Diffeomorphism-Informed 3D Gaussian Splattings via Screen-Space Optimal Transport (OT)

Xiang Gao[1]*, **Xiaoping Zhu**[4], **Xinmu Wang**[1]*, **Yuanpeng Liu**[1]*, **Haotian Yin**[2]*
Dichang Zhang[1]*, **Yazheng Chen**[1], **Yue Wang**[1]*, **Yubin Zhou**[3], **Yu Guo**[3]
Wei Chen[5], **Xianfeng David Gu**[1], **Xiyun Song**[3]†, **Heather Yu**[3]

[1]Stony Brook University, [2]New Jersey Institute of Technology, [3]Futurewei Technologies
[4]Rutgers University, [5]Capital Normal University

[1]{gao2, xinmuwang, yuanpliu, diczhang, yazchen, wang139, gu}@cs.stonybrook.edu
[2]eileenyinht@gmail.com, [3]{yzhou, yguo1, xsong, hyu}@futurewei.com
[4]xz349@scarletmail.rutgers.edu, [5]wei.chen.sbu@gmail.com

Abstract

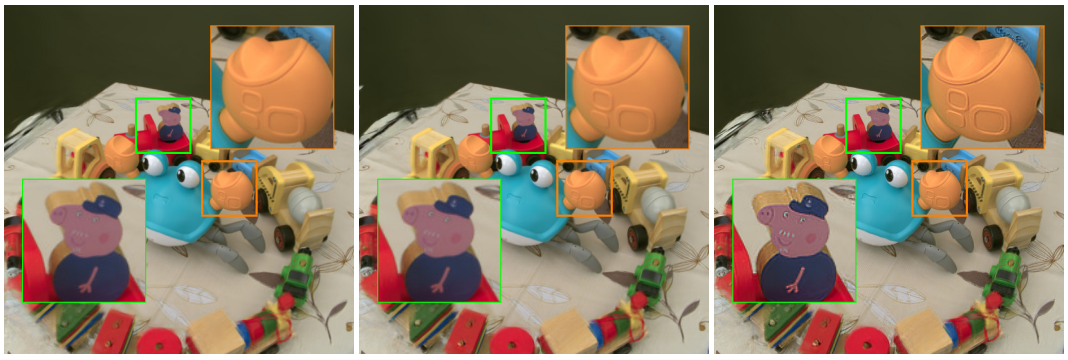
3D Gaussian Splatting (3DGS) has emerged as an effective solution for real-time novel view synthesis (NVS) with high rendering quality, but training memory scales with image resolution, making downsampled supervision common and leading to loss of sharp, high-frequency details and degraded perceptual quality. Moreover, uniform supervision assigns equal weight to all pixels, resulting in weak emphasis on detail-rich regions such as edges and thin structures. We propose **OT-3DGS**, which leverages optimal transport in screen space to reparameterize the image domain prior to downsampling, inducing importance-based non-uniform supervision that redistributes pixel budget toward geometry-critical regions and strengthens supervision on high-frequency content. Our method remains fully compatible with the original 3DGS pipeline, requiring no modification to loss functions, rendering settings, or pruning, cloning, and splitting strategies. Experiments across diverse datasets show that OT-3DGS produces sharper structures, clearer reflectance, and reduced over-smoothing; while PSNR and SSIM may slightly decrease due to deviation from downsampled references, reference-free metrics including BRISQUE, NIQE, and Laplacian Variance (LV) improve significantly, better reflecting perceptual quality and high-frequency detail. These results demonstrate that OT-3DGS aligns with the fundamental goal of NVS: generating perceptually realistic, high-quality unseen views.

1 Introduction

Novel view synthesis (NVS), the task of reconstructing a 3D scene from images and rendering it from novel viewpoints, is a fundamental problem in computer vision and graphics, with applications in VR/AR, medical imaging, autonomous driving, and 3D content creation. Beyond geometric accuracy, NVS aims to produce perceptually realistic renderings from unseen views. Neural Radiance Fields (NeRF) [25] achieve high-quality reconstruction with implicit representations but suffer from slow training and rendering. More recently, 3D Gaussian Splatting (3DGS) [18] enables real-time rendering with competitive perceptual quality using explicit Gaussian primitives. However, a key challenge remains the high GPU memory consumption during training, which scales with image resolution. In practice, high-resolution scenes are often infeasible to reconstruct on a single GPU, as

*Work done during internships at Futurewei Technologies.
†Corresponding Author

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



(a) 3DGS Trained on $\downarrow 2\times$ Peak GPU Use = 836 MB

(b) 3DGS Trained on $\downarrow 4\times$ 371 MB

(c) OT-3DGS (Ours) on $\downarrow 4\times$ 490 MB

Figure 1: **Visual comparison of novel view synthesis perceptual quality.** (a) 3DGS [18] trained with a twofold reduction in resolution. (b) 3DGS trained with a fourfold reduction in resolution. (c) Our method, using non-uniform supervision at a fourfold reduction in resolution, achieves superior perceptual quality, preserving sharp structures and fine details.

modern multi-megapixel inputs require substantial memory for scene parameters, optimization states, and intermediate computations.

