

AdR-Gaussian: Accelerating Gaussian Splatting with Adaptive Radius (Supplementary Material)

XINZHE WANG, Shanghai Jiao Tong University, China

RAN YI, Shanghai Jiao Tong University, China

LIZHUANG MA, Shanghai Jiao Tong University, China

1 OVERVIEW

In this supplementary material, more details about the proposed AdR-Gaussian method and more experimental results are provided, including:

- The detailed computation of adaptive radius. (Sec. 2)
- The detailed computation of axis-align bounding box tailored for Gaussian splatting. (Sec. 3)
- Detailed comparisons with other methods, especially for the sub methods focus on accelerating. (Sec. 4)
- The qualitative result for ablation. (Sec. 5)

2 DETAILED COMPUTATION OF ADAPTIVE RADIUS

To early cull pairs of Gaussian-Tile with low splatting opacity during *Preprocess* stage based on adaptive radius, which is the radius of the bounding circle of a ellipse constructed by minimum splatting opacity α_{low} , we need to begin with the culling inequality:

$$\alpha_i \geq \alpha_{low}, \quad (1)$$

where α_i represents the splatting opacity of Gaussian with index i at the current pixel.

Based on the concept of EWA [Zwicker et al. 2002], the splatting opacity, the splatting opacity α_i for each Gaussian-Pixel pair is calculated by the Gaussian's opacity σ_i , pair's 2D distance \mathbf{x} and projected covariance matrix Σ' :

$$\alpha = \sigma e^{-\frac{1}{2} \mathbf{x}^T \Sigma'^{-1} \mathbf{x}}. \quad (2)$$

By substituting Eq. (2) into the above inequality Eq. (1), we can get a quadratic expression, with calculation procedures as below:

$$\begin{aligned} \sigma_i e^{-\frac{1}{2} \mathbf{x}^T \Sigma'^{-1} \mathbf{x}} &\geq \alpha_{low}, \\ e^{\frac{1}{2} \mathbf{x}^T \Sigma'^{-1} \mathbf{x}} &\leq \frac{\sigma_i}{\alpha_{low}}, \\ \mathbf{x}^T \Sigma'^{-1} \mathbf{x} &\leq 2 \ln \left(\frac{\sigma_i}{\alpha_{low}} \right). \end{aligned} \quad (3)$$

With 2D distance \mathbf{x} expressed as its 1D components (x, y) , and projected covariance Σ' expressed as $[\Sigma'_X, \Sigma'_{XY}; \Sigma'_{XY}, \Sigma'_Y]$, we can get a elliptical inequality wherein the x, y serve as, with calculation procedures as follows:

$$(x \ y) \begin{pmatrix} \Sigma'_X & \Sigma'_{XY} \\ \Sigma'_{XY} & \Sigma'_Y \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} \leq 2 \ln \left(\frac{\sigma_i}{\alpha_{low}} \right),$$

Authors' Contact Information: Xinzhe Wang, hiroxzhang@sjtu.edu.cn, Shanghai Jiao Tong University, Shanghai, China; Ran Yi, ranyi@sjtu.edu.cn, Shanghai Jiao Tong University, Shanghai, China; Lizhuang Ma, ma-lz@cs.sjtu.edu.cn, Shanghai Jiao Tong University, Shanghai, China.

$$(x \ y) \begin{pmatrix} \Sigma'_Y & -\Sigma'_{XY} \\ -\Sigma'_{XY} & \Sigma'_X \end{pmatrix}^{-1} / \det(\Sigma') \begin{pmatrix} x \\ y \end{pmatrix} \leq 2 \ln \left(\frac{\sigma_i}{\alpha_{low}} \right),$$

$$Ax^2 + By^2 + Cxy + D \leq 0, \quad (4)$$

where

$$A = \Sigma'_Y, \quad B = \Sigma'_X, \quad C = -2\Sigma'_{XY},$$

$$D = -2 \left(\Sigma'_X \Sigma'_Y - \Sigma'_{XY}^2 \right) \ln \left(\frac{\sigma_i}{\alpha_{low}} \right).$$

