

GaussianHead: Impressive 3D Gaussian-based Head Avatars with Dynamic Hybrid Neural Field

Jie Wang¹, Xianyan Li¹, Jiucheng Xie¹, Feng Xu², Hao Gao,¹

¹Nanjing University of Posts and Telecommunications,

²Tsinghua University

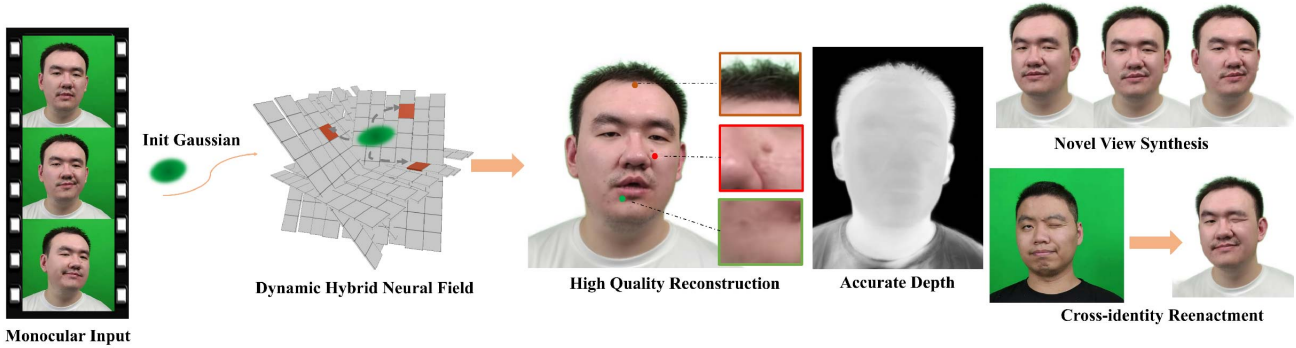


Figure 1. GaussianHead starts from monocular videos, integrating canonical 3D gaussian factors by constructing a dynamic tri-plane, and using it as the generation condition for gaussian attributes. Based on our novel factor representation, GaussianHead achieves optimal visual effects, efficient rendering, and features such as low model size. We evaluated it on various tasks, including self-reconstruction, novel view synthesis, and cross-identity reenactment.

Abstract

Previous head avatar methods have mostly relied on fixed explicit primitives (mesh, point) or implicit surfaces (Sign Distance Function) and volumetric neural radiance field, it challenging to strike a balance among high fidelity, training speed, and resource consumption. The recent popularity of hybrid field has brought novel representation, but is limited by relying on parameterization factors obtained through fixed mappings. We propose GaussianHead: an head avatar algorithm based on anisotropic 3D gaussian primitives. We leverage canonical gaussians to represent dynamic scenes. Using explicit "dynamic" tri-plane as an efficient container for parameterized head geometry, aligned well with factors in the underlying geometry and tri-plane, we obtain aligned canonical factors for the canonical gaussians. With a tiny MLP, factors are decoded into opacity and spherical harmonic coefficients of 3D gaussian primitives. Finally, we use efficient differentiable gaussian rasterizer for rendering. Our approach benefits significantly from our novel representation based on 3D gaussians, and the proper alignment transformation of underlying geometry structures and factors in tri-plane eliminates

biases introduced by fixed mappings. Compared to state-of-the-art techniques, we achieve optimal visual results in tasks such as self-reconstruction, novel view synthesis, and cross-identity reenactment while maintaining high rendering efficiency (0.12s per frame). Even the pores around the nose are clearly visible (Figure.1) in some cases. Code and additional video can be found on the project homepage: <http://chiehwangs/chiehwangs.github.io-gaussianhead>.

1. Introduction

Creating personalized avatars for everyone is essential for the future wide-ranging applications of virtual reality and metaverse technologies. Commercial deployment requires easy data capture, fast rendering, low model volume occupation, and high-fidelity visual effects. However, existing methods struggle to strike a balance in these aspects.

Early head avatars focused on coarse skinned surfaces. They mostly created surface templates for head avatars without eyes, mouth, and hair, based on face surface data obtained from extensive scans [5, 14, 25]. These templates were controlled by linear blend skinning algorithms. While

these methods could be rapidly driven, they lacked acceptable visual effects and were geometrically constrained by the limitations of the template. Building implicit head avatars using Signed Distance Fields (SDF) [53–55] helped overcome the deficiencies of skinned models and achieved better head geometry [62]. Nevertheless, implicit surfaces lacked excellent expressive power for fine structures and fast rendering. With the rise of neural radiance fields (NeRF [33]), some methods modulated implicit neural radiance fields based on time, expressions, or audio embeddings [11, 16, 37] to construct head avatars. However, their extended training times hindered their practical large-scale applications.

Combining explicit data structures to build volumetric neural radiance fields is a current trend. Methods like [4, 6, 7, 9, 34] store scenes using explicit data container, allowing the use of smaller MLPs to focus on decoding semantic information (i.e. density and color), resulting in faster optimization speed. Methods like [12, 24, 44, 49, 61] achieved realistic head avatars by storing scenes using voxels or multiple 2D planes instead of the weights of neural networks, but these methods are still limited by the large number of spatial sampling points required for NeRF, and they are not the optimal choice for achieving excellent performance in fine structures and compact model volume. They either require excessive training memory or time, or limited to complex facial textures. Additionally, they exhibit visual effects such as ‘smoothing’ and ‘blurring’. Compared to explicit representation (i.e. mesh, point) or implicit representations (i.e. NeRF [33], SDF [36]), our GaussianHead have greater advantages, as shown in Tab. 1

