RenderMe-360: Large Digital Asset Library and Benchmark Towards High-fidelity Head Avatars

Supplementary Material

Abstract

In this supplementary material, we provide more information about the proposed RenderMe-360 dataset and additional experimental discussions for comprehensive benchmarking. Specifically, (1) we unfold the related works that are not mentioned in the main paper (Section 1). (2) we introduce the dataset construction process in detail (Section 2 and 3). Section 2 includes hardware construction and data collection. Section 3 covers data annotation, and data statistics of the proposed RenderMe-360 dataset. (3) We provide additional experiments/detailed setting descriptions for the novel view synthesis benchmark. We also present more comprehensive benchmarks with respect to the rest four tasks that are not unfolded in the main paper(*i.e.,* novel expression synthesis, hair rendering, hair editing, and talking head generation) (Section 4). We analyze the phenomena both qualitatively and quantitatively. (4) We discuss some potential applications that can be benefited from our dataset, and list a toy example in the text-to-3D generation scenario, to show how to utilize our dataset in a flexible way (Section 5). (5) Checklist is attached at the end of this document.

1 Related Works

In the main paper, we discuss related work on multi-view head datasets and head rendering aspects. In this supplementary material, we further unfold the progress on algorithms with respect to the domains of head avatar representation, hair reconstruction, hair editing, and talking head generation.

1.1 Neural Rendering for Head Avatar

Representations. How to effectively represent and ren- der 3D scenes has been a long-term exploration of com- puter vision. The research efforts can be roughly classified into four categories at high-level: surface rendering, image-based rendering, volume rendering, and neural rendering. For surface rendering, the general idea is to first explicitly model the geometry, and then apply shading. For the geometry representation, polygonal meshes [3] are the most popular geometry representations for their compact and efficient nature with modern graphic engines. Other alternatives like point clouds [54], parametric surfaces [51], volumetric occupancy [32, 66], and constructive solid geometry [16] are less convenient. Implicit functions (*e.g.,* signed distance field (SDF)) have better flexibility in complex geometry modeling. Upon these representations, researchers have proposed various shading models to render images [83, 50, 103, 41, 66]. Whereas, all of these representations are better suited to surface reconstruction, rather than photo-realistic rendering, due to their inherent shortages in expressiveness. Traditional image-based rendering (IBR) methods [28, 63, 44] are texture-driven

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

counterparts. They focus on rendering images by using representations like multi-plane images (MPI) [47, 78, 105] or sweep plane [86, 13]. The core idea behind these representations is to leverage depth images and layers to obtain the discrete representations of light fields. Whereas, the view ranges are typically subjected to narrow view interpolations. Volume rendering [49, 40, 41] has great ability in modeling inhomogeneous media such as clouds, and allows rendering in full viewpoints when images are dense. The core idea behind volume rendering is accumulating the information along the ray with numerically approximated of integral. With the emergence of coordinate-based neural networks, neural rendering pops up and becomes a powerful complementarity of classic representations. Such a methodology combines the advantages of differential rendering and neural networks. For instance, neural surface rendering [97, 29, 83], and neural volume rendering [49, 40] ensure novel views of the target scene can be rendered by arbitrary camera pose trained by dense multi-view images. These methods achieve photo-realistic rendering and smooth view transition results in creating free-viewpoint videos compared to traditional ones. The follow-up researches lie on the directions of model efficiency [50, 67, 99], dynamic scene [55, 76, 18], large-scene compatibility [79, 91], class-specific robustness [22, 82], multi-modal extensiveness [19, 81], or generalizablitiy [100, 84, 5, 46, 7, 36].

Hair Reconstruction. High-fidelity hair reconstruction has been a long-standing challenging task due to the tremendous volume of strands, great diversity among different identities, and micro-scale structure. Dynamic hair rendering and animation are even more difficult since complex motion patterns and self-occlusions need to be additionally considered. Except for classical methods like hair modeling paradigms [27, 43, 93], multi-stereo methods [62] or physics-based simulations [31, 26, 10], some later research efforts utilize deep neural networks to extract temporal features of hair motion [95], infer 3D geometry [30], or localize valid mask region [68]. With the blooming of neural rendering, recent works make notable progress in both static and dynamic hair reconstruction. For example, to render high-fidelity hair strands, NeuralStrand [60] introduces a neural rendering framework for jointly modeling hair geometry and appearance. For dynamic hair modeling, general dynamic scene rendering methods such as [35, 77, 40, 41, 87] could be directly applied to the task. These methods have been proven as powerful tools to model the motion and interaction of hair strands. Upon the [41], HVH [87] designs a special volumetric representation for hair, and models the dynamic hair strands as the motion of the volumetric primitives.

1.2 Generative Models for Head Manipulation

Hair Editing. Finding a neat solution to support hairstyle or hair color editing is an exciting research problem. Related methods could be categorized into image-based editing and text-based ones. The general ideas behind the two trends follow a similar pipeline – (1) first, encode hair appearance, shape, and structure information from prompts. For image-based methods [33, 92, 70], the prompts could be masks, well-drawn sketches, or reference images. For text-driven ones, the core prompt is text descriptions. (2) The second step is style mapping, where input conditions are mapped into corresponding latent code changes. Image-based methods utilize sophisticated conditional generative module [71, 92] or modulate conditions into the prior space of a pre-trained generative model [58] via inversion strategies (*e.g.,* e2e[75], PTI [59], ReStyle [1], and HyperStyle [2]). As a flexible complementarity, text-driven methods graft the power of CLIP [56] to guide/regularize target attribute manipulation. StyleCLIP [52] is a general text-driven image manipulation framework and can be directly applied to hair editing. It provides a basic solution to tailor text information into latent optimization and mapper. Upon this, HairCLIP [74] designed specific latent mappers for hairstyle and color editing based on both reference images and text prompts.

Talking Head Generation. This task also known as face reenactment, aims to synthesize realistic human face videos according to the given source facial clips and the driving materials. It can be roughly divided into two categories by the driving modality: image-driven methods [73, 89, 65, 101, 4, 85, 21] and audio-driven methods [69, 72, 24, 80, 6, 104, 106, 25, 19, 38]. The major challenge for this task is to control the expressions and head pose of the synthesized video according to the driving materials, while reserving the identity information of the source images. several methods

2

used facial landmarks [65, 101], latent feature space [24, 104] or the parameters of parametric head model [73, 72] to model the facial expressions, and then use these intermediate representations to guide the animated video generation. More recently, AD-NeRF [19] and SSP-NeRF [38] condition the radiance field with audio fragments for the customized talking head generation. AD-NeRF trains two neural radiance fields for inconsistent movements between the head and torso without an explicit 3D face model. SSP-NeRF [38] uses one unified neural radiance field for portrait generation with the introduced torso deformation module and semantic-aware ray sampling strategy.

2 Data Collection Details

In this section, we first introduce our physical capturing environment, POrtrait Large-scale hIgh-quality Capturing sYstem, namely POLICY (Section 2.1). Then, we provide an elaborate data collection pipeline introduction(Section 2.2).

2.1 Capture System: POLICY
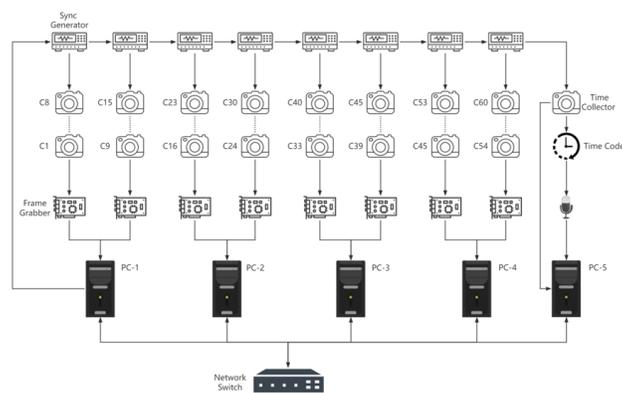
Figure S1: **The structures of the POLICY.** 60 industrial high-definition cameras and a high-quality recording device are connected through synchronous generators, frame grabbers, five high-performance computers, and a network switch.

Hardware Setup. We build a multi-video camera capture cylinder called POLICY to capture synchronized multi-view videos of the human head performance. The capture studio contains 60 synchronous cameras with a resolution of 2448×2048 (a multi-view data sample is visualized in Figure S2). The sensor model is LBAS-U350-35C, and the shutter speed is at 30 FPS for video capture. The cameras are arrayed in a cylindrical confined space, and they all point inward to the middle of the cylinder. We separate the camera array into four hierarchical layers. The first and the fourth layers use a large field of view to capture the overall head motion at a long distance, while the second and the third layers adopt a small field of view to capture more details of the head. 39 LED displays are used in the cylinder, where 6 are used to balance the lighting distribution in front of the human face.

In addition, POLICY also contains five computers with high-performance CPUs and RAIDs, a network switch, eight frame grabbers, an extra camera, a time-code viewer, a condenser microphone, and fiber optic USB capture cables. The fiber optic USB capture cables are used to link the other devices.

Hardware Synchronization. It is a great challenge to achieve high-bandwidth capturing and synchronization in both visual portrait data collection from 60 color cameras with different views, and audio-vision data collection from recording devices. We illustrate the structure design of POLICY in Figure S1, and show the reason why our POLICY can overcome the challenge in following paragraphs.

3

Figure S2: **Multi-view head data sample.** The captured human head visual data encompass 60 camera views with $360°$ left-to-right, and $160°$ up-to-down.

For visual data, POLICY connects every eight cameras to a frame grabber and a synchronization generator. Two frame grabbers are connected to a computer on the other end to achieve high-bandwidth transmission of the capturing data. A synchronization generator is connected in series to the next synchronization generator on the other end, and the first synchronization generator is linked to the first computer. During capturing visual data, the first computer controls all synchronization generators by launching a high-level trigger to achieve a microsecond error in the cameras' synchronization.

For audio data, POLICY uses the extra camera to connect to a synchronization generator and the time-code viewer. A high-quality microphone is placed in front of the human head. The time-code viewer is linked to the microphone for the collection of the time stamp of the audio voice. The microphone and the extra camera are connected to a computer. During capturing audio data, the time code of the microphone and the synchronized signal from the extra camera enable the high-precise synchronization of audio-vision data.

All computers are connected to the network switch to synchronize the capturing operations and store the capturing data at high bandwidth. With the connection of these devices, POLICY achieves high-bandwidth capturing with the speed of 90 GB/s, multi-view synchronization, and audio-vision synchronization at the speed of 30 Hz.

2.2 Data Collection Details

2.2.1 Criterion for Captured Attribute Design

We invite 500 people to be our capture subjects. We require each subject to perform three different parts during the data capture, namely expression, hair, and speech. We will detail the collection process in Section 2.2.2. In the current sub-section, we will describe the content design.

Expression. The design of expression collection is based on the standard proposed in i3DMM [98], in which 10 facial expressions are recorded as the train set and the other 5 are used as the test set. We capture 1 neutral expression and 11 facial expression (9 for the train set and 2 for the test set, if not specifically explained). It needs to be stressed that two of our design expressions (smile and mouth-open) are treated as the test expression, with the motivation that the smile and mouth-open are used to test extrapolation and interpolation of the benchmarks respectively. The expression capture example is visualized in Figure S3.

Hair. The design of the hair collection consists of three aspects – original outfit capture, 3D face capture (with hair cap to hide hair), and wig capture. Specifically, for the original outfit capture setting, each subject is captured with his/her original hairstyle. For performers dressed in different eras, the collection of 3D face capture and wig are skipped due to the inconvenience of wearing a wig or hair cap on the head with already wearing many different accessories. For the normal performers, one video of wearing the hair cap is captured and then the wig part follows. We prepare wigs with 7 daily styles ('Men's straight short hair', 'Men's curly short hair', 'Women's bobo hair', 'Women's pear curls', 'Women's long curls', 'Women's long straight hair', and 'Women's small curls'), and

4
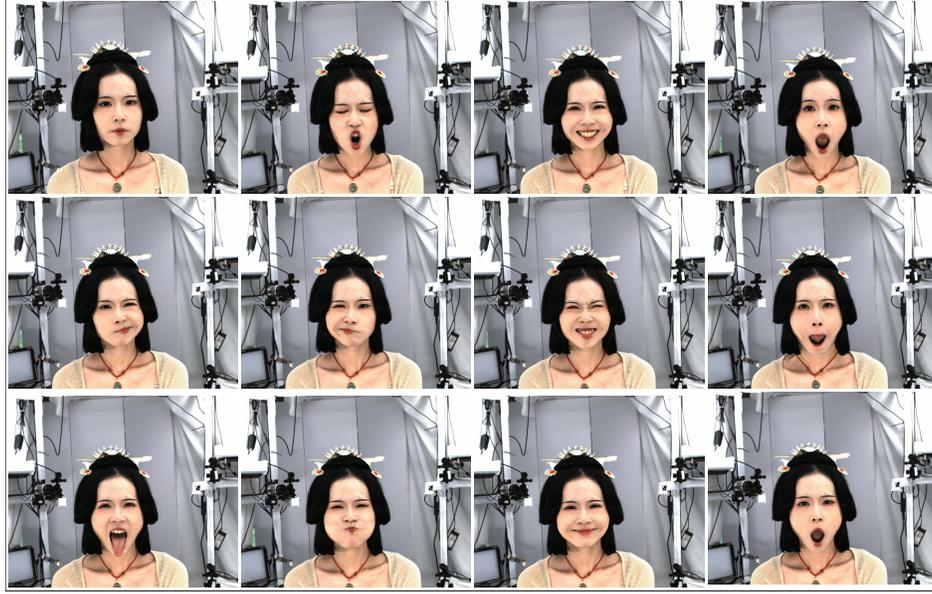
Figure S3: **Expression capture.** We capture 12 expressions, containing 1 expressionless and 11 exaggerated expressions.

6 color tones (black, blue, brown, green, gold and yellow). During the collection, the subject is asked to turn around his head in a whole circle. Such a design can benefit the emphasizing of the dynamic motion that relates to the wig. Different wig styles, colors, and head motions are visualized in Figure S4.

Speech. Since the subjects come from different countries all over the world, we provide the speech corpus in two languages – Mandarin for Chinese and English for the others. We also provide two versions of the corpus.

Concretely, in the first version, each subject speaks 42 sentences, which consist of sentences and short paragraphs. For Mandarin sentence design, we select 30 phonetically balanced sentences from [64] as our main part, and 10 sentences combined with single words from [14] in order to cover all the consonants, vowels, and tones. The composition of English sentences is similar to VOCASET [9], in which the main part covers 40 phonetically balanced sentences. Two short paragraphs are both added to the Mandarin and English collections as a supplement for continuous long-time talking recordings. Each subject has the same corpus in the first version. In the second version, we shorten the total number of sentences from 42 to 26 in order to speed up the collection. Moreover, we randomly sample the sentences from the corpus for each subject so as to improve differentiation. For Mandarin, we first reduce the single words-combined sentences from 10 to 5 but still keep their coverage of consonants, vowels, and tones. Then, for the main part, phonetically balanced sentences are shortened to 20, which consists of 10 fixed and 10 flexible sentences. Finally, we randomly sample one paragraph from the original two. As a result, we get 26 sentences in total for each subject. For the English collection, the main part, with respect to the original 40 phonetically balanced sentences, is shortened to covering 25 sentences (15 fixed and 10 flexible sentences). The paragraph part is processed the same as in Chinese. Since we have 500 identities in total, about 150 Chinese and 150 non-native Mandarin speaking subjects are captured with the first version and the rest with the second version.

2.2.2 Collection Protocol

As the dataset collection spans over months, to guarantee the accuracy of data collection, we design a collection protocol and execute it before every capture. The protocol consists of three steps, *i.e.,* pre-collection check, collection, and post-collection check.

5

Figure S4: **Daset sample of hair capture.** We capture 12 hairstyles for each subject (on average). The data includes one video of the original hairstyle, one video of wearing headgear, and ten video sequences of wearing wigs. The ten wigs are randomly picked from our wig set. We ask the participant to turn the head clockwise with different hairstyles.

Pre-Collection Check. To ensure proper operability of equipment and accurate camera position, two steps of inspection are applied:

1) *Hardware Check.* We manually check the status of all computers and cameras and make sure that all 60 video stream is ready-to-work and synchronized by testing collection. We prepare backup cameras for the broken ones.

2) *Fake Head Capture.* We put a fake head in the middle of the view and keep it static, and then capture one frame of all 60 cameras. Then we check all the frames, when the head offsets the imaging center, the pose of the correspondent camera needs to be fixed. The sharpness of the images is also checked in case one or part of the cameras are not focusing on the head.

Collection. The main collection consists of four parts:

1) *Camera Calibration.* A chessboard is held and turned around for 3 circles, then every camera can capture data with the chessboard in various poses. The data is used for calculating the camera parameters (intrinsic and extrinsic).

2) *Expression Capture.* Each subject's expression metadata is collected with 12 facial expressions. Each expression collection lasts about 3 to 5 seconds and the performer starts with the neutral expression, changes continuously to designated expressions, and then keeps the performance unchanged until this collection finish. Substandard or incorrect expressions will be discarded and re-recorded.

3) *Hair Capture.* The hair collection is separated into three parts: origin hair, hair cap, and wig capture. One video for the origin hair and one for the hair cap are captured for each subject. In these two parts, the subject always keeps still with eyes straight ahead. Then the wig part collection begins and we collect 10 videos for wigs with random hairstyles and colors. Generally each subject cover about 4 wig styles and 3 wig colors. In the wig collection, the performer starts with his head in the middle of the view and eyes straight ahead, then cranes his neck 360 degrees, relaxing it as usual but with as much amplitude as possible. When finishing the whole process, the subject returns to the original status and waits for the end of this part. We'll record it again when insufficient head rotation appears.

4) *Speech Capture.* We prepare a large corpus in two languages (Mandarin and English) for each subject. The whole speech collection is split into 4 or 6 parts according to the number of sentences.