For Gaussian with index i , all pixels whose 2D distance satisfies the above inequality will be rendered (*i.e.*, within this ellipse), thus the farthest Euclidean distance between it and the pixels it rendered, can be expressed as $\sqrt{x^2 + y^2}_{max}$, which is equal to the bounding circle radius of this ellipse, *i.e.*, half of the ellipse's major axis length. We define the adaptive radius r_{ad} of a projected Gaussian as this distance, and calculated through the standard equation of ellipse: $\lambda_1 x^2 + \lambda_2 y^2 = 1$, from which we know that the half length of an ellipse's major axis is equal to the square root of the reciprocal of the smaller eigenvalue $\lambda_{ad} = \min(\lambda_1, \lambda_2)$ of the ellipse's covariance matrix Σ_{ad} . Based on the elliptical inequality (Eq. (4)), the ellipse's covariance matrix Σ_{ad} can be expressed as Eq. (5), while the computations of eigenvalues λ_1, λ_2 and adaptive radius λ_{ad} are as follows:

$$\Sigma_{ad} = \begin{pmatrix} -A/D & -C/2D \\ -C/2D & -B/D \end{pmatrix}, \quad (5)$$

$$|\Sigma_{ad} - \lambda I| = 0,$$

$$\begin{vmatrix} -A/D - \lambda & -C/2D \\ -C/2D & -B/D - \lambda \end{vmatrix} = 0,$$

$$(-A/D - \lambda)(-B/D - \lambda) - (-C/2D)^2 = 0,$$

$$D^2 \lambda^2 + (A+B)D\lambda + AB - \frac{C^2}{4} = 0,$$

$$\lambda = \frac{-(A+B) \pm \sqrt{(A-B)^2 + C^2}}{2D},$$

$$\lambda_{ad} = \lambda_{min} = \frac{-(A+B) + \sqrt{(A-B)^2 + C^2}}{2D}, \quad (6)$$

$$r_{ad} = \sqrt{\frac{1}{\lambda_{ad}}} = \sqrt{\frac{2D}{-(A+B) + \sqrt{(A-B)^2 + C^2}}}. \quad (7)$$

With elliptical coefficients substituted, the calculation procedure of adaptive radius is formulated as follows:

$$r_{ad} = \sqrt{\frac{4 \times (\Sigma'_X \Sigma'_Y - \Sigma'_{XY}^2) \ln \left(\frac{\sigma_i}{\alpha_{low}} \right)}{(\Sigma'_X + \Sigma'_Y) - \sqrt{(\Sigma'_X - \Sigma'_Y)^2 + (-2\Sigma'_{XY})^2}}}$$

$$r_{ad} = \sqrt{\frac{4 \times (\Sigma'_X \Sigma'_Y - \Sigma'_{XY}^2) \ln\left(\frac{\sigma_i}{\alpha_{low}}\right)}{\sqrt{(\Sigma'_X + \Sigma'_Y)^2 - \sqrt{\Sigma'^2_X + \Sigma'^2_Y - 2\Sigma'_X \Sigma'_Y + 4\Sigma'_{XY}^2}}}}, \\ r_{ad} = \sqrt{2\lambda'_{max} \ln\left(\frac{\sigma_i}{\alpha_{low}}\right)}, \quad (8)$$

where λ'_{max} is the larger eigenvalue of the projected covariance Σ' , which has been computed during the original calculation of confidence interval, formulated as follows:

$$\lambda'_{max} = \left(\frac{\Sigma'_X + \Sigma'_Y}{2} + \sqrt{\left(\frac{\Sigma'_X + \Sigma'_Y}{2}\right)^2 - \left(\Sigma'_X \Sigma'_Y - \Sigma'_{XY}^2\right)} \right). \quad (9)$$

Finally, there may be situations where the Gaussian opacity is high or the predefined minimum splatting opacity is low, resulting in an adaptive radius r_{ad} that exceeds the original radius r_o determined based on the 99% confidence interval. Therefore, the final value of the 2D Gaussian radius should be the minimum of these two values:

$$r = \min(r_{ad}, r_o). \quad (10)$$

3 DETAILED COMPUTATION OF AABB FOR 3DGS

To further cull the pairs of Gaussian-Tile with low splatting opacity in the minor-axis direction, as the adaptive radius is determined by the major axis and can not cull the tiles between the ellipse determined by Eq. (4) and its bounding circle, we propose axis-aligned bounding box for Gaussian splatting, which achieves a more significant reduction in ineffective expenses by accurately calculating the Gaussian size in the 2D directions and culling the tiles out of its axis-aligned bounding box. Specifically, half of the bounding box's width w and height h equal to the maximum values in the two coordinate directions of the ellipse, x_{max} and y_{max} , respectively:

$$\frac{w}{2} = x_{max}, \quad \frac{h}{2} = y_{max}. \quad (11)$$

To obtain the maximum value in the two coordinate directions of the ellipse, we first define a function F for the ellipse from its inequality form (Eq. (4)):

$$F = Ax^2 + By^2 + Cxy + D. \quad (12)$$

Based on the ellipse function, we calculate the corresponding partial derivatives in two coordinate directions respectively:

$$\frac{\partial F}{\partial x} = 2Ax + Cy, \quad \frac{\partial F}{\partial y} = 2By + Cx. \quad (13)$$

