Reinforcement Learning for View-Adaptive Distillation in 3D Gaussian Compression

Hongji Zhao
Xidian University
zhaohj@stu.xidian.edu.cn

Mingrui Zhu✉
Xidian University
mrzhu@xidian.edu.cn

Xin Wei
Xidian University
weixin@xidian.edu.cn

Nannan Wang
Xidian University
nnwang@xidian.edu.cn

Abstract

3D Gaussian Splatting enables real-time novel view synthesis with high fidelity, yet its irregularly distributed Gaussian primitives and anchors make compression challenging without harming structure and cross-view consistency. A key challenge is that effective compression depends not only on reducing parameter redundancy, but also on identifying where the compressed model is most fragile and how limited representation capacity should be allocated accordingly. We present an RL-guided distillation framework for 3DGS compression (RLGS). RLGS first uses a lightweight reinforcement learning policy to predict informative viewpoint offsets for adaptive distillation. It then distills a compact student 3DGS from a high-quality teacher under multi-view rendering supervision. Finally, it introduces an Unbalanced Optimal Transport based structural selection module to align teacher-student anchor distributions in voxelized space, thereby retaining the most informative anchors during pruning and allocation. Across three benchmarks, RLGS achieves a favorable quality–size trade-off, preserving high rendering fidelity with compact models.

1 Introduction

Novel view synthesis aims to render photorealistic images from unseen viewpoints, with broad applications in augmented reality, virtual reality, digital twins, robotics, and free-viewpoint video. Recent neural radiance field methods have greatly improved synthesis quality by learning continuous scene representations from posed images [1–4]. However, their optimization and inference costs often limit interactive use. 3D Gaussian Splatting (3DGS) instead represents a scene explicitly with anisotropic Gaussians and learnable appearance attributes, enabling real-time rendering with high visual quality [5]. Nevertheless, this explicit representation introduces substantial storage, transmission, and memory overhead, as a typical scene contains many Gaussians with geometry and appearance parameters [6, 7].

Compressing 3DGS is challenging because the representation is sparse, irregular, and highly view-dependent. Unlike regular grid signals such as images or videos, Gaussians are non-uniformly distributed in 3D space and jointly encode position, covariance, scale, opacity, and appearance. As a result, aggressive pruning or quantization can easily damage thin structures, specular regions, and occlusion boundaries. Existing methods have improved 3DGS compression from several perspectives, including scaffold-style structured representations [6], pruning and simplification strategies [7–10],

✉Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and context-based entropy coding with structured signals such as hash-grid features [11, 12]. Despite these advances, many methods still depend on hand-crafted pruning schedules, static capacity allocation, or non-adaptive view supervision, making it difficult to consistently achieve strong quality–size trade-offs, especially in the low-storage regime.

Teacher–student distillation provides a natural way to preserve rendering quality under strong compression, where a compact student is guided by a high-quality teacher [13]. However, applying distillation to 3DGS compression is nontrivial, since compression artifacts are strongly view-dependent. The most useful supervision may not come from the original training views, but from viewpoints that expose regions where the student fails to match the teacher. This motivates an adaptive distillation strategy that can actively seek informative views. Reinforcement learning is well suited to this goal, as it can learn decision policies from optimization feedback and generate training conditions that improve the final objective [14–17].

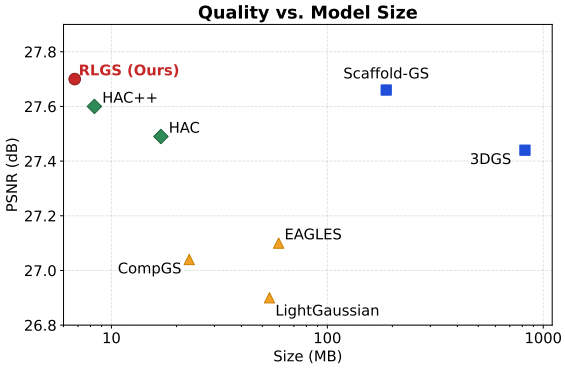


Figure 1: Rate–quality comparison on Mip-NeRF360 under the lowrate setting. Our method achieves a better quality–size trade-off than prior compact 3DGS baselines, especially in the low-storage regime. ■ denotes reference methods, ▲ denotes representative compression baselines, ♦ denotes compact coding methods, and ● denotes our method.

Motivated by this observation, we propose RLGS, a reinforcement learning guided compression framework for 3DGS. Our method learns a lightweight policy to predict continuous viewpoint offsets around reference cameras, producing more informative distillation views for teacher–student training. The compact student is then optimized through multi-view rendering supervision at these perturbed views, allowing it to better inherit the teacher's structural and appearance fidelity under a limited capacity budget. To further preserve scene structure, we introduce an Unbalanced Optimal Transport (UOT) based selection strategy, which aligns teacher and student anchor distributions in voxelized space and guides anchor pruning according to their structural importance. In this way, our framework jointly improves where distillation is applied in view space and how model capacity is allocated in structure space. It is also compatible with context-based entropy coding pipelines such as HAC++ [12], since it improves the compression process without changing the underlying coding scheme.

Our contributions are threefold. First, we introduce an RL-based viewpoint-adaptive distillation mechanism for 3DGS compression, which learns continuous viewpoint perturbations to provide more effective teacher-guided supervision. Second, we propose a UOT-guided structural selection module that aligns teacher and student anchor distributions in voxelized space, enabling more principled pruning than heuristic selection rules. Third, experiments across three benchmarks show that learned view selection and UOT-guided allocation improve the compact student's quality–size trade-off, with modest gains over strong competing methods. Fig. 1 illustrates this trade-off on Mip-NeRF360 under the lowrate setting.

2 Related Work

2.1 3D Compression

The emergence of 3D Gaussian Splatting (3DGS) [5] has promoted extensive research on compact scene representations and efficient compression, as summarized by recent surveys [18–21]. Unlike images, videos, or dense volumetric fields, 3DGS represents a scene with sparse and irregular Gaussian primitives, making redundancy difficult to exploit with conventional grid-based coding tools.

Neural Radiance Field (NeRF) compression provides an important precursor. While NeRF methods improve view synthesis through implicit scene functions [1–4], compact neural field representations further study pruning, codebooks, quantization, entropy constraints, and structured decom-

positions [22–31]. These works show the importance of both parameter modeling and structural regularity, which becomes more challenging for 3DGS due to its unstructured Gaussian distribution.

Existing 3DGS compression methods mainly follow three directions. The first reduces explicit Gaussian redundancy through pruning, clustering, quantization, or compact parameterization [7, 9, 8, 10, 32–35]. The second introduces more structured representations, such as anchor-based Scaffold-GS [6], self-organizing Gaussian grids [36], spectral or graph-inspired pruning [37], and compact spawning or implicit representations [38, 39]. The third focuses on coding-oriented compression with learned context models, including HAC [11], HAC++ [12], ContextGS [40], HEMGS [41], FCGS [42], PCGS [43], and related coding-aware or progressive methods [44–47]. Although these methods have greatly advanced 3DGS compression, most of them still rely on fixed scoring rules, hand-crafted pruning schedules, or static view supervision. Under strong compression, such fixed procedures may be suboptimal because both capacity allocation and teacher supervision need to adapt to the current student representation.

2.2 Adaptive Compression

Knowledge distillation transfers supervision from a stronger teacher to a smaller student [13], and is naturally useful for preserving appearance and geometry in compressed 3D representations. However, 3DGS distillation is highly view-dependent: some views provide mostly redundant supervision, while others reveal thin structures, specular regions, or occlusion boundaries. Therefore, effective compression requires not only matching the teacher representation, but also deciding where informative supervision should be applied.