In this paper, we employ pre-retrieved facial expression parameters as conditions to transform randomly initialized 3D gaussians, which have initial position, color, opacity and shape (parameterized by the covariance matrix), into canonical space to address dynamic facial movements. To simultaneously capture complex facial muscle dynamics and intricate textures, we utilize an explicit data structure – parameterized tri-plane – to access canonical factors of gaussians. The canonical gaussians are tracked, and their alignment with factors on feature planes is achieved by applying learnable rotations represented by quaternions. This approach reduces bias in representing factors of 3D gaussians compared to previous works employing axis-aligned mappings [30, 42, 56], resulting in improved visual effects and reduced errors. Canonical factors are used to decode opacity and view-dependent spherical harmonic coefficients of 3D gaussians. Finally, a differentiable gaussian rasterizer is employed to render realistic head avatars. Furthermore, we propose GaussianHead+, which achieves data dimensionality reduction through 2D planes and utilizes a hash map [34] to store canonical factors, resulting in a highly competitive model size as described in Sec.4.3, refer to Figure.2 for an

	Fast Rendering	Easy Animation	Flexible Primitives	Explicit Geometry	Low model size
Meshes	✓	✓	✗	✓	✓
Points	✓	✓	✗	✓	✗
Volumetric NeRF	✓	✗	✗	✗	✓
Implicit Surface	✗	✗	✗	✓	✗
3D Gaussians (Ours)	✓	✓	✓	✓	✓

Table 1. GaussianHead is easy to integrate with existing hardware to achieve efficient rasterization rendering, and can flexibly represent even hair-like structures. It also features deformable primitives to represent complex geometric structures with fewer elements compared to point. Notably, our method only requires about 4-5GB of GPU memory and maintains a model size of around 4.2MB.

overview of our approach.

In summary, our contributions are as follows:

- Using anisotropic 3D gaussian primitives to construct dynamic head avatars, allowing for high-fidelity reconstruction and fast rendering;
- Novel canonical space geometric perception-assisted reconstruction and efficient feature representation with tri-plane and hash map, enhancing the representation effect while achieving low resource consumption;
- The novel dynamic learnable mapping scheme in hybrid neural field is introduced to mitigate biases introduced by axis-aligned mapping, thereby facilitating a robust alignment between the underlying structure and factor planes.
- Extensive experiments on public datasets have shown that our method is superior to previous methods in terms of visual effects and efficient representation.

2. Related Work

Scene Primitives in Reconstruction. Whether it is a real-world scenario, the human body, or a head, the foundation of their construction lies in a simple form of scene primitives. Some past approaches have relied on implicit primitives, such as signed distance function [15, 19, 28, 35, 36, 46, 53–55] that builds objects by tracing points located on the zero level set of the function in space, or occupancy function [32, 39, 62] that represents three-dimensional surfaces as the continuous decision boundaries of deep neural network classifiers. However, these implicit primitives cannot represent complex structures, and their approximation capability for surfaces of rapidly changing objects is insufficient for high-fidelity reconstruction. Implicit neural radiance fields [8, 11, 13, 17, 26, 31, 33, 40, 59, 60] are a way of storing scenes using network weights as primitives, but due to network capacity limitations, they perform

poorly in complex scenes, simultaneously accompanied by a significant training cost. Explicit scene primitives, such as points [48, 63], can represent sufficiently complex structures by continuously refining point radii and increasing point counts during scene construction, but it introduces significant storage and training overhead, limiting the shape of primitives, which can only be addressed by either increasing quantity or reducing size.

Here, we use 3D anisotropic gaussians [21] as scene primitives. They have deformable shapes, allowing for the representation of intricate details by adjusting their own shapes instead of blindly increasing quantity and reducing radii. As a result, this approach maintains a smaller training overhead while expressing complex structures effectively.

Monocular-video-based Head Portrait Synthesis. The widespread application of head avatars requires convenient capture methods. Some studies [23, 45] utilize dense multi-view or sparse multi-view to capture dynamic head avatars in space. Although this brings better multi-view consistency, the difficulty in popularizing capture devices hinders the rapid adoption of such technologies. Meanwhile, they mostly reconstruct head dynamics along the temporal axis, sacrificing cross-subject generalization performance. Building head avatars from monocular videos, allowing users to capture data using smartphones, has played a positive role in the promotion of technologies such as virtual reality. However, a challenge lies in how to leverage multi-view information from monocular videos. Common approaches involve anchoring the head in space and using head poses as a substitute for camera poses [11, 12, 49]. Several researchs explore the use of monocular data for new view synthesis and high-fidelity reconstruction tasks. They employ implicit surfaces [11, 16, 62], explicit point cloud structures [63], or volumetric neural radiance fields [12, 24, 44, 49] as head representations, achieving increasingly realistic visual effects.

Hybrid Neural Field. Previous implicit methods such as NeRF [33] are limited to a large number of sampling points in space. Using MLP weights to implicitly represent the scene means that as the complexity of the scene increases, the network complexity and learning difficulty rise rapidly. Recently, hybrid neural radiation fields have become a hot topic [1, 2, 10, 12, 27, 34, 43, 49, 50, 58], storing scenes through explicit data structures (i.e. tri-plane [6], hexplane [4, 9], voxel [34]) to share the pressure of neural networks, allowing neural networks to focus on decoding semantic information (i.e. density and color) from explicit structures rather than storing the entire scene information. Instant neural graphics primitives [34] adopts multi-level hash map to augment a shallow MLP and achieves a combined speedup of several orders of magnitude. However, these methods could not be directly used for dynamic scenes due to its complex non-rigid deformations, and ex-

PLICIT data structures still need to store information for the entire spatial point. We use parametric tri-plane [9] and 3D gaussian differentiable rasterization, which automatically discards redundant gaussians after meeting the pixel rendering requirements, significantly reduced the storage burden of parametric data structures. At the same time, the newly added gaussians will appear in areas that are not well optimized, rather than being evenly distributed in space like NeRF [33].