According to geometric properties of ellipse, when the ellipse function's partial derivative with respect to one coordinate direction is 0, the coordinate value in another direction attains its extremum. To get the extremum of the ellipse on both coordinates, we let the each partial derivative to be 0 to obtain the coordinate relationship of each extreme point:

$$2Ax + Cy = 0, \quad 2By + Cx = 0 \\ x = -\frac{Cy}{2A}, \quad y = -\frac{Cx}{2B} \quad (14)$$

With the coordinate relationship from the partial derivative of one direction, we can substitute it into the border equation of the ellipse where $F = 0$. As an example, to get the maximum coordinate

value of the ellipse in horizontal direction, , i.e., x_{max} , we need to substitute $y = -Cx/2B$ into $Ax^2 + By^2 + Cxy + D = 0$:

$$Ax^2 + B\left(-\frac{Cx}{2B}\right)^2 + Cx\left(-\frac{Cx}{2B}\right) + D = 0 \\ \left(A + \frac{C^2}{4B} - \frac{C^2}{2B}\right)x^2 + D = 0 \\ \left(A - \frac{C^2}{4B}\right)x^2 + D = 0 \\ \left(\frac{C^2}{4B} - A\right)x^2 = D \\ x^2 = \frac{4BD}{C^2 - 4AB} \\ x_{max} = \sqrt{\frac{4BD}{C^2 - 4AB}} \quad (15)$$

With elliptical coefficients substituted, the calculation procedure of x_{max} is formulated as follows:

$$x_{max} = \sqrt{\frac{-8\Sigma'_X (\Sigma'_X \Sigma'_Y - \Sigma'_{XY}^2) \ln\left(\frac{\sigma_i}{\alpha_{low}}\right)}{4\Sigma'_{XY}^2 - 4\Sigma'_X \Sigma'_Y}} \\ x_{max} = \sqrt{2\Sigma'_X \ln\left(\frac{\sigma_i}{\alpha_{low}}\right)}. \quad (16)$$

Similarly, we can get the maximum coordinate value of the ellipse in vertical direction, , i.e., y_{max} , we need to substitute $x = -Cy/2A$ into $Ax^2 + By^2 + Cxy + D = 0$:

$$y_{max} = \sqrt{2\Sigma'_Y \ln\left(\frac{\sigma_i}{\alpha_{low}}\right)}. \quad (17)$$

Compared to bounding circle with adaptive radius, the axis-aligned bounding box for Gaussian splatting can achieve different extents of culling in horizontal and vertical directions, thereby obtaining different tile ranges for the two directions. Furthermore, similar to the adaptive radius for bounding circle, we take the original radius as the upper limit:

$$r_x = \min(x_{max}, r_o), \quad r_y = \min(y_{max}, r_o). \quad (18)$$

4 DETAILED COMPARISONS WITH OTHER METHODS

Due to the fact that the existing works to accelerate 3DGS are based on lightweight approaches, i.e. reducing the overall number of Gaussian, their effective part for acceleration is often part of their whole methods, such as the pruning and distillation operation in LightGaussian [Fan et al. 2023] and frequency-modulated loss in GES [Hamdi et al. 2024]. Since other sub-methods of these works may cause degradation of quality or rendering speed, we show the comparisons between their effective part and our AdR-Gaussian to further demonstrate the power of our method. Quantitative comparisons are shown in Table 1. where 3DGS denotes the baseline [Kerbl et al. 2023], LightG-Pd denotes LightGaussian with only pruning and distillation, GES-Freq denotes GES with only the frequency-modulated loss, Ours-AABB denotes our AdR-Gaussian with only axis-aligned bounding box, and Ours-Full denotes our full method, while we still utilize the same datasets as those used in 3DGS: all

Table 1. Quantitative evaluation of our method compared to the effective sub method of latest works, computed over the same full dataset as 3DGS. The bold results represent the best performance, while the underlined results stand for the second-best among all the results obtained in our own experiments.