To address these memory constraints, a widely adopted strategy in the static NVS research community is to train and evaluate on images downsampled by a factor of four from the original high-resolution inputs, enabling single-GPU feasibility and continued empirical progress. This practice is followed by state of the arts methods such as Neural Radiance Fields (NeRF) [25, 2], Plenoxels [46], Instant Neural Graphics Primitives (iNGP) [28], Mip-NeRF 360 [2], RobustNeRF [33], NeRF on-the-Go [32], 3D Gaussian Splatting (3DGS) [18], and 2D Gaussian Splatting (2DGS) [13], and has remained the de facto standard in subsequent static NVS research, largely due to its practicality in balancing computational efficiency with empirical performance under limited hardware resources.

However, this practice exhibits two fundamental limitations. First, downsampling irreversibly removes high-frequency information, leading to supervision and evaluation signals that favor blurred or aliased structures and fail to capture sharp features, small text, and localized details present in the original images. Second, uniform supervision assigns equal weight to all pixels, producing weak gradients for sparse high-frequency regions that occupy only a small portion of the image, resulting in their under-representation in the reconstructed 3D Gaussian splats, particularly for edges and thin structures. These limitations motivate the need for a strategy that preserves high-frequency information prior to downsampling while enabling importance-based non-uniform supervision.

In this work, we propose **OT-3DGS**, which leverages optimal transport in screen space to induce importance-based non-uniform supervision before downsampling. We construct an edge-guided density g from high-resolution images and compute a Brenier potential u, whose gradient $T = \nabla u$ defines a transport map that redistributes spatial density, allocating more area to geometry-critical regions. We apply T prior to downsampling so that these regions receive more pixel budget, effectively converting uniform supervision into importance-based non-uniform supervision. Training is performed in the transported domain, improving retention of sharp, high-frequency details without modifying the loss, rendering settings, or the pruning, cloning, or splitting strategies in 3DGS [18].

Furthermore, standard 3DGS uses pixel-wise losses that average errors, leading to regression-to-the-mean behavior in high-frequency regions. Under sparse views or complex geometry, densification may rely on larger Gaussians with averaged colors, producing blurred artifacts, while low-order spherical harmonics oversmooth view-dependent effects and miss sharp specularities. Existing evaluation protocols further obscure these issues by using PSNR, SSIM [40], and LPIPS [49] against downsampled ground truth, favoring agreement with degraded signals. As our supervision is defined in the transported domain, such comparisons are mismatched. We therefore report reference-free metrics, including BRISQUE [26], NIQE [27], and Laplacian Variance (LV), which better capture perceptual quality and high-frequency sharpness, aligning with the goal of novel view synthesis.

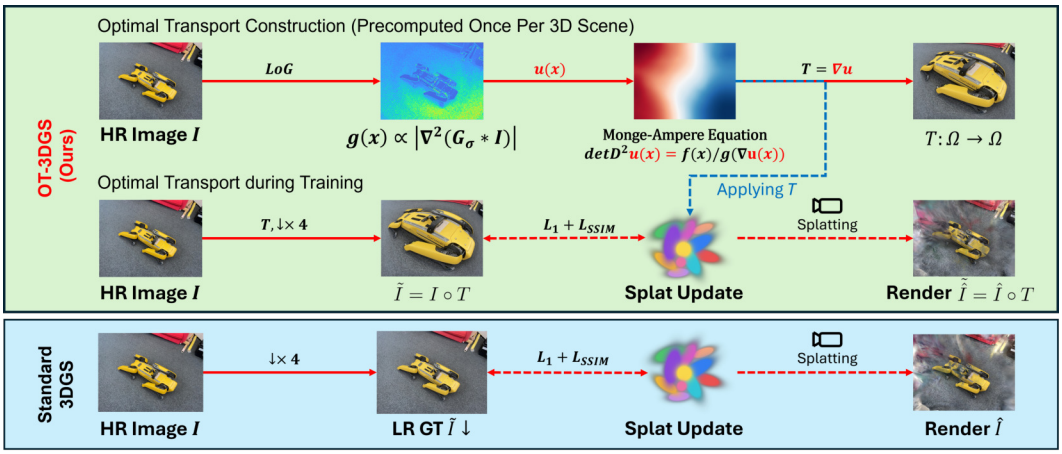


Figure 2: **Overview of the OT-3DGS pipeline.** Standard 3DGS [18] (bottom) struggles to retain sharp, high-frequency structures under $4\times$ downsampled supervision due to uniform supervision, reducing perceptual quality. OT-3DGS (top) precomputes a transport map T from an edge-guided density to reparameterize the image domain before downsampling. During training, both rendered images and ground truth are transformed by T, inducing importance-based non-uniform supervision with stronger gradients on detail-rich regions. As a result, edges, corners, and thin structures are better preserved, improving perceptual quality as reflected by BRISQUE, NIQE, and LV, directly supporting high-quality unseen view synthesis.

In summary, our contributions are: (i) We identify that uniform supervision on downsampled ground truth leads to loss of sharp, high-frequency details and poor representation of edges and thin structures. (ii) We propose OT-3DGS, which leverages optimal transport in screen space to induce importance-based non-uniform supervision. We construct an edge-guided density from high-resolution images and compute a Brenier potential u, whose gradient $T = \nabla u$ defines a geometry-aware transport map applied prior to downsampling, with optimization performed in the transported domain. (iii) This paradigm converts uniform supervision into importance-based non-uniform supervision, addressing limitations from both downsampling and uniform supervision. (iv) We demonstrate significantly improved perceptual quality over 3DGS [18] with downsampled supervision, particularly in sharpness, reflectance, and high-frequency details; we report reference-free metrics (BRISQUE, NIQE, LV) with negligible peak memory overhead per scene, aligning with the goal of novel view synthesis to generate perceptually high-quality unseen views.

