Research Article

Make-A-Character 2: Animatable 3D Character Generation From a Single Image

Lin Liu[1], Yutong Wang[1], Jiahao Chen[1], Jianfang Li[1], Tangli Xue[1], Longlong Li[1], Jianqiang Ren[1], Liefeng Bo[1]

1. Tongyi Lab, Alibaba Group (China), Hangzhou, China

This report introduces Make-A-Character 2, an advanced system for generating high-quality 3D characters from single portrait photographs, ideal for game development and digital human applications. Make-A-Character 2 builds upon its predecessor by incorporating several significant improvements for image-based head generation. We utilize the IC-Light method to correct non-ideal illumination in input photos and apply neural network-based color correction to harmonize skin tones between the photos and game engine renders. We also employ the Hierarchical Representation Network to capture high-frequency facial structures and conduct adaptive skeleton calibration for accurate and expressive facial animations. The entire image-to-3D-character generation process takes less than 2 minutes. Furthermore, we leverage transformer architecture to generate co-speech facial and gesture actions, enabling real-time conversation with the generated character. These technologies have been integrated into our conversational AI avatar products.

1. Introduction

Make-A-Character[1] introduced a system that generates 3D characters based on text descriptions. Although text is effective and lightweight, it struggles to describe the precise nuances of facial features, limiting its controllability on detailed attributes. As the saying goes, a picture is worth a thousand words, images can convey far more detailed information than text, allowing users to generate characters with greater intuitiveness and control.

Given the aforementioned content, we introduce Make-A-Character 2, a system for 3D characters generation from single images. We wish to generate a 3D character with consistent appearance (*e.g.*,

 doi.org/10.32388/H03RE1

face, hairstyle, skin color) given a frontal face portrait photo, and the 3D character is equipped with sophisticated underlying skeleton and ready for animation. The Make-A-Character 2 inherits key properties from Make-A-Character, such as highly-realistic rendering, full-body completed and industry-compatible. Further more, we propose the following major improvements:

Portraits Illumination Harmonization. To ensure the high quality of the generated 3D head, ideal input portraits should be captured under optimal lighting conditions. This includes uniform lighting that avoids asymmetry, excessive shadows, or overexposure. However, daily portraits often fail to meet these strict requirements. To address this issue, we utilize the latest diffusion-based illumination editing method IC-Light[2] to correct the lighting in the given photos, bringing them to an ideal illumination state for subsequent operations.

Color Correction in Game Engine. We aim to ensure that the generated 3D character maintains consistent lighting and skin tone with the input photo. However, predicting the lighting environment from an input portrait and replicating it in render engine is challenging. The final color appearance in a game engine cannot be precisely predicted from a diffuse texture map alone due to various influencing factors, such as lighting setup, material properties, reflections, subsurface scattering, and ambient occlusion. We draw inspiration from[3] and address the challenge of non-differentiability in game engines by training a neural network to learn the color discrepancies between diffuse maps and their corresponding renders from the game engine. After the network successfully models this color offset, we utilize the learned model to adjust the diffuse map, allowing the render engine to produce the desired color outcome more accurately.

Detailed Face Reconstruction. Our initial head geometry, reconstructed guided by facial dense landmarks, often lacks high-frequency details. To address this limitation, we leverage HRN[4] facial geometry to augment the surface details of our initial model. For essential facial regions, such as around the eyes and lips, we increase the density of landmark predictions, enabling to capture intricate nuances with greater precision.

Facial Skeleton Calibration. According to industry standards, the creation of 3D facial assets necessitates a well-designed skeletal structure and specialized rigging to achieve satisfactory animation outcomes. In our work, we conducted skeleton fitting on the reconstructed facial meshes, utilizing several neutral face models sourced from MetaHuman, which had already undergone comprehensive rigging processes as described in[5]. This approach guarantees that each facial mesh

produced through the aforementioned geometric reconstruction algorithm is equipped with a customized skeletal configuration.

Real-Time Speech-driven Animation. Although there are many researches[6][7][8][9][10] addressing the problem of audio-driven motion generation, the produced expression and gesture animations are far from satisfactory. They mainly adopt ARKit blendshapes to represent the facial expressions. The ARKit blendshapes are not able to express detailed mouth shapes. As for the gesture, their generated results are not restricted and the motions are jittering. In order to produce realistic motions, we follow Metahuman's control rig to animate facial expressions and employ two senior animators to revise motion capture data. Instead of generating gestures freely, we predict indices of motion pieces, which are intentionally designed by animators. This allows our system to generate natural and realistic body motions.

2. Related Work

Single-View 3D Reconstruction. Reconstructing a 3D object from a single image is an ill-posed problem that typically relies on specific prior knowledge. Inspired by the Scaling Law, which demonstrates notable success in natural language processing (NLP) tasks through the utilization of scalable network architectures and large training datasets, recent works[11][12] propose a large-scale 3D reconstruction model based on the Transformer architecture. These models aim to extract a generalized 3D prior from extensive datasets, thereby enabling the prediction of Neural Radiance Field (NeRF) representations from a single image. While these methods demonstrate remarkable reconstruction quality and generalization capabilities, they face significant challenges in the realm of 3D avatar generation, which demands sophisticated structures to ensure animatability and compatibility with existing industry CG pipelines.

Single-View 3D Head Generation. The 3D Morphable Model (3DMM)[13] captures the variability inherent in human face by representing facial geometries as a linear combination of blendshapes. This approach has become a foundational technique in 3D face reconstruction. By fitting the model parameters to 2D images, it enables face reconstruction from single images.

With the advent of deep learning, researchers[14][15] have developed methods to leverage neural networks for predicting 3D face shapes directly from images. To better capture fine details, many approaches aim to improve the 3D Morphable Model (3DMM)-based framework by introducing

nonlinear 3DMM[16], deformation maps generation[4], and displacement maps generation[17] for higher-quality geometry. Regarding texture, the traditional linear combination of texture bases often results in a blurred texture map. Thus, techniques like differentiable rendering[4] or directly unwrapping the input image into UV space[18] are more effective alternatives.

Advancements in neural 3D representations have led to the development of a variety of models for detailed human head reconstruction. The Neural Parametric Head Models (NPHM)[19] effectively use a signed distance field (SDF) to achieve high-fidelity capture of complete human head geometry. Despite their detailed geometrical focus, NPHMs do not support appearance modeling or hair reconstruction. In contrast, Rodin[20] employs diffusion models to create 3D digital avatars using neural radiance fields (NeRF), which incorporate both geometry and appearance. Additionally, the 3D Gaussian Parametric Head Model (GPHM)[21] leverages 3D Gaussian to deliver photorealistic rendering quality and real-time performance. Nonetheless, several challenges persist, including integrating the Gaussian head model with a full body and adapting these models to varying lighting conditions.

Speech-driven 3D Gesture Generation. Gesture generation is a complex task that requires understanding speech, gestures, and their relationships. To address this, BEAT[22] presents a high-quality, multi-modal motion data set captured from 30 speakers. Along with the data set a Cascaded Motion Network is proposed to generate gestures from speech. Based on this data set, EMAGE[23] and ProbTalk[24] utilize masked transformer to predict gestures from audio in the latent space of VQ-VAE. Although improved results are observed, the motion diversity is restricted. Diffusion based models[8][9][7] are also explored in the literature of speech-driven motion generation. Compared to the previous methods, these could generate diverse motions. However, the results produced by all of these methods are far from industrial standards. The predicted motions are usually jittering. Instead of unrestricted generation, we directly predict the indices of predefined motions, which are designed by senior animators. Based on this, it is able to generate realistic gestures motions.