Dataset Method Metric	Mip-NeRF360						Tanks&Temples						Deep Blending					
	Train↓	FPS↑	Mem↓	SSIM↑	PSNR↑	LPIPS↓	Train↓	FPS↑	Mem↓	SSIM↑	PSNR↑	LPIPS↓	Train↓	FPS↑	Mem↓	SSIM↑	PSNR↑	LPIPS↓
3DGS	29m15s	185.85	780.7MB	0.813	<u>27.49</u>	0.220	14m56s	253.39	437.5MB	<u>0.844</u>	23.64	0.178	24m14s	213.00	677.5MB	0.899	29.50	0.246
LightG-Full	34m30s	212.32	58.7MB	0.803	27.03	0.238	17m49s	341.30	33.9MB	0.834	23.41	0.196	28m19s	259.15	52.0MB	0.896	29.12	0.258
LightG-Pd	33m33s	247.33	<u>201.8MB</u>	0.811	27.34	0.230	16m53s	412.92	<u>116.0MB</u>	0.839	23.53	0.191	27m22s	303.14	<u>179.5MB</u>	0.901	29.44	0.257
GES-Full	<u>21m50s</u>	241.67	332.9MB	0.792	26.95	0.258	<u>10m56s</u>	339.22	214.5MB	0.836	23.36	0.197	<u>19m04s</u>	279.93	367.5MB	0.901	29.60	0.252
GES-Freq	24m45s	235.61	522.2MB	0.802	27.35	0.241	12m50s	331.71	323.0MB	0.839	<u>23.65</u>	0.190	21m25s	270.90	523.0MB	0.900	<u>29.61</u>	0.248
Ours-AABB	27m49s	<u>336.15</u>	786.7MB	0.814	27.51	0.219	14m3s	<u>419.07</u>	445.5MB	0.846	23.76	0.178	22m51s	<u>449.21</u>	693.0MB	0.899	29.50	0.246
Ours-Full	17m44s	589.61	274.3MB	0.783	26.90	0.272	<u>9m2s</u>	718.42	192.0MB	0.832	23.53	0.205	15m32s	716.46	335.5MB	0.901	29.65	0.254

scenes from Mip-NeRF360 [Barron et al. 2022], two scenes from Tanks&Temples [Knapitsch et al. 2017], and two scenes from Deep-Blending [Hedman et al. 2018], encompassing both bounded indoor scenes and unbounded outdoor environments.

The quantitative experiments demonstrate that LightGaussian with only pruning and distillation achieves better quality and rendering speed with bigger model size than its full model, while GES with only the frequency-modulated loss achieves better quality and equivalent rendering speed than its full model. Nevertheless, ours with only axis-aligned bounding box achieves faster rendering speed than all aforementioned methods as well as optimal rendering quality on every dataset, and our full AdR-Gaussian method with early culling and load balancing achieves the best rendering efficiency.

5 QUALITATIVE RESULT FOR ABLATION

In this section, we show the qualitative result for ablation studies, *i.e.* comparisons of our sub methods (*Adaptive Radius*, *AABB*, and *Load Balancing*) and the corresponding ground truth images from held-out test views, illustrated as Fig. 1 and Fig. 2. Although the process of sequentially adding each sub method gradually accelerates rendering, the qualitative comparison for ablation shows that there is no significant degeneration of quality during this process.

REFERENCES

- Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18–24, 2022*. IEEE, 5460–5469. <https://doi.org/10.1109/CVPR52688.2022.00539>
- Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. 2023. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. *CoRR* abs/2311.17245 (2023), 1–16. [https://doi.org/10.48550/ARXIV.2311.17245 arXiv:2311.17245](https://doi.org/10.48550/ARXIV.2311.17245)
- Abdullah Hamdi, Luke Melas-Kyriazi, Guocheng Qian, Jinjie Mai, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. 2024. GES: Generalized Exponential Splatting for Efficient Radiance Field Rendering. *CoRR* abs/2402.10128 (2024), 1–39. [https://doi.org/10.48550/ARXIV.2402.10128 arXiv:2402.10128](https://doi.org/10.48550/ARXIV.2402.10128)
- Peter Hedman, Julian Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.* 37, 6 (2018), 257. <https://doi.org/10.1145/3272127.3275084>
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (2023), 139:1–139:14. <https://doi.org/10.1145/3592433>
- Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.* 36, 4 (2017), 78:1–78:13. <https://doi.org/10.1145/3072959.3073599>
- M. Zwicker, H. Pfister, J. van Baar, and M. Gross. 2002. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics* 8, 3 (2002), 223–238. <https://doi.org/10.1109/TVCG.2002.1021576>

