

# KD-GS: Knowledge Distillation for Lightweight 3D Gaussian Splatting

Donggun Lee

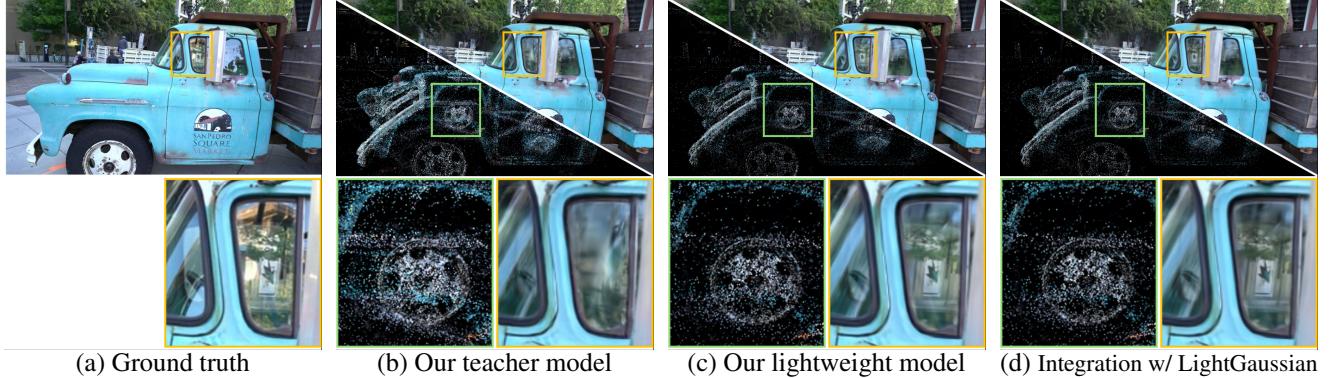


Figure 1. Visual comparison of different model configurations. For each method, the top row shows the rendered image (upper right) and Gaussian point cloud visualization (lower left), separated by a diagonal line. The bottom row shows zoomed-in regions (highlighted by colored boxes in the top row) for detailed comparison. (a) Ground truth image; (b) Our teacher model (612MB, PSNR 25.43); (c) Our lightweight model with knowledge distillation (203MB, PSNR 25.15); (d) Integration of (c) with LightGaussian [5] for additional compression (39MB, PSNR 24.89).

## Abstract

Recent advances in 3D Gaussian Splatting (3DGS) have demonstrated its effectiveness for real-time, high-quality novel view synthesis. However, representing complex scenes often requires millions of Gaussians, leading to significant memory and computational costs. We propose a knowledge distillation framework to generate a lightweight 3DGS model while preserving high visual fidelity. We identify the unique challenges of applying knowledge distillation to 3DGS as the significant differences in the spatial distribution and the number of Gaussians between teacher and student models, which make direct feature matching impractical for effective knowledge transfer. To address the challenges, we introduce two key techniques: visibility-aware neighbor matching and spherical harmonics coefficient distillation, enabling effective knowledge distillation across models with different Gaussian structures. In addition, our method preserves the original 3DGS representation, allowing for seamless integration with existing 3DGS techniques, e.g., for compression, in a plug-and-play manner. Experimental results show a 70% reduction in model size, a 2-3× speedup in rendering, and comparable visual quality, making high-quality 3D scene representation more accessible

for resource-limited environments, such as mobile devices.

## 1. Introduction

Recent advances in neural rendering, particularly since NeRF [15], have revolutionized novel view synthesis through volumetric rendering approaches [1, 3, 6, 16, 27, 28]. While most methods rely on volumetric representations, several approaches have explored differentiable point-based rendering [11, 30, 32–34] for its efficiency and explicit geometry representation. Building upon these point-based methods, 3D Gaussian Splatting (3DGS) [10] has emerged as a powerful approach by representing scenes with explicit 3D Gaussians. Each Gaussian serves as an enhanced point primitive with attributes such as position, size, rotation, and color, enabling efficient rendering through traditional rasterization pipeline. This explicit representation, combined with parallel rasterization, makes 3DGS highly suitable for real-time applications.

However, a key limitation of 3DGS is its reliance on a large number of Gaussians to accurately represent complex scenes. For representing unbounded scenes, 3DGS requires millions of Gaussians, leading to substantial memory demands and computational overhead. This overhead

significantly limits its practical applicability in resource-constrained environments, such as mobile devices, and large-scale scene reconstruction.

Several methods have been proposed to reduce this computational overhead of the 3DGS representation through various techniques. Some methods retain the Gaussian primitive but modify how features are stored and accessed, such as by using vector quantization, which requires additional codebook structures [5, 17, 18, 23]. Other approaches fundamentally alter the organization of Gaussians through techniques like hash-grid encoding or octree-based partitioning [8, 12, 20]. While these approaches are effective, they often require significant structural modifications to the original 3DGS framework, limiting compatibility with other techniques.

In this paper, we propose a novel knowledge distillation approach that achieves lightweight while preserving the original representation structure. Our method is designed to be plug-and-play, meaning it can be seamlessly integrated with existing 3DGS techniques, without requiring significant structural modifications.

For our knowledge distillation framework, we exploit the densification mechanism of 3DGS to control the number of Gaussians between teacher and student models. We define the teacher model as a pretrained dense 3D Gaussian representation with a large number of Gaussians, trained for capturing high-quality scene details. In contrast, the student model is designed to use fewer Gaussians, achieving a more compact representation of the same scene.

This setup presents two main challenges, which is not common in knowledge distillation for neural networks [2, 21, 26]. First, the spatial distribution difference between teacher and student models creates correspondence ambiguity. The significant difference in the numbers of Gaussians complicates the process of establishing meaningful correspondences between the models. Second, the Gaussian feature characteristics should be different between the models. The teacher model employs dense and smaller Gaussians for fine-grained detail capture, while the student model utilizes fewer, larger Gaussians to cover broader regions efficiently. This mismatch in size and density makes direct feature matching impractical, as the Gaussians serve different structural purposes in each model.

We overcome these challenges through two key technical components. First, we introduce a visibility-aware neighbor matching mechanism that effectively resolves the ambiguous correspondence problem between teacher and student Gaussians by using visibility filtering and geometric proximity. Second, we identify spherical harmonics coefficients as our distillation target to focus on appearance, rather than geometric features that could be inconsistent among teacher and student models due to different Gaussian densities.

Our experimental results show that our knowledge dis-

tillation framework achieves significant model size reduction (70%, from 700MB to 200MB) while maintaining comparable visual quality across different scene types, along with 3 $\times$  speedup in rendering performance (from 50 FPS to 140 FPS). Through our visibility-aware neighbor matching and spherical harmonics distillation, we effectively transfer knowledge from teacher to student models despite their different spatial configurations. In addition, our method’s plug-and-play design also allows it to be seamlessly integrated with other effective methods for 3DGS. This capability is demonstrated by successful integration with LightGaussian [5], where LightGaussian is combined with our approach to provide additional benefit of compression.

Our contributions can be summarized as follows:

- A novel knowledge distillation framework for that significantly reduces model size while maintaining high visual fidelity.
- The introduction of visibility-aware neighbor matching to resolve ambiguous correspondences between teacher and student models.
- The use of spherical harmonics coefficients for effective appearance-based distillation.
- A plug-and-play design that preserves the original 3DGS representation, enabling seamless integration with other optimization methods.

## 2. Related Work

### 2.1. Lightweight Neural Scene Representations

Neural Radiance Fields (NeRF) [15] revolutionized novel view synthesis through its continuous volumetric scene representation. However, its demanding ray marching process and network inference have led to various optimization efforts. Early approaches like KiloNeRF [19] and FastNeRF [7] focused on architectural improvements through network decomposition and caching strategies, while SNeRG [9] explored efficient data structures through sparse voxel grids.

Recent works have pursued different directions for efficiency. Some methods focus on alternative scene representations: TensoRF [3] represents scenes as 4D tensors with multi-channel features, while Plenoxels [6] uses explicit sparse voxel grids. By avoiding MLPs, these approaches achieve faster training and smaller models. Other approaches like Instant-NGP [16, 28] dramatically improved training and rendering speed through multiresolution hash encoding, allowing for smaller networks while preserving representation power.

