

RaRa Clipper: A Clipper for Gaussian Splatting Based on Ray Tracer and Rasterizer

Da Li, Donggang Jia, Yousef Rajeh, Dominik Engel, and Ivan Viola
 {da.li, donggang.jia, yousef.rajeh, dominik.engel, ivan.viola}@kaust.edu.sa

King Abdullah University of Science and Technology

Abstract

With the advancement of Gaussian Splatting techniques, a growing number of datasets based on this representation have been developed. However, performing accurate and efficient clipping for Gaussian Splatting remains a challenging and unresolved problem, primarily due to the volumetric nature of Gaussian primitives, which makes hard clipping incapable of precisely localizing their pixel-level contributions. In this paper, we propose a hybrid rendering framework that combines rasterization and ray tracing to achieve efficient and high-fidelity clipping of Gaussian Splatting data. At the core of our method is the RaRa strategy, which first leverages rasterization to quickly identify Gaussians intersected by the clipping plane, followed by ray tracing to compute attenuation weights based on their partial occlusion. These weights are then used to accurately estimate each Gaussian’s contribution to the final image, enabling smooth and continuous clipping effects. We validate our approach on diverse datasets, including general Gaussians, hair strand Gaussians, and multi-layer Gaussians, and conduct user studies to evaluate both perceptual quality and quantitative performance. Experimental results demonstrate that our method delivers visually superior results while maintaining real-time rendering performance and preserving high fidelity in the unclipped regions.

Keywords: Clipping, Gaussian Splatting, Rasterization, Ray tracing.

1 Introduction

With the rapid advancement of 3D vision and neural rendering, Gaussian Splatting (GS) Kerbl et al., 2023 has emerged as a powerful representation for real-time rendering and differentiable reconstruction. Unlike traditional discrete representations such as point clouds, meshes, or voxel grids, Gaussian primitives offer continuous spatial support, anisotropy, and sub-pixel anti-aliasing, enabling both high fidelity and efficient rendering. As a result, this representation has been widely adopted in state-of-the-art research and open-source systems.

In many practical applications, such as interactive editing Tang et al., 2025, medical image clipping Kleinbeck et al., 2025, robotic navigation Chen et al., 2024, and hair modification Zakharov et al., 2024; Zhou et al., 2024, it is often necessary to perform clipping operations to isolate regions of interest, remove occlusion, or accelerate rendering. While traditional graphics pipelines provide mature solutions such as z-buffering, frustum culling, and AABB-based rejection, these techniques Shirley et al., 2009 are not directly applicable to Gaussian Splatting.

The primary challenge lies in the volumetric nature of Gaussian primitives: their contribution to image formation is computed by integrating their density along viewing directions, rather than through surface-based geometry. Existing clipping strategies Kerbl et al., 2023; Kleinbeck et al., 2025 employ a binary filtering approach based solely on the position of the Gaussian center. In this paper, we refer to this as the **Hard Clipping** strategy. However, this method fails to account for the true image-space contribution of each Gaussian, often leading to visual artifacts such as aliasing, popping, and the unintended removal of partially visible primitives, shown in Figure 1, Figure 2, and Figure 9 - 12.

To address these issues, we propose a novel **Ray-Rasterizer (RaRa)** clipping framework that combines the parallel visibility analysis capabilities of rasterization with the precise intersection reasoning of ray tracing. Our method consists of two key stages: (1) we classify the visibility of each Gaussian by computing its signed distance to the clipping plane, labeling it as fully visible, fully invisible, or near the clipping boundary (cutoff); and (2) we apply ray tracing only to Gaussians in the cutoff region to compute ellipsoid-ray intersections and derive a decaying opacity weight based on the length of the visible ray segment.

This selective ray tracing avoids unnecessary computation, reduces errors caused by hard clipping, and maintains structural continuity across clipped regions. Moreover, the proposed strategy is general and can be integrated into any system that supports

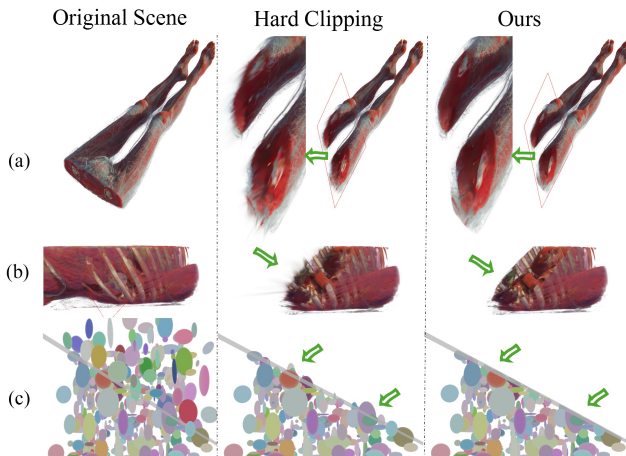


Figure 1. Comparison of different clipping strategies across representative Gaussian Splatting scenes. (a) Lower body dataset(a) and full body dataset(b): our method preserves anatomical structure without boundary artifacts, while hard clipping introduces visible distortions. (c) Schematic visualization using ellipsoids highlights the effect of clipping at the primitive level—RaRa selectively attenuates partially visible Gaussians, avoiding abrupt removal.

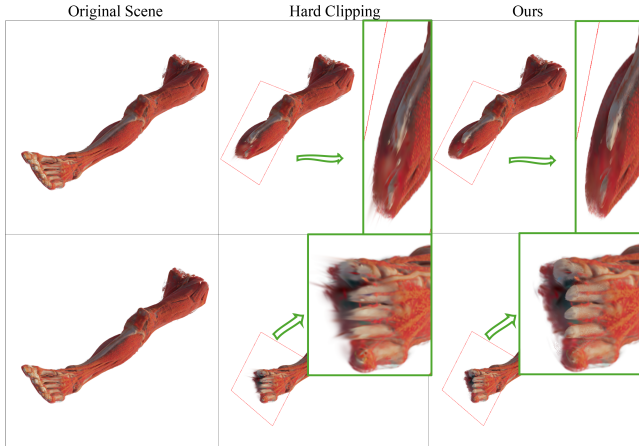


Figure 2. Comparison of clipping results on the full leg dataset using different clipping strategies and clipping planes. The dataset is clipped by two different planes. Our method preserves anatomical detail and structural continuity near the clipping boundary, while hard clipping introduces visible distortions and blurring artifacts.

Gaussian Splatting.