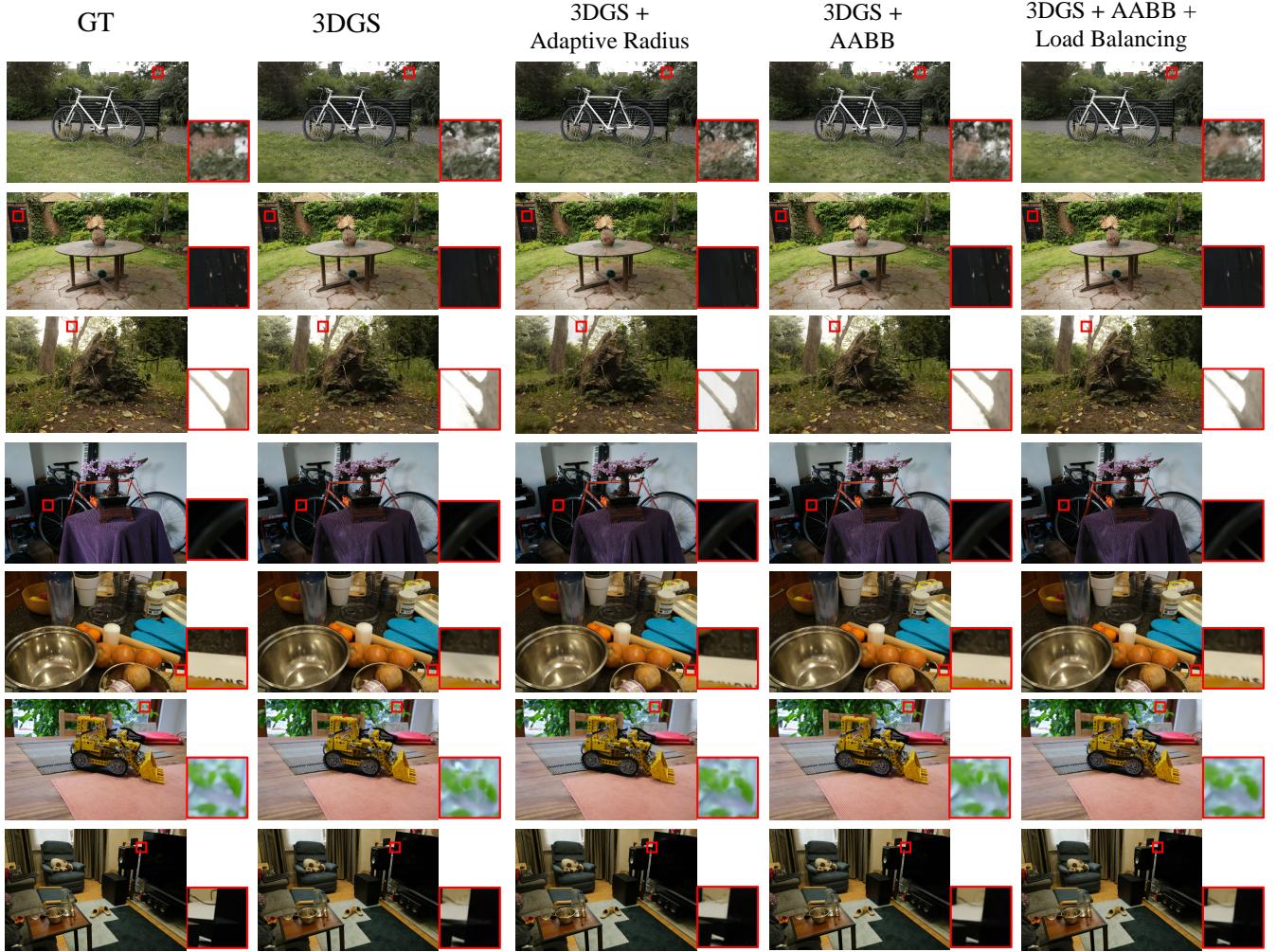


Fig. 1. Part of qualitative comparisons for ablation. Based on the 3DGS baseline, we sequentially introduce each sub-method for ablation experimentation, where the sub-methods are shown from left to right. The scenes are, from the top down: *BICYCLE*, *GARDEN*, *STUMP*, *BONSAI*, *COUNTER*, *KITCHEN*, and *ROOM* from the Mip-Nerf360 dataset. Detailed comparisons are highlighted with red rectangles, where the larger rectangle is the zoomed-in view of the smaller one.