3. Preliminary of 3D Gaussian Splatting

3D gaussian splatting [21] is an explicit representation method for scenes that utilizes explicit anisotropic 3D gaussian primitives to represent the underlying structure of the scene. Each 3D gaussian is an optimizable object with optimizable attributes: position $x \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, quaternion rotation $r \in \mathbb{R}^4$, scaling factor $s \in \mathbb{R}^3$, and spherical harmonic coefficients $c \in \mathbb{R}^k$. For ease of representation, quaternion rotation and scaling factor are merged into the covariance matrix Σ . Therefore, a 3D gaussian is represented as 3D gaussian distribution with a central position μ and a covariance matrix Σ :

$$\mathcal{G}(\mu) = e^{-\frac{1}{2}\mu^T \Sigma^{-1} \mu}, \quad (1)$$

In order to render, 3D gaussians are arranged above the camera plane and employ a differentiable rasterization technique [57] similar to alpha compositing. The covariance in world space undergoes transformations into the camera coordinate under the changes of the view transformation matrix W and the Jacobian matrix J resulting from affine approximation in the projective transformation [64, 65]:

$$\Sigma' = JW\Sigma W^T J^T, \quad (2)$$

For each pixel x on the camera plane, the color is synthesized by combining all the 3D gaussians above it:

$$C_x = \sum_{i \in N} c_{i,x} \alpha_{i,x} \prod_{j=1}^{i-1} (1 - \alpha_{j,x}), \quad (3)$$

Where

$$\alpha_{i,x} = \alpha_i e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}, \quad (4)$$

$c_{i,x}$ is the color of the i -th gaussian above the pixel. It is calculated using the spherical harmonic coefficients. $\alpha_{i,x}$ represents the opacity of the i -th gaussian in the region above the pixel x . It is obtained through the center opacity α_i of this gaussian, its center position μ , and the pixel position x .

4. GaussianHead Method

Given a segment of monocular video, we extract camera parameters, rgb frames, expression coefficients, and masks

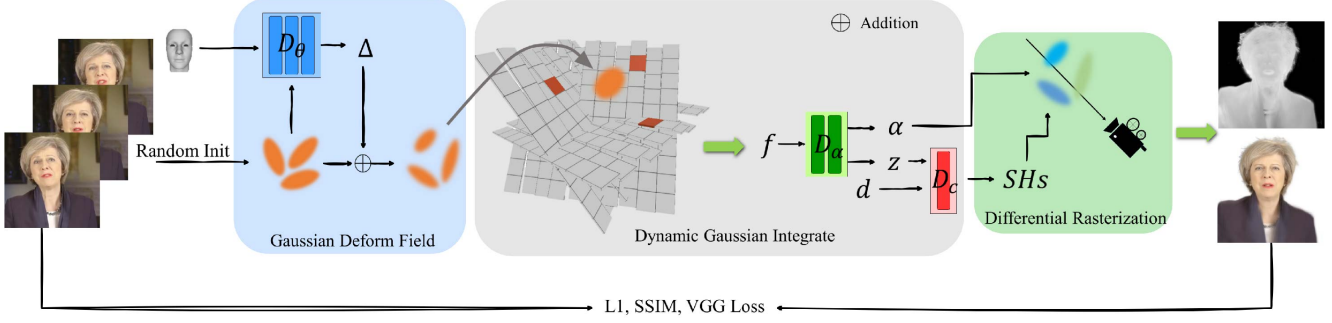


Figure 2. Overview of GaussianHead: We randomly initialize some 3D gaussians in the world space. Conditioned on expression, through a deformation field, we learn random 3D gaussians’ displacements to the canonical space, denoted as Δ . The canonical gaussians are fed into a dynamic hybrid neural field, where canonical factors f are queried on dynamic tri-plane. These canonical factors, along with intermediate latent variables z , are decoded into opacity attributes through an opacity decoder D_α . Furthermore, z , combined with the viewing direction d , are used by the color decoder D_c to generate spherical harmonic coefficients representing view-dependent color. Finally, the position, shape, opacity, and color attributes of 3D gaussians are collectively rendered into realistic images through a differentiable rasterizer.

as inputs to our method. The GaussianHead learns the spatial positions, shapes, and sizes of each 3D gaussian primitive. It combines geometric factors from explicit dynamic tri-plane to learn the opacity and spherical harmonic coefficients used in differentiable rasterization for image rendering. After training, the GaussianHead can render at a speed of 0.12 seconds per frame. An overview of our approach is presented in Figure.2.

4.1. Gaussian Deform Field

3D gaussian splatting is a high-fidelity reconstruction method for static scenes, while avatars exhibit complex dynamic properties. A feasible approach is to transform dynamic avatars into a canonical space with a common geometric representation. We randomly initialize 3D gaussians $\mathcal{G}_i = (x_i, r_i, s_i)$ with learnable position x , rotation r , and scale s in world space, where $i = 1, 2, \dots, N$ means the number of 3D gaussians. Subsequently, we use pre-acquired facial expression parameters e as conditions, employing a standard deformation multi-layer perceptron to predict position offsets δx , rotation offsets δr , and scale offsets δs based on the initialized 3D gaussians, resulting in 3D gaussians in the canonical space:

$$(\delta x, \delta r, \delta s) = \mathcal{D}_\theta(\gamma(x), e), \quad (5)$$

Where θ represents the optimized parameters of the standard deformable Multi-Layer Perceptron (MLP), and γ represents the encoding of the spatial coordinates of the 3D gaussian into a high-dimensional sine-cosine sequence, as described in [33]. In our experiments, a frequency of 10 was chosen for the positional encoding. Subsequently, we overlay offsets onto the initial 3D gaussian to obtain the canonical gaussian represented as $\mathcal{G}_o = (x_o, r_o, s_o)$.

