are adopted. To achieve more attribute degrees, we use larger step-sizes than the original setting, *i.e.*, $\varepsilon > 1.0$.

5.3 Evaluation Metrics

For 2D images, 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 [76], 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.

Identity Preservation Metric. We use the off-the-shelf face model *FaceNet* [76] to extract features for images before and after editing. Then we compute the Euclidean distance between features of the edited facial images and the features of the original facial image. The identity preservation metric is expressed as follows:

$$\text{IDPreserve} = \frac{1}{N}\sum_{i=1}^{N} \left\| FaceNet(I_i) - FaceNet(I_0) \right\|_2 , \tag{13}$$

where I_0 is the original image, I_i are edited images, and N is the total number of edited images.

Attribute Preservation Metric. We retrain an attribute predictor P' (different from the one we use for training), and

TABLE 1: **Quantitative Comparisons on 2D Images (CelebA domain).** All the methods adopt the pretrained StyleGAN on the CelebA dataset. 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. The best results are **bolded**, and the second best results are underlined.

Methods	Bangs	Eyeglasses	Beard	Smiling	Young
InterfaceGAN [24]	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 [37]	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**

TABLE 2: **Quantitative Comparisons on 2D Images (FFHQ domain).** The methods adopt the pretrained Diffusion model or StyleGAN on the FFHQ dataset. 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.

Method	Bangs	Eyeglasses	Beard	Smiling	Young
DiffusionCLIP [42]	0.5856 / 0.6418	0.6002 / 0.9644	**0.4840** / 0.8535	0.4777 / 0.7806	0.6359 / 1.2507
Asyrp [43]	0.9292 / 0.8297	1.1107 / 1.2896	0.9850 / 1.2008	1.0658 / 1.6126	1.2177 / 1.2881
Latent Transformer [41]	0.6811 / 0.5275	0.5636 / **0.5761**	0.7025 / 0.6125	0.6145 / 0.4691	0.6848 / 0.5597
Talk-to-Edit (Ours)	**0.5743** / **0.4306**	**0.5552** / 0.8265	0.5278 / **0.6103**	**0.3493** / **0.4555**	**0.5720** / **0.5132**

use the retrained predictor to output cross-entropy score. The attribute preservation metric is defined as follows:

$$\text{AttrPreserve} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,j\neq m}^{k}\sum_{c=0}^{C} y_{j,c} log(p'_{j,c}), \quad (14)$$

where N is the total number of edited images, k is the number of attributes, m is the index of the attribute being edited, $p'_{j,c}$ is the softmax output of predictor P', $y_{j,c}$ is the binary indicator with respect to the target class and it is obtained by feeding the original image to the attribute predictor.

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?*

As for 3D-aware images, in addition to identity preservation and attribute preservation, we compute the forward Chamfer distance to evaluate the 3D consistency of the images.

Forward Chamfer Distance. The manipulation of the latent space of 3D-aware generative models involves not only the changes in the color space but also the deformations in the density space. The forward Chamfer distance is computed between the original head mesh and the edited head mesh, and it is defined as follows:

$$d = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{|S_o|}\sum_{x\in S_o}\min_{y\in S_i}\|x-y\|_2^2\right], \quad (15)$$

where N is the total number of editing steps, S_o is the vertices of the original head mesh and S_i is the vertices in the head mesh of the i-th editing step.

5.4 Quantitative Evaluation on 2D Images

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 on CelebA domain. The generation model (*i.e.*, StyleGAN) is pretrained on the CelebA dataset. Our method achieves the best identity and attribute preservation scores. We also report quantitative comparisons with state-of-the-art transformer-based and diffusion-based methods which are built upon the pretrained models on the FFHQ dataset. The quantitative results are calculated with one-step editing for fair comparison. As shown in Table 2, our proposed method achieves competitive identity preservation and attribute preservation scores compared to baseline methods.

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 5.8. Compared to Multiclass SVM, our results confirm the necessity of different directions for different identities.