6
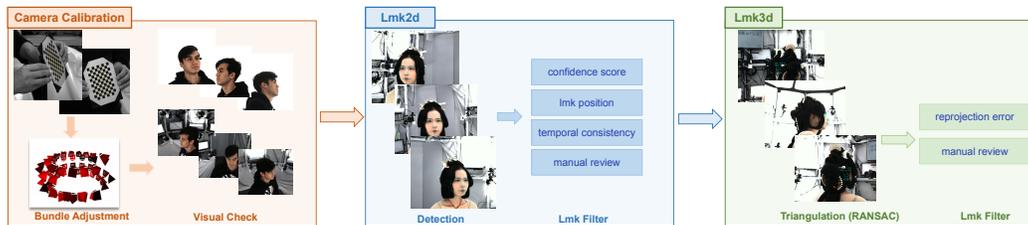
Figure S5: **Camera calibration and keypoint detection.** The camera calibration process contains chessboard data collection, calibration with bundle adjustment, and visual check. After the detection and filtering of the multi-view 2D landmarks, the 2D landmarks result, together with the camera parameters, are utilized to triangulate for robust 3D landmarks.

In each collection, the performer is asked to read the sentences which are shown on a screen and the collection lasts about 30 to 40 seconds. We do not require a standard mouthpiece but mispronunciation is not allowed.

Post-Collection Check. A script is applied to concatenate and visualize the multiview video synchronously. All the collected data is processed and checked manually to filter out source data issues. Due to the hardware limitation, the recording data of a few subjects miss one or two camera views. We demonstrate the necessity and importance of the data post-collection check with extensive trial and error experiences.

After the above processes finish, we obtain a large-scale dataset of 500 identities. Each identity is guided to perform 12 expressions, talking with 26 or 42 sentences, and more than 10 hairstyle collections.

3 Data Annotation Details

We obtain the raw data of RenderMe-360 from the collection pipeline. Then, we annotate the data to get rich annotations. In this section, we present the detailed annotation processes regarding each annotated dimension (Section 3.1- 3.5). We also analyze the data statistics of the proposed dataset in detail (Section 3.6).

3.1 Camera Parameter Annotation

Camera calibration is the basic step for fine-grained annotation in a multi-view capture system. The process in our pipeline is visualized in Figure S5. To make sure the availability and accuracy of the parameters, two checking procedures are performed besides basic camera pose estimation. First, we apply fast NeRF model training of Instant-NGP [50] via feeding all the camera views. We render images with the same views and manually check for potentially unreasonable rendering results caused by wrong extrinsic parameters. Secondly, we perform the keypoint annotation process with the same frames and re-project the 3D facial landmarks to manually check for the out-of-face result. The unqualified results will loop in re-calibration process.

3.2 Facial Keypoint Anotation

To filter out abnormal 2D landmarks and precisely triangulate to get robust landmark 3D, we apply the following rule-based and heuristic rules. 1) We use a enhanced version of facial landmark detection model [88], and discard the result with a low confidence score. 2) Since some unqualified landmark results have an abnormal scale or location, we heuristically set thresholds for the largest distance between landmarks and the mean location. 3) As there is no large head motion in the expression capture stage, we consider the temporal consistency of the detected landmarks and filter out the case with an overall offset of the keypoints. 4) We manually check the data to select inaccurate landmark results. We make sure that data of at least 3 views are applied to do the triangulation, and check

7
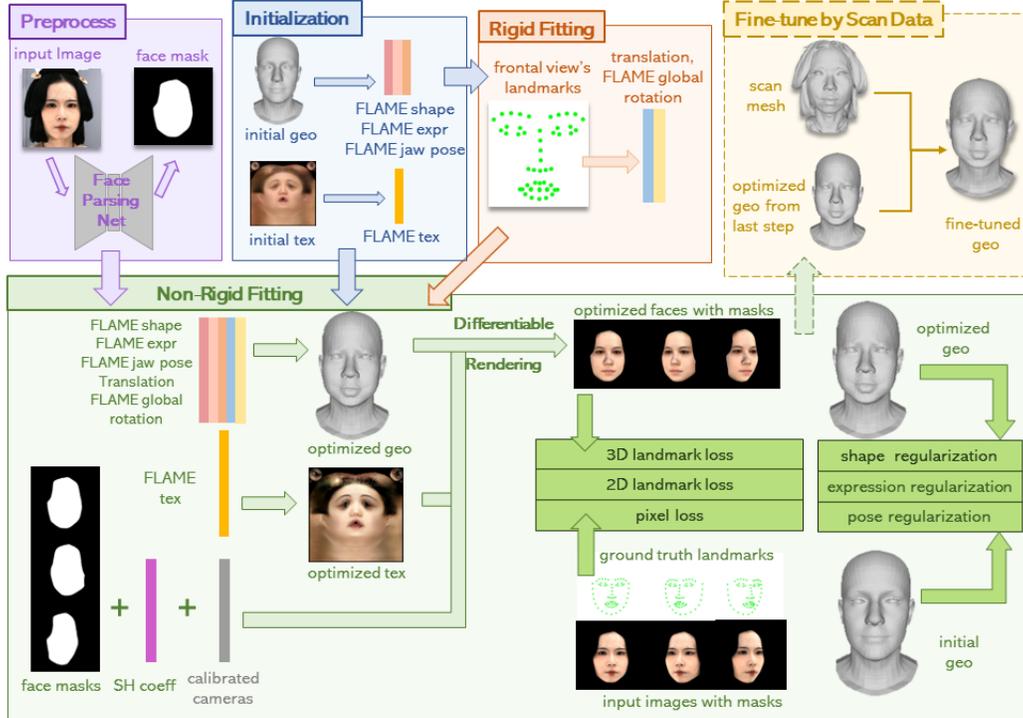
Figure S6: **FLAME fitting.** The fitting pipeline is focused on the subject's face region, and face masks for each view are preprocessed. Rigid fitting aims to solve translation and rotation roughly with 2D and 3D landmarks, values are improved in the non-rigid fitting. Non-rigid fitting optimizes FLAME's other parameters as well, but mainly on shape, expression, jaw pose and texture parameters to ensure better identity likeness of final geometry. The last fine-tuned step is not necessary for all frames, frames without scan mesh are optimized based on frames with it.

the reprojection error in all 60 views. When a significant location error or an abnormal reprojected location is detected, we manually label all 2D landmarks and re-run the triangulation process for an accurate result.

3.3 FLAME Fitting

The overall pipeline for FLAME fitting is illustrated in Figure S6. Raw captured images are first processed via masking out the background and non-facial head regions, in order to avoid fitting distractions. Then, a rigid fitting is applied to get rough values of translation and global rotation. Concretely, the 2D and 3D facial landmarks are both involved in this process. We use 51 facial landmarks due to the non-differentiable attribute of contour landmarks trajectory. 2D landmarks from the frontal views are used for rough estimation, and 3D landmarks are used for anchor 3D position. For the rigid fitting, the optimizing target can be viewed as

$$\mathcal{L}_{\text{rigid}} = \|\text{lmk}_{2d} - Proj(R \cdot \text{lmk}_{\text{flame}} + t)\| \tag{S1}$$

where lmk_{2d} is the detected 2D landmarks, $\text{lmk}_{\text{flame}}$ is the marked corresponding landmarks on the FLAME model, and R,t are the variables to be optimized, the loss is calculated through all frontal views and all 51 facial landmarks.

Non-rigid fitting is further applied to improve translation/global rotation, FLAME shape, expression, jaw pose, and texture parameters. We utilize landmarks in both 2D and 3D to constrain the optimization. Since 3D landmarks provide one more dimension value (*i.e.,* z value), while having a shortage of good face contour information. Thus, 2D landmarks around face contours are needed to improve shape. Moreover, with calibrated cameras, we are able to render geometry and texture in

8
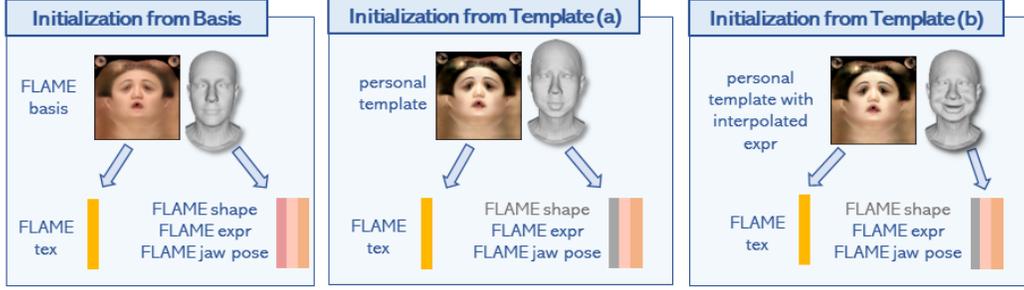
Figure S7: **Initialization modes for FLAME fitting.** There are three initialization modes according to different fitting purposes. Initialization from the basis is designed for getting a personal template. Initialization with the template is to fix shape parameters and do expression fitting, (a) is for frames with scan mesh, (b) is for frames without.

260 image space by using differentiable rendering and comparing pixel differences with input images.
261 However, texture parameters only map to albedo map based on texture basis, and skin tone from input
262 images is affected by environment lighting conditions. Thus, optimized spherical harmonics (SH)
263 coefficients are needed to adjust rendered faces. To ensure the reasonability of optimized geometry,
264 we provide shape, expression, and pose regularizations to avoid broken geometry. Scan meshes show
265 accurate facial shapes in world space, so a FLAME fitting process with scan can preserve better facial
266 edges and corners, but not all frames are grouped with it. As shown in Figure S6, the fine-tuned step
267 is surrounded with dotted lines, indicating that it is not necessary for all frames and is only applied
268 on frames with scan meshes to do further improvement. This strategy is useful for personalization
269 and getting expression prior knowledge for non-neutral frames without scan. In a nutshell, the full
270 loss function can be formulated as

$$\mathcal{L} = \mathcal{L}_{\text{lmk}} + \mathcal{L}_{\text{scan}} + \mathcal{L}_{\text{pix}} + \mathcal{L}_{\text{reg}} \tag{S2}$$

271

$$\mathcal{L}_{\text{lmk}} = \|\text{lmk}_{2d} - Proj(R \cdot \text{lmk}_{\text{FLAME}(s,e,p)} + t)\| \\ + \|\text{lmk}_{3d} - R \cdot \text{lmk}_{\text{FLAME}(s,e,p)} - t\| \tag{S3}$$

$$\mathcal{L}_{\text{scan}} = \min_{i \in \text{scan}} \|v_i - R \cdot v_{\text{FLAME}(s,e,p)} - t\| \tag{S4}$$

$$\mathcal{L}_{\text{pix}} = \|\text{rgb}_{Proj(R \cdot v_{\text{FLAME}(s,e,p)})} \\ - \text{tex} * (\gamma \cdot \text{SH}(n_{\text{FLAME}(s,e,p)}))\| \tag{S5}$$

$$\mathcal{L}_{\text{reg}} = \left\|\frac{s}{\sigma_s}\right\| + \left\|\frac{e}{\sigma_e}\right\| + \left\|\frac{p}{\sigma_p}\right\| \tag{S6}$$

272 where landmark loss includes 2D detected ones and 3D triangulated ones. Scan loss includes the
273 nearest point on scan with each FLAME vertex, which is only calculated at the last frame of each
274 sequence. For rendering, we calculate the RGB value at each float position with bilinear interpolation
275 within the face mask with rendered vertices using face normals n_{FLAME} and spherical harmonic
276 lighting SH. Regularization terms include shape parameter s, expression parameter e, and poses p
277 for jaw, neck and eyes.

278 We assume frames of neutral sequences are always neutral (expression parameter s and pose parameter
279 p are zero), sequences with non-neutral expressions start with neutral and end with exaggerated
280 expressions. Dense mesh reconstruction is at least applied on the last frame to generate scan mesh
281 for each expression sequence. The personalization step is inspired by [34]. It starts with FLAME
282 basis as an initial value, as shown in the left image of Figure S7, optimizes FLAME parameters, and
283 is fine-tuned with the help of scan mesh to get an accurate face shape template. With a personal
284 template provided, as shown in the middle image of Figure S7, non-neutral frames' fitting won't
285 optimize shape parameters anymore, and we solve the last frame paired with scan mesh firstly and
286 puts more effort into other parameters to ensure face expression as vivid as the input image. Due to
287 the assumption mentioned above, frames in between the first frame and the last frame are performed
288 with linear interpolation to get a rough initial value, as shown in the right image of Figure S7. For the

9

purpose of ensuring the annotation to the full extent of accuracy, the human annotators are asked to identify and rectify inaccurate annotation results of FLAME. The annotators must identify and select the incorrect results, and then we provide the necessary refinement to generate the accurate 3D head model.

In addition to FLAME fitting annotation, we also provide the UV texture map as an extra annotation upon the fitting. Specifically, since it is low quality and has few details, instead of using an albedo map optimized from our fitting pipeline, we take view-dependent texture maps unwrapped from captured images of selected views and composited them together with Poisson blending [53] to create the final high-quality texture map in Figure S8.
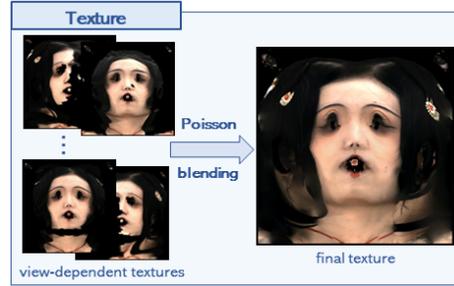
Figure S8: **Final UV texture map.** View-dependent texture maps are selected and composited together with Poisson blending to create the final full texture map as the UV map annotation.

3.4 Scan and Matting Refinement

The processing pipeline is illustrated in Figure S9.

Scan. Specifically, we apply NeuS [83] to multi-viewed images with known camera intrinsics and extrinsics. In practice, a rigid transformation is estimated from landmarks of a standard FLAME model to target detected 3D landmarks from triangulation. Then the bounding box of the head region is assumed to be 2 times the bounding box of the FLAME model. We follow the setting assuming that a background NeRF [48] modeling the rendered results outside the bounding box and a NeuS [83] modeling radiance field inside the bounding box. Both are modeled as an 8-layer multi-perceptual network (MLP) with skip connections in the 5-th layer, and the inputs are coded with positional encoding. For each video sequence, we apply this algorithm to the first frame and train from scratch to get the neutral scan mesh. For the following frames, we pick the keyframe where the expression seems to be the most exaggerated, add fine-tune to the static model to get a similar scanned result, where the bounding box is fixed as the first frame.

Matting. As for the matting annotation, a static background is captured before the formal recording of each round. Then, we use a video-based matting method [37] to estimate the foreground map of each image. To further improve matting accuracy, we additionally tailor the depth information into the pipeline. Concretely, we rasterize the scanned mesh to each camera view, and use this geometry prior to refine the video-based matting estimation, with graphical-based segmentation. Grabcut [61] is used with the intersection of both masks as the absolute foreground and areas outside the union with a fixed size of padding as the absolute background. We calculate Bayesian posterior for each pixel as the alpha value. We further employ human annotators to identify and rectify inaccurate annotation results of scan and matting. Then we provide the necessary parameters to generate the accurate dense mesh or manually label the foreground to yield precise matting maps.

Matting Annotation Discussion. We verified the accuracy of the annotation of matting by comparing our synthesized results with the manual-annotated matting results. In particular, the annotators are required to manually segment the foreground and the background among 800 images that are randomly selected from our

Table S1: **Quantative results of matting annotation.** We calculate the difference between the synthesized matting maps and the manual matting maps on a subset of our data.

	MSE	IoU	AUC
Synthesized Matting Annotation	0.009	0.971	0.990

dataset. We adopt several well-known metrics, including Area Under the Curve (AUC), Mean Square Error (MSE) and Intersection over Union (IoU), to measure the distances between our matting annotation and the hand-made matting annotation. As shown in Table. S1, the difference between the synthesized matting maps and the manually annotated matting maps is slight. It demonstrates that the

10
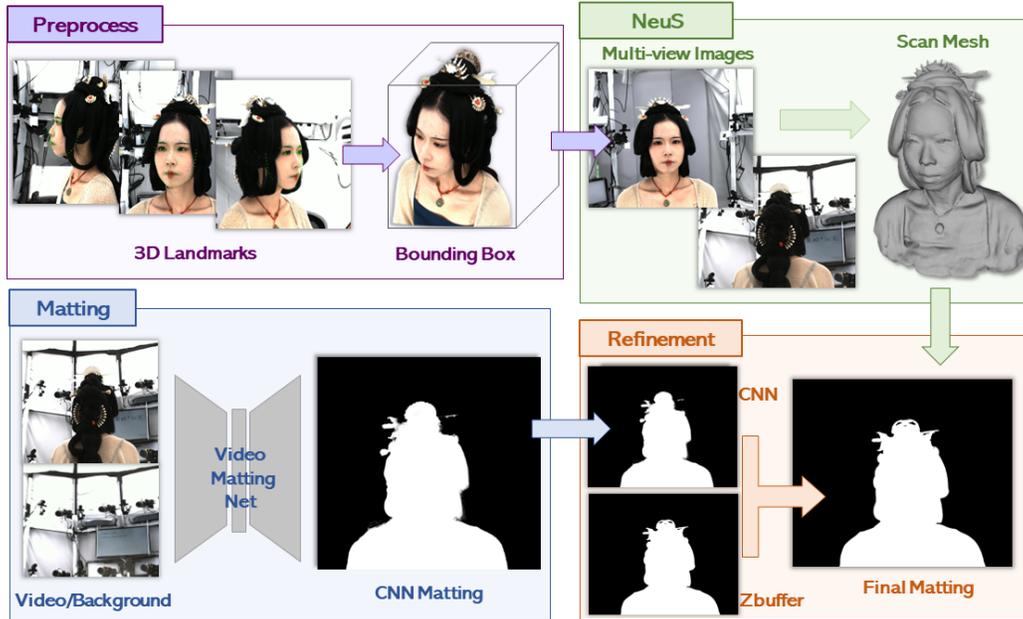
Figure S9: **Dense mesh reconstruction and matting.** Dense mesh reconstruction is supported by NeuS, it builds models for the subject(foreground) and background separately, bounding box is estimated by robust 3D landmarks for better separation. The final matting result is refined with a Z-buffer value. This is applied for refining the situation when the mask predicted from the video matting network cannot well handle detailed head accessories.
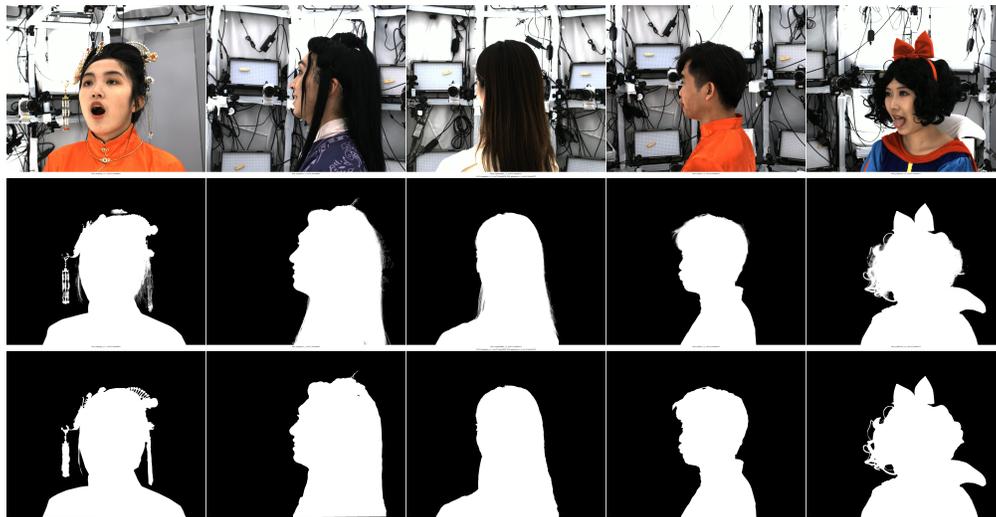
Figure S10: **Qualitative comparison of matting annotation.** We illustrate the qualitative comparison between the synthesized matting maps and the hand-made matting maps. From the top to the bottom rows are: the original images, our synthesized matting maps and the hand-made matting maps.

annotations synthesized by our algorithm are comparable to human annotations, with high reliability and usability.