Speech-driven 3D Face Animation. Speech-driven 3D face animation focuses on creating realistic facial animations from the input speech signals. This field has developed for a long history and achieved significant progress, which can be broadly categorized into visme-based and deep learning-based methods. Visme-based methods primarily focused on establishing mapping relationships from phonemes to visemes[25][26][27]. These methods allow for explicit control and can be easily integrated

into industrial animation pipelines. However, they have some disadvantages: they mainly concentrate on lip region animations and lack comprehensive strategies for animating the entire face. Additionally, the expressiveness of these methods is limited, making it difficult to capture detailed lip movements. Recently, deep learning-based methods have been increasingly researched. Karras et al. [28] proposed an deep neural network to learn a mapping from input waveforms to 3D vertex coordinates of a face model. VOCA[29] utilizes an encoder-decoder network where the encoder learns to transform audio features to a low-dimensional embedding and the decoder maps this embedding into a high-dimensional space of 3D vertex displacements. Faceformer[30] also leverages an encoder-decoder model, but with Transformer-based architecture and autoregressively generates a sequence of animated 3D face meshes from input raw audio. To avoid over-smoothed facial motions, Codetalker[31] adopts a pre-trained VQ-VAE motion prior. MeshTalk[32] achieves highly realistic motion synthesis results for the entire face based a categorical latent space that disentangles audio-correlated and audio-uncorrelated information. SelfTalk[33] involve a self-supervision framework to exchange cross-modals information to generating realistic and accurate lip movements with lipreading comprehensibility. However, the methods mentioned above are all based vertex offset, usually lacking components like teeth, tongue, and eyelashes, leading to flaws during animation. Additionally, they are difficult to integrate into existing animation production pipelines. However, these approaches can serve as valuable references.

3. 3D Character Generation

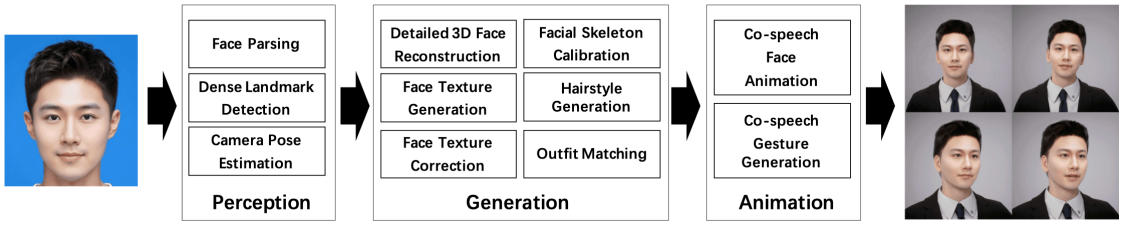


Figure 1. Pipeline for generating animatable 3D characters from a single image.

doi.org/10.32388/H03RE1

3.1. Geometry Generation

3.1.1. Hierarchical Facial Details Transfer

Unlike our prior work[1] which relied on textural descriptions, this research uses a single portrait image as input, requiring more accurate and detailed face generation. Our prior triplane-based method, which used total variation loss for smoothness, struggled to reconstruct intricate facial details like wrinkles and dimples in the initial head geometry. Lei et al.[4] proposed a Hierarchical Representation Network (HRN) enabling highly accurate single-image facial reconstruction. We utilize this HRN to transfer detailed facial features to our initial head model, overcoming previous geometric limitations. To align the detail-rich HRN geometry with our model, we apply a rigid transformation based on 7 facial landmarks, followed by details geometry transfer. However, direct transfer can induce artifacts at the junctions between the detail-enhanced facial areas and the fixed head region. To ensure a seamless transition and achieve a natural appearance, we employ a smoothing operation at these junctions. The complete pipeline is shown in Figure 2.

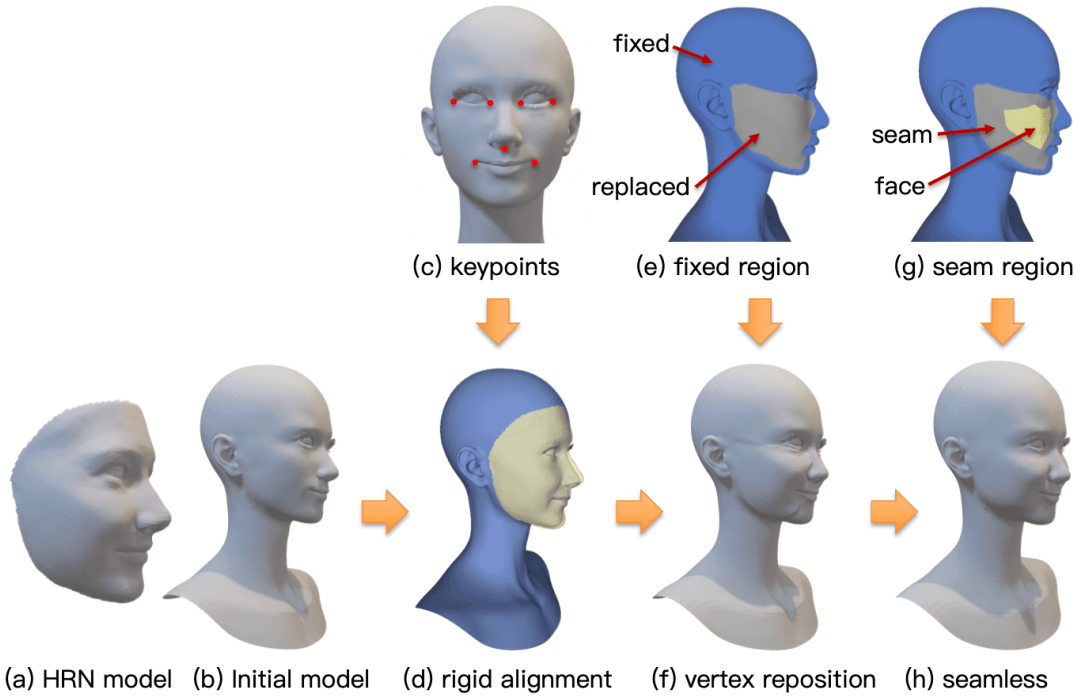


Figure 2. The facial details transfer pipeline. We align the HRN geometric model (a) with the initial geometric model (b) through a rigid alignment process (d) based on seven key points indicated in (c). Following this, we fix the positions of the points within the blue region, as depicted in (e) of our initial model, and subsequently transfer the HRN facial details to the replaceable gray region also shown in (e), leading to the creation of a replaced head model with HRN face details (f). Finally, we determine the gray transition region as shown in (g) and apply a smoothing technique to attain a final model (h) that seamlessly integrates facial details and exhibits a natural appearance.

Figure 3 showcases the effectiveness of our HRN-guided facial detail transfer. Using images from the SCUT-FBP5500[34] as input, the figure highlights a substantial enhancement in facial details. Notably, the pre-transfer faces appear smooth and lack fine features, whereas the post-transfer faces exhibit detailed characteristics, such as nasolabial folds and dimples.