Several methods have explored converting neural representations into more efficient formats. R2L [25] eliminates iterative sampling by distilling NeRF knowledge into a light field representation. MobileNeRF [4] and Neural Duplex

Radiance Fields [24] convert neural representations into traditional graphics primitives for hardware-accelerated rendering.

While these NeRF-based methods have significantly improved rendering efficiency through various approaches, achieving both real-time performance and photorealistic quality remains challenging with volumetric representations. Most methods face a trade-off: optimizing for speed often compromises visual quality, while maintaining high fidelity requires longer rendering times. This limitation has motivated exploration of alternative approaches, particularly point-based methods like 3D Gaussian Splatting.

## 2.2. Efficient 3D Gaussian Splatting

Based on the point-based radiance fields [11, 32, 34] and differentiable point rendering techniques [30, 33], 3D Gaussian Splatting (3DGS) [10] innovated neural rendering by representing scenes with explicit 3D Gaussians, enabling real-time rendering through traditional rasterization pipelines. Unlike NeRF-based methods that rely on computationally intensive ray marching, 3DGS achieves high rendering speed and efficiency through parallelized rasterization, outperforming even lightweight NeRF variants in terms of computational requirement. However, accurately representing complex scenes requires millions of Gaussians, leading to substantial memory demands for typical unbounded scenes.

Recent efforts have addressed efficient 3DGS representation while maintaining its rendering quality through various approaches. Feature compression methods maintain the original Gaussian structure while reducing memory requirements: CompGS [17] and CompressedGS [18] utilize vector quantization with codebook-based parameter sharing, while EAGLES [23] employs quantized embeddings with coarse-to-fine training. LightGaussian [5] combines multiple optimization strategies including pruning, SH coefficient distillation, and vector quantization for comprehensive model compression.

Alternative approaches explore different primitive representations and structural modifications. GES [8] replaces traditional Gaussians with generalized exponential functions that require fewer particles. ScaffoldGS [13] introduces an anchor-based hierarchical representation where each anchor point spawns multiple neural Gaussians, and ContextGS [29] builds upon this by adding an autoregressive context model to leverage spatial dependencies.

Spatial organization methods focus on efficient scene structuring: C3DGS [12] employs hash-grid-based encoding with masking strategies, while OctreeGS [20] introduces level-of-detail structures for consistent rendering performance. MesonGS [31] combines octree-based partitioning with attribute transformation for post-training compression.

While these methods achieve effective model compression using various techniques, they often require complex modifications to the original 3DGS framework. These structural modifications, while effective individually, make it challenging to combine different approaches due to potential incompatibilities in their implementations. Our work addresses this limitation through knowledge distillation, maintaining the original Gaussian representation while achieving lightweight models. The effectiveness of our plug-and-play approach is demonstrated by successfully integrating our method with LightGaussian [5] for additional compression while preserving the merits of both methods.

## 3. Method

### 3.1. 3D Gaussian Splatting

3D Gaussian Splatting [10] (3DGS) represents a 3D scene as a collection of Gaussian primitives. Each Gaussian  $G_i$  is parameterized by its mean position  $\mu_i \in \mathbb{R}^3$ , covariance matrix  $\Sigma_i \in \mathbb{R}^{3 \times 3}$ , opacity  $\alpha_i \in [0, 1]$ , and view-dependent color features:

$$G_i = \{\mu_i, \Sigma_i, \alpha_i, SH_i\}, \quad (1)$$

where  $SH_i$  represents the spherical harmonics coefficients used to encode view-dependent color information.

During rendering, these 3D Gaussians are projected to 2D screen space. For each pixel  $(u, v)$  in view  $t$ , we compute the accumulated opacity  $\sigma_i^t(u, v)$  for each Gaussian  $G_i$  through alpha blending:

$$\sigma_i^t(u, v) = \alpha_i \prod_{j < i} (1 - \alpha_j), \quad (2)$$

where the Gaussians are ordered by depths at pixel  $(u, v)$ . This accumulated opacity represents how much each Gaussian actually contributes to the final rendered image at that pixel location.

The final color of each pixel is then computed using these accumulated opacities:

$$C_t(u, v) = \sum_{i \in N} c_i \sigma_i^t(u, v), \quad (3)$$

where  $c_i$  represents the view-dependent color computed from spherical harmonics coefficients ( $SH_i$ ).

During training, model parameters are optimized using a combination of L1 color loss and structural similarity (SSIM) loss:

$$\mathcal{L}_{gt} = (1 - \lambda_{ssim}) \mathcal{L}_{L1} + \lambda_{ssim} \mathcal{L}_{ssim}, \quad (4)$$

where  $\mathcal{L}_{L1}$  and  $\mathcal{L}_{ssim}$  are common image-domain loss functions.

A key characteristic of 3DGS is the densification mechanism during training. The densification threshold controls

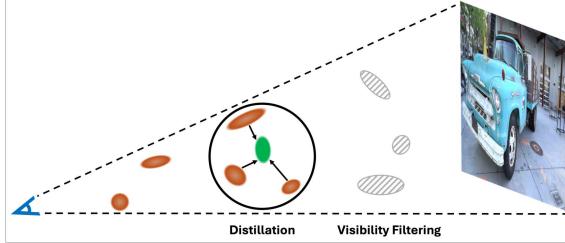


Figure 2. Visualization of our visibility-aware knowledge distillation mechanism. The dotted line represents the view frustum, with grey striped ellipses showing filtered-out teacher Gaussians. Brown and green ellipses indicate visible teacher and student Gaussians, respectively. The black circle highlights our weighted neighbor matching for knowledge distillation.

where new Gaussians should be added to improve the scene representation, directly affecting the final number of Gaussians in the model. This adaptive process ensures an efficient distribution of Gaussians: regions needing more details receive additional Gaussians through splitting, while well-represented regions maintain their current density. A smaller threshold leads to more frequent splitting, resulting in a denser representation that captures finer scene details. Conversely, a larger threshold yields a more compact model with fewer Gaussians. For a more detailed explanation of the 3DGS framework, we refer readers to [10].

### 3.2. Knowledge Distillation

Our objective is to achieve lightweight 3DGS by effectively training a student model having fewer Gaussians to match the rendering quality of a denser teacher model. Our knowledge distillation framework consists of two stages. First, we train a teacher model using the standard 3DGS objective (Eq. (4)) and a lower densification threshold, to create a high-quality scene representation with dense Gaussians. Once the teacher model has been fully trained and fixed, we train a student model with a higher densification threshold, naturally resulting in fewer Gaussians through less frequent splitting. During the student model training, we combine the original rendering loss with our knowledge distillation loss to effectively transfer appearance information from the teacher to the student.

#### 3.2.1. Knowledge Distillation in 3DGS

Knowledge distillation for 3DGS presents unique challenges, due to the evolving nature of the student model’s spatial distribution and the structural differences between the teacher and student models.

First, the student model’s spatial distribution evolves dynamically during training due to the densification mechanism. New Gaussians are added based on reconstruction error, leading to continuous changes in the student model’s Gaussian arrangement. This dynamic nature makes direct

feature matching between teacher and student models ambiguous.

Second, structural differences between the teacher and student models complicate defining appropriate distillation targets. For example, the teacher model uses a denser arrangement of smaller Gaussians to capture fine details, while the student model represents the same scene with fewer, larger Gaussians. Simply matching the teacher’s Gaussian properties would create gaps in the scene, as the student’s fewer Gaussians with the teacher’s Gaussian sizes cannot provide the same level of coverage. Therefore, it is essential to define distillation targets that account for these differences without compromising the scene’s visual fidelity.

To address these challenges, we introduce two complementary solutions. First, we propose a visibility-aware neighbor matching mechanism, which selects relevant teacher Gaussians based on their visual importance and proximity to each student Gaussian. This ensures that only the most important teacher Gaussians are used during student training, enabling effective knowledge transfer. Second, we use spherical harmonics coefficients as our distillation target since they encode color information independently of the underlying geometry. This allows the student model to retain its own geometric configuration while learning detailed color information from the teacher model.

