NeuWigs: A Neural Dynamic Model for Volumetric Hair Capture and Animation

Ziyan Wang[1,2] Giljoo Nam[2] Tuur Stuyck[2] Stephen Lombardi[2] Chen Cao[2]
Jason Saragih[2] Michael Zollhöfer[2] Jessica Hodgins[1] Christoph Lassner[2]
[1]Carnegie Mellon University [2]Meta Reality Labs

Abstract

The capture and animation of human hair are two of the major challenges in the creation of realistic avatars for the virtual reality. Both problems are highly challenging, because hair has complex geometry and appearance and exhibits challenging motion. In this paper, we present a two-stage approach that models hair independently of the head to address these challenges in a data-driven manner. The first stage, state compression, learns a low-dimensional latent space of 3D hair states including motion and appearance via a novel autoencoder-as-a-tracker strategy. To better disentangle the hair and head in appearance learning, we employ multi-view hair segmentation masks in combination with a differentiable volumetric renderer. The second stage optimizes a novel hair dynamics model that performs temporal hair transfer based on the discovered latent codes. To enforce higher stability while driving our dynamics model, we employ the 3D point-cloud autoencoder from the compression stage for de-noising of the hair state. Our model outperforms the state of the art in novel view synthesis and is capable of creating novel hair animations without relying on hair observations as a driving signal.[†]

Figure 1. **Animation from Single View Captures.** Our model can generate realistic hair animation from single view video based on head motion and gravity direction. Original captures of subjects wearing a wig cap are shown in red boxes.

1. Introduction

The ability to model the details of human hair with high fidelity is key to achieving realism in human avatar creation: hair can be very important to one's appearance! The realism of human hair involves many aspects like geometry, appearance and interaction with light. The sheer number of hair strands leads to a very complex geometry, while the interactions between light and hair strands leads to non-trivial view-dependent appearance changes. The dynamics of hair creates another axis for evaluating the realism of a human avatar while being similarly hard to capture and model due to the complexity of the hair motion space as well as severe self-occlusions.

These problems lead to two major challenges for creating realistic avatars with hair: appearance modeling and dynamics modeling. While modern capture systems can reconstruct high fidelity hair appearance from a sparse and discrete set of real world observations frame by frame, no dynamics information is discovered through this process. To capture dynamics, we need to perform tracking to align those reconstructions in both space and time. However, doing tracking and reconstruction do not directly solve the problem of novel motion generation. To achieve that, we

[†] Project page at https://ziyanw1.github.io/neuwigs/.

need to go beyond the captured data in space and time by creating a *controllable* dynamic hair model.

In conventional animation techniques, hair geometry is created by an artist manually preparing 3D hair grooms. Motion of the 3D hair groom is created by a physics simulator where an artist selects the parameters for the simulation. This process requires expert knowledge. In contrast, data-driven methods aim to achieve hair capture and animation in an automatic way while preserving metric photo-realism. Most of the current data-driven hair capture and animation approaches learn to regress a dense 3D hair representation that is renderable directly from per-frame driving signals, without modeling dynamics.

However, there are several factors that limit the practical use of these data-driven methods for hair animation. First of all, these methods mostly rely on sophisticated driving signals, like multi-view images [22, 33], a tracked mesh of the hair [24], or tracked guide hair strands [42], which are hard to acquire. Furthermore, from an animation perspective, these models are limited to rendering hair based on hair observations and cannot be used to generate novel motion of hair. Sometimes it is not possible to record the hair driving signals at all. We might want to animate hair for a person wearing accessories or equipment that (partially) obstructs the view of their hair, for example VR glasses; or animate a novel hair style for a subject.

To address these limitations of existing data-driven hair capture and animation approaches, we present a neural dynamic model that is able to animate hair with high fidelity conditioned on head motion and relative gravity direction. By building such a dynamic model, we are able to generate hair motions by evolving an initial hair state into a future one, without relying on per-frame hair observation as a driving signal. We utilize a two-stage approach for creating this dynamic model: in the first stage, state compression, we perform dynamic hair capture by learning a hair autoencoder from multi-view video captures with an evolving tracking algorithm. Our method is capable of capturing a temporally consistent, fully renderable volumetric representation of hair from videos with both head and hair. Hair states with different time-stamps are parameterized into a semantic embedding space via the autoencoder. In the second stage, we sample temporally adjacent pairs from the semantic embedding space and learn a dynamic model that can perform the hair state transition between each state in the embedding space given the previous head motion and gravity direction. With such a dynamic model, we can perform hair state evolution and hair animation in a recurrent manner which is not driven by existing hair observations. As shown in Fig. 1, our method is capable of generating realistic hair animation with different hair styles of single view captures of a moving head with a wig cap. In summary:

- We present NeuWigs, a novel end-to-end data-driven pipeline with a volumetric autoencoder as the backbone for real human hair capture and animation, learnt from multi-view RGB images.
- We learn the hair geometry, tracking and appearance end-to-end with a novel autoencoder-as-a-tracker strategy for hair state compression, where the hair is modeled separately from the head using multi-view hair segmentation.
- We train an animatable hair dynamic model that is robust to drift using a hair state denoiser realized by the 3D autoencoder from the compression stage.

2. Related Work

We discuss related work in hair capture, data-driven hair animation and volumetric avatars.

Static Hair Capture. Reconstructing static hair is challenging due to its complex geometry. Paris *et al.* [31] and Wei *et al.* [44] reconstruct 3D hair from multi-view images and 2D orientation maps. PhotoBooth [32] further improves this technique by applying calibrated projectors and cameras with patterned lights to reconstruct finer geometry. Luo *et al.* [25] and Hu *et al.* [12] leverage hair specific structure (strand geometry) and physically simulated strands for more robust reconstruction. The state-of-the-art work on static hair capture is Nam *et al.* [29]. They present a line-based patch method for reconstructing 3D line clouds. Sun *et al.* [39] extends this approach to reconstruct both geometry and appearance with OLAT images. Rosu *et al.* [38] presents a learning framework for hair reconstruction that utilizes a prior on hair growing direction trained on synthetic data. NeuralHDHair [48] and DeepMVSHair [17] both learn to regress a hair growing field conditioned on sparse 2d observations.