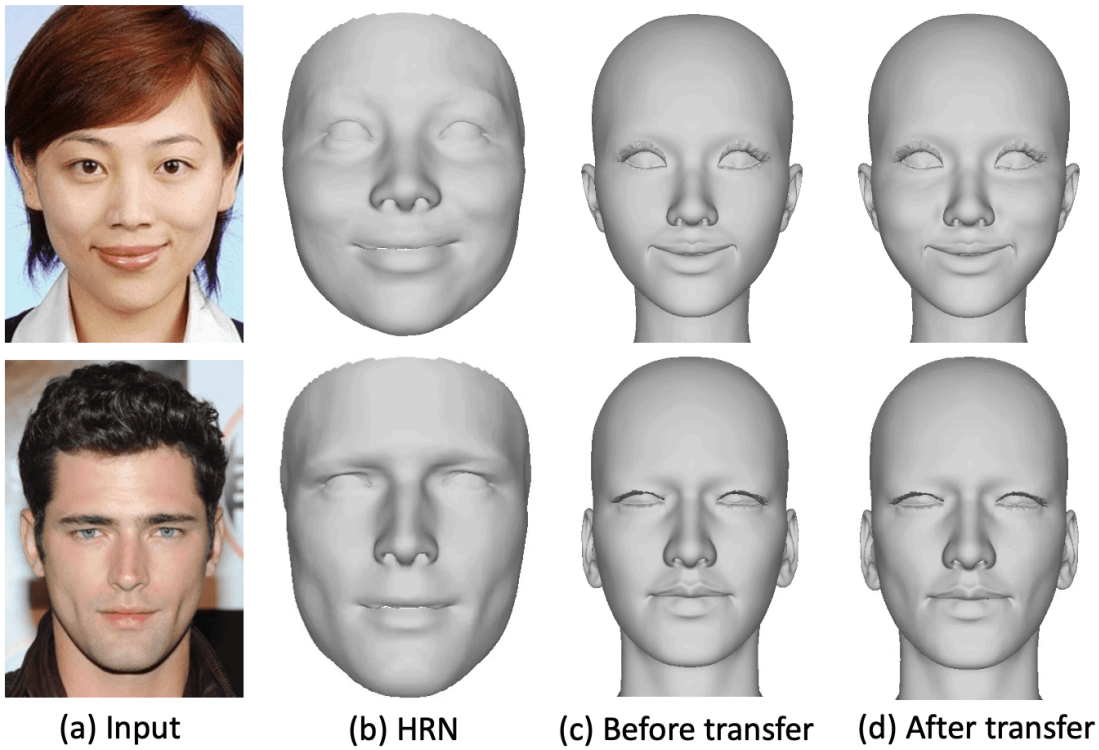


Figure 3. Results with facial details generated using our facial detail transfer pipeline.

The complexity of the eyes and mouth requires a more detailed representation than the original landmarks could provide. Therefore, we increased the landmark count to 20 for the eyes and 28 for the mouth. This crucial modification significantly improves the geometric precision of these facial features, resulting in superior alignment with the input image data. Figures 4 and 5 visually demonstrate this improvement.

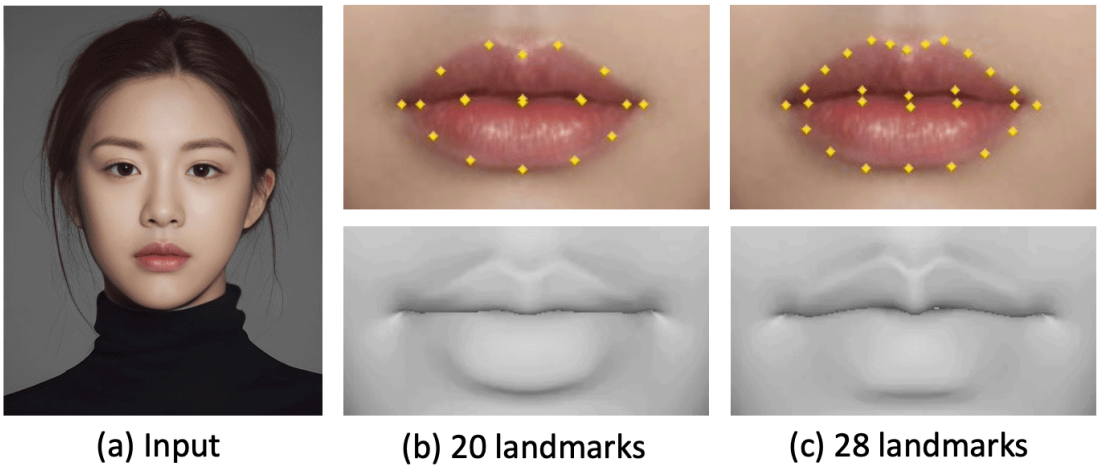


Figure 4. An example of mouth geometric results landmarks obtained by expanding the mouth landmarks

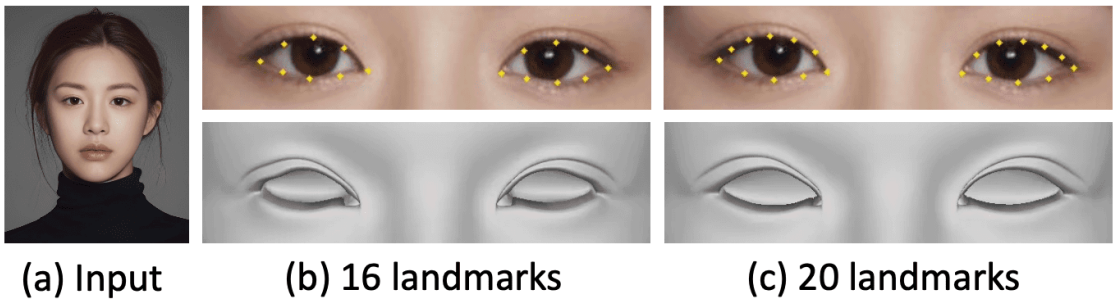


Figure 5. An example of eyes geometric results landmarks obtained by expanding the eyes landmarks

Because the nose and adjacent facial areas have similar skin tones, accurately detecting nose landmarks for shape reconstruction is a difficult task. To improve the precision of nose shape reconstruction, we leverage the shape knowledge encoded within the Basel Face Model (BFM). Specifically, we augment the nose representation with 80 shape bases from the BFM to obtain the final geometry V:

$$V = triplane[v, u, :] + \sum_{i=0}^{80} S_i \alpha_i \tag{1}$$

As shown in Figure 6, the nose shape of S_i is consistent with BFM shape basis, but the topology is geometrically identical to the triplane representation, and α_i is the nose shape coefficient.

doi.org/10.32388/H03RE1

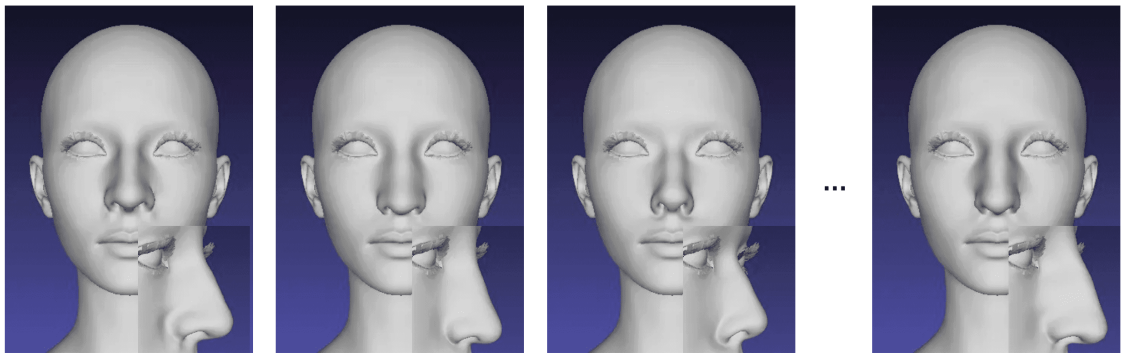


Figure 6. Nose Shape Basis Set Derived from the Basel Face Model.

3.1.2. Facial Skeletons Calibration

Once high-fidelity human face meshes are generated, accurate skeleton(*i.e.*, joint, bone) binding is crucial for ensuring the realistic and nuanced animation driving performance. We achieve this by fitting the skeleton (joints and bones) from a pre-rigged, neutral-shaped face mesh onto the newly generated target model.