4.2. Dynamic Canonical Gaussian Integrate

While conventional deformation networks can represent dynamic scenes, simply transforming the initial gaussian through an MLP to the canonical gaussian and then used for rendering often struggles to simultaneously capture both dynamic and appearance details due to the limited expressive power and capacity of a single deformation network [23, 45, 49]. Therefore, we propose using parametric tri-plane to construct a blended neural field. However, previous works based on explicit tri-plane [4, 6, 9] have all employed a straightforward fixed-axis alignment mapping to project tracked points onto factor planes, leading to axis-aligned mapping biases [30, 42]. Inspired by works [42, 56], we propose a dynamic (mapping) hybrid neural field with learnable transformations for constructing head avatars.

We parameterize the space tri-plane $\mathbf{P}_{xy}, \mathbf{P}_{xz}, \mathbf{P}_{yz}$, and set the shape of the space tri-plane to (H, W, F) , where H and W represent the size of each plane, and F represents the number of channels for factors on the plane, capturing the geometric and color properties of the head space.

We track the position x_o of each 3D gaussian \mathcal{G}_o on the canonical space and generate T sets of rotation vectors $\mathbf{v} = (v_x, v_y, v_z)$ to distort the positions of 3D gaussians through random initialization. Directly constructing the rotation vectors as rotation matrices would introduce a significantly increased number of parameters and optimization for rotation matrices is highly ill-conditioned. Therefore, we propose generating the rotation vectors as unit quaternions $r = (a, b_i, c_j, d_k)$, where the number of rotation vectors satisfies $F \bmod T = 0$. In more detail, after processing each quaternion to each rotation vector, we apply the rotation to the tracked 3D gaussians to get “aligned gaussians” and map them to tri-plane. By bilinear interpolation, we obtain canonical single-plane sub-factors (F/T channels).

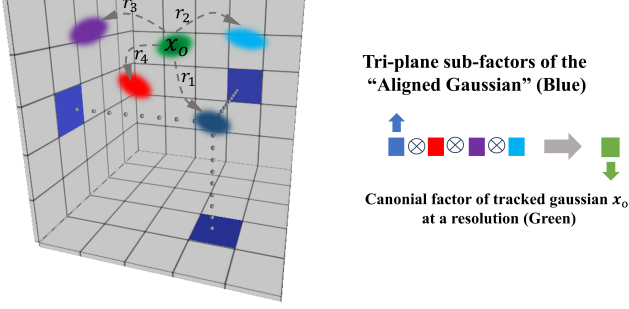


Figure 3. The tracked gaussian x_o is rotated T times (4 in our experiments), distorted to four 'aligned gaussians' and further mapped to the factor plane to obtain sub-factors (For simplicity, we only show one quadrant). Multiple sub-factors are combined to form the canonical factors of the tracked gaussian. For rendering, we still use the position of the tracked gaussian instead of the 'aligned gaussians'. Therefore, this process describes the transformation of the original gaussian factors obtained through fixed mappings to canonical factors using a learnable mapping, referred to as the 'factor distortion' on tri-plane.

Then, 3 sub-factors from tri-plane are multiplied to obtain the tri-plane sub-factors. This operation is performed for T times on one tracked gaussian at the same resolution, resulting in T tri-plane sub-factors. These T sub-factors are multiplicatively combined to obtain the aligned factor at one resolution (green block in Figure.3). Finally, the aligned factors at different resolutions are concatenated to obtain the multi-resolution aligned canonical factor f for the tracked 3D gaussian x_o :

$$f = \bigcup_{r \in \{reso\}} \prod_{i=1}^T \prod_{c=1}^3 \psi_r(\mathbf{P}_c, r(x_o)), \quad (6)$$

Where \mathbf{P}_c represents the c -th plane, \bigcup denotes the concatenation operation, and ψ_r represents the interpolation of gaussian rotated at resolution r onto the plane. A more intuitive process is demonstrated in Figure.3.

Due to manifold constraints, unit quaternions are restricted to the manifold rather than the entire Euclidean space. Simple first-order optimizers assume flat space and do not consider the geometric structure of the manifold. Directly applying standard optimizers to unit quaternions may lead to solutions that do not conform to the manifold. Therefore, we employ the Riemannian ADAM optimizer [3], where updates to unit quaternions induced by the learning rate α_k for the current step k and the loss backward in their tangent space gradient ∇_k to ensure adherence to the manifold:

$$r_{k+1} = r_k \text{Exp}(\alpha_k \nabla_k) \quad (7)$$

Compared to recent dynamic scene approaches based on 3D gaussians [29, 47, 51, 52] that directly set opacity and

	IMavatar	Point Avatar	NeRF BlendShape	GaussianHead (ours)	GaussianHead+ (ours)
Model Size (MB)	48.5	33.9	564.9	135.1	4.5
Render Time (Seconds)	128	1.5	0.24	0.1	0.13

Table 2. Model size and rendering speed. In terms of model lightweighting, both our GaussianHead and GaussianHead+ outperform NeRFBlendShape. Although GaussianHead has a larger model size compared to Imavatar and PointAvatar, it exhibits superior visual performance. Additionally, our GaussianHead+ surpasses the first three methods in both model lightweighting and visual effects.

color as optimizable parameters, we employ two small decoder MLPs to decode the canonical factors from the dynamic tri-plane into semantic information: opacity α and spherical harmonic coefficients Y_{lm} . This process ensures a more precise generation of opacity and color as mentioned in sec.5.7. Finally, differentiable rasterization in formula 3 is used for rendering.

$$(\alpha, z) = \mathcal{D}_\alpha(f), \quad (8)$$

$$Y_{lm}(\theta, \phi) = \mathcal{D}_c(d, z), \quad (9)$$

Where \mathcal{D}_α and \mathcal{D}_c represent MLPs for decoding opacity and spherical harmonic coefficients, respectively. z and d denotes latent variable and view direction, and we use 4th-order spherical harmonic coefficients to synthesize view-dependent colors.

