One Avatar, Any Budget: Robust LoD for Dynamic Gaussian Avatars

Haoyu Zhao[1,2], Wenjuan Gong[1,2*], Junang Qi[1,2]
Haiyu Zhang[1,2], Wenjian Zhao[1,2], Zimu YOU[3]

[1]Qingdao Software College, College of Computer Science and Technology,
China University of Petroleum (East China), Qingdao, China
[2]Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software
[3]Faculty of Engineering and IT, University of Melbourne
[*]Corresponding author: wenjuangong@upc.edu.cn

Abstract

Cross-device deployment of a single reconstructed avatar often requires adaptation to varying Gaussian budgets with minimal degradation in visual quality. Motivated by the need for a robust budget-adaptive 3D representation, we propose RobustAvatar, a train-once Level-of-Detail (LoD) adaptation framework for dynamic Gaussian avatars based on sequential selection and compensation. RobustAvatar supports flexible compression-ratio control from a single trained avatar, without requiring ratio-specific fine-tuning or the maintenance of separate LoD models. The framework consists of two coordinated modules: Semantic Distribution-Regularized Selection (SDS) and Canonical Budget-Conditioned Compensation (CBC). SDS performs budget-aware Gaussian selection guided by semantic distribution regularization, preserving the coverage of perceptually salient facial regions in the full avatar under constrained budgets. CBC refines the selected primitives in a canonical, expression- and pose-neutral space by predicting ratio-conditioned attribute residuals, reducing detail loss from Gaussian removal and improving robustness to decreasing Gaussian budgets. Experiments on the NeRSemble dataset show that RobustAvatar achieves more robust budget adaptation than state-of-the-art methods, with slower quality degradation as Gaussian budgets decrease. The code will be made publicly available.

1 Introduction

High-fidelity dynamic head avatars are essential for social VR, telepresence, and multi-user virtual environments. A single reconstructed avatar often needs to adapt to varying memory, compute, bandwidth, and latency constraints across deployment environments. 3D Gaussian Splatting provides an explicit and real-time representation for such avatars Kerbl et al. [2023], and has recently become a common backbone for animatable head modeling Qian et al. [2024], Xu et al. [2024], Chen et al. [2024], Ma et al. [2024], Giebenhain et al. [2024]. The key difficulty is not merely to reduce the number of Gaussian primitives, but to preserve identity appearance, expression-sensitive facial details, and dynamic consistency as the available Gaussian budget decreases. This motivates the need for robust LoD control in dynamic Gaussian avatars, where the representation can adapt to continuous compression ratios. Yet existing methods typically operate at a fixed representation capacity and lack a mechanism to select and compensate Gaussians at reduced budgets. As a result, this can lead to noticeable rendering-quality degradation under tight budgets, making it harder to deploy the same avatar across diverse deployment settings.

Robust LoD control for dynamic Gaussian avatars remains challenging and underexplored. Prior UV-parameterized methods Dongye et al. [2025], Yan et al. [2025] enable continuous LoD control by resampling a 2D feature field, but this controls capacity through the UV parameterization rather

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

than through the spatial distribution of Gaussian primitives in canonical 3D space. Recent work Song et al. [2026] targets progressive transmission and realizes LoD by monotonically adding Gaussian primitives according to an importance-ordered hierarchy. Although this progressive formulation facilitates reuse of transmitted content, the low-budget representations are determined by prefixes of the learned hierarchy. It does not select a more suitable Gaussian subset for each target budget, nor does it explicitly optimize the attributes of retained Gaussians. These limitations raise a natural question: how can a single reconstructed Gaussian avatar support robust LoD control, keeping visual quality from degrading rapidly as the Gaussian budget decreases?

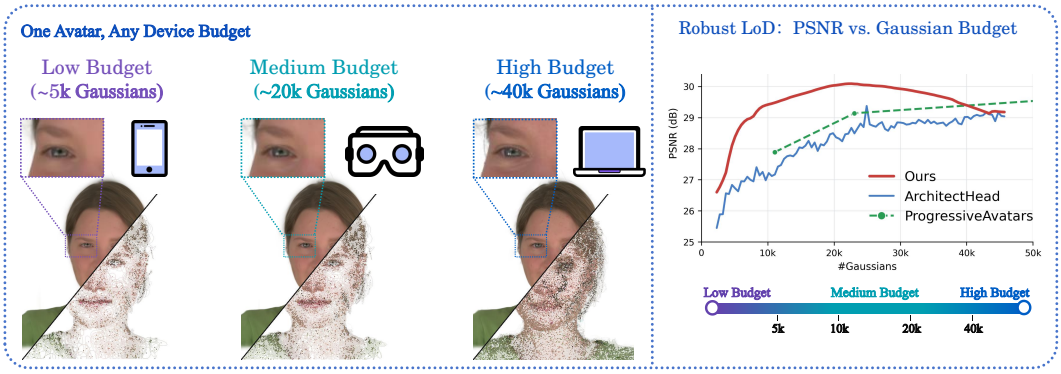

Figure 1: RobustAvatar is motivated by the need to deploy a single dynamic Gaussian avatar across devices with different memory and computation budgets. As shown by the qualitative(left) and quantitative(right) results, RobustAvatar alleviates the rapid visual quality degradation caused by reducing the number of Gaussians.

To address this question, we propose RobustAvatar, a learning framework for LoD adaptation of dynamic Gaussian avatars under varying Gaussian budgets. Unlike existing methods, RobustAvatar directly selects a budget-specific Gaussian subset and compensates the selected Gaussians for each target budget. Specifically, we leverage FLAME to define a canonical space, where Gaussian primitives maintain stable semantic associations and dynamic consistency under expression changes and head motion. To this end, we introduce Semantic Distribution-Regularized Selection (SDS) and Canonical Budget-Conditioned Compensation (CBC): the former learns budget-aware Gaussian selection with semantic distribution constraints, while the latter performs attribute compensation for the selected Gaussian primitives in canonical space. Together, SDS and CBC allow the selected Gaussian primitives to preserve the semantic distribution of the full avatar and compensate for details lost at lower Gaussian budgets, leading to slower quality degradation as the budget decreases. Experiments show that this design improves budget robustness over existing LoD baselines, supporting deployment across settings with different resource constraints, as shown in Figure 1.

In summary, the main contributions of this work are as follows:

- We propose RobustAvatar, a learning framework that enables robust LoD adaptation of dynamic Gaussian avatars through sequential selection and compensation. Given a reconstructed avatar and a target Gaussian budget, RobustAvatar selects an effective subset of Gaussian primitives and compensates the selected subset, enabling flexible budget control without fine-tuning for each ratio or maintaining multiple LoD models.
- We introduce Semantic Distribution-Regularized Selection and Canonical Budget-Conditioned Compensation. SDS preserves facial semantic distributions during budget-aware selection, while CBC performs compensation in canonical 3D Gaussian space to reduce degradation and improve dynamic consistency.

2 Related Work

Animatable Head Avatars. Animatable head avatars have progressed from implicit neural representations to explicit Gaussian-based models. Early methods reconstruct dynamic heads with neural radiance fields or volumetric avatar representations Gafni et al. [2021], Grassal et al. [2022],

Bai et al. [2023], while FLAME Li et al. [2017] provides a common scaffold for facial shape, expression, and pose. With 3D Gaussian Splatting Kerbl et al. [2023], GaussianAvatars Qian et al. [2024], Gaussian Head Avatar Xu et al. [2024], MonoGaussianAvatars Chen et al. [2024], 3D Gaussian Blendshapes Ma et al. [2024], NPGA Giebenhain et al. [2024], and GAGA Chu and Harada [2024] show that Gaussian primitives can be rigged, deformed, or generated for high-fidelity and real-time head animation. Recent work further improves projection efficiency Feng et al. [2025], relighting Zhang et al. [2025a], online modeling Li et al. [2025a], mesh-Gaussian editing Wang et al. [2025a], full-head editing Zhang et al. [2025b], statistical head priors Zielonka et al. [2025a], dynamic appearance modeling Wang et al. [2025b], geometry adaptation Moon et al. [2025], hybrid surface-volumetric reconstruction Sun et al. [2025], hair modeling Kim et al. [2025], and generative animation Yu et al. [2025]. A parallel line reduces capture requirements or improves controllability with morphable multi-view diffusion Taubner et al. [2025], zero-shot or few-shot inversion Zhou et al. [2025], Zielonka et al. [2025b], large animatable reconstruction models Kirschstein et al. [2025], personalized or one-shot generation Cha et al. [2025], Chen et al. [2025], Li et al. [2025b], and lifelike audio-driven or talking-head animation Xu et al. [2025], Li et al. [2025c], Zhen et al. [2025], Tan et al. [2025], Hu et al. [2025b], Teotia et al. [2025]. These works advance avatar reconstruction, generation, editing, and animation, but they typically operate at a chosen representation capacity rather than studying how one reconstructed dynamic Gaussian avatar can remain robust under arbitrary Gaussian budgets.