Dynamic Hair Capture. Hair dynamics are hard to capture as a result of the complex motion patterns and self-occlusions. Zhang *et al.* [51] refine hair dynamics by applying physical simulation techniques to per-frame hair reconstruction. Hu *et al.* [11] invert the problem of hair tracking by directly solving for hair dynamic parameters through iterative grid search on thousands of simulation results under different settings. Xu *et al.* [49] perform hair strand tracking by extracting hair strand tracklets from spatio-temporal slices of a video volume. Liang *et al.* [21] and Yang *et al.* [50] design a learning framework fueled by synthetic hair data to regress 3D hair from video. Winberg *et al.* [45] perform dense facial hair and underlying skin tracking under quasi-rigid motion with a multi-camera system. While achieving good results on the capture of the hair geometry from dynamic sequences, those methods either do not model hair appearance with photo-realism or do not solve the problem of drivable animation.

Data-driven Hair Animation. Using physics-based simulation for hair animation is a common practice in both, academia and the film/games industry [1, 43]. However, generating hair animations with physics-based simulation can be computationally costly. To remedy this problem, reduced data-driven methods [5, 6, 9] simulate only a small portion of guide hair strands and interpolate the rest using skinning weights learned from full simulations. With the latest advances in deep learning, the efficiency of both dynamic generation [26] and rendering [4,30] of hair has been improved using neural networks. Lyu *et al.* [26] uses deep neural networks for adaptive binding between normal hair and guide hair. Olszewski *et al.* [30] treats hair rendering as an image translation problem and generates realistic rendering of hair conditioned on 2D hair masks and strokes. Similarly, Chai *et al.* [4] achieves faster rendering with photorealistic results by substituting the rendering part in the animation pipeline with screen-space neural rendering techniques. Temporal consistency is enforced in this pipeline by conditioning on hair flow. However, those methods still build on top of conventional hair simulation pipelines and use synthetic hair wigs, which require manual efforts by an artist to set up and are non-trivial to metrically evaluate. Wu *et al.* [46] propose to use a secondary motion graph (SDG) for hair animation without relying on a conventional hair simulation pipeline at runtime. However, this method is limited by artist's design of hair wigs and control of hair simulation parameters and is not able to capture or animate hair with realistic motion.

Volumetric Avatars. With the recent advent of differentiable volumetric raymarching [22, 28], many works attempt to directly build volumetric avatars from images or videos. To the best of our knowledge, Neural Volumes [23] is the earliest work that creates a volumetric head avatar from multiview images using differentiable volumetric raymarching. One of the many strengths of this work is that it directly optimizes a volume grid from multiview images while still producing high quality renders for semi-transparent objects like hair. One followup work [41] combines volumetric and coordinate-based representations into a hybrid form for better rendering quality and drivability. However, the level of detail either methods can capture is limited by the resolution of the volume grid.

Another early work on differentiable volumetric rendering is NeRF [28], which parameterizes radiance fields implicitly with MLPs instead of using a volumetric grid. Due to the success of NeRF [28] for modeling 3D scenes from multiple images, there are many works that build avatars with NeRF [7, 8, 10, 14, 16, 18, 27, 33, 34, 36, 53]. PVA [36] and KeypointNeRF [27] utilize pixel aligned information to extend NeRF's drivability and generalization over sequence data. Nerfies [33], HyperNeRF [34] and TAVA [18] optimize a deformation field together with a NeRF in a canonical space given videos. NeRFace [7], IM Avatar [53] and HeadNeRF [10] substitute the deformation field with face models like 3DMM [2] or FLAME [19] for better controllablity. However, those methods mostly assume hair to be rigidly attached to the head without motion and most of the approaches suffer from prohibitively long rendering time.

In contrast to NeRF-based avatars, a mixture of volumetric primitives (MVP) [24] builds a volumetric representation that can generate high-quality *real time* renderings that look realistic even for challenging materials like hair and clothing. Several follow up works extend it to model body dynamics [37], moderate hair dynamics [42] and even for in-the-wild captures [3]. However, animating such volumetric representations with dynamics is still an unsolved problem.

3. Method

Our method for hair performance capture and animation consists of two stages: state compression and dynamic modeling (see also Fig. 2). The goal of the first stage is to perform dynamic hair capture from multi-view video of a head in motion. To be more specific, in this stage, we aim to distill a 3D renderable representation of hair from multi-view images at each frame into an embedding space. To achieve that, we train a volumetric autoencoder in a self-supervised manner to model the hair geometry, tracking and appearance. The output of this model is a set of tracked hair point clouds p_t, their corresponding local radiance fields in the form of volumetric primitives $\boldsymbol{V}_t$ and a compact embedding space that is spanned by the 1D hair state encoding $\boldsymbol{z}_t$. In the second stage, we perform modeling of hair dynamics based on the hair capture from the first stage. The goal of this stage is to create a controllable, self-evolving representation of hair without relying on online observations of hair. We achieve this by learning a neural network (MLP) to regress the next possible hair state, which is conditioned on the previous hair state as well as the previous head motion and head-relative gravity direction. Equipped with the hair encoding space acquired from the first stage, we can train that model in a supervised manner by simply sampling data pairs of temporally adjacent hair states from the encoding space. Using both stages, we can perform dynamic hair animation at test time given an initialization using a recurrent strategy, without relying on direct observations of hair as a per-frame driving signal.

3.1. State Compression

We assume that multi-view image captures I_{cam_i} with their corresponding calibrated cameras are given. We denote the extrinsics of each camera i as $\mathbf{R}_i$ and $\mathbf{t}_i$. We then run l-MVS [29] and non-rigid tracking [47] to obtain the per-frame hair reconstructions $p_t \in \mathbb{R}^{N_{p_t} \times 3}$ and head tracked vertices $x_t \in \mathbb{R}^{N_{x_t} \times 3}$ at time frame t, where N_{p_t} and N_{x_t} denote the size of each. The p_t and x_t together serve as a coarse representation for the hair and head. Dif-