5.5 Qualitative Evaluation on 2D Images

Visual Photorealism. Qualitative comparisons are shown in Fig. 8. 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

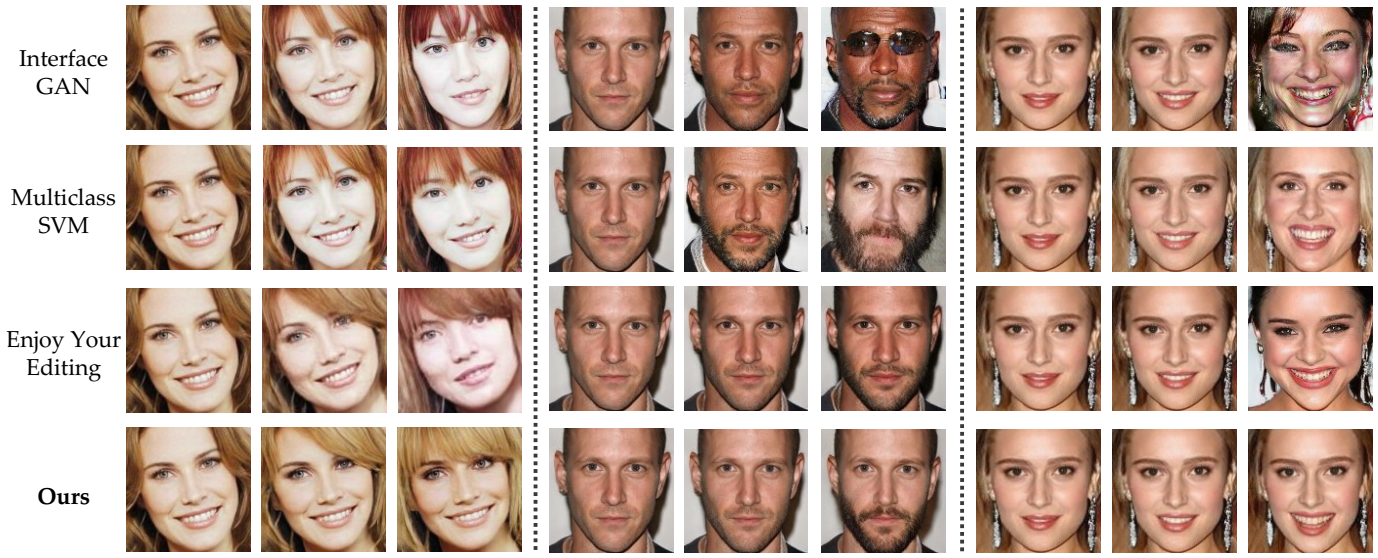


Fig. 8: **Qualitative Comparison on 2D Images (CelebA domain).** We compare our approach with InterfaceGAN, Multiclass SVM, and Enjoy Your Editing. Our editing results are more realistic. Besides, our method is less likely to change the identity and other attributes.

TABLE 3: **Quantitative Comparisons on 3D-aware Images.** We report Identity preservation, Attribute preservation, and Chamfer distances. 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. A smaller Chamfer distance means better 3D consistency. The best results are **bolded**, and the second best results are underlined.

Methods	Bangs	Eyeglasses	Beard	Smiling
InterfaceGAN [24]	0.4494 / 0.6345 / 96.5007	0.5556 / 1.0403 / 198.4897	0.4362 / 0.7480 / 72.5549	0.6065 / 0.7566 / 239.3674
Multiclass SVM	0.3882 / **0.5912** / 49.9648	0.4325 / **0.8606** / 63.1457	0.4390 / 0.7349 / 47.7721	0.3819 / 0.6047 / 44.6267
Enjoy Your Editing [37]	0.3346 / 0.6013 / 56.4048	0.4247 / 0.9168 / 72.2978	**0.4244** / **0.7128** / **39.8632**	0.5803 / 0.7234 / 291.1915
Talk-to-Edit (Ours)	**0.2980** / 0.5950 / **37.7279**	**0.3663** / 0.8996 / **46.4000**	0.4304 / 0.7164 / 64.4676	**0.2733** / **0.5774** / **32.4784**

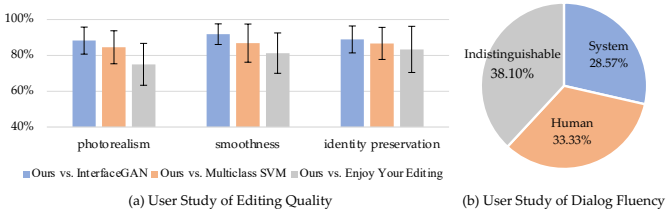


Fig. 9: **User Study.** (a) The percentage of participants favoring our results against existing methods. Our results are preferred by the majority of participants. (b) Over half of the participants think the system feedback is natural.

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. 10, 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.