Optimal transport offers a principled way to align distributions [48, 49]. In particular, unbalanced optimal transport relaxes strict mass preservation and allows partial matching when the support size changes [50], making it suitable for teacher–student compression where the student only retains a compact subset of the teacher representation. Reinforcement learning further provides a framework for adaptive decision making under long-horizon objectives [15, 17, 14, 16]. From this perspective, 3DGS compression can be viewed as a joint decision problem over which structures to preserve and which views should provide the most useful teacher guidance.

Our method combines these ideas by making distillation adaptive through RL-based viewpoint offset control and using UOT to guide teacher-aware structural pruning. Compared with prior methods that mainly improve local pruning, coding efficiency, or entropy modeling, our framework treats 3DGS compression as a learned rate–distortion decision process over both view space and structure space.

3 Methods

3.1 Overview

Our goal is to compress a high-quality 3D Gaussian scene representation into a compact student model while preserving rendering fidelity under a constrained storage budget. As illustrated in Fig. 2, our framework consists of three tightly coupled components: viewpoint offset learning, policy-guided view-adaptive distillation, and UOT-guided structural pruning.

Given a set of posed training images, we first optimize a full-capacity teacher model based on the Scaffold-GS representation. Unlike unstructured Gaussian formulations, Scaffold-GS organizes Gaussian primitives around spatial anchors and predicts their attributes in an anchor-conditioned manner. This representation provides a more stable and compact basis for compression, and both the teacher and the student in our framework follow the same underlying formulation.

The central observation behind our method is that distillation in Gaussian compression is strongly view dependent. Different viewpoints activate different visible anchors and local Gaussian groups, and even small local pose perturbations may substantially change the difficulty of teacher–student alignment. Therefore, instead of relying only on the original training views or on fixed random perturbations, we learn a viewpoint offset policy that transforms a sampled reference camera into a more informative distillation view. The resulting perturbed view is then used to distill the student under teacher guidance, so that the compressed model is trained more frequently on regions where additional corrective signals are most beneficial.

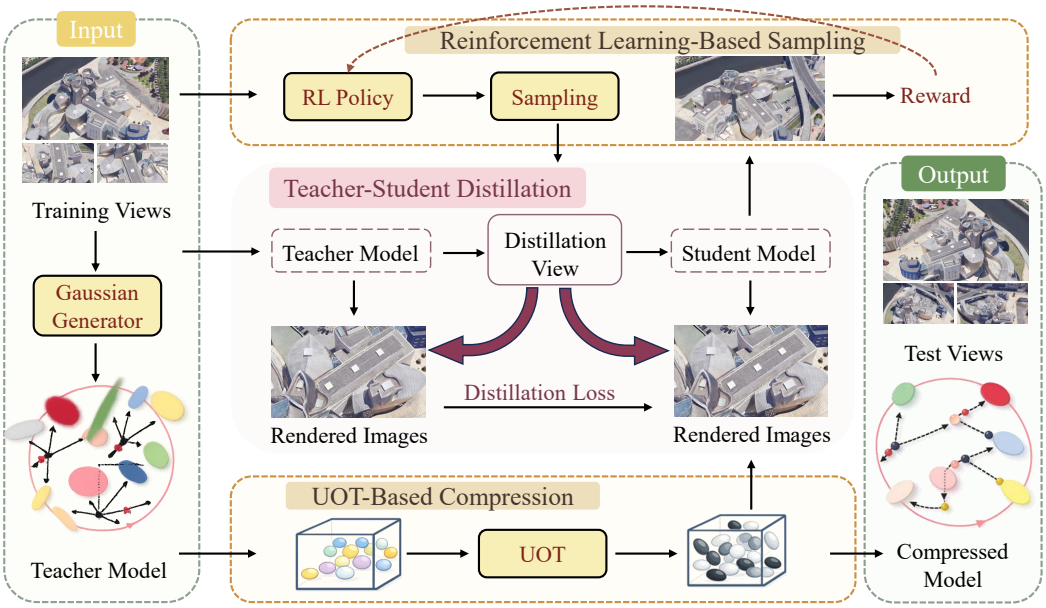


Figure 2: Overview of the proposed framework. Given posed training views, we train a high-quality teacher and a compact student under the anchor-based Scaffold-GS representation. A reinforcement learning policy predicts viewpoint offsets to generate informative distillation views, and the student is optimized through teacher–student rendering distillation. Meanwhile, a UOT-guided structural pruning module aligns teacher and student anchors in voxelized space to allocate the limited anchor budget. Together, these two components improve where distillation is applied in view space and how representation capacity is retained in structure space, leading to a compact model with strong rendering fidelity.

Viewpoint adaptation alone, however, is insufficient to achieve strong compression quality under a limited model budget. The final performance also depends on how the available anchor capacity is distributed across space and which anchors are ultimately preserved. To address this structural aspect, we further introduce a UOT-guided pruning module that periodically aligns the spatial distributions of teacher and student anchors and allocates the student keep budget accordingly. In this way, the proposed framework jointly improves *where* distillation is applied and *how* the compact representation is structurally retained, leading to a student model that better matches the teacher under strict storage constraints.

3.2 RL-Based Viewpoint Offset

A distillation view is not equally informative for compressing Gaussian scene representations: even small pose perturbations around the same reference camera can substantially change the visibility of thin structures, specular regions, and occlusion boundaries, and therefore lead to distillation signals of very different training value. This observation is particularly important in teacher–student compression, where the goal is not merely to expose the student to diverse views, but to expose it to views that produce the most effective distillation updates. Motivated by this, we do not formulate reinforcement learning as a discrete selection problem over a fixed candidate set. Instead, we learn a continuous viewpoint

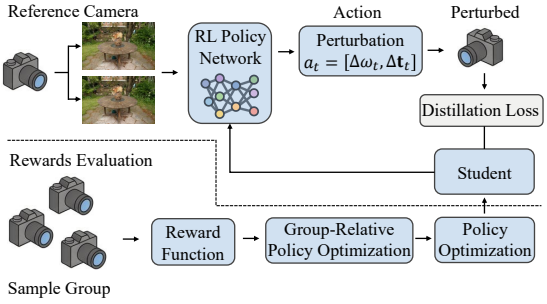


Figure 3: RL-based viewpoint adaptation. The policy perturbs a reference camera for student distillation and is updated using action rewards through group-relative optimization.

offset policy that transforms a sampled reference camera into a more beneficial distillation view. As illustrated in Fig. 3, the policy predicts a local perturbation around a reference camera, the resulting perturbed view is used for teacher–student distillation, and the policy is further updated through reward-driven group-relative optimization.

At training step t, we sample a reference camera

$$v_t = (R_t, \mathbf{t}_t, K_t), \tag{1}$$

where R_t, $\mathbf{t}_t$, and K_t denote the camera rotation, translation, and intrinsic matrix, respectively. The teacher is fixed and denoted by T_ψ, while the compressed student at step t is denoted by S_{ϕ_t}. We first render the reference view from both models and construct the policy state as

$$s_t = \Big[\phi_{\mathrm{err}}(v_t),\ \phi_{\mathrm{pose}}(v_t),\ \phi_{\mathrm{vis}}(v_t),\ \phi_{\mathrm{hist}}(t),\ \phi_{\mathrm{probe}}(v_t)\Big]. \tag{2}$$

Here, $\phi_{\mathrm{err}}(v_t)$ captures the current teacher–student discrepancy at the reference view, $\phi_{\mathrm{pose}}(v_t)$ encodes the camera pose, $\phi_{\mathrm{vis}}(v_t)$ summarizes view-dependent rendering statistics, $\phi_{\mathrm{hist}}(t)$ provides lightweight training context, and $\phi_{\mathrm{probe}}(v_t)$ supplies local directional cues around the current camera. Together, these features allow the policy to reason not only about where the reference camera lies in pose space, but also about how the student currently fails around that view and in which nearby directions distillation is likely to be more beneficial.