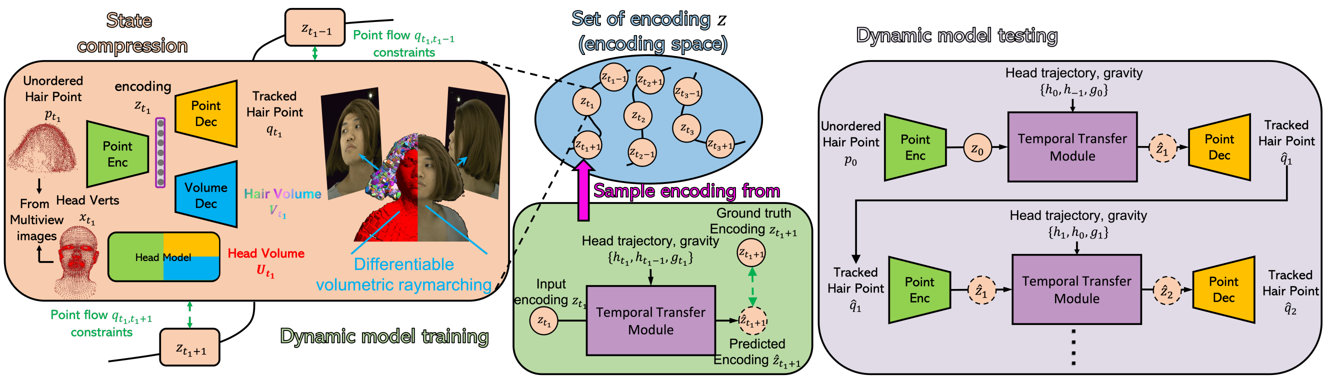

Figure 2. **Method Overview.** Our method is comprised of two stages: state compression and dynamic modeling. In the first stage, we train an autoencoder for hair and head appearance from multiview RGB images using differentiable volumetric raymarching; at the same time we create an encoding space of hair states. In the dynamic modeling stage, we sample temporally adjacent hair encodings to train a temporal transfer module ($\mathtt{T^2M}$) that performs the transfer between the two, based on head motion and head-relative gravity direction.

ferent from the head vertices, here the p_t represent an unordered set of hair point clouds. Due to the difference between hair and head dynamic patterns, we model them separately by training two different volumetric autoencoders. For the head model, we use an autoencoder to regress the volumetric texture in an unwrapped UV layout conditioned on the tracked head mesh, similar to [24, 42]. For the hair model, we optimize the hair volumetric texture and its tracking simultaneously. To better enforce the disentanglement between hair and head, we attach head volumes only to the head mesh and hair volumes only to the hair point clouds. Moreover, we use a segmentation loss to constrain each of them to only model the texture of their assigned category of hair or head.

Autoencoder as a Tracker. Learning to track hair in a supervised manner with manual annotation is infeasible. We automatically discover the hair keypoints as well as their tracking information by optimizing a variational autoencoder (VAE) [15] in a semi-supervised manner. By doing autoencoding on hair point clouds, we find that the VAE representing the hair point cloud can automatically align hair shapes along the temporal axis and is capable of tracking through both long and discontinuous hair video segments such as captures of different hair motions.

The input to the point encoder $\mathcal{E}$ is the point coordinates of the unordered hair point cloud p_t. Given its innate randomness in terms of point coverage and order, we use PointNet [35] to extract the corresponding encoding $\boldsymbol{z_t} \in \mathbb{R}^{256}$. Besides being agnostic to the order of p_t, it also can process varying numbers of points and aggregate global information from the input point cloud. The point decoder $\mathcal{D}$ is a simple MLP that regresses the coordinate and point tangential direction of the tracked point cloud $q_t \in \mathbb{R}^{N_{prim} \times 3}$, $\boldsymbol{dir}(q_t) \in \mathbb{R}^{N_{prim} \times 3}$ and $s_t \in \mathbb{R}^{N_{prim}}$ from $\boldsymbol{z_t}$, where N_{prim} is the number of tracked hair points. We denote $\boldsymbol{dir}(x)$ as the tangential direction of x. s_t is a per-point scale factor. We optimize the following loss to train the point autoencoder:

$$\mathcal{L}_{geo} = \mathcal{L}_{cham} + \omega_{temp}\mathcal{L}_{temp} + \omega_{KL}\mathcal{L}_{KL}.$$

The first term is the Chamfer distance loss which aims to align the shape of tracked point cloud q_t to p_t:

$$\begin{aligned}\mathcal{L}_{cham} = &||q_t - \mathtt{N}_{q_t,p_t}||_2 - \boldsymbol{cos}(\boldsymbol{dir}(q_t), \boldsymbol{dir}(\mathtt{N}_{q_t,p_t})) \\ &+||p_t - \mathtt{N}_{q_t,p_t}||_2 - \boldsymbol{cos}(\boldsymbol{dir}(p_t), \boldsymbol{dir}(\mathtt{N}_{q_t,p_t})),\end{aligned}$$

where $\boldsymbol{cos}(\cdot,\cdot)$ is the cosine similarity and $\mathtt{N}_{x,y} \in \mathbb{R}^{N_x \times 3}$ are the coordinates of the nearest neighbor of each point of x in y. To further enforce temporal smoothness, we use point flow $\overrightarrow{fl}(p_t)$ and $\overleftarrow{fl}(p_t)$ denoting forward and backward flow from p_t to p_{t+1} and p_{t-1} as additional supervision and formulate $\mathcal{L}_{temp}$ as follows:

$$\begin{aligned}\mathcal{L}_{temp} = &||\overleftarrow{fl}(q_t) - \overleftarrow{fl}(\mathtt{N}_{q_t,p_t})||_2 + ||\overleftarrow{fl}(p_t) - \overleftarrow{fl}(\mathtt{N}_{p_t,q_t})||_2 \\ &+||\overrightarrow{fl}(q_t) - \overrightarrow{fl}(\mathtt{N}_{q_t,p_t})||_2 + ||\overrightarrow{fl}(p_t) - \overrightarrow{fl}(\mathtt{N}_{p_t,q_t})||_2,\end{aligned}$$

where as q_t is the tracked point, we can simply have $\overleftarrow{fl}(q_t) = q_t - q_{t-1}$ and $\overrightarrow{fl}(q_t) = q_t - q_{t+1}$. Please see the supplemental materials for how we estimate $\overleftarrow{fl}(p_t)$ and $\overleftarrow{fl}(\mathtt{N}_{q_t,p_t})$. The last term $\mathcal{L}_{KL}$ is the KL-divergence loss [15] on the encoding $\boldsymbol{z_t}$ to enforce similarity with a normal distribution $\mathcal{N}(0,1)$.

