ProgressiveAvatars: Progressive Animatable 3D Gaussian Avatars

Kaiwen Song Jinkai Cui Juyong Zhang*
University of Science and Technology of China

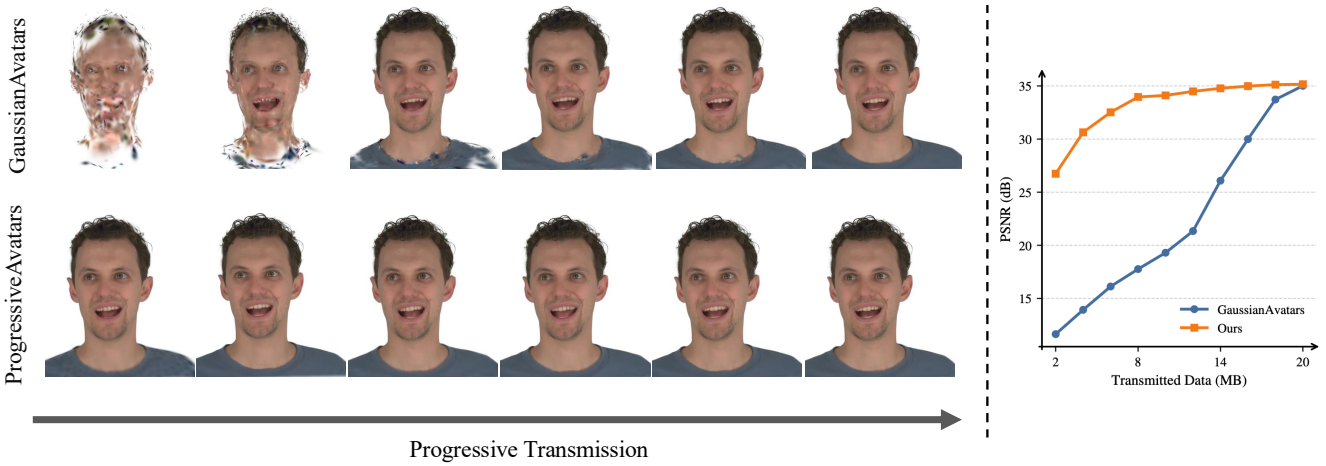


Figure 1. **ProgressiveAvatars** is a novel progressive representation that supports adaptive rendering quality of 3D Gaussian avatars under bandwidth or compute constraints. Qualitative (left) and quantitative (right) results demonstrate that ProgressiveAvatars rapidly attains high quality and continues to refine the avatar as more data arrives, whereas GaussianAvatars [15] only becomes usable once nearly the entire asset has been transmitted.

Abstract

In practical real-time XR and telepresence applications, network and computing resources fluctuate frequently. Therefore, a progressive 3D representation is needed. To this end, we propose ProgressiveAvatars, a progressive avatar representation built on a hierarchy of 3D Gaussians grown by adaptive implicit subdivision on a template mesh. 3D Gaussians are defined in face-local coordinates to remain animatable under varying expressions and head motion across multiple detail levels. The hierarchy expands when screen-space signals indicate a lack of detail, allocating resources to important areas. Leveraging importance ranking, Progressive-Avatars supports incremental loading and rendering, adding new Gaussians as they arrive while preserving previous content, thus achieving smooth quality improvements across varying bandwidths. ProgressiveAvatars enables progressive delivery and progressive rendering under fluctuating network bandwidth and varying compute and memory resources. Project page: https://ustc3dv.github.io/ProgressiveAvatars/

*Corresponding author.

1. Introduction

High-fidelity, real-time rendered head avatars are crucial for immersive interaction, visual communication, and digital human creation. In dynamic multi-user scenarios like Social VR, transmitting high-fidelity avatars as conventional static assets causes severe start-up latency and bandwidth spikes, breaking immersion by forcing users to wait for complete downloads before any rendering can begin. This calls for a progressive 3D representation that provides an animatable avatar capable of minimal start-up latency and continuously refining details as more data arrives. Due to its high fidelity and efficient rendering, 3D Gaussian Splatting [18, 43] and its variants are gradually becoming the mainstream explicit representation. Recent advancements have rapidly expanded across high-fidelity head and body avatars [15, 32, 35, 42, 44, 51], real-time SLAM [33, 34], and advanced rendering techniques [47]. However, existing 3DGS-based avatar methods rarely support progressive, streamable 3D representations. They lack incremental loading mechanisms, failing to smoothly accumulate details with increasing input bandwidth while main-

taining controllability and temporal stability. Therefore, current 3D Gaussian-based digital human representations are unsuitable for these latency-sensitive scenarios.

Constructing a progressive 3D avatar faces several challenges. First, the constructed model must remain animatable across varying transmission budgets. Second, it must maintain temporal stability across various dynamic scenarios under different expressions and head movements. Directly increasing the number of Gaussians can generally improve modeling accuracy, but it heavily increases transfer size, loading times, and instantiation overhead, while excessively restricting capacity compromises modeling accuracy. Prior works [3, 46] typically employ a uniform capacity expansion approach, such as increasing UV resolution or uniformly subdividing the template mesh. This may over-refine smooth areas while under-refining high-frequency areas, resulting in wasted resources. Moreover, existing methods [3, 5, 26, 36] that attempt to offer multiple detail levels typically rely on a discrete LOD paradigm. This requires generating and storing multiple independent copies of the same avatar for different quality budgets, leading to severe storage redundancy and rigid, latency-inducing asset switching. As a result, details cannot be added incrementally, and true progressive transmission and rendering is not supported.