2 Related Work

3D Gaussian Splatting. 3D Gaussian Splatting [18] represents scenes using anisotropic Gaussian primitives with differentiable rasterization, enabling real-time novel view synthesis with competitive perceptual quality. Subsequent work improves rendering [34, 16, 22], anti-aliasing and multi-scale modeling [43, 47], densification [6, 51, 30, 50], efficiency [29, 48], and extends to surfaces [9, 13, 14], dynamics [41, 45, 4], generative modeling [5, 36], and large-scale scenes [24, 23, 15]. Despite this progress, existing methods rely on uniform photometric supervision from downsampled images such as L1 and SSIM losses, where downsampling and uniform weighting dilute high-frequency signals and lead to weak representation of edges and thin structures in detail-rich regions. In contrast, we use optimal transport to reparameterize the image domain prior to downsampling, inducing importance-based non-uniform supervision that allocates more sampling density to geometry-critical regions. This improves reconstruction of edges, corners, and thin structures without modifying rendering settings or the pruning, cloning, or splitting strategies used in 3DGS [18], and aligns with the goal of novel view synthesis to generate perceptually high-quality unseen views.

Supervision and Loss Design in Neural Rendering. Prior work addresses limitations of uniform supervision through frequency-aware losses [44, 42], perceptual losses [17, 49], and importance sampling [35, 20]. However, these approaches rely on predefined weighting, operate in feature space, or act at the ray level, and do not adapt the spatial distribution of supervision in the image domain,

often leading to weak emphasis on high-frequency regions such as edges and thin structures. In contrast, we reparameterize the image domain via optimal transport to induce importance-based non-uniform supervision, with optimization performed in the transported domain, allocating more supervision to geometry-critical regions and improving reconstruction of high-frequency details, consistent with the perceptual goals of novel view synthesis.

Optimal Transport. Optimal transport (OT) [38, 39, 21, 12] provides a mathematical framework for computing cost-minimizing mappings between probability measures. Under quadratic cost, Brenier's theorem [3, 38] guarantees the existence of a unique measure-preserving transport map $T = \nabla u$, where u is a convex Brenier potential satisfying the Monge–Ampère equation (see Appendix A for a self-contained derivation). OT has been widely applied used in computer vision and machine learning, including colour transfer in image processing [31, 10], unsupervised domain adaptation [8, 7], and generative modelling [1, 11]. In this work, we apply optimal transport in screen space to full-resolution images before downsampling, using an edge-guided density that reflects local geometric structure. The resulting transport map, precomputed with the FFT-OT algorithm [21] in $O(N \log N)$ time, reallocates screen-space resolution toward detail-rich regions. As a result, fine-scale structures occupy larger regions in the transported space, retain more information after downsampling, and provide stronger training signals during 3DGS optimization, as detailed in Section 3.

3 Method

Our goal is to address the loss of sharp, high-frequency detail in 3DGS [18] caused by $4\times$ downsampling and uniform supervision by reparameterizing the image domain prior to downsampling. Unlike standard 3DGS, which applies supervision directly on downsampled images, we first transform high-resolution images via optimal transport to redistribute spatial density toward detail-rich regions such as edges, text, and thin structures, and then perform downsampling. This induces importance-based non-uniform supervision, strengthening gradients in geometry-critical regions. The overall pipeline is illustrated in Figure 2. We begin with a brief review of 3DGS, followed by the construction of the target density, the transport map, and the training objective.

3.1 Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [18] represents a 3D scene as a set of anisotropic Gaussian splats $\mathcal{G} = \{G_i\}$, where each splat $G_i \in \mathcal{G}$ is parameterized by a mean $\mu_i \in \mathbb{R}^3$, covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity $\sigma_i \in \mathbb{R}$, and view-dependent color c_i represented by spherical harmonics coefficients. To ensure positive semi-definiteness property during the optimization, Σ_i is factorized using a rotation matrix $R_i \in \mathbb{R}^{3\times3}$ and a scale matrix $S_i \in \mathbb{R}^{3\times3}$:

$$G_i(x) = e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)} \qquad \Sigma_i = R_i S_i S_i^T R_i^T. \tag{1}$$

To render the 3D scene representation, the splats are rasterized to screen coordinates via projective transformation. Following Zwicker et al. [52], the projection of the 3D covariance to 2D screen space can be approximated as:

$$\Sigma_i' = JW\Sigma_i W^T J^T. \tag{2}$$

where $W \in \mathbb{R}^{3\times3}$ is the viewing transformation rotation and $J \in \mathbb{R}^{2\times3}$ is the Jacobian of the projection matrix, which provides a linear approximation of the non-linear projection process. The color $\hat{C}$ of a pixel can be computed using the volume rendering equation along its corresponding ray:

$$\hat{C} = \sum_{i=1}^{N} c_i \alpha_i T_i, \quad T_i = \prod_{j=1}^{i-1}(1-\alpha_j), \tag{3}$$

where N is the number of Gaussian splats that the ray passes through, i is the index in the sorted order by depth, α_i is the blending weight of the i-th Gaussian splat along the ray, and $T_i \in \mathbb{R}$ is the accumulated transmittance through preceding gaussians.