Hair Volumetric Decoder. In parallel to the point decoder $\mathcal{D}$, we optimize a hair volumetric decoder that regresses a volumetric radiance field around each of the hair points. The hair volumetric primitives $\boldsymbol{V_t} \in \mathbb{R}^{N_{prim} \times 4 \times m^3}$ store RGB and alpha in resolution of m^3. We use a decoder similar to HVH [42] to regress the volume payload. The pose of each volumetric primitive is directly determined by the output of the point decoder: q_t and $\boldsymbol{dir}(q_t)$. We denote $\boldsymbol{R}_t^{p,n} \in SO(3)$ and $\boldsymbol{d}_t^{p,n} \in \mathbb{R}^3$ as the $\boldsymbol{n}$th volume-to-world rotation and translation of hair volume and per-hair-volume scale $s_t^{p,n}$ as the $\boldsymbol{n}$th element in scale $\boldsymbol{s_t}$. Similarly, $\boldsymbol{d}_t^n = q_t^n$ is the $\boldsymbol{n}$th element of q_t. Given the head center $\boldsymbol{x}_t^c$ extracted from the head vertices $\boldsymbol{x}_t$ and hair head direction as $\bar{\boldsymbol{h}}_t^n = q_t^n - \boldsymbol{x}_t^c$, we formulate the rotation $\boldsymbol{R}_t^{p,n}$ as $\boldsymbol{R}_t^{p,n} = [\boldsymbol{l}(q_t^n), \boldsymbol{l}(q_t^n \times \bar{\boldsymbol{h}}_t^n), \boldsymbol{l}(q_t^n \times (q_t^n \times \bar{\boldsymbol{h}}_t^n))]^T$, where $\boldsymbol{l}(x) = x/||x||_2$ is the normalization function. The output of the head model is similar to the hair except that it is modeling head related (non-hair) regions. We denote the head volume payload as $\boldsymbol{U}_t \in \mathbb{R}^{N_{prim} \times 3 \times m^3}$ and head related rotation $\boldsymbol{R}_t^{x,n} \in SO(3)$, translation $\boldsymbol{d}_t^{x,n} \in \mathbb{R}^3$ and scale $\boldsymbol{s}_t^{x,n} \in \mathbb{R}$.

Differentiable Volumetric Raymarching. Given all volume rotations $\boldsymbol{R}_t^{all} = [\boldsymbol{R}_t^{x,n}, \boldsymbol{R}_t^{p,n}]$, translations $\boldsymbol{d}_t^{all} = [\boldsymbol{d}_t^{x,n}, \boldsymbol{d}_t^{p,n}]$, scales $\boldsymbol{s}_t^{all} = [\boldsymbol{s}_t^{x,n}, \boldsymbol{s}_t^{p,n}]$ and local radiance fields $\boldsymbol{V}_t^{all} = [\boldsymbol{U}_t, \boldsymbol{V}_t]$, we can render them into image $\mathcal{I}_{cam_i}$ and compare it with I_{cam_i} to optimize all volumes. Using an optimized BVH implementation similar to MVP [24], we can efficiently determine how each ray intersects with each volume. We define a ray as $\boldsymbol{r}(\mathfrak{p}, l) = \boldsymbol{o}(\mathfrak{p}) + l\boldsymbol{v}(\mathfrak{p})$ shooting from pixel $\mathfrak{p}$ in direction of $\boldsymbol{v}(\mathfrak{p})$ with a depth l in range of (l_{min}, l_{max}). The differentiable formation of an image given the volumes can then be formulated as below:

$$\mathcal{I}_{\mathfrak{p}} = \int_{l_{min}}^{l_{max}} \mathbf{V}_{t,rgb}^{all}(\mathbf{r}_{\mathfrak{p}}(l)) \frac{dT(l)}{dl} dl,$$
$$T(l) = min(\int_{l_{min}}^{l} \mathbf{V}_{t,\alpha}^{all}(\mathbf{r}_{\mathfrak{p}}(l)) dl, 1),$$

where $\boldsymbol{V}_{t,rgb}^{all}$ is the RGB part of $\boldsymbol{V}_t^{all}$ and $\boldsymbol{V}_{t,\alpha}^{all}$ is the alpha part of $\boldsymbol{V}_t^{all}$. To get the full rendering, we composite the rendered image as $\tilde{\mathcal{I}}_{\mathfrak{p}} = \mathcal{I}_{\mathfrak{p}} + (1 - \mathcal{A}_{\mathfrak{p}}) I_{\mathfrak{p},bg}$ where $\mathcal{A}_{\mathfrak{p}} = T(l_{max})$ and $I_{\mathfrak{p},bg}$ is the background image. We optimize the following loss to train the volume decoder:

$$\mathcal{L}_{pho} = ||\tilde{\mathcal{I}}_{\mathfrak{p}} - I_{\mathfrak{p},gt}||_1 + \omega_{VGG} \mathcal{L}_{VGG}(\tilde{\mathcal{I}}_{\mathfrak{p}}, I_{\mathfrak{p},gt}),$$

where $\mathcal{L}_{VGG}$ is the perceptual loss in [13] and $I_{\mathfrak{p},gt}$ is the ground truth pixel value of $\mathfrak{p}$. We find that the usage of a perceptual loss yields more salient rendering results.

However, as we optimize both $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ from the images, texture bleeding between the hair volume $\boldsymbol{V}_t$ and the head volume $\boldsymbol{U}_t$ becomes a problem. The texture bleeding issue is especially undesirable when we want to treat the hair and head separately, for example, when we want to animate new hair. To prevent this, we additionally render a hair mask map to regularize both $\boldsymbol{V}_{t,\alpha}$ and $\boldsymbol{U}_{t,\alpha}$. We denote the ground truth hair mask as $M_{\mathfrak{p},gt}$ and the rendered hair mask as $\mathcal{M}_{\mathfrak{p}}$:

$$\mathcal{M}_{\mathfrak{p}} = \int_{l_{min}}^{l_{max}} \mathbf{V}_{t,\mathbb{1}}^{all}(\mathbf{r}_{\mathfrak{p}}(l)) \frac{dT(l)}{dl} dl,$$
$$T(l) = min(\int_{l_{min}}^{l} \mathbf{V}_{t,\alpha}^{all}(\mathbf{r}_{\mathfrak{p}}(l)) dl, 1),$$

where $\mathbf{V}_{t,\mathbb{1}}^{all}$ is all one volume if it belongs to $\boldsymbol{V}_t$ otherwise zero. We formulate the segmentation loss as $\mathcal{L}_{mask} = ||\mathcal{M}_{\mathfrak{p}} - M_{\mathfrak{p},gt}||_1$. The final objective for training the whole autoencoder is $\mathcal{L} = \mathcal{L}_{geo} + \mathcal{L}_{pho} + \omega_{mask} \mathcal{L}_{mask}$.