Based on these practical needs and the shortcomings of existing representations, we propose a progressive head avatar representation utilizing adaptive implicit subdivision and continuous detail accumulation. Unlike discrete LOD pipelines that rigidly switch between redundant models, our single, unified asset allows any received subset of Gaussians to be rendered immediately at arbitrary transmission percentages. Built upon the FLAME model [24] to provide a base animatable structure, we construct a mesh-anchored subdivision hierarchy. By binding 3D Gaussians to the local coordinates of each triangle, the avatar remains animatable under mesh deformation and structurally consistent as details incrementally accumulate. Furthermore, rather than using uniform subdivision, we rely on screen-space gradient signals to adaptively refine only detail-rich regions, optimizing resource allocation. Ultimately, as more data streams in, newly arrived Gaussians are seamlessly merged without discarding prior content, enabling true progressive rendering and eliminating the need for cumbersome full-model replacements. In summary, our main contributions are:

- We propose ProgressiveAvatars, a novel progressive 3D avatar representation. It shifts the paradigm from discrete, redundant LOD switching to a single, continuous streamable asset. It grows progressively through adaptive implicit subdivision on a mesh-anchored 3D Gaussian hierarchy, remaining animatable across all streaming stages and recovering fine-scale details via adaptive refinement.
- We achieve incremental loading and progressive rendering through gradual activation. Each streaming increment adds new 3D Gaussians to the previous content, allowing a coarse but animatable avatar to emerge instantly, with visual quality smoothly improving as bandwidth allows.
- Extensive experiments demonstrate that ProgressiveAvatars can rapidly generate high-quality rendering results as data streams in, and its finest modeling results achieve modeling quality comparable to state-of-the-art methods.

2. Related Work

2.1. Animatable Head Avatars

Classical head avatar pipelines fit a morphable mesh with fixed topology to drive animation. Such templates are robust for tracking and retargeting, but they inherently miss delicate geometry and struggle to reproduce complex, view-dependent appearance [7, 12, 17, 23, 27]. To increase realism, many works augment meshes with implicit fields like NeRF [30]. NerFace [8] learns deformation fields to animate heads, but can produce floaters and struggles to match precise target expressions. HeadNeRF [13] builds a parametric head NeRF and uses lightweight 2D neural rendering for efficiency. INSTA [50] maps query points to a canonical space via nearest-triangle search on a FLAME [24] mesh and couples this with InstantNGP [31] for fast rendering. NeRFBlendShape [9] models dynamics by blending hash-grid features conditioned on 3DMM parameters. PointAvatar [48] represents heads as colored, deformable points and is effective for animation.

Recently, 3D Gaussian Splatting has emerged as a compelling explicit representation for head avatars, combining high fidelity with real-time rendering [18]. GaussianAvatars [35] rigs Gaussians to tracked FLAME meshes to obtain controllable, high quality head animation. Splatting Avatar [37], embed trainable Gaussians on a template mesh, initializing them by random surface sampling and refining them through a walking on triangles strategy. FlashAvatar [44] presents a lightweight mesh-embedded Gaussian field with residual offsets in UV-space. MeGA [42] mixes meshes and Gaussians by regions to leverage the strengths of each representation, achieving superior quality and enabling head editing after a short optimization stage. However, under tight bandwidth or compute budgets, these methods lack mechanisms to adapt level of detail at inference time, making them difficult to stream or to scale gracefully.

2.2. Level of Detail

Level of Detail traditionally balances scene complexity against performance in interactive graphics. A large body of work integrates LoD into classic pipelines [1, 2, 4, 11, 14, 25, 28, 29] to reduce memory traffic, stream-

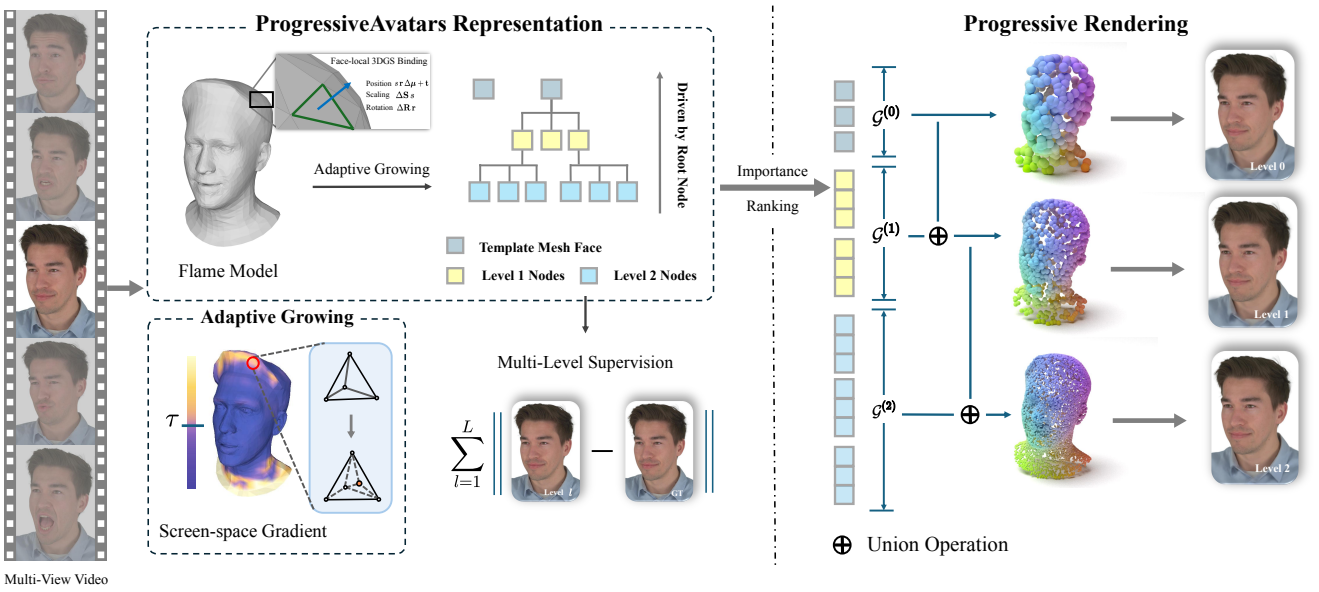


Figure 2. **Overview.** We take head video as input then recover a tracked FLAME mesh sequence. We bind 3D Gaussians to the local coordinate frame of each FLAME face. During training, screen-space gradients of the Gaussians drive implicit subdivision of the template mesh across multiple levels, yielding a triangle face forest. At rendering time, we precompute per-face importance score and progressively transmit and render the corresponding Gaussians in decreasing order of importance.

