GaussianVAE: Adaptive Learning Dynamics of 3D Gaussians for High-Fidelity Super-Resolution

Shuja Khalid
Huawei Canada
shuja.khalid@huawei.com

Mohamed Ibrahim
Huawei Canada
mohamed.ibrahim9@huawei.com

Yang Liu
Huawei Canada
yang.liu8@huawei.com

Abstract

We present a novel approach for enhancing the resolution and geometric fidelity of 3D Gaussian Splatting (3DGS) beyond native training resolution. Current 3DGS methods are fundamentally limited by their input resolution, producing reconstructions that cannot extrapolate finer details than are present in the training views. Our work breaks this limitation through a lightweight generative model that predicts and refines additional 3D Gaussians where needed most. The key innovation is our Hessian-assisted sampling strategy, which intelligently identifies regions that are likely to benefit from densification, ensuring computational efficiency. Unlike computationally intensive GANs or diffusion approaches, our method operates in real-time (~0.015s per inference on a single consumer-grade GPU), making it practical for interactive applications. Comprehensive experiments demonstrate significant improvements in both geometric accuracy and rendering quality compared to state-of-the-art methods, establishing a new paradigm for resolution-free 3D scene enhancement.

Keywords: 3D Gaussian Splatting, Generative Modeling, Resolution-Free Refinement, Hessian-Based Sampling, Real-Time Inference

1. Introduction

The advent of 3D Gaussian Splatting (3DGS) [4] has revolutionized the synthesis of novel views in real time by offering an explicit and differentiable representation that combines the benefits of point-based rendering with neural scene optimization. However, despite its impressive capabilities, 3DGS inherits a fundamental limitation from its image-based training paradigm: the reconstruction quality is strictly bounded by the resolution of the input images, the VRAM footprint of which scales quadratically (Supp. 7). This constraint manifests itself in two critical ways: Once trained on lower-resolution data, missing high-frequency details cannot be recovered through traditional

optimization, and existing methods provide no mechanism to enhance Gaussians beyond their trained resolution without access to higher-quality source imagery.

Current solutions [2, 3, 8] either require retraining with higher-resolution inputs as a signal (using Single-Image Super Resolution - SISR [12]) or apply 2D super-resolution as a post-process [11], which often produces blurry results or geometric inconsistencies. We instead propose to directly refine the 3D Gaussian representation itself, enabling true resolution-free enhancement. The core innovations of our work include:

- The first resolution-free 3DGS refinement method that enhances Gaussian representations beyond their trained resolution without requiring higher-resolution input images
- A novel Hessian-guided importance sampling strategy that focuses computational resources on regions with the highest potential for quality improvement
- A highly efficient generative architecture capable of predicting new Gaussians in 15ms - two orders of magnitude faster than conventional generative approaches while using a fraction of the compute resources

Our experiments demonstrate that the proposed method successfully recovers details that would otherwise require higher-resolution training data. The approach maintains full compatibility with existing 3DGS pipelines and introduces negligible overhead (<1% runtime impact) during rendering. This work opens new possibilities for enhancing legacy captures (as a post-processing step), and adaptive level-of-detail systems.