3.2 Screen-Space Reparameterization via Optimal Transport

Overview. Given a high-resolution image $I : \Omega \to \mathbb{R}^3$, defined on the rectangular pixel domain $\Omega = [-1, 1] \times [-1, 1]$, our goal is to construct a transport map $T : \Omega \to \Omega$ that reallocates screen-space resolution toward geometrically informative regions before downsampling and training. Instead

of applying uniform supervision on the downsampled image, we first compute an edge-guided density from the full-resolution ground truth and use optimal transport to obtain a screen-space reparameterization. This allows fine-scale structures, such as sharp boundaries and small text, to occupy larger regions in the transported domain and retain more information after downsampling.

Edge-Guided Density. We construct the target density from the high-resolution ground truth I using the Laplacian of Gaussian (LoG) response:

$$\phi(x) = \left|\nabla^2 (G_\sigma * I)(x)\right| + \epsilon, \qquad g(x) = \frac{\phi(x)}{\int_\Omega \phi}, \tag{4}$$

where G_σ is a Gaussian kernel with scale σ, $\epsilon > 0$ ensures strict positivity, and g is normalized to integrate to one. The density g is larger around edges, corners, and fine structures, and small in smooth regions. Since g is computed from the full-resolution image, it captures geometric detail that would be weakened or removed by direct downsampling. The edge-guided density construction is presented as an algorithm in Appendix 3, with ablations in Appendix C.6. The density choice is very task-specific, and our method can accommodate different density definitions.

Screen-Space OT Map. We compute the optimal transport map from the uniform density $f = 1/|\Omega|$ to the edge-guided density g. Under quadratic cost, Brenier's theorem [3] gives a unique transport map of the form $T = \nabla u$, where u is a convex Brenier potential satisfying the Monge–Ampère equation,

$$\det D^2 u(x) = \frac{f(x)}{g(\nabla u(x))}. \tag{5}$$

In practice, we solve for T using the FFT-OT algorithm [21], which reformulates this equation as a fixed-point iteration and solves each step with DCT-based Poisson solvers in $O(N \log N)$ time. The OT map is precomputed once for each training view and reused throughout training. We provide the mathematical background and solver details in Appendix A.

Training Objective. During training, for each sampled camera view, we render the image $\hat{I}$ using the current 3D Gaussians splats and evaluate the photometric loss in the transported screen space. Let $\tilde{I} = I \circ T$ and $\tilde{\hat{I}} = \hat{I} \circ T$ denote the transported ground truth and rendered image, respectively, computed by bilinear interpolation. The OT-guided training loss is defined as

$$\mathcal{L}_{\mathrm{OT}} = (1-\lambda)\,\|\tilde{\hat{I}} - \tilde{I}\|_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}(\tilde{\hat{I}}, \tilde{I}), \tag{6}$$

where $\lambda = 0.2$ follows standard 3DGS practice. To suppress over-densification in regions where the transport concentrates mass, we optionally enhance edge preservation while smoothing noise with a Relative Difference Prior (RDP) [19] regularization term:

$$\mathcal{L} = \mathcal{L}_{\mathrm{OT}} + \beta\,\mathcal{R}(\hat{I}), \tag{7}$$

where $\mathcal{R}(\hat{I})$ penalizes large relative intensity differences between spatially neighboring pixels and β=0.5 by default. We analyze its effect in Appendix C.5. However, throughout this work, we use $\beta = 0$ (no regularization), which produces a superior perceptual quality. Since directly applying T is memory-intensive, we precompute T^{-1} offline per camera via fixed-point iteration, keeping peak GPU memory within $1.6\times$ of standard 3DGS; we find $K = 10$ sufficient for accurate precomputation (See Appendix C.2). Full training procedures are provided as algorithms in Appendix C, including both a direct implementation without considering GPU memory and a memory-efficient variant.

Because the transport map allocates more screen-space area to high-density regions before downsampling, fine-scale structures occupy larger regions in the transported image and retain more information, leading to stronger gradients during optimization. In contrast, standard 3DGS applies uniform supervision on downsampled images, where small high-frequency structures may collapse into only a few pixels and provide weak training signals.

4 Experiments and Results

4.1 Experimental Setup

Dataset. Following 3D Gaussian Splatting [18], we evaluate on Mip-NeRF360 [2] dataset, which contains clean large-scale scenes with wide viewpoint variation. To assess robustness, we additionally evaluate on RobustNeRF [33] and NeRF-On-the-Go [32] datasets. RobustNeRF [33] dataset

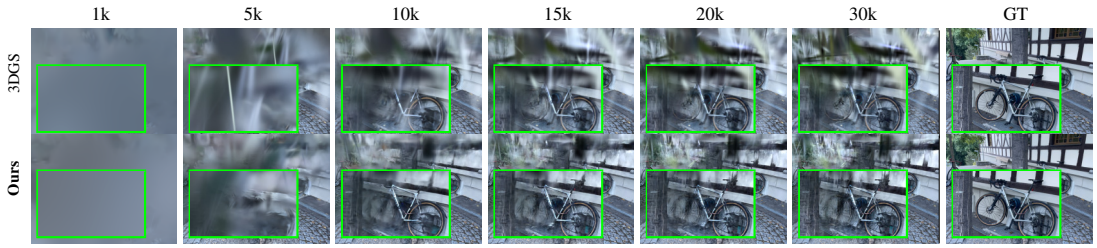


Figure 3: **Visual ablation of convergence over 30k iterations on Tree scene from NeRF On-the-Go [32].** Green insets show high-frequency structures across iterations. Our method starts to converge around 10k iterations, with sharper structures in the green boxes, while 3DGS [18] does not reach comparable sharp detail. GT denotes standard $4\times$ downsampled high-resolution images.