We evaluate our approach on three representative datasets—general Gaussians scenes, strand Gaussians scenes [Zakharov et al., 2024](#); [Zhou et al., 2024](#); [Luo et al., 2024](#), and multi-layer Gaussians scenes [Tang et al., 2025](#); [Kleinbeck et al., 2025](#)—and conduct both quantitative and perceptual evaluations via user studies. Results demonstrate that our method consistently preserves visual fidelity, shown in [Figure 1](#) and real-time performance.

Our main contributions can be summarized as follows:

- We propose the RaRa hybrid clipping framework—the first to integrate rasterization and ray tracing for efficient and precise Gaussian Splatting clipping, which can be seamlessly integrated as an add-on to a general Gaussian Splatting renderer.
- We design a physically-consistent decaying opacity function based on visible ray segment length, enabling accurate modeling of partial occlusion.
- We conduct comprehensive experiments and user studies on multiple Gaussian datasets, demonstrating superior visual quality and robustness compared to hard clipping.

2 Related Research

2.1 Gaussian Splatting and Its Applications

Gaussian Splatting has recently gained significant attention as a continuous 3D representation technique for real-time rendering and differentiable scene reconstruction. The seminal work by Kerbl et al. [Kerbl et al., 2023](#) introduced 3D Gaussian Splatting, which uses anisotropic Gaussian primitives as rendering units and employs efficient image-space rasterization without requiring explicit meshes similar to previous work [Westover, 1991](#); [Zwicker et al., 2001](#). Compared to point-based [Kivi et al., 2022](#) or neural implicit methods [Mildenhall et al., 2021](#); [Barron et al., 2021](#), GS provides higher reconstruction fidelity while maintaining interactive performance.

Subsequent research has explored various applications and extensions of GS, including sparse-view reconstruction [Liu et al.,](#)

[2025](#), dynamic scene reconstruction [Duan et al., 2024](#); [Lin et al., 2024](#); [Yang et al., 2024](#), avatar reconstruction [Qian et al., 2024](#); [Z. Li et al., 2024](#); [Pang et al., 2024](#); [Hu et al., 2024](#), and multi-scale modeling [J. Li et al., 2024](#); [Seo et al., 2024](#); [Ren et al., 2024](#). Highly anisotropic Gaussians have been applied to fine-grained strand-level hair modeling [Zakharov et al., 2024](#); [Zhou et al., 2024](#); [Luo et al., 2024](#), and anatomical data has been converted to layered Gaussian fields [Kleinbeck et al., 2025](#) for efficient rendering on consumer VR devices [Niedermayr et al., 2024](#). Works like iVR [Tang et al., 2025](#) further introduced editable Gaussian attributes and multi-layer fusion, enhancing interaction and structural representation. These studies demonstrate the versatility of GS beyond appearance modeling, indicating a growing need for standardized operations such as clipping, which remains underexplored and motivates this paper.

2.2 Limitations of Traditional Clipping in Gaussian Representations

Clipping and culling are standard operations in traditional graphics pipelines, primarily using techniques such as view frustum culling and Z-buffer depth testing [Shirley et al., 2009](#); [Tran, 1986](#). These methods perform well on rigid, surface-based geometry like meshes or point clouds. However, Gaussian primitives are volumetric in nature and provide soft spatial support defined by covariance matrices, making them incompatible with conventional binary visibility tests.

Some systems attempt hard clipping by measuring the distance from the Gaussian center to the clipping plane, but such heuristics often produce artifacts—flickering, popping, or structural loss—especially near clipping boundaries. For example, the original 3DGS [Kerbl et al., 2023](#) includes basic frustum-based visibility culling based on Gaussian centers, and Multi-layer GS [Kleinbeck et al., 2025](#) implements bounding box clipping but still relies on center-based hard culling logic. Other domain-specific works include ellipsoidal volume clipping on the digital earth platform using voxel-aligned slicing [Wang et al., 2023](#), fast proxy-based clipping of subterranean networks [Konev et al., 2018](#), and image-space clipping in textured volume rendering [Weiskopf et al., 2003](#). These methods, however, operate on voxel grids, regular proxy geometry, or raster-aligned volumes, and cannot be directly applied to the soft-support, anisotropic primitives used in Gaussian Splatting.

Thus, implementing an efficient and accurate clipping framework for Gaussian fields remains an open problem. The core difficulty lies in accounting not only for binary visibility but also for the integral contribution of partially visible Gaussians to each pixel.

2.3 Ray Tracing for Gaussian Rendering

To enhance photorealism and physical modeling in GS-based rendering, recent works have introduced ray tracing into Gaussian rendering pipelines. For instance, 3DGRT [Moenne-Loccoz et al., 2024](#) utilizes bounding volumes for efficient ray-Gaussian intersection and supports reflection/refraction modeling. 3DGUT [Wu et al., 2024](#) retains rasterization but integrates ray tracing for performance optimization. REdiSplats [Byrski et al., 2025](#) approximates Gaussians as mesh-based flat shapes, enabling standard mesh-editing operations. Relightable 3D Gaussians [Gao et al., 2024](#)

combine ray tracing with BRDF decomposition for realistic re-lighting. Stochastic Ray Tracing Sun et al., 2025 adopts efficient sampling strategies to accelerate transparency modeling in Gaussian scenes.

Although these techniques significantly improve global rendering quality and light transport modeling, they rarely address localized clipping or fine-grained visibility estimation. Many require custom ray tracing pipelines or mesh priors, which increase training costs or reduce rendering performance Mai et al., 2024. In contrast, our RaRa strategy focuses on real-time, localized clipping by selectively applying ray tracing only to cutoff Gaussians. Rasterization is first used to identify potentially clipped Gaussians, followed by ray-ellipsoid intersection and visible volume ratio computation. Our approach is plugin-style and can be directly integrated into native GS pipelines without altering training or data structures, enabling precise and interactive clipping at negligible runtime overhead.

3 Method

In the rendering process of Gaussian Splatting, we need to project each 3D Gaussian distribution onto the 2D image plane to compute its contribution to pixel color. However, due to the infinite support of the Gaussian function, we typically limit the effective range of influence to within three standard deviations. Although this constraint improves rendering efficiency, it still cannot explicitly represent the complete point set on the ellipsoidal surface. As a result, standard depth-buffer-based rasterization is incapable of accurately determining whether a Gaussian intersects with the clipping plane. Our method addresses this shortcoming using ray-tracing for the clipped Gaussians.

In this section, we first introduce Gaussian Splatting formally (Sec. 3.1), before presenting our approach to facilitate clipping during rendering. This approach begins by identifying the Gaussians that can be trivially rendered or discarded, because they are completely on one side of the clipping plane (Sec. 3.2). Next, we focus on the Gaussians that are intersected by the clipping plane. For these Gaussians, we compute their exact intersection with the clipping plane (Sec. 3.3) and adapt their appearance based on this intersection using a decaying function (Sec. 3.4). The full method is summarized in [algorithm 1](#).