#### 3.2.2. Knowledge Distillation Loss

To implement our solutions, we define a knowledge distillation loss that is based on our visibility-aware neighbor matching and effectively transfers the appearance knowledge of spherical harmonics coefficients from the teacher to the student model..

First, we filter teacher Gaussians based on their visibility in each training view  $t$ . Using the opacity accumulation defined in Eq. (2), we identify visible teacher Gaussians by thresholding their accumulated opacity:

$$\mathcal{G}_T^t = \{G_i | \sigma_i^t > \tau\}, \quad (5)$$

where  $\tau$  is the opacity threshold and  $\mathcal{G}_T^t$  is the set of filtered teacher Gaussians visible in training view  $t$ . For each Gaussian  $G_i$ , we take the maximum accumulated opacity across all pixels it projects onto, ensuring we capture its highest contribution to the view. This filtering guarantees that we focus on teacher Gaussians that meaningfully contribute to the scene appearance, avoiding distillation from occluded or negligible Gaussians.

After filtering, we establish correspondences using  $K$ -nearest neighbor matching. For each student Gaussian  $G_s$ , we find its  $K$  nearest neighbors from the filtered set  $\mathcal{G}_T^t$  based on Euclidean distance in 3D space. Let  $\mathcal{K}(s, t)$  denote the set of  $K$  nearest teacher Gaussians of a student Gaussian  $s$  in training view  $t$ . The contribution weight  $w_k^t$  for each

neighbor is computed inversely proportional to its distance:

$$w_k^t = \frac{1/d_k^t}{\sum_{j \in \mathcal{K}(s,t)} 1/d_j^t} \quad (6)$$

where  $d_k^t$  is the Euclidean distance between the student Gaussian and its k-th neighbor.

These weighted correspondences guide our knowledge distillation through a weighted average of spherical harmonics coefficients:

$$\mathcal{L}_{kd} = \sum_t \sum_s \left\| SH_s - \sum_{k \in \mathcal{K}(s,t)} w_k^t SH_k \right\|^2 \quad (7)$$

where  $SH_s$  and  $SH_k$  are the spherical harmonics coefficients of student and teacher Gaussians, respectively. This approach allows each student Gaussian to learn from multiple relevant teacher Gaussians, making the knowledge transfer robust to different spatial distributions between models. During training, we augment the views by adding small random perturbations to their positions and orientations, to create diverse visibility filtering results and improve the robustness of the distillation process.

The final student loss combines the rendering loss in Eq. (4) with our knowledge distillation loss:

$$\mathcal{L}_{student} = \mathcal{L}_{gt} + \lambda_{kd} \mathcal{L}_{kd}, \quad (8)$$

where  $\lambda_{kd}$  controls the strength of knowledge distillation. By combining these losses, we ensure that the student model learns essential appearance information from the teacher, enabling high-quality rendering with significantly fewer Gaussians.

### 3.3. Student Model Initialization

Following the standard 3DGS pipeline, we initialize the student model using the COLMAP-reconstructed point cloud and train it with a higher densification threshold than the teacher model. This results in fewer Gaussians due to less frequent splitting, creating a compact representation while learning from the denser teacher model.

We can also explore an alternative initialization strategy by subsampling Gaussians from the trained teacher model, similarly to point cloud subsampling. This approach provides the student with pre-optimized features from the teacher as its initial condition. Then, the student model can be trained with our knowledge distillation loss and higher densification threshold in the same way as the case of using the COLMAP-reconstructed point cloud.

Our experiments demonstrate that both initialization strategies achieve comparable final rendering quality, confirming that our knowledge distillation framework is insensitive to initialization. Therefore, we adopt the COLMAP-based initialization, which is simpler and also maintains the standard 3DGS pipeline.

Initialization	PSNR↑	SSIM↑	LPIPS↓
COLMAP	30.15	0.909	0.259
Teacher Subsample	30.30	0.910	0.259

Table 1. Quantitative comparison between different initialization strategies. Scene: **Playroom**. Both COLMAP-based initialization and teacher model subsampling achieve comparable rendering quality when combined with our knowledge distillation process, demonstrating the robustness of our approach against the initialization method. Arrows ( $\uparrow/\downarrow$ ) indicate whether higher or lower values are better.

## 4. Experiments

### 4.1. Experimental Setup and Implementation

We implement our knowledge distillation framework on top of the original 3DGS [10] to demonstrate its effectiveness. We evaluate our framework on two standard datasets: Tanks & Temples and Deep Blending.

As described in Sec. 3.2, we control the scene representation density by adjusting the densification parameter. In our experiments, the parameter is set to 0.002 for the teacher model and 0.003 for the student model, creating models with different Gaussian densities. For the visibility-aware neighbor matching, we set  $K = 3$  for the K-nearest neighbor search. We set the knowledge distillation weight  $\lambda_{kd}$  to 0.1, balancing between the original rendering loss and our distillation loss. Following the standard 3DGS pipeline, we initialize our models using COLMAP-reconstructed point clouds except the case of comparing initialization strategies. All experiments were conducted on a single Quadro RTX 5000 GPU with 16GB memory.

### 4.2. Initialization Comparison

We compare the initialization methods, which we discussed in Sec. 3.3, to validate the robustness of our knowledge distillation framework. While the alternative subsampling approach inherits the teacher's optimized spatial distribution and could potentially provide better initial setting, our experiments show that both initialization strategies achieve similar final quality when combined with our knowledge distillation process. Tab. 1 shows the quantitative results with comparable PSNR, SSIM, and LPIPS scores.

These results demonstrate that our knowledge distillation framework effectively transfers appearance information regardless of the initialization method. Furthermore, our choice of COLMAP-based initialization offers an additional practical advantage of the consistency with the teacher model's pipeline, simplifying the overall implementation. Consequently, we use the COLMAP-based initialization for all subsequent experiments in this paper.

	Tanks & Temples					Deep Blending				
Method	PSNR↑	SSIM↑	LPIPS↓	FPS↑	Size↓	PSNR↑	SSIM↑	LPIPS↓	FPS↑	Size↓
3DGS	23.71	0.848	0.177	87.2	434MB	29.57	0.904	0.244	51.1	666MB
3DGS+KD	23.58	0.836	0.207	200	144MB	29.70	0.906	0.256	142	205MB
LightGaussian [5]	23.32	0.829	0.205	200	29MB	29.11	0.894	0.262	133	43MB
LightGaussian+KD	23.27	0.828	0.217	198	28MB	29.51	0.905	0.264	134	39MB

Table 2. Quantitative comparison on Tanks & Temples and Deep Blending datasets. 3DGS represents the dense teacher model, while 3DGS+KD is our lightweight model trained with knowledge distillation. LightGaussian shows the results of applying pruning and compression steps of [5] to the teacher model, and LightGaussian+KD represents the results of applying the compression step of [5] to our 3DGS+KD model. Our 3DGS+KD achieves significant model size reduction while maintaining comparable rendering quality to 3DGS. Though LightGaussian and LightGaussian+KD show similar metrics, LightGaussian+KD preserves more visual details as shown in Fig. 6.



Figure 3. Qualitative results of the effectiveness of knowledge distillation on 3DGS. (a)-(d) Ground truth images from different views. (b)-(e) 3DGS, our teacher model with dense Gaussians. (c)-(f) 3DGS+KD, our lightweight model trained with knowledge distillation. The lightweight 3DGS+KD shows comparable visual quality to the teacher 3DGS model.

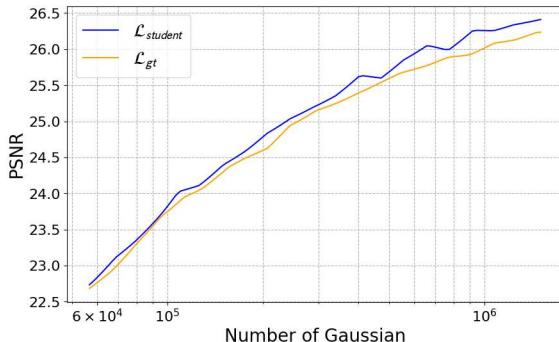


Figure 4. PSNR comparison between models trained with Eq. (4) (orange) and Eq. (8) (blue) for varying numbers of Gaussians (log scale). Scene: **Truck**. With the same number of Gaussians, models trained with our knowledge distillation consistently achieve higher PSNR, demonstrating the effectiveness of our approach.