Our approach begins by implementing the forward propagation of a skeleton-driven skinning system. This system uses a hierarchical, multi-branch, loop-free tree structure to organize hundreds of skeleton nodes, mirroring the rig logic systems of MetaHuman[35]. Each node stores its local translation and rotation relative to its parent. Once a skeleton moves, we update the absolute positions of all descendant nodes of this skeleton using forward kinematics. These displacements in absolute coordinate, combined with Linear Blend Skinning (LBS) weights, determine the displacements of mesh vertices and deform the mesh. Since our geometry reconstruction maintains consistent face-vertex topology, we can compute the vertex differences between the target mesh and the deformed mesh on a per-vertex basis. These differences are then used as the loss function for skeleton fitting:

$$L_v = \sum_{n=1}^{N} \|\mathbf{v}_i^{\text{current}} - \mathbf{v}_i^{\text{target}}\|_2^2 \tag{2}$$

where N denotes the number of vertices in the mesh, $\mathbf{v}_i^{\text{current}}$ and $\mathbf{v}_i^{\text{target}}$ are the (x, y, z) coordinates of the i-th vertex in the current deformed mesh and the target mesh, respectively. We aim to move the current vertices towards their corresponding target positions.

doi.org/10.32388/H03RE1

Leveraging the complete forward path from skeleton displacement to mesh vertex displacement, we can directly utilize automatic differentiation to optimize skeleton parameters. This process iteratively adjusts the skeleton position, aiming to minimize the loss and drive the current mesh closer to the target shape. This iterative fitting pipeline, visualized in Figure 7, continues until convergence is achieved.

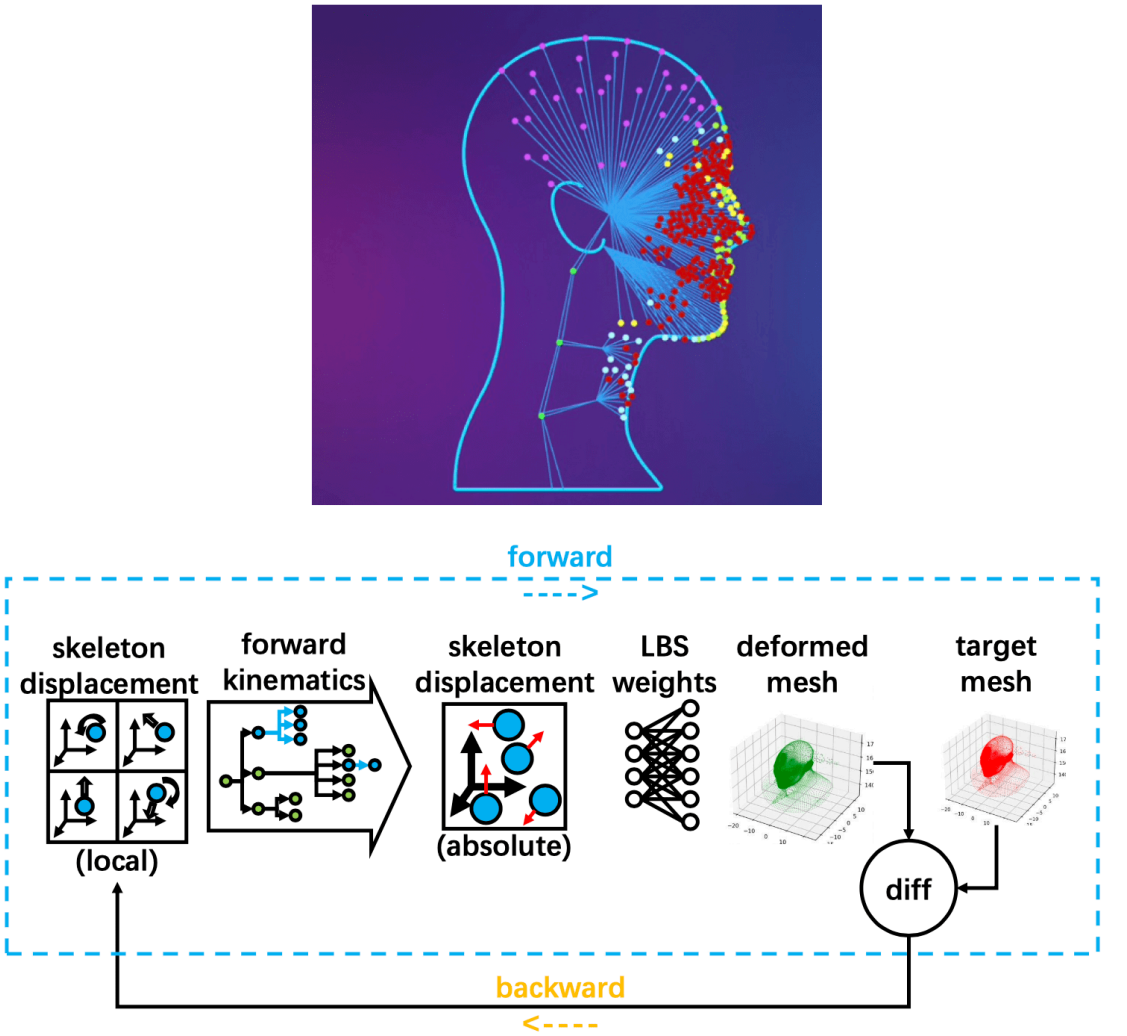


Figure 7. The skeleton calibration pipeline.

We also observe significant spatial overlap between the leaf node skeletons and the mesh vertices on both the neutral and rigged face meshes. We denote the index set of these overlapping leaf skeletons as Φ:

$$M(i) = j, \quad i \in \Phi \tag{3}$$

doi.org/10.32388/H03RE1

$$\mathbf{s}_i = \mathbf{v}_{M(i)}, \quad i \in \Phi \tag{4}$$

the function $M(.)$ maps each index of a skeleton to the corresponding index of its associated vertex in the mesh. Therefore, the positional differences between these overlapping skeleton nodes and their mesh counterparts are used as a supervision component in the fitting process:

$$L_s = \sum_{i \in \Phi} \| \mathbf{s}_i^{\text{current}} - \mathbf{v}_{M(i)}^{\text{current}} \|_2^2 \tag{5}$$

the total loss function is given by $L_{\text{total}} = L_v + L_s$.

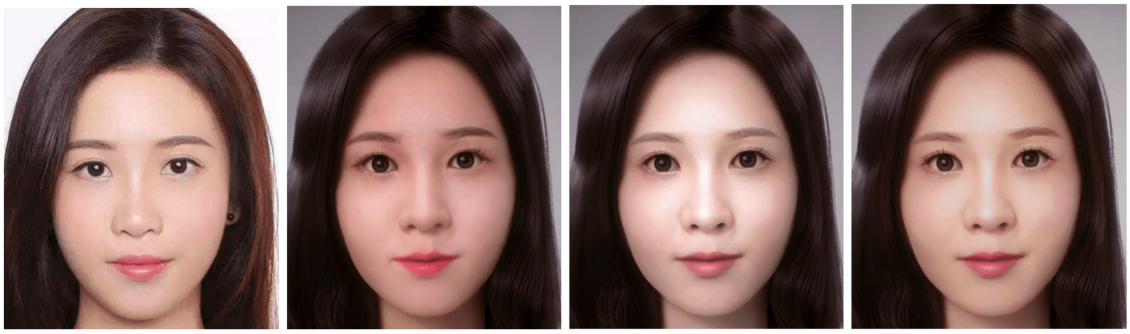
Through practical experimentation, we observed that enabling updates for all skeleton parameters during the optimization process can lead to unreasonable skeleton configurations. Specifically, the optimizer tends to induce exaggerated movements in the skeleton in an attempt to align the deformed mesh with the target mesh. To mitigate this issue, we impose a constraint by allowing updates only for the leaf node skeletons within the skeleton tree system.