We define the action as a continuous local camera perturbation,

$$a_t = \big[\Delta\boldsymbol{\omega}_t,\ \Delta\mathbf{t}_t\big], \tag{3}$$

where $\Delta\boldsymbol{\omega}_t$ and $\Delta\mathbf{t}_t$ denote rotational and translational offsets, respectively. The policy predicts a bounded perturbation, which is mapped through predefined pose scales to a perturbed distillation view $\tilde{v}_t = \mathcal{T}(v_t, a_t)$. In this way, the policy does not decide which training camera to use; rather, it learns how to move a sampled reference camera toward a more effective distillation viewpoint in continuous pose space.

Given the perturbed view $\tilde{v}_t$, we render the teacher and student outputs and optimize the student using the teacher–student distillation loss

$$\mathcal{L}_{\mathrm{dist}} = (1-\lambda_{\mathrm{ssim}})\|\tilde{I}_t^S - \tilde{I}_t^T\|_1 + \lambda_{\mathrm{ssim}}\big(1 - \mathrm{SSIM}(\tilde{I}_t^S, \tilde{I}_t^T)\big). \tag{4}$$

The policy selects where image-space teacher supervision is applied, while the distillation loss encourages the student renderings to match the teacher at the selected views. This loss does not directly reduce anchor count or bitstream size; compactness is controlled by the student capacity, UOT-guided pruning, and the rate objective.

To stabilize exploration, we adopt a group-relative policy optimization strategy. For the same state s_t, we sample a small group of candidate actions, generate the corresponding perturbed views, and evaluate each action with an action-specific reward, as shown in the lower branch of Fig. 3. Rather than relying on the absolute scale of individual rewards, we normalize rewards within each sampled group and optimize the policy using a clipped relative objective with entropy regularization. This design retains the practical stability of group-based relative policy updates while remaining lightweight and well matched to our continuous viewpoint-control setting.

The key design choice lies in the reward. Since the policy is introduced to improve distillation rather than merely to seek visually difficult views, the reward should reflect how much a sampled offset actually benefits teacher–student alignment after the student update. We therefore define a proxy distillation discrepancy at any view v as

$$\mathcal{D}(v;\phi) = (1-\lambda_{\mathrm{ssim}})\|S_\phi(v) - T_\psi(v)\|_1 + \lambda_{\mathrm{ssim}}\big(1 - \mathrm{SSIM}(S_\phi(v), T_\psi(v))\big). \tag{5}$$

Based on this quantity, the reward combines three considerations: the reduction of teacher–student discrepancy at the perturbed view after distillation, a weak preference for informative views, and regularization of the perturbation magnitude. For the g-th candidate action, the reward used in our experiments is

$$r_t^{(g)} = \Delta_{\mathrm{off}}^{(g)} + \lambda_{\mathrm{hard}}\mathcal{R}_{\mathrm{hard}}^{(g)} - \lambda_{\mathrm{mag}}\mathcal{R}_{\mathrm{mag}}(a_t^{(g)}). \tag{6}$$

Here, $\Delta_{\mathrm{off}}^{(g)}$ measures the reduction in teacher–student discrepancy at the perturbed view after the student update. The hardness term mildly favors views that expose informative discrepancies relative to the reference view, while the magnitude penalty discourages unnecessarily large camera perturbations. The post-update improvement is the primary signal: the policy favors offsets that yield useful distillation updates.

3.3 Policy-Guided View-Adaptive Distillation

Given the policy-generated views defined above, we optimize the compact student using teacher-rendered images as distillation targets. The image-space loss improves teacher–student rendering agreement at these views, but does not itself impose a smaller representation. The student capacity and rate objective constrain model size, while UOT-guided pruning determines how the anchor budget is distributed.

The policy conditions its viewpoint offsets on the current teacher–student state, so the views used for distillation can adapt as student training proceeds. This provides complementary supervision to the original training views within the same compact student configuration. We do not assume a prescribed coarse-to-fine progression of selected views; the policy is trained to favor offsets that yield useful post-update alignment gains.

3.4 UOT-Guided Structural Pruning

While policy-guided view-adaptive distillation determines where stronger distillation signals should be delivered, compression quality also depends on how the limited anchor budget is structurally distributed. In our framework, this structural step is handled by a UOT-guided pruning module, which is applied periodically during training rather than being introduced as a dense differentiable loss at every iteration. Concretely, at scheduled pruning stages, we compute a teacher-guided keep mask and directly update the student anchor set. Its goal is to identify which student regions should retain more anchors under teacher guidance, and which regions can be compressed more aggressively.

Let the current student and teacher anchor sets be

$$\mathcal{A}_S = \{\mathbf{g}_i^S\}_{i=1}^N, \qquad \mathcal{A}_T = \{\mathbf{g}_j^T\}_{j=1}^M, \tag{7}$$

where each anchor contains its spatial position and learned anchor feature under the Scaffold-GS representation. Rather than matching individual Gaussians directly, we first assign an importance score to each student anchor according to its accumulated rendering utility and optimization sensitivity. Specifically, for anchor $\mathbf{g}_i^S$, we define

$$\chi_{ik} = \mathbb{I}[\sigma(\tilde{\chi}_{ik}) > 0.01], \qquad \chi_i = \mathbb{I}\left[\sum_k \chi_{ik} > 0\right], \tag{8}$$

where $\tilde{\chi}_{ik}$ is the mask logit of the k-th Gaussian associated with anchor i, and σ is the sigmoid function. The importance score is

$$\rho_i = (s_i^{\mathrm{opa}} + \delta)(s_i^{\mathrm{grad}} + \delta)(0.1 + \chi_i). \tag{9}$$

where s_i^{opa} denotes the opacity-based rendering statistic, s_i^{grad} denotes the accumulated gradient-based sensitivity, χ_i is the current anchor-validity mask, and $\delta > 0$ is a small numerical stabilizer. This score assigns larger weights to anchors that are both more useful in rendering and more responsive to training updates.

We then voxelize the student and teacher anchors into source and target voxel sets. Following the actual implementation, the source and target voxel sizes are initialized from the scene voxel size and scaled by preset multipliers; if the resulting voxel count is excessively large, the voxel size is increased adaptively to keep the matching problem tractable. On the student side, the source mass of each voxel is obtained by aggregating the scores of anchors inside that voxel,

$$\mu_p^S = \sum_{i \in V_p^S} \rho_i. \tag{10}$$

On the teacher side, we construct the target mass using an opacity-volume formulation,

$$\mu_q^T = \sum_{j \in V_q^T} \tilde{\alpha}_j \prod_{d=1}^{3} \tilde{s}_{j,d}, \tag{11}$$

where $\tilde{\alpha}_j$ and $\tilde{s}_{j,d}$ denote the opacity and scaling of teacher anchor j, respectively. When a teacher-validity mask is available, only active teacher anchors are retained in this construction. Intuitively,

this definition assigns larger target mass to teacher regions that are both more visually influential and spatially more significant.