5.6 Quantitative Evaluation on 3D Images

Table 3 shows the quantitative comparisons on 3D-aware images. For the Bangs and Eyeglasses attributes, our method achieves the best identity preservation score and the smallest Chamfer distance. The attribute preservation score is comparable to Multiclass SVM. As for the beard attribute, our method achieves comparable identity preservation score and attribute preservation score as the Enjoy Your Editing. For the Smiling attribute, our method outperforms the baseline methods in all these metrics.

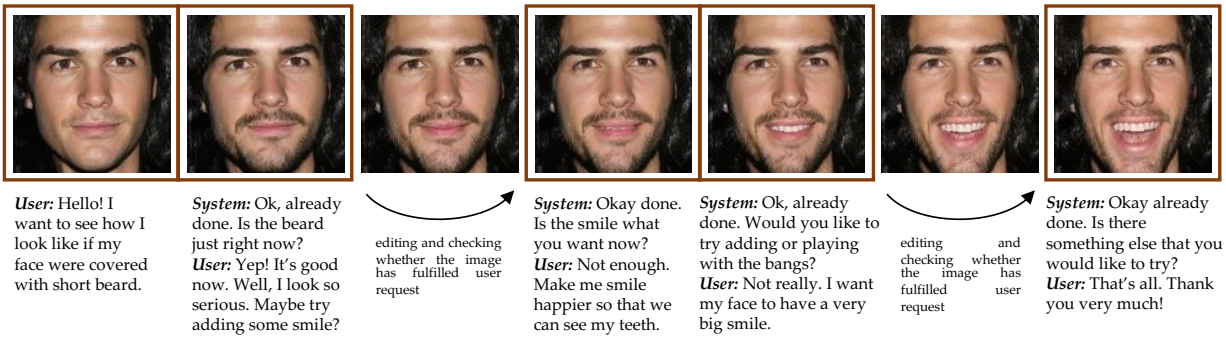


Fig. 10: **Results of dialog-based facial editing.** The whole process is driven by the dialog between the user and the system.

(a) Bangs (b) Smiling

Fig. 11: **Quantitative Comparisons on 3D-aware Images.** We visualize the edited images and meshes. Our method preserves the identity better.

(a) Bangs (b) Eyeglasses

(c) Beard (d) Smiling

Fig. 12: **High-Resolution Image Editing.** Our method can be generalized to 1024×1024 images.

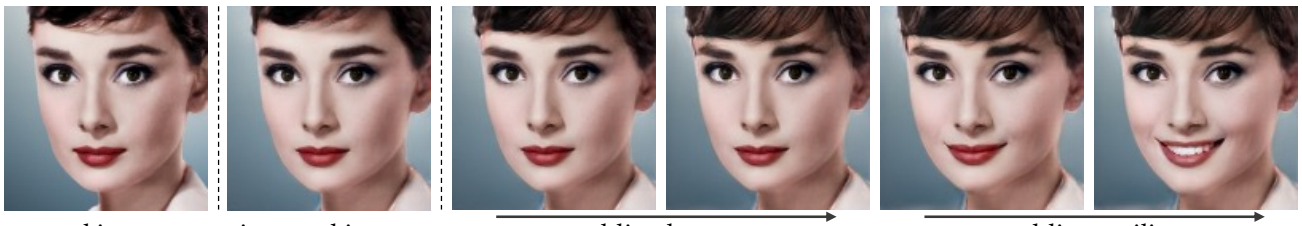


Fig. 13: **Real Image Editing.** Given a real image, we first inverse the image and find its corresponding latent code in latent space. We firstly add bangs and then add smiling.

5.7 Qualitative Evaluation on 3D Images

In Fig. 11, we compare the visual results obtained by different methods. Our method achieves better results. The identity is well preserved and the changes in the meshes are smooth.

5.8 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. 12.

Real Image Editing. In Fig. 13, we show an example of real image editing results. The image is firstly inversed by the inversion method proposed by Pan *et al.* [77]. The inversion process would finetune the weight of StyleGAN, and we observe that the trained semantic field still works.

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. 14. 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.

