Perception-Aware Training for 3D Gaussian Splatting under Limited Resources

Jie Peng (Frank) Chen

Abstract—3D Gaussian Splatting (3DGS) enables real-time view synthesis but remains difficult to train on resource-constrained platforms such as mobile and AR/VR devices due to its high computational and memory demands. While recent work like Taming 3DGS improves training efficiency through budgeted densification, it relies on structure-based heuristics that treat all regions equally and ignore perceptual relevance. In this work, we propose a perception-aware extension that integrates lightweight cues—such as saliency, segmentation, edge, and depth maps—to guide densification toward regions more important to human viewers. We also introduce an unseen-view depth regularization strategy to improve scene consistency. Together, these contributions help allocate limited resources more effectively and enhance reconstruction quality under constrained budgets. Our method builds on prior work while advancing toward practical, perceptually guided 3DGS training for deployment on lightweight systems.

Index Terms—3D Reconstruction, Gaussian Splatting, Novel View Synthesis

1 Introduction

3D Gaussian Splatting (3DGS) [1] has recently gained attention for its ability to generate high-quality, real-time view synthesis results. It models a scene using millions of 3D Gaussian primitives, each with attributes such as position, scale, opacity, and view-dependent appearance represented by spherical harmonics (SH). However, this high fidelity comes at a cost: 3DGS is extremely resource-intensive. A single scene can contain millions of Gaussians, and each primitive stores 48 SH coefficients for the three color channels, along with other attributes such as position and opacity, which has total 59 float-32 attributes per Gaussian. This leads to substantial memory usage and computational overhead as training progresses.

Moreover, the training process lacks any constraint on model growth. Gaussians are added adaptively during training through a densification process, but without an upper bound. On resource-constrained devices such as mobile phones or AR/VR headsets, this unrestricted growth can quickly exceed memory limits and cause training to fail.

There has been substantial research on compressing 3DGS models after training to make them lighter for deployment [2], [3]. Techniques such as pruning, quantization, and encoding have been shown to reduce memory consumption and improve rendering speed. However, these methods are designed for inference and do not address the challenge of training 3DGS models efficiently on low-resource hardware.

Recent work such as Taming 3DGS [4] takes an important step toward addressing this challenge by introducing budgeted densification and performance optimizations. Their approach guides training using structural heuristics such as gradient magnitude, pixel coverage, and opacity of each Gaussian. However, it does not consider perceptual relevance and treats all regions of the scene equally, regardless of how important they are to human viewers. In resource-constrained settings, this can lead to inefficient use of the computational budget.

In this work, we explore how perception-aware guidance can improve training efficiency under strict resource constraints. We incorporate lightweight visual cues, including saliency, segmentation, edge, and depth maps, into the scoring function used for densification. This allows the system to prioritize detail in regions that are perceptually important. Additionally, we introduce an unseen-view depth regularization loss to improve scene consistency across the entire scene. Together, these components aim to make 3DGS training more perception-aware by allocating resources in a way that aligns with human visual perception, while maintaining low computational overhead. Our method is lightweight and well suited for deployment on mobile and embedded platforms.

Our contributions are summarized as follows:

- We propose a perception-aware scoring mechanism that integrates saliency, segmentation, edge, and depth cues to guide Gaussian densification toward perceptually important regions.
- We introduce an unseen-view depth regularization loss to improve geometric consistency across the whole scene.
- We show that our method produces visually improved reconstructions in perceptually important areas, even under strict Gaussian budget constraints, while maintaining comparable quantitative performance to Taming 3DGS.