4.3. GaussianHead+

GaussianHead has made significant progress in generating high-fidelity head avatars. However, in terms of model size, directly storing canonical factors on explicit tri-plane is not the optimal choice. To maintain the consistency of our GaussianHead, we further propose GaussianHead+. We store the canonical factors of gaussians in a multi-resolution learnable hash map [24, 34]. Instead of introducing a large number of parameters as in a 3D voxel structure, we use a tri-plane structure and a learnable mapping to project 3D coordinates onto a 2D plane. The reduced-dimensional coordinates are then fed into the hash map to query canonical factors, further reducing data volume and increasing efficiency. In comparison to GaussianHead, explicit tri-plane are no longer used as a data storage structure but as a data dimensionality reduction tool. A more compact and efficient multi-resolution hash map is used as a factor container. GaussianHead+ achieves competitive visual results while maintaining an extremely competitive model size (only 4.5MB) in terms of model lightweighting. We compare the model sizes of various methods in Table.2 to demonstrate our advantage, parameters we used in Table.3.



Figure 4. Qualitative comparison with other methods. The original NeRFBlendShape [12] paper only provides head reconstruction, so we adhere to its algorithm implementation. All methods are run under the hardware conditions and parameter configurations specified by their respective authors. Our method outperforms others in aspects such as wrinkle rendering, hair representation, and facial expression fidelity.

Parameter	value
number of levels	12
Max. entries per level (hash table size)	2^{14}
Number of feature dimensions per entry	2
Coarsest resolution	16
Finest resolution	2^{16}

Table 3. The main parameter settings for the GaussianHead+ hash map

Method	L1↓	PSNR↑	SSIM↑	LPIPS↓
IMavatar [62]	0.00296	25.29	0.8776	0.190
PointAvatar [63]	0.00211	26.76	0.8611	0.162
NeRFBlendShape [12]	0.00117	29.32	0.911	0.102
Ours (GaussianHead+)	0.00078	31.12	0.936	0.122
Ours (GaussianHead)	0.00071	31.55	0.940	0.091

Table 4. Quantitative Comparison with baselines: Our GaussianHead achieves the best quality results. Although there is a slight decrease in quality with GaussianHead+, it still surpasses the baseline metrics while maintaining an extremely compact model size.

4.4. Training Objectives

Compared to the past popular methods that process a subset of pixels at a time [11, 16, 18, 33, 49] using neural radiance fields and implicit reconstruction, we employ differ-

entiable rasterization based on 3D gaussians to reconstruct an entire image at once. Consequently, a greater variety of image-based losses can be utilized.

Image-based Loss. We utilize ssim loss and perceptual loss to measure the quality loss of the rendered images:

$$\mathcal{L}_{\text{vgg}} = \|\text{vgg}(\mathbf{I}) - \text{vgg}(\mathbf{I}^{\text{GT}})\|_1, \quad (10)$$

$$\mathcal{L}_{\text{ssim}} = 1 - \text{ssim}(\mathbf{I}, \mathbf{I}^{\text{GT}}), \quad (11)$$

Here, we use the first four layers of the VGG16 model [41] and assess the similarity between the features in the intermediate layers of the rendered images and the real images:

Pixel-based Loss. L1 loss is employed to measure the pixel-wise difference between the rendered images and the real images:

$$\mathcal{L}_{\text{rgb}} = \|\mathbf{I} - \mathbf{I}^{\text{GT}}\|_1, \quad (12)$$

The final loss function is as follows, where we set $\lambda_1 = 0.8$, $\lambda_2 = 0.01$, $\lambda_3 = 0.2$.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{rgb}} + \lambda_2 \mathcal{L}_{\text{vgg}} + \lambda_3 \mathcal{L}_{\text{ssim}} \quad (13)$$

5. Experiments

5.1. Baselines

We choose recently state-of-the-art methods based on mainstream representation for comparison: IMavatar [62]