3.1 Gaussian Splatting

3D Gaussian Splatting Kerbl et al., 2023 is a framework for 3D reconstruction using point clouds with Gaussian kernels. Each point is represented by a Gaussian centered at $\boldsymbol{\mu} \in \mathbb{R}^3$ with a covariance matrix \mathbf{V} which can be represented by the scale and rotation matrix as $\mathbf{V} = \mathbf{R}\mathbf{S}^2\mathbf{R}^T$. The Gaussian kernel is defined as:

$$G(\mathbf{x}, \boldsymbol{\mu}, \mathbf{V}) = \frac{1}{(2\pi)^{3/2}|\mathbf{V}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{V}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right). \quad (1)$$

Each point has attributes such as transparency α_k , opacity density δ_k , and color c_k . α_k is calculated by δ_k , shown in Sec. 3.4. The rendering process involves projecting the point cloud into camera space and performing splatting operations to generate the final image. To blend color attributes accurately, points are sorted by depth, and pairwise compositing is applied to account for trans-

parency. The compositing formula is:

$$C = \sum_{k \in N} c_k \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \quad (2)$$

3.2 Identifying non-trivial Gaussians

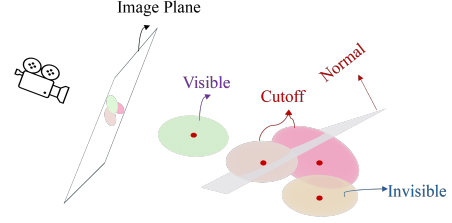


Figure 3. Multiple Gaussians classified in different categories in the scene. The visible region is defined as the half-space in the direction of the clipping plane’s normal.

In order to clip these Gaussian representations, we first aim to categorize the set of Gaussians, to decide if they will be discarded, rasterized, or ray-traced, based on where their centers lie in comparison to the clipping plane. Our goal is to restrict expensive ray-tracing to the Gaussians that cannot be trivially handled through rasterization or discarding. To distinguish the Gaussians, we begin by defining the clipping region using an implicit plane:

$$\pi(\mathbf{x}) = \mathbf{n}\mathbf{x} + d = 0 \quad (3)$$

where $\mathbf{n} \in \mathbb{R}^3$ is the normal of the clipping plane and $d \in \mathbb{R}$ is the signed distance from the origin to the plane. In a scene containing multiple Gaussians, as illustrated in [Figure 3](#), where we define the area the normal pointing to is the visible region for the viewer, each Gaussian G_k with center $\boldsymbol{\mu}_k$ can be classified into one of three categories based on its signed distance to the clipping plane:

1. **Invisible:** If the center is far behind the clipping plane, i.e., $d_{e.p.} = \mathbf{n}\boldsymbol{\mu}_k + d < -r_k$, where $r_k = 3 \max(\sigma_{x_k}, \sigma_{y_k}, \sigma_{z_k})$, the Gaussian is discarded. σ is the standard deviation. $d_{e.p.}$ denotes the signed distance from the Gaussian center (ellipsoid) to the clipping plane.
2. **Visible:** If the center lies clearly in front of the plane, i.e., $d_{e.p.} > r_k$, the Gaussian is fully visible and rasterized normally.
3. **Cutoff:** If the center is within the proximity of the clipping boundary, i.e., $|d_{e.p.}| \leq r_k$, the Gaussian is partially visible. We compute precise intersections and modify the appearance.

Note that some Gaussians could be falsely classified in the cutoff region, as we ignore anisotropy by using the maximum σ for this classification. To reduce computational cost and keep rendering fidelity, we limit exact intersection tests to Gaussians in the cutoff category. These are the only cases where visibility is ambiguous and visual adaptation of the Gaussians is required.

3.3 Ray-Gaussian and Clipping Plane Intersection

To enable precise clipping of Gaussian ellipsoids, we now derive the exact intersections of a viewing ray with the clipped Gaussians, as well as the clipping plane. To compute the intersection

with the Gaussian, we model the transformed Gaussian using a linear transformation from the unit sphere. Specifically:

- An ellipsoid is represented as $E(S, \mathbf{R}, \boldsymbol{\mu})$, where S is the scale matrix, \mathbf{R} is the rotation matrix, and $\boldsymbol{\mu}$ is the center.
- The transformation matrix $\mathbf{M} = \mathbf{R}\mathbf{S}$ maps the unit sphere to the ellipsoid, and the transformed point is given by $\mathbf{x} = \mathbf{M}\mathbf{u} + \boldsymbol{\mu}$ with $\mathbf{u} \in \mathbb{S}^2$, where \mathbb{S}^2 represents the unit sphere embedded in \mathbb{R}^3 .

Consider a ray $\mathbf{r}(t) = \mathbf{e} + t\mathbf{d}$, where \mathbf{e} is the ray origin, \mathbf{d} is the normalized direction vector, and $t \geq 0$. Transform this ray into the local space of the unit sphere:

$$\tilde{\mathbf{r}}(t) = \mathbf{M}^{-1}(\mathbf{e} - \boldsymbol{\mu}) + t\mathbf{M}^{-1}\mathbf{d} \quad (4)$$

We then solve for the intersection of this ray with the unit sphere using:

$$|\tilde{\mathbf{r}}(t)|^2 = 1 \quad (5)$$

which leads to the quadratic equation:

$$At^2 + Bt + C = 0 \quad (6)$$

With coefficients:

$$\begin{cases} A = \|\mathbf{M}^{-1}\mathbf{d}\|^2 \\ B = 2\langle \mathbf{M}^{-1}(\mathbf{e} - \boldsymbol{\mu}), \mathbf{M}^{-1}\mathbf{d} \rangle \\ C = \|\mathbf{M}^{-1}(\mathbf{e} - \boldsymbol{\mu})\|^2 - 1 \end{cases}$$

Solving this quadratic gives the two intersection parameters (for simplicity, here we only consider the case of two intersection points):

$$t_{-e_{1,2}} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (7)$$

Substituting t_{-e_1} and t_{-e_2} into $\mathbf{r}(t)$ gives the two world-space intersection points with the transformed ellipsoid.