line rendering, and preserve responsiveness. In neural implicit reconstruction, NGLoD [40] organizes features in a sparse voxel octree whose depth indexes detail, and it enables continuous LOD and real-time SDF rendering by interpolating across levels and traversing the octree sparsely. Takikawa et al.[41] compress multi-resolution feature grids with a vector-quantized dictionary, which reduces memory by up to orders of magnitude and supports hierarchical streaming at variable bitrate. BungeeNeRF [45] trains progressively for extreme multi-scale scenes. It first fits distant views with a shallow base block, then appends new blocks as training proceeds while activating high-frequency channels in the positional encoding to reveal finer details. Tri-MipRF [16] prefilters the 3D feature space into three orthogonal mipmaps and performs cone-casting for area sampling, which delivers anti-aliased renderings with fast reconstruction. LoD-NeuS [49] adopts a multi-scale tri-plane representation and aggregates features within a conical frustum along each ray, recovering high-frequency surface detail while suppressing aliasing.

With advancements in 3DGS, researchers have started exploring modeling different LOD in explicit 3D Gaussian scenes [5, 6, 19, 22, 26, 36, 38, 39]. CityGaussian compresses trained large-scale 3DGS at multiple ratios by LightGaussian [5] to obtain LoD variants. 4DGF [6] integrated 3D Gaussians as an efficient geometry scaffold while utilizing neural fields as a compact and flexible appearance model. Octree-GS [36] organizes multi-scale Gaussians in an octree and selects levels dynamically by viewing distance and projected texture frequency, achieving rendering at stable frame rates. Hierarchical 3D Gaussians [19] and LODGE [22] choose levels based on the 2D projected scale of Gaussians for street-scale scenes, enabling LoD training and rendering. These methods typically pre-materialize all levels and, at runtime, select or switch levels from camera distance or primitive projected are. In contrast to prior work that mainly targets rendering throughput, we focus on streaming a 3DGS avatar. We aim get a usable head avatar appears quickly at a coarse LoD, and the remaining Gaussians are incrementally transmitted to refine quality with controllable compute and bandwidth.

A closely related line of work includes LoDAvatar [3] and ArchitectHead [46]. Neither method supports progressive rendering, since both require switching rendering resources. LoDAvatar embeds 3D Gaussian splats into a template mesh and performs uniform subdivision across levels, which tends to over-refine uninformative regions and waste computation. To partially address this issue, it applies a hand-crafted mask at render time to selectively densify specified areas and regulate the number of Gaussians. In contrast, our approach grows a multi-level representation through adaptive optimization, achieving a more favorable trade-off between visual fidelity and primitive count. ArchitectHead parameterizes Gaussians in a 2D UV feature space and defines a UV feature field composed of multi-level learnable maps. A neural decoder then maps

the latent features to 3D Gaussian attributes for rendering. Although this design offers a tunable balance between fidelity and efficiency, its network-based formulation causes the frame rate to deteriorate rapidly as the number of Gaussians increases, which in turn limits applicability in production settings when compared with purely explicit pipelines.

3. Method

As shown in Fig. 2, ProgressiveAvatars takes video records of a human head as input. For each time step t, we fit a FLAME [24] mesh $M_t = (\mathbf{V}_t, \mathbf{F})$ using a photometric multi-view tracker, where the vertex positions $\mathbf{V}_t$ vary with expressions and poses while the mesh topology $\mathbf{F}$ remains fixed over time. Leveraging this topological consistency, we build a progressive representation by implicitly subdividing the template topology to obtain a hierarchy $\{\mathbf{F}^{(\ell)}\}_{\ell=0}^{L}$, and we associate 3D Gaussians with faces at each level ℓ, where $\mathcal{G}^{(\ell)} = \bigcup_{f \in \mathbf{F}^{(\ell)}} \mathcal{G}_f^{(\ell)}$. We render images from $\mathcal{G}^{(\ell)}$ using a differentiable Gaussian rasterizer and supervise them with multi-view ground-truth images. This yields an animatable head avatar that supports progressive refinement. During training, the hierarchy grows on the fly via implicit subdivision guided by screen-space signals so detail concentrates where needed. During inference, we realize progressive rendering through incremental loading. Newly available Gaussians from finer levels are added while previously loaded content remains unchanged, and quality improves smoothly as more data arrives.

3.1. ProgressiveAvatars Representation

Our goal is to construct a progressive representation for head avatars. We grow a hierarchical tree on each template mesh face through implicit subdivision and bind 3D Gaussians to faces at each level. Gaussians are defined in the corresponding face-local frame so they remain stably animatable under varying expressions and poses and preserve identity across levels. At the coarsest level, base bindings cover all template faces, providing a complete global avatar before any fine-level refinement arrives.

Implicit Subdivision. Starting from $\mathbf{F}^{(0)} = \mathbf{F}$, we apply recursive triangle subdivisions to obtain $\{\mathbf{F}^{(\ell)}\}_{\ell=0}^{L}$. For a parent face $f = (i, j, k) \in \mathbf{F}^{(\ell)}$, a new child vertex $\mathbf{p}$ is created by barycentric interpolation of the parent vertices $\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k\}$:

$$\mathbf{p} = \beta_1 \mathbf{v}_i + \beta_2 \mathbf{v}_j + \beta_3 \mathbf{v}_k. \tag{1}$$

The barycentric coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$ are initialized to $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and are softly optimized during training under the simplex constraint $\beta_i \geq 0$ and $\sum_i \beta_i = 1$. This lets $\mathbf{p}$ move within the triangle and induces different effective scales and shapes per face so the hierarchy adapts to facial regions of different sizes and structures while keeping a consistent per-level construction. Under varying expressions and poses, the positions of subdivided points are computed at each level by the same barycentric mapping. These per-face hierarchical trees collectively form a forest over the template mesh.