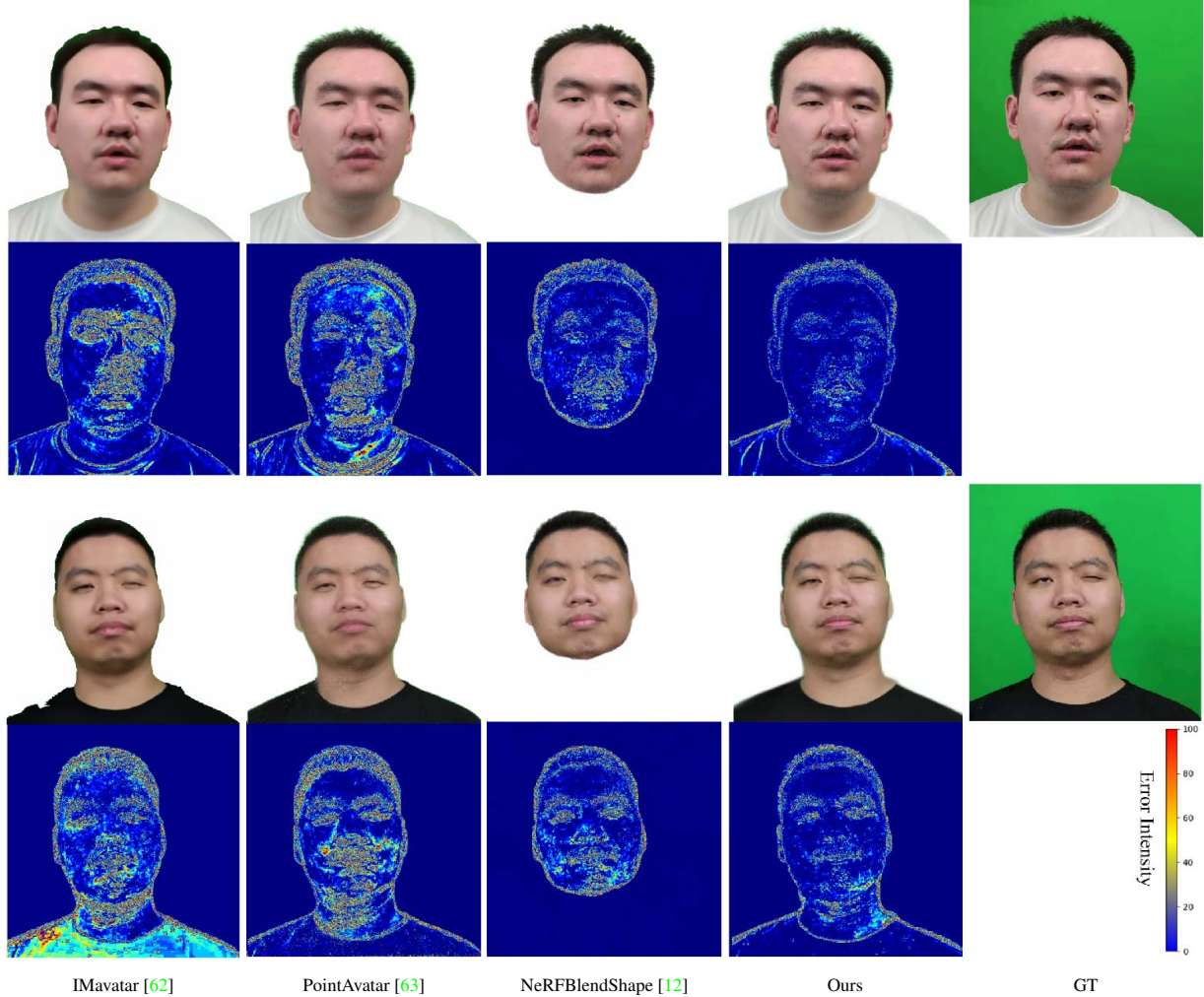


Figure 5. We visualize the error map by computing the Mean Squared Error (MSE) between visualized pixels and ground truth, mapped to the RGB domain. This makes it easier to identify problematic areas in the reconstructed results, where brighter colors indicate poorer reconstruction. Among all compared methods, our results exhibit the overall minimum error.

based on implicit surfaces, reconstructs avatars by building a neural implicit surface on the foundation of 3DMM; PointAvatar [63] based on explicit points, constructs detailed avatars using iteratively refined and upsampled point clouds. NeRFBlendShape [12], the excellent work based on volumetric neural radiance fields, utilizes a hash grid as the storage structure for the facial expression base to construct high-fidelity head avatars.

5.2. Dataset

All subjects are open-source [12] and come with rich raw data. The training data we use for each subject consists of approximately 2000 frames, and evaluate the results using the last 200 frames. Our approach streamlined the required parameters and additionally utilized MODNet [20] to obtain binary masks. Specifically, the data for each subject we

used only includes four parts: RGB head images with background, expression parameters fitted using a blendshape model, camera intrinsic and extrinsic parameters, and binary masks.

5.3. Implementation Details

We implemented GaussianHead with PyTorch [38], where GaussianHead+ with hash grid and differentiable 3D Gaussian rasterization is implemented based on CUDA kernels [21, 34]. The initialization includes 10k 3D gaussians. Apart from optimizing the manifold using Riemannian ADAM [3], all other optimizers use first-order ADAM [22], and linear learning rate scheduler is applied. In our experiments, we set the batch size to 1 and trained for 80k iterations (about 30 epochs) on a single RTX 3090, taking approximately 5GB GPU memory and 3 hours.

After 3000 iterations, we perform densification of 3D Gaussians and clipping of overly large Gaussians with very low opacity every 500 iterations. We set the opacity threshold to 0.0002 and the size threshold for large Gaussians to be greater than 1% of the scene. Other details follow the settings in [21]. For network configurations and other details, please refer to the supplementary materials.

5.4. Qualitative and Quantitative Comparison in Reconstruction

We showcase our state-of-the-art visual quality in Figure.4 and Figure.5. Imavatar [62] and Pointavatar [63] reconstruct avatars based on implicit surfaces and explicit points, providing effective driving capabilities. However, their reliance on the FLAME [25] 3DMM model introduces shortcomings in expressing expressions. Expression movements controlled by Linear Blend Skinning (LBS) fail to capture extreme expressions, and consequently, information such as wrinkles induced by extreme expressions cannot be properly represented. NerfBlendshape [12], which relies on an explicit hash map to store expression bases, combines expression coefficients to reconstruct avatars. However, the direct storage of 3D spatial points in multiple hash maps results in excessively large model sizes (in Table.2). Meanwhile, all of the above methods come with a strong smoothing effect. From a visual perspective, we are able to effectively represent subtle details such as slightly closed eyes and wrinkles when smiling, which these models struggle to accurately depict. Furthermore, we report the results of quantitative comparisons in Table.4.

5.5. Novel View Synthesis

The generated avatars by GaussianHead exhibit excellent multi-view consistency. Even for perspectives not encountered during training, GaussianHead generalizes to produce high-fidelity geometry and appearance. We achieve this by observing the avatars from multiple viewpoints obtained by rotating the camera position, as illustrated in Figure.6. The multi-view results show no occurrence of artifacts or unrealistic facial expressions even at intermediate rotation angles. At the same time, our method can also effectively represent details such as teeth (such as Figure.8). all this characteristic is demonstrated in video format on our project homepage.