Based on these voxelized source and target distributions, we estimate structural correspondence through entropic unbalanced optimal transport. The transport cost between a student voxel and a teacher voxel is defined as

$$C_{pq} = \|\mathbf{c}_p^S - \mathbf{c}_q^T\|_2^2 + \lambda_f \|\bar{\mathbf{f}}_p^S - \bar{\mathbf{f}}_q^T\|_2^2, \tag{12}$$

where $\mathbf{c}_p^S$ and $\mathbf{c}_q^T$ are voxel centers, $\bar{\mathbf{f}}_p^S$ and $\bar{\mathbf{f}}_q^T$ are voxel-level averaged anchor features, and λ_f controls the contribution of feature similarity. We then solve the following unbalanced transport problem:

$$\mathbf{P}^\star = \arg\min_{\mathbf{P}\geq 0} \left[\langle \mathbf{P}, \mathbf{C} \rangle + \varepsilon \,\mathrm{KL}(\mathbf{P} \,\|\, \exp(-\mathbf{C}/\varepsilon)) + \tau \,\mathrm{KL}(\mathbf{P}\mathbf{1} \,\|\, \boldsymbol{\mu}^S) + \tau \,\mathrm{KL}(\mathbf{P}^\top \mathbf{1} \,\|\, \boldsymbol{\mu}^T) \right]. \tag{13}$$

Here, the transport cost encourages structurally consistent matching, the entropy term stabilizes the transport plan, and the relaxed marginal constraints make the formulation suitable for compression under mismatched teacher and student capacities. In practice, the resulting source-side marginal provides a voxel-level estimate of how strongly each student region should be preserved under the teacher-guided structural prior.

We next convert this continuous structural prior into a discrete pruning decision under a fixed global keep budget. Let

$$K = \lfloor rN \rfloor \tag{14}$$

be the total number of student anchors to keep, where r is the keep ratio and N is the current number of student anchors. According to the transport-induced regional importance, each student voxel is assigned an integer keep quota under exact budget satisfaction. Anchors are then ranked locally by their scores ρ_i, and only the top anchors are retained within each voxel. This two-stage design is important: UOT determines how the budget should be spatially distributed across regions, while score-based local selection determines which anchors survive inside each retained region.

Therefore, UOT serves in our framework as a teacher-guided structural pruning operator rather than a standalone reconstruction objective. It is invoked periodically to update the student topology, while the standard reconstruction and distillation objectives continue to optimize the surviving student representation between pruning stages. In this way, UOT-guided structural pruning complements policy-guided view-adaptive distillation: the latter improves where informative distillation is delivered in view space, while the former improves how the compact representation is preserved in structure space.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of the proposed method for compressing 3D Gaussian scene representations. Our evaluation focuses on three questions: (i) whether the proposed framework improves the quality–size trade-off on standard 3DGS compression benchmarks, (ii) whether each component contributes consistently to the final performance, and (iii) how view-adaptive distillation and UOT-guided pruning affect the behavior of the compressed representation.

4.1 Experimental Setup

Datasets and metrics. We evaluate our method on three widely used real-scene benchmarks: Mip-NeRF360 [3], Tanks&Temples [51], and Deep Blending [52]. These datasets cover indoor and outdoor scenes with diverse geometry, occlusion, and appearance complexity, providing a suitable testbed for 3DGS compression. We report PSNR, SSIM, and LPIPS for rendering quality, together with the reported storage size. For RLGS, size refers to the final entropy-coded bitstream; the storage formats reported by other methods may differ.

Representation and implementation. Both the teacher and the student are built on Scaffold-GS [6]. The teacher is trained in the full-capacity setting, while the student adopts the same representation and is progressively compressed during training. The anchor feature dimensions are set to $m = 50$ and $k = 10$. We train the model for $40{,}000$ iterations using Adam with mixed-precision training. The viewpoint offset policy is implemented as a lightweight MLP and optimized with a group-relative

policy update. The transport-based matching is solved with an entropic-regularized Sinkhorn-style solver. For the complete RLGS configuration, we follow HAC++ [12] in using learned entropy modeling during optimization and entropy coding to produce the final bitstream. The lowrate and highrate operating points use $\lambda = 4 \times 10^{-3}$ and $\lambda = 1 \times 10^{-3}$, respectively; the other reported architecture, policy, pruning, training, and entropy-backend settings are unchanged between these two operating points. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

Training cost. On a single NVIDIA RTX 4090, RLGS requires approximately 90 minutes of offline training per scene on average across Mip-NeRF360, Tanks&Temples, and Deep Blending, with average peak GPU memory usage of approximately 14 GB. The main computational costs arise from teacher–student rendering at candidate perturbed views and policy optimization; voxel-level UOT is solved only at scheduled pruning stages. The teacher, viewpoint policy, and UOT solver are not included in the final compressed model and incur no overhead during rendering.

Baselines. We compare our method with both uncompressed Gaussian rendering methods and representative 3DGS compression approaches, including 3DGS [5], Scaffold-GS [6], Mini-Splatting [38], CompGS [44], Compact-3DGS [9], C3DGS [7], LightGaussian [8], EAGLES [10], SOG [36], HAC [11], LocoGS [45], HAC++ [12], and Smol-GS [46]. These baselines span pruning-based, quantization-based, entropy-modeling-based, and locality-aware compact 3DGS methods, providing a broad comparison across different compression paradigms.

Table 1: Quantitative comparison on Mip-NeRF360, Tanks&Temples, and Deep Blending. We report PSNR, SSIM, LPIPS, and the storage size reported for each method; RLGS size is the final entropy-coded bitstream. Rankings are computed only among the methods below the horizontal rule, excluding the first three reference methods. The best and second-best results are marked in bold and with underlining, respectively.

Method	Mip-NeRF360				Tanks&Temples				Deep Blending			
	PSNR ↑	SSIM ↑	LPIPS ↓	Size (MB) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Size (MB) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Size (MB) ↓
3DGS [5]	27.44	0.813	0.218	822.6	23.67	0.844	0.179	452.4	29.48	0.900	0.246	692.5
Scaffold-GS [6]	27.66	0.812	0.223	187.3	24.11	0.855	0.165	154.3	30.28	0.907	0.243	121.2
Mini-Splatting [38]	27.39	0.822	0.216	119.5	23.41	0.846	0.180	67.6	30.04	0.910	0.241	124.9
CompGS [44]	27.04	0.804	0.243	22.93	23.29	0.835	0.201	14.23	29.89	0.907	0.253	15.15
Compact-3DGS [9]	26.95	0.797	0.244	26.31	23.33	0.831	0.202	18.97	29.71	0.901	0.257	21.75
C3DGS [7]	27.09	0.802	0.237	29.98	23.52	0.837	0.188	18.58	29.53	0.899	0.254	24.96
LightGaussian [8]	26.90	0.800	0.240	53.96	23.32	0.829	0.204	29.94	29.12	0.895	0.262	45.25
EAGLES [10]	27.10	0.807	0.234	59.49	23.14	0.833	0.203	30.18	29.72	0.906	**0.249**	54.45
SOG [36]	27.01	0.800	**0.226**	43.77	23.54	0.833	0.188	24.42	29.21	0.891	0.271	19.32
HAC [11]	27.49	0.807	0.236	16.95	24.08	0.846	0.186	8.42	29.99	0.902	0.268	4.51
LocoGS [45]	27.04	0.806	0.232	7.90	23.63	0.847	**0.169**	6.59	30.06	0.904	**0.249**	7.64
HAC++ [12]	27.60	0.803	0.253	8.34	24.23	0.847	0.191	5.34	30.24	0.908	0.266	3.31
Smol-GS [46]	27.76	0.807	0.248	**5.86**	24.25	0.843	0.199	4.78	30.12	0.905	0.262	2.86
Ours RLGS (lowrate)	27.63	0.803	0.251	6.96	24.26	0.845	0.193	**4.33**	30.26	0.906	0.266	**2.74**
Ours RLGS (highrate)	**27.81**	**0.810**	0.231	16.72	**24.32**	**0.850**	0.181	7.85	**30.32**	**0.910**	0.256	4.59

4.2 Results Analysis

Quantitative comparison. Table 1 reports the results on Mip-NeRF360, Tanks&Temples, and Deep Blending. Overall, our method achieves a favorable quality–size trade-off across all three datasets, maintaining competitive rendering quality in a compact storage regime. It is especially effective on Tanks&Temples and Deep Blending, where it attains smaller model sizes than strong recent baselines while preserving comparable or better reconstruction quality. Compared with LocoGS [45] and HAC++ [12], our method consistently improves PSNR and substantially reduces storage on the latter two benchmarks. Although some baselines show slightly better SSIM or LPIPS on individual datasets, our method delivers a more balanced performance under strict compression budgets.