Code is available:
https://frankjc2022.github.io/perception-aware-3dgs/

2 Related Work

2.1 Novel View Synthesis

Novel View Synthesis (NVS) aims to generate photorealistic images from novel viewpoints given a set of captured views. Neural Radiance Fields (NeRFs) [5] represent scenes as continuous volumetric fields and use multilayer perceptrons (MLPs) to predict color and density. While NeRFs achieve high visual quality, they are computationally expensive to

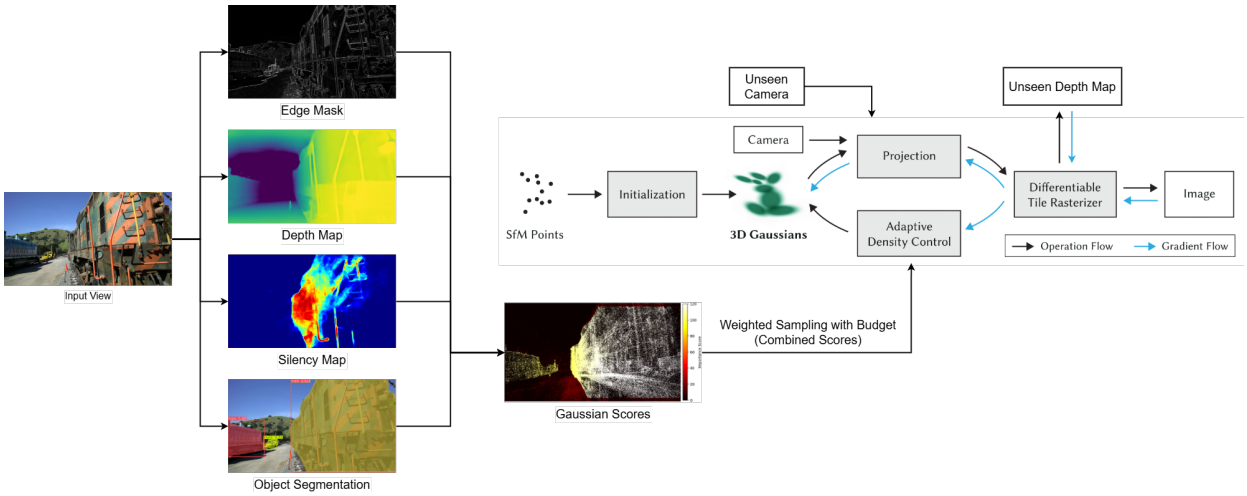

Fig. 1. Overview of the proposed training pipeline. For each input view, we precompute four lightweight perceptual cues: saliency, segmentation, edge, and depth maps. The 3D Gaussians are projected onto these maps to compute importance scores for densification. The final score is a weighted combination of perception-aware signals and the structural score from Taming 3DGS, and is used to guide weighted sampling under a pre-defined Gaussian budget. Additionally, we introduce an unseen-view depth regularization loss, computed from depth maps rendered at novel viewpoints, to improve geometric consistency across the scene.

train and render. Follow-up works such as MipNeRF [6] and Instant-NGP [7] have improved performance, but real-time training and rendering remain challenging.

3D Gaussian Splatting offers a promising alternative to NeRF by representing scenes with explicit 3D Gaussian primitives and enabling real-time rendering through fast screen-space splatting and blending. Each primitive stores attributes such as position, scale, opacity, and view-dependent color. While efficient at inference, training 3DGS is resource-intensive due to the dynamic growth of primitives and the large number of parameters per Gaussian. In complex scenes, the number of Gaussians can reach millions, which leads to high memory usage and long training times.

2.2 Model Compression for 3DGS

To enable deployment of 3DGS on limited-resource devices, many works focus on post-training compression to reduce memory and storage requirements. These approaches include pruning [8]–[10], quantization [11], [12], and entropy-based encoding [13]. Structured techniques such as hash grids and anchor-based representations [14], [15] further improve efficiency by organizing Gaussian parameters more compactly. Survey papers [2], [3] summarize these developments. However, most of these methods focus on inference and assume that training is performed on high-end hardware, leaving training-time efficiency unaddressed.