Figure 4: **Visual comparison with 3DGS [18] on the NeRF On-the-Go [32] dataset.** Our method with non-uniform supervision better retains high-frequency structures which occupy small pixel regions, such as sharp edges and small text, while 3DGS often struggles under uniform supervision. Here, the ground truth (GT) refers to standard $4\times$ downsampled high-resolution images. Please see the Appendix E for full qualitative results.

introduces real-world degradations such as motion blur, occlusions, and transient objects, while NeRF-On-the-Go dataset captures casual handheld scenes with dynamic content, varying illumination, and pose noise. For all datasets, we follow the provided train/test splits.

Implementation Details. Our method builds on 3DGS [18] with default settings (SH degree 3, 30k iterations, Adam with decayed position LR $1.6 \times 10^{-4} \rightarrow 1.6 \times 10^{-6}$, opacity reset every 3k, and densification from 500–15k with $\tau = 2 \times 10^{-4}$). Standard 3DGS [18] uses $4\times$ downsampled images (`images_4`), while our method uses OT-transported downsampled images and potentials (`ot_image_4`, `ot_potential_4`). For each camera, we construct an edge-guided density g from the high-resolution image using LoG ($\sigma = 1.0$), and solve the Monge–Ampère equation via an FFT-based solver to obtain the potential u on a shared grid with tolerance $\epsilon = 10^{-6}$. We precompute

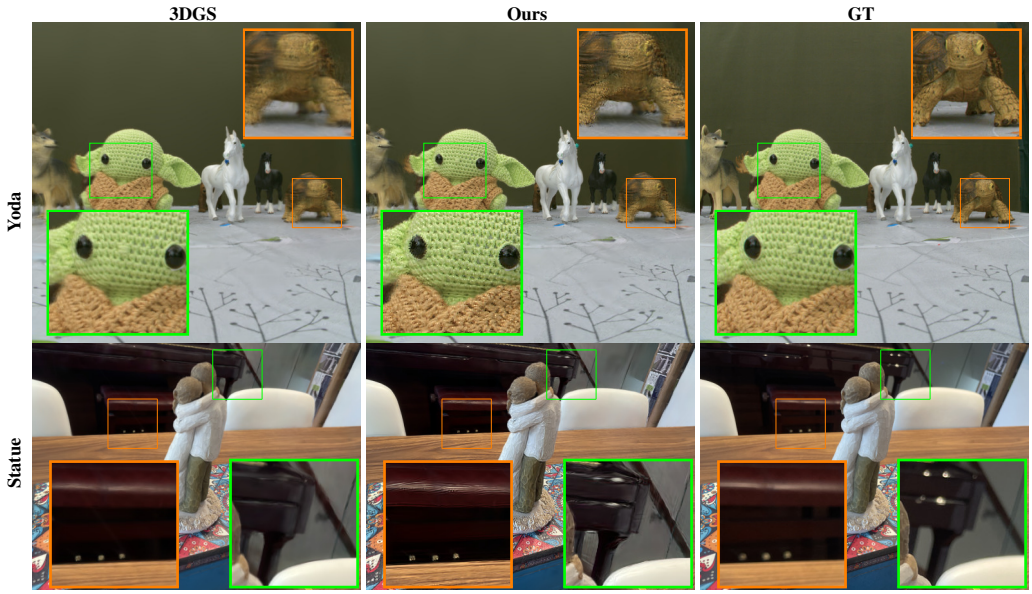


Figure 5: **Visual comparison with 3DGS [18] on RobustNeRF [33] scenes.** Our method achieves superior perceptual quality in zoomed-in object regions, while 3DGS remains blurred. Here, GT denotes standard $4\times$ downsampled high-resolution images. Please see the Appendix E for full qualitative results.

$T = \nabla u$ and its inverse T^{-1} (via fixed-point) per view. During training, $\hat{I}(T^{-1}(\mathbf{x}))$ is evaluated using `grid_sample`, requiring an $(H \times W \times 2)$ grid and $\sim$1.6$\times$ peak GPU memory relative to 3DGS [18]. **All experiments run on a single RTX 4090 GPU**; precomputation takes $\sim$2–3 minutes per scene, and training time is within $\sim$5–10% of 3DGS [18] under the same downsampled supervision setting.

Evaluation Metrics. Reference-based metrics used in 3DGS [18], such as PSNR and SSIM [40], evaluate against $4\times$ downsampled ground truth and tend to favor low-variance, smoothed reconstructions, making them less sensitive to high-frequency details. In OT-3DGS, supervision is defined after reparameterization, where pixel budget is redistributed toward detail-rich regions, making such comparisons less aligned with the model's objective. We therefore report reference-free metrics, including BRISQUE [26] and NIQE [27], which capture perceptual naturalness, and Laplacian Variance (LV) (Appendix C.3), which measures image sharpness via high-frequency content, better reflecting perceptual quality and the goal of novel view synthesis.