2. Related works

2.1. 3D Gaussian Splatting Enhancements

Recent advances in 3D Gaussian splatting (3DGS) have generated multiple enhancement approaches. SuperGS [11] pioneered super-resolution for 3DGS through joint optimization with a 2D SR-CNN, achieving improvement, but requiring paired multi-resolution training data. SRGS [2] later proposed a diffusion-based up-sampler that gener-
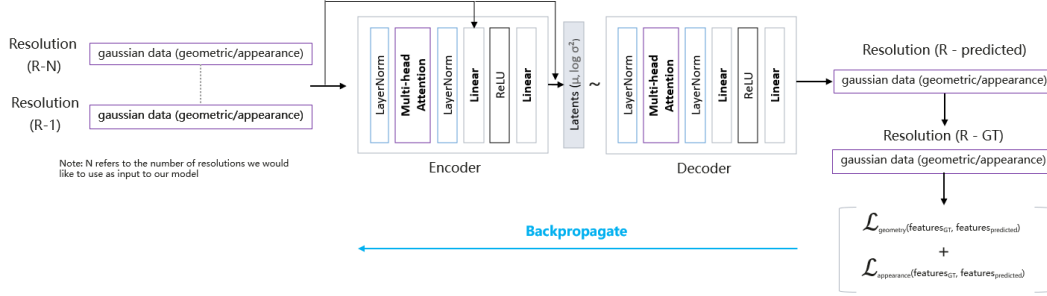
Figure 1. **Model architecture:** A transformer-based VAE with separate streams for geometry and appearance. Self-attention heads process Gaussian lineages, while the Hessian-weighted loss prioritizes regions with high densification.
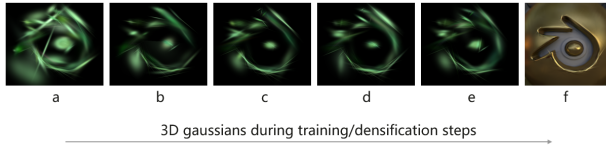

Figure 2. **The value of lineage tracking:** Selected Gaussians visualized across training steps. (a) Without lineage awareness, Gaussians may inconsistently appear/disappear across resolutions. (b-e) Our approach enforces continuity, leading to stable extrapolation and fewer artifacts when reconstructing (f).

ates new Gaussians conditioned on low-resolution renders, demonstrating superior quality but suffering generation times that are impractical for real-time applications.

2.2. Neural Radiance Field Super-Resolution

The broader field of neural representation enhancement offers relevant insights. NeRF-SR [9] introduced frequency-aware up-sampling through learned positional encoding adjustments, while Instant-NGP [6] demonstrated real-time enhancement via hash-grid distillation. However, these implicit methods struggle with explicit control over scene elements. Point-based approaches present alternative strategies: PointSR [5] used a diffusion model for point cloud upsampling, and PU-GCN [7] incorporated geometric priors. Although effective for static point clouds, these methods cannot handle view-dependent effects critical for 3DGS rendering.

3. Method

Our approach consists of a generative Variational Autoencoder (VAE) with a transformer-based architecture designed to process and extrapolate the dynamic properties of 3D Gaussians across varying resolutions. Below, we detail the core components of our method, including pre-processing, model architecture, and training strategy.

3.1. Pre-processing and Gaussian Lineage Tracking

The input to our system is a sequence of 3D Gaussian Splatting (3DGS) reconstructions at different resolutions. During pre-processing, we extract the lineage of each Gaussian, tracking its parametric evolution across densification steps which will be used to extrapolate its properties to unseen resolutions. This process, illustrated in Figure 7, is similar to meta-learning, as the model learns to predict how Gaussians evolve under resolution changes rather than relying solely on direct observations. By treating Gaussian lineages as temporal sequences, we enable the model to infer structural and appearance continuities that are not explicitly present in any single resolution. We leave the pre-processing steps to the supplementary section (8, 9).

3.2. Model Architecture

Our model is a Hessian-aware VAE with a transformer-based encoder and a symmetric decoder (Figure 1). The encoder processes Gaussian lineages as temporal data, employing a self-attention mechanism to capture long-range dependencies. This is critical because the importance of a Gaussian at one resolution may only become evident when observing its behavior across multiple scales. The decoder mirrors the encoder but reconstructs both geometry (position, scale, rotation) and appearance (color, opacity) features separately. This disentanglement ensures that geometric stability does not interfere with appearance refinement and vice versa.