Rate–quality trade-off. Fig. 1 further visualizes this trend. Our method lies in a favorable region of the rate–quality plot, showing that it makes more effective use of the available storage budget than baselines. This advantage comes from the combination of policy-guided view-adaptive distillation and UOT-guided pruning: the former improves the usefulness of teacher supervision, while the latter preserves model capacity in structurally important regions. Together, they enable more effective compression.

Qualitative comparison. The visual results in Fig. 4 are consistent with the quantitative findings. In the *truck* scene, our method better preserves background structures and local appearance consistency. In the *room* scene, it more faithfully reconstructs thin leaves and fine foreground details. These



Figure 4: Qualitative comparison on representative scenes under the lowrate setting. Our method better preserves thin structures, background consistency, and fine appearance details under strong compression.

comparisons show that our method is especially effective at retaining visually important content that is often degraded under strong compression.

4.3 Ablation Studies

Table 2 reports the module ablation on all three benchmarks. Orig. uses only the original views, Rand. replaces the learned viewpoint policy with fixed random perturbations, and Heur. replaces UOT-guided allocation with heuristic pruning. The full method obtains the highest PSNR among these variants on each dataset. Its gains are more pronounced relative to original-view training than relative to fixed random perturbations, especially on Tanks&Temples. The heuristic-pruning variant is also competitive on some metrics, but the full method gives the smallest final bitstream on Tanks&Temples and Deep Blending. Because the bitstream sizes are close but not identical, these quality differences should be considered together with size, rather than treated as strictly equal-rate comparisons. Random perturbation provides a useful non-learned baseline; the separate comparison below further tests whether a learned policy adds benefit beyond evaluating multiple random candidates.

Comparison with non-RL view selection. The loss-based sampler prioritizes views by discrepancy; Random-4 evaluates four offsets within the RLGS bounds. Table 3 shows competitive baselines. RLGS reduces size from 7.15 to 6.96 MB with small quality gains, suggesting a modest benefit from student-conditioned proposals.

Table 2: Module ablation on three benchmarks under the lowrate setting. Size denotes the final bitstream size in MB.

Variant	PSNR↑	SSIM↑	LPIPS↓	Size↓
Mip-NeRF360				
RLGS (full)	27.63	0.803	0.251	6.96
Original-view	27.44	0.799	0.259	6.93
Random VP	27.55	0.802	0.254	7.14
Heuristic pruning	27.59	0.803	0.255	7.07
Tanks&Temples				
RLGS (full)	24.26	0.845	0.193	4.33
Original-view	24.10	0.838	0.199	4.35
Random VP	24.23	0.842	0.198	4.41
Heuristic pruning	24.21	0.841	0.203	4.36
Deep Blending				
RLGS (full)	30.26	0.906	0.266	2.74
Original-view	29.99	0.902	0.272	2.74
Random VP	30.17	0.905	0.268	2.85
Heuristic pruning	30.21	0.901	0.272	2.77

Table 3: Comparison with non-RL view selection on Mip-NeRF360 under the lowrate setting. Size denotes the final bitstream size in MB.

Method	PSNR↑	SSIM↑	LPIPS↓	Size↓
Loss-based hard sampler	27.53	0.802	0.254	7.19
Random best-of-4	27.57	0.802	0.252	7.15
RLGS	27.63	0.803	0.251	6.96

Table 4: Joint sensitivity to rotation and translation bounds on Deep Blending. The default bounds are 4° and 0.2.

Rotation	Translation	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓
2°	0.1	30.22	0.907	0.266	2.72
4°	0.2	30.26	0.906	0.266	2.74
6°	0.3	30.26	0.903	0.268	2.79

Table 5: Effect of the number of candidate actions on Deep Blending. Other settings are held fixed.

Group size	PSNR↑	SSIM↑	LPIPS↓	Time (min)↓
2	30.22	0.905	0.268	80
4	30.26	0.906	0.266	90
8	30.28	0.906	0.265	120

4.4 Policy Sensitivity and Training Cost

Candidate action group. Increasing the group size from 4 to 8 improves PSNR by only 0.02 dB while increasing training time from 90 to 120 minutes (33.3%). Increasing the group size from 2 to 4 improves PSNR by 0.04 dB and LPIPS by 0.002 at an additional training cost of 10 minutes. We therefore use four candidate actions as a practical quality–cost trade-off, not as a universally optimal choice.

Perturbation range. Table 4 examines the bounds of local viewpoint exploration. We vary the two bounds jointly while keeping the policy, reward, pruning configuration, training schedule, and compression target fixed. The default setting matches the highest PSNR in the tested range, while the smaller range remains competitive. The larger range provides no additional PSNR gain and slightly worsens SSIM, LPIPS, and model size. Since the bounds change together, this experiment does not isolate their individual effects.

Interpreting the quality–cost trade-off. Group size controls how many offsets are evaluated, whereas pose bounds define the exploration region. Table 5 shows diminishing returns: doubling the default group adds 30 minutes for 0.02 dB higher PSNR, unchanged SSIM, and 0.001 lower LPIPS. Candidate evaluation is only required during training and adds no rendering overhead to the final student.

PSNR alone does not determine the preferred perturbation range. The default and larger ranges both attain 30.26 dB, but the larger range worsens SSIM, LPIPS, and size. The smaller range instead gives higher SSIM and a smaller bitstream at lower PSNR. Thus, the default balances the reported metrics without dominating every alternative. These separate studies on Deep Blending neither characterize interactions between group size and pose bounds nor establish a scene-independent optimum; joint and scene-level evaluations would be needed for broader conclusions. The small metric differences should also be interpreted cautiously, since these sensitivity tables do not report uncertainty for each configuration.

5 Conclusion and Limitations

In this paper, we presented a reinforcement learning guided framework for view-adaptive distillation in 3D Gaussian compression. By combining viewpoint offset learning with UOT-guided structural pruning, our method improves both the selection of informative distillation views and the allocation of limited anchor capacity. This leads to a better balance between rendering fidelity and representation compactness across multiple benchmarks.

Despite its effectiveness, the framework still involves manually designed choices, such as the perturbation range and pruning schedule, which may not generalize optimally to all scenes or compression settings. Future work could further improve adaptability through automatic schedule learning or data-dependent parameter prediction.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Program No. 62576261, U22A2096, and 62506280), in part by the Natural Science Basic Research Program of Shaanxi (Program No. 2026SYS-SYSZD-105), in part by the Key Research and Development Program of Shaanxi (Program No. 2026JC-ZDGG-42), in part by the Scientific and Technological Innovation Teams in Shaanxi Province (Program No. 2025RS-CXTD-011), in part by the Shaanxi Province Core Technology Research and Development Project (Program No. 2024QY2-GJHX-11), in part by the Establishment Fund of the State Key Laboratory of Wakefullness/Sleep and Cognition (Chongqing Institute for Brain and Intelligence), in part by the Fundamental Research Funds for the Central Universities (Program No. QTZX26138).