Lastly, we compute the intersection of the ray with the clipping plane to determine if and where the Gaussian is actually clipped. To achieve this, we solve for the intersection of the ray with the clipping plane $\pi(x) = 0$. Substituting $\mathbf{r}(t)$ into the plane equation yields:

$$t_{-p} = \frac{-(\mathbf{n}\mathbf{e} + d)}{\mathbf{n}\mathbf{d}} \quad (8)$$

If the resulting t_{-p} is between t_{-e_1} and t_{-e_2} , the ray intersects the ellipsoid and is also clipped by the plane. Otherwise, the Gaussian is either fully visible or fully occluded and treated accordingly. In the following section, we will use these intersection points to determine how clipped Gaussians need to be adjusted visually.

3.4 Decaying Function Design

In the standard rendering formulation of Gaussian Splatting, each 3D Gaussian splat contributes to pixel color by integrating its projected density along the ray direction. For any position in the space $\mathbf{x} \in \mathbb{R}^3$, and its corresponding pixel coordinate $\hat{\mathbf{x}} \in \mathbb{R}^2$ in the image space, we can compute the opacity contribution for this pixel of any Gaussian points $\boldsymbol{\mu}_k \in \mathbb{R}^3$ by Zwicker et al., 2001:

$$\alpha_k(\hat{\mathbf{x}}) = \delta_k \int_{\mathbb{R}} \frac{1}{|\mathbf{J}^{-1}| |\mathbf{W}^{-1}|} G(\mathbf{x}, \boldsymbol{\mu}_k, \mathbf{J}\mathbf{W}\mathbf{V}_k \mathbf{W}^T \mathbf{J}^T) dx_z \quad (9)$$

where δ_k indicates the opacity density, \mathbf{J} denotes the Jacobian matrix of the projective transformation, and \mathbf{W} is the transformed matrix from world to camera space. \mathbf{V}_k represents the 3D Gaussian covariance matrix in world space, while $G(\cdot)$ is the 3D Gaussian kernel function. $\hat{\boldsymbol{\mu}}_k$ denotes the projection of the Gaussian center in the image space, and \mathbf{x}_z is the projection direction.

We can directly get the analytical result of Equation 9 as:

$$\alpha_k(\hat{\mathbf{x}}) = \frac{1}{|\mathbf{J}^{-1}| |\mathbf{W}^{-1}|} \delta_k \hat{G}(\hat{\mathbf{x}}, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{V}}_k), \quad (10)$$

where \hat{G} is the 2D Gaussian kernel, $\hat{\mathbf{V}}_k$ is obtained by skipping the last row and column of $\mathbf{J}\mathbf{W}\mathbf{V}_k \mathbf{W}^T \mathbf{J}^T$.

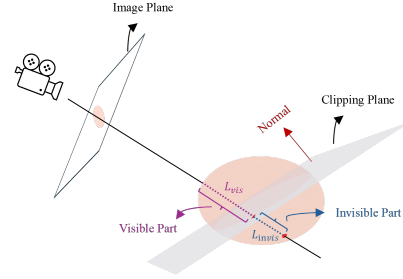


Figure 4. Illustration about calculation for decay weight for cutoff Gaussians.

When a Gaussian is partially removed by a clipping plane, like in Figure 4, we cannot use the analytic result shown in Equation 10 and need to compensate for the missing portion. To approximate the residual visible contribution, we propose a 1D length-ratio-based decay function $w(\hat{\mathbf{x}})$. The key idea is to estimate the proportion of the Gaussian volume contributing to opacity by measuring the length of the visible segment along the ray within the Gaussian. As shown in Figure 4, we define L_{vis} as the segment between ray-ellipsoid intersections that lies within the visible region, and L_{invis} as the portion falling in the invisible region. The full extent of the intersection path through the ellipsoid is then given by $L_{\text{vis}} \cup L_{\text{invis}}$. We define the decay weight function as:

$$w(\hat{\mathbf{x}}) = \begin{cases} \frac{L_{\text{vis}}}{L_{\text{vis}} \cup L_{\text{invis}}}, & \text{if two intersections exist} \\ 1, & \text{otherwise (keep original rasterization)} \end{cases} \quad (11)$$

Based on the decay weight $w(\hat{\mathbf{x}})$, we can then update the opacity of the Gaussian to adjust its contribution to the pixel:

$$\alpha_k^{\text{new}} = \alpha_k(\hat{\mathbf{x}})w(\hat{\mathbf{x}}) \quad (12)$$

This adjusted opacity is then used in the final splatting step. This approach ensures that clipping operations do not introduce visible discontinuities or flickering along the edges. The core logic of the full RaRa strategy is elucidated in algorithm 1.

4 Experiment

In this section, we experimentally validate the effectiveness of our algorithm. We first introduce the evaluation datasets, then verify that our clipping approach retains the real-time performance of Gaussian Splatting. Ablation studies further reveal that naively

ALGORITHM 1: RaRaClip: Gaussian Visibility and Decaying Weight Computation

Input: Gaussian parameters μ_i, R_i, S_i , clip plane (\mathbf{n}, d) , camera pose T , image resolution $W \times H$, pixel coordinate (p_x, p_y)

Output: Opacity contribution α_i

// Step 1: Visibility Pre-check (Hard Clipping)

$d_{vis} \leftarrow \text{dot}(\mathbf{n}, \mu_i) + d; r_i \leftarrow 3 \max(S_i);$

if $d_{vis} < -r_i$ **then**

 | **return** $\alpha_i \leftarrow 0;$ // Completely invisible

end

// Step 2: RaRa Strategy - Ray Tracing for Cutoff

if $|d_{vis}| \leq r_i$ **then**

$\mathbf{d} \leftarrow \text{ProjectRay}(p_x, p_y, T);$ // pixel to ray direction

$\mathbf{e} \leftarrow \text{CameraOrigin}(T)$

$(\tilde{\mathbf{d}}, \tilde{\mathbf{e}}) \leftarrow \text{TransformToUnitSphere}(\mathbf{d}, \mathbf{e}, M_i^{-1}, \mu_i)$

$(t_{e1}, t_{e2}) \leftarrow \text{RayEllipsoidIntersect}(\tilde{\mathbf{d}}, \tilde{\mathbf{e}})$

$t_p \leftarrow \text{RayPlaneIntersect}(\mathbf{e}, \mathbf{d}, \mathbf{n}, d)$

if $t_{e1}, t_{e2} > 0$ **and** $t_p > 0$ **then**

$\mathbf{x}_1 = \mathbf{e} + t_{e1}\mathbf{d}, \mathbf{x}_2 = \mathbf{e} + t_{e2}\mathbf{d};$ // Compute 3D Points

if $\text{IsVisible}(\mathbf{x}_1, \mathbf{n}, d)$ **AND** $\text{IsVisible}(\mathbf{x}_2, \mathbf{n}, d)$ **AND** $t_{e1} < t_p < t_{e2}$ **then**

 | $w \leftarrow 1.0;$

end

else if $(\text{IsVisible}(\mathbf{x}_1, \mathbf{n}, d)$ **OR** $\text{IsVisible}(\mathbf{x}_2, \mathbf{n}, d))$ **then**

 | $w \leftarrow L_{vis} / (L_{vis} \cup L_{invis})$

end

end

end

else

 | $w \leftarrow 1.0;$ // Fully visible, keep opacity

end

// Step 3: Final Opacity Computation

$\alpha_i^{\text{orig}} \leftarrow \text{Equation 10}$

$\alpha_i \leftarrow \min(0.99, \alpha_i^{\text{orig}} w)$

return α_i

applying ray tracing degrades quality, while our RaRa strategy effectively mitigates such artifacts. All experiments are conducted on an NVIDIA RTX 4090 GPU.