Simply fine-tuning the skeletons alone is insufficient to perfectly match the target shape (the loss cannot be reduced to zero). Therefore, our process first applies the residual offset as a blend shape to the neutral face mesh. Subsequently, skeletons are adjusted according to the results of the above solution. This two-step calibration enables a well-rigged face mesh.

3.2. Texture Generation

3.2.1. Texture Correction

In our endeavor to accurately replicate the textures and shadings of the input facial images within Unreal Engine (UE), the illumination-independent diffuse albedos[1]prove to be unsuitable for this scenario, as demonstrated in Figure 8(b). While textures generated via differentiable rendering initially present as a promising choice, they inherently incorporate baked illumination, leading to significant discrepancies in light and shadow effects when rendered in UE compared to those observed in the input facial images, as depicted in Figure 8(c). Moreover, recreating identical lighting conditions in UE is particularly challenging due to the inherently ill-posed problem of inferring illumination from a single real-world image. To address these issues, we opt to establish a fixed lighting environment within UE and formulate the task of lighting replication as a classical image-to-image translation problem. By learning the differences between the rendered output and the input facial image, it corrects the textures generated by the differentiable rendering and reproduces the lighting effects of the input images, as illustrated in Figure 8(d).



(a) (b) (c) (d)

Figure 8. Comparison of different types of textures rendered in UE. (a) is the input image, (b), (c), (d) are generated 3D characters using diffuse albedo[1], texture generated from differentiable rendering and the corrected textures.

It is well known that the rendering process in UE is not differentiable. To overcome this limitation and facilitate back-propagation of shadings onto facial textures, we propose a neural network, denoted as $\mathcal{N}$, to emulate the inverse rendering process of the UE. As in Figure 9, $\mathcal{N}$ maps the rendered color $\mathcal{C}$ to a corrected color $\mathcal{C}'$, such that

$$\mathcal{C}' = \mathcal{N}(\mathcal{C}) \tag{6}$$

When the corrected color $\mathcal{C}'$ is rendered through UE, the rendered result yields to $\mathcal{C}$, indicating successful replication of the rendered color. To achieve the replication of the input facial images' illumination, we apply $\mathcal{N}$ to adjust the textures generated by the differentiable rendering process.

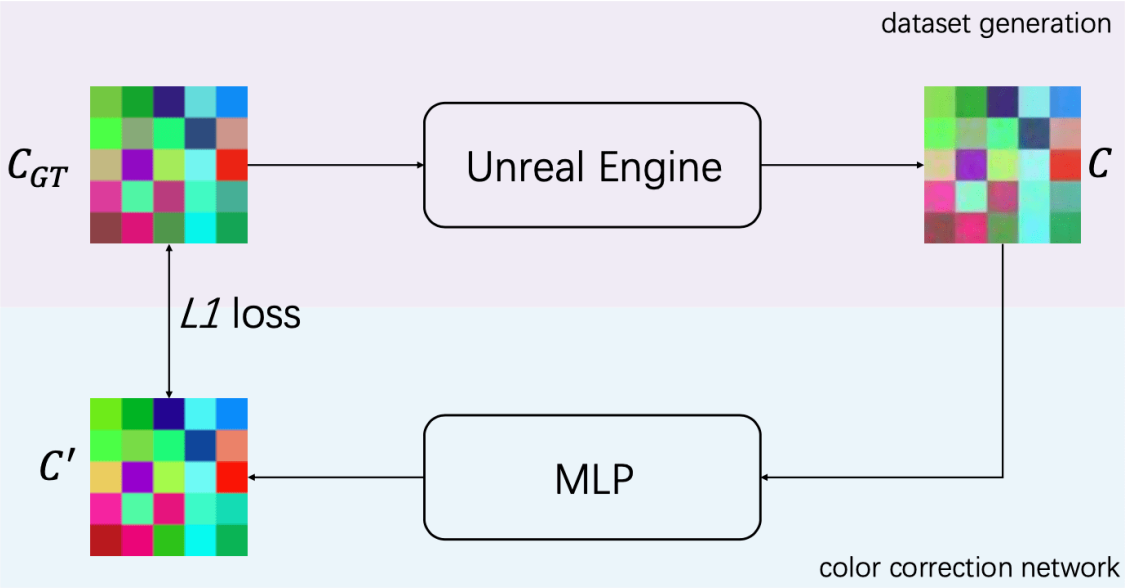


Figure 9. The color correction pipeline.

We implement the network $\mathcal{N}$ as a three-layer MLP with 32 hidden units, and define the loss function as $\sum_{i=1}^{n} \|\mathcal{C}_i' - \mathcal{N}(\mathcal{C}_i)\|$. The training data is generated through an intuitive approach: we render a variety of colors $\mathcal{C}'$ in UE and store the results as $\mathcal{C}$, yielding over 10,000 color pairs for training.

3.2.2. Facial Lighting Normalization

When dealing with facial images under atypical lighting, such as non-uniform, colored, or overexposed lighting conditions, the corrected textures often retain these lighting features, including shadings and highlights. As a result, it presents a dilemma, while these lighting features enhance the fidelity of the rendered face compared to the input, they also create visual inconsistencies within the uniformly lit UE environment. To address this issue, we employ the IC-light algorithm[2] to attenuate these atypical lighting features and achieve normalized facial lighting. Given a facial image and the uniform light map, the IC-light algorithm re-lights the face according to the new lighting condition. To partially maintain the original shadings and highlights, we blend the re-lit face with the input face, yielding a similar yet even lighting facial image. Figure 10(a) and Figure 10(b) present the 3D characters generated by the input facial image and the normalized facial image, demonstrating that the latter's shadings and highlights are more natural and harmonious within our UE environment.

Figure 10. Results of the relighted image. This figure presents the 3D characters generated by the input facial image and the normalized facial image, respectively.

3.3. Hair Generation

When generating 3D avatars from single images, capturing the hairstyle is crucial for achieving an accurate likeness. To facilitate this, we utilize a specialized hairstyle classification model powered by convolutional neural networks. This model directly maps a portrait image to a specific hairstyle within our asset library, bypassing the need for intermediate feature extraction. Many hairstyles naturally showcase asymmetry, as seen in the way long hair flows over the chest or back and the placement of bangs. In our asset library, we've enhanced these asymmetrical hairstyles by applying a horizontal flip, treating the flipped versions as unique styles. Consequently, when labeling the training data, we distinctly differentiate between the left and right variations of these hairstyles.

4. 3D Character Animation

In our application we mainly concerned to animate the generated avatars via speech audio. We leveraged bone skeleton and facial control rig to generate gesture and facial motion, respectively.

4.1. Speech-driven Gesture Motions

Researchers have explored masked transformers[23][24][10] or diffusion models[8][9][7] to predict gestures from speech with or without additional conditions, like word text, emotional labels or semantic labels. They all freely generate gestures by learning distribution from motion capture data. Their jittering results do not meet the industrial requirements. In contrast, we suggest to predict predefined motion pieces for realistic animations.

In order to produce realistic animations, we use Vicon, which is a sophisticated motion capture system, to collect body motion data, and employ two senior animators to adapt the data to our generated avatars. We capture more than one hundred pieces of motion data. The length of each piece of data falls between 2 seconds and 3 seconds. These data are common nonverbal gestures made by people while speaking, and they can be divided into 5 categories, which responds to 5 human poses (see Figure 11). In each category, the start and end pose of each data are same. This allows the motion data in each category to be smoothly concatenated together to form longer animation. Besides, there is motion data for transition between different poses. These captured data form the foundation of our training data.