3.3. Training Strategy

Similar to language models [1], we train our system by masking the final nodes in multi-step Gaussian lineages and tasking the model with predicting the missing states. This forces the model to learn robust representations of Gaussian dynamics rather than memorizing fixed-resolution snapshots. We adopt a coarse-to-fine training regimen, where the model first learns low-resolution Gaussian distributions before progressively refining them. This hierarchical ap-
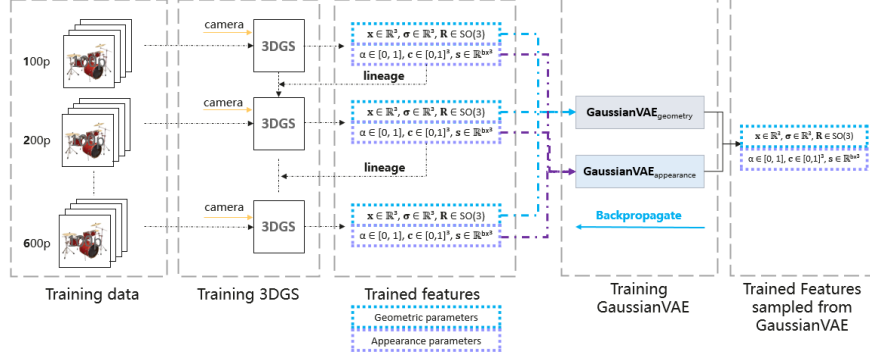
Figure 3. **Pipeline:** Input Gaussians are tracked across resolution steps, their geometric properties (x: position, σ: scale, R: rotation) and their appearance properties (c: colour, s: spherical harmonics, α: opacity) are logged, and their lineages are extrapolated for missing resolutions via learned interpolation. (C) The resulting Gaussians are extrapolated from the lineage.

proach improves convergence and stabilizes gradient flow. Our Hessian-inspired objective, which identifies and prioritizes Gaussians in regions undergoing densification (Figure 2). By analyzing where the model allocates more Gaussians, we implicitly learn which areas are perceptually or structurally significant, effectively replicating the adaptive densification behavior of traditional 3DGS pipelines, but this time, without a training signal. Our approach enforces continuity, leading to stable extrapolation and fewer artifacts when reconstructing. The model is trained in an end-to-end manner.

3.4. Loss Functions

Our training objective combines multiple loss terms to ensure geometric consistency, appearance fidelity, and latent space regularization. The total loss $\mathcal{L}_{\text{total}}$ is a weighted sum of the following components:

$$\mathcal{L}_{\text{total}} = \underbrace{\lambda_{\text{KL}}\mathcal{L}_{\text{KL}}}_{\text{Latent regularization}} + \underbrace{\lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}}}_{\text{Appearance matching}} + \underbrace{\lambda_{\text{chamfer}}\mathcal{L}_{\text{chamfer}}}_{\text{Geometric alignment}} \quad (1)$$

The weighting factors $\lambda_{\{\text{KL,MSE,chamfer}\}}$ balance the contribution of each term during training. We select 1e-6, 1.0 and 0.01 as our respective weighting factors. The KL divergence weighting factor is set dynamically using a cyclic annealing approach that cycles between fine-structural quality and generalizability.

3.5. Integration with 3D Gaussian Splatting

As depicted in Figure 3, our method acts as a post-processing step for conventional 3DGS models. Instead of modifying the core splatting pipeline, we refine its output by predicting Gaussian properties at untrained resolutions, effectively *inpainting* missing information while preserving the efficiency of the original pipeline. This positions our work as a plug-in enhancement for existing systems.
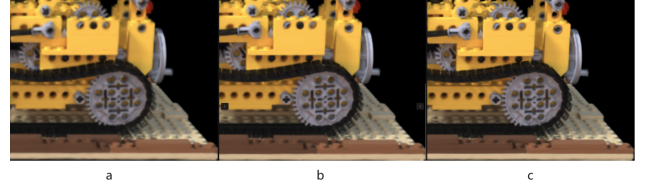

Figure 4. **Results:** NeRF Synthetic/Blender dataset - Lego scene. **a**) Image generated at 600p, **b**) *Ours-∇^2+2*, **c**) Image generated at 800p

Figure 5. **Results:** NeRF Synthetic/Blender dataset - Chair scene. **a**) Image generated at 600p, **b**) *Ours-∇^2+2*, **c**) Image generated at 800p

Figure 6. **Results:** NeRF Synthetic/Blender dataset - Ficus scene. **a**) Image generated at 600p, **b**) *Ours-∇^2+2*, **c**) Image generated at 800p

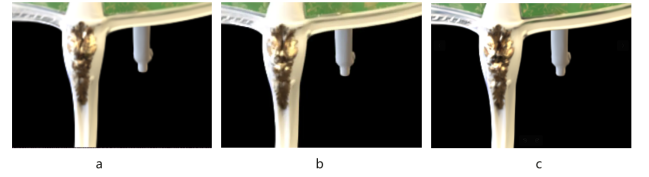### 4. Experiments

Our experiments demonstrate significant improvements over the 3DGS baseline in all evaluation metrics and scenes.