Level of Detail. Level-of-detail methods adapt representation cost to memory, bandwidth, and rendering budgets. For static 3DGS, LightGaussian Fan et al. [2024], Compact 3DGS Lee et al. [2024], and memory-footprint reduction Papantonakis et al. [2024] compress Gaussian parameters, while LP-3DGS Zhang et al. [2024], PUP 3D-GS Hanson et al. [2025b], MaskGaussian Liu et al. [2025c], and Speedy-Splat Hanson et al. [2025a] prune or sparsify primitives; CodecGS Lee et al. [2025], FlexGS Liu et al. [2025b], and LOD-GS Shen et al. [2025] further explore codec-style compression, flexible model-size control, or structured LoD. However, these methods mainly target static or general scenes, where global importance criteria do not need to preserve facial semantics, expression-sensitive details, or deformation consistency. For dynamic Gaussian content, 4DGC Hu et al. [2025a] studies rate-aware 4D Gaussian compression, and HGC-Avatar Tang et al. [2025] builds hierarchical compression for streamable dynamic avatars; however, their emphasis is on bitrate, streaming, or hierarchical coding rather than budget-specific selection and compensation for a reconstructed head avatar. For digital humans, URAvatar Li et al. [2024], LUCAS Liu et al. [2025a], SqueezeMe Iandola et al. [2025], and HRM^{2}Avatar Shi et al. [2025] study codec priors, layered representations, distillation, or mobile-ready deployment, while LoDAvatar Dongye et al. [2025], ArchitectHead Yan et al. [2025], and ProgressiveAvatars Song et al. [2026] provide avatar-specific LoD through UV/template parameterization or additive progressive hierarchies. However, these approaches either redesign the avatar representation, tie budget control to UV/template structures, or rely on nested subsets, leaving open robust selection and compensation of native 3D Gaussian primitives under any target budget.

3 Method

The overview of RobustAvatar is shown in Figure 2. RobustAvatar aims to learn a single ratio-conditioned model on top of an already reconstructed dynamic Gaussian avatar, enabling continuous LoD control over Gaussian budgets at inference time while making the budget-quality curve more robust as the available Gaussian budget decreases. Instead of treating LoD control as a simple primitive removal problem Song et al. [2026], our goal is to slow down the degradation of identity appearance, expression-sensitive facial details, and dynamic consistency under reduced Gaussian budgets.

Let the Gaussian primitive set of the complete input model, referred to as the full avatar, be

$$\mathcal{G} = \{g_i\}_{i=1}^{N}, \qquad g_i = (\mathbf{x}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{h}_i, b_i), \tag{1}$$

where $\mathbf{x}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{h}_i$ denote the position, log-scale, rotation, opacity logit, and SH/color coefficients of the i-th Gaussian, respectively, and b_i denotes its bound FLAME face. Given a target budget ratio $r \in (0, 1]$, RobustAvatar renders approximately rN Gaussians while preserving visual quality close to that of the full avatar.

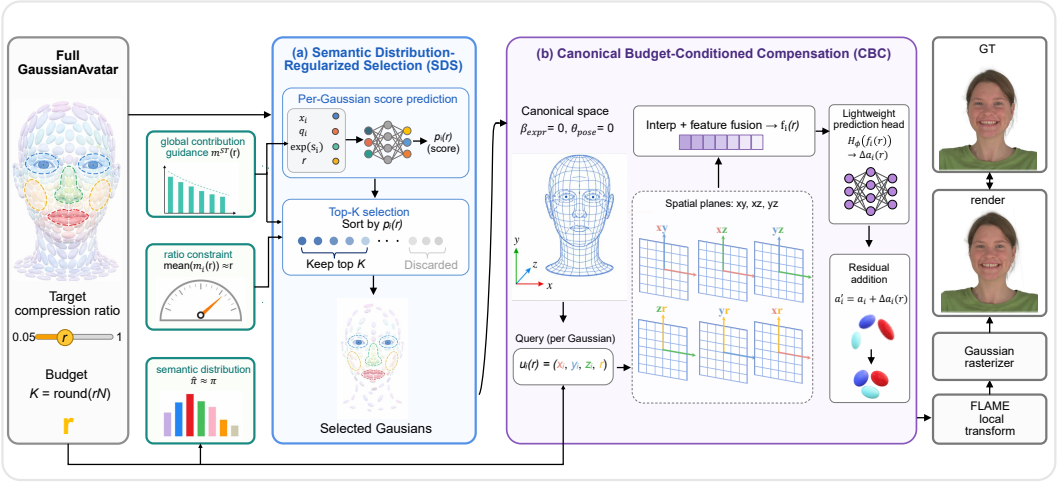

Figure 2: Overview of RobustAvatar. SDS selects a budget-specific Gaussian subset under semantic distribution regularization, while CBC compensates the retained primitives in canonical space with ratio-conditioned attribute residuals. The compensated subset is then transformed to posed avatar space and rasterized, enabling continuous Gaussian-budget control while preserving visual details under low budgets.

To this end, we formulate LoD control as a learning-based budget-conditioned adaptation problem. RobustAvatar consists of two core modules: Semantic Distribution-Regularized Selection (SDS) and Canonical Budget-Conditioned Compensation (CBC). For each ratio r, SDS first learns, through Semantic Distribution-Regularized Selection, to select a more suitable Gaussian subset under the current budget ratio. CBC then applies a budget-conditioned network to adapt the attributes of the retained Gaussians, allowing these retained primitives to compensate for the geometric support and appearance details lost from the removed Gaussians under low budgets. Finally, RobustAvatar learns to adapt the Gaussians available at each budget, thereby slowing down quality degradation as the budget decreases.

3.1 Semantic Distribution-Regularized Selection (SDS)

In a dynamic head avatar, low-budget selection should not be determined only by global rendering contribution. Regions such as the eyes, lips, nose, and eyelids may contain relatively few Gaussians, but they are crucial for identity preservation and expression perception. If only the globally most contributive primitives are retained, the low-budget subset may over-concentrate on a few visually dominant regions, causing rapid degradation of local facial structure and expression-sensitive details.

Therefore, SDS formulates Gaussian selection as a semantic distribution-regularized budgeted subset learning problem. For a target budget ratio r, the model predicts the retention probability of each Gaussian:

$$p_i(r) = S_\theta(\mathbf{x}_i, \mathbf{q}_i, \exp(\mathbf{s}_i), r), \tag{2}$$

where S_θ is a lightweight budgeted subset prediction network. During training, $m^{\mathrm{ST}}(r) \in \{0,1\}^N$ denotes the straight-through binary keep mask obtained from the keep/drop logits by hard Gumbel-Softmax, while $p_i(r)$ is the corresponding soft keep probability. We constrain the selected Gaussians such that their distribution over facial semantic regions remains close to the semantic distribution of the full avatar. At inference time, we discard the stochastic mask and sort Gaussians according to $p_i(r)$, retaining the top-K Gaussians, where $K = \max(1, \lfloor rN \rceil)$.

To learn selection strategies under different budgets, we estimate the contribution of each Gaussian from the rendering process of the full avatar. This full-avatar contribution is determined by the Gaussian's visibility, opacity, and screen-space footprint in full-avatar rendering, and is accumulated across training views. For a given budget ratio r, the top-K Gaussians with the highest full-avatar contribution form a binary global importance guidance mask $m^{\mathrm{full}}(r) \in \{0,1\}^N$.

Each Gaussian inherits a FLAME semantic region label $\ell_i \in \{1, \ldots, R\}$ through its bound face b_i. Let w_i^{full} denote the full-avatar contribution of the i-th Gaussian. We normalize the average contribution of the full avatar in the k-th semantic region into a target region distribution:

$$\pi_k = \frac{\frac{1}{n_k} \sum_{i:\ell_i=k} w_i^{\text{full}}}{\sum_{j=1}^{R} \frac{1}{n_j} \sum_{i:\ell_i=j} w_i^{\text{full}}}, \tag{3}$$

where n_k is the number of Gaussians in the k-th region. This distribution characterizes the relative importance of different facial regions in the full avatar.