4.1 Dataset

We evaluate our method on three representative types of Gaussian datasets:

General Gaussian scenes are from SplatViz [Barthel et al., 2024](#) framework using Mip-NeRF360, representing typical surface-level reconstructions.

Strands Gaussian scene is derived from the GaussianHair [Zakharov et al., 2024](#), containing highly anisotropic primitives to showcase fine-scale effects such as hair clipping.

Multi-layer Gaussian scenes are based on anatomical reconstructions [Kleinbeck et al., 2025](#) with layered Gaussian fields, enabling visualization of internal clipping within volumetric structures.

4.2 Render Efficiency

To ensure that our proposed clipping framework maintains real-time interactivity, we evaluate its runtime performance under dynamic clipping scenarios. To further verify the scalability of our method under dense scenes, we conduct evaluations on two large-scale datasets: 360garden (~4 million Gaussians) and Hair (~3 million Gaussians). Results in [Table 1](#) show that our method achieves comparable frame rates to hard clipping and preserves real-time interactivity even in dense point-based representations. Original denotes the baseline rendering performance measured at a fixed camera view without any clipping. For both Hard Clipping and our RaRa Clipping method (Ours), we introduce a single clipping plane and gradually move it across the entire scene over the course of 30 frames, while keeping the camera view unchanged. The reported FPS values are averaged over these 30 frames to reflect interactive rendering performance under active clipping.

FPS	360garden (~4M)			Hair (~3M)		
	Original	Hard	Ours	Original	Hard	Ours
	77.82	81.49	81.47	128.02	146.88	145.46

Table 1

Average rendering performance (FPS) under dynamic clipping.

4.3 Ablation Study

To validate the necessity and correctness of the proposed RaRa strategy, we conduct an ablation study that compares rendering results with and without it. Our goal is to demonstrate that ray tracing should only be applied to Gaussians that lie near the clipping boundary, as naively tracing all Gaussians may introduce visual inconsistencies.

Since the exact ground truth of a scene after partial clipping is hard to define, we simulate a reference condition by placing the clipping plane at infinity. This ensures that no Gaussians are clipped and the entire scene remains fully visible. Under this assumption, the correct rendering should match that of the original unmodified Gaussian scene.

We compare the following two strategies:

- **wo RaRa**: ray tracing and attenuation weights are performed on all visible Gaussians.
- **w RaRa**: ray tracing is only applied to Gaussians that are classified as *cutoff*.

As shown in Table 2, the RaRa strategy results in perfect alignment with the ground truth, with zero L1 error and SSIM of 1.0 on both datasets because no cutoff Gaussians exist. In contrast, without RaRa, the results exhibit numerical deviation and perceptual distortion, particularly along Gaussian boundaries as shown in Figure 6.

	360bonsai		Hair	
	L1	SSIM	L1	SSIM
wo RaRa	8.5359	0.5759	0.3210	0.9968
w RaRa	0.0	1.0	0.0	1.0

Table 2

Ablation study comparing rendering results with and without the RaRa strategy under a simulated infinite clipping plane. Our method ensures perfect consistency with the ground truth.

5 User Study

To systematically evaluate the user experience of our clipping strategies, we conducted a controlled user study comparing RaRa Clip—our proposed clipping method—with hard clipping.

5.1 Procedure

The entire user study was carried out *online* by our deployed website. Before starting the task, each participant read a concise task description as well as step-by-step instructions that covered the clipping techniques to be compared, the rating workflow, and all privacy guidelines.

Participants. We recruited a total of 22 participants (P1 to P22, 8 female, 14 male; mean age = 26.1 ± 3.9 years). In terms of academic background, 55% held a Master’s degree, 27% were PhD candidates or graduates, and 18% held a Bachelor’s degree. Regarding computer-graphics experience, 55% reported no prior CG experience, 23% were beginners, 14% intermediate users, and 9% experts. This distribution offers a balanced mix of novices and specialists for evaluating our methods across varying expertise levels. They were rewarded with gift vouchers worth approximately 20 USD.

Stimuli. We collected **20 evaluation cases** drawn from **eight Gaussian splatting datasets**. Every case contained two strictly matched visualizations:

- A static rendering produced with the *Hard Clip* method.
- A static rendering produced with our *RaRa Clip* method.
- A short animation of the cutting plane sweeping through the datasets (shared by both methods).

Camera pose, cutting-plane orientation, and the plane’s start/end positions were kept *identical* between the two methods, ensuring that the clipping algorithm was the *only* varying factor.

Evaluation metrics. Each case was assessed on the following six subjective dimensions:

1. **Flatness / Smoothness**: Degree to which the clipping surface is visually continuous and free of aliasing, stair-steps, or jagged curvature.
2. **Temporal Stability**: The degree to which the rendered surface evolves smoothly and continuously as the clipping plane moves, without popping artifacts or abrupt changes in visibility that disrupt user perception.
3. **Topological Integrity**: Absence of unintended holes, breaks, or floating fragments that would disrupt anatomical or structural continuity.
4. **Naturalness**: How seamlessly the clipped surface blends with surrounding textures, giving the impression of a “natural dissection” rather than an artificial cut.
5. **Region of Interest (ROI) Exposure**: Extent to which the region of interest is revealed clearly, without over-occluding adjacent important structures.
6. **Context Completeness**: The surrounding anatomy or environment is left visible to help users stay oriented.

Fatigue control. Participants were randomly divided into two equally sized groups. Each group rated exactly **three metrics** per case, so that no participant completed more than 60 case–metric combinations, keeping the session within a comfortable time window.

Rating workflow. For every *case–metric* combination, the following two-step procedure was performed:

1. **Forced choice.** The two results were displayed side-by-side in random order. The participant selected the method that performed better on the current metric.
2. **Strength rating.** Immediately afterwards, the participant indicated *how much* better the chosen method was, using a **7-point Likert scale** (1 = “barely better”, 7 = “extremely better”).