We illustrate some examples for qualitative comparison in Figure S10. As shown, our synthesized matting maps are similar to the hand-made ones and can precisely segment the human head and the background of the original image.

11
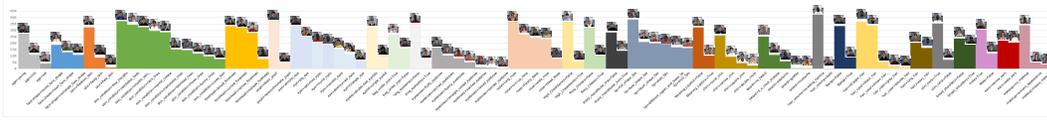
Figure S11: **Statistical chart of static facial features.** The properties that lie in the same attribute group of facial features are highlighted in the same color. An exemplar image of each attribute is shown in the corresponding histogram column. We use ">" to distinguish the group and attribute. Better zoom in for details.

3.5 Text Annotation

Both static and dynamic text-based descriptions are involved in our text annotation to further facilitate multi-modality research on human head avatar creation. The text combines four types of annotations: *static facial features*, *static information of non-facial regions*, *dynamic facial actions*, and *dynamic video activity descriptions*. With these four aspects of text annotation, we could provide a comprehensive description of each human head to boost various downstream tasks.

Static Facial Features. This aspect of text annotation seeks to comprehensively detail attributes of the subject's facial features in facial regions. Based on the text annotation proposed in CelebA [39], we further annotate new facial features, with extending the original 40 annotations to 95 common fixed types of facial attributes and 2 non-fixed text-based salient attributes. The fixed facial attributes refer to the universal and frequent properties, which are annotated through pre-defined attribute item selection. The non-fixed attribute provides flexible supplemental additions to the 95 fixed attributes, which aim at encompassing a broader range of facial depictions and is annotated through natural language. The combination of fixed attribute and non-fixed attribute annotations could outline human faces with more complete and precise text descriptions than the original category definition in CelebA.

Specifically, the fixed facial attributes and the corresponding example images are illustrated in Figure S11. For every attribute, we employ five annotators to vote on whether the collected subjects contain the particular attribute, and the final annotation is determined by the majority decision. In particular, we carefully analyze common facial traits, and divide these 95 facial attributes into 28 major groups, including facial properties like face shape, skin condition, eye shape, eyebrow shape, lip shape, nose shape, hair shape, etc. Each major group of facial features contains several detailed shape attributes. Compare with the original facial attributes of CelebA [39], we introduce more facial feature attributes to describe facial features in detail. For instance, CelebA only defines one single label for eyes, namely "narrow eyes", we provide more variant shapes for comprehensive depictions, including "almond eyes", "big eyes", "upturned eyes", "round eyes", "monolid eyes", "downturned eyes" and "triangle eye". More examples like the skin condition, a newly introduced property group, is a significantly conspicuous facial attribute and has been ignored by CelebA. For this group, we describe it with several detailed attributes, containing "tear troughs", "nasolabial folds", "neck lines", "mental creases", "marionette lines", "forehead lines", "frown lines", "bunny lines", "crows feet" and "smooth skin". Through such a fine-grained category enrichment, a fixed common types annotation with 95 attributes of facial attributes is constructed.

In addition, we provide two non-fixed attributes: *the salient facial feature*, which describes significant attributes of the facial features, and *the salient features of the makeup*, which depicts the significant features of the makeup styles. The two attributes do not overlap with any of the fixed attributes. We require annotators to observe the overall features of the subject and describe salient features of the subject's face and makeup style in natural language. The annotated descriptions from 5 annotators are collected and manually removed redundant or nonexistent attributes to yield the final annotation. For example, the salient facial attribute of Figure S3 is that *she possesses visible collarbones with a mole above the left eyebrow, round pupil, multiple eyelids, slightly flattened eyebrows, pale forehead, and applies light foundation, draws long and thin eyebrows, wears petal-like lipstick with pink eyeshadow and black mascara.* This flexible attribute further complements salient facial features based on subjective observations, including some color, position and shape of facial features, and some attributes not covered by fixed attributes.

12

Static Information of Non-Facial Regions.
This aspect of text annotation aims to depict the attributes of the subjects' non-facial regions, such as the tops of outfits and accessories. In addition to the attributes of inherent facial features, we also consider static information of non-facial regions. Since these properties are distinctive to describe different human heads. We focus on the material, shape, color, and lighting conditions of the subjects' wearing accessories. For holistic head rendering, information on non-facial regions is also critical. However, few studies have involved annotation of these parts, with most research focusing solely on the labels of static facial features. While static facial features have been a primary focus for modeling human appearance, additional qualities corresponding to the wearing elements promote photorealism. Unprecedentedly, we introduced annotations related to these aspects. By including non-facial attributes in our annotation, we provided a broader, and more integrated knowledge to model human heads in their full individual characteristics.

Similar to static facial features, we provide two types of annotation, including fixed attributes and non-fixed attributes. As shown in Figure S12, the fixed attributes contain 36 attributes derived from 7 major groups, such as accessories shape, clothing transparency, headwear shape, *etc*. For every attribute annotation, we require five annotators to label whether the subject has the attribute. The final annotation of this attribute is voted by the majority choice. Additionally, the annotators are required to describe the non-fixed attributes in natural language. The non-fixed attributes contain 1) *the color of the tops of the outfits*, in which the annotators describe the colors and in the order from large areas to small areas; 2) *the color of the head accessories*, in which the annotators mark the colors included in the order from large areas to small areas; and 3) *the salient features of the accessories*, in which the annotators describe the significant features of the accessories. For instance, the non-fixed attributes in Figure S3

Figure S12: **Statistical chart of static information of non-facial regions.** The properties that lie in the same attribute group of non-facial features are highlighted in the same color. An exemplar image of each attribute is shown in the corresponding histogram column. We use ">" to split the group and attribute.

are that 1) *her tops of the outfits are yellow and black*, 2) *her accessories are multiple colors of golden, white, blue and red*, and 3) *she wears an ancient jade pendant and a golden hairpiece with red stones and a blue circlet in the crown*. There are no overlapped descriptions between non-fixed attributes and fixed attributes. The proposed text annotation on static information of non-facial regions involves diverse and rich descriptions for the non-inherent attributes, which could promote text-aware generation with detailed and high-fidelity textures.

Dynamic Facial Actions The text annotation of dynamic facial actions refers to explicitly describing the dynamic changes in the local facial features of the collected subjects at each timestamp. Here,

13

Table S2: **Action Units of expression.** Each of the collected expressions (Exp) is defined as a set of AUs. Please note that *Exp-4 is a left-toward expression while *Exp-5 is a right-toward expression, and they contain the same set of AUs.

Expression No.	Action Units
Exp-1	AU-18, AU-22, AU-25, AU-27, AU-43
Exp-2	AU-6, AU-12, AU-13, AU-14, AU-25, AU-26, AU-27
Exp-3	AU-1, AU-5, AU-25, AU-26, AU-27
Exp-4*	AU-4, AU-6, AU-9, AU-11, AU-13, AU-14, AU-17, AU-44
Exp-5*	AU-4, AU-6, AU-9, AU-11, AU-13, AU-14, AU-17, AU-44
Exp-6	AU-4, AU-7, AU-9, AU-10, AU-15, AU-25, AU-41
Exp-7	AU-16, AU-25, AU-26, AU-28
Exp-8	AU-13, AU-25, AU-26, AU-27
Exp-9	AU-13, AU-17, AU-18, AU-23
Exp-10	AU-6, AU-12, AU-13
Exp-11	AU-25, AU-26, AU-27

we only focus on the collected expression-related videos and ignore speech-related and wig-related videos because expression-related videos already contain a large number of dynamic changes in local facial features.

Based on Facial Action Coding System, FACs [12], facial expression can be described into specific action units (AUs), which are the fundamental facial actions of individual muscles or groups of muscles. The detailed descriptions of each AU can be found in `https://www.cs.cmu.edu/~face/facs.htm`. Each of the 11 collected expression categories can be further divided into a set of multiple action units (AUs), as shown in Table S2. We provide AU annotations for each frame of changes in expression videos. As shown in Figure S13, we statistically analyzed the proportion of each AU category in the annotations. AU-25, representing that the lips part, appears most frequently, accounting for 12.28%, while AU-1, representing that the inner brow raise, appears least frequently, only 1.74%. The top 3 most prevalent

Figure S13: **Statistical chart of dynamic facial actions.** We illustrate the proportion of every AU in our text annotation data of dynamic facial actions.

AUs are AU-25 (lips parting), AU-13 (cheek puffing) and AU-27 (mouth stretching), while the least prevalent top 3 AUs are AU-1 (inner brow raising), AU-5 (upper lid raising) and AU-7 (lid tightening). It indicates that our dataset encompasses more extensive mouth movement variations, which are significant facial motions while paying comparatively little attention to subtle brow and lid regions motions.

Dynamic Video Activity Descriptions. The text annotation of dynamic video activity descriptions is video-linguistic annotation and aims to globally describe the overall activity of the subjects in the collected videos in complete sentences.

To globally describe facial activity with diversity, four annotators were employed to introduce each video action from four different perspectives: dynamic changes in facial actions, dynamic changes in facial state, dynamic changes in facial features, and dynamic changes in facial muscles. We collected videos in three scenarios: expressions (Exp), hairstyles (HS) and speeches (Sp). Thus, each video has a corresponding template, and the annotators describe each video type from the collection templates, allowing us to obtain text descriptions for each video type. The descriptions of the actions performed by the subject can be found in Figure S14. Each type of action has four corresponding descriptions. In particular, for hairstyle videos, we describe wig color, shape, texture, etc., which

14

does not overlap with our previous annotations, since the previous annotations did not involve wigs. For every individual video, providing merely a subject (i.e., "a man" or "a female") and integrating this with the relevant template of dynamic action descriptions yields a complete descriptive sentence.

As shown in Figure S14, we provide comprehensive and diverse video activity descriptions composed of user-friendly natural language sentences, which can facilitate video generation or video editing.

Video Type	Text-1: Facial Actions	Text-2: Facial State	Text-3: Facial Features	Text-4: Facial Muscles
Exp-0	One has no expression.	One's face is expressionless.	One's face is motionless.	One's face muscles do not move.
Exp-1	One closes her eyes, her mouth protrudes O-shaped.	One's closing eyes. She is opening wide and protruding the mouth.	One's jaw drops, lips part up and down and mouth stectch in O-shape.	One's cheeks' muscle is being pulled down and her lips are pushing outside.
Exp-2	One smiles with her teeth and cheeks up.	One is grinning and showing her teeth.	One raises her cheeks, shrinks her eyes and opens wide her mouth with her teeth shown.	One's cheeks' muscle is raising and her chins are pulling down. Her muscle around eyes is shrinking while the muscle around mouth is extending.
Exp-3	One surprises, chins pulled down, eyebrows raised	One is surprised with her chins pulled down and eyebrows raised.	One's upper lip, cheeks, eyebrows raise, her jaw drops and her mouth stretched as O-shape.	One lifts the muscles of her cheeks and forehead, and she stretches the muscles of her jaw downward. She stretches the muscles around her eyes to open them wide.
Exp-4	One purses mouth moving to the left.	One is making her mouth to the left side.	One's lips are wiping to the left side.	One's left cheek is shrinking and her right cheek is stretching.
Exp-5	One purses mouth moving to the right.	One is making her mouth to the right side.	One's lips are wiping to the right side.	One's right cheek is shrinking and her left cheek is stretching.
Exp-6	One is angry, her brow is tightened, her nose is raised, and her upper teeth are exposed.	One looks angry with her eyebrows shrinked, nose upward and the upper row of teeth exposed.	One's upper lip and nose raise and her eyebrows clamps.	One' muscle around eyes is shrinking and her nose and upper lip is extending upward.
Exp-7	One wraps inner lips.	One's mouth is opening and her upper lip is warping inside the mouth.	One's upper lip is sucking and lips are parting.	One'upper lip is tightening inward and her chin is stretching downward.
Exp-8	One opens her mouth wide and sticks out her tongue down	One's mouth is opening wide and her tongue is being shown outside.	One's mouth is stretching and her tongue show.	One's mouth and tongue are stretching.
Exp-9	One puffs cheeks	One's cheeks are puffed.	One's cheeks are puffing.	One is stretching cheeks.
Exp-10	One smiles without her teeth.	One is smiling without teeth.	One's mouth is stretching wide.	One is stretching mouth.
Exp-11	One does not show her teeth and open her mouth wide.	One's mouth is opening in O-shape.	One' upper lip is streching upward and the bottom lip is stretching downward.	One is contracting her upper lip and chin and stretching bottom lip.
HS-0	One remains still.	One stay still.	One do not move.	One's face is relaxed.
HS-1	One wearing a mid-length and black wig turns around her head.	One wearing a mid-length and black wig is turning around the head.	One' head is turning right, up, left and down with a mid-length and black wig.	One's neck is stretching the head toward right, up, left and down with a mid-length and black wig.
HS-2	One wearing a mid-length and brown wig turns around her head.	One wearing a mid-length and brown wig is turning around the head.	One' head is turning right, up, left and down with a mid-length and brown wig.	One's neck is stretching the head toward right, up, left and down with a mid-length and brown wig.
HS-3	One wearing a long and black wig turns around her head.	One wearing a long and black wig is turning around the head.	One' head is turning right, up, left and down with a long and black wig.	One's neck is stretching the head toward right, up, left and down with a long and black wig.
HS-…	…	…	…	…
Sp-1	One reads a Chinese/ English text word by word.	One is speaking Chinese/ Englishwords.	One is talking with lips apart and strentched.	One is saying with the mouth streching and contracting.
Sp-2	One reads a Chinese/ English text sentence by sentence.	One is speaking Chinese/ Englishsentences.	One is talking with lips apart and strentched.	One is saying with the mouth streching and contracting.
Sp-6	One reads a Chinese/ English paragraph of a story.	One is speaking a Chinese/ English paragraph.	One is talking with lips apart and strentched.	One is saying with the mouth streching and contracting.
Sp-…	…	…	…	…

Figure S14: **Example of dynamic video activity descriptions.** We provide four perspectives of text descriptions about each video type's activity. Exp refers to expression-based video, HS refers to hairstyle-based video, and SP refers to speech-based video. "One" can be replaced by a subject.

3.6 Dataset Statistics Details

Since RenderMe-360 is a large-scale head dataset with multiple data, identity, and annotation, we unfold the statistic analysis into six aspects as below.

Identity. As shown in Figure S15 (a), we summarize data of captured identities in four dimensions, including age, height-weight, gender, and ethnicity. The subjects' ages range from 8 and 80 with approximate normal distribution, where teenagers and adults form the major part. A relatively large number of children and the elderly increase diversity of our assets. We show a height-weight distribution map, which indicates a large part of the models is located in height between 155cm and 185cm, and weight between 50kg and 90kg. Notably, the recorded height and weight data can support the physical nature perception of humans, which is an important question in commonsense reasoning. Our dataset is gender-balanced and divided into 4 ethnicities (217 Asian, 140 White, 88 Black, and 55 Brown). Ethnic diversity poses significant challenges and helps explore the margin and limitations of head avatar research.