Given the retention probabilities $p_i(r)$ under the current budget ratio, the soft selected region distribution is defined as

$$\hat{\pi}_k(r) = \frac{\sum_{i:\ell_i=k} p_i(r)}{\sum_{j=1}^{R} \sum_{i:\ell_i=j} p_i(r)}. \tag{4}$$

The budgeted subset learning objective is

$$\mathcal{L}_{\text{sub}} = \lambda_{\text{imp}} \| m^{\text{ST}}(r) - m^{\text{full}}(r) \|_1 + \lambda_{\text{ratio}} \left| \frac{1}{N} \sum_i m_i^{\text{ST}}(r) - r \right| + \lambda_{\text{sem}} \| \hat{\pi}(r) - \pi \|_1. \tag{5}$$

This objective imposes three constraints simultaneously: retaining high-contribution Gaussians from the full avatar, satisfying the target budget ratio, and preserving the semantic-region importance distribution of the full avatar.

3.2 Canonical Budget-Conditioned Compensation (CBC)

Primitives in a dynamic Gaussian avatar move across frames under expression changes, head pose, and FLAME-driven deformation. We refer to the frame-dependent coordinate domain after applying expression coefficients, pose coefficients, and head motion as the posed avatar space. In this space, the world-space position and orientation of the same Gaussian change with the expression and pose of each frame. Therefore, directly learning budget compensation in posed avatar space would entangle budget adaptation with frame-dependent motion, weakening dynamic consistency and causing temporal flickering.

We therefore define a canonical space. Specifically, we set both the FLAME expression coefficients and pose coefficients to 0, obtaining the canonical FLAME state. This state removes frame-dependent motion factors introduced by expression deformation and pose-driven motion. We refer to the coordinate domain that represents Gaussian attributes in the FLAME face-local frame under this canonical state as the canonical space, denoted by Ω_c. When rendering an arbitrary frame, the canonical local attributes are transformed to posed avatar space through the local coordinate transform of the corresponding FLAME face under the current expression and pose, and are then passed to the Gaussian rasterizer.

Under low-budget settings, removing unselected Gaussians can cause insufficient local geometric support, broken boundaries, blurred expression-sensitive regions, and appearance drift. Thus, after obtaining a suitable Gaussian subset under the current budget, CBC further adapts the attributes of the retained Gaussians in Ω_c, allowing retained primitives to compensate for the local representation capacity originally carried by the removed Gaussians. Specifically, we define a canonical-ratio field:

$$\Psi_\phi : \Omega_c \times [r_{\min}, 1] \to \mathbb{R}^C, \qquad f_i(r) = \Psi_\phi(\mathbf{x}_i, r), \tag{6}$$

where $\mathbf{x}_i = (x_i, y_i, z_i)$ is the position of the Gaussian in the canonical space.

Directly storing a dense four-dimensional field is expensive. We factorize Ψ_ϕ into six two-dimensional feature planes:

$$\mathcal{P} = \{xy, xz, yz, xr, yr, zr\}. \tag{7}$$

Given a query point $u_i(r) = (x_i, y_i, z_i, r)$, we perform bilinear interpolation on each plane and fuse the resulting features to obtain a budget-conditioned feature:

$$f_i(r) = F_{\text{fuse}} \left(\{ \text{Interp}\left(P_{ab}, u_{i,a}, u_{i,b}\right) \}_{(a,b)\in\mathcal{P}} \right). \tag{8}$$

The spatial planes $\{xy, xz, yz\}$ capture the canonical geometric structure, while the ratio planes $\{xr, yr, zr\}$ capture how the compensation at the same canonical location changes with the target budget. A lightweight prediction head then predicts full-attribute residuals from $f_i(r)$:

$$\Delta a_i(r) = H_\phi\big(f_i(r)\big), \qquad \Delta a_i(r) = (\Delta \mathbf{x}_i, \Delta \mathbf{s}_i, \Delta \mathbf{q}_i, \Delta o_i, \Delta \mathbf{h}_i). \tag{9}$$

The compensated Gaussian attributes are

$$(\mathbf{x}_i', \mathbf{s}_i', \mathbf{q}_i', o_i', \mathbf{h}_i') = (\mathbf{x}_i + \Delta \mathbf{x}_i,\ \mathbf{s}_i + \Delta \mathbf{s}_i,\ \mathrm{normalize}(\mathbf{q}_i + \Delta \mathbf{q}_i),\ o_i + \Delta o_i,\ \mathbf{h}_i + \Delta \mathbf{h}_i). \tag{10}$$

CBC is the key to the robustness of RobustAvatar. As the budget decreases, the retained Gaussians learn to provide additional geometric support and appearance compensation for missing regions, thereby reducing the impact of removed primitives on the final rendering quality.

3.3 Training Objective and Inference

During training, we sample a target budget ratio r from a predefined budget set. The full avatar renders the reference image I^{full}. RobustAvatar first selects the Gaussian subset under the current budget, then applies budget-conditioned compensation to the retained Gaussians, and finally renders the image I^r. The reconstruction loss is

$$\mathcal{L}_{\text{rec}} = (1 - \lambda_{\text{ssim}}) \|I^r - I^{\text{full}}\|_1 + \lambda_{\text{ssim}}(1 - \text{SSIM}(I^r, I^{\text{full}})). \tag{11}$$

To prevent low-budget adaptation from overly altering the identity structure of the avatar, we introduce preservation and regularization terms. The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{sub}} + \lambda_{\text{pres}} \mathcal{L}_{\text{pres}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}. \tag{12}$$

Here, $\mathcal{L}_{\text{pres}}$ constrains full-avatar rendering and the local attributes of compensated Gaussians from deviating from the full avatar, while $\mathcal{L}_{\text{reg}}$ includes geometric range constraints and smoothness regularization for the budget-conditioned feature field.

At inference time, given an arbitrary target budget ratio r, RobustAvatar first selects the top-K Gaussians under the current budget and directly discards the remaining primitives. It then applies budget-conditioned compensation and rasterization only to the retained subset. Therefore, the same avatar representation can be deployed continuously under different budgets, while the actual number of rendered primitives is strictly controlled by r.

4 Experiments

4.1 Experiment Settings

Implementation Details. We use the Adam optimizer Kingma and Ba [2015] for all parameter updates. The full avatar is initialized from a 60k-iteration GaussianAvatars Qian et al. [2024] model. For each scene, we train our model for 60k iterations on a single NVIDIA RTX 4090 GPU. Learning rates are set to 5×10^{-4} for Gaussian positions, 2.5×10^{-3} for SH features, 5×10^{-2} for opacity, 1×10^{-2} for scales, and 1×10^{-3} for rotations. Both the budget-conditioned compensation field in CBC and the subset prediction network in SDS are optimized with a learning rate of 1×10^{-4}.

We set $\lambda_{\text{imp}} = 0.5$, $\lambda_{\text{ratio}} = 0.01$, $\lambda_{\text{sem}} = 0.2$, and $\lambda_{\text{pres}} = 1.0$. During training, the budget condition is randomly sampled from $\{10\%, 30\%, 60\%, 90\%\}$ at each iteration, and at inference time the same trained model is queried at deployment ratios of $\{5\%, 10\%, 20\%, 50\%\}$ through top-K Gaussian selection.

Datasets. We evaluate our method under the preprocessed GaussianAvatars Qian et al. [2024]/NeRSemble Kirschstein et al. [2023] setting. For each subject, we use the calibrated camera parameters, white-background preprocessing, foreground masks, and train/validation/test splits provided by the GaussianAvatars data pipeline. A GaussianAvatars model trained for 60k iterations is used as the full avatar for each subject.

Baselines and Metrics. We compare RobustAvatar with ArchitectHead and ProgressiveAvatars. ArchitectHead is evaluated under the same input setting as ours, where both methods use the full GaussianAvatars trained for 60k iterations. Since ProgressiveAvatars does not release its code, we cite its reported results and align the comparison by the actual number of rendered Gaussians. As the ratio definitions differ across methods, we report both the method-specific ratio and the rendered Gaussian count: ours denotes the retained proportion relative to the full avatar, ArchitectHead uses its own LoD control parameter, and ProgressiveAvatars reports the progressive transmission ratio. We evaluate Novel View Synthesis (NVS) and Novel Expression Synthesis (NES) quality using PSNR, SSIM, and LPIPS, and assess temporal stability with optical-flow-based TWE, warped tLPIPS, and rTWE, computed on valid foreground regions after forward–backward flow consistency filtering.

4.2 Reconstruction and Animation Results

Comparison with ArchitectHead Across all tested budgets, RobustAvatar achieves better reconstruction and animation quality than ArchitectHead under comparable numbers of rendered Gaussians. As shown in Table 1, RobustAvatar improves NVS PSNR by approximately 2 dB and NES PSNR by approximately 7 dB on average, while using a similar number of Gaussians.