2.3 Training Efficiency for 3DGS

Taming 3DGS [4] improves training efficiency by introducing budgeted densification and a structure-based scoring function using features such as gradient magnitude, pixel coverage, and opacity. However, it does not consider perceptual relevance and treats all regions equally, regardless of visual importance. This can lead to suboptimal use of the limited Gaussian budget, especially in AR/VR settings, where it is more desirable to focus detail on regions that attract human attention.

Outside 3DGS, perception-guided strategies have been explored in image generation tasks [16], [17], where cues such as segmentation maps or detection priors are used to improve generation quality or perceptual alignment. However, such signals have not been applied to guide the densification process in 3DGS training.

Recent methods such as SparseGS [18] and DNGaussian [19] use depth from input views to improve geometry reconstruction under sparse-view conditions. We explore a lightweight alternative that applies a simple regularization loss on depth maps rendered from unseen viewpoints to encourage smoother geometry.

To explore this direction, we propose a lightweight perception-aware scoring mechanism that integrates saliency, segmentation, edge, and depth cues to better allocate Gaussians under a fixed budget. We also include a depth-based regularization loss computed from unseen viewpoints to further improve geometric consistency. Details are described in Section 3.

3 Proposed Method

Our goal is to improve the training efficiency of 3DGS in resource-constrained settings by guiding the training process to focus on perceptually important regions, while avoiding over-allocation of the Gaussian budget to low-priority areas. In addition, we introduce a lightweight depth regularization loss based on unseen views to improve geometric consistency. Our method builds on Taming 3DGS [4], which introduced structural importance scores, pre-defined training budgets, and fused SSIM loss to accelerate training and control model growth. Figure 1 provides an overview of our training pipeline, which integrates both perception-aware scoring and unseen-view regularization.

TABLE 1
Comparison across three datasets. Metrics are averaged over scenes. We compare Taming 3DGS with our variants: using only perceptual scores, combining perceptual and structural scores, adding unseen-view loss, and applying both.

	Tanks&Temples					MipNeRF-360					Deep Blending				
	SSIM↑	PSNR↑	LPIPS↓	Train	#Gauss.	SSIM↑	PSNR↑	LPIPS↓	Train	#Gauss.	SSIM↑	PSNR↑	LPIPS↓	Train	#Gauss.
Taming 3DGS	0.834	23.91	0.212	2.9m	318,524	0.789	27.21	0.267	5.1m	633,583	0.899	29.74	0.277	3.1m	294,383
Perceptive Score Only	0.828	23.61	0.219	3.1m	318,524	0.782	27.03	0.278	5.4m	633,583	0.897	29.49	0.282	3.2m	294,383
Combined Scores	0.837	23.95	0.210	3.1m	318,524	0.790	27.22	0.266	5.4m	633,583	0.900	29.88	0.277	3.2m	294,383
Unseen View Loss	0.835	23.86	0.211	4.3m	318,524	0.789	27.20	0.268	6.7m	633,583	0.897	29.58	0.277	4.7m	294,383
Combined Scores+Loss	0.835	23.89	0.211	4.3m	318,524	0.789	27.21	0.268	6.7m	633,583	0.899	29.76	0.278	4.9m	294,383

3.1 Perception-Aware Densification

We extend the densification method introduced in Taming 3DGS [4], which uses a budgeted control mechanism to periodically densify Gaussians through weighted sampling. Each Gaussian is assigned an importance score based on structural heuristics, including screen-space gradient magnitude (computed via Laplacian-filtered photometric loss), pixel coverage, opacity, and view-dependent features such as scale and sharpness. At each densification step, a subset of Gaussians is selected based on these scores, constrained by a user-defined budget, and duplicated through split and clone operations. This ensures that model growth remains bounded and predictable throughout training.

While effective, this scoring strategy is entirely structure-driven and does not account for human perceptual relevance. For example, high-gradient regions such as grass or tree leaves in the background may receive high scores despite having limited impact on perceived visual quality. To address this, we incorporate lightweight perception-aware cues into the scoring function.