Table 1. Quantitative comparison of reconstruction quality across resolutions and methods using Chamfer distance. Lower values indicate better performance. The rightmost column shows percentage improvement of our best result over the 600p 3DGS baseline. +1, +2, +3, +4 are each extrapolations/inferences from our trained model, which only sees images with a max resolution of 600p. Top three values are highlighted using a green heatmap: dark green for the best, medium green for second-best, and light green for third-best.

Scene	3DGS Baseline			Ours				Ours-∇^2				% ↑
	600p	700p	800p	+1	+2	+3	+4	+1	+2	+3	+4	
Materials	0.001290	0.001388	0.001390	0.001382	0.001389	0.001327	0.001326	0.001247	0.001255	0.001246	0.001203	7
Lego	0.149601	0.149734	0.149882	0.149735	0.149883	0.149876	0.149880	0.149434	0.149674	0.149579	0.149509	1
Mic	0.001159	0.001101	0.001164	0.001100	0.001163	0.001117	0.001110	0.001013	0.001002	0.001025	0.001006	14
Chair	0.004691	0.004695	0.004773	0.004696	0.004774	0.004664	0.004669	0.004644	0.004643	0.004660	0.004652	1
Ficus	0.000663	0.000639	0.000639	0.000639	0.000638	0.000671	0.000658	0.000582	0.000566	0.000605	0.000568	15
Drums	0.001393	0.001370	0.001380	0.001370	0.001380	0.001362	0.001362	0.001332	0.001326	0.001333	0.001322	5
Hotdog	0.004187	0.004021	0.004068	0.004314	0.004020	0.004067	0.004066	0.003914	0.003914	0.003928	0.004008	6.5
Ship	0.002042	0.002026	0.002009	0.002025	0.002008	0.002007	0.002003	0.001957	0.001927	0.001870	0.001903	8

Table 2. Quantitative comparison using CenseoQoE metric (higher is better) across different resolutions and methods. The rightmost column shows percentage improvement of our best result over the 600p 3DGS baseline. Top three values are highlighted using a green heatmap: dark green for the best, medium green for second-best, and light green for third-best.

Scene	3DGS Baseline			Ours				Ours-∇^2				% ↑
	600p	700p	800p	+1	+2	+3	+4	+1	+2	+3	+4	
Materials	0.741	0.745	0.749	0.749	0.740	0.740	0.747	0.782	0.788	0.795	0.790	7
Lego	0.546	0.561	0.559	0.559	0.554	0.555	0.550	0.588	0.592	0.584	0.579	8
Mic	0.711	0.707	0.719	0.719	0.722	0.725	0.728	0.710	0.721	0.729	0.712	3
Chair	0.612	0.607	0.603	0.602	0.603	0.603	0.597	0.639	0.639	0.627	0.628	4
Ficus	0.696	0.691	0.689	0.689	0.688	0.687	0.686	0.704	0.702	0.705	0.701	1
Drums	0.615	0.616	0.616	0.614	0.616	0.618	0.622	0.673	0.674	0.682	0.672	11
Hotdog	0.790	0.790	0.789	0.789	0.789	0.790	0.792	0.795	0.809	0.809	0.813	3
Ship	0.701	0.692	0.699	0.699	0.695	0.699	0.700	0.708	0.689	0.731	0.728	4