Comparison with GaussianAvatars As shown in Table 1, at the 50% retained-Gaussian budget, RobustAvatar matches or slightly exceeds the full GaussianAvatars on several metrics while rendering only half of the primitives. This does not indicate that primitive removal alone improves the original avatar; rather, the gains come from budget-conditioned adaptation. SDS selects a high-utility subset whose semantic coverage better matches the 50% budget, reducing the loss of expression-sensitive regions under naive pruning. CBC further adjusts the retained primitives in canonical space, allowing them to compensate for part of the geometric and appearance capacity carried by the removed Gaussians. Since 50% remains a moderate-capacity regime, the selected subset still provides enough support for CBC to refine geometry and appearance. The resulting compact avatar can therefore slightly outperform the unadapted full GaussianAvatars under the evaluation protocol. This effect is most visible at moderate budgets; under very low budgets, reduced representation capacity still leads to inevitable degradation.

Comparison with ProgressiveAvatars. ProgressiveAvatars is designed for monotonic progressive transmission, where low-budget representations are formed by accumulating Gaussian prefixes from a fixed hierarchy. As shown in Table 1, RobustAvatar achieves better NVS and NES quality under comparable Gaussian counts, and in some settings reaches comparable or better quality with fewer Gaussians. For example, RobustAvatar at 20% uses fewer Gaussians than ProgressiveAvatars at 5% while improving all reported NVS and NES metrics. This suggests that a fixed progressive hierarchy does not necessarily provide the most effective representation for each target budget, whereas budget-conditioned selection and compensation make more effective use of limited Gaussian budgets.

Table 1: Quantitative comparison under different Gaussian budgets. Different methods use different ratio definitions; therefore, we report the method-specific setting together with the actual number of rendered Gaussians. The missing entries are marked with "/" because ProgressiveAvatars is not publicly available and the corresponding temporal metrics are not reported in its paper.

Method / Setting	NVS			NES			Temporal			#Gaussians
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	TWE↓	tLPIPS↓	rTWE↓	
Ours (5%)	26.36	0.912	0.197	24.17	0.900	0.188	0.0413	0.0968	0.0366	2,307
Ours (10%)	**28.42**	**0.923**	**0.182**	**25.05**	**0.906**	**0.177**	**0.0381**	**0.0905**	**0.0317**	4,615
ArchitectHead (0.90)	26.18	0.881	0.201	17.83	0.799	0.284	0.0583	0.1453	0.0663	4,871
Ours (20%)	**29.47**	**0.929**	**0.164**	**25.54**	**0.910**	**0.164**	**0.0374**	**0.0872**	**0.0295**	9,230
ArchitectHead (0.75)	26.62	0.886	0.191	18.75	0.813	0.257	0.0579	0.1432	0.0658	8,788
ProgressiveAvatars (5%)	27.89	0.851	0.186	25.13	0.804	0.176	/	/	/	10,144
Ours (50%)	**29.72**	**0.930**	0.152	**25.67**	**0.910**	0.155	**0.0379**	**0.0877**	**0.0292**	23,076
ArchitectHead (0.45)	27.72	0.897	0.159	18.72	0.811	0.241	0.0590	0.1409	0.0668	23,133
ProgressiveAvatars (25%)	29.14	0.892	**0.080**	25.58	0.846	**0.124**	/	/	/	37,302
GaussianAvatars (100%)	29.19	0.927	0.153	25.55	0.909	0.154	0.0382	0.0887	0.0296	46,152

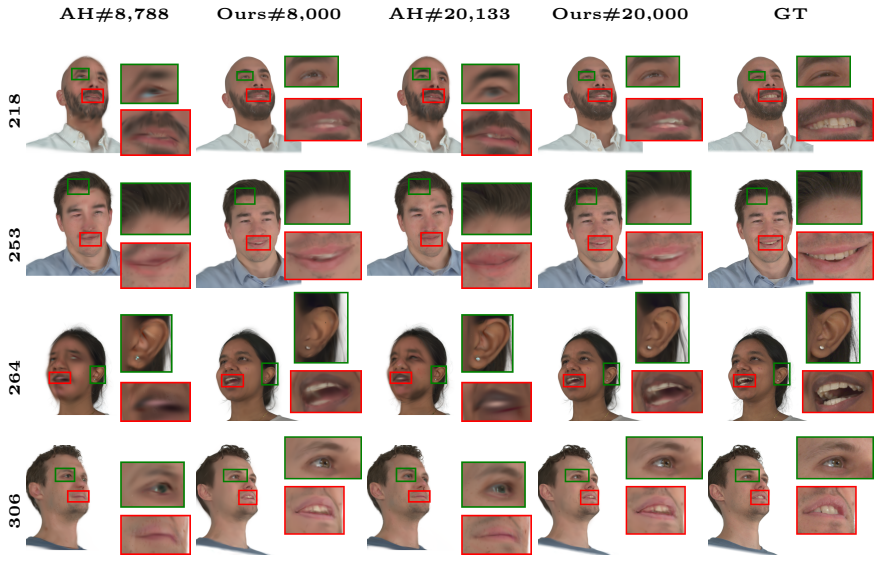

Figure 3: Qualitative comparison with ArchitectHead (AH) under comparable Gaussian budgets. RobustAvatar preserves sharper local details, especially around expression-sensitive regions such as eyes, lips, teeth, and ears, while using a similar number of rendered Gaussians.

Table 2: Ablation results on the validation and test datasets, reporting average PSNR, SSIM, and LPIPS over all avatar subjects.

Dataset	Ratio	10%				20%				50%			
	Strategy	a	b	c	Full	a	b	c	Full	a	b	c	Full
NVS	PSNR↑	27.51	28.02	28.09	28.42	28.40	27.92	28.83	29.47	29.15	29.01	29.68	29.72
	SSIM↑	0.87	0.88	0.87	0.923	0.90	0.90	0.91	0.93	0.92	0.92	0.93	0.93
	LPIPS↓	0.24	0.23	0.23	0.182	0.22	0.20	0.20	0.16	0.16	0.16	0.15	0.15
NES	PSNR↑	24.67	24.57	24.93	25.05	25.39	24.98	25.03	25.54	24.21	25.04	25.82	25.67
	SSIM↑	0.89	0.80	0.90	0.91	0.90	0.91	0.91	0.91	0.90	0.91	0.92	0.91
	LPIPS↓	0.22	0.22	0.21	0.18	0.20	0.19	0.19	0.16	0.18	0.19	0.15	0.16

Visual Quality Comparison Figure 3 compares RobustAvatar with ArchitectHead under comparable Gaussian budgets. Although ArchitectHead renders a similar number of Gaussians, it often produces over-smoothed facial regions and loses expression-sensitive details around the eyes, lips, and teeth. In contrast, RobustAvatar better preserves local structures such as eye contours, mouth interiors, and tooth boundaries under novel expressions. These visual results are consistent with the NVS and NES gains in Table 1, indicating that budget-conditioned selection and compensation more effectively preserve reconstruction- and animation-critical Gaussians. Figure 4 further shows RobustAvatar under different budget ratios.

4.3 Ablations and Analysis

For compactness, Table 2 denotes three ablated variants as a, b, and c. Variant a removes the canonical-space formulation, variant b disables the deformation branch of CBC, and variant c removes the semantic regularization in SDS. The full model achieves the best or comparable performance in most settings, with the clearest gains at 10% and 20% budgets, where Gaussian removal most strongly affects local facial structures. This indicates that robust LoD adaptation benefits from joint canonical-space compensation and semantic-aware selection.