For each training view, we generate four perceptual maps using pretrained models: saliency (U2Net [20]), object segmentation (YOLOv11 [21]), edge (Sobel filter), and monocular depth (ZoeDepth [22]). These models were selected for their efficiency, as each can be computed quickly and with minimal overhead, making them suitable for real-time or resource-constrained settings. Each Gaussian is projected into the image space of all views, and the corresponding values from the four maps are computed at each projected location. These values are then averaged across views and combined using a weighted sum to form a perception-aware score:

$$\text{PerceptionScore}_g = w_d \cdot D_g + w_s \cdot S_g + w_e \cdot E_g + w_c \cdot C_g \quad (1)$$

where D_g, S_g, E_g, and C_g denote the per-Gaussian depth, saliency, edge, and segmentation scores, respectively. The weights are set empirically as $w_d = 50$, $w_s = 100$, $w_e = 25$, and $w_c = 50$, reflecting the relative importance of each cue, with saliency receiving the highest weight due to its strong alignment with human attention.

The final importance score for each Gaussian is computed by combining the structural and perceptual scores:

$$\text{Score}_g = \alpha \cdot \text{StructureScore}_g + \beta \cdot \text{PerceptionScore}_g \quad (2)$$

where we set $\alpha = 0.7$ and $\beta = 0.3$ in all experiments. The final scores guide densification under a fixed Gaussian budget, so the model focuses on regions that are both structurally and perceptually important.

3.2 Unseen-View Depth Regularization

To further improve the quality of novel view synthesis, we introduce a lightweight depth-based regularization loss using unseen viewpoints. While prior work typically applies smoothness constraints to depth maps from input views, we explore the use of depth information from novel, unseen viewpoints as an additional consistency signal.

We predefine a set of unseen cameras distributed around the scene using the initial point cloud and input camera poses. During training, we randomly select one unseen camera $\mathcal{C}_u$ and render its RGB image I_u and depth map D_u. We then compute a depth smoothness loss $\mathcal{L}_{\text{smooth}}(D_u)$ to reduce local depth variations, and a total variation loss $\mathcal{L}_{\text{tv}}(I_u)$ to make the rendered image more spatially consistent.

The total unseen-view regularization loss is defined as:

$$\mathcal{L}_{\text{unseen}} = \lambda_s \cdot \mathcal{L}_{\text{smooth}}(D_u) + \lambda_{\text{tv}} \cdot \mathcal{L}_{\text{tv}}(I_u) \quad (3)$$

We set $\lambda_s = 0.7$, $\lambda_{\text{tv}} = 0.3$, and apply this loss with a small weight $\lambda_{\text{unseen}} = 0.2$, alongside the standard photometric and SSIM losses used in 3DGS training.

4 Experimental Results

4.1 Experimental Setup

We evaluate our method on three commonly used datasets for view synthesis and neural rendering: MipNeRF-360 [6], Tanks and Temples [23], and Deep Blending [24], which contain 9, 2, and 2 scenes, respectively. These datasets include both bounded indoor and unbounded outdoor environments with diverse geometry and background complexity. We follow the same train/test splits and evaluation setup as used in the original 3DGS and Taming 3DGS works.

Our model and Taming 3DGS are trained for 30,000 iterations using the same Gaussian budget settings, while the original 3DGS is trained for 30,000 iterations without a budget constraint. We also adopt the fully fused differentiable SSIM [4] from Taming 3DGS, an efficient replacement for standard SSIM that improves training speed.

We report standard view synthesis metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). We also assess resource usage by reporting the total training time and the final number of Gaussians. All experiments are run on a single NVIDIA RTX 4080 GPU.