Annotation. As mentioned before, we obtain a dataset with more than 243M frames which are fine-grained annotated. As Figure S15 (b) shows, there are three data collection parts of RenderMe-360, including Expression-Part, Wig-Part, and Speech-Part. Since frames in all the collection parts

15
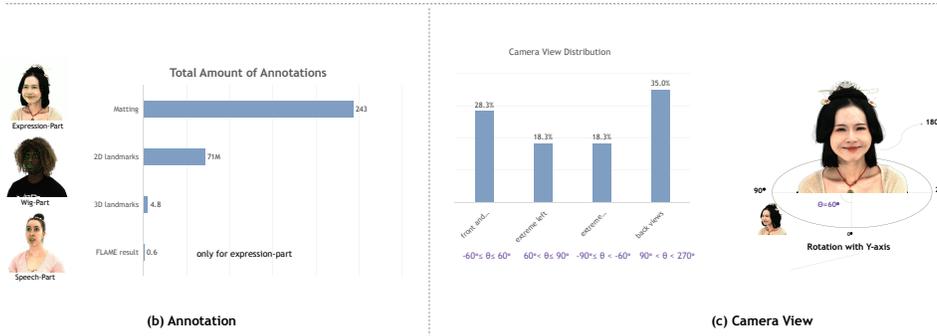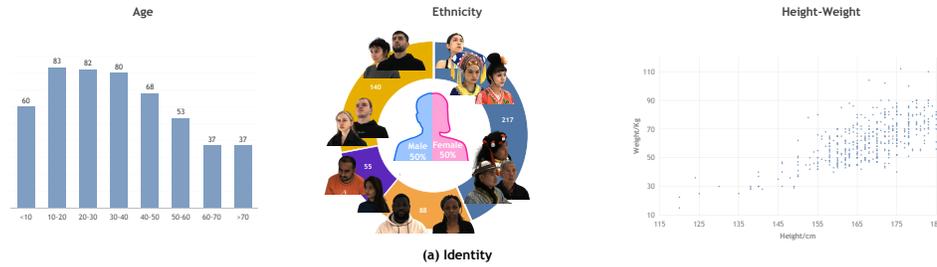
Figure S15: **General data distribution.** The data is summarized in three aspects, identity attributes, annotation, and camera view.
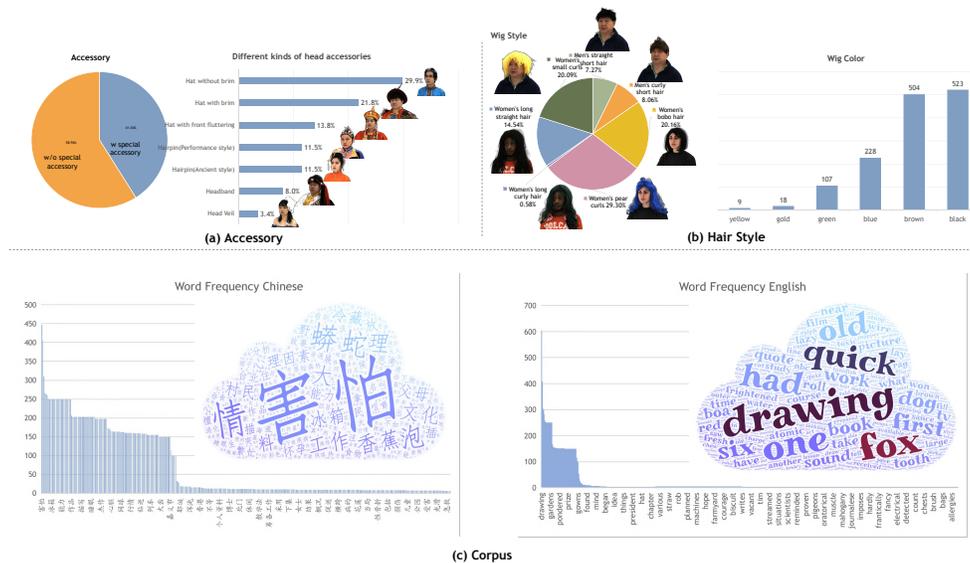
Figure S16: **Collection statistic.** We demonstrate the collection statistic on three sides, namely accessory, wig, and corpus. Better zoom in for details.

are annotated, there have over 243M frames with matting, 71M frames with 2D landmarks, and 4.8M frames with 3D landmarks. Since only frames in the expression collection are annotated with FLAME, we have 0.6M FLAME result in total. Besides, we also provide UV maps, AUs, appearance annotation, and text annotation. Rich and multimodal annotation provides more possibilities for downstream research and application.

Camera View. Since the POLICY contains 60 cameras which form four layers, we demonstrate the camera view distribution in Figure S15 (c). Camera views are divided into four groups based on rotation angle with the y-axis. Front and mild side views are convenient for face fitting algorithms,

16

extreme left and extreme right views are challenged for landmark detection, while back views are helpful with hair reconstruction.

Accessory. Parts of Asians (about 40%) are captured with special clothing and head accessories, while others are not, therefore, distributions of head accessories are only calculated among Asians, which is summarized in Figure S16 (a). The high diversity of accessories types, materials and textures presents huge challenges for head rendering and reconstruction.

Hair Style. As shown in Figure S16 (b), we have 7 styles for wigs, 2 with men's styles, and 5 with women's styles. We randomly sampled about 10 wigs for captured subjects, wig styles are not specified for gender. 6 colors are not evenly distributed among each wig. Therefore, subjects captured with black and brown are the majority in our dataset, while yellow color has the least portion. Due to the hair-related benchmark, the complexity of hair structure and the dynamic deformation during large head motion challenge the SOTA methods, and the large hair assets provide a great database for the application of hair rendering and reconstruction as well as the potential research opportunity for cross-identity hairstyle transfer and animation.

Corpus. We calculated the word frequency for Chinese and English separately. From the cloud visualization, word frequency is indicated by the size of each character. The most frequent word "Hai Pa" in Chinese appears nearly 450 times among all sentences, while the least frequent one "Ji Jiu" is less than 50. We only summarize the phrased in Chinese, but not single characters like "de", "shi", "wo" and etc., since there have no specific implications. Among English, the most frequent word, "Drawing", occurs more than 600 times, while the least frequent one "Ambitious" is close to 0. The corpus statistic and "word cloud" are demonstrated in Figure S16 (c). Since our collection contains cross-identity repeated corpus and also different corpus, it is beneficial for the construction of the generalizable talking model.

4   Benchmarks Details

Based on the RenderMe-360 dataset, we construct comprehensive benchmarks on five critical tasks to showcase the potential usage of our data, and reflect the status quo of relative methods. Due to the space limitation, some experiments and settings are not described in the main paper in detail. In this section, we first introduce the criterion to divide our dataset splits. Then, we provide a detailed discussion on benchmarks – 1) We analyze the novel view synthesis benchmark with more qualitative results, and additional quantitative ablations. 2) For the intra-dataset evaluation, we provide additional experiments with different training settings from the main paper. 3) We provide more experiments and qualitative visualizations for the Cross-Dataset Evaluation to serve as complementary demonstrations to the ones in the main paper. 4) We provide novel expression synthesis, hair rendering, hair editing, and talking head generation benchmarks with different training and testing settings.

4.1   Benchmark Splits

When it comes to rendering the human head, different attributes of head performance have impacts on rendering tasks with different magnitudes. For example, the high-frequency texture, detailed geometry, the reflection effects under different materials, and the accessories which have different deformation caused by human head, all these factors are challenging and crucial for rendering tasks. To conduct a thorough evaluation of state-of-the-art methods, we split benchmark data for head-centric rendering tasks, with spanning difficulties in the hierarchy. Figure S17 shows a preview of split data samples. Concretely, we follow the defined rules to split data: (1) Normal Case. Normal cases are identities without any accessories; (2) With Deformable Accessories. Identities who wear deformable accessories, , hair band, normal hat, .(3) With Complex Accessories. Identities have accessories with sophisticated structures or textures, , gauze kerchiefs, complex earrings, or hats with pendants. For each task, we sample data from these three groups with different sampling principles, according to the characteristics of specific tasks. Please refer to the corresponding subsections for more details.

17
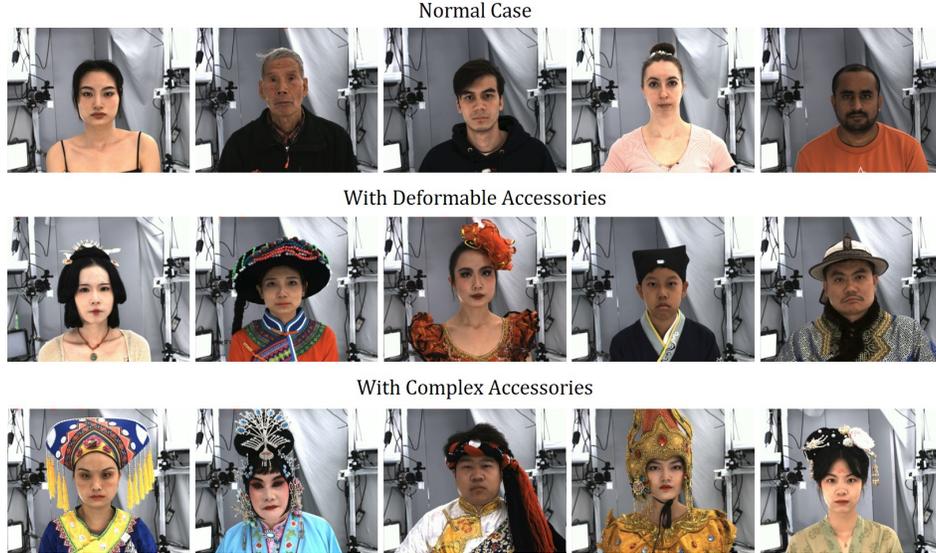
Figure S17: **Samples in benchmark splits.** We create three splits for benchmark evaluation, depending on the accessory difficulty, namely, 'Normal Case', 'With Deformable Accessories', and 'With Complex Accessories'.

4.2 Novel View Synthesis

Detailed Settings. As mentioned in the main experiment part, for *#Protocol-1* we evaluate the performance of novel view synthesis among four state-of-the-art methods. Specifically, we select two expressions from each subject, which means we train 40 models for Instant-NGP [50] and NeuS [83] respectively, and 20 models for MVP [42] and NV [40] respectively. Note that two expression sequences of one identity are trained with same configuration. For each model of Instant-NGP and NeuS, we have 38 camera view images for training and 22 camera view images for testing, while the whole sequences of the selected expressions, which has in total about 8000 frames of 38 training views, are fed into the training of MVP and NV. For preprocessing, images are resized and matted to 512×512 with white background. Note that to get more stable rendering results, we do not resize the image and use a black background for Instant-NGP. We train $30k$ iterations for Intant-NGP to get sufficient convergence of the model, $200k$ iterations for MVP, and $50k$ iterations with batch size 16 for NV. The other settings of these four methods are as same as the default implementations in [50, 83, 41, 40]. If not specified, we use the V100 GPUs to train the models.

Additional Qualitative Results. The qualitative result is shown in Figure S18, all four methods function normally in reconstructing the selected subjects, but with different performances. For the normal case, we mainly focus on high-frequency parts like hair and beard. As shown in the zoom-in regions of the first and three rows, NeuS and Neural Volume can reconstruct the head shape and most of the facial features, but fail to render hair and beard in detail. Instant-NGP and MVP perform well in hair/fur, whereas there is still a gap between rendered image and ground truth. For the subjects with deformable accessories, we pay attention to the accessories with different textures. As demonstrated in the middle left case, NeuS fails to reconstruct the bead-like shape of the fabric hat, and tends to smooth and form long stripes. This indicates Neus' disability to recover objects with complex textures. From the subject in the middle right, we can observe that Neural Volume produces many artifacts in the neck, eyes, and flower-like semi-transparent accessory. Finally, for the identities with complex accessories, we observe that Instant-NGP and MVP can render rigid or non-rigid accessories, like pendants, gemstones, feathers, and fabric slings, with high-frequency texture results. Scattered hair on the skin is failed to synthesize properly in all methods.

18
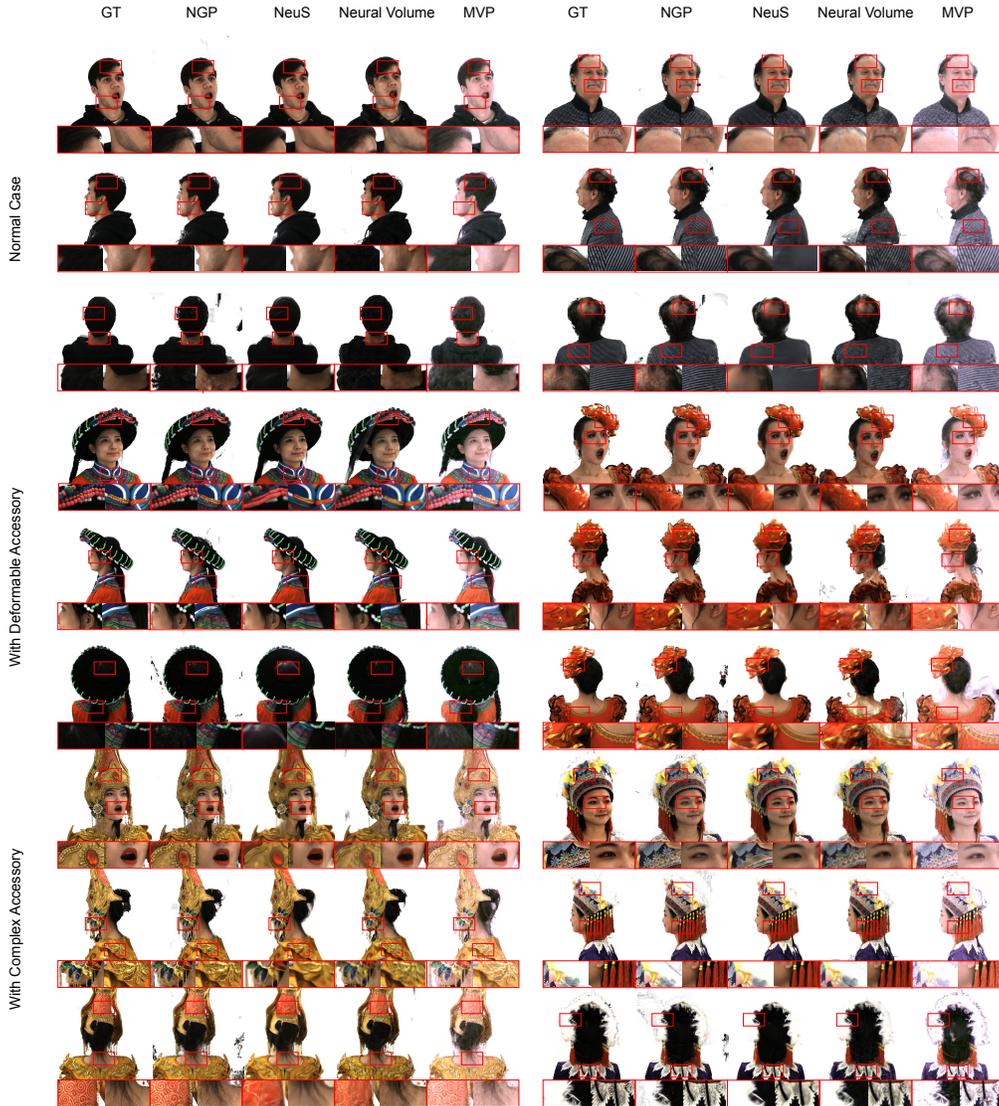
Figure S18: **Illustration of qualitative novel view synthesis (*#Protocol-1*).** We sample two subjects in each data split and show the novel view synthesis results in three different test views (frontal, side, back) among four methods. NeuS performs well with almost no surrounding noise but has a much smoothing surface, while Instant-NGP produces a lot of surrounding noise and can recover some high-frequency parts. MVP renders lighter and more refined results, and Neural Volume renders skins mostly with many artifacts.

19

Table S3: **Ablation study of camera split (#Protocol-2).** We set up the experiments with three camera splits and four methods.

Split	Metrics	NGP [50]	NeuS [83]	NV [40]	MVP [42]
Cam Split 0 [train 56, test 4]	PSNR↑	26.27	23.34	18.29	23.87
	SSIM↑	0.879	0.892	0.717	0.887
	LPIPS↓	0.11	0.14	0.33	0.13
Cam Split 1 [train 38, test 22]	PSNR↑	22.46	23.39	18.56	23.1
	SSIM↑	0.808	0.888	0.723	0.876
	LPIPS↓	0.15	0.14	0.33	0.15
Cam Split 2 [train 26, test 34]	PSNR↑	22.07	22.48	18.12	23.02
	SSIM↑	0.789	0.846	0.72	0.868
	LPIPS↓	0.17	0.22	0.33	0.14
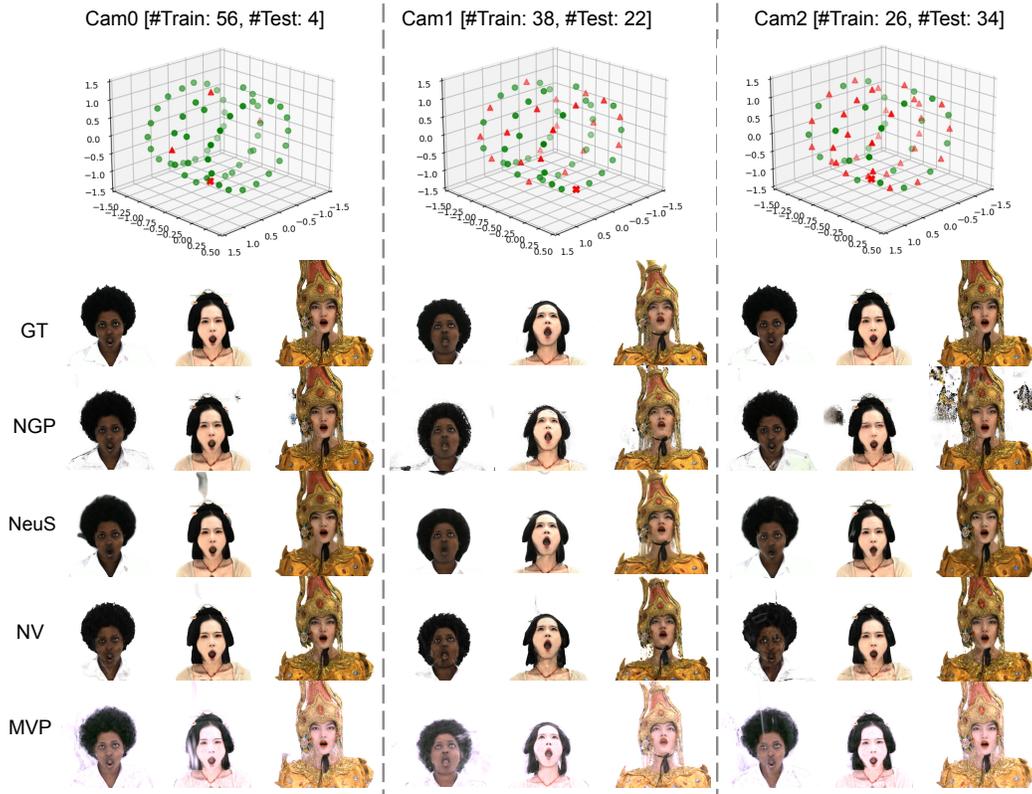
Figure S19: **Illustration of camera split ablation (#Protocol-2).** We select and visualize three different camera settings, which are visualized on the top side of the figure. Green circles stand for training views, red triangles stand for testing views. We demonstrate three subjects in different data groups rendered with same expression. The visualized novel camera views are marked as ✗ in the camera split visualization.