At the 50% budget, the variant without semantic regularization can match or slightly outperform the full model on some PSNR, SSIM, and LPIPS entries. This does not contradict the purpose of semantic regularization, which is to preserve the facial-region distribution of retained Gaussians rather than directly maximize image-level metrics. Since PSNR, SSIM, and LPIPS do not explicitly

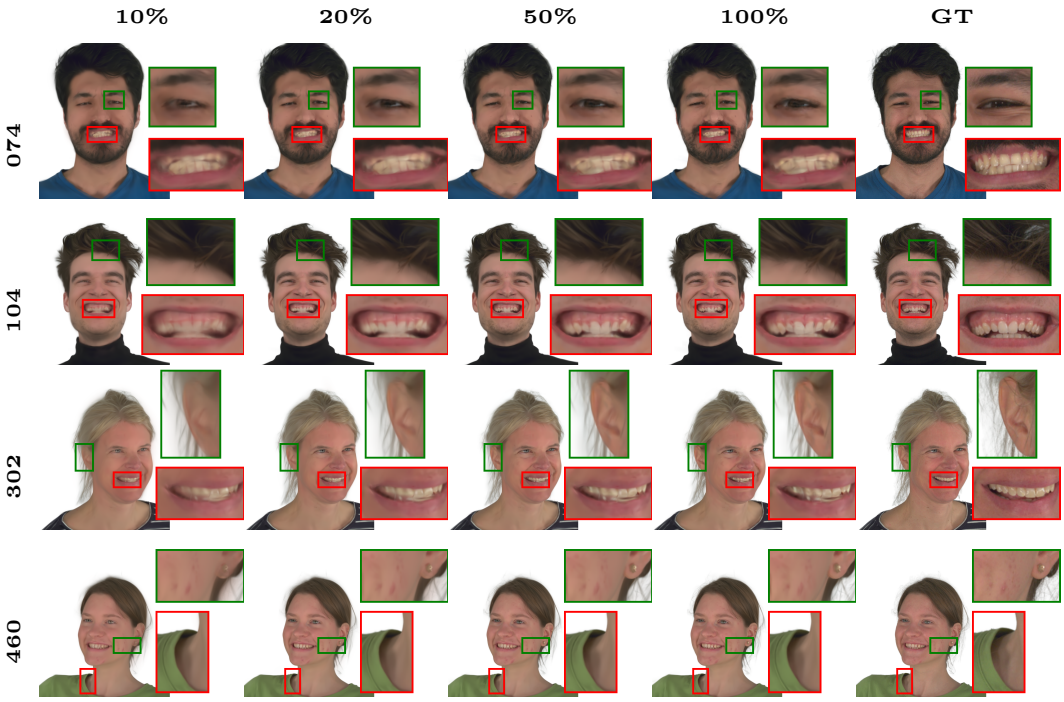

Figure 4: Qualitative results of RobustAvatar under different budget ratios. As the Gaussian budget increases from 10% to 100%, local facial details are progressively recovered and the rendering approaches the ground truth.

measure whether expression-sensitive regions are proportionally preserved, the semantic module should be viewed as a structural constraint for robust budgeted selection.

5 Conclusion

We presented RobustAvatar, a budget-robust LoD framework for dynamic Gaussian avatars. Starting from a single full avatar, RobustAvatar enables continuous Gaussian-budget control through budget-conditioned selection and compensation, without ratio-specific fine-tuning or separate LoD models. SDS selects budget-specific Gaussian subsets under semantic distribution constraints, while CBC compensates the retained primitives in canonical space to reduce geometric and appearance loss from Gaussian removal. Experiments show that RobustAvatar better preserves identity appearance, expression-sensitive details, and dynamic consistency, especially in low- and mid-budget regimes. These results demonstrate that selection followed by compensation is an effective paradigm for deploying one reconstructed avatar across diverse resource constraints.

Acknowledgments and Disclosure of Funding

This work was supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2023MF041).

Thanks to all coauthors for their efforts and to the area chair and reviewers for their constructive feedback.

I am also grateful for the mathematics I have always loved, and for programming as another way of writing equations.

Last but not least, special thanks to C for inspiring me to reflect and grow throughout this project. When GPU memory is limited, our method only renders a fraction of the Gaussians. When there is room again, those left out can find their way back into the scene.

Competing interests. The authors declare no competing interests.

References

Ziqian Bai, Feitong Tan, Zeng Huang, Kripasindhu Sarkar, Danhang Tang, Di Qiu, Abhimitra Meka, Ruofei Du, Mingsong Dou, Sergio Orts-Escolano, Rohit Pandey, Ping Tan, Thabo Beeler, Sean Fanello, and Yinda Zhang. Learning personalized high quality volumetric head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16890–16900, 2023.

Hyunsoo Cha, Inhee Lee, and Hanbyul Joo. PERSE: Personalized 3D generative avatars from a single portrait. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15953–15962, 2025.

Jinshu Chen, Bingchuan Li, Fan Zhang, Songtao Zhao, and Qian He. OneGT: One-shot geometry-texture neural rendering for head avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11294–11304, 2025.

Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. In *ACM SIGGRAPH Conference Proceedings*, pages 1–9, 2024. doi: 10.1145/3641519.3657499.

Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-1838. URL `https://openreview.net/forum?id=gVM2AZ5xA6`.

Xiaonuo Dongye, Hanzhi Guo, Le Luo, Haiyan Jiang, Yihua Bao, Jie Guo, Zeyu Tian, and Dongdong Weng. Lodavatar: hierarchical embedding and selective detail enhancement for adaptive levels of detail gaussian avatars. *Virtual Reality*, 29(4), November 2025. ISSN 1434-9957. doi: 10.1007/s10055-025-01249-3. URL `http://dx.doi.org/10.1007/s10055-025-01249-3`.

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

Wei-Qi Feng, Dong Han, Ze-Kang Zhou, Shunkai Li, Xiaoqiang Liu, Pengfei Wan, Di Zhang, and Miao Wang. GPAvatar: High-fidelity head avatars by learning efficient gaussian projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 250–259, 2025.

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

Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. NPGA: Neural parametric gaussian avatars. In *SIGGRAPH Asia Conference Papers*, 2024. URL `https://arxiv.org/abs/2405.19331`.

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

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-Splat: Fast 3D gaussian splatting with sparse pixels and sparse primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21537–21546, 2025a.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. PUP 3D-GS: Principled uncertainty pruning for 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5949–5958, 2025b.

Qiang Hu, Zihan Zheng, Houqiang Zhong, Sihua Fu, Li Song, Xiaoyun Zhang, Guangtao Zhai, and Yanfeng Wang. 4DGC: Rate-aware 4D gaussian compression for efficient streamable free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 875–885, 2025a.

Wentao Hu, Shunkai Li, Ziqiao Peng, Haoxian Zhang, Fan Shi, Xiaoqiang Liu, Pengfei Wan, Di Zhang, and Hui Tian. GGTalker: Talking head synthesis with generalizable gaussian priors and identity-specific adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10108–10117, 2025b.

Forrest Iandola, Stanislav Pidhorskyi, Igor Santesteban, Divam Gupta, Anuj Pahuja, Nemanja Bartolovic, Frank Yu, Emanuel Garbin, Tomas Simon, and Shunsuke Saito. SqueezeMe: Mobile-ready distillation of gaussian full-body avatars. In *ACM SIGGRAPH Conference Proceedings*, 2025. URL `https://arxiv.org/abs/2412.15171`.

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.

Byungjun Kim, Shunsuke Saito, Giljoo Nam, Tomas Simon, Jason Saragih, Hanbyul Joo, and Junxuan Li. HairCUP: Hair compositional universal prior for 3D gaussian avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL `https://arxiv.org/abs/2507.19481`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

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

Tobias Kirschstein, Javier Romero, Artem Sevastopolsky, Matthias Nießner, and Shunsuke Saito. Avat3r: Large animatable gaussian reconstruction model for high-fidelity 3D head avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12089–12100, 2025.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024.

Soonbin Lee, Fangwen Shu, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. Compression of 3D gaussian splatting with optimized feature planes and standard video codecs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 25496–25505, 2025.

Junxuan Li, Chen Cao, Gabriel Schwartz, Rawal Khirodkar, Christian Richardt, Tomas Simon, Yaser Sheikh, and Shunsuke Saito. URAvatar: Universal relightable gaussian codec avatars. In *SIGGRAPH Asia Conference Papers*, 2024. URL `https://arxiv.org/abs/2410.24223`.

Linzhou Li, Yumeng Li, Yanlin Weng, Youyi Zheng, and Kun Zhou. RGBAvatar: Reduced gaussian blendshapes for online modeling of head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10747–10757, 2025a.

Peng Li, Yisheng He, Yingdong Hu, Yuan Dong, Weihao Yuan, Yuan Liu, Zilong Dong, and Yike Guo. PanoLAM: Large avatar model for gaussian full-head synthesis from one-shot unposed image, 2025b.

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*, 36(6):194:1–194:17, 2017. doi: 10.1145/3130800.3130813.

Yuan Li, Ziqian Bai, Feitong Tan, Zhaopeng Cui, Sean Fanello, and Yinda Zhang. IM-portrait: Learning 3D-aware video diffusion for photorealistic talking heads from monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21107–21116, 2025c.

Di Liu, Teng Deng, Giljoo Nam, Yu Rong, Stanislav Pidhorskyi, Junxuan Li, Jason Saragih, Dimitris N. Metaxas, and Chen Cao. LUCAS: Layered universal codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21127–21137, 2025a.