4.2 Results

Quantitative Results. Traditional metrics such as PSNR and SSIM tend to favor low-variance, overly smoothed reconstructions, often suppressing high-frequency details and fine structures. In contrast, we focus on perceptual quality and report no-reference metrics, including BRISQUE [26], NIQE [27], and Laplacian Variance (LV), which better capture image naturalness and sharpness by reflecting high-frequency content. As shown in Table 1, across the RobustNeRF [33], NeRF-On-the-Go [32], and Mip-NeRF360 [2] datasets, OT-3DGS achieves significantly improved BRISQUE, NIQE, and LV scores compared to 3DGS [18], indicating consistently stronger perceptual quality and sharper reconstructions across diverse scenes. While PSNR and SSIM decrease slightly, this reflects a known trade-off: optimizing for perceptual sharpness and high-frequency detail can deviate from strict pixel-wise agreement with the reference. We also observe a marginal increase in LPIPS, which is consistent with prior observations that sharper, perceptually plausible details may introduce small deviations from ground-truth pixel values, even when they better match human visual perception.

As shown in the last two columns of Table 1, despite incorporating an additional block, the OT block, in the pipeline, OT-3DGS maintains comparable GPU memory usage ($0.91\times$ to $1.43\times$ compared to the baseline) while preserving fine-scale textures and structural details. In contrast, the baseline 3DGS must rely on significantly higher input resolutions to achieve comparable visual fidelity, resulting in a

Table 1: **Quantitative comparison with 3DGS [18] on the Mip-NeRF360 [47], RobustNeRF [33], and NeRF On-the-Go [32] datasets.** Mean results over all scenes. Metrics include BRISQUE, NIQE, and Laplacian Variance (LV), which capture perceptual quality and high-frequency sharpness. Training uses OT-transported images with $4\times$ downsampling and importance-based non-uniform supervision, redistributing pixel budget toward detail-rich regions. † denotes published baseline results reported by 3DGS [18], for which BRISQUE, NIQE, and LV are additionally evaluated.

	No-Reference			*Reference-Based*			*GPU Memory*	
Method	BRISQUE↓	NIQE↓	LV↑	PSNR↑	SSIM↑	LPIPS↓	Peak (MB)↓	GPU Rel.↓
Mip-NeRF 360 Dataset (9 scenes)								
Plenoxels†	55.92	6.58	33	23.087	0.626	0.456	–	–
iNGP†	27.33	4.88	322	25.306	0.671	0.371	–	–
iNGP-big†	27.53	4.55	363	25.590	0.699	0.331	–	–
MipNeRF360†	28.60	3.29	392	27.694	0.792	0.238	–	–
3DGS†	20.93	2.70	673	27.298	0.815	0.206	–	–
3DGS (Retrained)	20.69	2.69	699	**27.787**	**0.827**	**0.171**	2249	1.00×
Ours (w/o reg)	**18.39**	**2.52**	**1356**	25.702	0.788	0.200	2498	1.11×
Ours (w/ reg)	22.19	2.77	738	26.858	0.808	0.192	**2232**	**0.99**×
RobustNeRF Dataset (5 scenes)								
3DGS (Retrained)	41.47	4.37	130	**22.930**	**0.774**	**0.325**	**728**	**1.00**×
Ours (w/o reg)	**27.71**	**3.41**	**582**	22.296	0.734	0.345	1042	1.43×
Ours (w/ reg)	30.84	3.55	287	22.770	0.755	0.332	1021	1.40×
NeRF On-the-Go Dataset (12 scenes)								
3DGS (Retrained)	24.40	3.34	409	**15.918**	**0.501**	**0.459**	858	1.00×
Ours (w/o reg)	**18.30**	**2.79**	**1917**	15.414	0.460	0.469	857	1.00×
Ours (w/ reg)	23.13	3.15	505	15.837	0.487	0.468	**778**	**0.91**×

substantial increase in GPU memory consumption that frequently exceeds hardware limits and leads to memory exhaustion or system crashes.

Qualitative Results. As shown in Figures 1, 4, 5, and 6, OT-3DGS produces reconstructions with sharper edges, clearer structures, and reduced over-smoothing compared to 3DGS. On NeRF-On-the-Go [32], our method better retains fine textures and local details under challenging capture conditions. On RobustNeRF [33], OT-3DGS improves structural clarity in sparse-view and complex regions, reducing the characteristic smearing artifacts of the baseline. On Mip-NeRF360 [2], we observe sharper large-scale scene reconstruction with improved reflectance and high-frequency details across wide-view settings. These improvements are particularly important for novel view synthesis, where perceptual plausibility and consistency are more critical than exact pixel matching. By emphasizing high-frequency detail, OT-3DGS yields more visually realistic reconstructions, while baseline results exhibit blurred structures and muted highlights.