4.2.1 Camera Split Ablation for Single ID NVS

Settings. For *#Protocol-2*, in order to ablate methods with various training and testing camera splits on rendering results, we design three kinds of camera distribution and retrain the above methods, comparing the metrics. Three kinds of camera splits contain 'train 56, test 4', which means most of the camera views are used in training, 'train 38, test 22', which is the original distribution, 'train 26, test 34', which means more testing views than training views, and all testing views in 3 splits are uniformly distributed. We select 3 representative subjects from the above-mentioned subset, and 1 from each predefined split. The training settings are the same as in Section 4.2, except for the distribution of the training views.

20

Results. The quantitative result is shown in Table S3. As the number of training views decreases, a decline in the metrics appears in Instant-NGP [50]. Interestingly, when adding up the number of training views from 38 to 56, the performance of the other three methods remains roughly consistent, which indicates the number of training cameras above a certain threshold may not play a key role in performance. When we decrease the number of training views to 26, all methods have a decline of metrics, and NeuS [83] performs relatively better.

As the demonstration of qualitative result in Figure S19, there is no large gap in the visual result between Cam0 and Cam1 in all three subjects. For Instant-NGP [50], more details on accessories are reconstructed as more training views provided, while with fewer training views, more noise and artifacts occur on the face and the surrounding area. For NV [40], artifacts also gets more when fewer views are involved into training , and it smooths the high-frequency details in all three settings. There is not much difference among three camera settings for MVP [42] and NeuS [83], but they fail to render high-frequency details with fewer training cameras and generate artifacts as well.

4.2.2 Generalizable NVS

Detailed Settings. As mentioned in the main paper, we train all models in both protocols with 10 expressions performed by 187 identities. For *#Protocol-1* we evaluate novel view synthesis on two unseen expressions on a subset of the training identities. Specifically, we select 20 identities in total – 10 normal cases, 5 with deformable accessories, and 5 with complex accessories. For *#Protocol-2*, 20 unseen identities are tested with the same splitting strategy. Noted that during training and testing, three source views are used in all experiments, and we crop and resize the source and target views to the 512×512 resolution, and render the images with white background.

Additional Results. Recall that, in the main paper, we find that KeypointNeRF [46] achieves good visual quality while getting the worst quantitative results among all generalizable methods. We discuss the possible reasons behind the phenomenon in the main paper, where the major miss-alignment comes from the non-facial parts, like body parts of the rendered images(such as missing shoulders). Since KeypointNeRF [46] tends to anchor the geometry using the relative encoding of facial key points, the body part with no keypoint encoding tends to reconstruct the intersection region from source views. Here, we further provided a quantitative demonstration from another perspective. Concretely, we re-compute the benchmark results in Tab.4 of the main paper under a different masked region. In the main paper, we calculate metrics of rendered raw full images compared with ground truth. Here, in Tab. S4, we only calculate the regions that KeypointNeRF could render. As shown in the Table, The PSNR results of all methods get higher under this new setting, and KeypointNeRF [46] outperforms IBRNet [84] and VisionNeRF [36] in SSIM and LPIPS, which accords with our visual observation.

Table S4: **Masked results on generalizable NVS.** We re-calucuated the overall metrics on masked images in Table 4 Unseen ID NVS.

Train Setting	Test Setting	Methods	PSNR↑	SSIM↑	LPIPS↓*
Fixed Views	Fixed Views	IBRNet [84]	23.70	0.889	135.16
		VisionNeRF [36]	24.32	0.893	139.27
		KeypointNeRF [46]	24.75	0.901	103.78
	Random Views	IBRNet [84]	22.25	0.895	157.96
		VisionNeRF [36]	22.58	0.874	157.54
		KeypointNeRF [46]	22.40	0.861	143.265
Random Views	Fixed Views	IBRNet [84]	24.84	0.903	102.57
		VisionNeRF [36]	25.80	0.902	118.72
		KeypointNeRF [46]	25.12	0.910	85.39
	Random Views	IBRNet [84]	24.24	0.895	102.50
		VisionNeRF [36]	23.11	0.879	149.62
		KeypointNeRF [46]	24.715	0.890	85.94

21

Figure S20: **Qualitative results of generalizable novel view synthesis (*#Protocol-1&2*).** We illustrate some qualitative results of the generalizable methods, including IBRNet, KeypointNeRF, and VisionNeRF in two different settings, namely synthesizing the novel identifies and synthesizing the novel expressions. Two samples for a case are shown, and the regions in red boxes are zoomed in for better comparison.

22

Table S5: **Explanation for training-testing settings in generalizable NVS.** All settings are evaluated on the same camera split of target views, and source views are selected apart from the target views. Tested random views are constrained under a certain angle range. At inference, three source views are provided.

Training Setting	Testing Setting	Explanation
Fixed Source Views	Fixed Source Views	The model is trained given fixed source camera views and tested with the same source view indexes.
	Random Source Views	The model is trained given fixed source camera views and tested with random source view indexes.
Random Source Views	Fixed Source Views	The model is trained given random source camera views and tested with the fixed source view indexes.
	Random Source Views	The model is trained given random source camera views and tested with re-random selected source view indexes.

4.3 Additional Results of Intra-Dataset Evaluation

In order to evaluate the relationship between the performance of the model and the size of input data, we additionally split the training set into 3 parts with different settings similar to the split of the test set depending on the similarity of decorations between different identities. Besides, random samples with different amounts of data (30%, 50%) are also evaluated in training. The results can be seen in Tab S6. Consistent with common experience, we find that the metrics declined when decreasing the number of identities in training. For all methods, there's an abrupt increase when more data is included in training, whatever the difficulty of the training identities, which shows that a complete set of whole data is necessary for training a satisfying model that can generate on the novel identity of person. As for the different settings in the training split, we find that whatever the setting in the test set, with more data in subset 1, the trained model shows more advanced results in evaluation, with only a few exceptions that may due to random perturbations. Moreover, we also visualize the overall metrics with masked regions computed in Fig. S21, we can find the same phenomenon with non-masked metrics yet with better absolute values. Also in Fig. S21, we can find models train only on split achieve the best quality on the same test split while generalizing poorly on other splits, when the data coverage has no bias, eg. random 30%, random 50%, and full set, the performance variance between splits get relative smaller. Moreover, when the data scale gets larger, the more robust the metrics are across different splits. Interestingly, we observe that the VisionNeRF [36] model trained on subset 3 which contains the smallest scale of data in all experiments gets the worst result. The main reason might be the codebook training in VisionNeRF [36] typically highly rely on the amount of data.

4.4 Cross-Dataset Evaluation

We further compare the results of training in our dataset with other multi-view face datasets. FaceScape [94] is a dataset with multiview captured faces in ideal experimental conditions. All the people captured covered their hair with a cloth so as to show only the quality of the face region. Most of the people are Asians, and overall 359 identities and 20 different expressions are captured. Note that we do not follow the same setting in MofaNeRF [107], where only synthetic renderings of reconstructed mesh are treated as training sets. For MultiFace [90], a multiview capture system photoing 13 different identities of human heads with different expressions. Most of the people are Europeans, and the light condition is darker. Since we want to find the performance in real-world circumstances, we pre-process the photos initially captured to align with our dataset, and evaluate on those images. We further train different methods on both datasets and evaluate the results on ours, facescape, multiface with 3 different models trained on each one. Models tested on cross dataset is performed directly without any further finetuning.

Detailed Settings. For Multiface dataset, we train on the 10 identities of v1 version and the rest 3 identities of v2 version is left for testing. For Facescape dataset, the first 300 subjects are selected as the training set and the rest 59 as testset. Since a registered head is provided as the standard face coordinate, we map the mean face with our FLAME model, and re-calculate the world matrix of

23

Table S6: **Intra-Dataset evaluation.** We qualitatively evaluate the impact of data distribution and data scale of the proposed dataset. The reported numbers are from models.

IBRNet[84]

Training Set	Normal Case			With Deformable Accessories			With Complex Accessories			Overall		
	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*
Subset 1	21.87	0.893	162.60	17.06	0.798	245.81	15.83	0.722	327.81	19.16	0.827	224.71
Subset 2	20.43	0.871	183.88	17.71	0.809	220.61	16.80	0.732	293.37	18.84	0.821	220.44
Subset 3	18.76	0.844	214.59	17.06	0.795	236.54	16.40	0.718	306.26	17.75	0.800	243.00
Random 30%	21.06	0.883	167.73	17.09	0.797	237.74	16.63	0.730	293.96	18.96	0.823	216.79
Random 50%	21.69	0.892	158.18	17.98	0.815	212.82	17.35	0.748	276.99	19.68	0.837	201.54
Full set	22.53	0.897	154.05	18.75	0.830	195.12	18.10	0.749	250.72	20.48	0.843	188.49

KeypointNeRF[46]

Training Set	Normal Case			With Deformable Accessories			With Complex Accessories			Overall		
	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*
Subset 1	18.79	0.883	139.09	14.49	0.753	243.03	16.25	0.767	234.20	17.08	0.822	188.85
Subset 2	17.89	0.868	172.29	14.54	0.744	260.71	16.61	0.768	228.94	16.73	0.812	208.56
Subset 3	17.68	0.863	179.45	14.37	0.746	259.20	16.88	0.774	219.60	16.65	0.812	209.43
Random 30%	18.47	0.876	148.56	14.39	0.747	239.90	16.51	0.765	213.30	16.96	0.816	187.58
Random 50%	18.26	0.871	167.86	14.84	0.743	253.25	16.69	0.766	226.85	17.01	0.813	203.96
Full set	18.02	0.865	145.30	15.75	0.794	194.16	16.15	0.747	227.49	16.99	0.818	178.06

VisionNeRF [36]

Training Set	Normal Case			With Deformable Accessories			With Complex Accessories			Overall		
	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*	PSNR↑	SSIM↑	LPIPS↓*
Subset 1	21.34	0.878	147.30	18.41	0.791	208.70	16.75	0.721	207.10	19.33	0.812	179.00
Subset 2	20.32	0.891	159.21	19.16	0.806	201.76	16.60	0.737	218.60	18.98	0.826	186.30
Subset 3	18.74	0.791	190.45	17.59	0.756	218.60	16.62	0.745	280.50	17.86	0.769	222.88
Random 30%	20.29	0.883	129.20	17.05	0.812	203.10	16.91	0.714	213.86	18.55	0.817	170.98
Random 50%	20.65	0.886	153.20	17.01	0.817	203.90	16.86	0.724	214.78	18.70	0.823	182.86
Full set	24.77	0.918	110.40	20.22	0.858	149.30	19.35	0.797	196.90	22.28	0.873	141.75

those images to match all the images with our input. To make all the input images with the same size as we trained in RenderMe-360, we find the nearest z-axis of the rotation matrix in our captured data as the marker, and place the head with an additional affine matrix between the two camera-to-world extrinsics. Then all the inputs and source views become similar for different datasets, and we start training in these images. For each experimental setting, we train on one dataset's train split, and test on another's test split (train and test may belong to the same kind of dataset), to testify the generalization ability of the trained model for different datasets. We follow the training setting with random source view and random test view from Section 4.1.2 in the main paper.

Result. The qualitative results with different settings can be seen in S22. Since our dataset has more data than Multiface and large variance (hair and clothes variance v.s. only face region), the testing metrics show superior results over models trained in our dataset. From the results, we can also see, that with only a few identities training, most methods cannot show a meaningful generation result on unseen identity, although KeypointNeRF [46], with 3D facial landmarks as anchors for face position can roughly sketch the head contour, they do not perform well with training on Multiface. However, with plenty of training data in RenderMe-360, we can detect a convincing result even without any finetune on unseen data in Multiface. That proves the generalization ability of training with a large number of person identities.

Another visualization in Figure S23 shows the comparison of the Facescape rendering result between the two methods. The model trained on our dataset has the ability to generate competitive results compared to the inference result trained from Facescape dataset. IBRNet can produce more reasonable results of the face part although parts of the face are missing. This also proves the robustness of the generalization ability when training a generalizable methods with our dataset.

4.5 Novel Expression Synthesis

This task refers to the setting of reconstructing a 4D facial avatar based on *monocular* video sequences[1]. We study three representative methods with different expression settings – 1) *#Protocol-1* for investigating the interpolation/extrapolation abilities of training on intentional expression structures and testing on novel ones. 2) *#Protocol-2* for exploring the robustness of training on normal

---

[1]Note that, differing from unseen expression NVS protocol, the novel expression should be synthesized under the guidance of *non-target person's image* prompts, such as facial expression parameters. The main focus of this setting is to evaluate methods' effectiveness in *dynamic changes* of the surface of a face.

24
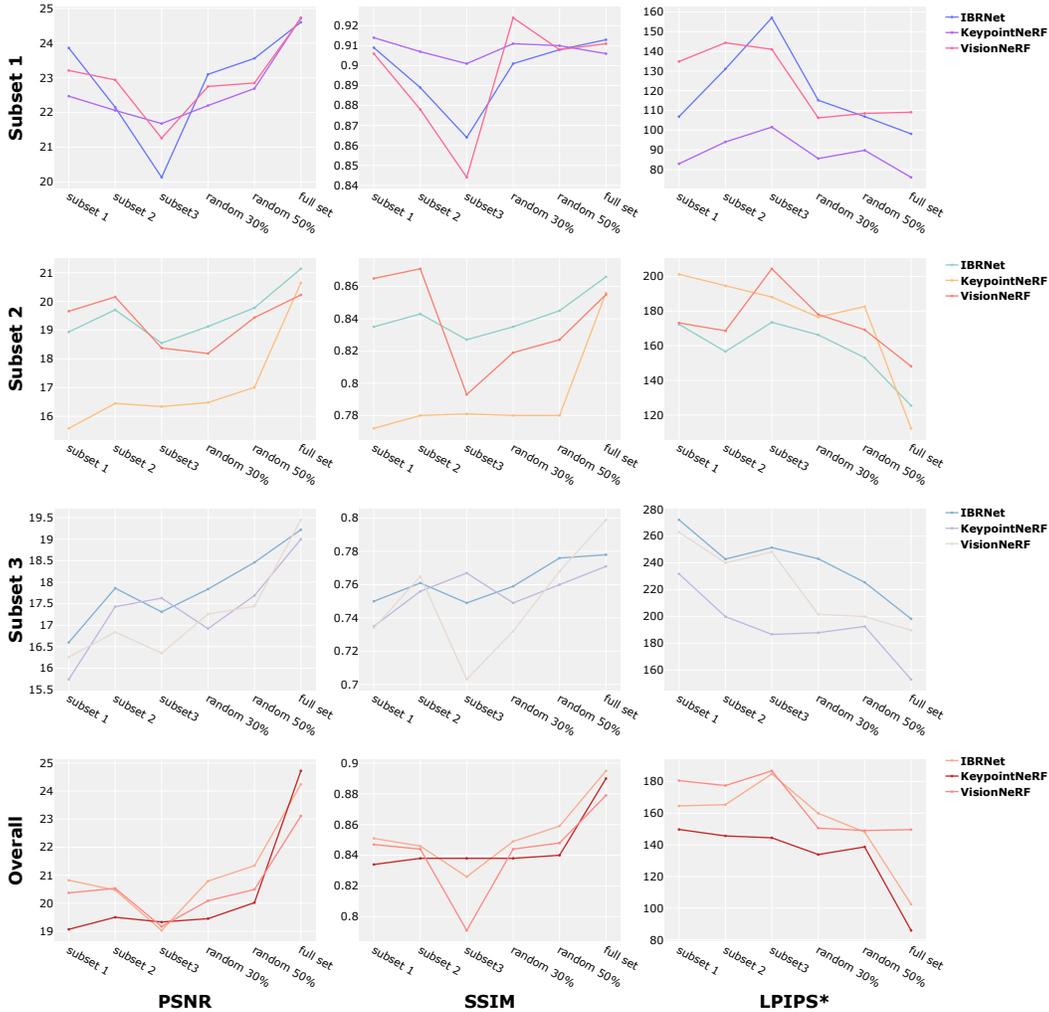
Figure S21: **Intra-dataset metrics with masked region calculation.** We also visualize the intra-dataset metrics of the masked region on each split. With only the masked region evaluated the absolute value is typically higher than Tab S6, while the trend fits. Also, models training on only one subset easily overfit on specific split of data, while generalizing poorly on other subsets.

conversation sequences, then testing on both new conversations and intentional expression structures. The normal conversation scenarios include subtle expression changes. They can help to verify a method's reconstruction on local motion transformation. The intentional expression structures provide challenges of reconstructing 4D information in high-frequence texture/geometry, and multi-scale motion changes.

***#Protocol-1* Settings.** We study three case-specific, deformable head avatar methods: NeRFace [17], IM Avatar [102], and Point Avatar [103]. These methods showcase different paradigms of leveraging neural implicit representations for dynamic head avatars. The official implementation of IM Avatar suffers from unstable training when not using specific GPU [2] We find one of the sensitive factors might relate to the FLAME parameters. We follow the official released data preprocessing pipeline of IM Avatar, where the FLAME parameters are initialized from DECA [15] and refined with single-view facial keypoints[3]. To obtain relatively stable results (shown in Table S7), we also compare

---

[2]This problem is frequently raised in GitHub Issues, *e.g.,* `https://github.com/zhengyuf/IMavatar/issues/3`, of the official release version.

[3]We abbreviate the preprocessing pipeline as DECA in the follow-up sections with less rigorous.