Hengyu Liu, Yuehao Wang, Chenxin Li, Ruisi Cai, Kevin Wang, Wuyang Li, Pavlo Molchanov, Peihao Wang, and Zhangyang Wang. FlexGS: Train once, deploy everywhere with many-in-one flexible 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16336–16345, 2025b.

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. MaskGaussian: Adaptive 3D gaussian representation from probabilistic masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 681–690, 2025c.

Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3D gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH Conference Proceedings*, 2024. URL `https://arxiv.org/abs/2404.19398`.

SeungJun Moon, Hah Min Lew, Seungeun Lee, Ji-Su Kang, and Gyeong-Moon Park. GeoAvatar: Adaptive geometrical gaussian splatting for 3D head avatar. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL `https://arxiv.org/abs/2507.18155`.

Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1):1–17, 2024. doi: 10.1145/3651282.

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 (CVPR)*, pages 20299–20309, 2024.

Jianqiang Ren, Lin Liu, and Steven Hoi. OMG-Avatar: One-shot multi-LOD gaussian head avatar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11017–11028, 2026. URL `https://arxiv.org/abs/2603.01506`.

Jianxiong Shen, Yue Qian, and Xiaohang Zhan. LOD-GS: Achieving levels of detail using scalable gaussian soup. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 671–680, 2025.

Chao Shi, Shenghao Jia, Jinhui Liu, Yong Zhang, Liangchao Zhu, Zhonglei Yang, Jinze Ma, Chaoyue Niu, and Chengfei Lv. HRM^{2}avatar: High-fidelity real-time mobile avatars from monocular phone scans. In *SIGGRAPH Asia Conference Papers*, 2025. doi: 10.1145/3757377.3763894. URL `https://arxiv.org/abs/2510.13587`.

Kaiwen Song, Jinkai Cui, and Juyong Zhang. Progressiveavatars: Progressive animatable 3D gaussian avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. doi: 10.48550/arXiv.2603.16447. URL `https://arxiv.org/abs/2603.16447`.

Heyi Sun, Cong Wang, Tian-Xing Xu, Jingwei Huang, Di Kang, Chunchao Guo, and Song-Hai Zhang. SVG-head: Hybrid surface-volumetric gaussians for high-fidelity head reconstruction and real-time editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2508.09597.

Shuai Tan, Bill Gong, Bin Ji, and Ye Pan. FixTalk: Taming identity leakage for high-quality talking head generation in extreme cases. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24–36, 2025.

Haocheng Tang, Ruoke Yan, Xinhui Yin, Qi Zhang, Xinfeng Zhang, Siwei Ma, Wen Gao, and Chuanmin Jia. HGC-avatar: Hierarchical gaussian compression for streamable dynamic 3D avatars. In *Proceedings of the ACM International Conference on Multimedia*, 2025. doi: 10.1145/3746027.3755317. URL https://arxiv.org/abs/2510.16463.

Felix Taubner, Ruihang Zhang, Mathieu Tuli, and David B. Lindell. CAP4D: Creating animatable 4D portrait avatars with morphable multi-view diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5318–5330, 2025.

Kartik Teotia, Helge Rhodin, Mohit Mendiratta, Hyeongwoo Kim, Marc Habermann, and Christian Theobalt. Audio-driven universal gaussian head avatars. In *SIGGRAPH Asia Conference Papers*, 2025. URL https://arxiv.org/abs/2509.18924.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26274–26284, 2025a.

Yating Wang, Xuan Wang, Ran Yi, Yanbo Fan, Jichen Hu, Jingcheng Zhu, and Lizhuang Ma. 3D gaussian head avatars with expressive dynamic appearances by compact tensorial representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21117–21126, 2025b.

Sicheng Xu, Guojun Chen, Jiaolong Yang, Yizhong Zhang, Yu Deng, Stephen Lin, and Baining Guo. VASA-3D: Lifelike audio-driven gaussian head avatars from a single image. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iLbXyPNTeb.

Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1941, 2024.

Peizhi Yan, Rabab Ward, Qiang Tang, and Shan Du. Architecthead: Continuous level of detail control for 3D gaussian head avatars, 2025.

Zhengming Yu, Tianye Li, Jingxiang Sun, Omer Shapira, Seonwook Park, Michael Stengel, Matthew Chan, Xin Li, Wenping Wang, Koki Nagano, and Shalini De Mello. GAIA: Generative animatable interactive avatars with expression-conditioned gaussians. In *ACM SIGGRAPH*, 2025. URL https://research.nvidia.com/labs/amri/projects/gaia/.

Dongbin Zhang, Yunfei Liu, Lijian Lin, Ye Zhu, Kangjie Chen, Minghan Qin, Yu Li, and Haoqian Wang. HRAvatar: High-quality and relightable gaussian head avatar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26285–26296, 2025a.

Jiawei Zhang, Zijian Wu, Zhiyang Liang, Yicheng Gong, Dongfang Hu, Yao Yao, Xun Cao, and Hao Zhu. FATE: Full-head gaussian avatar with textural editing from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5535–5545, 2025b.

Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. LP-3DGS: Learning to prune 3D gaussian splatting. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=kzJ9P7VPnS.

Dingcheng Zhen, Shunshun Yin, Shiyang Qin, Hou Yi, Ziwei Zhang, Siyuan Liu, Gan Qi, and Ming Tao. Teller: Real-time streaming audio-driven portrait animation with autoregressive motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21075–21085, 2025.

Zhenglin Zhou, Fan Ma, Hehe Fan, and Tat-Seng Chua. Zero-1-to-a: Zero-shot one image to animatable head avatars using video diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15941–15952, 2025.

Wojciech Zielonka, Timo Bolkart, Thabo Beeler, and Justus Thies. Gaussian eigen models for human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a. URL `https://arxiv.org/abs/2407.04545`.

Wojciech Zielonka, Stephan J. Garbin, Alexandros Lattas, George Kopanas, Paulo Gotardo, Thabo Beeler, Justus Thies, and Timo Bolkart. Synthetic prior for few-shot drivable head avatar inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. URL `https://arxiv.org/abs/2501.06903`.

One Avatar, Any Budget: Robust LoD for Dynamic Gaussian Avatars

Supplementary Material

A Overview

This supplement presents additional method comparisons, limitations, implementation details, and experiments on deployment efficiency, pruning, and compensation. The accompanying videos, `demo-render-only.mp4` and `demo-render-gaussian.mp4`, show multi-scene novel-view rendering under a continuous sweep of Gaussian retention ratios.

B Relationship to Existing Methods and OMG-Avatar Comparison

B.1 Flexible Gaussian representations and dynamic-avatar adaptation

FlexGS Liu et al. [2025b] studies flexible capacity for static Gaussian scenes through learned selection and attribute transformation. RobustAvatar follows the select-then-compensate formulation but addresses a reconstructed, animatable identity. In this setting, the retained representation must preserve facial semantics and remain compatible with deformation driven by facial expressions and head poses. A primitive with limited contribution in a neutral frontal frame can become important under another expression or viewpoint; a selection criterion tied to a single animation state can therefore discard useful support. Globally important primitives can concentrate in large regions, leaving small but perceptually sensitive regions such as the eyes and lips insufficiently represented under a small budget.

SDS addresses budget-dependent allocation through a shared ratio-conditioned scorer and facial semantic distribution regularization. CBC predicts residuals in canonical FLAME-local space before applying the original face-local transform to the posed avatar.

Progressive Gaussian methods, including ProgressiveAvatars Song et al. [2026], organize content for incremental transmission and reuse. A fixed importance hierarchy typically defines its low-budget models through prefixes. RobustAvatar selects Gaussian subsets, which need not be nested across budgets, and compensates the attributes of the retained primitives for each requested budget. Our objective is to preserve fidelity at each budget rather than support incremental transmission.

B.2 Cross-identity and cross-budget generalization

OMG-Avatar Ren et al. [2026] is a one-shot, cross-identity head-avatar model with three predefined LoD levels. RobustAvatar starts from an already reconstructed GaussianAvatar and optimizes a model for that identity across budgets. The two approaches generalize along different axes: unseen identities from a single input image versus unseen budgets for the same optimized identity.

Table 3: OMG-Avatar reproduction at its three predefined LoD levels. Input and other denote the input viewpoint and the remaining evaluated viewpoints, respectively.

	Input viewpoint			Other viewpoints		
LoD	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
2	23.85	0.85	0.11	21.28	0.82	0.18
1	23.84	0.85	0.11	21.24	0.82	0.18
0	23.80	0.85	0.12	21.25	0.82	0.18

C Limitations

Similar to existing Gaussian avatar methods, our approach mainly targets controlled multi-view capture settings and relies on accurate calibration, segmentation, and face tracking. Its quality may

degrade under in-the-wild videos, severe occlusions, extreme illumination, or large appearance changes caused by hair and accessories. Since we use FLAME as the underlying facial prior, details that are difficult to represent with the FLAME topology, such as complex hairstyles, glasses, teeth, or non-rigid accessories, may also remain challenging.