#### 4.3. Knowledge Distillation Analysis

We evaluate the effectiveness of our lightweight model trained with knowledge distillation. In the following, 3DGS refers to the dense teacher model, and 3DGS+KD denotes our lightweight model with knowledge distillation. The results are shown in Tab. 2 and Fig. 3.

K	PSNR↑	SSIM↑	LPIPS↓
1	21.84	0.792	0.249
3	<b>22.02</b>	<b>0.797</b>	<b>0.244</b>
5	21.81	0.793	0.246
7	21.80	0.794	0.247

Table 3. Ablation study on the number of neighbors ( $K$ ) in our K-nearest neighbor matching. Scene: **Train**.

Our student model achieves about 70% reduction in the model size while preserving comparable visual quality to the original 3DGS. This significant size reduction is achieved by reducing the number of Gaussians with our knowledge distillation technique. As a direct benefit, the resulting smaller number of Gaussians improves the rendering performance, demonstrating 2-3× speedup. The visual comparison in Fig. 3 shows that our lightweight model successfully preserves the rendering quality, despite the drastic reduction in the number of Gaussians.

Interestingly, as shown in Fig. 3, our knowledge distillation approach sometimes produces cleaner results in regions where the teacher model shows noisy appearance. This indicates that our method provides a regularization

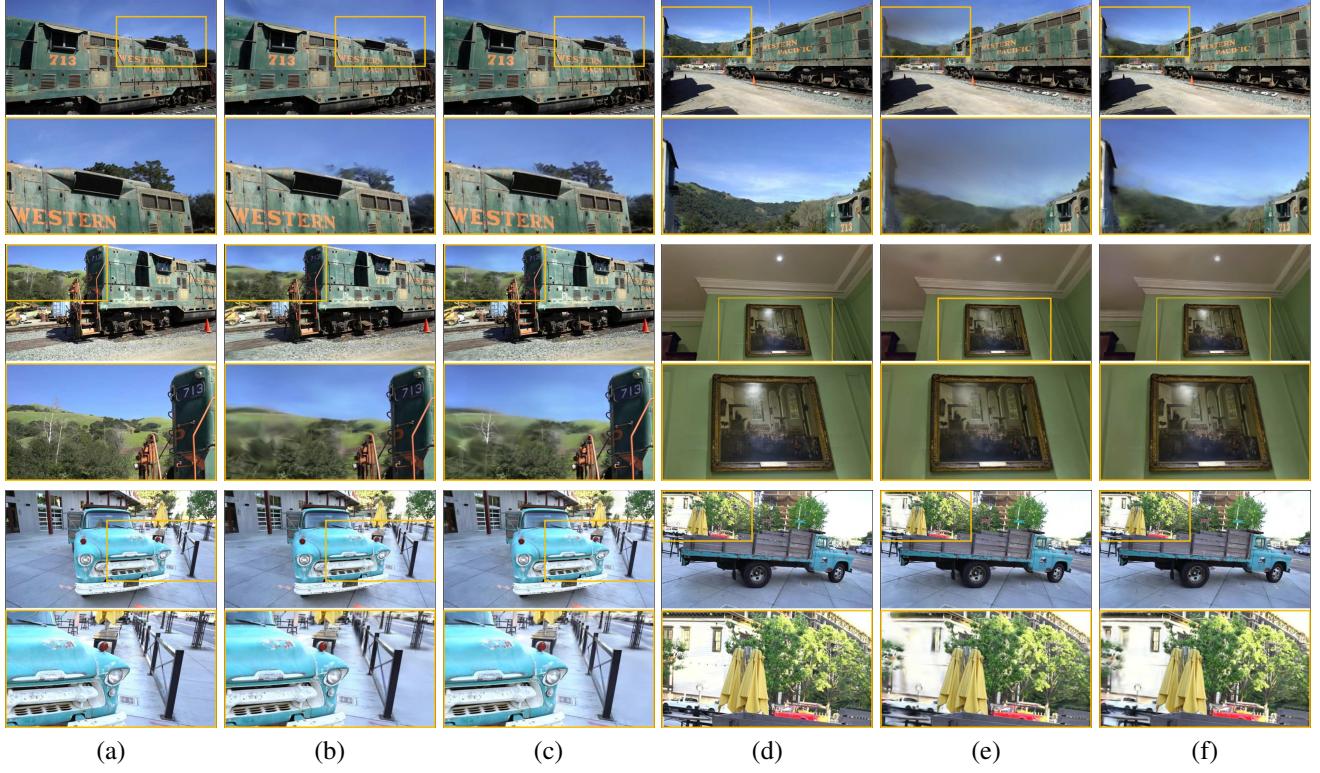


Figure 5. Visual comparison between lightweight models trained with and without our knowledge distillation loss. (a)(d) Ground truth images. (b)(e) 3DGS+LW, lightweight model trained with only ground truth loss (Eq. (4)). (c)(f) 3DGS+KD, lightweight model trained with our knowledge distillation loss (Eq. (8)). Lightweight models show quality degradation due to the reduced number of Gaussians; however, our knowledge distillation effectively mitigates this by transferring appearance information from the teacher model.

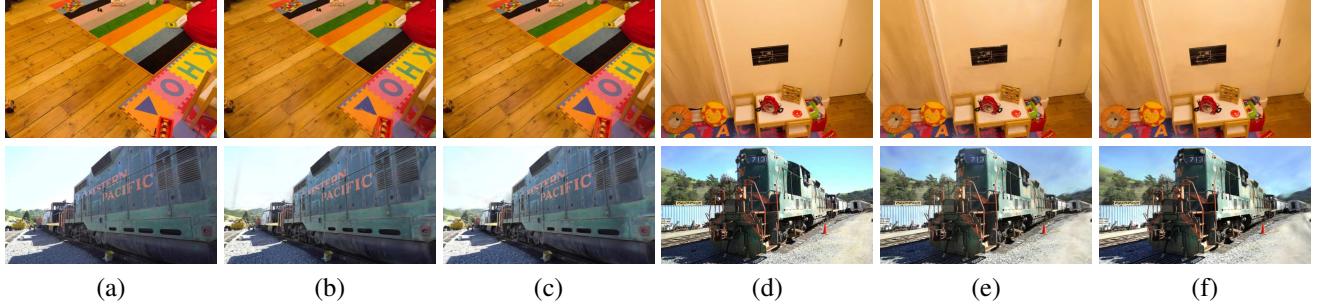


Figure 6. Qualitative results of integrating our knowledge distillation with LightGaussian [5]. (a)(d) Ground truth images from different views. (b)(e) LightGaussian applied to our teacher model. (c)(f) LightGaussian+KD, our method combined with LightGaussian. Our integrated approach (LightGaussian+KD) shows better preservation of fine details compared to the original LightGaussian, showing that our knowledge distillation remains effective even when combined with other 3DGS methods.

effect that can help reduce potential visual artifacts in the teacher model.

To further analyze the benefits of knowledge distillation, we compare two training strategies: one using only the original 3DGS objective (Eq. (4)) and another incorporating our knowledge distillation loss (Eq. (8)). We evaluate PSNRs across different model sizes by controlling the number of Gaussians through the densification threshold param-

eter. As shown in Fig. 4, for any given number of Gaussians, the model trained with Eq. (8) consistently achieves higher PSNR compared to the model trained with Eq. (4). This consistent improvement in rendering quality validates that our knowledge distillation effectively transfers appearance information from the teacher to the student model.

The effectiveness of our knowledge distillation is further demonstrated in Fig. 5, where we compare the visual

quality of our approach (3DGS+KD) with the lightweight model trained without knowledge distillation (3DGS+LW). While 3DGS+LW shows quality degradation due to the reduced capacity, 3DGS+KD achieves better rendering quality. Specifically, 3DGS+KD reduces gray artifacts in smooth regions, and better preserves the appearance in high view-dependency region, demonstrating improved overall visual fidelity.