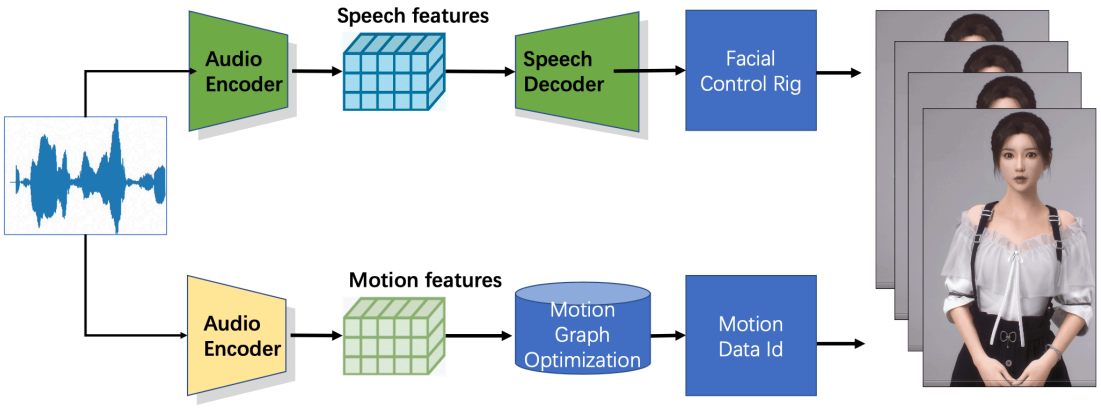


Figure 11. Pipeline of animations. Coefficients of facial control rig were directly predicted from speech audio. Gesture motion clips were selected from the captured data set by optimizing graph path.

doi.org/10.32388/H03RE1

Figure 12. Captured motion data can be divided into 5 categories corresponding to 5 poses. In each category, the start and end poses of each data are same. This enables us to generate longer animation by concatenating any two pieces of motion data in a single category.

The animators generate sequences of motion data based on speech audios as our training data set. To align audio with motion sequences in the data set, we use graph-based optimization method to generate motion sequences. We take each piece of the data as a graph node, and there is a directed edge between two nodes if these two nodes are in continuous order in the training data set. The weight associated with a edge is defined as follows:

$$T(N_i, N_j) = \lambda_1 T_p(N_i, N_j) + \lambda_2 T_r(N_i, N_j) \tag{7}$$

where N_i denotes the ith graph node, $T_p(N_i, N_j)$ is the translation loss between two nodes, $T_r(N_i, N_j)$ is the rotation loss of corresponding joints between two nodes. We also associate each node with an audio embedding feature which extracted from Wav2Vec2. Given an audio sequence $M_1, M_2, .., M_n$, the corresponding motion sequence is obtained by an optimal path in the directed graph. The cost is defined as follows:

$$C = \sum_{i}^{n} C_a(A_i, \hat{A}_i) + \sum_{i}^{n-1} T(i, i+1) \tag{8}$$

where $C_a(A_i, \hat{A}_i)$ is the audio embedding loss. A_i and $\hat{A}_i$ denote the embedding features of audio clip M_i and graph node N_i, respectively. $T(i, i+1)$ is the edge loss between the selected nodes with audio clip M_i and M_{i+1}. We use The viterbi algorithm[36] to solve this optimization problem.

In application we optimize the graph path within a specific motion data category. For lengthy speech audio, we randomly choose transition data to generate motions in another category. This strategy

enables us to generate a wide range of realistic animations.

4.2. Speech-Driven 3D Facial Animations

There are typically two parametric representation for creating 3D facial animations: vertex displacement and blendshapes. While vertex displacement methods can control facial movements with great precision, it is challenging to animate components such as teeth, tongue, and eyelashes, which can result in noticeable imperfections. On the other hand, blendshape-based solutions are stable and less prone to imperfections. However, the expressive capability of such solutions depends on the number of blendshapes available. A larger number of blendshapes allows for more nuanced expressions and improved animation quality, whereas a smaller number may result in less satisfactory outcomes. We adopt the blendshape-based approach, leveraging the robust Control Rig system in Metahuman[37] to generate realistic facial animations. In order to obtain high-quality facial animations data, we employ a tool called MetaHuman Animator[37] to capture ctrlrig coefficients and then refine them with the expertise of experienced animators.

The entire pipeline of speech-driven 3D Facial Animations is shown in Figure 11. Firstly, we utilize the state-of-the-art self-supervised pre-trained speech model, wav2vec 2.0[38], to encode audio signals. Secondly, a non-autoregressive architecture is employed as the decoder to directly regress facial animation weights from audio features. In contrast to autoregressive transformer-based methods, the non-autoregressive method only call decoder once, so it more efficient than autoregressive method, which need a iterative decoding loop. We only employ reconstruction loss and velocity loss which is also used in Selftalk[33] to train the model. The loss function is formulated as:

$$L = L_{\text{rec}} + L_{\text{vel}} \tag{9}$$

The reconstruction loss L_{rec} measures the difference between the predicted ctrlrig coefficients and the ground-truth ctrlrig coefficients. Here, $b_{t,i}$ denotes the ground truth coefficients of the i^{th} ctrlrig at timestamp t, whereas $\hat{b}_{t,i}$ represents the predicted value. T is the length of the sequence, N is the number of ctrlrig.

$$L_{\text{rec}} = \sum_{t=1}^{T} \sum_{i=1}^{N} \| b_{t,i} - \hat{b}_{t,i} \|_2^2 \tag{10}$$

The velocity loss L_{vel} is used to reduce lip jittery over time.

$$L_{\text{vel}} = \sum_{t=1}^{T} \sum_{i=1}^{N} \| (b_{t,i} - b_{t-1,i}) - (\hat{b}_{t,i} - \hat{b}_{t-1,i}) \|_2^2 \tag{11}$$

In order to enhance the facial expressions of digital avatars, we pre-create templates for words with strong emotions, such as interjections. Figure 13 show some screenshots of the Chinese interjections we created. When these interjections are detected in the input audio, the corresponding facial expression templates are triggered to enhance the expression of emotions.