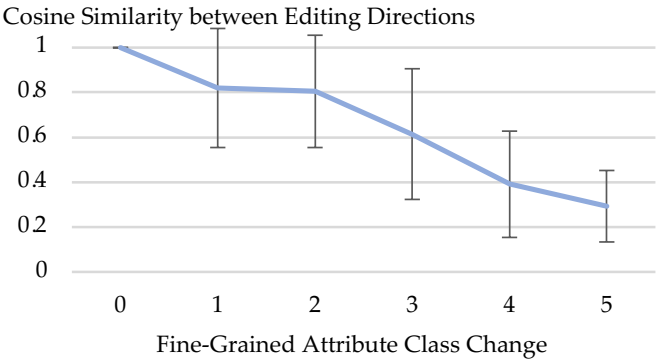
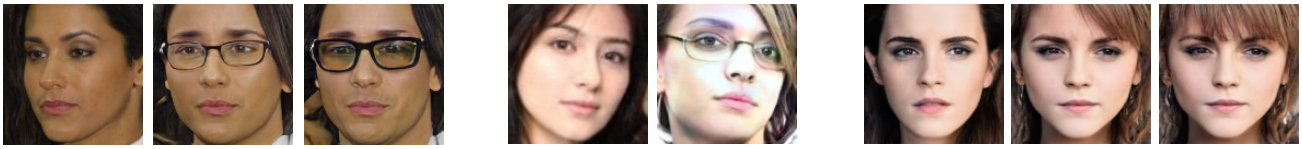


Fig. 14: **Cosine Similarity.** We compute the average cosine similarity between the initial direction and directions of later steps. As the attribute class changes, the cosine similarity decreases, indicating that the editing trajectories for most facial images are curved.

Comparisons on the Editing in Z Space and W+ space. Since the W+ space has a better disentanglement property compared to the Z space, we propose a variant of W+ space editing. Quantitative comparisons of the editing in these two spaces are shown in Table 4. The editing in W+ space has significant improvement in Bangs, Beard, Smiling, and Young attributes. On the Eyeglasses attribute, the editing in W+ space has a comparable identity score as that in Z space

TABLE 4: **Quantitative Comparisons on Z space and W+ space editing.** 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. The best results are **bolded**, and the second best results are underlined.

Methods	Bangs	Eyeglasses	Beard	Smiling	Young
Z Space	0.6047 / 0.3660	**0.6229** / 0.7720	0.8324 / 0.6891	0.6434 / 0.5028	0.6309 / 0.4814
W+ Space	**0.5276** / **0.2902**	0.6670 / **0.6345**	**0.7634** / **0.5425**	**0.4580** / **0.3573**	**0.6234** / **0.2731**



(a) Identity Loss (b) Artifacts (c) Real Cases

Fig. 15: **Failure Case on 2D Facial Image Editing.**

Fig. 16: **Failure Case on 3D Facial Image Editing.**

and a better attribute preservation score.

5.9 Limitations and Future Work

Here, we take the eyeglasses attribute as an example to illustrate the failure case of synthetic image editing. As shown in Fig. 15 (a), identity loss could be observed in some cases, and this issue is severer on female images. The problem may attribute to the dataset bias and the mode collapse issue of the pretrained GAN. For example, the CelebA dataset [68] has only a small number of females with eyeglasses. Thus, females with eyeglasses are only a minority in the image distribution of the pretrained GAN. In this case, given a randomly sampled female without eyeglasses as the initial image, it is sometimes difficult to wear a pair of eyeglasses for her in a well-disentangled manner. Another issue is the artifacts problem shown in Fig. 15 (b). For some latent code, it is difficult to change the attribute from degree 0 to degree 1. After many latent code updating iterations, the latent code falls into the outlier region of the latent space so that the corresponding image would bear artifacts. Our proposed semantic field may not perfectly model the non-linearity property for this attribute due to the dataset imbalance.

As for editing real images, it is more prone to change the identities. As shown in Fig. 15 (c), adding bangs would change the face shape. This is because that GAN-inversion, as an ill-posed problem, may introduce an additional gap between the inverted latent code and the original latent space. This could potentially be addressed by adopting more advanced GAN-inversion techniques [78], [79], [80], [81], [82] that better keep the latent codes within the latent domain.

Figure 16 shows one failure case of 3D-aware facial image editing. Artifacts would appear during the editing. As shown in this example, there are some artifacts around the cheek when the smiling is added to a large degree.

For the language feedback, we currently use template-based sentences according to the system states. Although we randomly replace some phrases with their synonyms to extend the language richness, the diversity of the feedback is still limited. In future works, we can use ChatGPT to further refine the feedback by feeding the template-based sentences generated by our system to the ChatGPT.