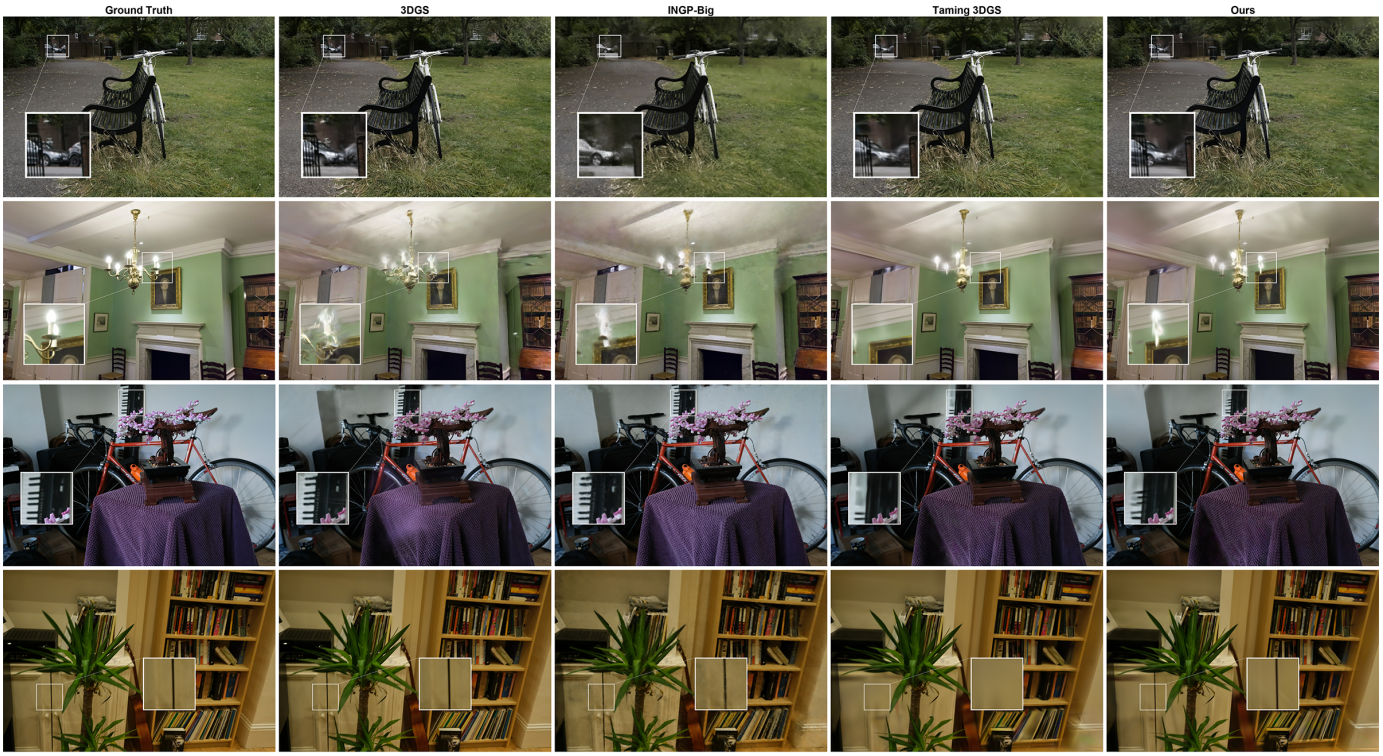

Fig. 2. Qualitative comparison across multiple methods. From left to right: Ground Truth, original 3DGS (30k iterations), Instant-NGP (INGP-Big), Taming 3DGS (with budget), and Ours (with budget). Our method better preserves perceptually important details, such as salient regions and object surfaces, compared to Taming 3DGS under the same Gaussian budget. It also remains visually competitive with the original 3DGS and Instant-NGP-Big, despite using significantly fewer Gaussians.

4.2 Quantitative Results

Table 1 shows the performance of our method compared to Taming 3DGS across three datasets. Our method shows marginal improvements and indicates that perceptual-aware guidance improves visual quality without sacrificing reconstruction accuracy.