We also analyze the effect of  $K$  in our K-nearest neighbor matching through experiments shown in Tab. 3. When  $K = 1$ , the model shows lower performance as it relies on information from only a single teacher Gaussian, limiting the knowledge transfer. Setting  $K = 3$  provides the best results in the rendering quality by incorporating information from multiple teacher Gaussians while keeping the computational overhead small. Increasing  $K$  to larger values ( $K = 5, 7$ ) does not show additional performance gains, but it increases computational cost. The similar or slightly decreased rendering quality with larger  $K$  values implies that considering too many neighbor Gaussians may introduce undesirable noisy information in the knowledge transfer process. Based on these results, we set  $K = 3$  as our default setting.

#### 4.4. Integration with Existing Methods

Our method can be integrated with existing 3DGS techniques in a plug-and-play manner, enabling additional improvement or compression depending on the combined method. To demonstrate this capability, we integrate our approach with LightGaussian [5], a compression method for 3DGS that consists of three stages: Gaussian pruning based on global significance score (opacity and volume), spherical harmonics compression through degree reduction and view-dependent color matching, and vector quantization of spherical harmonics coefficients with additional precision optimization.

We integrate our method by replacing LightGaussian’s pruning stage with our knowledge distillation approach while keeping the subsequent compression stages. We refer the integrated version to LightGaussian+KD in the following experiments, while the original LightGaussian applied to our teacher model is denoted as LightGaussian.

The experimental results in Tab. 2 show that both LightGaussian and LightGaussian+KD achieve comparable performance in quantitative measures. However, the visual results in Fig. 6 reveal that LightGaussian+KD better preserves visual details, thanks to appearance features inherited through the knowledge distillation process that persist even after applying the compression steps of LightGaussian.

### 5. Conclusion

In this paper, we introduced a novel knowledge distillation framework for 3D Gaussian Splatting, addressing two key

challenges: the unaligned nature of point-based representations of teacher and student models and the unclear definition of appropriate distillation target. We addressed the challenges by proposing visibility-aware neighbor matching and using spherical harmonics coefficients as the distillation target. Our framework effectively transfers knowledge from a dense teacher model to a lightweight student model, achieving significant model size reduction (approximately 70%) while preserving high visual quality. The rendering performance is also improved with our results by 2-3 $\times$ , making high-quality 3D scene representation more accessible for resource-limited applications.

In addition, our framework preserves the fundamental Gaussian representation structure, enabling seamless plug-and-play integration with other effective 3DGS techniques. We demonstrated this capability by integrating with LightGaussian [5] to achieve additional data compression. Our knowledge distillation approach offers a practical solution for lightweight 3DGS with simplicity and flexibility. Adapting our approach to other applications of 3DGS, such as avatar modeling [14, 22], would be interesting future work.

### References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 1
- [2] Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022. 1, 2
- [4] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *The Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [5] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*, 2023. To appear in NeurIPS 2024 (Spotlight). 1, 2, 3, 6, 7, 8
- [6] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinrong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 1, 2
- [7] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF*

- International Conference on Computer Vision (ICCV)*, pages 14346–14355, 2021. 2
- [8] Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. Ges : Generalized exponential splatting for efficient radiance field rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19812–19822, 2024. 2, 3
- [9] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 2
- [10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 3, 4, 5
- [11] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, pages 29–43. Wiley Online Library, 2021. 1, 3
- [12] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024. 2, 3
- [13] Tao Lu, Mulin Yu, Lining Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 3
- [14] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH Conference Proceedings, Denver, CO, United States, July 28 - August 1, 2024*, 2024. 8
- [15] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 1, 2
- [16] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1, 2
- [17] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbas Koohpayegani, and Hamed Pirsiavash. Compogs: Smaller and faster gaussian splatting with vector quantization. In *European Conference on Computer Vision*, 2024. 2, 3
- [18] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10349–10358, 2024. 2, 3
- [19] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *International Conference on Computer Vision (ICCV)*, 2021. 2
- [20] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Lining Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2, 3
- [21] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 2
- [22] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 8
- [23] Abhinav Shrivastava Sharath Girish, Kamal Gupta. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, 2024. 2, 3
- [24] Ziyu Wan, Christian Richardt, Aljaž Božič, Chao Li, Vijay Rengarajan, Seonghyeon Nam, Xiaoyu Xiang, Tuotuo Li, Bo Zhu, Rakesh Ranjan, and Jing Liao. Learning neural duplex radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8307–8316, 2023. 3
- [25] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *European Conference on Computer Vision*, 2022. 2
- [26] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3048–3068, 2021. 2
- [27] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1
- [28] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 1, 2
- [29] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *arXiv preprint arXiv:2405.20721*, 2024. To appear in NeurIPS 2024. 3
- [30] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7467–7477, 2020. 1, 3
- [31] Shuzhao Xie, Weixiang Zhang, Chen Tang, Yunpeng Bai, Rongwei Lu, Shijia Ge, and Zhi Wang. Mesongs: Post-training compression of 3d gaussians via efficient attribute transformation. In *European Conference on Computer Vision*, 2024. 3
- [32] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Pointnerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 1, 3

- [33] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. [3](#)
- [34] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–12, 2022. [1](#), [3](#)

# KD-GS: Knowledge Distillation for Lightweight 3D Gaussian Splatting

## Supplementary Material

### 6. Supplementary

#### 6.1. Effectiveness of Knowledge Distillation

To analyze the contributions of knowledge distillation (KD) loss and the densification process for improving the performance of the student model, we conduct experiments on the *Truck scene* from the Tanks & Temples dataset. We evaluate three settings to disentangle the effects of KD loss and densification:

- **Setting 1: Subset of Teacher Gaussians** A static subset of 0.85 million Gaussians was randomly sampled from the teacher model, matching the final Gaussian count of our standard 3DGS+KD model. This subset was directly used for evaluating the rendering performance of the teacher model when reduced to a smaller, subsampled set of Gaussians.
- **Setting 2: KD Loss without Densification** Using the same 0.85 million Gaussians from Setting 1, we trained the student model with KD loss but without enabling densification. The Gaussian count remained fixed throughout training. This configuration isolated the effect of KD loss by focusing on its ability to refine the parameters of a static Gaussian set.
- **Setting 3: KD Loss with Densification** Setting 3 follows the same approach as the default setting of our experiments described in Sec. 4. The student model was initialized using the COLMAP-reconstructed point cloud. During training, KD loss was applied in conjunction with the densification process, allowing the model to dynamically add new Gaussians in underrepresented regions. The final Gaussian count was 0.85 million, similar to that of Settings 1 and 2, but the distribution of Gaussians was adaptively optimized to account for scene complexity.

The results of these settings are presented in Fig. 7, Fig. 8 and Tab. 4.

In Setting 1, rendering with a static subset of Gaussians from the teacher model led to significant visual artifacts, such as detail loss and gaps in scene coverage. In Setting 2, the application of KD loss reduces visual artifacts by refining the parameters of the fixed Gaussians. By learning detailed appearance information from the teacher model, the student model improves rendering quality, with an increase in PSNR of approximately 3.5. However, gaps and coverage holes persisted due to the fixed Gaussian count. Setting 3 achieved the highest rendering quality by combining the KD loss and densification. The densification process dynamically added Gaussians in underrepresented regions, while KD loss enabled effective appearance learning from the teacher. This combination ensured comprehensive scene

Method	PSNR↑	SSIM↑	LPIPS↓
Setting 1	17.38	0.605	0.371
Setting 2	20.92	0.751	0.280
Setting 3	<b>25.15</b>	<b>0.874</b>	<b>0.170</b>

Table 4. Quantitative comparison on the effectiveness of knowledge distillation.

Method	MipNeRF-360				
	PSNR↑	SSIM↑	LPIPS↓	FPS↑	Size↓
3DGS	28.00	0.833	0.167	183	738MB
KD	27.53	0.806	0.218	417	197MB
LG [5]	27.39	0.820	0.187	389	48MB
LG+KD	27.08	0.798	0.226	435	38MB

Table 5. Quantitative comparison on Mip-NeRF 360 dataset. (downscaled by 4x due to RTX3090 GPU memory constraints) Method abbreviations: KD refers to our 3DGS+KD model, LG denotes LightGaussian [5], and LG+KD represents the integration of our knowledge distillation with LightGaussian as described in Sec. 4.4. 3DGS represents the original dense teacher model.

coverage and better detail preservation.