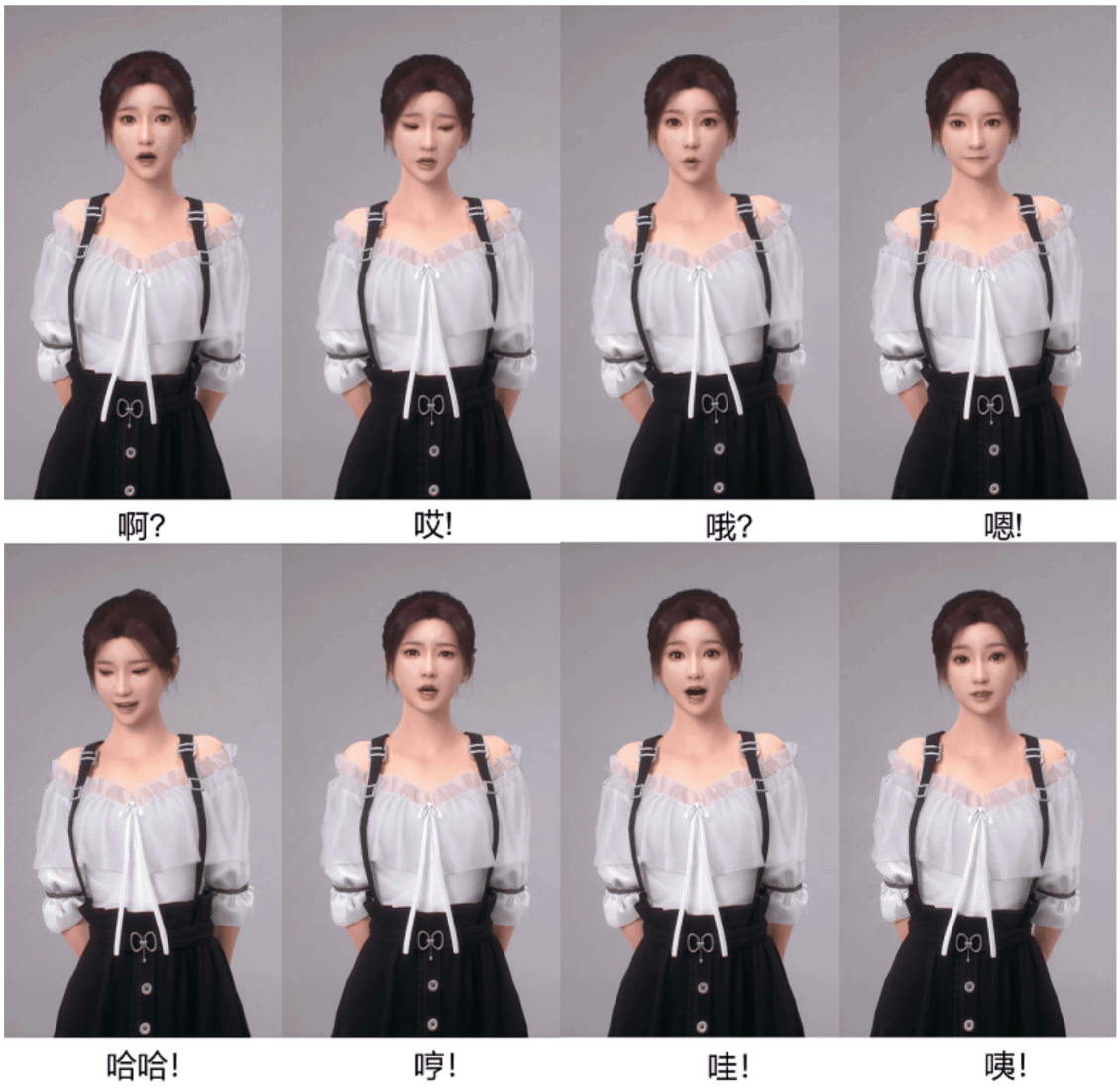


Figure 13. Screenshots of some Chinese interjections template animations

doi.org/10.32388/H03RE1

5. Results

5.1. 3D Character Generation

Figure 14 presents the 3D characters created from single images with our method. We compare our method with several state-of-art 3D avatar generation algorithms, including two text-based avatar generation methods(DreamFace[39], Rodin[40]), and two single image-based avatar generation methods (MeInGame[41] and MoSAR[42]). As shown in Table 1, Our method generates complete, production-ready 3D character assets compatible with modern CG pipelines, offering enhanced rendering fidelity, highly accurate rigging, and improved editability compared to other approaches.

Figure 14. Textured 3D character models generated from frontal portrait images. We utilize Unreal Engine for high-fidelity rendering.

doi.org/10.32388/H03RE1

Method	High Fidelity Generation	Full-Body Completeness	Facial Rigging Fidelity	CG Engine Compatible	Outfit Editable
DreamFace[39]	⋆⋆⋆	×	⋆	✓	✓
Rodin[40]	⋆⋆	×	×	×	×
MeInGame[41]	⋆	✓	⋆	✓	✓
MoSAR[42]	⋆⋆	×	⋆	✓	✓
Ours	⋆⋆⋆	✓	⋆⋆⋆	✓	✓

Table 1. Comparison of latest related 3D avatar generation algorithms.

5.2. 3D Character Animation

Our system combines the facial animation and gesture animation, it can generate holistic full-body motions. Figure 15 shows holistic gesture motions and facial animations generated from English and Chinese speeches. Benefit from high-quality mouth animations data, and non-autoregressive decoder architecture, our system can generate realistic lip-sync with efficiency exceeding real-time performance. Additionally, suitable expressions are also incorporated into facial animations during speech to enhance emotions. Our predefined motion data set guarantees no jittering gestures. And the gesture motions are aligned well with speech rhythm(See results in appendix videos).

Figure 15. Full-body animations sequences generated by our system with english speech(top row) ,Chinese speech(middle row) and Chinese speech with interjection(bottom row).

6. Limitation

Although our method excels in generating high-quality 3D character assets and animations, it still has several limitations. Firstly, when the input face does not have a neutral expression, our expression correction process may introduce artifacts, leading to unsatisfactory facial rig results. Additionally, our generated characters currently lack the ability to perform complex dynamic animations, such as singing or dancing, due to a lack of specific training data. Furthermore, the general nature of our animation model limits the personalization of facial and body motion synthesis. We are actively researching ways to overcome these limitations.

References

1. a, b, c, dRen J, He C, Liu L, Chen J, Wang Y, Song Y, Li J, Xue T, Hu S, Chen T, Zheng K, Xiang J, Bo L (2023). "Make-A-Character: High Quality Text-to-3D Character Generation within Minutes". arXiv preprint arXiv:2312.15430. Available from: https://arxiv.org/abs/2312.15430.
2. a, bZhang L, Rao A, Agrawala M (2024). "IC-Light GitHub Page". GitHub. Available from: https://github.com.
3. ^Shi T, Yuan Y, Fan C, Zou Z, Shi Z, Liu Y (2019). "Face-to-parameter translation for game character auto-creation". Proceedings of the IEEE/CVF International Conference on Computer Vision. 161–170.
4. a, b, c, dLei B, Ren J, Feng M, Cui M, Xie X (2023). "A hierarchical representation network for accurate and detailed face reconstruction from in-the-wild images". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 394–403.
5. ^Epic Games, Inc (2022). "MetaHuman-DNA-Calibration GitHub Page". Available from: https://github.com/EpicGames/MetaHuman-DNA-Calibration.
6. ^Liu H, Zhu Z, Becherini G, Peng Y, Su M, Zhou Y, Zhe X, Iwamoto N, Zheng B, Black MJ. "EMAGE: Towards Unified Holistic Co-Speech Gesture Generation via Expressive Masked Audio Gesture Modeling". arXiv [cs.CV]. 2024. Available from: https://arxiv.org/abs/2401.00374.
7. a, b, cZhu L, Liu X, Liu X, Qian R, Liu Z, Yu L (2023). "Taming Diffusion Models for Audio-Driven Co-Speech Gesture Generation". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pages 10544–10553.
8. a, b, cYang S, Wu Z, Li M, Zhang Z, Hao L, Bao W, Cheng M, Xiao L. "DiffuseStyleGesture: Stylized Audio-Driven Co-Speech Gesture Generation with Diffusion Models." In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. International Joint Conferences on Artificial Intelligence Organization; 2023. p. 5860-5868. doi:10.24963/ijcai.2023/650.
9. a, b, cChen J, Liu Y, Wang J, Zeng A, Li Y, Chen Q (2024). "DiffSHEG: A Diffusion-Based Approach for Real-Time Speech-driven Holistic 3D Expression and Gesture Generation". In: CVPR.
10. a, bZhang X, Li J, Zhang J, Dang Z, Ren J, Bo L, Tu Z (2024). "SemTalk: Holistic Co-speech Motion Generation with Frame-level Semantic Emphasis". arXiv. arXiv:2412.16563 [cs.CV].
11. ^Hong Y, Zhang K, Gu J, Bi S, Zhou Y, Liu D, Liu F, Sunkavalli K, Bui T, Tan H (2023). "Lrm: Large reconstruction model for single image to 3d". arXiv preprint arXiv:2311.04400.