When using only our perceptual score (*Perceptive Score Only*), performance slightly drops, especially in PSNR, which measures pixel-level accuracy compared to ground truth. This is expected, since our score focuses more on regions that are important to human perception rather than evenly matching all pixels across the image. When combining our perceptual score with the original Taming 3DGS structural score (*Combined Scores*), the metrics slightly improve, which shows that the two types of information are complementary.

Adding the unseen-view regularization loss (*Unseen View Loss*) leads to a slight drop in metrics. This likely results from the randomly sampled unseen views differing from the train/test camera distribution, which shifts the model's focus toward global scene consistency instead of fine-tuned local reconstruction. This regularization still helps improve visual consistency when rendering from novel viewpoints.

All methods are trained using the same Gaussian budget settings as in Taming 3DGS. We follow their scene-specific budget allocation, based on spatial extent and the number of SfM points. Specifically, we use 2× the SfM points for small-scale indoor scenes in MipNeRF-360 and for Tanks and Temples (due to its high SfM point count), 5× for the larger indoor scenes in Deep Blending, and 15× for unbounded outdoor scenes. This setting ensures a fair comparison under constrained resource conditions, unlike the original 3DGS, which allows unbounded Gaussian growth.

As expected, our additional components increase training time slightly. However, the overall runtime remains reasonable and practical for low-resource training environments.

4.3 Qualitative Results

Figure 2 shows rendered views from several test scenes. Our method better reconstructs regions that are important to human perception, such as salient areas, where Taming 3DGS fails to recover fine detail. In other regions, our results remain visually comparable to Taming 3DGS without any noticeable loss in quality.

The visual comparisons show that our method recovers finer details in regions that are marked as perceptually important. As illustrated in Figure 2, the first row shows a car in the background that appears blurry in Taming 3DGS, but is more clearly reconstructed in our result because it is detected in the object segmentation map. In the second row, we recover the structure of several ceiling lights that are missed entirely by Taming 3DGS. These areas are highlighted in the saliency map and receive additional Gaussians based on our scoring function.

Overall, Figure 2 demonstrates that our method can recover more detail in regions likely to be noticed by viewers, particularly when perceptual maps identify them as visually important. While quantitative metrics such as PSNR and SSIM show only minor improvements, the qualitative

results reveal better reconstruction in perceptually sensitive areas. Our method also produces results that are visually competitive with the original 3DGS (trained without a budget constraint) and Instant-NGP, while using significantly fewer Gaussians. These observations highlight the benefit of incorporating perception-aware guidance under tight resource constraints.

4.4 Ablation Study

As shown in Table 1 and discussed in Section 4.2, we evaluate each proposed component individually against the Taming 3DGS baseline to understand its contribution. Specifically, we ablate the perception-aware scoring, the unseen-view depth regularization, and their combination. These variants help determine whether each component improves reconstruction quality or perceptual alignment.

We also tested other settings from Taming 3DGS, including a different densification interval and an alternative opacity activation function, but found they had little effect when used with our method. For consistency, we keep the same hyperparameters reported in their paper.

Fig. 3. Perception-aware scoring helps recover overlooked regions. The ceiling light is missing in Taming 3DGS but partially reconstructed in our result, guided by high values in the saliency and depth maps that identify it as perceptually important.

Figure 3 shows a qualitative example where our method recovers part of a ceiling light that is entirely missed by Taming 3DGS. This structure appears in only a few training views, so Taming's structural cues fail to assign it enough importance. Our perception-aware scoring identifies the region as visually relevant and allocates more Gaussians to it. Although the reconstruction remains blurry due to limited input views, our method is able to capture it to some extent.

Figure 4 shows the complementary effect of combining lightweight pretrained models. In this case, both the saliency and depth maps miss a car in the background, likely due to occlusion or low contrast, but the segmentation mask detects it. Since our scoring function fuses all cues, the region receives a higher score, allowing our method to allocate Gaussians and reconstruct the shape more accurately than Taming 3DGS. This example shows that combining multiple weak cues can improve robustness, even when some fail individually.