We note that PSNR and SSIM rely on comparing outputs to ground-truth images, which are unavailable in our case due to extrapolation into unseen dimensions. Tables 1 and 2 report reconstruction and perceptual quality metrics across multiple resolutions, with training stopped at 600p and evaluation extrapolated to higher resolutions (+1 to +4 steps, each of which are sequential inferences from our trained model). We compare our base method (*Ours*), which extrapolates Gaussians across the entire scene to the more targeted (*Ours-∇^2*), our Hessian-optimized variant, which consistently yields better geometric fidelity and perceptual quality. achieves up to 15% lower reconstruction error on complex scenes like Ficus and 14% improvement on specular objects like Mic compared to the 600p 3DGS baseline, while maintaining superior perceptual quality (11% higher CenseoQoE scores for Drums). The resolution-agnostic nature of our approach proves particularly effective, as evidenced by consistent performance when extrapolating beyond the training resolution (+1 to +4 steps beyond 800p), whereas traditional 3DGS exhibits degradation during upsampling. Visual comparisons in Figures 4-6 corroborate these findings, showing enhanced preservation of thin structures (chair legs), high-frequency textures (ficus leaves), and specular highlights (Lego reflections). It is worth noting that we use the NeRF synthetic dataset because the dataset has a ground truth mesh we can use to compare against to isolate the geometric contribution of our model. To quantify the appearance, we use CenseoQoE metric which is a non-reference metric to capture perceptual differences in appearance [10]. Without ground truth references, these metrics cannot provide meaningful evaluations and we thus use Chamfer distance (Supp. 6) and CenseoQoE [10] (appearance) for our analysis

The success of our method stems from three key innovations: (1) temporal modeling of Gaussian lineages enables stable extrapolation across resolutions, (2) Hessian-weighted optimization prioritizes geometrically significant regions, and (3) dynamic pruning reduces memory overhead without quality loss. While most scenes show substantial gains, the Lego and Chair scene's minimal improvement (1% error reduction) suggests limitations with homogeneous textures, likely due to insufficient lineage variation for our temporal attention mechanism. This edge case notwithstanding, our approach consistently outperforms the baseline in both geometric accuracy and perceptual quality, validating the efficacy of treating Gaussian evolution as a continuous temporal process rather than discrete resolution-specific optimizations.

5. Conclusions

We presented a novel resolution-agnostic framework for 3D Gaussian Splatting that introduces lineage-aware Gaussian modeling and a Hessian-weighted generative architecture. Our key innovation lies in treating Gaussian evolution across resolutions as a temporal sequence, enabling meta-learning of continuous scale priors through transformer-based VAEs with normalizing flow regularization. Experiments demonstrate that our approach outperforms traditional per-resolution optimization both qualitatively and quantitatively. As for limitations, our method assumes smooth Gaussian property transitions across scales, which may break for abrupt scene topology changes.

References

[1] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024. 2

[2] Xiang Feng, Yongbo He, Yubo Wang, Yan Yang, Wen Li, Yifei Chen, Zhenzhong Kuang, Jianping Fan, Yu Jun, et al. Srgs: Super-resolution 3d gaussian splatting. *arXiv preprint arXiv:2404.10318*, 2024. 1

[3] Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. Gaussiansr: High fidelity 2d gaussian splatting for arbitrary-scale image super-resolution. *arXiv preprint arXiv:2407.18046*, 2024. 1

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[5] Kai Luan, Chenghao Shi, Neng Wang, Yuwei Cheng, Huimin Lu, and Xieyuanli Chen. Diffusion-based point cloud super-resolution for mmwave radar data. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11171–11177. IEEE, 2024. 2

[6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[7] Guocheng Qian, Abdulellah Abualshour, Guohao Li, Ali Thabet, and Bernard Ghanem. Pu-gcn: Point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11683–11692, 2021. 2

[8] Yuan Shen, Duygu Ceylan, Paul Guerrero, Zexiang Xu, Niloy J Mitra, Shenlong Wang, and Anna Frühstück. Supergaussian: Repurposing video models for 3d super resolution. In *European Conference on Computer Vision*, pages 215–233. Springer, 2024. 1

[9] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6445–6454, 2022. 2

[10] Shaoguo Wen and Junle Wang. A strong baseline for image and video quality assessment. *arXiv preprint arXiv:2111.07104*, 2021. 4

[11] Shiyun Xie, Zhiru Wang, Yinghao Zhu, and Chengwei Pan. Supergs: Super-resolution 3d gaussian splatting via latent feature field and gradient-guided splitting. *arXiv preprint arXiv:2410.02571*, 2024. 1