3.2. Dynamic Model

In the second stage, we aim to build a temporal transfer module ($\mathtt{T^2M}$) of hair dynamic priors that can evolve hair states over time without relying on per-frame hair observation as a driving signal. In a high level, the design of $\mathtt{T^2M}$ is similar to a hair simulator except that it is fully data-driven. To acquire such a model, we sample data pairs of adjacent time frames $(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t)$ from the embedding space of hair states built at the state compression stage and train $\mathtt{T^2M}$ to perform the hair state transfer in a supervised manner. To be more specific, we use a MLP as the backbone for $\mathtt{T^2M}$. One of the inputs to $\mathtt{T^2M}$ is the encoding $\boldsymbol{z}_{t-1}$ of the previous time step. At the same time, $\mathtt{T^2M}$ is also conditioned on the head per-vertex displacement $\boldsymbol{h}_t = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$ and $\boldsymbol{h}_{t-1} = \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}$ from the previous two time steps as well as head relative gravity direction $\boldsymbol{g}_t \in \mathbb{R}^3$ at the current time step. $\mathtt{T^2M}$ will then predict the next possible state $\hat{\boldsymbol{z}}_t$ from $\mathtt{T^2M}$ based on those inputs. Similar to the design of the VAE, the output of $\mathtt{T^2M}$ is a distribution instead of a single vector. Thus $\mathtt{T^2M}$ predicts both the mean and standard deviation of $\hat{\boldsymbol{z}}_t$ as $\mu(\hat{\boldsymbol{z}}_t), \delta(\hat{\boldsymbol{z}}_t) = \mathtt{T^2M}(\boldsymbol{z}_{t-1} | \boldsymbol{h}_t, \boldsymbol{h}_{t-1}, \boldsymbol{g}_t)$. During training, we take $\hat{\boldsymbol{z}}_t = \mu(\hat{\boldsymbol{z}}_t) + \mathfrak{n} \odot \delta(\hat{\boldsymbol{z}}_t)$ and during testing $\hat{\boldsymbol{z}}_t = \mu(\hat{\boldsymbol{z}}_t)$. The per point normal distribution vector, $\mathfrak{n}$, is the same shape as $\delta(\hat{\boldsymbol{z}}_t)$ and $\odot$ is the element-wise multiplication.

Training Objectives. We denote the point encoder as $\mathcal{E}(\cdot)$ and the point decoder as $\mathcal{D}(\cdot)$. Across the training of $\mathtt{T^2M}$, we freeze the parameters of both, $\mathcal{E}$ and $\mathcal{D}$. We denote the unordered point cloud at frame t as p_t, its corresponding encoding as $\mu(\boldsymbol{z}_t), \delta(\boldsymbol{z}_t) = \mathcal{E}(p_t)$ and the tracked point cloud as $q_t = \mathcal{D}(\boldsymbol{z}_t)$. The following loss enforces the prediction of $\mathtt{T^2M}$ to be similar to its ground truth:

$$\mathcal{L}_{mse} = ||\mu(\hat{\boldsymbol{z}}_{t+1}) - \mu(\boldsymbol{z}_{t+1})||_2 + ||\delta(\hat{\boldsymbol{z}}_{t+1}) - \delta(\boldsymbol{z}_{t+1})||_2$$
$$\mathcal{L}_{cos} = -\boldsymbol{cos}(\mu\hat{\boldsymbol{z}}_{t+1}, \mu\boldsymbol{z}_{t+1}) - \boldsymbol{cos}(\delta\hat{\boldsymbol{z}}_{t+1}, \delta\boldsymbol{z}_{t+1})$$
$$\mathcal{L}_{ptsmse} = ||\mathcal{D}(\hat{\boldsymbol{z}}_{t+1}) - \mathcal{D}(\boldsymbol{z}_{t+1})||_2,$$

where we not only minimize the ℓ_2 distance between $\hat{\boldsymbol{z}}_{t+1}$ and $\boldsymbol{z}_{t+1}$, but also enforce the cosine similarity and the corresponding tracked point cloud to be equivalent. To adapt $\mathtt{T^2M}$ to the tracked point cloud q_t, we compute the above loss again but using q_t as input, where we generate the corresponding encoding as z'_t from $\mathcal{E}(q_t)$ and its prediction $\hat{z}'_{t+1}$ from $\mathtt{T^2M}(z'_t | h_t, h_{t-1}, g_t)$:

$$\mathcal{L}_{mse,cyc} = ||\mu(\hat{z}'_{t+1}) - \mu(z_{t+1})||_2 + ||\delta(\hat{z}'_{t+1}) - \delta(z_{t+1})||_2$$
$$\mathcal{L}_{cos,cyc} = -\boldsymbol{cos}(\mu\hat{z}'_{t+1}, \mu z_{t+1}) - \boldsymbol{cos}(\delta\hat{z}'_{t+1}, \delta z_{t+1})$$
$$\mathcal{L}_{ptsmse,cyc} = ||\mathcal{D}(\hat{z}'_{t+1}) - \mathcal{D}(z_{t+1})||_2.$$

Similar to how we train our autoencoder, we also enforce two KL divergence losses on both the predicted $\hat{z}_{t+1}$ and $\hat{z}'_{t+1}$ with a normal distribution $\mathcal{N}$. The final objective for training the $\mathtt{T^2M}$ is a weighted sum of the above eight terms.

Animation. Given an initialized hair state, our dynamic model $\mathtt{T^2M}$ can evolve the hair state into future states conditioned on head motion and head-relative gravity direction. One straightforward implementation of the $\mathtt{T^2M}$ is to directly propagate the hair state encoding z_t. However, in practice, we find this design leads to severe drift in the semantic space. As a simple feed forward neural network, $\mathtt{T^2M}$ can not guarantee that its output is noise free. The noise in the output becomes even more problematic when we use $\mathtt{T^2M}$ in a recurrent manner, where the output noise will aggregate and lead to drift. To remedy this, instead of propagating the encoding z_t directly, we reproject the predicted encoding z_t by the point autoencoder $\mathcal{E}$ and $\mathcal{D}$ every time for denoising. To be more specific, we acquire the de-noised predicted hair encoding $\hat{z}_{t+1} = \mathcal{E}(\mathcal{D}(\hat{z}_{t+1}))$ from the raw prediction $\hat{z}_{t+1}$ of $\mathtt{T^2M}$. The use of the point autoencoder $\mathcal{E}$ and $\mathcal{D}$ can help us remove the noise in z_{t+1}, as the point cloud encoder $\mathcal{E}$ can regress the mean μ_{t+1} and standard deviation δ_{t+1} of z_{t+1} separately by using q_{t+1} as an intermediate variable. Thus, we can extract the noise free part of z_{t+1} by taking the mean μ_{t+1} regressed from $\mathcal{E}$. Please see our experiments for further details of this approach.