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. We demonstrate the effectiveness of our proposed framework on 2D and 3D face generative models. 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 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). It is also supported by NTU NAP and MOE AcRF Tier 1 (2021-T1-001-088).

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[2] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[3] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *CVPR*, 2020, pp. 8110–8119.

[7] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3d generative adversarial networks," in *CVPR*, 2022, pp. 16 123–16 133.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017, pp. 2223–2232.

[9] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018, pp. 8789–8797.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 1125–1134.

[11] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *ICML*. PMLR, 2017, pp. 1857–1865.

[12] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *arXiv preprint arXiv:1703.00848*, 2017.

[13] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *CVPR*, 2020, pp. 5549–5558.

[14] J. Sun, X. Wang, Y. Shi, L. Wang, J. Wang, and Y. Liu, "Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–10, 2022.

[15] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*. Springer, 2016, pp. 694–711.

[16] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *arXiv preprint arXiv:1705.08086*, 2017.

[17] W. Yin, Z. Liu, and C. C. Loy, "Instance-level facial attributes transfer with geometry-aware flow," in *AAAI*, 2019, pp. 9111–9118.

[18] X. Li, S. Zhang, J. Hu, L. Cao, X. Hong, X. Mao, F. Huang, Y. Wu, and R. Ji, "Image-to-image translation via hierarchical style disentanglement," *arXiv preprint arXiv:2103.01456*, 2021.

[19] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, "Language-based image editing with recurrent attentive models," in *CVPR*, 2018, pp. 8721–8729.

[20] T. Zhang, H.-Y. Tseng, L. Jiang, W. Yang, H. Lee, and I. Essa, "Text as neural operator: Image manipulation by text instruction," *arXiv preprint arXiv:2008.04556*, 2020.

[21] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: manipulating images with natural language," *arXiv preprint arXiv:1810.11919*, 2018.

[22] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *ICCV*, 2017, pp. 1680–1688.

[23] W. Xia, Y. Yang, J.-H. Xue, and B. Wu, "Tedigan: Text-guided diverse image generation and manipulation," *arXiv preprint arXiv:2012.03308*, 2020.

[24] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *CVPR*, 2020, pp. 9243–9252.

[25] Y. Shen, C. Yang, X. Tang, and B. Zhou, "Interfacegan: Interpreting the disentangled face representation learned by gans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[26] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the gan latent space," in *ICML*. PMLR, 2020, pp. 9786–9796.

[27] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in gans," *arXiv preprint arXiv:2007.06600*, 2020.

[28] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," *arXiv preprint arXiv:2004.02546*, 2020.

[29] Y. Jiang, Z. Huang, X. Pan, C. C. Loy, and Z. Liu, "Talk-to-edit: Fine-grained facial editing via dialog," in *ICCV*, 2021, pp. 13 799–13 808.

[30] H. Yang, D. Huang, Y. Wang, and A. K. Jain, "Learning face age progression: A pyramid architecture of gans," in *CVPR*, 2018, pp. 31–39.

[31] W. Wang, Z. Cui, Y. Yan, J. Feng, S. Yan, X. Shu, and N. Sebe, "Recurrent face aging," in *CVPR*, 2016, pp. 2378–2386.

[32] K. Olszewski, D. Ceylan, J. Xing, J. Echevarria, Z. Chen, W. Chen, and H. Li, "Intuitive, interactive beard and hair synthesis with generative models," in *CVPR*, 2020, pp. 7446–7456.

[33] J. Xing, K. Nagano, W. Chen, H. Xu, L.-y. Wei, Y. Zhao, J. Lu, B. Kim, and H. Li, "Hairbrush for immersive data-driven hair modeling," in *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 263–279.

[34] W. Wang, X. Alameda-Pineda, D. Xu, P. Fua, E. Ricci, and N. Sebe, "Every smile is unique: Landmark-guided diverse smile generation," in *CVPR*, 2018, pp. 7083–7092.

[35] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *ECCV*. Springer, 2016, pp. 597–613.

[36] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," *arXiv preprint arXiv:1609.07093*, 2016.

[37] P. Zhuang, O. Koyejo, and A. G. Schwing, "Enjoy your editing: Controllable gans for image editing via latent space navigation," in *ICLR*, 2021.

[38] A. Jahanian, L. Chai, and P. Isola, "On the" steerability" of generative adversarial networks," *arXiv preprint arXiv:1907.07171*, 2019.

[39] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 3, pp. 1–21, 2021.