Post-study questionnaire. After completing all rating tasks, participants answered three open-ended questions to gather qualitative insights:

1. *What key properties should an ideal clipping possess to support effective data exploration and interpretation?*
2. *What limitations or missing capabilities can be identified in the current two clipping methods?*
3. *What strengths or advantageous features do the current two clipping methods offer?*

All interactions and timestamps were logged automatically for subsequent analysis.

5.2 Result

Across all six perceptual metrics, participants exhibited a clear and consistent preference for **RaRa Clip** over **Hard Clip**. Median ratings fell between 3 and 4 on the $[-7, 7]$ scale, and Wilcoxon signed-rank [Rosner et al., 2006](#) tests confirmed that every metric differed significantly from 0 in favor of **RaRa Clip** (all $p < .001$).

The largest effects emerged for *Topological Integrity* ($M = 3.53 \pm 2.47$, $d = 1.43$, 91% positive responses) and *Context Completeness*

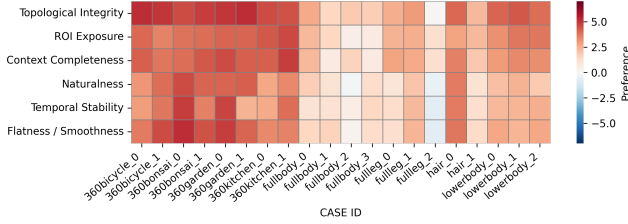


Figure 5. Heatmap of per-case preference scores for the six evaluation metrics; columns represent the 20 cases, rows the metrics, with blue cells indicating stronger favour toward Hard Clip and red cells toward RaRa Clip.

($M = 3.04 \pm 2.28$, $d = 1.33$, 91% positive), indicating that users strongly valued preservation of structural relationships and surrounding context. Here, M denotes the mean rating with standard deviation, and d is the effect size [Cohen, 2013](#), reflecting the magnitude of difference compared to the *Hard Clip*. *ROI Exposure* also showed a large effect ($M = 2.95 \pm 2.44$, $d = 1.21$, 90% positive), suggesting that the proposed technique improves visibility of regions of interest without sacrificing context. The user rating results are demonstrated in the [Figure 7](#) and [Figure 8](#).

Moderate, yet still substantial, benefits were observed for *Flatness/Smoothness* ($M = 2.70 \pm 3.06$, $d = 0.88$, 82% positive), *Naturalness* ($M = 2.36 \pm 3.05$, $d = 0.78$, 81% positive) and *Temporal Stability* ($M = 2.38 \pm 3.04$, $d = 0.78$, 80% positive). Across all metrics, the proportion of negative ratings never exceeded 20%, underscoring the robustness of user preference for **RaRa Clip**. Collectively, these results demonstrate statistically and practically significant gains in perceptual quality, with particularly strong advantages in maintaining topological fidelity, contextual completeness, and region-of-interest visibility.

We further conducted a case-by-case analysis across the six evaluation metrics, as shown in [Figure 5](#). Red indicates a user preference for **RaRa Clip**, while blue indicates a preference for **Hard Clip**. The results show that, for nearly all cases and metrics, users consistently preferred RaRa Clip. An exception was observed in *Case 2* of the *full leg* dataset, where users rated Hard Clip higher in terms of *naturalness*, *temporal stability*, and *flatness*. Additionally, the preference for RaRa Clip was more pronounced in real-world datasets such as *bicycle*, *bonsai*, and *kitchen*, while the preference was slightly less evident in medical datasets.

5.3 User Feedback

To complement the qualitative results, we collected open-ended responses from participants regarding their preferences and expectations for effective clipping plane methods. Several key themes emerged:

Smoothness and Visual Continuity. Many users emphasized that an ideal clipping plane should be visually smooth and free of jagged edges or flickering artifacts. P3 specifically noted that one method maintained clear boundaries even during motion, while the other showed sudden changes or ghosting effects. P4 further pointed out that one method produced smoother transitions and fewer aliasing artifacts compared to another method.

Precision and Low Distortion. Users stressed the importance of preserving the geometric and perceptual fidelity of the clipped surface. This includes maintaining relative distances and orien-

tations between data points to ensure the interpretability of the visualization. P13 explicitly likened the need for clarity to medical imaging, where preserving structure without overwhelming detail is critical.

Dataset-dependent Preferences. Participants (e.g., P6, P15, P16) observed that the effectiveness of each method varies by dataset. Viewpoint also played a role—differences between methods became more apparent when the cutting plane was not perpendicular to the camera.

Advantages and Limitations. Several users noted that both methods were accurate in general (P13, P21), but they also pointed out limitations such as insufficient background context preservation (P13) or visual inconsistencies near the clipping boundary (P11).

6 Discussion

Underperforming scenarios: While our RaRa clipping strategy consistently outperforms hard clipping across most datasets and metrics, we observed a notable outlier in the *full_leg_2* viewpoint of the full-body dataset shown in [Figure 5](#). In this case, participants in the user study slightly favored the hard clipping results. We attribute this to two key factors: (1) To ensure side-by-side comparison in our web-based user study interface, we downsampled the rendering resolution from 1024×1024 to 512×512 . This resolution reduction likely caused the fine-grained differences, especially in boundary smoothness, to become less perceptible to users. (2) The Gaussians in the full-leg dataset are inherently small and compact. In such cases, both hard clipping and RaRa strategies yield similar visual results due to limited clipping overlap per primitive.

Applicable scenarios: RaRa is particularly suited for multi-layer Gaussian fields, where internal structures—not just surfaces—are of interest. While original Gaussian Splatting focuses on surface appearance, clipping enables volumetric exploration. As Gaussian-based representations evolve toward anatomy visualization or semantic layer decomposition, the need for precise and artifact-free clipping will grow, making RaRa increasingly relevant.

Limitations: Despite its generality and compatibility, our method currently supports only planar clipping surfaces. However, our method can be extended to support other clipping primitives, provided that efficient ray-primitive intersection tests are available. Moreover, our method operates only at the rendering level—the clipped result exists only during visualization and is not baked into the underlying data representation. Future work could explore methods to re-fit or re-optimize Gaussians post-clipping to persistently store editable or extractable subregions.