4. Experiments

In order to test the proposed model, we conduct experiments on both, the hair motion dataset presented in HVH [42] and our own dataset with longer sequences following a similar capture protocol as HVH. We collect a total of four different hair wig styles with scripted head motions like nodding, swinging and tilting. We also collect an animation test set with the same scripted head motions performed by different actors wearing a wig cap, which we will refer to as "bald head motion sequence". The animation test set contains both, single view captures from a smart phone and multiview captures. The total length of each hair wig capture is around 1-1.5 minutes with a frame rate of 30Hz. A hundred cameras are used during the capture where 93 of them are used to obtain training views and the rest are providing held-out test views. We split each sequence into two folds with similar amounts of frames and train our model exclusively on the training portion of each sequence. Please see the supplemental page[‡] for all video results and comparisons.

	seq01				seq02				seq03			
	MSE↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	PSNR↑	SSIM↑	LPIPS↓
PFNeRF	51.25	31.16	0.9269	0.3717	103.41	28.15	0.8659	0.5067	76.59	29.50	0.9000	0.2949
NSFF	50.13	31.21	0.9346	0.3672	90.06	28.75	0.8885	0.4728	83.18	29.10	0.8936	0.3292
NRNeRF	56.78	30.78	0.9231	0.3554	132.16	27.13	0.8549	0.5241	79.83	29.33	0.8987	0.3067
MVP	47.54	31.60	0.9476	0.2587	77.23	29.62	0.9088	0.3051	73.78	29.66	0.9224	0.2455
HVH	41.89	32.17	0.9543	0.2019	59.84	30.69	0.9275	0.2353	71.58	29.81	0.9314	0.2021
Ours	40.34	32.28	0.9558	0.1299	56.47	30.94	0.9329	0.1254	73.65	29.69	0.9247	0.1496

Table 1. **Novel view synthesis.** We compute MSE↓, PSNR↑, SSIM↑ and LPIPS↓ comparing rendered and ground truth images on hold-out views. First and second best results are highlighted.

4.1. Evaluation of the State Compression Model

We first test our state compression model to evaluate its ability to reconstruct the appearance of hair and head.

Novel View Synthesis. We compare with volumetric methods including NeRF based methods [20, 40] and volumetric primitives based methods [24, 42] on the dataset from HVH. In Tab. 1, we show the reconstruction related metrics MSE, SSIM, PSNR and LPIPS [52] between predicted images and the ground truth images on hold out views. Our method shows a good balance between perceptual and reconstruction losses while keeping both of them relatively low. Furthermore, our method achieves a much higher perceptual similarity with ground truth images. In Fig. 3 we show that our method can capture high frequency details and even preserve some fly-away hair strands.

Ablation on Hair/Head Disentanglement. We test how well our model handles hair/head disentanglement compared to previous work. As hair and head are exhibiting different dynamic patterns, disentanglement is usually required, especially to achieve independent controllability for both. We compare with HVH, which implicitly separates the hair and head using the dynamic discrepancy between them and optical flow. In our model, we further facilitate the disentanglement by using semantic segmentation. We visualize the difference in Fig. 4. Our method generates a more opaque hair texture with less texture bleeding between hair volumes and non-hair volumes. Moreover, our model creates the hair shape in an entirely data-driven fashion, which yields higher fidelity results than the artist prepared hair in HVH.

Ablation on $\mathcal{L}_{VGG}$. We examine the synergy between $\mathcal{L}_{VGG}$ and the ℓ_1 loss for improving the rendering quality. As shown in Tab. 2, we find that the perceptual loss has positive effects on the reconstruction performance while the improvements are negated when the weight is too large. In Fig. 5, we compare the rendered images using different $\mathcal{L}_{VGG}$ weights. The results are more blurry when not using $\mathcal{L}_{VGG}$ and fewer details are reconstructed, for example fly-away strands are not visible.

[‡] https://ziyanw1.github.io/neuwigs/resources

Figure 3. **Novel View Synthesis.** Compared with previous methods our method captures hair with more details, including fly-away hair strands, and creates an overall more accurate hair reconstruction with perceptually better rendering results.

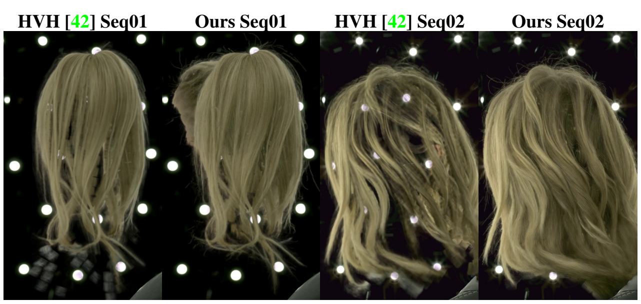

Figure 4. **Hair/Head Disentanglement.** By explicitly enforcing the semantic segmentation of head and hair through additional supervision, we learn a more opaque hair texture while the result suffers less from texture bleeding.

	vgg=0.0	vgg=0.1	vgg=0.3	vgg=1.0	vgg=3.0	vgg=10.0
MSE	42.84	42.40	39.94	40.34	40.98	42.82
PSNR	32.04	32.09	32.34	32.28	32.25	32.07
SSIM	0.9544	0.9564	0.9576	0.9558	0.9541	0.9518
LPIPS	0.2021	0.1765	0.1511	0.1299	0.1238	0.1257

Table 2. **Ablation on $\mathcal{L}_{VGG}$.** We find using an additional complementary perceptual loss leads to better appearance reconstruction.

Ablation on Point Flow Supervision. Even though with $\mathcal{L}_{cham}$ we can can already optimize a reasonably tracked point cloud p_t, we find that point flow can help remove the jittering in appearance. We show the temporal smoothness enforced by the point flow supervision in Fig. 6. Our model learns a more consistent hair texture with less jittering when trained with point flow.

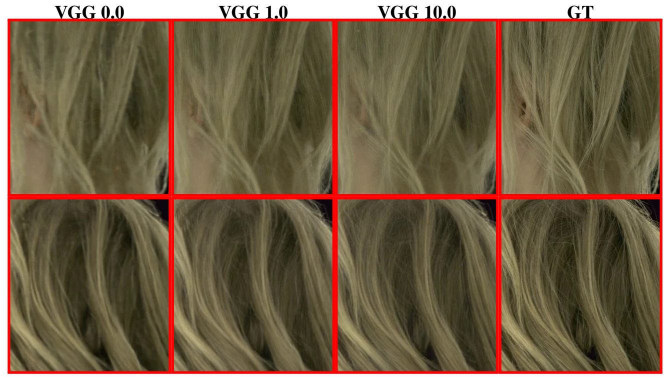

Figure 5. **Ablation on $\mathcal{L}_{VGG}$.** Adding a perceptual loss leads to sharper reconstruction results.