25
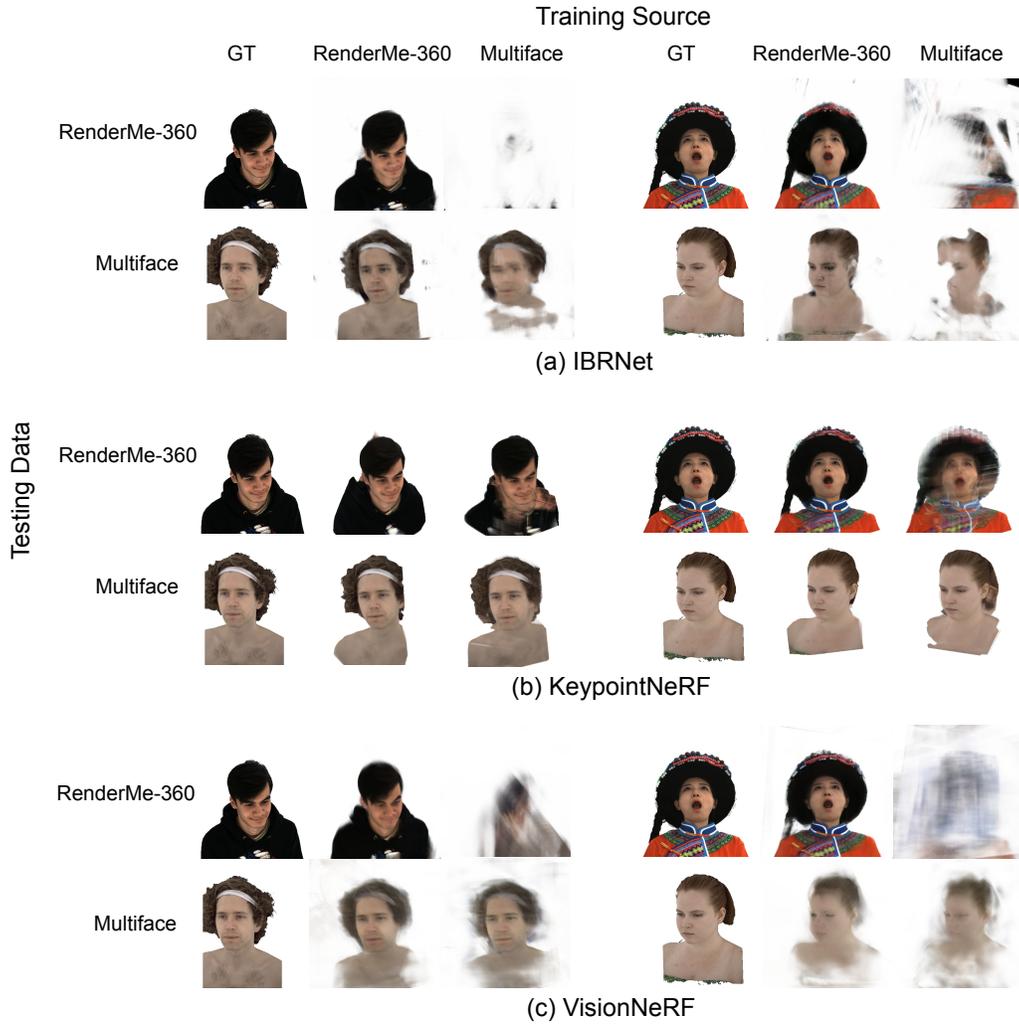
Figure S22: **Illustration of cross-dataset evaluation.** We visualize the result from three methods, IBRNet, KeypointNeRF, and VisionNeRF, between two datasets, RenderMe-360 and Multiface.
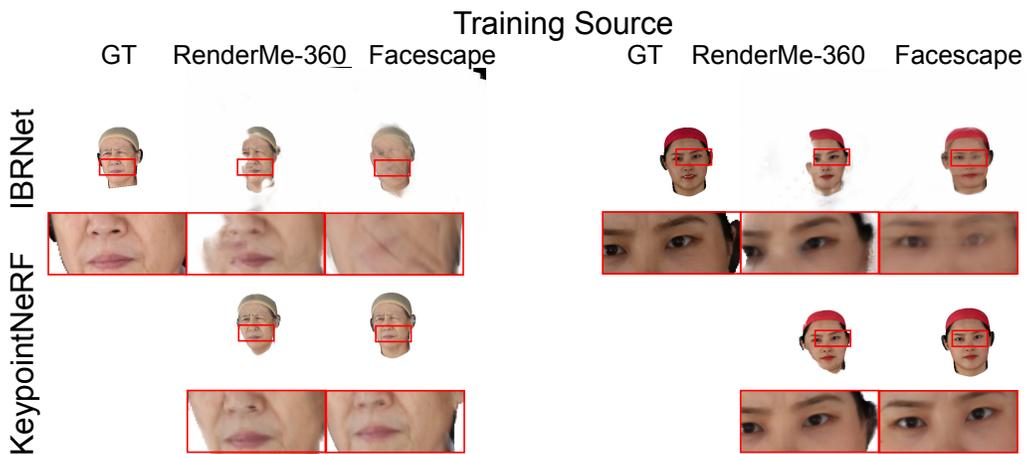
Figure S23: **Illustration of cross dataset experiment on Facescape [94].** We visualize the model rendering results from Facescape, which take RenderMe-360 and Facescape as training source respectively.

26
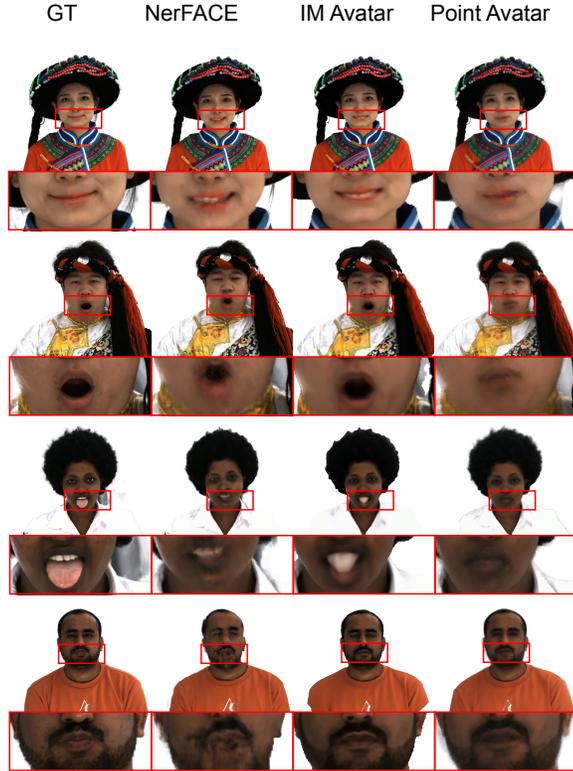
Figure S24: **Illustration of novel expression synthesis (*#Protocol-1*).** We showcase four samples from both normal expression and hard expression splits.

the results from DECA and our optimized FLAME parameters, which are shown in Supplementary Materials. All methods are evaluated in terms of PSNR, SSIM, LPIPS, and L1 Distance, similar to [103]. For *#Protocol-1*, we select 20 identities from the three categories to form the benchmark data. We use 6 expression sequences for per-identity training and the other 6 expressions for testing.

#Protocol-1 **Results.** The quantitative result is presented in Table S7. We split the novel expressions into normal and hard subsets according to their similarity to the training expression structures. We find PointAvatar outperforms the two implicit-based methods on both splits under most of the metric measurements. The comparison suggests that combing explicit point-based representation with implicit one helps increase the robustness of new expression synthesis. This is reasonable since point cloud provides more flexibility and specificity in geometry deformation than pure implicit ones. But such a merit does not always exist. The granularity of points limits PointAvatar's performance on subtle motions (*e.g.,* 'pout' in the last row of Figure S24). In addition, we observe that all methods suffer from out-of-distribution cases like the 'tongue out' in the third row of the Figure. Moreover, from the whole-head rendering aspect, we find that IM Avatar struggles with thin structures like twisted hair band and hair strands. This is because IM Avatar constrains reconstruction on the surface. NerFace has fine rendering results in a global manner, while facing problems in robustly modeling dynamic motion.

#Protocol-2 **Settings.** As mentioned in the main experiment part, we evaluate the performance of novel expression synthesis among three state-of-the-art methods, namely NeRFace [17], IM Avatar [102] and Point Avatar [103]. Here we elaborately discuss the experiments for *#Protocol-2*, in which we select the same 20 identities to form the benchmark data. We use 2 sequences of verbal (about 1700 to 2000 frames) for training, another 1 unseen verbal sequence and 11 expression sequences (exclude the natural expression) for testing. All data samples used in *#Protocol-1&2* are resized and matted to 512×512 with white background. We train $1000k$ iterations for NeRFace, 100 epochs for IM Avatar, 65 epochs for Point Avatar. We keep other training configurations the same as

27

Table S7: **Novel expression synthesis (*#Protocol-1*).** We benchmark three methods on different splits of RenderMe-360. **N**: Normal Expression, **H**: Hard Expression.

Method	Split	$L_1 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
IM Avatar [102]	N	0.047	22.61	0.903	0.134
	H	0.047	21.91	0.895	0.149
NerFace [17]	N	0.034	20.46	0.876	0.114
	H	0.037	18.89	0.865	0.121
PointAvatar [103]	N	0.0057	24.57	0.878	0.089
	H	0.0055	25.05	0.883	0.086

the default one, whose details are referred to [17, 102, 103]. All methods are evaluated in PSNR, SSIM, LPIPS, and L1 Distance, similar to [103].

***#Protocol-2* Results.** The quantitative result is shown in Table S8. We find that Point Avatar [103] achieves the best performance on the 'Speech' set in terms of the average for 'PSNR', 'SSIM', 'LPIPS', while NeRFace [17] performs relatively better on the expression test data in total. Since the official implementation of IM Avatar is unstable in training, we can only show the results with the intermediate saved checkpoint. This contributes to IM Avatar's underperforming over other methods by a large margin. There exists a clear gap in the quantitative result between the speech and expression data in IM Avatar [102] and Point Avatar [103]. We attribute this difference to a different distribution of data. Since the speech data is mostly interpolation data, and the expression data tends to be extrapolation data. In addition, the qualitative result provides pieces of evidence from another perspective, which are shown in Figure S26. IM Avatar collapses in the mouth parts and fails in detail synthesis (such as hair, and accessories). PointAvatar shows a high-quality performance in generating a 3D avatar, which reconstructs tiny strands of hair, while suffering from dynamic unseen expressions. NerFace also shows a strong ability to generate a 3D avatar that can extrapolate to simple unseen expressions. These methods all perform fine when interpolating into another verbal video, whereas struggle with extrapolation like Speech-to-Expression.
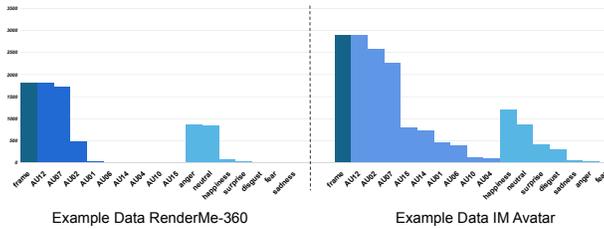
Figure S25: **Comparison of training data between RenderMe-360 and IMavatar.** We summarize the frames, AUs, head poses, and expressions between the example data from RenderMe-360 and data from IMavatar.

We also perform the ablation experiments that trained with different FLAME fitting parameters, as shown in the last two rows of Table S8. Specifically, DECA applies a model-based single-view fitting process, while our annotation pipeline designs a multi-view fitting process with the supervision of corresponding scan and images. We quantitatively compare the fitting quality, by calculating the facial landmark distance metric, which stands for the fitting error and reflects the quality of the expression parameters. For 99.3% of the data, the fitting result from our pipeline has better fitting quality. We further calculate the L2 difference of the shape parameter from the mean face to aligned identities, and obtain the result (14.115 in our pipeline, compared to 2.77 from DECA). This phenomenon reflects that DECA tends to produce results converging to the mean face.

We further sample and visualize the FLAME result between two methods in Figure S27. Our produced results mimic the motion of the mouth and eyes better, and cover richer details in geometry.

28
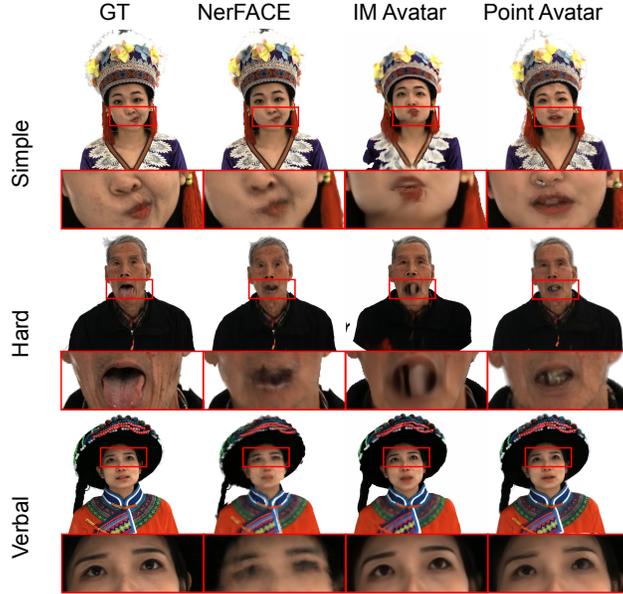
Figure S26: **Illustration of novel expression synthesis (#Protocol-2).** We select three different identities from different levels of difficulty. The first line is the simple expression, the middle line is the hard expression and the last line is the interpolation result of another verbal video.

Table S8: **Novel expression synthesis (#Protocol-2).** We evaluate three methods on the novel expression synthesis task on different splits of RenderMe-360. **EN**: Normal Expression, **EH**: Hard Expression, **S**: Speech.

Method	Split	$L_1 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
NerFace [17]	EN	0.0338	22.23	0.826	0.1264
	EH	0.0369	21.4	0.815	0.1351
	S	0.03	20.51	0.848	0.1499
IM Avatar [102]	EN	0.148	14.45	0.723	0.2751
	EH	0.1522	14.5	0.718	0.2812
	S	0.071	20.61	0.828	0.1754
PointAvatar [103]	EN	0.01	21.99	0.854	0.1097
	EH	0.0103	21.83	0.852	0.1112
	S	0.0032	26.95	0.917	0.0598
PointAvatar [103] (with DECA [15])	EN	0.0093	22.68	0.861	0.103
	EH	0.0099	22.3	0.856	0.107
	S	0.0034	26.83	0.914	0.0607

Interestingly, a better FLAME fitting result does not contribute too much performance boost on Point-Avatar. As shown in the table, Point-Avatar trained with better FLAME parameters performs slightly better on the conversation sequences, but lags behind on intentional expression sequences. We guess the possible reason lies in the characteristics of the training and testing data. Compared with the training data used in the original paper (two of the subjects used in Point-Avatar are from IMavatar's dataset), our conversation sequences are more challenging for Speech-to-Expression settings (*i.e.,*, EN, EH in the Table S8). As shown in Figure S25, the facial attributes of our data are more challenging, as the main changes are around the mouth and fewer expressions pop up during the speech sequence. This leads to a larger distribution gap between training and testing scenarios. Moreover, since our FLAME pipeline produces better-aligned results in expression parts that are far away from the mean face (Figure S27), the trained model struggles with these out-of-distribution cases, and has relatively lower metric performances than the ones trained on the FLAME version that is inaccurate but smooth across the sequence.

29
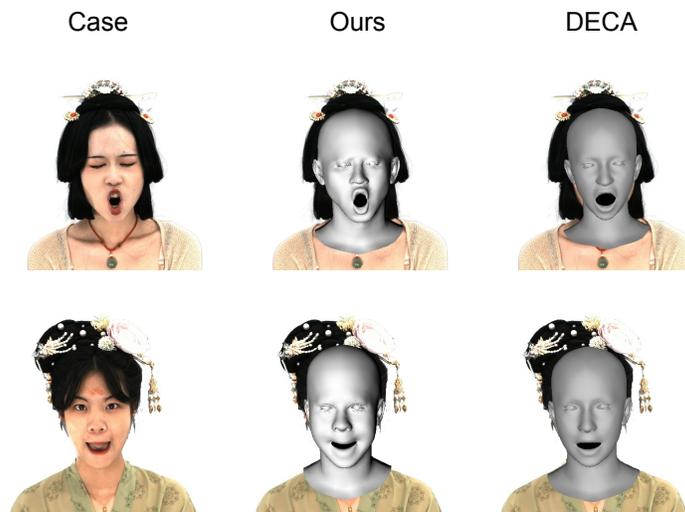
Figure S27: **Examples for comparison of different FLAME fitting quality.** We compare and visualize FLAME fitting results from RenderMe-360 and DECA. DECA is the processing pipeline of the official implementation of IM Avatar and Point Avatar.

4.6 Hair Rendering

This task refers to the setting of modeling accurate hair appearance across changes of viewpoints or dynamic motions. We focus on three sub-problems of hair rendering: 1)*#Protocol-1* for probing current methods' effectiveness on static hair reconstruction, in which methods are trained on multi-view images and tested on novel views; 2) *#Protocol-2* for evaluating the algorithms' capability on dynamic hair performance capture, in which methods are trained on multi-view video sequences and tested on the motion sequences under novel views; 3) *#Protocol-3* for investigating the methods' interpolation ability on dynamic hair motion, in which the methods are trained on frames sampled from a monocular video, and tested on the rest frames of the video.

Settings. We select a subset from RenderMe-360 to form the benchmark for this task, with 20 representative wig collections from 8 randomly picked human subjects. This subset is further split into three groups, *i.e.,* short hair, long hair, and curls, according to the complexity of hair strand intersections. In total, we study six representative methods under the three mentioned protocol settings (Instant-NGP [50] and NeuS [83] for *#Protocol-1*, MVP [42] and NV [40] for *#Protocol-2*, NSFF [35] and NR-NeRF [77] for *#Protocol-3*). The evaluation metrics are PSNR, SSIM, and LPIPS. Concretely, we discuss Instant-NGP [50] as well as NeuS [83] for *#Protocol-1*. We train the models with 38 camera views of a specific frame (the one with the largest motion magnitude in the video) and evaluate their performances with the rest 22 views. The distribution of camera split is the same as the one in the main paper. For *#Protocol-2*, we study two dynamic neural rendering methods – MVP [42] and NV [40]. The methods are evaluated under 4 held-out views of motion sequences. The four views are distributed around the front, double side, and back of the human head. For training, the other 56 views of the motions are fed into the models. For *#Protocol-3*, we reveal the effectiveness of NSFF [35] and NR-NeRF [77]. We take a camera from a frontal view as the monocular camera, and sample the input sequence in 10 FPS. The rest frames are used as evaluation data. This strategy results in about 30 frames for training per motion sequence and 60 frames for testing. The training data volume is similar to the original papers, while the testing data volume is larger for a more comprehensive evaluation. Note that, hair rendering is a long-standing task, and there are many instructive methods. For example, state-of-the-art multi-view hair rendering methods like HVH [87], and Neural Strand [60] are also valuable. However, most of the methods are not open-sourced, and difficult to be re-implemented with aligned performances claimed in the original papers. Also, there are various quantitative evaluation settings among the hair rendering research efforts, and these settings emphasize many different aspects. We discuss six neural rendering

30

methods that are not customized for hair but representative in rendering, to explore their adaption ability and provide open-source baselines for this task. We leave the exploration of more interesting and challenging scenarios upon RenderMe-360 dataset to the community for future work.