[40] X. Hu, Q. Huang, Z. Shi, S. Li, C. Gao, L. Sun, and Q. Li, "Style transformer for image inversion and editing," in *CVPR*, 2022, pp. 11 337–11 346.

[41] X. Yao, A. Newson, Y. Gousseau, and P. Hellier, "A latent transformer for disentangled face editing in images and videos," in *ICCV*, 2021, pp. 13 789–13 798.

[42] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," in *CVPR*, 2022, pp. 2426–2435.

[43] M. Kwon, J. Jeong, and Y. Uh, "Diffusion models already have a semantic latent space," in *ICLR*, 2023.

[44] G. Kwon and J. C. Ye, "Diffusion-based image translation using disentangled style and content representation," in *ICLR*, 2023.

[45] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *ICCV*, 2017, pp. 5907–5915.

[46] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*. PMLR, 2016, pp. 1060–1069.

[47] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *CVPR*, 2018, pp. 1316–1324.

[48] J. Shi, N. Xu, T. Bui, F. Dernoncourt, Z. Wen, and C. Xu, "A benchmark and baseline for language-driven image editing," in *ACCV*, 2020.

[49] Y. Cheng, Z. Gan, Y. Li, J. Liu, and J. Gao, "Sequential attention gan for interactive image editing," in *ACM MM*, 2020, pp. 4383–4391.

[50] J.-H. Kim, N. Kitaev, X. Chen, M. Rohrbach, B.-T. Zhang, Y. Tian, D. Batra, and D. Parikh, "Codraw: Collaborative drawing as a testbed for grounded goal-driven communication," *arXiv preprint arXiv:1712.05558*, 2017.

[51] R. Y. Benmalek, C. Cardie, S. Belongie, X. He, and J. Gao, "The neural painter: Multi-turn image generation," *arXiv preprint arXiv:1806.06183*, 2018.

[52] T.-J. Fu, X. E. Wang, S. Grafton, M. Eckstein, and W. Y. Wang, "Sscr: Iterative language-based image editing via self-supervised counterfactual reasoning," *arXiv preprint arXiv:2009.09566*, 2020.

[53] Y. Liu, M. De Nadai, D. Cai, H. Li, X. Alameda-Pineda, N. Sebe, and B. Lepri, "Describe what to change: A text-guided unsupervised image-to-image translation approach," in *ACM MM*, 2020, pp. 1357–1365.

[54] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in *CVPR*, 2020, pp. 7880–7889.

[55] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "Chatpainter: Improving text to image generation using dialogue," *arXiv preprint arXiv:1802.08216*, 2018.

[56] Y. Liao, K. Schwarz, L. Mescheder, and A. Geiger, "Towards unsupervised learning of generative models for 3d controllable image synthesis," in *CVPR*, 2020, pp. 5871–5880.

[57] A. Szabó, G. Meishvili, and P. Favaro, "Unsupervised generative 3d shape learning from natural images," *arXiv preprint arXiv:1910.00287*, 2019.

[58] M. Gadelha, S. Maji, and R. Wang, "3d shape induction from 2d views of multiple objects," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 402–411.

[59] P. Henzler, N. J. Mitra, and T. Ritschel, "Escaping plato's cave: 3d shape from adversarial rendering," in *ICCV*, 2019, pp. 9984–9993.

[60] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "Hologan: Unsupervised learning of 3d representations from natural images," in *ICCV*, 2019, pp. 7588–7597.

[61] T. H. Nguyen-Phuoc, C. Richardt, L. Mai, Y. Yang, and N. Mitra, "Blockgan: Learning 3d object-aware scene representations from unlabelled images," *NeurIPS*, vol. 33, pp. 6767–6778, 2020.

[62] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," *NeurIPS*, vol. 29, 2016.

[63] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. Tenenbaum, and B. Freeman, "Visual object networks: Image generation with disentangled 3d representations," *NeurIPS*, vol. 31, 2018.

[64] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[65] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis," in *CVPR*, 2021, pp. 5799–5809.

[66] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *NeurIPS*, vol. 33, pp. 20 154–20 166, 2020.

[67] Y. Lan, X. Meng, S. Yang, C. C. Loy, and B. Dai, "E3dge: Self-supervised geometry-aware encoder for style-based 3d gan inversion," *arXiv preprint arXiv:2212.07409*, 2022.

[68] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015, pp. 3730–3738.