Face-Local Gaussian Binding. On this hierarchy, we bind each 3D Gaussian to a triangle using a residual, face-local parameterization similar to GaussianAvatars [15]:

$$\mathbf{R} = \Delta\mathbf{R}\,\mathbf{r}, \qquad \mathbf{S} = \Delta\mathbf{S}\,s, \qquad \boldsymbol{\mu} = s\,\mathbf{r}\,\Delta\boldsymbol{\mu} + \mathbf{t}, \tag{2}$$

where $\Delta\mathbf{R}$, $\Delta\boldsymbol{\mu}$, and $\Delta\mathbf{S}$ are trainable residuals, $\mathbf{r}$ is a rotation aligned with the triangle normal, $\mathbf{t}$ is the triangle centroid, and s is a face scale computed as the mean of the three edge lengths. This binding makes Gaussians co-move with their binding faces under varying expressions and poses and preserves consistent appearance across levels.

Progressive Rendering. ProgressiveAvatars is a natural representation for progressive transmission and rendering. After training we sort the face hierarchy and assign each face at every level an importance score W_i. The score determines the order of transmission and activation and is obtained from aggregated screen-space gradients and view statistics collected during training, as described in Sec. 3.2. At rendering time, the coarsest avatar is transmitted first, followed by a stream of refinement records. Each record carries the barycentric coordinates $\boldsymbol{\beta}$ of the subdivision point and the residual parameters $\Delta\mathbf{R}$, $\Delta\mathbf{S}$, and $\Delta\boldsymbol{\mu}$ for the newly introduced 3D Gaussians. The receiver incrementally adds these Gaussians as records arrive and drives them with the template mesh by recursively evaluating the barycentric mapping from the root face to the target face. Crucially, this realizes continuous detail accumulation rather than discrete asset switching. This progressive rendering has the following advantages.

- **Progressive transmission.** Uses a priority-ordered stream guided by learned per-face scores, enabling bandwidth-adaptive refinement and faster time to a usable avatar without retransmitting earlier content.
- **Continuous detail accumulation.** Delivers pop-free and temporally stable transitions via additive refinement, and supports continuous view- and content-aware adjustment through selective subdivision.
- **Adaptive refinement.** Thanks to our hierarchical tree with inherited parent–child order on each template face, we can realize adaptive refinement. For example, we can apply adaptive subdivision only to facial regions such as the eyes or mouth. This further focuses resources on user-attended regions.

3.2. ProgressiveAvatars Construction

Adaptive Subdivision. We couple 3DGS adaptive density control with the hierarchical trees to build the structure on

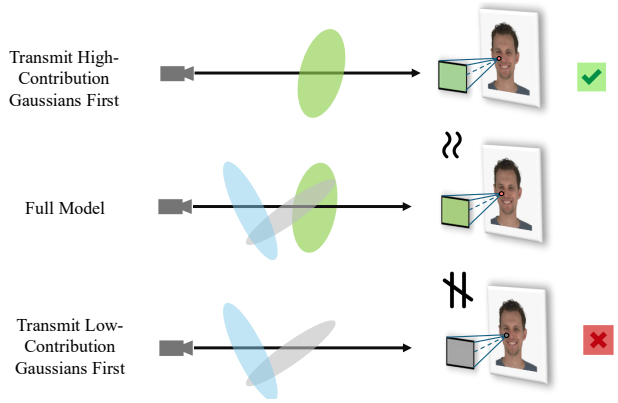


Figure 3. The center row shows the full model containing all 3D Gaussians within one level. Transmitting in descending importance makes early partial renderings closely match the full-model pixel color because dominant contributors arrive first. In contrast, sending low-importance Gaussians first re-normalizes partial weights and amplifies weak contributors, causing noticeable color drift from the full model. This motivates an importance-first schedule within each level for faithful progressive rendering.

the fly during training. We accumulate the screen-space gradient g_i only at the current finest level $\ell_{\max}$, since the finest structure most faithfully reflects whether the current representation has sufficient detail, and then grow the structure at a fixed step size k. Every k iterations, we select leaf faces satisfying $g_i > \varepsilon$ and subdivide them. For each new child face f', we attach a unique Gaussian as described in Sec. 3.1. This training–growth loop continues until the maximum level L is reached for each face. This strategy grows the per-face hierarchical trees in the most demanding fine-scale regions, yielding better marginal quality at a fixed budget.

Importance Ranking. After training, we linearize the multi-level hierarchy into a one-dimensional 3D asset stream to enable progressive transmission. Inspired by [5, 10], we compute an importance score for each face at every level and sort faces within the level accordingly. For face i, we define the score as the aggregated rendering contribution of its bound Gaussians over all pixels p:

$$W_i = \sum_{j \in \mathcal{G}_i} \sum_{p} \alpha_{j,p}\, T_{j,p}, \tag{3}$$

where $\mathcal{G}_i$ is the set of Gaussians bound to face i, $\alpha_{j,p}$ is the per-pixel opacity, and $T_{j,p}$ is the accumulated transmittance up to Gaussian j along the viewing ray through pixel p. In progressive transmission or continuous streaming settings, we prioritize higher scored faces and their Gaussians into the renderer. Fig. 3 shows that importance guided scheduling reduces color drift and suppresses artifacts, improving perceptual quality under tight bandwidth and compute.

3.3. Training

Multi-Level Supervision. We jointly supervise multiple levels to encourage cross-level consistency. Let $\mathcal{S}$ be the set of supervised levels, the photometric loss is

$$\mathcal{L}_{\text{rgb}} = \sum_{\ell \in \mathcal{S}} w_\ell \Big[(1 - \lambda_s)\, \mathcal{L}_1 + \lambda_s\, \mathcal{L}_{\text{ssim}} \Big], \tag{4}$$

with per-level supervision weights w_ℓ. We employ a coarse-to-fine optimization strategy that incrementally relaxes the upper bound on the per-face hierarchical subdivision depth. At initialization, the depth cap is set to 1. Every 50k iterations we increase this cap and invoke adaptive subdivision to expand the forest under the updated budget, continuing until the maximum depth D is reached.