[12] Shutong Ye, Shengyu Zhao, Yaocong Hu, and Chao Xie. Single-image super-resolution challenges: a brief review. *Electronics*, 12(13):2975, 2023. 1

GaussianVAE: Adaptive Learning Dynamics of 3D Gaussians for High-Fidelity Super-Resolution

Supplementary Material

6. Training Details

The training objective for our GaussianVAE framework integrates:

- **KL Divergence Loss** ($\mathcal{L}_{\text{KL}}$): Regularizes the latent space using a normalizing flow prior $p_\psi(z)$ and cyclic annealing to avoid posterior collapse. For a flow transformation $z = T_\psi(\epsilon)$ where $\epsilon \sim \mathcal{N}(0, I)$:

$$\mathcal{L}_{\text{KL}} = \beta(t) \cdot D_{\text{KL}}\left(q(z \mid x) \,\|\, p_\psi(z)\right), \quad (2)$$

$$p_\psi(z) = \mathcal{N}(0, I)\left|\det \frac{\partial T_\psi^{-1}}{\partial z}\right|, \quad (3)$$

  where $\beta(t) = \min(\alpha \cdot t \mod R, 1)$ is the cyclic annealing schedule with ramp length R and scaling factor α. This schedule periodically increases β from 0 to 1 during training to gradually introduce KL regularization.
- **MSE Loss** ($\mathcal{L}_{\text{MSE}}$): Supervises appearance attributes (RGB colors $\mathbf{c}$ and opacities α):

$$\mathcal{L}_{\text{MSE}} = \|\mathbf{c} - \hat{\mathbf{c}}\|_2^2 + \|\alpha - \hat{\alpha}\|_2^2, \quad (4)$$

  where $\hat{\mathbf{c}}$, $\hat{\alpha}$ are predicted values.
- **Chamfer Loss** ($\mathcal{L}_{\text{chamfer}}$): Measures geometric fidelity between predicted ($\mathcal{P}$) and ground-truth ($\mathcal{Q}$) Gaussian centers:

$$\mathcal{L}_{\text{chamfer}} = \sum_{p \in \mathcal{P}} \min_{q \in \mathcal{Q}} \|p - q\|_2^2 + \sum_{q \in \mathcal{Q}} \min_{p \in \mathcal{P}} \|q - p\|_2^2. \quad (5)$$

7. Resolution scaling

Computational resources required to train 3DGS at higher resolutions scales quadratically and training using commercial hardware becomes infeasible.

Table 3. Video RAM (VRAM) requirements by output resolution

Resolution (p)	VRAM (GB)
144	>0.6
240	>1.0
360	>1.5
480	>2.2
720	>4.0
1080	>8.0
1440	>14.0

8. Pre-processing

Table 4. Model comparison of size, training time, and inference time

Model	Size	Training time (h)	Inference time (s)
Geometric	0.5MB	coarse (0.5) + fine (2)	0.005
Appearance	1.0MB	coarse (2) + fine (0.5)	0.010

9. Feature composition

Table 5. Feature composition of geometric and appearance models

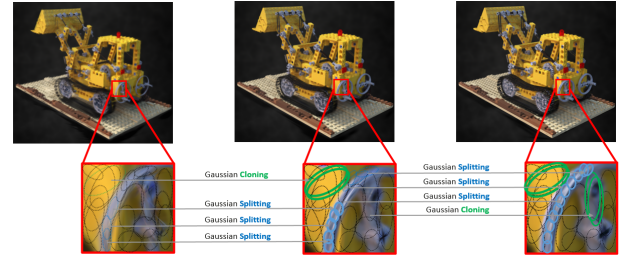| Model	Total Features	Breakdown
Geometric	11	Position (3), Rotation (4), Scale (3), Exist (1)
Appearance	50	SPH (45), RGB (3), Opacity (1), Exist (1)

10. 3DGS densification

Figure 7. **3DGS**: The core 3DGS densification process uses a gradient based threshold for increasing the density of Gaussians in regions that require fine-detail.