4.3 Ablation Studies

We analyze the behavior of OT-3DGS through a series of ablation studies. Figure 3 illustrates how finer details progressively emerge over iterations on the Tree example. Complementing this visualization, Table 4 in Appendix C.4 studies convergence across iterations (1k, 5k, 10k, 15k, 30k) on both Train Station and Tree, showing steady improvements in perceptual quality from initially underdeveloped results. Table 6 in Appendix C.6 examines the choice of edge-guided density, where our LoG-based formulation achieves better perceptual quality with lower BRISQUE and NIQE and strong LV compared to other edge-based densities. Table 5 in Appendix C.5 evaluates Relative Difference Prior (RDP) regularization, showing improved stability but reduced LV, indicating a trade-off with perceptual quality; we therefore favor the unregularized setting with $\beta = 0$. Table 7 in Appendix C.7 compares Naive OT and grid-sampled OT, showing nearly equivalent perceptual quality with comparable BRISQUE, NIQE, and LV, while significantly reducing memory usage. Tables in Appendix D report full quantitative results for all scenes individually, consistently showing

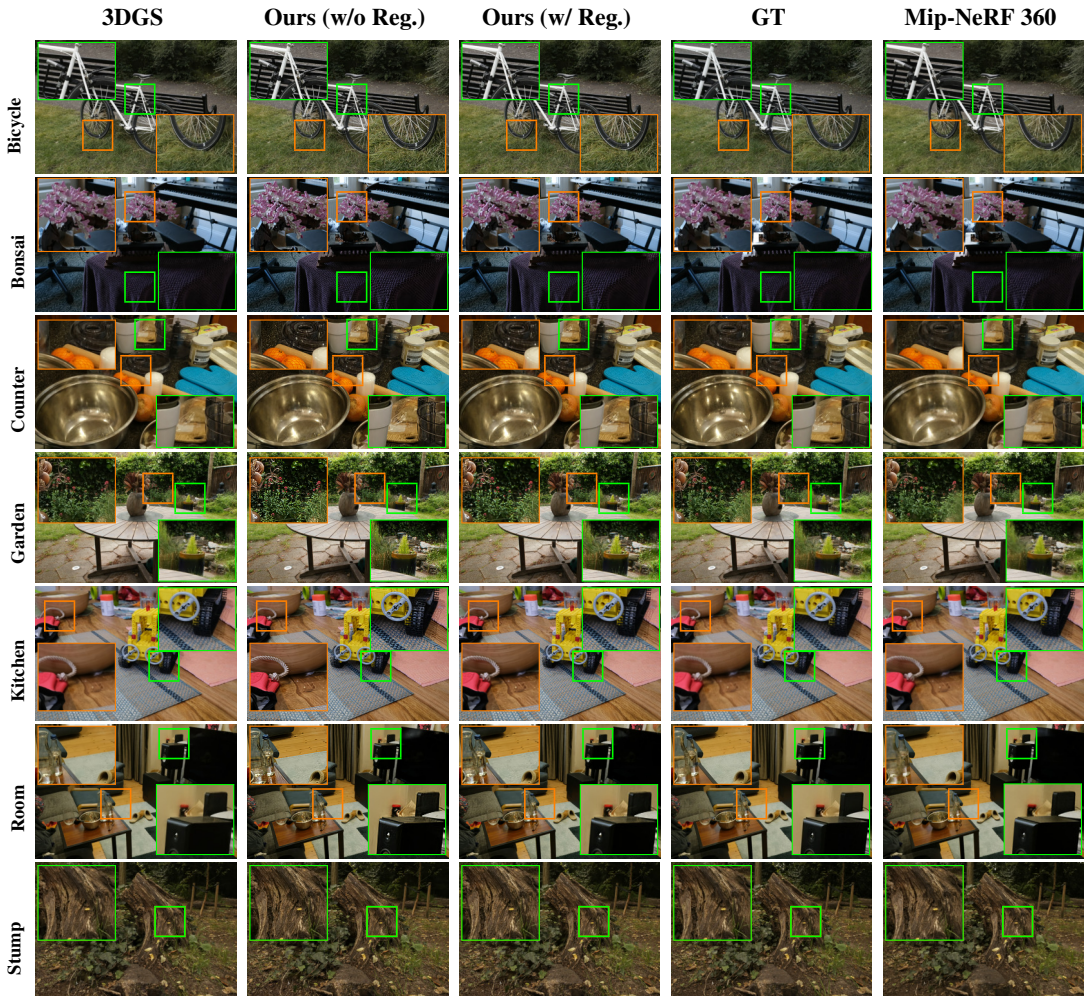


Figure 6: **Qualitative comparison with 3DGS [18] on the Mip-NeRF 360 [2] dataset.** Our method achieves sharper structural details through geometry-aware, non-uniform supervision, improving retention of structures occupying small pixel regions compared to 3DGS [18]. Here, GT denotes standard $4\times$ downsampled high-resolution images.

improved perceptual quality across datasets. Finally, Table 3 in Appendix C.2 reports peak GPU memory on Mip-NeRF360 [2] scenes, where the ratio column indicates memory overhead relative to 3DGS [18], remaining close to $1\times$ on average.

5 Conclusion

We presented OT-3DGS, which addresses the loss of sharp, high-frequency detail in 3D Gaussian Splatting caused by downsampling and uniform supervision. By leveraging optimal transport to reparameterize the image domain prior to downsampling, our method induces importance-based non-uniform supervision that emphasizes geometry-critical regions such as edges and thin structures. This results in reconstructions with sharper structures, clearer reflectance, and reduced over-smoothing, while remaining fully compatible with the original 3DGS [18] pipeline without modifying loss functions, rendering settings, or pruning, cloning, and splitting strategies. Experiments across diverse datasets show consistent improvements in perceptual quality, as reflected by reference-free metrics including BRISQUE [26], NIQE [27], and Laplacian Variance (LV), which better capture sharpness and high-frequency detail. Overall, OT-3DGS advances novel view synthesis toward generating perceptually realistic, high-quality unseen views while maintaining GPU memory efficiency on par with 4× downsampling in the baseline 3DGS.