Although our ratio-conditioned model supports multiple LoD levels, very low retention ratios inevitably lose some high-frequency appearance and subtle expression details, reflecting the trade-off between compression and rendering quality. Moreover, the compression ratio mainly controls the number of retained Gaussians and does not directly correspond to hardware latency or frame rate, which may vary across rasterizers, GPU architectures, and batching strategies.

Per-avatar training and hardware scope. RobustAvatar optimizes an identity-specific asset and does not provide one-shot cross-identity generalization. Per-subject training is a scalability cost when creating many avatars; sharing knowledge across identities is a future direction. Our efficiency measurements are limited to an RTX 4090. Evaluation on mobile devices remains future work.

Temporal evaluation and uncertainty. Temporal metrics are evaluated at fixed budgets. We do not quantitatively evaluate artifacts during budget switching or report repeated-run variability.

Broader impacts. Budget-adaptive avatars may broaden access to telepresence and social VR, but more deployable and visually convincing avatars can also facilitate impersonation, deceptive media, and unauthorized likeness use. Responsible deployment should respect consent, privacy, and the licenses of datasets, models, and software. Demographic coverage and differential quality are not evaluated.

D Reproducibility Statement

This section describes the data protocol, training settings, network architectures, and loss definitions used in our experiments.

Data and split protocol. All variants use the train/validation/test splits provided by the GaussianAvatars data pipeline. The data follow the standard GaussianAvatars-style preprocessing format, including calibrated cameras, per-frame FLAME parameters, mesh bindings, masks, and RGB images.

Training protocol. All reported RobustAvatar models are initialized from the corresponding full-resolution GaussianAvatars teacher model and trained for $60{,}000$ iterations. The teacher checkpoint is taken at iteration $60{,}000$. Unless otherwise stated, the same optimizer settings, learning-rate schedule, loss weights, rendering settings, and data split are used for all scenes and ablations. The loss weights are summarized in Table 5. All experiments use a white background and evaluate the same validation and test views.

Training is performed on one NVIDIA RTX 4090 using Adam. Position, SH, opacity, scale, and rotation learning rates are 5×10^{-4}, 2.5×10^{-3}, 5×10^{-2}, 10^{-2}, and 10^{-3}, respectively; selector and compensation-field learning rates are 10^{-4}. Training ratios are $\{0.1, 0.3, 0.6, 0.9\}$. The GaussianAvatars-preprocessed NeRSemble evaluation uses calibrated cameras, FLAME tracking, foreground masks, and white backgrounds. NVS/NES quality is measured with PSNR, SSIM, and LPIPS; temporal metrics use valid foreground regions after forward–backward flow consistency filtering.

D.1 SDS architecture and budget-conditioned selection

The selector receives canonical local position $\mathbf{x}_i \in \mathbb{R}^3$, normalized quaternion $\mathbf{q}_i \in \mathbb{R}^4$, activated scale $\exp(\mathbf{s}_i) \in \mathbb{R}^3$, and target retention ratio $r \in \mathbb{R}$. Four two-layer encoders with ReLU activations map these inputs to four 32-dimensional features. Each encoder has dimensions $d_{\text{in}} \to 32 \to 32$. Concatenation gives a 128-dimensional feature, which is mapped to two keep/drop logits. Training uses straight-through hard Gumbel-Softmax with temperature $\tau = 1$. At inference, the soft keep probabilities are ranked globally and the method retains exactly $K = \max(1, \text{round}(rN))$ primitives.

The scorer is optimized using contribution guidance, the ratio constraint, semantic distribution regularization, and reconstruction supervision. The ratio penalty encourages the selected proportion to match the target retention ratio during training.

Conditioning the scorer on the retention ratio allows Gaussian rankings to vary across budgets.

D.2 CBC feature planes and residual prediction

The canonical-ratio field uses all six coordinate pairs $\{xy, xz, yz, xr, yr, zr\}$. Each plane has 32 channels. The base four-dimensional resolution is $(64, 64, 64, 25)$, with spatial scales $\{1, 2, 4, 8\}$ and ratio resolution fixed at 25. Consequently, a spatial plane at scale s has resolution $64s \times 64s$, while a plane involving r has one spatial dimension of length $64s$ and a ratio dimension of length 25.

Features are bilinearly sampled from the planes, multiplied within each scale, and concatenated across the four scales, producing a 128-dimensional vector. A two-layer MLP with hidden width 64 predicts five residual heads: position (3), scale (3), rotation (4), opacity (1), and SH coefficients (48), for 59 output components in total. The fourth field coordinate is the target retention ratio r. Residuals are predicted in canonical space and composed with the original FLAME animation transform.

Table 4: Network dimensions used by SDS and CBC. Ratio-plane resolution remains fixed across spatial scales.

Component	Specification
SDS position encoder	$3 \to 32 \to 32$
SDS quaternion encoder	$4 \to 32 \to 32$
SDS physical-scale encoder	$3 \to 32 \to 32$
SDS ratio encoder	$1 \to 32 \to 32$
SDS fusion and output	$4 \times 32 = 128$ features; 2 logits
CBC coordinate planes	xy, xz, yz, xr, yr, zr; 32 channels each
CBC base resolution	$(64, 64, 64, 25)$
CBC spatial scales	$1, 2, 4, 8$; ratio resolution 25
CBC fused feature	$4 \times 32 = 128$ dimensions
CBC prediction	Hidden width 64; heads $3, 3, 4, 1, 48$

D.3 Image losses and field regularization

Let I_{teacher} denote the teacher-rendered reference I^{full} in Section 3.3. The image I^r is rendered from the selected Gaussians after CBC compensation at retention ratio r. The reconstruction loss uses $\lambda_{\text{ssim}} = 0.2$:

$$\mathcal{L}_{\text{rec}} = 0.8\|I^r - I_{\text{teacher}}\|_1 + 0.2\big(1 - \text{SSIM}(I^r, I_{\text{teacher}})\big). \tag{13}$$

Let $I^{\text{full}}_{\text{student}}$ denote the rendering of all current student Gaussians before selection and CBC compensation. The image component of the preservation objective compares this rendering with the teacher reference:

$$\begin{aligned}\mathcal{L}_{\text{pres,img}} = 0.25\|I^{\text{full}}_{\text{student}} - I_{\text{teacher}}\|_1 \\ + 0.10\big(1 - \text{SSIM}(I^{\text{full}}_{\text{student}}, I_{\text{teacher}})\big).\end{aligned} \tag{14}$$

Here, $\|\cdot\|_1$ denotes the mean absolute image error.

Following FlexGS Liu et al. [2025b], the feature field is regularized using squared second differences and a deviation-from-one penalty. Let $P \in \mathbb{R}^{B\times C\times H\times W}$ denote a stored feature plane, and define

$$\mathcal{D}_2(P) = \frac{1}{BC(H-2)W} \sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{h=1}^{H-2}\sum_{w=1}^{W} \left(P_{b,c,h+2,w} - 2P_{b,c,h+1,w} + P_{b,c,h,w}\right)^2. \tag{15}$$

Let $\mathcal{P}_s = \{xy, xz, yz\}$ and $\mathcal{P}_r = \{xr, yr, zr\}$. Across resolution levels ℓ, the field regularizer is defined as

$$\mathcal{L}_{\text{reg}}^{\text{grid}} = \alpha_s \sum_{\ell} \sum_{ab \in \mathcal{P}_s} \mathcal{D}_2(P_{ab}^{\ell}) + \alpha_r \sum_{\ell} \sum_{ab \in \mathcal{P}_r} \mathcal{D}_2(P_{ab}^{\ell}) + \alpha_1 \sum_{\ell} \sum_{ab \in \mathcal{P}_r} \text{mean} \left|1 - P_{ab}^{\ell}\right|. \tag{16}$$

The deviation penalty encourages ratio-plane features to remain near 1, the multiplicative identity.

Table 5 lists the image-loss and field-regularization coefficients. The field coefficients follow FlexGS. The remaining weights are $\lambda_{\text{imp}} = 0.5$, $\lambda_{\text{ratio}} = 0.01$, $\lambda_{\text{sem}} = 0.2$, and $\lambda_{\text{pres}} = 1.0$.

Table 5: Image-loss and field-regularization weights.

