Faster and Better 3D Splatting via Group Training

A. The Parameters Independence and the Opacity Regularization

A.1. Assumption of Parameters Independence

The different attributes of 3DGS primitives can be specified arbitrarily without affecting each other, by explicit regularization [1, 2]. Therefore, the attributes can be considered to be independent of each other.

A.2. Opacity Regularization

Some regularization for opacity tends to use low opacity Gaussians [2], which may bring about a potential reduction in rendering speed, which in turn leads to longer training time with lower opacity shown **Table 4** of [2].

B. Proof of Property 1: Opacity-based Effective Gaussians Densification

Under the assumptions of mutual independence between Gaussian attributes and intra-primitive parameter independence, the partial derivatives $[\frac{\partial \Delta x}{\partial x_\mathrm{m}}, \frac{\partial \Delta y}{\partial y_\mathrm{m}}]$ for any arbitrary Gaussian admit the following derivation:

$$\frac{\partial \alpha}{\partial G} = O + \frac{\partial O}{\partial G} G = O + 0 \cdot G = O, \tag{1}$$

$$\begin{aligned} \frac{\partial L}{\partial x_\mathrm{m}} &= \sum_{\text{pixel}} \frac{\partial L}{\partial G_m^{\text{2D}}} \frac{\partial G_m^{\text{2D}}}{\partial \Delta x} \frac{\partial \Delta x}{\partial x_\mathrm{m}} \\ &= \sum_{\text{pixel}} o_m \frac{\partial L}{\partial \alpha_m} \frac{\partial G_m^{\text{2D}}}{\partial \Delta x} \frac{\partial \Delta x}{\partial x_\mathrm{m}} \\ &= o_m \sum_{\text{pixel}} \frac{\partial L}{\partial \hat{C}} \frac{\partial \hat{C}}{\partial \alpha_m} \frac{\partial G_m^{\text{2D}}}{\partial \Delta x} \frac{\partial \Delta x}{\partial x_\mathrm{m}}, \end{aligned} \tag{2}$$

$$\frac{\partial L}{\partial y_\mathrm{m}} = o_m \sum_{\text{pixel}} \frac{\partial L}{\partial \hat{C}} \frac{\partial \hat{C}}{\partial \alpha_m} \frac{\partial G_m^{\text{2D}}}{\partial \Delta y} \frac{\partial \Delta y}{\partial y_\mathrm{m}}, \tag{3}$$

where $[\frac{\partial \Delta x}{\partial x_\mathrm{m}}, \frac{\partial \Delta y}{\partial y_\mathrm{m}}]$ remain constant parameters determined by the resolution $[W, H]$; $[\frac{\partial G_m^{\text{2D}}}{\partial \Delta x}, \frac{\partial G_m^{\text{2D}}}{\partial \Delta y}]$ derive from the scale, the rotation and the world coordinates of Gaussian primitives (independent of their opacity); and $\frac{\partial L}{\partial \hat{C}}$ represents the loss gradient with respect to the current pixel value.

Given that $\hat{C}$ is formulated as the composite rendering of N Gaussians in Eq. (4), the derivative $\frac{\partial \hat{C}}{\partial \alpha_m}$ admits computation via Eq. (5).

$$\begin{aligned} \hat{C} = &\underbrace{\sum_{i=1}^{m-1} \alpha_i c_i \prod_{j=1}^{i-1} (1-\alpha_j)}_{\text{Before Gaussian } m} + \underbrace{\alpha_m c_m \prod_{j=1}^{m-1} (1-\alpha_j)}_{\text{Gaussian } m} \\ &+ \underbrace{\sum_{i=m+1}^{N} \overbrace{\prod_{j=1}^{m-1} (1-\alpha_j)}^{\text{Before G } m} \alpha_i c_i (1-\alpha_m) \overbrace{\prod_{j=m+1}^{i-1} (1-\alpha_j)}^{\text{After G } m}}_{\text{After Gaussian } m} \\ &+ \underbrace{\prod_{i=1}^{N} (1-\alpha_i) c_{\text{bg}}}_{\text{background}}, \end{aligned} \tag{4}$$

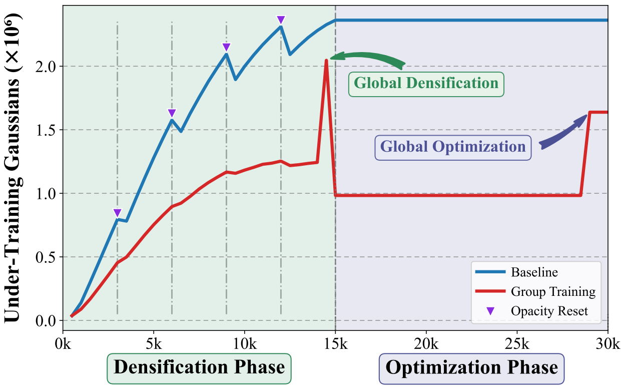

Figure 1. **Comparsion of Under-Training Gaussian Primitives.** Our Group-Training methodology selectively trains a subset of Gaussian primitives, demonstrating enhanced computational efficiency while mitigating loss of potentially critical points during opacity reset operations.