Regularization. We further adopt the scaling loss $\mathcal{L}_{\text{scale}}$ and the position loss $\mathcal{L}_{\text{pos}}$ from [15] to constrain the relative position and scale with respect to its bound face, thereby preventing artifacts that occur when the 3D Gaussians move with the FLAME mesh.

Our final loss function is

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{scale}}\, \mathcal{L}_{\text{scale}} + \lambda_{\text{pos}} \mathcal{L}_{\text{pos}}. \tag{5}$$

4. Experiments

We evaluate our method on the NeRSemble dataset [21], which comprises multi-view videos for each subject with calibrated parameters for all 16 cameras. Following prior work [35, 42], we downsample images to 802×550 pixels and generate a foreground mask for each image. We adopt the same train/test split as GaussianAvatars [35]: 9 of the 10 expression sequences and 15 of the 16 cameras are used for training, with the remaining camera and the remaining expression sequence reserved for evaluation. All metrics are computed on pixels within the foreground mask.

Implementation Details. We use the Adam optimizer [20] for all parameter updates. Learning rates are set to 1×10^{-2} for the barycentric coordinates of subdivided points, 5×10^{-3} for positions, 2×10^{-2} for scales, and 1×10^{-3} for rotations. The FLAME parameters vertex offsets, joint rotations, and expression coefficients are also optimized. We set $\lambda_{\text{pos}} = 0.01$ and $\lambda_{\text{scale}} = 1.0$. The maximum hierarchy level is $D = 4$. Training runs for 60k iterations, with the hierarchy expanded adaptively every 2k iterations.

4.1. Reconstruction and Animation Results

We conduct comparative experiments with GaussianAvatars [35] and PointAvatar [48]. All baselines are trained from scratch using their public implementations on the same training data, camera calibration, and preprocessing as ours. We compare reconstruction quality at our 100% budget and 5% base budget to state-of-the-art approaches. We report self-reenactment and novel view synthesis in terms of PSNR, SSIM, and LPIPS.

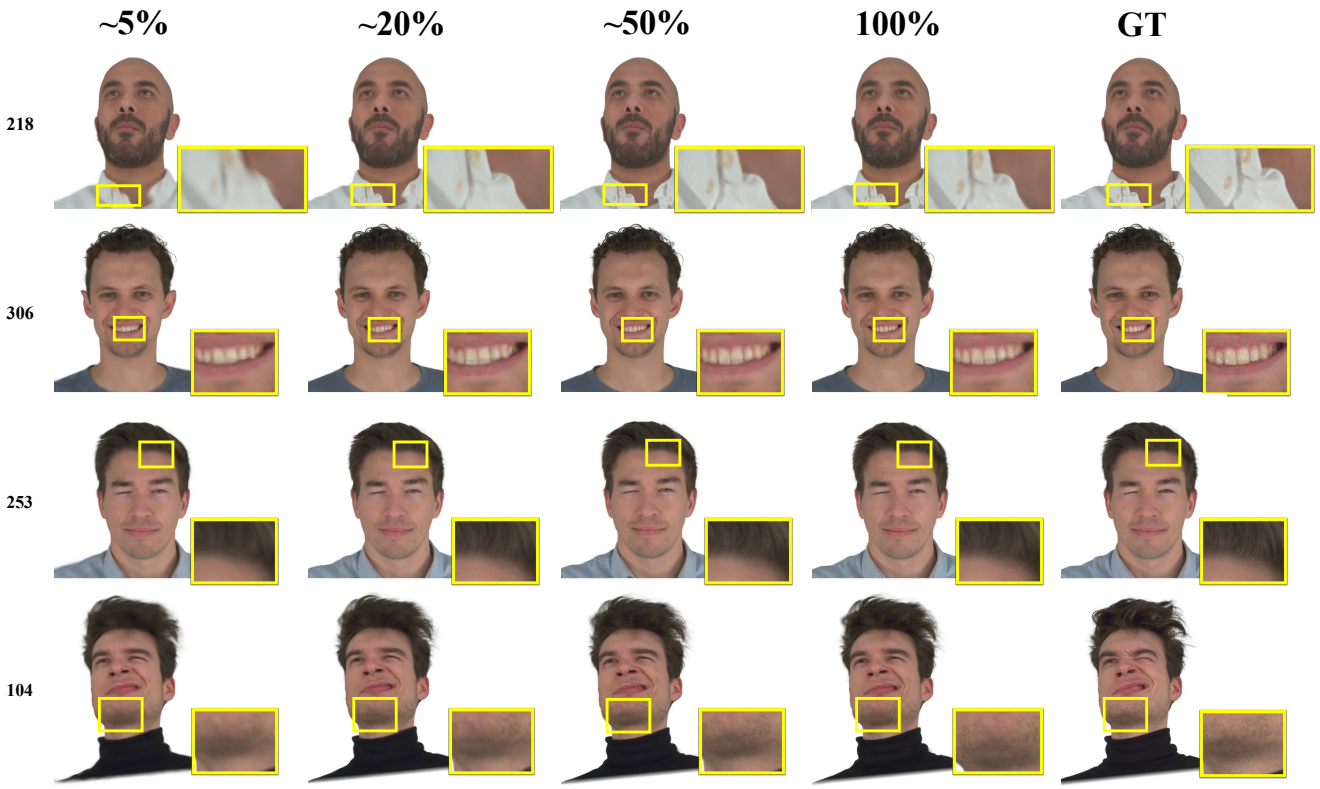


Figure 4. **Qualitative results on NeRSemble dataset across different transmission percentages.**

Table 1. **Performance comparison across varying transmission budgets.** We report Novel View Synthesis (NVS) and Novel Expression Synthesis (NES) using PSNR/SSIM/LPIPS. We also list the number of Gaussians, the amount of data to transmit (in Megabytes), and rendering speed. Rendering speed is measured on an RTX 4090 at 550×802 resolution.