Weight	Value
Spatial-plane smoothness α_s	2×10^{-4}
Ratio-plane smoothness α_r	10^{-3}
Ratio-plane deviation penalty α_1	10^{-4}
Reconstruction SSIM weight λ_{ssim}	0.2
Preservation image L1 coefficient	0.25
Preservation image SSIM coefficient	0.10

E Export and Rasterization Efficiency

E.1 Three-stage deployment and measurement protocol

Deployment separates three operations: offline training of an identity-specific LoD network; one-time generation of a fixed avatar model for a requested retention ratio; and repeated rasterization of that exported model. The same trained network generates different budgets without ratio-specific retraining. The export stage includes Gaussian scoring, exact top-K selection, attribute compensation, and construction of the fixed-budget avatar. An export takes 2.995–4.406 ms on an NVIDIA RTX 4090 across the evaluated ratios.

Each exported avatar is independently reloaded for rasterization. We use FP32, a white background, resolution 550×802, and all 480 test views. CUDA events bracket exactly one GaussianRasterizer call, and each reported latency result contains 1,440 measurements. FPS is computed from the unrounded mean rasterization latency. Table 6 reports mean and 95th-percentile latency.

GPU allocation is measured in a fresh CUDA process for each model, without loading SDS or CBC. We report PyTorch allocated memory after loading the model and the maximum allocated memory observed during rasterization over the 480 views. These are allocated-memory statistics, not whole-process GPU usage or reserved allocator memory. Storage includes both the ratio-dependent Gaussian PLY and shared FLAME metadata. The latter is unchanged across ratios, so total storage does not decrease in direct proportion to Gaussian count.

Table 6: Fixed-model deployment measurements on RTX 4090. Latency and FPS refer to one rasterizer call, not complete avatar-system latency. Storage includes PLY and shared FLAME metadata. Loaded and peak denote PyTorch allocated GPU memory, excluding SDS/CBC.

Ratio	K	Storage MiB	Mean ms	P95 ms	FPS	Loaded MiB	Peak MiB
5%	2,307	6.73	0.312	0.330	3,207	44.51	64.64
10%	4,615	7.28	0.343	0.361	2,919	45.04	66.99
20%	9,230	8.39	0.384	0.409	2,601	46.09	71.69
30%	13,845	9.50	0.412	0.444	2,427	47.15	76.39
50%	23,076	11.72	0.489	0.551	2,044	49.26	85.80
100%	46,152	17.26	0.685	0.789	1,458	54.55	109.31

At 5% retention relative to 100%, mean rasterization latency decreases by approximately 54.5%, throughput increases by approximately $2.20\times$, and storage decreases by approximately 61.0%. Loaded and peak allocated memory decrease by approximately 18.4% and 40.9%, respectively.

F Controlled Pruning and Fine-Tuning Baselines

We compare random pruning, opacity top-K, and rendering-contribution top-K to separate the benefits of learned selection and compensation from pruning followed by fine-tuning. All methods start from the same full GaussianAvatar and retain the same number of primitives at each target ratio. The pruning baselines are fine-tuned under matched optimization settings. RobustAvatar uses a single model trained across multiple budgets, without ratio-specific fine-tuning. The contribution score aggregates visibility, opacity, and screen-space footprint over training views, corresponding to the full-avatar guidance in Section 3.1. Unlike SDS, the fixed-ranking alternatives do not use a ratio-conditioned scorer or semantic-distribution regularization.

Table 7: Controlled pruning and fine-tuning comparisons. All methods use the same input avatar and retain the same number of Gaussians at each target ratio.

Task	Ratio	Metric	Random	Opacity	Contribution	Ours
NVS	10%	PSNR	14.49	18.65	22.98	28.42
		SSIM	0.73	0.79	0.86	0.923
		LPIPS	0.27	0.24	0.18	0.182
NES	10%	PSNR	15.04	18.72	21.07	25.05
		SSIM	0.73	0.80	0.86	0.91
		LPIPS	0.25	0.23	0.20	0.18
NVS	20%	PSNR	19.09	22.44	25.95	29.47
		SSIM	0.78	0.84	0.91	0.93
		LPIPS	0.23	0.22	0.18	0.16
NES	20%	PSNR	19.24	20.62	23.23	25.54
		SSIM	0.79	0.83	0.89	0.91
		LPIPS	0.22	0.21	0.18	0.16
NVS	50%	PSNR	24.30	25.05	26.46	29.72
		SSIM	0.91	0.90	0.92	0.93
		LPIPS	0.16	0.12	0.17	0.15
NES	50%	PSNR	24.13	23.18	24.78	25.67
		SSIM	0.86	0.89	0.90	0.91
		LPIPS	0.19	0.20	0.16	0.16

Relative to contribution top-K, RobustAvatar improves NVS PSNR by 5.44, 3.52, and 3.26 dB at 10%, 20%, and 50%, and NES PSNR by 3.98, 2.31, and 0.89 dB. The largest gains occur at constrained budgets. Opacity top-K achieves lower NVS LPIPS at the 50% budget, with 0.12 compared with 0.15 for RobustAvatar.

G Complete CBC Ablation and Full-Budget Controls

G.1 Disabling all CBC compensation

The variant without CBC retains the canonical avatar and FLAME animation pipeline and sets all budget-conditioned attribute residuals to zero:

$$(\Delta\mathbf{x}_i, \Delta\mathbf{s}_i, \Delta\mathbf{q}_i, \Delta o_i, \Delta\mathbf{h}_i) = \mathbf{0}. \tag{17}$$

The variant *w/o CBC compensation* corresponds to variant (b) in Table 2. It uses the same SDS module, budget sampling, varying subsets, training iterations, and optimization of the base Gaussian attributes as the full method. This comparison isolates the contribution of the CBC residuals.

Compared with this variant, the full model improves NVS PSNR by 0.40, 1.55, and 0.71 dB and NES PSNR by 0.48, 0.56, and 0.63 dB at the 10%, 20%, and 50% budgets.

G.2 Retaining all Gaussians and disabling SDS/CBC jointly

The GaussianAvatars (100%) baseline in Table 1 is the input avatar before applying our adaptation framework. We separately evaluate RobustAvatar while retaining all Gaussians and a control that uses the same training data, optimization settings, training procedure, and reconstruction supervision while disabling both SDS and CBC.

Table 8: Full-budget reconstruction quality of the input GaussianAvatar, RobustAvatar, and the control trained without SDS/CBC.

Setting	Task	PSNR	SSIM	LPIPS
Input GaussianAvatar	NVS	29.19	0.93	0.15
Input GaussianAvatar	NES	25.55	0.91	0.15
RobustAvatar at 100%	NVS	29.60	0.94	0.14
RobustAvatar at 100%	NES	25.13	0.93	0.15
Same training, without SDS/CBC	NVS	29.20	0.92	0.15
Same training, without SDS/CBC	NES	25.53	0.91	0.15

At full budget, the control without SDS and CBC remains close to the input avatar under the same training procedure. After adaptation, NVS PSNR increases from 29.19 to 29.60 dB, whereas NES PSNR decreases from 25.55 to 25.13 dB. SSIM increases for both tasks, while LPIPS decreases for NVS and remains unchanged for NES at the reported precision.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's scope and main contributions: RobustAvatar, SDS, and CBC for budget-conditioned LoD adaptation of reconstructed dynamic Gaussian avatars. These claims are supported by the method sections and the NeRSemble evaluation in Section 4, and are scoped to avatar-specific Gaussian-budget adaptation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a limitations section in the supplementary material (Appendix C).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 describes the main experimental settings and hyperparameters, while the supplementary material provides additional implementation and evaluation details. See Appendix D for the additional details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The authors confirm open access to the code, data access instructions, and instructions for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We follow the official GaussianAvatars preprocessing protocol and its NeRSemble train/validation/test splits for all methods, including calibrated cameras, foreground masks, FLAME tracking, and evaluation views.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports quantitative metrics under the specified evaluation protocol, but does not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 reports the GPU type and training iterations used for each subject. Appendix E additionally reports export and rasterization measurements.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper is in all respects consistent with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix C discusses telepresence accessibility alongside impersonation, deceptive media, likeness misuse, privacy, and unmeasured demographic coverage.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The authors confirm that appropriate safeguards are in place for any high-risk assets released with this work. Appendix C discusses consent, privacy, responsible release, and misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the original papers for the existing datasets, models, protocols, and baseline methods used in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The authors confirm that the new research assets are accompanied by documentation describing usage, dependencies, and licensing.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or new human-subject studies. Experiments use existing data protocols and automatic metrics.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not conduct new human-subject data collection, crowdsourcing, or user studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology, training, evaluation, or experiments.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.