These experimental results demonstrate that KD loss alone significantly enhances rendering quality by refining Gaussian parameters and transferring appearance knowledge. However, combining KD loss with densification offers the best performance, balancing scene fidelity and adaptability. Therefore, we adopt this combined approach as the standard configuration for our framework.

#### 6.2. KD Loss with Varing Number of Gaussian

To provide a more comprehensive evidence for our quantitative analysis in Fig. 4, we present additional visual comparisons in Fig. 9. The visual comparisons consistently demonstrate that while both approaches experience quality degradation with fewer Gaussians, our knowledge distillation approach maintains noticeably better visual fidelity compared to the baseline model. These results further validate that our knowledge distillation framework effectively transfers appearance information across different model capacities, providing consistent quality improvement regardless of the number of Gaussians used.

#### 6.3. Additional Experimental Results

To further validate the effectiveness of our knowledge distillation approach, we present additional experimental results

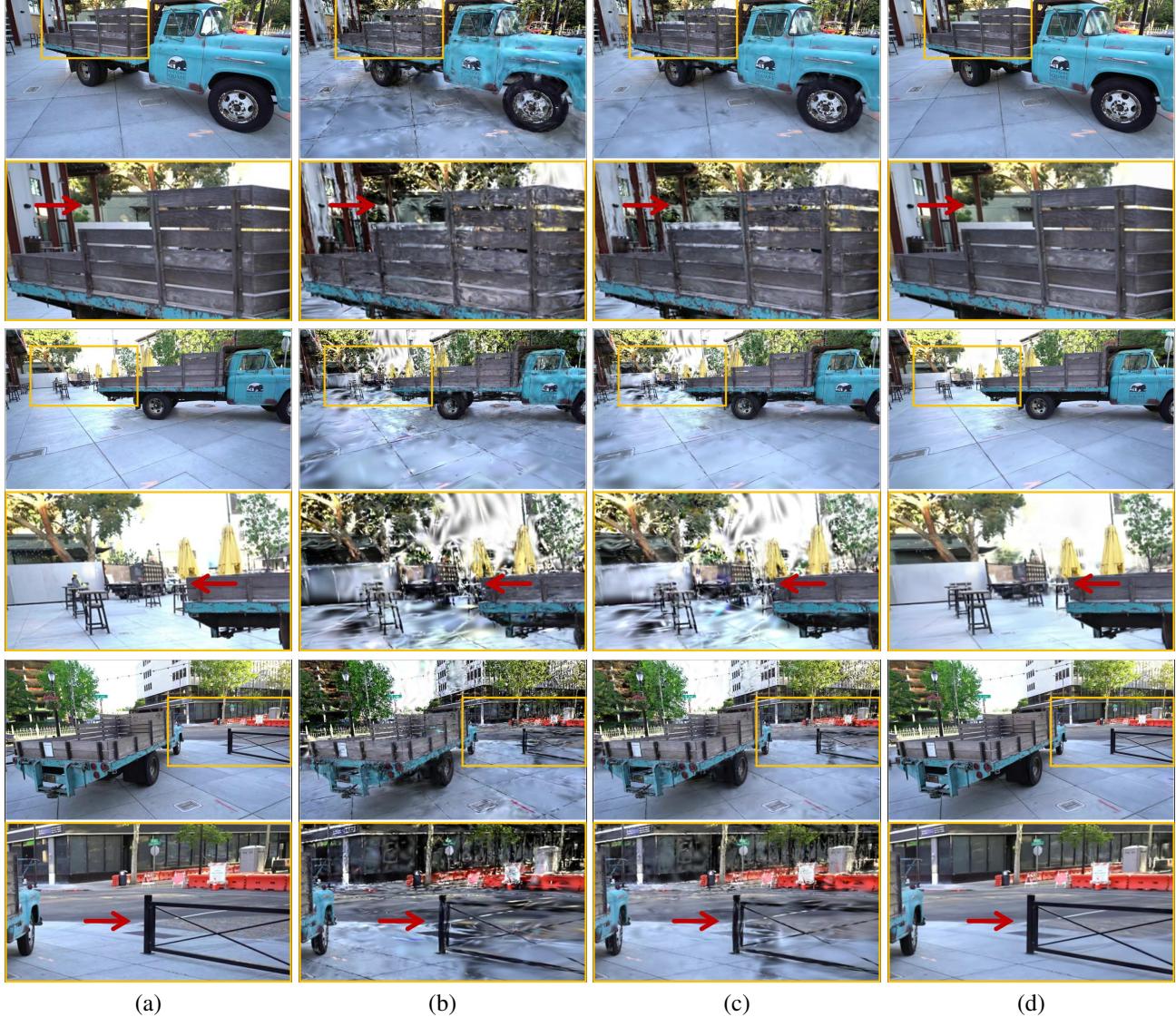


Figure 7. Qualitative comparison on the effectiveness of knowledge distillation. (a) Ground truth images. (b) Results of Setting 1. (c) Results of Setting 2. (d) Results of Setting 3.

in two aspects.

First, we extend our evaluation to the Mip-NeRF 360 dataset, which features challenging unbounded scenes with complex geometry and lighting conditions. For the experiments, we downsampled the input images by a factor of 4 to accommodate the memory constraints of our RTX3090 GPU. The quantitative results in Tab. 5 demonstrate that our method maintains its effectiveness even on these challenging scenes, achieving quality preservation through knowledge distillation.

Second, we provide more comprehensive visual comparisons on previously evaluated datasets. Fig. 10 shows detailed comparisons between our lightweight model and

the original 3DGS. Fig. 11 demonstrates the impact of our knowledge distillation loss. Fig. 12 present results from combining our approach with LightGaussian. The zoomed-in regions in the figures highlight that our lightweight model successfully preserves the fine details despite using significantly fewer Gaussians.