Figure 5 compares the depth maps from the final models. Taming 3DGS shows several semi-transparent Gaussians near the camera, which cause bright floaters in the depth output. Our unseen-view depth regularization reduces these artifacts. The loss penalizes inconsistent or noisy depth in

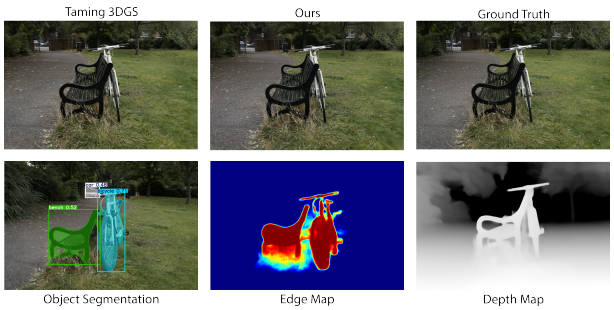

Fig. 4. Complementary cues from multiple lightweight models improve robustness. In this example, the saliency and depth maps fail to capture the car in the background, but the segmentation mask detects it correctly. This allows our method to reconstruct the shape more effectively than Taming 3DGS. Combining multiple cues helps reduce the impact of errors from individual models.

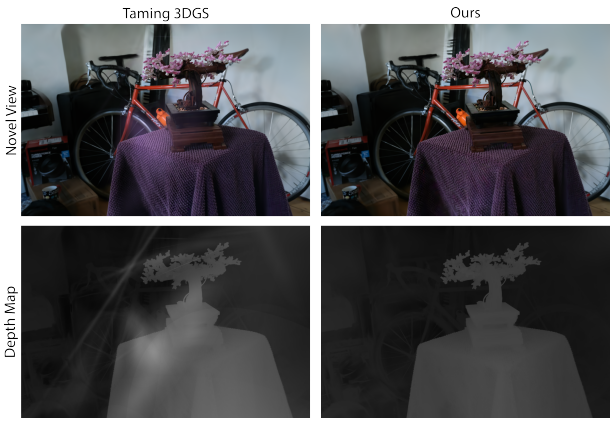

Fig. 5. Unseen-view depth regularization reduces floating artifacts. In Taming 3DGS, semi-transparent Gaussians near the front produce noisy depth with visible floaters (appearing as bright patches). Our method reduces these artifacts by using unseen-view supervision to produce more stable geometry.

unseen views and helps produce cleaner geometry with fewer floaters.

For unseen-view sampling, our initial strategy used random camera positions and distances. Many of these fell far outside the scene bounds and degraded quality. We now constrain sampling to a radius defined by the minimum and maximum extents of the original camera poses and aim the views toward the scene center. We also found that a few hundred sampled viewpoints were not enough. Performance improved when using 5,000 unseen views, so we fixed this number for all experiments without further tuning.

While our approach adds minimal training overhead under constrained Gaussian budgets, training time increases significantly when the budget is unbounded. This is especially noticeable in high-resolution scenes. For example, in the *train* scene (final Gaussian count $\approx$ 1M), training time remains manageable. In contrast, the *bicycle* scene grows to over 6M Gaussians without a budget limit, which causes a dramatic slowdown. This suggests that the current implementation does not scale well with unbounded model growth. Optimizing key components, such as perceptual scoring, through CUDA or other hardware acceleration could improve runtime efficiency.

5 Conclusion and Discussion

We explored strategies to improve 3D Gaussian Splatting training under resource-constrained settings by introducing perception-aware scoring and unseen-view depth regularization. While quantitative improvements in metrics such as PSNR, SSIM, and LPIPS are modest, our method shows qualitative gains in reconstructing perceptually important regions and in producing more consistent depth, particularly near salient areas.

Our findings suggest that perceptual cues can guide resource allocation during densification, even under strict Gaussian budgets. Although our goal was not to outperform existing methods across all metrics, we aimed to examine whether perceptual and spatial priors can steer training more effectively in low-resource conditions.