12. ^Tochilkin D, Pankratz D, Liu Z, Huang Z, Letts A, Li Y, Liang D, Laforte C, Jampani V, Cao Y (2024). "Tri posr: Fast 3d object reconstruction from a single image". arXiv preprint arXiv:2403.02151. Available fro m: https://arxiv.org/abs/2403.02151.

13. ^Blanz V, Vetter T (1999). "A morphable model for the synthesis of 3D faces". In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 187–194.

14. ^Deng Y, Yang J, Xu S, Chen D, Jia Y, Tong X (2019). "Accurate 3d face reconstruction with weakly-super vised learning: From single image to image set". In: Proceedings of the IEEE/CVF Conference on Comput er Vision and Pattern Recognition Workshops. 2019: 0–0.

15. ^Tewari A, Zollhofer M, Kim H, Garrido P, Bernard F, Perez P, Theobalt C. "Mofa: Model-based deep co nvolutional face autoencoder for unsupervised monocular reconstruction." In: Proceedings of the IEEE I nternational Conference on Computer Vision Workshops. 2017. p. 1274–1283.

16. ^Tran L, Liu X (2018). "Nonlinear 3d face morphable model". In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7346–7355.

17. ^Feng Y, Feng H, Black MJ, Bolkart T (2021). "Learning an animatable detailed 3D face model from in-t he-wild images". ACM Transactions on Graphics (ToG). **40** (4): 1–13.

18. ^Lin J, Yuan Y, Zou Z (2021). "Meingame: Create a game character face from a single portrait". Proceedi ngs of the AAAI conference on artificial intelligence. **35** (1): 311–319.

19. ^Giebenhain S, Kirschstein T, Georgopoulos M, Rünz M, Agapito L, Nießner M. "Learning neural param etric head models." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogn ition. 2023. p. 21003–21012.

20. ^Wang T, Zhang B, Zhang T, Gu S, Bao J, Baltrusaitis T, Shen J, Chen D, Wen F, Chen Q, et al. "Rodin: A g enerative model for sculpting 3d digital avatars using diffusion". In: Proceedings of the IEEE/CVF confer ence on computer vision and pattern recognition. 2023:4563-4573.

21. ^Xu Y, Su Z, Wu Q, Liu Y (2024). "GPHM: Gaussian Parametric Head Model for Monocular Head Avatar Reconstruction". arXiv preprint arXiv:2407.15070. Available from: https://arxiv.org/abs/2407.15070.

22. ^Liu H, Zhu Z, Iwamoto N, Peng Y, Li Z, Zhou Y, Bozkurt E, Zheng B. BEAT: A Large-Scale Semantic and Emotional Multi-modal Dataset for Conversational Gestures Synthesis. In: Avidan S, Brostow G, Cissé M, Farinella GM, Hassner T, editors. Computer Vision -- ECCV 2022. Cham: Springer Nature Switzerlan d; 2022. p. 612-630. ISBN: 978-3-031-20071-7.

23. a, bLiu H, Zhu Z, Becherini G, Peng Y, Su M, Zhou Y, Zhe X, Iwamoto N, Zheng B, Black MJ. "EMAGE: Tow ards Unified Holistic Co-Speech Gesture Generation via Expressive Masked Audio Gesture Modeling." In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. p. 1144-1154.

24. a, b *Liu Y, Cao Q, Wen Y, Jiang H, Ding C. "Towards Variable and Coordinated Holistic Co-Speech Motion Generation." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. p. 1566-1576.*
25. ^ *Taylor SL, Mahler M, Theobald BJ, Matthews I (2012). "Dynamic units of visual speech". In: Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation. pp. 275–284.*
26. ^ *Edwards P, Landreth C, Fiume E, Singh K (2016). "Jali: an animator-centric viseme model for expressive lip synchronization". ACM Transactions on Graphics (TOG). 35 (4): 1–11.*
27. ^ *Bao L, Zhang H, Qian Y, Xue T, Chen C, Zhe X, Kang D (2023). "Learning audio-driven viseme dynamics for 3d face animation". arXiv preprint arXiv:2301.06059. arXiv:2301.06059.*
28. ^ *Karras T, Aila T, Laine S, Herva A, Lehtinen J (2017). "Audio-driven facial animation by joint end-to-end learning of pose and emotion". ACM Transactions on Graphics (ToG). 36 (4): 1–12.*
29. ^ *Cudeiro D, Bolkart T, Laidlaw C, Ranjan A, Black MJ (2019). "Capture, learning, and synthesis of 3D speaking styles". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10101–10111.*
30. ^ *Fan Y, Lin Z, Saito J, Wang W, Komura T (2022). "Faceformer: Speech-driven 3d facial animation with transformers". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18770–18780.*
31. ^ *Xing J, Xia M, Zhang Y, Cun X, Wang J, Wong TT (2023). "Codetalker: Speech-driven 3d facial animation with discrete motion prior". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 12780–12790.*
32. ^ *Richard A, Zollh\uoof6fer M, Wen Y, De la Torre F, Sheikh Y (2021). "Meshtalk: 3d face animation from speech using cross-modality disentanglement". In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 1173--1182.*
33. a, b *Peng Z, Luo Y, Shi Y, Xu H, Zhu X, Liu H, He J, Fan Z (2023). "Selftalk: A self-supervised commutative training diagram to comprehend 3d talking faces". In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 5292–5301.*
34. ^ *Liang L, Lin L, Jin L, Xie D, Li M (2018). "SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction". ICPR.*

35. ^Epic Games, Inc (2022). "RIG LOGIC, RUNTIME EVALUATION OF METAHUMAN FACE RIGS". Available from: https://cdn2-unrealengine-1251447533.file.myqcloud.com/rig-logic-whitepaper-v2-5c9f23f7e210.pdf.
36. ^Forney GD (1973). "The viterbi algorithm". Proceedings of the IEEE. **61** (3): 268–278. doi:10.1109/PROC.1973.9030.
37. a, bEpic Games. MetaHuman Creator [Internet]. 2021. Available from: https://www.unrealengine.com/en-US/metahuman Accessed: 2024-06-01.
38. ^Baevski A, Zhou Y, Mohamed A, Auli M (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations". Advances in neural information processing systems. 33: 12449–12460.
39. a, bZhang L, Qiu Q, Lin H, Zhang Q, Shi C, Yang W, Shi Y, Yang S, Xu L, Yu J (2023). "Dreamface: Progressive generation of animatable 3d faces under text guidance". arXiv preprint arXiv:2304.03117.
40. a, bWang T, Zhang B, Zhang T, Gu S, Bao J, Baltrusaitis T, Shen J, Chen D, Wen F, Chen Q, Guo B (2022). "Rodin: A Generative Model for Sculpting 3D Digital Avatars Using Diffusion". arXiv. arXiv:2212.06135 [cs.CV].
41. a, bLin J, Yuan Y, Zou Z (2022). "MeInGame: Create a Game Character Face from a Single Portrait". Proceedings of the AAAI Conference on Artificial Intelligence. page 311–319. doi:10.1609/aaai.v35i1.16106.
42. a, bDib A, Hafemann L, Got E, Anderson T, Fadaeinejad A, Cruz RM, Carbonneau MA (2023). "[FFHQ-UV-Intrinsics] MoSAR: Monocular Semi-Supervised Model for Avatar Reconstruction using Differentiable Shading".

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.