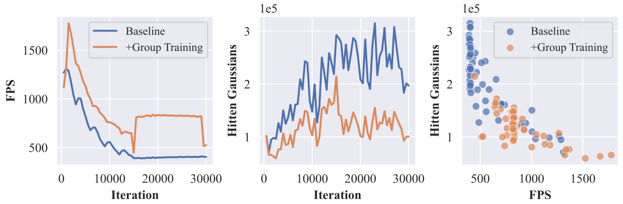

Figure 2. **Training dynamics of rendering performance and Gaussians utilization.** **Left:** Group-Training (GT) maintains higher FPS than the baseline throughout training. **Middle:** GT requires fewer hit Gaussians per rendered frame compared to the baseline. **Right:** GT achieves faster rendering (higher FPS) with reduced under-training Gaussians, demonstrating lower overall computational cost.

$$\frac{\partial \hat{C}}{\partial \alpha_m} = \overbrace{\prod_{j=1}^{m-1} (1-\alpha_j)}^{\text{Before G } m} \left[c_m - \sum_{i=m+1}^{N} \alpha_i c_i \overbrace{\prod_{j=m+1}^{i-1} (1-\alpha_j)}^{\text{After G } m} \right] - \frac{c_{\text{bg}} T_N}{1-\alpha_m} \tag{5}$$

Subsequently, the mathematical expectation of this derivative is formally established through Eq. (6)

$$\begin{aligned} \mathbb{E}\left[\frac{\partial \hat{C}}{\partial \alpha_m}\right] &= \overbrace{(1-\alpha_0)^{m-1}}^{\text{Before G } m} \left\{ c_0 - c_0\alpha_0 \sum_{i=m+1}^{N} \mathbb{E}\left[\overbrace{\prod_{j=m+1}^{i-1} (1-\alpha_j)}^{\text{After G } m} \right] \right\} - \frac{c_{\text{bg}} T_{\text{sta.}}}{1-\alpha_0} \\ &= (1-\alpha_0)^{m-1} \left[c_0 - c_0\alpha_0 \sum_{i=m+1}^{N} (1-\alpha_0)^{i-m-1} \right] - \frac{c_{\text{bg}} T_{\text{saturation}}}{1-\alpha_0} \\ &= \frac{(c_0 - c_{\text{bg}}) T_{\text{saturation}}}{1-\alpha_0} \\ &= \frac{(c_0 - c_{\text{bg}}) T_{\text{saturation}}}{1-\mathbb{E}\left[o_i\right]\mathbb{E}\left[G_i\right]}, \end{aligned} \tag{6}$$

C. Temporal Evolution of Under-Training Gaussian Primitives

We visually compare the number of under-training Gaussian primitives between Group-Training and 3DGS during scene reconstruction in Fig. 1. 3DGS with Group-Training reduces the training overhead by avoiding full optimization of all Gaussian primitives. Furthermore, during each opacity reset operation, the proposed method retains a higher proportion of geometrically significant primitives compared to baseline. These retained elements, despite their low-opacity values, preserve critical structural information that contributes to scene geometry fidelity.

D. Comparison of Scene Representation Efficiency

Fig. 2 presents a comparative analysis for the ”train” [3] scene reconstruction using 3D Gaussian Splatting (3DGS) under baseline conditions versus Group Training. It compares the per-iteration FPS and the number of hitten Gaussians. And the baseline method required 12.5 minutes to reach a PSNR of 21.985 dB, whereas Group Training with OPS acceleration attained a PSNR of 22.156 dB in just 9.3 minutes. These measurements confirm that Group Training consistently accelerates rendering with a substantial reduction in hitten Gaussians count during training. Consequently, it demonstrates higher scene representation efficiency by utilizing fewer Gaussian primitives without compromising reconstruction quality.

References

[1] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. *arXiv preprint arXiv:2311.13398*, 2023. 1

[2] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024. 1

[3] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. In *SIGGRAPH*, 2017. 2