We also identified limitations in our current implementation. The method becomes less practical when Gaussian counts grow significantly, such as in unbounded training scenarios. Optimizing key components for runtime efficiency, for example through CUDA or hardware acceleration, remains an important direction for future work.

Finally, our experiments with unseen-view supervision show potential for improving consistency when rendering from novel viewpoints. Further refinement of the sampling strategy and loss formulation may enhance this effect. We hope this work encourages continued research on perception-aware and resource-efficient training techniques for 3D Gaussian Splatting.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[2] M. T. Bagdasarian, P. Knoll, Y.-H. Li, F. Barthel, A. Hilsmann, P. Eisert, and W. Morgenstern, "3dgs.zip: A survey on 3d gaussian splatting compression methods," *arXiv preprint arXiv:2407.09510*, 2024.

[3] M. S. Ali, C. Zhang, M. Cagnazzo, G. Valenzise, E. Tartaglione, and S.-H. Bae, "Compression in 3d gaussian splatting: A survey of methods, trends, and future directions," 2025. [Online]. Available: https://arxiv.org/abs/2502.19457

[4] S. S. Mallick, R. Goel, B. Kerbl, M. Steinberger, F. V. Carrasco, and F. De La Torre, "Taming 3dgs: High-quality radiance fields with limited resources," in *SIGGRAPH Asia 2024 Conference Papers*, ser. SA '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3680528.3687694

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[6] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022.

[7] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[8] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani, and H. Pirsiavash, "Compgs: Smaller and faster gaussian splatting with vector quantization," *ECCV*, 2024.

[9] P. Papantonakis, G. Kopanas, B. Kerbl, A. Lanvin, and G. Drettakis, "Reducing the memory footprint of 3d gaussian splatting," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 7, no. 1, May 2024. [Online]. Available: https://repo-sam.inria.fr/fungraph/reduced_3dgs/

[10] M. S. Ali, M. Qamar, S.-H. Bae, and E. Tartaglione, "Trimming the fat: Efficient compression of 3d gaussian splats through pruning," 2024. [Online]. Available: https://arxiv.org/abs/2406.18214

[11] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang, "Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps," 2023.

[12] S. Girish, K. Gupta, and A. Shrivastava, "Eagles: Efficient accelerated 3d gaussians with lightweight encodings," 2024. [Online]. Available: https://arxiv.org/abs/2312.04564

[13] W. Morgenstern, F. Barthel, A. Hilsmann, and P. Eisert, "Compact 3d scene representation via self-organizing gaussian grids," 2024. [Online]. Available: https://arxiv.org/abs/2312.13299

[14] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.

[15] Y. Chen, Q. Wu, W. Lin, M. Harandi, and J. Cai, "Hac: Hash-grid assisted context for 3d gaussian splatting compression," in *European Conference on Computer Vision*, 2024.

[16] Y. Wang, R. Gao, K. Chen, K. Zhou, Y. Cai, L. Hong, Z. Li, L. Jiang, D.-Y. Yeung, Q. Xu, and K. Zhang, "Detdiffusion: Synergizing generative and perceptive models for enhanced data generation and perception," 2024. [Online]. Available: https://arxiv.org/abs/2403.13304

[17] R. Wang, X. Hou, S. Schmedding, and M. F. Huber, "Stay diffusion: Styled layout diffusion model for diverse layout-to-image generation," 2025. [Online]. Available: https://arxiv.org/abs/2503.12213

[18] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, "Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting," *Arxiv*, 2023.

[19] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization," *arXiv preprint arXiv:2403.06912*, 2024.

[20] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," vol. 106, 2020, p. 107404.

[21] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[22] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[23] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[24] P. Hedman, T. Ritschel, G. Drettakis, and G. Brostow, "Scalable inside-out image-based rendering," *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, vol. 35, no. 6, December 2016. [Online]. Available: http://www-sop.inria.fr/reves/publis/2016/HRDB16