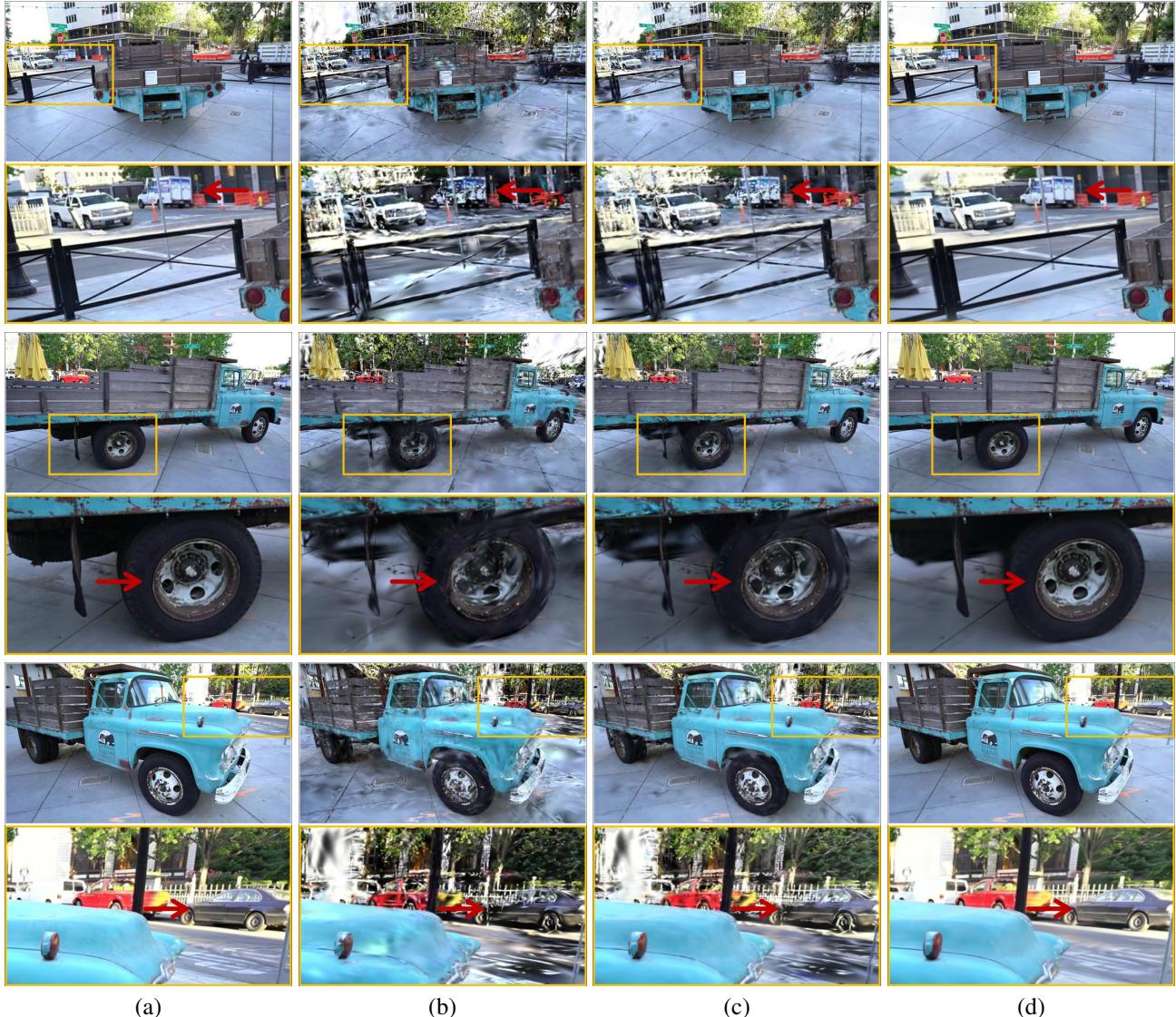


Figure 8. Qualitative comparison on the effectiveness of knowledge distillation. (a) Ground truth images. (b) Results of Setting 1. (c) Results of Setting 2. (d) Results of Setting 3.

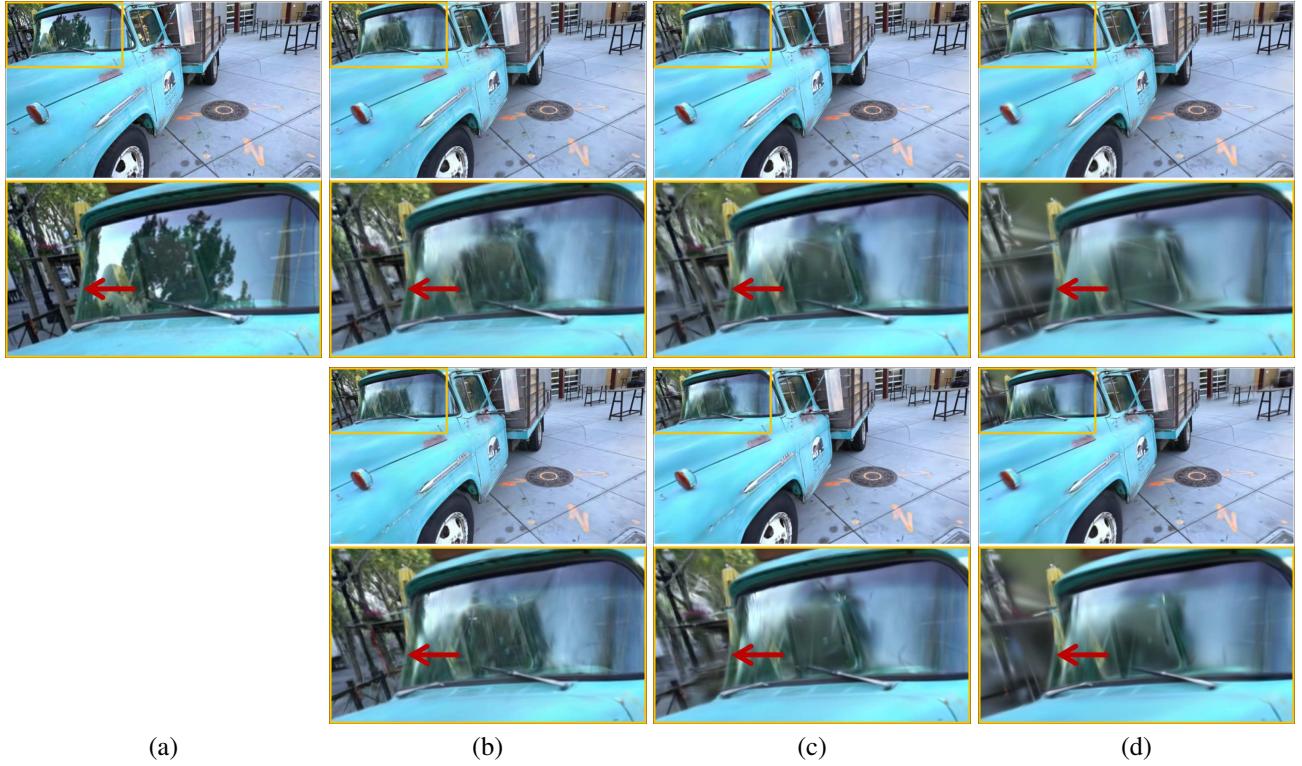


Figure 9. Visual comparison of the rendering quality with different numbers of Gaussians, complementing the quantitative analysis in Fig. 4. Top row: Results from our lightweight model trained with knowledge distillation loss. Bottom row: Results from the baseline model trained only with the ground truth loss (Eq. (4)). (a) Ground truth image; (b-d) Rendered results with progressively decreasing Gaussian counts ((b):85%, (c):20%, (d):3%). Our knowledge distillation approach demonstrates better preservation of visual quality even with reduced Gaussian counts compared to the baseline approach.

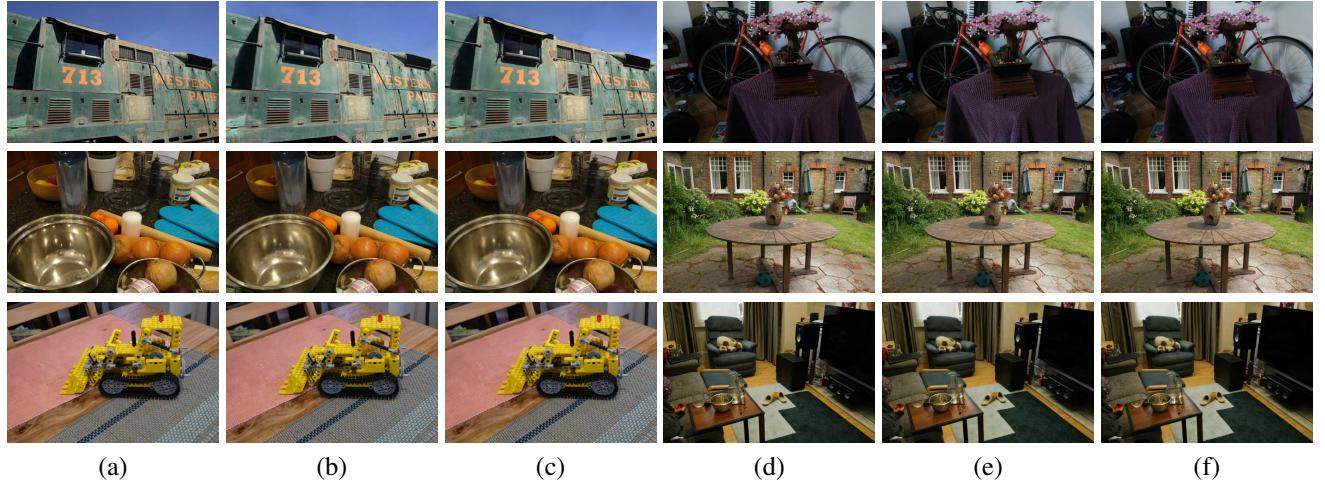


Figure 10. Additional detailed visual comparisons for Fig. 3. (a)(d) Ground truth images from different views. (b)(e) 3DGS, our teacher model with dense Gaussians. (c)(f) 3DGS+KD, our lightweight model trained with knowledge distillation. The zoomed patches show that our model shows comparable visual quality to the teacher 3DGS model.