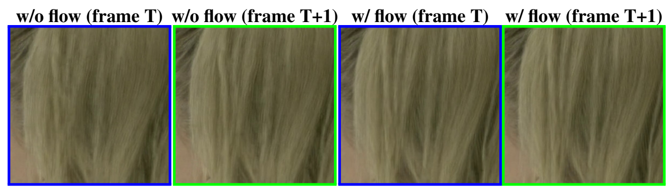

Figure 6. **Ablation on Point Flow.** We find that adding point flow to regularize the offsets between temporally adjacent tracked points prevents jittering.

4.2. Evaluation of the Dynamic Model

Lastly, we perform tests of our animation model. Compared to a per-frame model that takes hair observations as input, the input to our dynamic model and any of its variations is a subset of the head motion trajectory and hair point cloud at the initialization frame. As a quantitative evaluation, we compare our model with per-frame driven models using either hair observations or head observations as driving signals. For qualitative evaluation, we render new hair animations for the bald head motion sequences.

Quantitative Test of the Dynamic Model. We evaluate our dynamic model on the test sequences of scripted hair motion capture. The goal is to test whether our dynamic model generates reasonable novel content rather than only testing how well it reconstructs the test sequence. To test the performance of our dynamic model, we treat the model driven by per-frame (**pf**) hair observations as an oracle paradigm to compare with, while our dynamic model does not use any per-frame hair observation as a driving signal. In Tab. 3, we compare the rendering quality of our dynamic model with **pf** models and ablate several designs. The best performing dynamic model **(dyn)** has similar performance with the **pf** model even without the per-frame hair observation as driving signal. We find that both adding a cosine similarity

	MSE(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	ChamDis(↓)
pf w/ hair img	**37.36**	**32.94**	**0.9559**	**0.1333**	**10.47**
dyn w/o cos	44.96	32.26	0.9458	0.1327	25.49
dyn w/o cyc	45.22	32.23	0.9453	0.1335	26.79
dyn w/o grav	40.12	32.64	0.9504	0.1268	13.76
dyn	**38.49**	**32.80**	**0.9532**	**0.1211**	**11.12**

Table 3. **Ablation of Different Dynamic Models.** We compare different models in terms of rendering quality and tracking accuracy.

loss as an additional objective to MSE and adding a cycle consistency loss helps improve the stability of the dynamic model. Meanwhile, we find adding gravity as an auxiliary input stabilizes our model on slow motions, presumably because during slow motions of the head, the hair motion is primarily driven by gravity.

Ablation for the Point Autoencoder $\mathcal{E} + \mathcal{D}$. Here, we analyze how well the point autoencoder acts as a stabilizer for the dynamic model. We compare two different models in Fig. 7: encprop, that propagates the encoding directly, and ptsprop that propagates the regressed hair point cloud and generates the corresponding encoding from the point encoder. The results are more salient with ptsprop. The improvement of the ptsprop over the encprop is partially because the mapping from point cloud to encoding is an injective mapping and the point encoder serves as a noise canceller in the encoding space. To further study this behavior, we perform a cycle test on the point encoder, where we add noise n to a certain encoding z and get a noisy version of the encoding $\hat{z}=z+n$ and its corresponding noisy point cloud $\hat{p}$. Then, we predict a cycled encoding $\bar{z}=\mathcal{E}(\mathcal{D}(\hat{z}))$. We compare z, $\hat{z}$ and $\bar{z}$ in Fig. 7. z and $\bar{z}$ are consistently close while $\hat{z}$ jitters. This result suggests that the remapping of $\hat{z}$ using $\mathcal{E}$ and $\mathcal{D}$ counteracts the noise n.

Animation on Bald Head Sequences. We show animation results driven by head motions on in-the-wild phone video captures in Fig 1. We find our model generates reasonable motions of hair under natural head motions like swinging or nodding.

5. Discussion

We present a two-stage data-driven pipeline for volumetric hair capture and animation. The core of our method is a 3D volumetric autoencoder that we find useful for both automatic hair state acquisition and stable hair dynamic generation. The first stage of our pipeline simultaneously performs hair tracking, geometry and appearance reconstruction in a self-supervised manner via an autoencoder-as-a-tracker strategy. The second stage leverages the hair states acquired from the first stage and creates a recurrent hair dynamics model that is robust to moderate drift with the autoencoder as a denoiser. We empirically show that our method performs stable tracking of hair on long and segmented video captures while preserving high fidelity for hair appearance. Our model also supports generating new animations in both lab- and in-the-wild conditions and does not rely on hair observations.

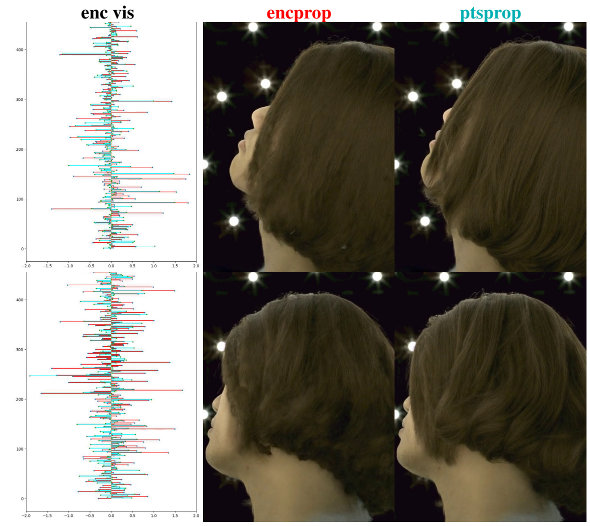

Figure 7. **encprop v.s. ptsprop.** ptsprop generates sharper results with less drifting than encprop.

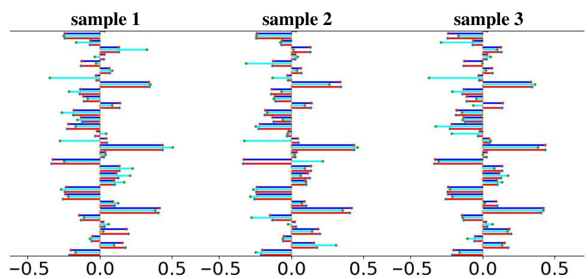

Figure 8. **Point Encoder $\mathcal{E}$ as a Stabilizer.** We sample several $\hat{z}$ and corresponding $\bar{z}=\mathcal{E}(\mathcal{D}(\hat{z}))$ with a fixed z and visualize part of them above. As we can see, $\bar{z}$ stays similar to z while $\hat{z}$ jitters.