Result. The quantitative results are shown in Table S9. We observe several interesting phenomena. 1) For methods under the NVS tracks of static hair rendering and dynamic hair rendering, their performances all show a declining trend with the increasing complexity of hair geometry. Specifically, the 'curls' scenario leads the methods to sharp performance drops under all metrics. This is reasonable, as curls data provides more challenges than the other two categories in terms of the difficulties in modeling more diverse intersections, complex motion situations, and high-frequency details. 2) NSFF and NR-NeRF remain roughly flat performances under the time-interpolation synthesis protocol. NSFF models the dynamic scene as a continuous function with the utility of a time-dependent neural scene flow field, and optimizes the function with spatial and temporal constraints. Its design help to achieve robustness in different motion interpolation scenarios. NR-NeRF has merits in dynamic reconstruction for disentangling dynamic motion into rigid and non-rigid parts. It introduces the ray-bending network to model the non-rigid motion, and a rigidity network to constrain the rigid regions. 3) From the hair motion aspect, long hair/curls scenarios contribute mostly to non-rigid deformation, whereas NSFF is superior to NR-NeRF in terms of three metrics. We infer that the deformation model of NR-NeRF has a flaw in capturing exact correspondences between images at different time steps, which leads to blur accumulated results along multiple frames. 4) In the static rendering, Instant-NGP has overall better 'PSNR' and 'SSIM' than NeuS, corresponding to the qualitative result in Figure S28 (a) , we can also observe that Instant-NGP renders hair in better high-frequency patterns. We infer that the multi-resolution data structure and individual local-part reconstruction strategy in Instant-NGP helps in fine-detail pattern reconstruction. 5) MVP performs better in all three metrics compared to NV. Whereas, these two methods show more blur reconstruction than static methods. The phenomenon suggests the efforts of dynamic field designs should also be paid to the preservation of per-frame precision, rather than only focusing on deformation to new frames.

Figure S28 (a) shows the visualization among methods under NVS track of static hair rendering and dynamic hair rendering. With the increase of the hair geometry complexity, we do not observe an obvious quality degradation of the hair rendering, while the corresponding metrics have a declining trend. We guess the main difference is on thin hair strand, which is the main challenge during hair rendering. As the complexity of the hairstyle increases, more hair strands spread out around the head (this can be discovered from the zoom-in area in the Figure), which are partially dismissed or smoothed during the rendering, causing degradation of metrics. Comparing the visualization of 4 methods, we found some method-specific characteristics. Instant-NGP [50] reconstructs the hair geometry not perfectly, but relatively well among four methods, since most of the diffusing hair strands can be reconstructed. We guess the multi-resolution data structure from NGP helps model the fine-grained geometry details. NeuS [83] produces overall correct geometry, but strongly smooths the hair. Specifically, in the 'curls' scenario, all the curly hairs are smoothed to form a general shape, which losses edge details. This is reasonable, as the SDF-based representation has advantages in modeling single-contour objects, but struggles with multiple contours objects, especially with thin structures. Neural Volume [40] produces lots of smoothness and blur, and most of the thin hair parts are dismissed, observed from the visualization. Since we feed the whole sequences with large motion into the model, it seems that Neural Volume can not handle this scenario. MVP [41] can preserve the hair details, but from all observed results, there are always artifacts surrounding the whole hair area. One possible reason is the size and quantity limitation of the volumetric primitives in the training procedure. As thin geometry, the hair parts need thousands of small primitives for high-quality representation, which requires great demands on training and is not training-friendly. A special primitive design is needed to be applied for hair rendering to improve performance.

In Figure S28 (b) we show the time-interpolation results of two methods. NSFF [35] has better performance than NR-NeRF [77] in different hairstyles. For the head motion, NSFF preserves most of the strand details regardless of the motion blur, while NR-NeRF produces more blur and artifacts in

31

the hair areas and face. The possible reason is that NSFF builds the structure correspondences among timestamps, which can be helpful for thin structure modeling. To improve the modeling capability of the deformable scenario, NR-NeRF introduces per-frame learned latent code, which may lead to smoothness and blurring with the interpolation of the latent code between two timestamps.

Table S9: **Quantitative results of hair rendering.** We study six methods for the hair rendering task under three settings. In static rendering and dynamic rendering, we evaluate the novel view synthesis result, while we render the image of the same camera view but evaluate an inter-novel time stamp in the time-interpolation part.

Aspects	Benchmarks	Short Hair			Long Hair			Curls			Over All		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Static Rendering	Instant-NGP [50]	25.53	0.848	0.274	24.99	0.834	0.29	21.06	0.789	0.355	23.75	0.822	0.309
	NeuS [83]	23.54	0.851	0.108	21.05	0.746	0.239	21.3	0.789	0.261	21.76	0.787	0.214
Dynamic Rendering	MVP [41]	24.31	0.821	0.148	22.56	0.868	0.197	20.97	0.795	0.262	22.61	0.856	0.201
	NV [40]	21.19	0.816	0.289	20.48	0.829	0.263	19.275	0.764	0.351	20.32	0.806	0.297
Time-Interpolation	NSFF [35]	27.98	0.856	0.094	28.27	0.867	0.094	28.231	0.846	0.112	28.19	0.858	0.098
	NR-NeRF [77]	27.14	0.851	0.114	27.62	0.865	0.122	27.825	0.84	0.136	27.563	0.854	0.124

4.7 Hair Editing

Table S10: **Different inversions for hair editing.** We showcase different inversion configurations for the hair editing task on the identities from our testset. **N** is short for normal cases. **H** is short for hard cases with deformable accessories.

Configuration	Split	ID-score ↑	MS-SSIM ↑	LPIPS ↓	L2 ↓	Configuration	Split	ID-score ↑	MS-SSIM ↑	LPIPS ↓	L2 ↓
e4e [75]	N	0.57	0.81	0.16	0.032	PTI [59]	N	0.88	0.95	0.06	0.003
	H	0.52	0.74	0.23	0.069		H	0.86	0.93	0.11	0.015
Restyle_e4e [1]	N	0.58	0.82	0.16	0.027	Hyperstyle [2]	N	0.81	0.90	0.07	0.012
	H	0.54	0.76	0.22	0.059		H	0.77	0.87	0.10	0.033

Table S11: **Quantitative results for hair editing.** We showcase eight configurations for the hair editing task on the normal split from the neutral expression subset of RenderMe-360.

Editing	Inversion	ID-score ↑	CLIP-score ↑	Editing	Inversion	ID-score ↑	CLIP-score ↑
HairCLIP [74]	e4e [75]	0.50	0.76	StyleCLIP [52]	e4e [75]	0.55	0.68
	Restyle_e4e [1]	0.55	0.69		Restyle_e4e [1]	0.58	0.67
	PTI [59]	0.73	0.68		PTI [59]	0.83	0.70
	Hyperstyle [2]	0.78	0.69		Hyperstyle [2]	0.80	0.69

Editing hair attributes, *e.g.,* color, hairstyle, and hair position, is an interesting but challenging task. The operations could be done in 2D [71, 92, 58] or 3D [87, 60] manner with various conditions. Here, we showcase one sub-direction – text-aware 2D hair editing, to give an example of the possible usages of our text annotation. This task refers to the setting of editing the hair attributes, given the source image and target text prompt.

Settings. For the evaluated data, we select 45 representative head images from the neutral expression subset of RenderMe-360. These images consist of 30 normal hairstyles, and 15 identities with deformable head accessories. The data samples vary from each other with distinctive attributes, such as hair color, hairdo, skin tone, and makeup. Upon the data, we present two configurations of possible ways to utilize our text annotation under the hair editing task. Concretely, we assemble two state-of-the-art text-based hair editing methods (*i.e.,* HairCLIP [74] and StyleCLIP [52]) with popular inversion strategies [75, 59, 1, 2] to form the configurations. For the first configuration, we apply HairCLIP [74], which designs specific mappers for hair color and hairstyle editing, based on text or image references. We follow the official implementation to test the capability of text-based editing after face alignment and e4e [75] inversion. For the second, third, and fourth configurations, we still focus on HairCLIP, but replace e4e [75] with other inversion methods, *i.e.,* Restyle_e4e [1], PTI [59], and Hyperstyle [2]. Since the latter three inversion strategies theoretically have better identity preserving ability. For the other four configurations, we combine another famous text-based pre-trained model StyleCLIP [52], with utilizing all the four inversion methods (e4e [75], Restyle_e4e [1], PTI [59] and HyperStyle [2]). We choose StyleCLIP's global direction style editing for adapting arbitrary text references. For the evaluation metrics, we follow the metrics used in HyperStyle [2]: identity similarity score (ID-score [11]), MS-SSIM, LPIPS, and pixel-wise L2 distance to evaluate the inversion results with the source images. For the edited images, We use

32

ID-score [11] and CLIP-score [20] to correspondingly evaluate the identity preservation ability and the similarity to text input. We first crop the original 2448×2048 images to 2048×2048 and then use the alignment code from PTI [59] to do the crop and align. For the following HairCLIP and StyleCLIP editing with different inversion methods, we use open-source pre-trained models and inference code without any further training or fine-tuning. The reference text of hairstyle and hair color basically follows the definition of HairCLIP [74]. We totally use 50 different hairstyles and 12 hair colors.

Results. Table S10 shows the quantitative inversion results. Overall, all configurations function normally with our text annotation and data samples, which demonstrates the feasibility of utilizing our data in the hair editing domain. Among the four configurations, we could observe that PTI and HyperStyle show better quantitative results than the first two. The superiority is most significant in terms of identity preservation. From the aspect of methods' effectiveness on the out-of-distribution (OOD) samples, we can observe that PTI inversion is the most robust, while the performances of other methods decrease more from normal hairstyles to images with the deformable accessory. This is reasonable as high-quality datasets for training inversion methods are typically under the shortage of complex hair accessories, *e.g.,* traditional high hats with ethnic characteristics. Additionally, the standard pre-processing requires cropped aligned faces, which often ignores partial hair and head accessories, as also been mentioned in [96]. This phenomenon reflects that there should be more research attention on the OOD problem, and the completeness regions that are associated with hair. Figure S29 shows the results of qualitative face inversion and hair manipulation on the normal split from the neutral expression subset of RenderMe-360, and Table S11 shows the quantitative results for hair manipulation. Based on the inversion results, PTI and HyperStyle can preserve more details such as face shape and hair texture compared to e4e and Restyle_e4e, which is consistent with the inversion metrics presented in Table S10. In terms of editing results, e4e+HairCLIP, which is specifically designed for hairstyle and hair color editing, performs well on both inputs. Although e4e inversion does not preserve all facial details, thanks to StyleCLIP's pre-training that follows e4e, e4e+StyleCLIP also performs well in editing most hair colors and hairstyles. When using the other three inversion methods besides e4e, HairCLIP and StyleCLIP have their respective strengths and weaknesses. For example, StyleCLIP is better at editing brown hair color and receding hairline hairstyles, while HairCLIP is better at editing black hair color and cornrow hairstyles. Restyle_e4e+HairCLIP, PTI+HairCLIP, Hyperstyle+HairCLIP may produce no change when our reference text is gray hair, and Restyle_e4e+StyleCLIP, PTI+StyleCLIP, Hyperstyle+StyleCLIP may not generate desired mohawk hairstyles. In summary, the e4e+HairCLIP model has a good effect on hair editing, but identity maintenance limited by the inversion methods which needs to be improved, which is consistent with the quantitative results shown in Table S11. On the other hand, although the inversion results of PTI and HyperStyle are superior compared with e4e and Restyle_e4e, the further text-based editing results following StyleCLIP are not equally satisfactory.

4.8 Talking Head Generation

With the phoneme-balanced corpus videos, our dataset can also serve as a standard benchmark for case-specific audio-driven talking head generation. This task refers to the setting of reenacting a specific person, with generating high-fidelity video portraits that are in sync with arbitrary speech audio as the driving source. We include two state-of-the-art talking-head methods to showcase the potential of our multi-sensory data. Previous approaches in this track mainly evaluate their performance on self-selected data. They manually extract several-minute video clips from TV programs or celebrity speeches for training and testing [19, 38, 69, 72]. Thus, there is a lack of unified selection criteria, and no benchmark agreement is achieved across different institutions yet. Additionally, some data sources (*e.g.,* YouTube videos) may suffer from license issues. We hope our attempt could provide a standard benchmark for this task.

Settings. For evaluation data, we choose two subsets that cover two languages (*i.e.,* English and Mandarin) from RenderMe-360. Each subset contains five distinctive identities, with six phoneme-balanced front-face videos per identity. Under this setting, we study two NeRF-based representative

33

baselines, namely AD-NeRF [19] and SSP-NeRF [38]. Compared with 2D generative model-based methods [24, 6, 104] and explicit 3D mesh-aware ones [72], these two methods bridge audio sources with implicit scene representation of neural radiance fields. Specifically, the two NeRF-based methods leverage pose and shape prior, along with audio information, to directly condition the semantic-aware NeRF. Such a methodology could theoretically help represent fine-scale head components (such as teeth and hair) with better photo-realistic synthesis quality. Following SSP-NeRF [38], we utilize PSNR and SSIM metrics to evaluate image quality, while landmark distance (LMD) and SyncNet confidence (Sync) [8] are used to assess the accuracy of the lip movements. Following AD-NeRF [19], we first convert videos to 450×450 resolution and we trim one second from the beginning and the end of each video to eliminate the interference from hitting board at the start and the end of recording. Then we use 90% frames for training and the remaining for testing. We process each video segment separately, and the video data for each identity has an average length of 6,018 frames at 25 fps. To obtain more accurate training data, we utilize the landmark detection model from our data processing pipeline and use the same number of corresponding landmarks at the corresponding positions. Additionally, we use our own pipeline to obtain more accurate parsing results in the face parsing step. We utilize the open-source code of AD-NeRF and the code provided by the author of SSP-NeRF for training and testing. The results we present are generated by models trained for $400k$ iterations using the corresponding official default configurations.

Table S12: **Quantitative evaluation on the talking head generation.** We benchmark AD-NeRF [19] and SSP-NeRF [38] on two subsets of RenderMe-360.

Method	Split	PSNR ↑	SSIM ↑	LMD ↓	Sync ↑	Method	Split	PSNR ↑	SSIM ↑	LMD ↓	Sync ↑
AD-NeRF [19]	English	18.44	0.83	2.29	2.75	SSP-NeRF [38]	English	18.22	0.85	1.20	3.88
	Mandarin	18.42	0.80	2.45	2.26		Mandarin	18.31	0.81	0.95	4.20

Results. Table S12 and Figure S30 present the quantitative results and qualitative illustration of talking head models. From Table S12, AD-NeRF and SSP-NeRF exhibit similar PSNR and SSIM scores, but SSP-NeRF outperforms AD-NeRF in terms of LMD and Sync confidence. This phenomenon indicates that SSP-NeRF produces more accurate mouth shapes. The inference could be further supported by the qualitative results shown in Figure S30, where SSP-NeRF's mouth shapes are closer to the ground truth. Additionally, the images generated by SSP-NeRF are clearer at the head and torso junctions. From the training language aspect, we can observe from Table S12 that, there is no significant difference between the two splits in Mandarin and English. Both methods have similar support for these languages. This reflects that even though the DeepSpeech model is used for extracting speech features that are primarily trained on non-Mandarin data, it still has good support for Mandarin due to its underlying word relationship capture ability. Moreover, the qualitative results are not ideal, if we compare models' performance to the test videos used in recent work [19, 38]. This demonstrates our dataset's potential as a new test set, uncovering more challenges for the case-specific audio-driven talking head generation.

5 Applications of RenderMe-360

There are a large number of down-streaming applications that could be enabled by our RenderMe-360 dataset, but have not been included in our current benchmark, such as 1) head generation, 2) image/video-based face reenact, and 3) cross-modal new avatar generation. Below, we demonstrate a specific task, Text to 3D Head Generation, which preliminarily reveals the broad possibilities of RenderMe-360 in abundant down-streaming applications.

5.1 Text to 3D Head

We apply our data on three typical Text to 3D Generation pipelines, ie, Dream Fields [23], Latent-NeRF [45], and TEXTure [57]. Although these methods are all general-object-centric, they are distinctive in different aspects. Specifically, Dream Fields uses NeRF to implicitly represent 3D object, and optimize the radiance fields with CLIP guidance. Latent-NeRF brings the NeRF into

34

latent space, and guides the generation with both text and proxy geometry. TEXTure requires a precise mesh alongside the text prompt, to serve as input. It leverages a pre-trained depth-to-image diffusion to iteratively inpaint the 3D model.

We select three identities from RenderMe-360 with different head characteristics. The first row in Figure S31 is the simplest sample without any makeup or extra accessories. The second row is a bit complicated, we select it from the set 'With Deformable Accessories'. The last row shows the sample in the most complicated set, in which we can see the subject has unique makeup and wears complex accessories. We use the corresponding text annotation of the samples to serve as the prompt input, which covers distinguishing descriptions of human heads in fine-grained details. We follow the original setting of the three methods, in which the scan annotation for each identity sample is used in Latent-NeRF and TexTure.

As shown in Figure S31, TEXTure can generate more reasonable results than the other two methods. The reasons are two folds. First, it only needs to learn a representation that relates to texture, and geometrically wrap the texture into a 3D mesh to generate the 3D head. Second, it uses depth-to-image diffusion, which can generate high-quality 2D head images. In contrast, Dream Fields can not produce a complete 3D head with text prompt only. Latent-NeRF can not produce fine-grained texture, although it also uses geometry prior and text prompt as TEXTure. We infer that is because it cannot well embed the text prompt into the neural implicit rendering field during training. In a nutshell, this toy example showcases several interesting suggestions for future researches on Text-to-3D-Head: 1) With the rich annotations of RenderMe-360, it is possible to generate a high-fidelity 3D head avatar corresponding to text prompts. 2) There might be a bottleneck in using text to describe complex geometry, which might be one of the reasons why current text-to-3D paradigms struggle to generate realistic human-centric 3D targets. 3) As our data annotation covers multiple modalities and dimensions, it allows the researchers to explore new paradigms with different prompt conditions.