7 Conclusion

We presented RaRa Clipper, a novel clipping strategy for Gaussian Splatting that seamlessly integrates rasterization-based visibility classification with selective ray tracing. By applying ray-ellipsoid intersection only to Gaussians near the clipping boundary, and computing visibility-aware opacity using a decaying function, our method achieves high-fidelity, smooth, and temporally stable clipping without sacrificing real-time rendering performance. Extensive experiments and user studies confirm that our method outperforms the hard clipping strategy in both objective accuracy and perceptual quality. RaRa Clipper is plugin-compatible with exist-

ing GS renderers and provides an essential step toward precise, editable, and interactive Gaussian volume exploration.

References

- Barron, Jonathan T et al. (2021). "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5855–5864.
- Barthel, Florian et al. (2024). *Gaussian Splatting Decoder for 3D-aware Generative Adversarial Networks*. arXiv: 2404.10625 [cs.CV].
- Byrski, Krzysztof et al. (2025). "REdiSplats: Ray Tracing for Editable Gaussian Splatting". In: *arXiv preprint arXiv:2503.12284*.
- Chen, Guikun and Wenguan Wang (2024). "A survey on 3d gaussian splatting". In: *arXiv preprint arXiv:2401.03890*.
- Cohen, Jacob (2013). *Statistical power analysis for the behavioral sciences*. routledge.
- Duan, Yuanxing et al. (2024). "4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes". In: *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11.
- Gao, Jian et al. (2024). "Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing". In: *European Conference on Computer Vision*. Springer, pp. 73–89.
- Hu, Shoukang, Tao Hu, and Ziwei Liu (2024). "Gauhuman: Articulated gaussian splatting from monocular human videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20418–20431.
- Kerbl, Bernhard et al. (2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering." In: *ACM Trans. Graph.* 42.4, pp. 139–1.
- Kivi, Petrus EJ et al. (2022). "Real-time rendering of point clouds with photorealistic effects: a survey". In: *IEEE Access* 10, pp. 13151–13173.
- Kleinbeck, Constantin et al. (2025). "Multi-layer gaussian splatting for immersive anatomy visualization". In: *IEEE Transactions on Visualization and Computer Graphics*.
- Konev, Artem et al. (2018). "Fast cutaway visualization of sub-terrain tubular networks". In: *Computers & Graphics* 75, pp. 25–35.
- Li, Jiameng et al. (2024). "Mipmap-GS: Let Gaussians Deform with Scale-specific Mipmap for Anti-aliasing Rendering". In: *arXiv preprint arXiv:2408.06286*.
- Li, Zhe et al. (2024). "Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19711–19722.
- Lin, Youtian et al. (2024). "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21136–21145.
- Liu, Haitian et al. (2025). "A review on 3D Gaussian splatting for sparse view reconstruction". In: *Artificial Intelligence Review* 58.7, p. 215.
- Luo, Haimin et al. (2024). "Gaussianhair: Hair modeling and rendering with light-aware gaussians". In: *arXiv preprint arXiv:2402.10483*.
- Mai, Alexander et al. (2024). "Ever: Exact volumetric ellipsoid rendering for real-time view synthesis". In: *arXiv preprint arXiv:2410.01804*.
- Mildenhall, Ben et al. (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1, pp. 99–106.
- Moenne-Loccoz, Nicolas et al. (2024). "3D Gaussian Ray Tracing: Fast Tracing of Particle Scenes". In: *ACM Transactions on Graphics (TOG)* 43.6, pp. 1–19.
- Niedermayr, Simon et al. (2024). "Application of 3d gaussian splatting for cinematic anatomy on consumer class devices". In: *arXiv preprint arXiv:2404.11285*.
- Pang, Haokai et al. (2024). "Ash: Animatable gaussian splats for efficient and photoreal human rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1165–1175.
- Qian, Zhiyin et al. (2024). "3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5020–5030.
- Ren, Kerui et al. (2024). "Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians". In: *arXiv preprint arXiv:2403.17898*.
- Rosner, Bernard, Robert J Glynn, and Mei-Ling T Lee (2006). "The Wilcoxon signed rank test for paired comparisons of clustered data". In: *Biometrics* 62.1, pp. 185–192.
- Seo, Yunji et al. (2024). "FLoD: Integrating Flexible Level of Detail into 3D Gaussian Splatting for Customizable Rendering". In: *arXiv preprint arXiv:2408.12894*.
- Shirley, Peter, Michael Ashikhmin, and Steve Marschner (2009). *Fundamentals of computer graphics*. AK Peters/CRC Press.
- Sun, X et al. (2025). "Stochastic Ray Tracing of 3D Transparent Gaussians". In: *arXiv preprint arXiv:2504.06598*.
- Tang, Kaiyuan, Siyuan Yao, and Chaoli Wang (2025). "iVR-GS: Inverse Volume Rendering for Explorable Visualization via Editable 3D Gaussian Splatting". In: *IEEE Transactions on Visualization and Computer Graphics*.
- Tran, Chan-Hung (1986). "Fast clipping algorithms for computer graphics". PhD thesis. University of British Columbia.
- Wang, Jiarun and Fan Yang (2023). "BLH-Slice Modeling Used in Human-Computer Interaction of Ellipsoidal Volume Clipping". In: *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*. IEEE, pp. 267–273.
- Weiskopf, Daniel, Klaus Engel, and Thomas Ertl (2003). "Interactive clipping techniques for texture-based volume visualization and volume shading". In: *IEEE Transactions on Visualization and Computer Graphics* 9.3, pp. 298–312.
- Westover, Lee Alan (1991). *Splatting: a parallel, feed-forward volume rendering algorithm*. The University of North Carolina at Chapel Hill.
- Wu, Qi et al. (2024). "3DGUT: Enabling Distorted Cameras and Secondary Rays in Gaussian Splatting". In: *arXiv preprint arXiv:2412.12507*.
- Yang, Ziyi et al. (2024). "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20331–20341.
- Zakharov, Egor et al. (2024). "Human hair reconstruction with strand-aligned 3d gaussians". In: *European Conference on Computer Vision*. Springer, pp. 409–425.
- Zhou, Yuxiao et al. (2024). "Groomcap: High-fidelity prior-free hair capture". In: *ACM Transactions on Graphics (TOG)* 43.6, pp. 1–15.
- Zwicker, Matthias et al. (2001). "EWA volume splatting". In: *Proceedings Visualization, 2001. VIS'01*. IEEE, pp. 29–538.