	NVS			NES			#Gaussians	Transmit Data	FPS
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓			
GaussianAvatars	31.10	0.937	0.064	25.80	0.911	0.076	163,829	41.90 MB	271
5% (Base)	27.89	0.851	0.186	25.13	0.804	0.176	10,144	2.60 MB	291
25%	29.14	0.892	0.080	25.58	0.846	0.124	37,302	9.56 MB	278
50%	30.03	0.904	0.073	25.71	0.884	0.105	84,132	21.56 MB	258
100%	31.47	0.929	0.068	25.89	0.908	0.080	169,438	43.42 MB	260

As shown in Fig. 5, our method reconstructs sharper details in several regions, particularly around the neck, shoulders, and clothing. These areas are relatively coarsely tessellated in the FLAME template compared with high-saliency facial zones (e.g., the periocular region). Consequently, prior methods often allocate too few 3D Gaussians to these regions to faithfully capture their fine-scale detail. In contrast, our adaptive growing strategy increases the number of Gaussians and refines the hierarchy only where needed, making allocation insensitive to FLAME's non-uniform tessellation. Quantitative results in Tab. 2 show that our approach is on par with state-of-the-art methods, and even at a minimal 5% transmission budget, it yields a usable avatar suitable for bandwidth-constrained streaming.

4.2. Progressive Rendering Results

We emulate a practical streaming setting in which the avatar is transmitted and rendered progressively. At inference, the renderer begins from the base structure (a 5% data budget) and, within each structural level, activates refinement groups in descending order of a per-face importance

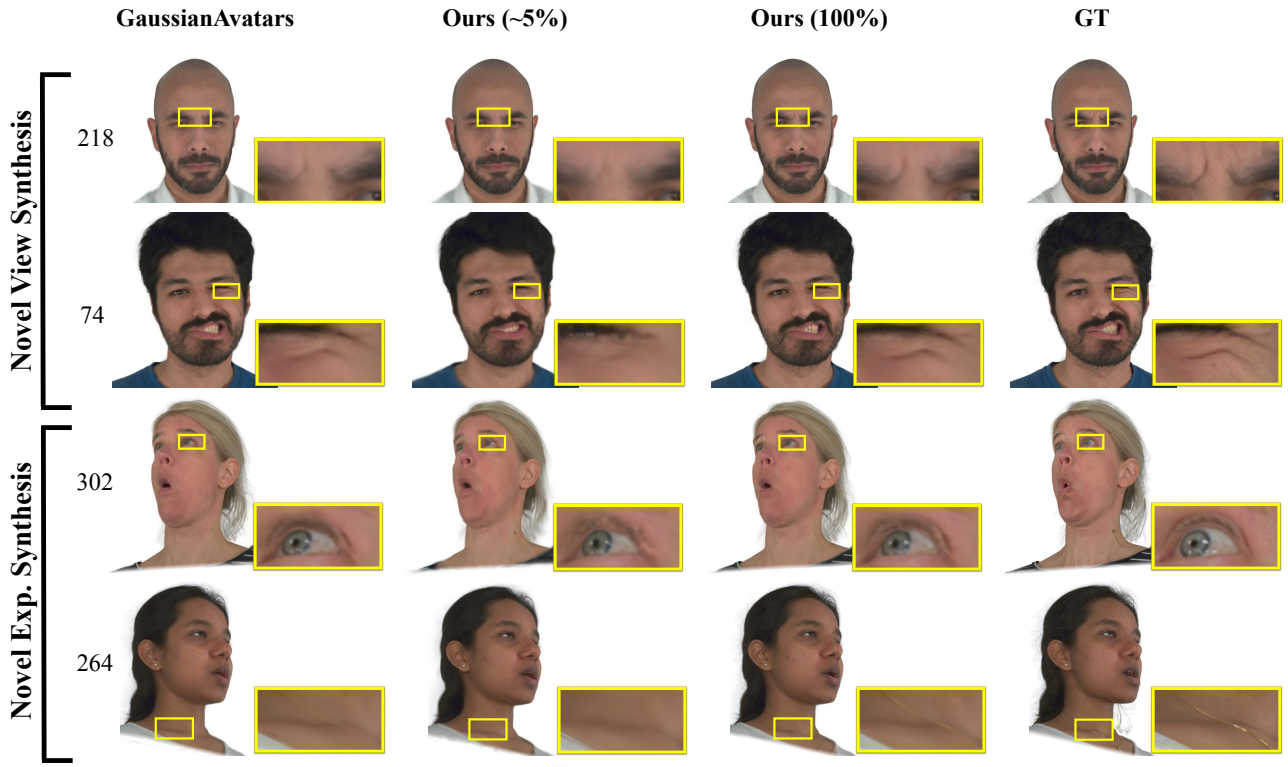


Figure 5. **Qualitative comparison with state-of-the-art methods.**

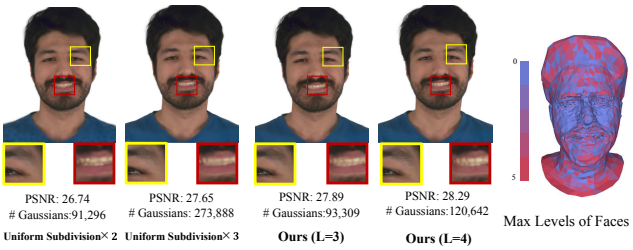


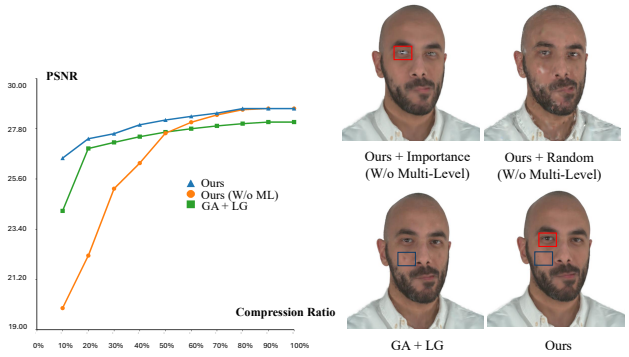
Figure 6. **Comparison of adaptive and uniform subdivision.** Right: visualization of per-face subdivision levels. High-frequency regions like facial hair receive more aggressive splitting, whereas smoother areas require substantially fewer subdivisions.