References

[1] 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. *Communications of the ACM*, 65 (1):99–106, 2021.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] 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.

[4] 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.

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

[6] Tao Lu, Mulin Yu, Linning 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.

[7] 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*, pages 10349–10358, 2024.

[8] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in Neural Information Processing Systems*, 37:140138–140158, 2024.

[9] 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*, pages 21719–21728, 2024.

[10] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024.

[11] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024.

[12] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac++: Towards 100x compression of 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[14] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[15] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[17] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[18] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.

[19] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[20] Milena T Bagdasarian, Paul Knoll, Y Li, Florian Barthel, Anna Hilsmann, Peter Eisert, and Wieland Morgenstern. 3dgs. zip: A survey on 3d gaussian splatting compression methods. In *Computer Graphics Forum*, volume 44, page e70078. Wiley Online Library, 2025.

[21] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3d gaussian splatting. *Computational Visual Media*, 10(4):613–642, 2024.

[22] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Liefeng Bo. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4222–4231, 2023.

[23] Chenxi Lola Deng and Enzo Tartaglione. Compressing explicit voxel grid representations: fast nerfs become also small. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1236–1245, 2023.

[24] Lingzhi Li, Zhongshu Wang, Zhen Shen, Li Shen, and Ping Tan. Compact real-time radiance fields with neural codebook. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2189–2194. IEEE, 2023.

[25] Seungjoo Shin and Jaesik Park. Binary radiance fields. *Advances in Neural Information Processing Systems*, 36:55919–55931, 2023.

[26] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17513–17524, 2023.

[27] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20680–20690, 2023.

[28] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Compressible-composable nerf via rank-residual decomposition. *Advances in Neural Information Processing Systems*, 35:14798–14809, 2022.

[29] Zetian Song, Jiaqi Zhang, Wenhong Duan, Yuhuai Zhang, Xinfeng Zhang, and Siwei Ma. Spc-nerf: Spatial predictive compression for voxel-based radiance field. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.

[30] Yihang Chen, Qianyi Wu, Mehrtash Harandi, and Jianfei Cai. How far can we compress instant-ngp-based nerf? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2024.

[31] Sicheng Li, Hao Li, Yiyi Liao, and Lu Yu. Nerfcodec: Neural feature compression meets neural radiance fields for memory-efficient scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21274–21283, 2024.

[32] Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1):1–17, 2024.

[33] Muhammad Salman Ali, Maryam Qamar, Sung-Ho Bae, and Enzo Tartaglione. Trimming the fat: Efficient compression of 3d gaussian splats through pruning. *arXiv preprint arXiv:2406.18214*, 2024.

[34] Henan Wang, Hanxin Zhu, Tianyu He, Runsen Feng, Jiajun Deng, Jiang Bian, and Zhibo Chen. End-to-end rate-distortion optimized 3d gaussian representation. In *European Conference on Computer Vision*, pages 76–92. Springer, 2024.

[35] 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*, pages 434–452. Springer, 2024.

[36] Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *European Conference on Computer Vision*, pages 18–34. Springer, 2024.

[37] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation. *arXiv preprint arXiv:2405.00676*, 2024.

[38] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024.

[39] Minye Wu and Tinne Tuytelaars. Implicit gaussian splatting with efficient multi-level tri-plane representation. *arXiv preprint arXiv:2408.10041*, 2024.

[40] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *Advances in Neural Information Processing Systems*, 37:51532–51551, 2024.

[41] Lei Liu, Zhenghao Chen, Wei Jiang, Wei Wang, and Dong Xu. Hemgs: A hybrid entropy model for 3d gaussian splatting data compression. *arXiv preprint arXiv:2411.18473*, 2024.

[42] Yihang Chen, Qianyi Wu, Mengyao Li, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Fast feedforward 3d gaussian splatting compression. *arXiv preprint arXiv:2410.08017*, 2024.

[43] Yihang Chen, Mengyao Li, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Pcgs: Progressive compression of 3d gaussian splatting. *arXiv preprint arXiv:2503.08511*, 2025.

[44] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compgs: Smaller and faster gaussian splatting with vector quantization. In *European Conference on Computer Vision*, pages 330–349. Springer, 2024.

[45] Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware gaussian compression for fast and high-quality rendering. *arXiv preprint arXiv:2501.05757*, 2025.

[46] Haishan Wang, Mohammad Hassan Vali, and Arno Solin. Smol-gs: Compact representations for abstract 3d gaussian splatting. *arXiv preprint arXiv:2512.00850*, 2025.

[47] Soonbin Lee, Fangwen Shu, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. Compression of 3d gaussian splatting with optimized feature planes and standard video codecs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25496–25505, 2025.

[48] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[49] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

[50] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

[51] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[52] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

[53] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024.

[54] Yangming Zhang, Wenqi Jia, Wei Niu, and Miao Yin. Gaussianspa: An" optimizing-sparsifying" simplification framework for compact and high-quality 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26673–26682. IEEE, 2025.

A Full Pipeline Description

Our framework follows a teacher–student compression paradigm. Given a set of posed training images, we first train a full-capacity teacher model using the anchor-based Scaffold-GS representation. The student adopts the same underlying representation but is progressively compressed during training. Instead of relying only on the original training views, we introduce a lightweight reinforcement learning policy that predicts informative viewpoint offsets around sampled reference cameras. The resulting perturbed views are used to render teacher and student outputs, and the student is optimized through teacher-guided multi-view distillation. Meanwhile, at scheduled pruning stages, a UOT-guided structural pruning module aligns teacher and student anchor distributions in voxelized space and allocates the limited keep budget accordingly. In this way, the overall framework is adaptive in both view space and structure space.

More specifically, RLGS improves compression from two complementary perspectives. In view space, the policy searches for locally perturbed views that are more informative for distillation than the original training views alone. In structure space, UOT estimates how the student keep budget should be distributed across regions under teacher guidance, while local anchor ranking determines which anchors are ultimately retained within each region. Together, these two components improve the quality–size trade-off under compact storage budgets, especially in the low-rate regime.

B RL-Based Viewpoint Offset

B.1 State Construction

At training step t, we sample a reference camera

$$v_t = (R_t, t_t, K_t), \tag{15}$$

where R_t, t_t, and K_t denote the camera rotation, translation, and intrinsic matrix. Given the fixed teacher T_ψ and the current student S_{ϕ_t}, we first render the reference view and construct the policy state as

$$s_t = [\phi_{\mathrm{err}}(v_t),\ \phi_{\mathrm{pose}}(v_t),\ \phi_{\mathrm{vis}}(v_t),\ \phi_{\mathrm{hist}}(t),\ \phi_{\mathrm{probe}}(v_t)]\,. \tag{16}$$

In our implementation, these terms are instantiated using compact numerical statistics. Concretely, the state includes student-versus-ground-truth discrepancy, teacher-versus-student discrepancy, pose features derived from the camera center, visibility-related rendering statistics, lightweight history signals, and six directional probe losses around the current base camera. This design allows the policy to reason not only about where the reference camera lies in pose space, but also about how the compressed student currently fails near that view and in which nearby directions additional distillation is likely to be most beneficial.