5.6. Cross-identity Reenactment

Compared to some previous works that reconstructed dynamic scenes or dynamic avatars along the temporal dimension, limiting the generalization performance of scene motion and preventing the replication of motion attributes from other scenes, GaussianHead employs expression parameters as conditions for motion modeling. This allows for the excellent replication of the subject’s own movements while



Figure 6. Novel view synthesis results of GaussianHead: These perspectives have never appeared in the training data, demonstrating the strong spatial consistency of our approach. In these new viewpoints, even details such as teeth and wrinkles are clearly represented.

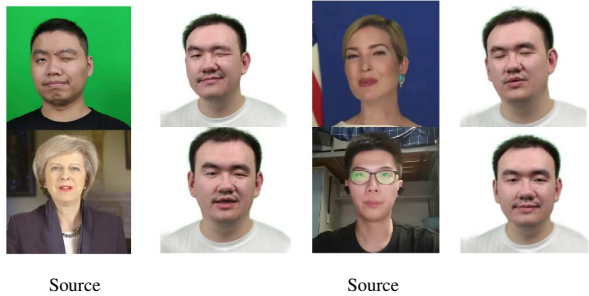


Figure 7. We use the source subject to drive the execution of new poses and expressions for the other subject.

also enabling motion control by other subjects, facilitating cross-identity reenactment. The results of this evaluation project are presented in Figure.7.

5.7. Ablation studies

Parametric tri-plane. We transform randomly initialized 3D gaussians into canonical space using a deformation MLP, and store canonical gaussian factors using parametric tri-plane. Due to limitations in the capacity and expressive power of a single MLP, it is challenging to simultaneously capture the dynamic facial muscle movements and textures. To address this limitation, we introduce parametric tri-plane as an additional component, where a multi-resolution grid representation is crucial for achieving high-detail expression [34]. We ablate the explicit structure of parametric tri-plane, directly optimizing the opacity and spherical harmonic coefficients of dynamic 3D gaussians as trainable parameters, without modulation through canonical factors. The canonical gaussians obtained by the deformation MLP are used directly for differentiable rasterization. We present the ablation results in Figure.9. The depth maps we visualize reveal numerous gaussians around the head that are

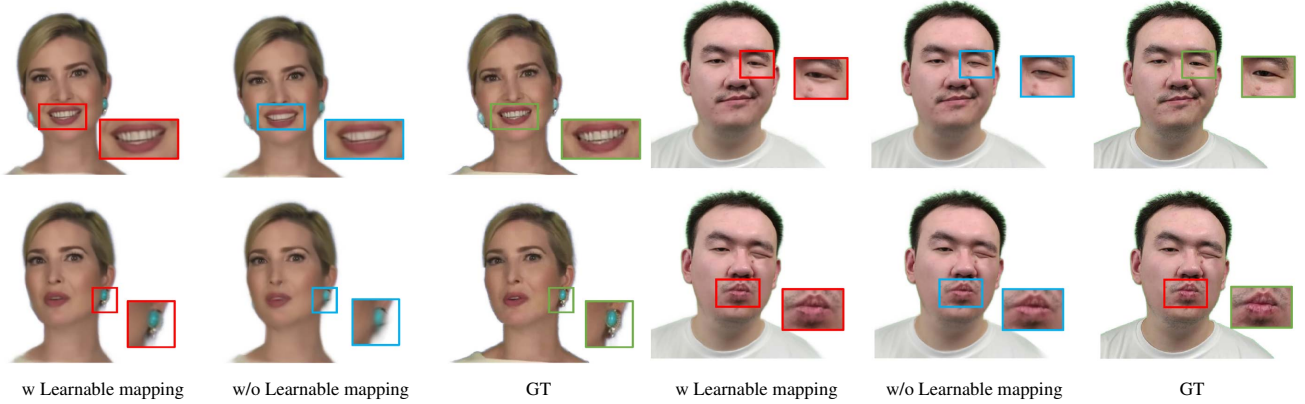


Figure 8. Using a fixed mapping, due to the lack of alignment between geometric structures and factor planes, factors in complex and subtle regions fail to reflect accurate information, resulting in blurriness (as seen in the second column with teeth) and distortion (as seen in the fifth column with eyes) in these areas.

not eliminated through cropping. Although their opacity is greatly reduced, making them imperceptible in the rendered results, the presence of these meaningless gaussians can somewhat slow down the training speed and increase training overhead. The low opacity gaussians may not affect the overall visual effect, but they have a greater impact on detailed areas. Therefore, a single MLP is not sufficient to capture extremely accurate gaussian properties.

parameters that control the coordinate mapping. This allows the tracked 3D coordinates to find the optimal grid factor representation, resulting in improved metrics and finer details. Compared to neural networks with parameters in the order of millions, several tens of optimizable rotation do not significantly increase the training burden. We conducted ablation experiments using a fixed mapping approach referred to as "w/o Learnable mapping", and the results are reported in Figure 8.

6. Limitations

GaussianHead achieves high-fidelity reconstruction of avatars with superior generalization performance in motion. However, reconstructing the interior of the oral cavity remains a challenging aspect, particularly in scenarios with limited training data. Addressing the precise reconstruction of the oral cavity, including the tongue, with fewer data, remains a focal point for our future research. Additionally, our method currently encompasses only the shoulders and above. For further applications in commercial domains, a comprehensive representation of the upper body may be necessary. For accurate emotional expression, consideration of reconstructing the complete upper body, including hand gestures, is crucial. Therefore, the reconstruction of the entire upper body and hand limbs holds significant importance for generating digitally rendered portraits with enhanced emotional expressiveness.