Table 2. **Quantitative comparison with SOTA methods.** We denote the best and second best scores in different colors.

	NVS			NES		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PointAvatar	25.8	0.893	0.097	23.4	0.884	0.102
GaussianAvatars	31.1	0.937	0.064	25.8	0.911	0.076
Ours (5%)	27.9	0.851	0.186	25.1	0.804	0.176
Ours (100%)	31.5	0.929	0.068	25.9	0.908	0.080

score until the next level is entered. This procedure induces a continuum of transmission percentages. For example, rendering at a 25% budget uses all level-1 Gaussians together with the top subset of level-2 Gaussians by importance, while preserving all previously transmitted content. After every activation step, we evaluate novel-view synthesis (NVS) and novel-expression synthesis (NES), and we record the number of active Gaussians, the amount of data that must be transmitted, and the achieved frame rate (FPS). Rendering speed is measured on an RTX 4090 at 550×802 resolution. Tab. 1 summarizes these measurements together with image quality. For direct reference to a widely used baseline, we additionally report GaussianAvatars in the same table.

Tab. 1 exhibits monotonic improvements as additional data arrive. With only 2.60 MB transmitted (5% budget), the avatar already attains reasonable quality. As higher-level Gaussians are streamed, fine structures such as shirt buttons, teeth, and hair gradually sharpen while temporal stability are maintained. At 100% transmission, our approach achieves rendering quality comparable to SOTA methods. Notably, the frame rates do not drop significantly, likely because the 3DGS workload has not yet saturated the GPU. However, multi-user VR scenarios can easily accumulate enough 3D Gaussians to hit GPU rasterization bottlenecks. In such computationally demanding cases, our method provides a critical advantage by flexibly balancing



(a) PSNR-compression comparison (b) Multi-level ablation

Figure 7. **Progressive streaming and multi-level ablation.**

primitive count and visual fidelity

We further evaluate the continuous streaming capability by comparing our method against a discrete compression pipeline: GaussianAvatars combined with LightGaussian (GA+LG). As Fig. 7(a) shows, GA+LG yields discrete operating points and demands 227.2 MB to store 10 levels. In contrast, our single progressive asset requires only 43.4 MB while supporting continuous, arbitrary-rate rendering and smooth quality refinement as data arrives.

4.3. Ablation Study

Adaptive Growing. We uniformly subdivide the initial mesh two and three times to highlight the importance of adaptive growing. For the uniform baseline, we recursively subdivide the template mesh two or three times at the beginning of training and then disable adaptive growing. As shown in Fig. 6, compared with uniform subdivision, our adaptive growing strategy achieves higher reconstruction quality while using fewer 3D Gaussians.

Multi-Level. To verify the benefits of multi-level over importance ranking alone, we evaluate an importance-only variant at a 50% transmission budget. As Fig. 7(b) shows, relying solely on importance ranking causes local coverage holes. In contrast, our multi-level design guarantees global base-level coverage for smooth progressive rendering.

Supervision on Multi-Level. As shown in Fig. 8 and Tab. 3, supervising only the finest level while ignoring intermediate ones prevents lower levels from learning a complete head, which conflicts with our goal of quickly streaming a usable head under bandwidth constraints. In "W/ Freeze", we freeze the parameters of previous level every 50k iterations, optimizing only the newly added level. This stage-wise schedule forces high-resolution 3D Gaussians to reconstruct residual details on fixed, low-detail ones, reducing the model's degrees of freedom and yielding lower quality than joint training where all levels are optimized.

Importance Ranking. In "W/ Random" of Tab. 3, we randomly permute the 3DGS within each level and load them in that order under the current bandwidth budget. Compared with randomly ordering the 3DGS at each level for progressive transmission and rendering, our importance ranking sorts Gaussians by their contribution to the rendered image, ensuring that high contribution Gaussians are loaded first and yielding finer rendering quality.

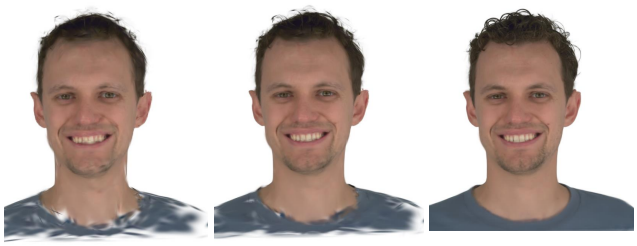


Figure 8. **Ablation study on multi-level supervision.**

Table 3. **Ablation study across all subjects.** We report average Novel View Synthesis (NVS) and Novel Expression Synthesis (NES) metrics over all subjects. "W/o MLS" supervises only the finest level, whereas "W/ MLS" supervises all levels.

		NVS			NES		
Strategy	Budget	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
W/o MLS	35%	20.06	0.812	0.284	19.86	0.798	0.327
W/ MLS	35%	29.87	0.897	0.076	25.65	0.859	0.114
W/ Freeze	100%	31.08	0.925	0.090	25.64	0.901	0.080
W/ MLS	100%	31.47	0.929	0.068	25.89	0.908	0.080
W/ Random	25%	28.40	0.867	0.109	25.46	0.838	0.143
W/ Ranking	25%	29.14	0.892	0.080	25.58	0.846	0.124

5. Conclusion

We proposed ProgressiveAvatars, a progressive, animatable 3D Gaussian avatar representation that enables progressive rendering of 3D avatars. By growing a mesh-anchored hierarchy via adaptive implicit subdivision and binding Gaussians in face-local frames, our representation supports incremental loading, progressive transmission, adaptive refinement, and progressive rendering. A usable avatar appears quickly and improves smoothly as additional data arrives, without replacing or deleting previously transmitted content. Experiments demonstrated strong time-to-quality under tight budgets and parity with state-of-the-art reconstruction at full detail. We believe ProgressiveAvatars plays an important role in progressive avatar streaming, a capability that will be vital for bringing 3DGS content to end users across heterogeneous networks and devices. While our work focuses on digital head avatars, the proposed progressive, mesh-anchored Gaussian hierarchy is generic and can be readily extended to general-purpose scenarios, such as progressive streaming and rendering of 3D assets and 4D volumetric video.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No.62272433), Anhui Provincial Natural Science Foundation (No.2508085ZD011) and the Fundamental Research Funds for the Central Universities.