B.2 Action Parameterization

Rather than selecting from a discrete set of candidate views, we predict a continuous local perturbation

$$a_t = [\Delta\omega_t,\ \Delta\mathbf{t}_t], \tag{17}$$

where $\Delta\omega_t$ denotes a rotational offset and $\Delta\mathbf{t}_t$ denotes a translational offset. In the implementation, the raw action is six-dimensional, with the first three dimensions controlling rotation and the last three controlling translation. The action is bounded through a $\tanh$ squashing function and mapped through predefined pose scales to obtain the perturbed distillation view

$$\tilde{v}_t = \mathcal{T}(v_t, a_t). \tag{18}$$

This formulation is important for two reasons. First, it avoids the need to pre-enumerate a large candidate set of cameras. Second, it better matches the local nature of view-dependent rendering difficulty: even small perturbations around the same reference camera can substantially change visibility, occlusion, and local appearance.

B.3 Distillation Loss

Given the perturbed view $\tilde{v}_t$, the teacher and student are rendered and the student is optimized by

$$\mathcal{L}_{\mathrm{dist}} = (1-\lambda_{\mathrm{ssim}})\|\tilde{I}_t^S - \tilde{I}_t^T\|_1 + \lambda_{\mathrm{ssim}}\left(1 - \mathrm{SSIM}(\tilde{I}_t^S, \tilde{I}_t^T)\right). \tag{19}$$

This is the same image-space teacher–student distillation objective used in the main paper. The policy determines the views at which this image-space loss is applied. The loss supervises rendering fidelity, whereas the student budget, pruning procedure, and rate objective control compactness.

B.4 Group-Relative Policy Optimization

To stabilize exploration, we adopt a group-relative optimization strategy. For the same state s_t, we sample a small group of candidate actions, generate the corresponding perturbed views, and compute an action-specific reward for each candidate. The rewards are normalized within the group, and the policy is updated using a clipped relative objective together with entropy regularization.

The motivation is that absolute reward values may vary substantially across training stages, camera poses, and scenes. By normalizing rewards within each action group, the policy update depends primarily on the relative quality of candidate offsets under the same local context, which improves stability and reduces sensitivity to reward scale.

B.5 Reward Design

We define the proxy distillation discrepancy at a view v as

$$D(v;\phi) = (1-\lambda_{\text{ssim}})\|S_\phi(v) - T_\psi(v)\|_1 + \lambda_{\text{ssim}}\left(1 - \text{SSIM}(S_\phi(v), T_\psi(v))\right). \tag{20}$$

For each action sampled in the group, the reward used in the reported experiments is

$$r_t^{(g)} = \Delta_{\text{off}}^{(g)} + \lambda_{\text{hard}}\mathcal{R}_{\text{hard}}^{(g)} - \lambda_{\text{mag}}\mathcal{R}_{\text{mag}}(a_t^{(g)}). \tag{21}$$

The first term evaluates the decrease in $\mathcal{D}$ at the perturbed view following the student update. The hardness bonus favors informative teacher–student discrepancies relative to the reference view, and the action-magnitude term limits unnecessarily large offsets. We set $\lambda_{\text{hard}} = 0.15$ and $\lambda_{\text{mag}} = 0.01$ in all reported experiments.

C UOT-Guided Structural Pruning

C.1 Anchor Scoring

Let the current student and teacher anchor sets be

$$\mathcal{A}_S = \{\mathbf{g}_i^S\}_{i=1}^N, \qquad \mathcal{A}_T = \{\mathbf{g}_j^T\}_{j=1}^M. \tag{22}$$

For each student anchor $\mathbf{g}_i^S$, we define an importance score. The anchor-validity mask is obtained from the mask logits as

$$\chi_{ik} = \mathbb{I}[\sigma(\tilde{\chi}_{ik}) > 0.01], \qquad \chi_i = \mathbb{I}\left[\sum_k \chi_{ik} > 0\right], \tag{23}$$

where $\tilde{\chi}_{ik}$ is the mask logit of the k-th Gaussian associated with anchor i, and σ is the sigmoid function. The importance score is

$$\rho_i = (s_i^{\text{opa}} + \delta)(s_i^{\text{grad}} + \delta)(0.1 + \chi_i), \tag{24}$$

where s_i^{opa} is the opacity-related rendering statistic, s_i^{grad} is the accumulated gradient-based sensitivity, χ_i is the anchor-validity mask, and $\delta > 0$ is a small numerical stabilizer. This score assigns larger weights to anchors that are both visually useful and responsive to optimization.

C.2 Voxelization and Mass Construction

Instead of directly matching individual anchors, we first voxelize the student and teacher anchors into source and target voxel sets. The voxel size is initialized from the scene voxel size and scaled by preset multipliers. If the resulting number of voxels becomes too large, the voxel size is increased adaptively to keep the matching problem tractable.

The student-side source mass for voxel p is defined as

$$\mu_p^S = \sum_{i \in V_p^S} \rho_i. \tag{25}$$

The teacher-side target mass for voxel q is defined through an opacity–volume formulation

$$\mu_q^T = \sum_{j \in V_q^T} \tilde{\alpha}_j \prod_{d=1}^{3} \tilde{s}_{j,d}, \tag{26}$$

where $\tilde{\alpha}_j$ and $\tilde{s}_{j,d}$ are the teacher anchor opacity and scale. When a teacher-validity mask is available, only active teacher anchors are retained. This construction assigns larger target mass to teacher regions that are both visually influential and spatially significant.

C.3 Transport Cost and UOT Objective

The transport cost between student voxel p and teacher voxel q is

$$C_{pq} = \|\mathbf{c}_p^S - \mathbf{c}_q^T\|_2^2 + \lambda_f \|\bar{\mathbf{f}}_p^S - \bar{\mathbf{f}}_q^T\|_2^2, \tag{27}$$

where $\mathbf{c}_p^S$ and $\mathbf{c}_q^T$ are voxel centers, $\bar{\mathbf{f}}_p^S$ and $\bar{\mathbf{f}}_q^T$ are voxel-level averaged features, and λ_f controls the weight of feature similarity.

Let $\boldsymbol{\mu}^S = (\mu_p^S)_p$ and $\boldsymbol{\mu}^T = (\mu_q^T)_q$. We then solve the entropic unbalanced optimal transport problem

$$\mathbf{P}^\star = \arg\min_{\mathbf{P}\geq 0} \left[\langle \mathbf{P}, \mathbf{C} \rangle + \varepsilon \,\mathrm{KL}(\mathbf{P} \,\|\, \exp(-\mathbf{C}/\varepsilon)) + \tau \,\mathrm{KL}(\mathbf{P}\mathbf{1} \,\|\, \boldsymbol{\mu}^S) + \tau \,\mathrm{KL}(\mathbf{P}^\top \mathbf{1} \,\|\, \boldsymbol{\mu}^T) \right]. \tag{28}$$

This objective combines three effects. The transport cost encourages spatially and structurally consistent matching, the entropy term stabilizes the transport plan, and the relaxed marginal constraints allow partial matching, which is particularly suitable for compression because teacher and student capacities are inherently mismatched.

C.4 Budget Allocation and Discrete Pruning

After solving UOT, we convert the continuous transport prior into a discrete keep decision. Let

$$K = \lfloor rN \rfloor \tag{29}$$

be the total number of anchors to keep, where r is the keep ratio.

Each student voxel is assigned an integer keep quota according to the UOT-induced regional importance. Within each voxel, anchors are ranked by ρ_i, and only the top anchors are retained. This two-stage design is important: UOT determines *where* the limited budget should be distributed, while local ranking determines *which* anchors should survive inside each region.