[69] X. Pan, B. Dai, Z. Liu, C. C. Loy, and P. Luo, "Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans," in *ICLR*, 2021.

[70] X. Wang, Y. Li, H. Zhang, and Y. Shan, "Towards real-world blind face restoration with generative facial prior," in *CVPR*, 2021.

[71] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

[72] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[74] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[75] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[76] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[77] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting deep generative prior for versatile image restoration and manipulation," in *ECCV*. Springer, 2020, pp. 262–277.

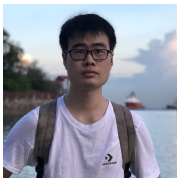
[78] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," in *ECCV*. Springer, 2020, pp. 592–608.

[79] T. M. Dinh, A. T. Tran, R. Nguyen, and B.-S. Hua, "Hyperinverter: Improving stylegan inversion via hypernetwork," in *CVPR*, 2022, pp. 11 389–11 398.

[80] T. Wang, Y. Zhang, Y. Fan, J. Wang, and Q. Chen, "High-fidelity gan inversion for image attribute editing," in *CVPR*, 2022, pp. 11 379–11 388.

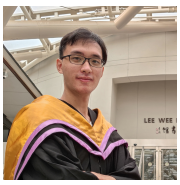
[81] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano, "Hyperstyle: Stylegan inversion with hypernetworks for real image editing," in *CVPR*, 2022, pp. 18 511–18 521.

[82] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, "Pivotal tuning for latent-based editing of real images," *ACM Transactions on graphics (TOG)*, vol. 42, no. 1, pp. 1–13, 2022.



Yuming Jiang is currently a Ph.D. student at MMLab@NTU, Nanyang Technological University, supervised by Prof. Ziwei Liu and Prof. Chen Change Loy. He got his bachelor degree in computer science from Yingcai Honors College, University of Electronic Science and Technology of China (UESTC) in 2019. He received the Google PhD Fellowship in 2022. His research interests include image generation, manipulation and restoration.

Ziqi Huang is currently a Ph.D. student at MMLab@NTU, Nanyang Technological University (NTU), supervised by Prof. Ziwei Liu. She received her Bachelor's degree from NTU, School of Electrical and Electronic Engineering in 2022. Her current research interests include generative models, visual generation and manipulation.



Tianxing Wu is currently a Research Associate at MMLab@NTU, Nanyang Technological University, supervised by Prof. Ziwei Liu. He received the BEng degree from Harbin Engineering University in 2020, and the MSc degree from Nanyang Technological University in 2021. His current research interests include visual generation and multi-modal deepfake detection.

Xingang Pan received the PhD degree in information engineering from The Chinese University of Hong Kong in 2021. He is currently an assistant professor at School of Computer Science and Engineering, Nanyang Technological University. Previously, he was a postdoctoral researcher at Max Planck Institute for Informatics. His research interests include generative models and neural rendering. He has won the 2017 Hong Kong PhD Fellowship Scheme (HKFPS) award, the 2017 Tusimple Lane Detection Challenge, and the 2018 WAD Drivable Area Segmentation Challenge. He is a member of the IEEE.

Chen Change Loy (Senior Member, IEEE) is a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his Ph.D. (2010) in Computer Science from the Queen Mary University of London. Prior to joining NTU, he served as a Research Assistant Professor at the MMLab of The Chinese University of Hong Kong, from 2013 to 2018. He was a postdoctoral researcher at Queen Mary University of London and Vision Semantics Limited, from 2010 to 2013. He serves as an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision and Computer Vision and Image Understanding. He also serves/served as an Area Chair of major conferences such as ICCV, CVPR, ECCV, NeurIPS and ICLR. He is a senior member of IEEE. His research interests include image/video restoration and enhancement, generative tasks, and representation learning.

Ziwei Liu is currently a Nanyang Assistant Professor at Nanyang Technological University, Singapore. His research revolves around computer vision, machine learning and computer graphics. He has published extensively on top-tier conferences and journals in relevant fields, including CVPR, ICCV, ECCV, NeurIPS, ICLR, ICML, TPAMI, TOG and Nature - Machine Intelligence. He is the recipient of Microsoft Young Fellowship, Hong Kong PhD Fellowship, ICCV Young Researcher Award, HKSTP Best Paper Award and WAIC Yunfan Award. He serves as an Area Chair of CVPR, ICCV, NeurIPS and ICLR, as well as an Associate Editor of IJCV.