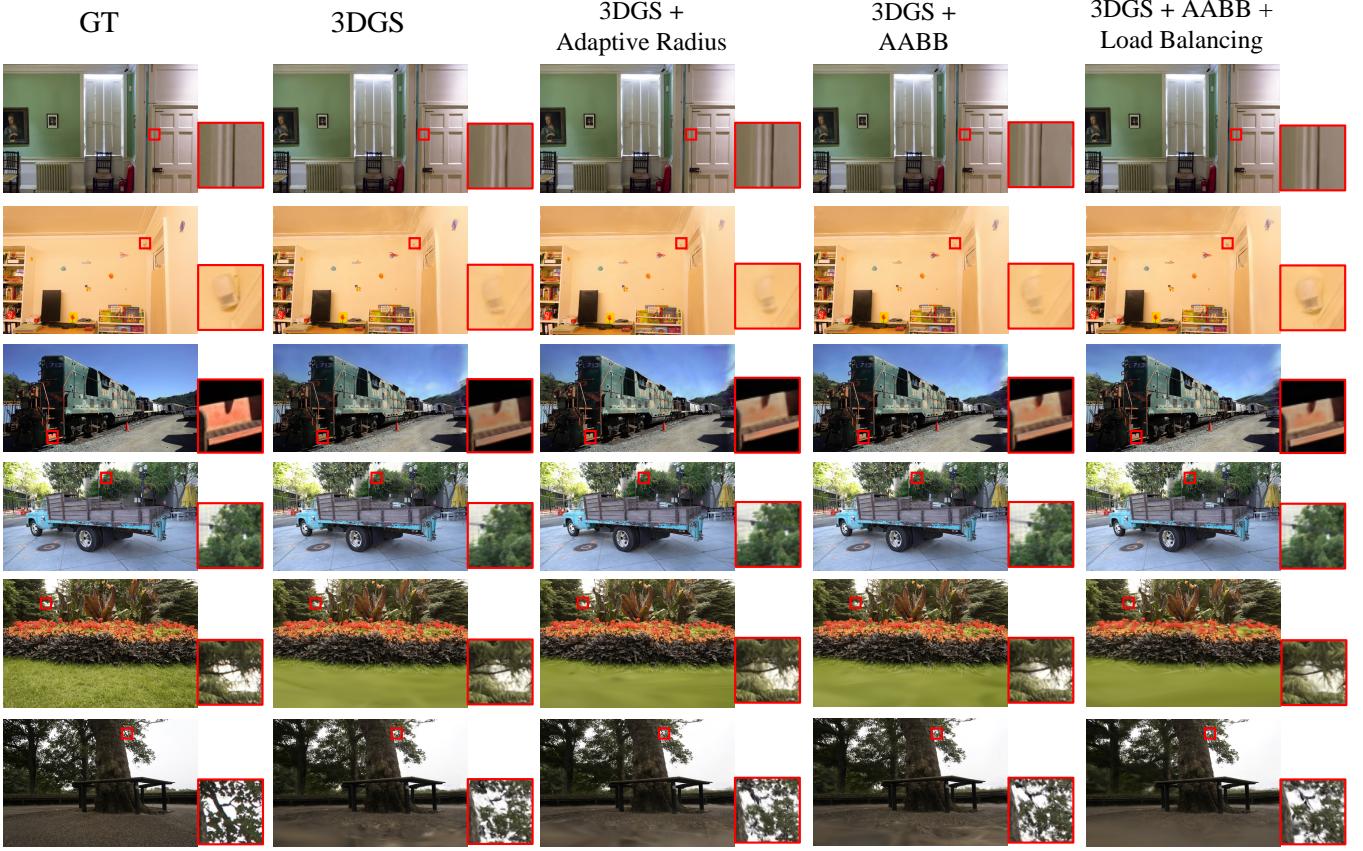


Fig. 2. Part of qualitative comparisons for ablation. Based on the 3DGS baseline, we sequentially introduce each sub-method for ablation experimentation, where the sub-methods are shown from left to right. The scenes are, from the top down: *DRJOHNSON* and *PLAYROOM* from the Deep Blending dataset; *TRAIN* and *TRUCK* from Tanks&Temples dataset; *FLOWERS*, and *TREEHILL* from the Mip-NeRF360 dataset. Detailed comparisons are highlighted with red rectangles, where the larger rectangle is the zoomed-in view of the smaller one.