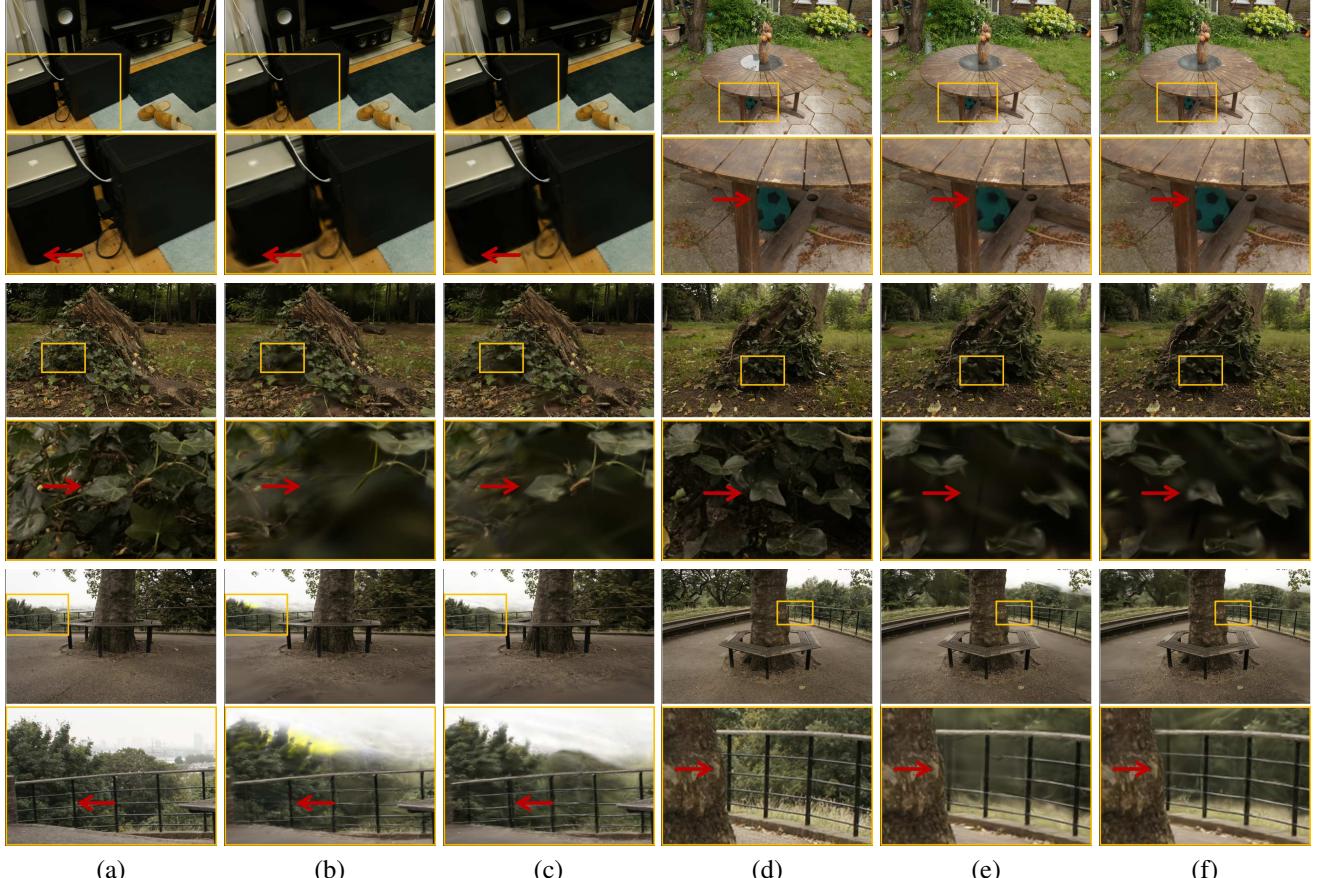


Figure 11. More detailed visual comparisons for Fig. 5. (a)(d) Ground truth images. (b)(e) 3DGS+LW, lightweight model trained with only ground truth loss (Eq. (4)). (c)(f) 3DGS+KD, lightweight model trained with our knowledge distillation loss (Eq. (8)). Lightweight models show quality degradation due to the reduced number of Gaussians; however, our knowledge distillation effectively mitigates this by transferring appearance information from the teacher model. The zoomed patches show that our model shows comparable visual quality to the teacher 3DGS model.

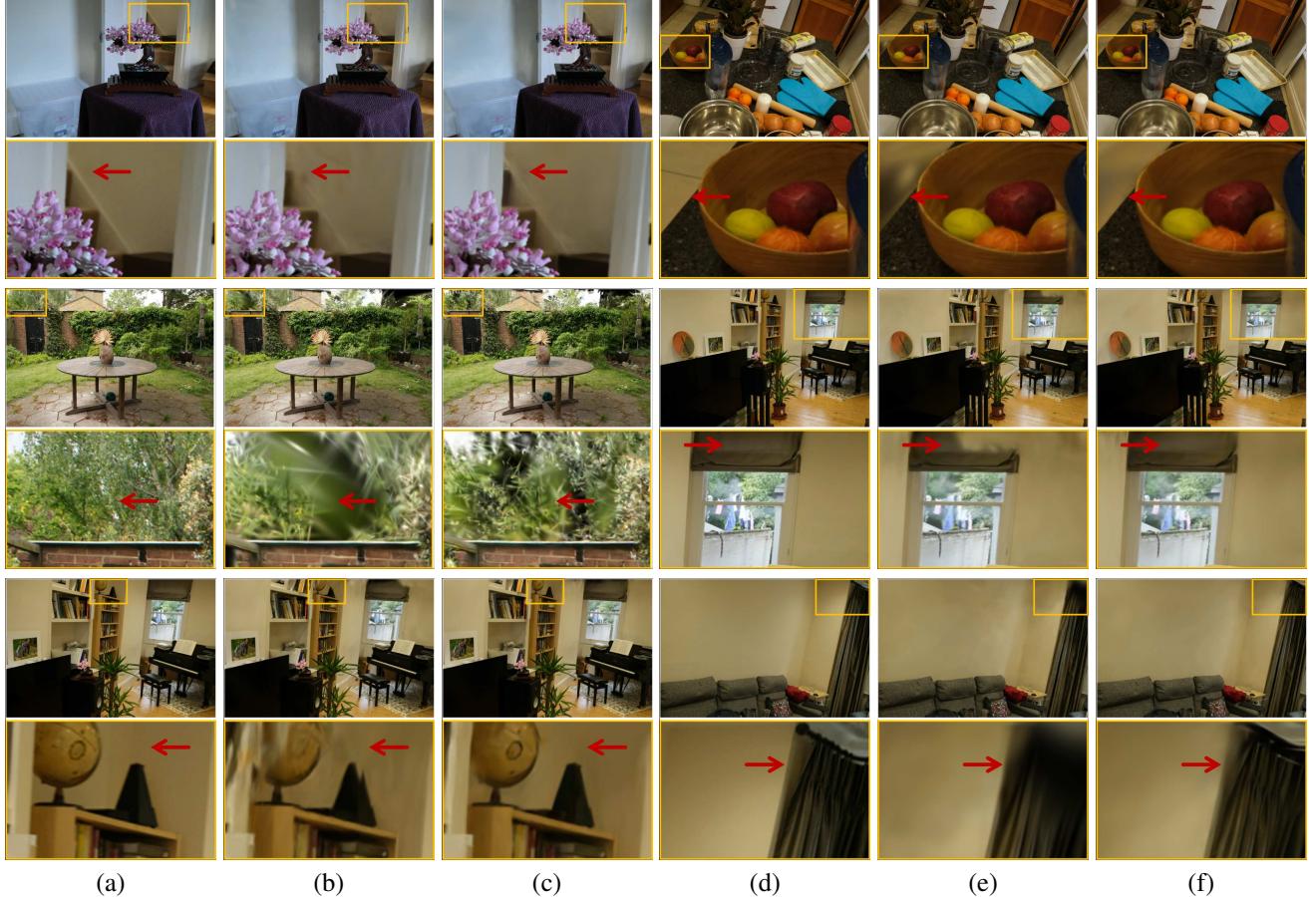


Figure 12. More detailed visual comparisons for Fig. 6. (a)(d) Ground truth images from different views. (b)(e) LightGaussian, our teacher model with dense Gaussians. (c)(f) LightGaussian+KD, our lightweight model trained with knowledge distillation. The zoomed patches show that our model shows comparable visual quality to the teacher 3DGS model.