7. Conclusion

We propose GaussianHead: a realistic head avatars generation algorithm based on 3D gaussians with dynamic hybrid canonical factors. We leverage 3D gaussians as efficient and accurate scene primitives, obtaining gaussian attributes through dynamic tri-plane queries and canon-

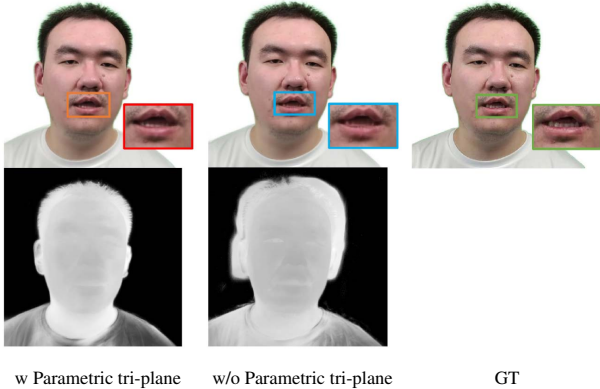


Figure 9. We melted down the results of parameterized three planes. Depth map shows many Gaussians that do not contribute to the rendering results being incorrectly excluded. This demonstrates that a single deformation MLP is insufficient to handle fine structures in the head, and introducing parameterized three planes can better learn Gaussian properties (i.e. opacity, color).

Fixed Mapping of Spatial Gaussians. Almost all previous methods have followed the practice of tracing 3D coordinates in space and directly mapping them to a two-dimensional plane or a three-dimensional voxel grid through axis-aligned means, which can introduce biases in the final results [42, 56]. In GaussianHead, we employ a learnable mapping by introducing a set of quaternion pa-

ical factors aligned with the underlying scene structure. This approach yields more precise gaussian attributes, capable of representing even complex structures such as hair, blemishes, and teeth. Finally, we utilize efficient differentiable rasterization, compatible with modern hardware, to rapidly render impressive results in tasks such as self-reconstruction, cross-identity driving, and new perspective generation.

References

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3
- [3] Gary Bécigneul and Octavian-Eugen Ganeu. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018. 5, 7
- [4] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2, 3, 4
- [5] Chen Cao, Yanlin Weng, Shun Zhou, Yiyi Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013. 1
- [6] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2, 3, 4
- [7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 2
- [8] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2
- [9] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2, 3, 4
- [10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinlong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3
- [11] Guy Gafni, Justus Thies, Michael Zollhofer, 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*, pages 8649–8658, 2021. 2, 3, 6
- [12] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (TOG)*, 41(6):1–12, 2022. 2, 3, 6, 7, 8
- [13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2
- [14] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018. 1
- [15] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2
- [16] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 6
- [17] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. in 2021 IEEE. In *CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 2
- [18] 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*, pages 20374–20384, 2022. 6
- [19] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1251–1261, 2020. 2
- [20] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1140–1147, 2022. 7
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 3, 7, 8
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

- [23] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *arXiv preprint arXiv:2305.03027*, 2023. 3, 4
- [24] Jiahe Li, Jiawei Zhang, Xiao Bai, Jun Zhou, and Lin Gu. Efficient region-aware neural radiance fields for high-fidelity talking portrait synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7568–7578, 2023. 2, 3, 5
- [25] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 1, 8
- [26] 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. 2
- [27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3
- [28] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020. 2
- [29] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 5
- [30] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 2, 4
- [31] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2
- [32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2
- [33] 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 of the ACM*, 65(1):99–106, 2021. 2, 3, 4, 6
- [34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3, 5, 7, 8
- [35] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2
- [36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2
- [37] 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
- [38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7
- [39] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2
- [40] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2
- [41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6
- [42] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. *arXiv preprint arXiv:1206.6418*, 2012. 2, 4, 9
- [43] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3
- [44] Jiaxiang Tang, Kaisiyuan Wang, Hang Zhou, Xiaokang Chen, Dongliang He, Tianshu Hu, Jingtuo Liu, Gang Zeng, and Jingdong Wang. Real-time neural radiance talking portrait synthesis via audio-spatial decomposition. *arXiv preprint arXiv:2211.12368*, 2022. 2, 3
- [45] Kartik Teotia, Xingang Pan, Hyeonwoo Kim, Pablo Garrido, Mohamed Elgharib, Christian Theobalt, et al. Hq3davatar: High quality controllable 3d head avatar. *arXiv preprint arXiv:2303.14471*, 2023. 3, 4
- [46] 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 preprint arXiv:2106.10689*, 2021. 2
- [47] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 5
- [48] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixian Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the*

- IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 3
- [49] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2, 3, 4, 6
- [50] Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Han Huang, Guojun Qi, and Yebin Liu. Latentavatar: Learning latent expression code for expressive neural head avatar. *arXiv preprint arXiv:2305.01190*, 2023. 3
- [51] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 5
- [52] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 5
- [53] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2
- [54] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Baked sdf: Meshing neural sdfs for real-time view synthesis. *arXiv preprint arXiv:2302.14859*, 2023. 2
- [55] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2
- [56] Brent Yi, Weijia Zeng, Sam Buchanan, and Yi Ma. Canonical factors for hybrid neural fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3414–3426, 2023. 2, 4, 9
- [57] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 3
- [58] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 3
- [59] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2
- [60] Kai Zhang, Gernot Riegler, Noah Snaveley, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2
- [61] Weichen Zhang, Xiang Zhou, YuKang Cao, WenSen Feng, and Chun Yuan. Ma-nerf: Motion-assisted neural radiance fields for face synthesis from sparse images. *arXiv preprint arXiv:2306.10350*, 2023. 2
- [62] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Böhler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13545–13555, 2022. 2, 3, 6, 7, 8
- [63] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21057–21067, 2023. 3, 6, 7, 8
- [64] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001. 3
- [65] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 3