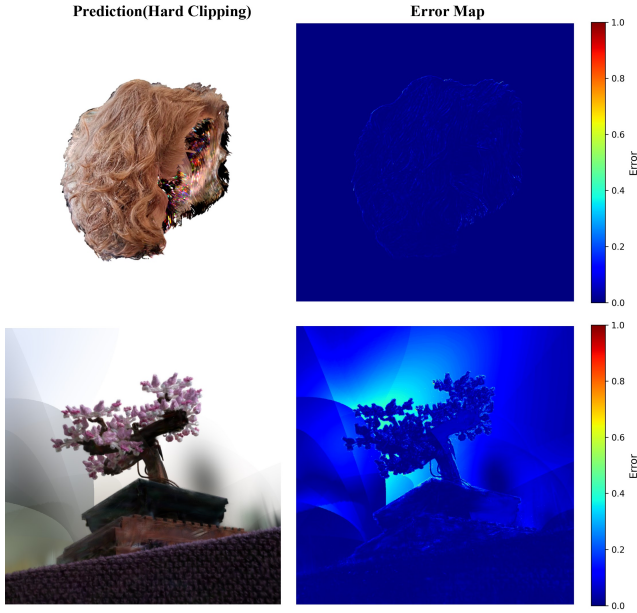


Figure 6. Error visualization for the hard clipping strategy. Top row: hair dataset; Bottom row: 360bonsai dataset. We omit the RaRa strategy here, as it produces artifact-free rendering under our setup and yields zero error.

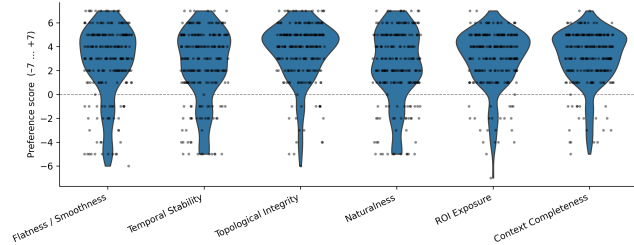


Figure 7. Violin plots users' ratings for the six metrics; positive scores favor RaRa Clip, revealing a clear overall bias toward it. Strip plots with a jitter in the figure show the original data amount.

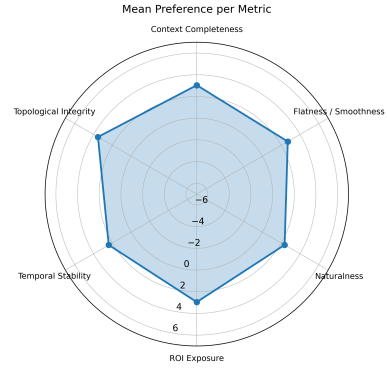


Figure 8. Radar chart of the mean preference scores for each of the six evaluation metrics; larger radii indicate stronger favor toward RaRa Clip, showing that participants consistently preferred it across all metrics.

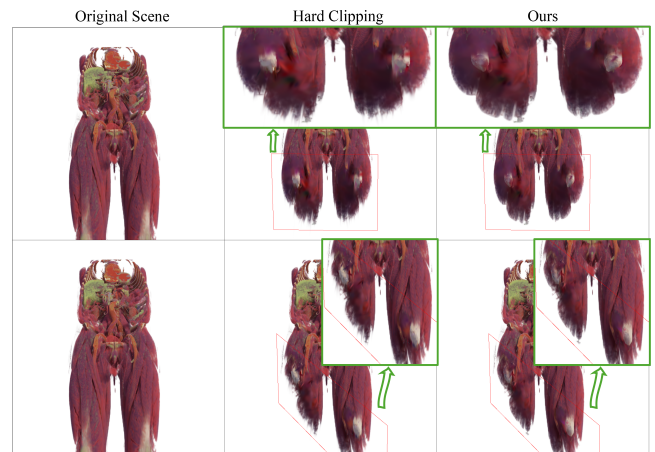


Figure 9. Clipping comparison on the full body dataset using different clipping strategies and clipping planes. Our method preserves internal structures and soft tissue continuity across boundaries, whereas hard clipping introduces aliasing and visible distortions.

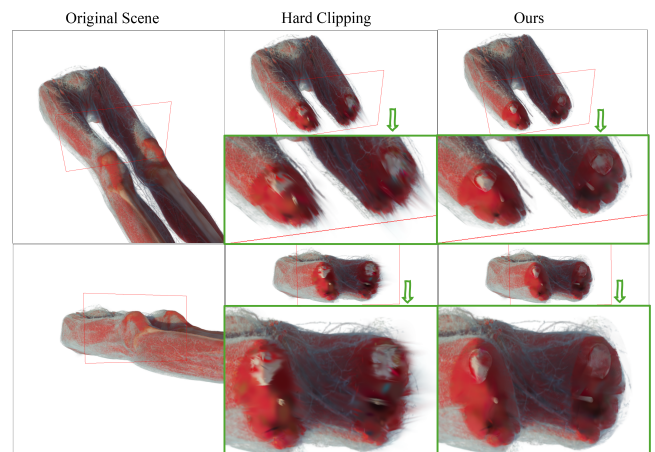


Figure 10. Comparison of clipping results on the lower body dataset using different clipping strategies and views. Each row presents a different camera viewpoint, but under the same clipping plane configuration. Our method effectively preserves structural integrity and surface smoothness near the clipping interface, while hard clipping produces visible artifacts such as aliasing and detail loss.

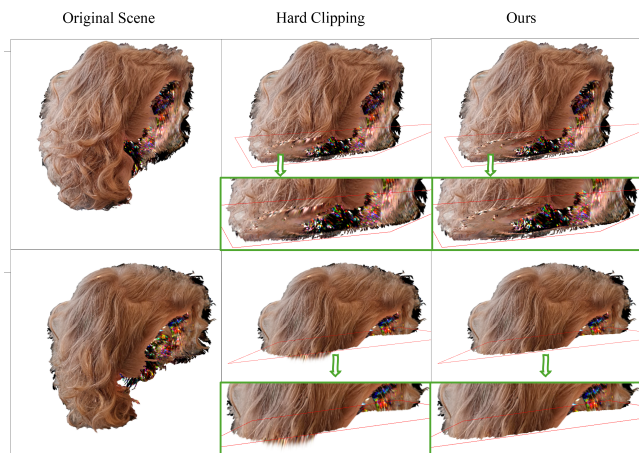


Figure 11. Clipping comparison on the Hair dataset using different clipping strategies. Our method eliminates artifacts along the clipping boundary, while the hard clipping approach introduces noticeable spiky distortions at the ends of hair strands. Each row corresponds to a different clipping plane configuration.



Figure 12. Clipping results on general appearance modeling datasets, showcasing the impact of different strategies near the clipping boundary. Hard clipping introduces spiky artifacts, particularly visible in edge structures. In contrast, our method produces clear contour cutting. The last row shows results on the 360bonsai dataset, where the red line indicates the visualized clipping plane.