References

[1] James H Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10): 547–554, 1976. 2

[2] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre, and Elmar Eisemann. Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 15–22, 2009. 2

[3] Xiaonuo Dongye, Hanzhi Guo, Haiyan Jiang, and Dongdong Weng. Adaptive levels of detail for human gaussian splats with hierarchical embedding. *2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2024. 2, 3

[4] Mark Duchaineau, Murray Wolinsky, David E Sigeti, Mark C Miller, Charles Aldrich, and Mark B Mineev-Weinstein. Roaming terrain: Real-time optimally adapting meshes. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, pages 81–88. IEEE, 1997. 2

[5] Zhiwen Fan, Kevin Wang, Zhangyang Wang, Kairun Wen, Dejia Xu, and Zehao Zhu. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in Neural Information Processing Systems*, 37, 2024. 2, 3, 5

[6] Tobias Fischer, Peter Kontschieder, Jonas Kulhanek, Marc Pollefeys, Lorenzo Porzi, and Samuel Rota Bulò. Dynamic 3d gaussian fields for urban areas. *Advances in Neural Information Processing Systems*, 37, 2024. 3

[7] Yun Fu, Renxiang Li, Thomas S Huang, and Mike Danielsen. Real-time multimodal human–avatar interaction. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(4):467–477, 2008. 2

[8] 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)*, 2021. 2

[9] 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

[10] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024. 5

[11] Stefan Guthe, Michael Wand, Julius Gonser, and Wolfgang Straßer. Interactive rendering of large volume data sets. In *IEEE Visualization, 2002. VIS 2002.*, pages 53–60. IEEE, 2002. 2

[12] Andrew Hogue, Sunbir Gill, and Michael Jenkin. Automated avatar creation for 3d games. In *Proceedings of the 2007 conference on Future Play*, pages 174–180, 2007. 2

[13] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[14] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 99–108, New York, NY, USA, 1996. Association for Computing Machinery. 2

[15] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 634–644, 2024. 1, 4, 5

[16] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19774–19783, 2023. 3

[17] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):1–14, 2015. 2

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[19] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 5

[22] Jonas Kulhanek, Marie-Julie Rakotosaona, Fabian Manhardt, Christina Tsalicoglou, Michael Niemeyer, Torsten Sattler, Songyou Peng, and Federico Tombari. Lodge: Level-of-detail large-scale gaussian splatting with efficient rendering. *arXiv preprint arXiv:2505.23158*, 2025. 3

[23] Wing Ho Leung, Belle L Tseng, Zon-Yin Shae, Ferdinand Hendriks, and Tsuhan Chen. Realistic video avatar. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, pages 631–634. IEEE, 2000. 2

[24] 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, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 4

[25] Peter Lindstrom and Valerio Pascucci. Visualization of large terrains made easy. In *Proceedings Visualization, 2001. VIS'01.*, pages 363–574. IEEE, 2001. 2

[26] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. *European Conference on Computer Vision*, 2024. 2, 3

[27] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. 2

[28] Frank Losasso and Hugues Hoppe. Geometry clipmaps: terrain rendering using nested regular grids. In *ACM SIGGRAPH 2004 Papers*, pages 769–776, New York, NY, USA, 2004. Association for Computing Machinery. 2

[29] David Luebke. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003. 2

[30] 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

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2

[32] Bo Peng, Yunfan Tao, Haoyu Zhan, Yudong Guo, and Juyong Zhang. Pica: Physics-integrated clothed avatar. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 1

[33] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1

[34] Zhexi Peng, Kun Zhou, and Tianjia Shao. Gaussian-plus-sdf slam: High-fidelity 3d reconstruction at 150+ fps. *Computational Visual Media*, 11(6):1195–1208, 2025. 1

[35] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 1, 2, 5

[36] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2, 3

[37] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024. 2

[38] Kaiwen Song, Xiaoyi Zeng, Chenqu Ren, and Juyong Zhang. City-on-web: Real-time neural rendering of large-scale scenes on the web. In *European Conference on Computer Vision*, pages 385–402. Springer, 2024. 3

[39] Kaiwen Song, Jinkai Cui, Zherui Qiu, and Juyong Zhang. Structuredfield: Unifying structured geometry and radiance field. *arXiv preprint arXiv:2501.18152*, 2025. 3

[40] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 3

[41] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*. Association for Computing Machinery, 2022. 3

[42] Cong Wang, Di Kang, Heyi Sun, Shenhan Qian, Zixuan Wang, Linchao Bao, and Song-Hai Zhang. Mega: Hybrid mesh-gaussian head avatar for high-fidelity rendering and head editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26274–26284, 2025. 1, 2, 5

[43] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3d gaussian splatting. *Computational Visual Media*, pages 1–30, 2024. 1

[44] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2

[45] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022. 3

[46] Peizhi Yan, Rabab Ward, Qiang Tang, and Shan Du. Architecthead: Continuous level of detail control for 3d gaussian head avatars. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1632–1642, 2026. 2, 3

[47] Keyang Ye, Tianjia Shao, and Kun Zhou. When gaussian meets surfel: Ultra-fast high-fidelity radiance field rendering. *ACM Transactions on Graphics*, 2025. 1

[48] 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 (CVPR)*, pages 21057–21067, 2023. 2, 5

[49] Yiyu Zhuang, Qi Zhang, Ying Feng, Hao Zhu, Yao Yao, Xiaoyu Li, Yan-Pei Cao, Ying Shan, and Xun Cao. Anti-aliased neural implicit surfaces with encoding level of detail. *SIGGRAPH Asia 2023 Conference Papers*, 2023. 3

[50] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *CVPR*, pages 4574–4584, 2023. 2

[51] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. *2025 International Conference on 3D Vision (3DV)*, 2025. 1