Limitations. Like many other data-driven methods, our method requires a large amount of diverse training data and might fail in scenarios that are far from the training distribution. For example, our model might not work well with motions like random head shakes or head motions under a different gravity. Using a physics prior with the aid of explicit motion modeling to create a more robust dynamics model seems to be an interesting direction to mitigate this problem. Our model is currently not relightable and can not animate new hairstyles. This could be addressed by separately modeling appearance and lighting. As our model is created persons specific, it will not work well for hairstyles it has not been trained on. How to design a universal hair representation that stores motion and appearance priors for diverse hairstyles in a shared manifold is an interesting question to answer in future work.

References

[1] Florence Bertails, Sunil Hadap, Marie-Paule Cani, Ming Lin, Tae-Yong Kim, Steve Marschner, Kelly Ward, and Zoran Kačić-Alesić. Realistic hair simulation: animation and rendering. In *ACM SIGGRAPH 2008 classes*, pages 1–154. 2008. 3

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 3

[3] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. Authentic volumetric avatars from a phone scan. *ACM Trans. Graph.*, 41(4), jul 2022. 3

[4] Menglei Chai, Jian Ren, and Sergey Tulyakov. Neural hair rendering. In *European Conference on Computer Vision*, pages 371–388. Springer, 2020. 3

[5] Menglei Chai, Changxi Zheng, and Kun Zhou. A reduced model for interactive hairs. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014. 3

[6] Menglei Chai, Changxi Zheng, and Kun Zhou. Adaptive skinning for interactive hair-solid simulation. *IEEE transactions on visualization and computer graphics*, 23(7):1725–1738, 2016. 3

[7] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021. 3

[8] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18653–18664, June 2022. 3

[9] Peng Guan, Leonid Sigal, Valeria Reznitskaya, and Jessica K Hodgins. Multi-linear data-driven dynamic hair model with efficient hair-body collision handling. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pages 295–304, 2012. 3

[10] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20374–20384, June 2022. 3

[11] Liwen Hu, Derek Bradley, Hao Li, and Thabo Beeler. Simulation-ready hair capture. In *Computer Graphics Forum*, volume 36, pages 281–294. Wiley Online Library, 2017. 2

[12] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014. 2

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[14] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. CoNeRF: Controllable Neural Radiance Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[16] Anastasiia Kornilova, Marsel Faizullin, Konstantin Pakulev, Andrey Sadkov, Denis Kukushkin, Azat Akhmetyanov, Timur Akhtyamov, Hekmat Taherinejad, and Gonzalo Ferrer. Smartportraits: Depth powered handheld smartphone dataset of human portraits for state estimation, reconstruction and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21318–21329, June 2022. 3

[17] Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepmvshair: Deep hair modeling from sparse views. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 2

[18] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. 2022. 3

[19] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):194–1, 2017. 3

[20] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 6

[21] Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. Video to fully automatic 3d hair model. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 2

[22] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4), 2019. 2, 3

[23] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4), July 2019. 3

[24] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40(4), July 2021. 2, 3, 4, 5, 6, 7

[25] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1490–1497. IEEE, 2012. 2

[26] Qing Lyu, Menglei Chai, Xiang Chen, and Kun Zhou. Real-time hair simulation with neural interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 3

[27] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing

image-based volumetric avatars using relative spatial encoding of keypoints, 2022. 3

[28] 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 *European Conference on Computer Vision*. Springer, 2020. 3

[29] Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 155–164, 2019. 2, 3

[30] Kyle Olszewski, Duygu Ceylan, Jun Xing, Jose Echevarria, Zhili Chen, Weikai Chen, and Hao Li. Intuitive, interactive beard and hair synthesis with generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7446–7456, 2020. 3

[31] Sylvain Paris, Hector M Briceno, and François X Sillion. Capture of hair geometry from multiple images. *ACM Transactions on Graphics (TOG)*, 23(3):712–719, 2004. 2

[32] Sylvain Paris, Will Chang, Oleg I Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Transactions on Graphics (TOG)*, 27(3):30, 2008. 2

[33] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2, 3

[34] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 3

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 4

[36] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11733–11742, June 2021. 3

[37] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 3

[38] 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. *ECCV*, 2022. 2

[39] Tiancheng Sun, Giljoo Nam, Carlos Aliaga, Christophe Hery, and Ravi Ramamoorthi. Human Hair Inverse Rendering using Multi-View Photometric data. In Adrien Bousseau and Morgan McGuire, editors, *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2021. 2

[40] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 6

[41] Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. Learning compositional radiance fields of dynamic human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5704–5713, 2021. 3

[42] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Michael Zollhoefer, Jessica Hodgins, and Christoph Lassner. Hvh: Learning a hybrid neural volumetric representation for dynamic hair performance capture, 2021. 2, 3, 4, 6, 7

[43] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R Marschner, Marie-Paule Cani, and Ming C Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE transactions on visualization and computer graphics*, 13(2):213–234, 2007. 3

[44] Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. Modeling hair from multiple views. In *ACM SIGGRAPH 2005 Papers*, pages 816–820. 2005. 2

[45] Sebastian Winberg, Gaspard Zoss, Prashanth Chandran, Paulo Gotardo, and Derek Bradley. Facial hair tracking for high fidelity performance capture. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 2

[46] Chenlei Wu and Takashi Kanai. Data-driven detailed hair animation for game characters. *Computer Animation and Virtual Worlds*, 27(3-4):221–230, 2016. 3

[47] Chenglei Wu, Takaaki Shiratori, and Yaser Sheikh. Deep incremental learning for efficient high-fidelity face tracking. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 3

[48] Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2022. 2

[49] Zexiang Xu, Hsiang-Tao Wu, Lvdi Wang, Changxi Zheng, Xin Tong, and Yue Qi. Dynamic hair capture using spacetime optimization. *ACM Transactions on Graphics (TOG)*, 33(6), nov 2014. 2

[50] Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. Dynamic hair modeling from monocular videos using deep neural networks. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 2

[51] Qing Zhang, Jing Tong, Huamin Wang, Zhigeng Pan, and Ruigang Yang. Simulation guided hair dynamics modeling from video. In *Computer Graphics Forum*, volume 31, pages 2003–2010. Wiley Online Library, 2012. 2

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[53] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I m avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13545–13555, June 2022. 3