6 Discussion

Boarder impact and limitations. The proposed RenderMe-360 dataset, together with the comprehensive benchmark, is expected to effectively facilitate modern head rendering and generation research. RenderMe-360 contains over 243 million high-fidelity video frames and their corresponding meticulous annotations. However, as the field of human head avatar is consistently blooming, we could not include all of the related research topics, and all of the state-of-the-art methods at one time. Thus, we treat the construction of benchmarks based on RenderMe-360 as a long-standing mission of our team. We will construct more and more benchmarks on different topics unflaggingly, to support the sustainable and healthy development of the related research community. Also, we will build an open platform based on RenderMe-360. We sincerely encourage and welcome contributions to RenderMe-360 from the community, to boost the development of human head avatars together.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, 2021.

[2] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing, 2021.

[3] Bruce G Baumgart. A polyhedron representation for computer vision. In *NCCE*, 1975.

[4] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *CVPR*, 2020.

[5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021.

[6] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *CVPR*, 2019.

35

[7] Wei Cheng, Su Xu, Jingtan Piao, Chen Qian, Wayne Wu, Kwan-Yee Lin, and Hongsheng Li. Generalizable neural performer: Learning robust radiance fields for human novel view synthesis. *arXiv*, 2022.

[8] J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[9] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. Capture, learning, and synthesis of 3d speaking styles. In *CVPR*, 2019.

[10] Gilles Daviet. Simple and scalable frictional contacts for thin nodal objects. *TOG*, 2020.

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[12] Paul Ekman and Wallace V Friesen. Facial action coding system. *EPNB*, 1978.

[13] Ricardo Farias, Joseph SB Mitchell, and Cláudio T Silva. Zsweep: An efficient and exact projection algorithm for unstructured volume rendering. In *SVV*, 2000.

[14] JI Fei, Chen Aiting, Zhao Yang, XI Xin, and Han Dongyi. Development of a script of phonemically balanced monosyllable lists of mandarin–chinese. *JO*, 2010.

[15] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. 2021.

[16] James D Foley. *Computer graphics: principles and practice*. 1996.

[17] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, 2021.

[18] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021.

[19] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *ICCV*, 2021.

[20] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[21] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *CVPR*, 2022.

[22] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *CVPR*, 2022.

[23] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. 2021.

[24] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. You said that?: Synthesising talking faces from audio. *IJCV*, 2019.

[25] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. In *CVPR*, 2021.

[26] Chenfanfu Jiang, Theodore F. Gast, and Joseph Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *TOG*, 2017.

[27] James T Kajiya and Timothy L Kay. Rendering fur with three dimensional textures. *SIGGRAPH*, 1989.

[28] Sing Bing Kang. Survey of image-based rendering techniques. In *VI*, 1998.

[29] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv*, 2020.

[30] Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepmvshair: Deep hair modeling from sparse views. In *SIGGRAPH Asia*, 2022.

[31] Tassilo Kugelstadt and Elmar Schömer. Position and orientation based cosserat rods. In *SIGGRAPH*, 2016.

[32] Samuli Laine and Tero Karras. Efficient sparse voxel octrees–analysis, extensions, and implementation. *NVIDIA Corporation*, 2010.

[33] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020.

[34] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *TOG*, 2017.

[35] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

36

[36] Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *WACV*, 2023.

[37] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *WACV*, 2022.

[38] Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. Semantic-aware implicit neural audio-driven video portrait generation. In *ECCV*, 2022.

[39] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *RA*, 2018.

[40] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv*, 2019.

[41] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *TOG*, 2021.

[42] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *TOG*, 2021.

[43] Stephen R Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *TOG*, 2003.

[44] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *CGIT*, 1995.

[45] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv*, 2022.

[46] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *ECCV*, 2022.

[47] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *TOG*, 2019.

[48] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[49] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications*, 2021.

[50] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 2022.

[51] Robert Osserman. *A survey of minimal surfaces*. 2013.

[52] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

[53] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*. 2003.

[54] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *CGIT*, 2000.

[55] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[57] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv*, 2023.

[58] Saha Rohit, Duke Brendan, Shkurti Florian, Taylor Graham, and Aarabi Parham. Loho: Latent optimization of hairstyles via orthogonalization. In *CVPR*, 2021.

[59] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *TOG*, 2021.

[60] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In

37

1133 *ECCV*, 2022.

[61] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *TOG*, 2004.

[62] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[63] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *VCIP*, 2000.

[64] Jyh-Shing Shyuu and Jhing-Fa Wang. An algorithm for automatic generation of mandarin phonetic balanced corpus. In *ICSLP*, 1998.

[65] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *NeurIPS*, 2019.

[66] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, 2019.

[67] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.

[68] Tiancheng Sun, Giljoo Nam, Carlos Aliaga, Christophe Hery, and Ravi Ramamoorthi. Human hair inverse rendering using multi-view photometric data. 2021.

[69] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *TOG*, 2017.

[70] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: multi-input-conditioned hair image generation for portrait editing. *TOG*, 2020.

[71] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: Multi-input-conditioned hair image generation for portrait editing. *TOG*, 2020.

[72] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, 2020.

[73] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016.

[74] Wei Tianyi, Chen Dongdong, Zhou Wenbo, Liao Jing, Tan Zhentao, Yuan Lu, Zhang Weiming, and Yu Nenghai. Hairclip: Design your hair by text and reference image. In *CVPR*, 2022.

[75] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv*, 2021.

[76] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021.

[77] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021.

[78] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020.

[79] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 2022.

[80] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *IJCV*, 2019.

[81] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *CVPR*, 2022.

[82] Daoye Wang, Prashanth Chandran, Gaspard Zoss, Derek Bradley, and Paulo Gotardo. Morf: Morphable radiance fields for multiview neural head modeling. In *SIGGRAPH*, 2022.

[83] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv*, 2021.

[84] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

38

[85] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021.

[86] WP Wang and KK Wang. Geometric modeling for swept volume of moving solids. *CGA*, 1986.

[87] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Hvh: Learning a hybrid neural volumetric representation for dynamic hair performance capture. In *CVPR*, 2022.

[88] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018.

[89] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV*, 2018.

[90] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Alexander Hypes, et al. Multiface: A dataset for neural face rendering. *arXiv*, 2022.

[91] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Citynerf: Building nerf at city scale. *arXiv*, 2021.

[92] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *TOG*, 2021.

[93] Ling-Qi Yan, Chi-Wei Tseng, Henrik Wann Jensen, and Ravi Ramamoorthi. Physically-accurate fur reflectance: modeling, measurement and rendering. *TOG*, 2015.

[94] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *CVPR*, 2020.

[95] Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. Dynamic hair modeling from monocular videos using deep neural networks. *TOG*, 2019.

[96] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Styleganex: Stylegan-based manipulation beyond cropped aligned faces. *arXiv*, 2023.

[97] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020.

[98] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *CVPR*, 2021.

[99] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

[100] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[101] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *ECCV*, 2020.

[102] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *CVPR*, 2022.

[103] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. *arXiv*, 2022.

[104] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI*, 2019.

[105] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv*, 2018.

[106] Hao Zhu, Huaibo Huang, Yi Li, Aihua Zheng, and Ran He. Arbitrary talking face generation via attentional audio-visual coherence learning. In *IJCAI*, 2020.

[107] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. In *ECCV*, 2022.

39
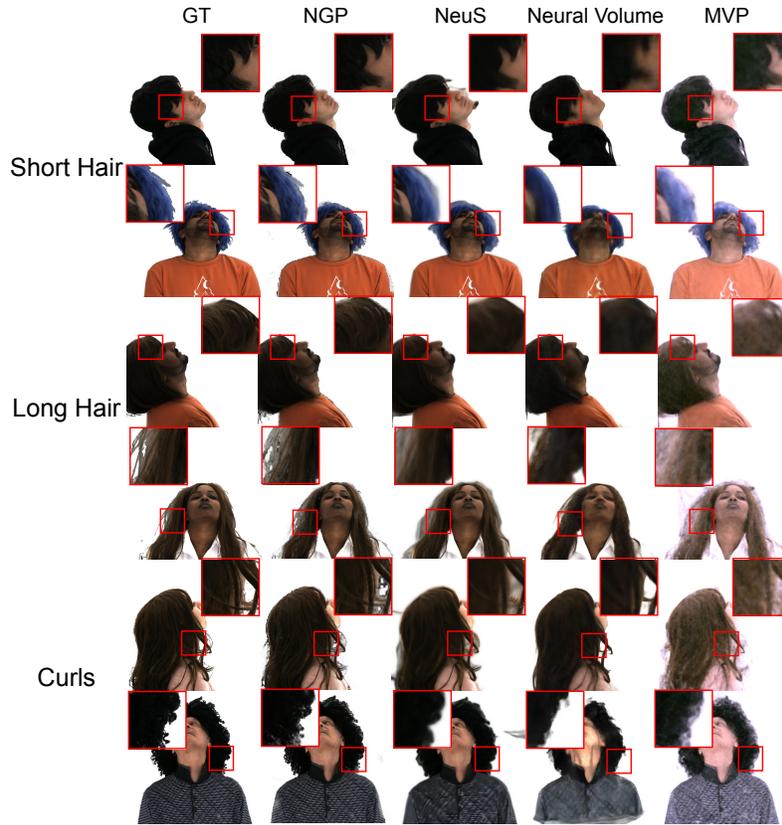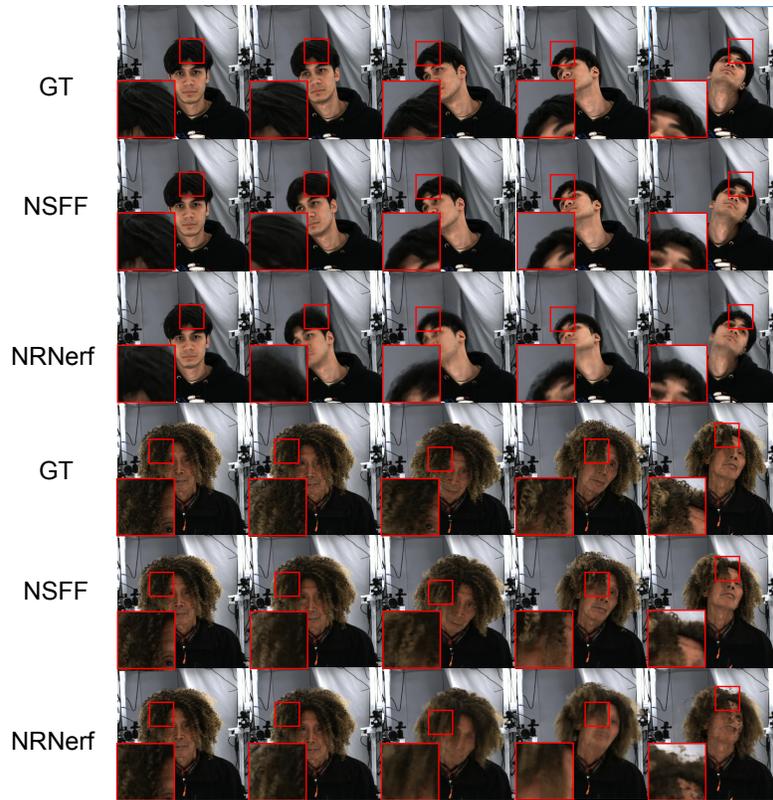
Figure S28: **Illustration of hair rendering.** (a) We show subjects in three kinds of hairstyles, and for the dynamic rendering methods (NV and MVP), we demonstrate the same frame as the static rendering methods. (b) We select keyframes of the sequence (novel inter-timestamp). Better zoom in for more details.

40
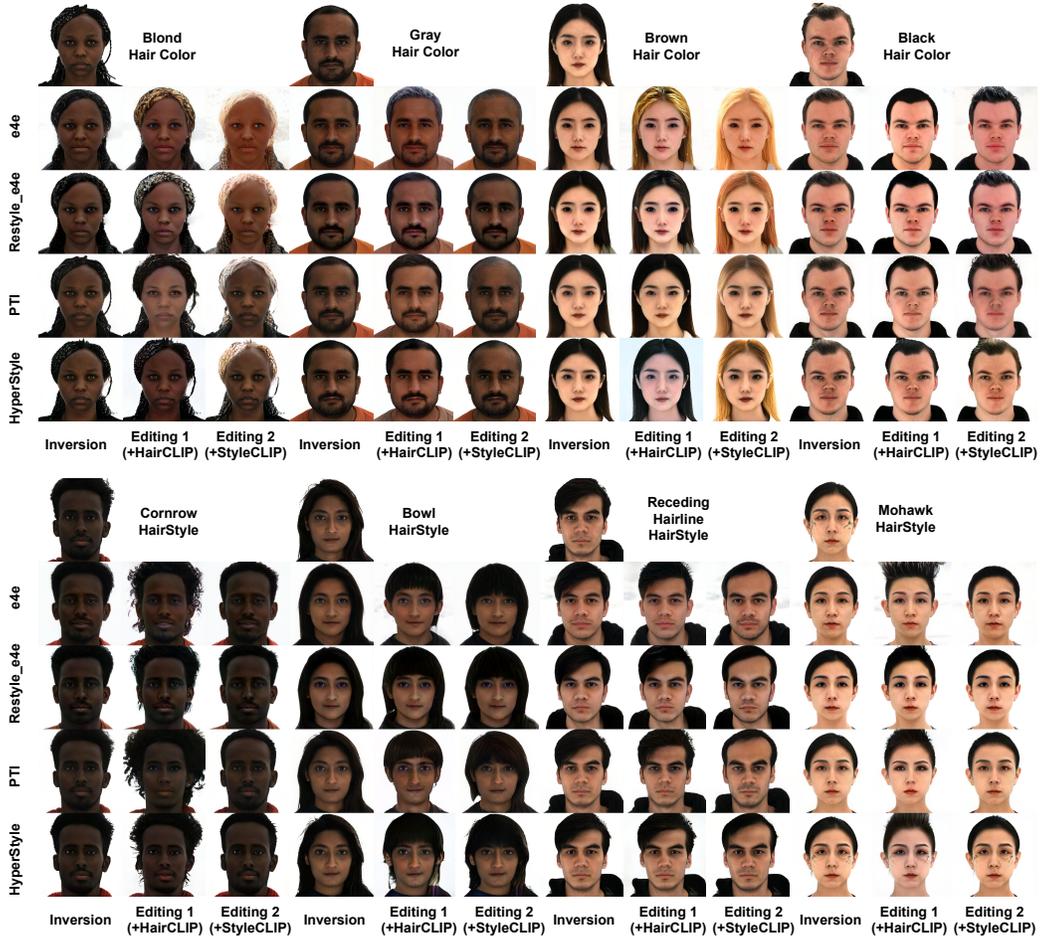
Figure S29: **Illustration of qualitative face inversion and hair editing.** For each identity, we show the aligned face, the text reference, and the combinations of face inversion and further hair manipulation. Better zoom in for more details.
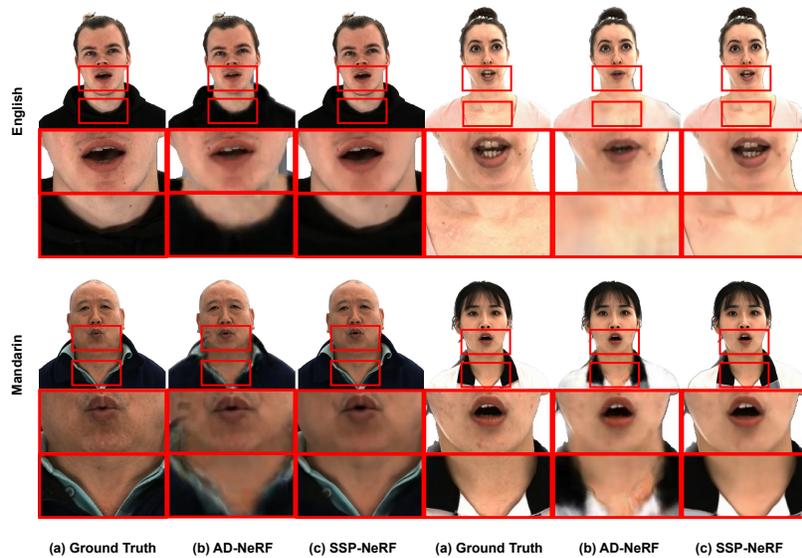
Figure S30: **Qualitative illustration of talking head generation.** We showcase results from AD-NeRF [19] and SSP-NeRF [38] on four representative samples of RenderMe360.

41
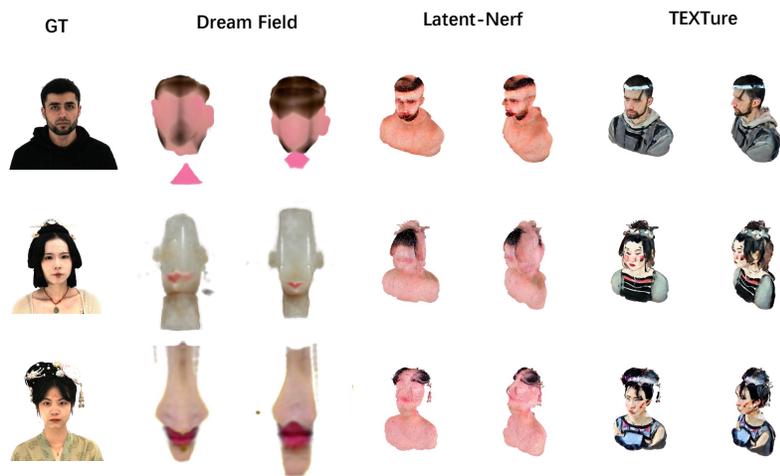
Figure S31: **Text-based application.** We select three identities and generate the result with the same text prompt, while Latent-NeRF and TEXTure additionally use the scan as geometry prior. TEXTure performs best among these three methods, and the remaining two methods are not robust in human head scenarios.

42

Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

   (b) Did you describe the limitations of your work? [Yes]

   (c) Did you discuss any potential negative societal impacts of your work? [Yes]

   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [NO] Due to the computational cost of the models, we were unable to produce error bars.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] The existing assets we use were all cited in our paper.

   (b) Did you mention the license of the assets? [Yes] Please refer to the "License" part in our online submission.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Please refer to the datasheet.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Please refer to the datasheet.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] Please refer to Section 2.2 in the Supplementary Material.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

43