D Training Details

Both the teacher and the student are based on Scaffold-GS. The teacher is trained in the full-capacity regime, while the student is progressively compressed during training. Following the main paper, the anchor feature dimensions are set to $m = 50$ and $k = 10$, and the model is trained for 40,000 iterations using Adam with mixed-precision training. The viewpoint offset policy is implemented as a lightweight MLP and trained jointly with the student through group-relative policy updates. The transport problem is solved with an entropic-regularized Sinkhorn-style solver. In the complete RLGS configuration, the HAC++-based entropy model contributes to the rate objective during training, and the final encoder produces the compressed bitstream after training. This backend is shared by the lowrate and highrate RLGS settings.

In practice, training alternates between dense student optimization and scheduled structural updates. Teacher–student distillation is injected at a predefined interval, while UOT-guided pruning is invoked only at designated pruning stages. This design keeps the structural update informative while avoiding the cost of solving transport at every iteration. A concise summary of the main hyperparameters is provided below.

E Hyperparameter Settings

We report only the principal hyperparameters that directly control the optimization objective, the viewpoint-adaptation module, and the UOT-based pruning process. Both the teacher and the student are built on Scaffold-GS. Following the main paper, the anchor feature dimensions are set to $m = 50$ and $k = 10$. The student is trained for 40,000 iterations using Adam with mixed-precision training on a single NVIDIA RTX 4090 GPU.

In the overall objective, the rate weight is $\lambda = 4 \times 10^{-3}$ for lowrate and $\lambda = 1 \times 10^{-3}$ for highrate; the distillation weight is $\lambda_{\mathrm{dist}} = 0.05$ in both settings. The distillation branch is activated every 2 iterations and uses one distillation view at each activation by default. For the distillation loss in Eq. (19), we set the structural similarity coefficient to $\lambda_{\mathrm{ssim}} = 0.1$.

For viewpoint offset learning, the bounded action in Eq. (17) is mapped to a maximum rotational offset of 4.0° and a maximum translational offset of 0.2. In the group-relative policy update, the group size is set to 4, the clipping coefficient is set to 0.2, and the entropy regularization coefficient is set to 0.01. For the reward in Eq. (21), the default implementation mainly instantiates the direct post-update improvement term together with a hardness bonus and an action-magnitude penalty. Accordingly, we set $\lambda_{\mathrm{hard}} = 0.15$ and $\lambda_{\mathrm{mag}} = 0.01$.

For the transport cost in Eq. (27), the feature-matching weight is set to $\lambda_f = 0.05$ in the default setting. For the entropic unbalanced optimal transport objective in Eq. (28), we set the entropic regularization coefficient

to $\varepsilon = 0.05$ and the marginal relaxation coefficient to $\tau = 1.0$. For the discrete keep-budget formulation in Eq. (29), the keep ratio is set to $r = 0.95$. In addition, the source and target voxel multipliers are set to 1 and 2, respectively, and the Sinkhorn solver is run for 60 iterations.

F Complexity and Practical Considerations

The additional computational overhead of RLGS mainly comes from two sources: policy evaluation at perturbed views and periodic UOT matching. The RL component is lightweight because the policy network is a small MLP and operates on compact state features rather than raw images. Its main runtime cost comes from rendering teacher and student outputs for candidate perturbed views. However, this cost remains manageable because perturbations are local and the action group size is small.

The UOT module is more expensive than heuristic pruning, but it is invoked only periodically rather than at every iteration. Moreover, the matching is performed at the voxel level instead of the anchor level, which makes the problem more tractable. Adaptive voxel resizing is further used when the voxel count becomes excessively large. In practice, this keeps the structural alignment cost under control while still providing a meaningful teacher-guided spatial prior.

G Additional Experimental Results

G.1 Structural Compactness Without Entropy Modeling

To separate structural compactness from the entropy backend, we train RLGS without entropy modeling and coding and compare its native model size on Mip-NeRF360 with the results reported for Octree-GS and GaussianSpa. Table 6 shows that this RLGS configuration reaches 27.87 dB at 118.0 MB. It is smaller than the reported Octree-GS model, but it does not outperform GaussianSpa in the quality–size trade-off.

The highrate result of the complete, entropy-coded RLGS configuration is included only as a reference. Learned entropy modeling affects training through the rate objective, whereas final entropy coding is applied after training. The configuration without entropy modeling and coding is therefore trained separately, rather than obtained by simply serializing the complete model without a coder.

Table 6: Mip-NeRF360 Comparison of native model sizes, with the entropy-coded RLGS result shown separately for reference. Octree-GS and GaussianSpa results are taken from their original papers; RLGS results use our evaluation pipeline. The complete entropy-modeled RLGS result is shown only for reference.

Method	PSNR↑	SSIM↑	LPIPS↓	MB↓
Octree-GS [53]	28.05	0.819	0.214	139.6
GaussianSpa [54]	27.85	0.825	0.214	25.0
RLGS without entropy modeling and coding	27.87	0.819	0.226	118.0
RLGS (highrate; entropy-coded bitstream)	27.81	0.810	0.231	16.72

G.2 Structural Compactness on Deep Blending

Table 7: Native model-size comparison on Deep Blending. Neither row includes learned entropy modeling or coding.

Method	PSNR↑	SSIM↑	LPIPS↓	MB↓
Scaffold-GS	30.28	0.907	0.243	121.2
RLGS without entropy modeling and coding	30.26	0.905	0.247	88.9

Within the Scaffold-GS anchor representation, RLGS reduces native model size from 121.2 to 88.9 MB on Deep Blending while retaining similar rendering quality. This comparison is informative about the compact student representation and structural pruning. The complete lowrate RLGS result of 2.74 MB is a final entropy-coded bitstream, whereas the 88.9 MB result is the native size of a separately trained configuration without entropy modeling or coding.

Table 8: Standard deviations across three independent RLGS runs on Deep Blending under the lowrate setting.

	PSNR (dB)	SSIM	LPIPS	Size (MB)
Standard deviation	0.04	0.003	0.003	0.05

G.3 Run-to-Run Variation

We repeat the complete RLGS training procedure with three independent random seeds. Table 8 indicates the scale of run-to-run variation on Deep Blending. Three seeds on one benchmark are insufficient to establish statistical significance for small differences from competing methods or for results on the other two benchmarks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions and scope of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results. The mathematical formulations are used to describe the proposed optimization objectives and algorithmic components.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the datasets, evaluation metrics, training procedure, and key hyperparameters needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments are conducted on publicly available benchmark datasets, and the paper provides detailed descriptions of the method and experimental settings. The code is not publicly released at submission time, but the implementation details and reproduction instructions are included to support reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the benchmark datasets, evaluation protocol, training configuration, loss functions, compression setting, and major hyperparameters. These details are sufficient to understand the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports standard quantitative metrics averaged over benchmark scenes and test views, following common practice in 3DGS compression evaluation.We do not report error bars or multi-seed statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: The paper reports the main computational resources used for training and evaluation, including the GPU setting and implementation environment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses standard public datasets and focuses on efficient 3D scene representation compression. It does not involve human subjects, private data collection, or applications that violate the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses that the proposed compression framework can reduce storage, transmission, and deployment costs for 3D scene representations, which may benefit VR, robotics, and digital twin applications. It also acknowledges that more efficient 3D representations could potentially be misused in privacy-sensitive reconstruction or unauthorized 3D content distribution, and such applications should follow appropriate data usage and privacy regulations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk assets such as pretrained language models, image generators, or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the original sources of the datasets, baselines, and related codebases used in the experiments. The use of public benchmark datasets and existing methods follows their corresponding licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset or benchmark. The main contribution is a compression algorithm evaluated on existing public datasets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, and therefore IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.