Limitations. Our method has several limitations. First, the importance density g is task-dependent, and our LoG-based formulation primarily prioritizes high-frequency structures. Its effectiveness depends on the filter scale and may decrease under severe blur or low contrast, while strong noise can introduce misleading edge responses. Second, concentrating supervision around edges may occasionally over-emphasize high-contrast patterns, producing over-sharpening or localized artifacts. Although the optional Relative Difference Prior (RDP) regularization mitigates excessive local variation, it may also suppress fine details, reflecting a trade-off between sharpness and reconstruction fidelity. Moreover, no-reference metrics such as LV may reward noise or excessive sharpening; therefore, improvements in these metrics do not necessarily translate into gains in PSNR, SSIM, or LPIPS. Third, transport maps are precomputed for each training view and remain fixed during optimization, limiting adaptation to evolving reconstruction errors. Furthermore, our method does not directly address challenges arising from inaccurate camera poses, transient objects, or severe motion blur. Finally, OT precomputation introduces additional computational and memory overhead relative to standard 3DGS under identical downsampling settings. Exploring adaptive importance densities and more general transport domains remains a promising direction for future work.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 214–223. PMLR, 2017.

[2] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2021. URL `https://api.semanticscholar.org/CorpusID:244488448`.

[3] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991.

[4] W. Cai, W. Ye, P. Ye, T. He, and T. Chen. Dynasurfgs: Dynamic surface reconstruction with planar-based gaussian splatting. *arXiv preprint arXiv:2408.13972*, 2024.

[5] Z. Chen, F. Wang, Y. Wang, and H. Liu. Text-to-3d using gaussian splatting, 2024. URL `https://arxiv.org/abs/2309.16585`.

[6] K. Cheng, X. Long, K. Yang, Y. Yao, W. Yin, Y. Ma, W. Wang, and X. Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. *arXiv preprint arXiv:2402.14650*, 2024.

[7] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[8] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[9] P. Dai, J. Xu, W. Xie, X. Liu, H. Wang, and W. Xu. High-quality surface reconstruction using gaussian surfels, 2024. URL `https://arxiv.org/abs/2404.17774`.

[10] O. Frigo, N. Sabater, V. Demoulin, and P. Hellier. Optimal transportation for example-guided color transfer. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *Computer Vision – ACCV 2014*, pages 655–670, Cham, 2015. Springer International Publishing. ISBN 978-3-319-16811-1.

[11] A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with Sinkhorn divergences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1608–1617. PMLR, 2018.

[12] X. Gu, F. Luo, J. Sun, and S. T. Yau. Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations, 2016. URL `https://arxiv.org/abs/1302.5472`.

[13] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

[14] H. Huang, Y. Wu, C. Deng, G. Gao, M. Gu, and Y.-S. Liu. Fatesgs: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[15] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, 2023.

[16] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces, 2023. URL https://arxiv.org/abs/2311.17977.

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016. URL https://arxiv.org/abs/1603.08155.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[19] C. Knaus and M. Zwicker. Progressive image denoising. *IEEE Transactions on Image Processing*, 23(7): 3114–3125, 2014. doi: 10.1109/TIP.2014.2326771.

[20] A. Kurz, T. Neff, Z. Lv, M. Zollhöfer, and M. Steinberger. Adanerf: Adaptive sampling for real-time rendering of neural radiance fields, 2022. URL https://arxiv.org/abs/2207.10312.

[21] N. Lei and X. Gu. Fft-ot: A fast algorithm for optimal transportation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6280–6289, October 2021.

[22] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia. Gs-ir: 3d gaussian splatting for inverse rendering, 2024. URL https://arxiv.org/abs/2311.16473.

[23] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan, and W. Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction, 2024. URL https://arxiv.org/abs/2402.17427.

[24] S. S. Mallick, R. Goel, B. Kerbl, M. Steinberger, F. V. Carrasco, and F. De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687694. URL https://doi.org/10.1145/3680528.3687694.

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

[26] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. doi: 10.1109/TIP.2012.2214050.

[27] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20:209–212, 2013. URL https://api.semanticscholar.org/CorpusID:16892725.

[28] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. doi: 10.1145/3528223.3530127.

[29] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani, and H. Pirsiavash. Compact3d: Smaller and faster gaussian splatting with vector quantization. *arXiv preprint arXiv:2311.18159*, 2023.

[30] H. C. Nguyen, W. Mao, J. M. Alvarez, and M. Liu. Ibgs: Image-based gaussian splatting, 2025. URL https://arxiv.org/abs/2511.14357.

[31] F. Pitié, A. C. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1–2):123–137, 2007.

[32] W. Ren, Z. Zhu, B. Sun, J. Chen, M. Pollefeys, and S. Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20626–20636, June 2023.

[34] H. Sun, Y. Gao, J. Xie, J. Yang, and B. Wang. Svg-ir: Spatially-varying gaussian splatting for inverse rendering, 2025. URL https://arxiv.org/abs/2504.06815.

[35] S. Sun, M. Liu, Z. Fan, Y. Liu, C. Lv, L. Dong, and L. Kong. Efficient ray sampling for radiance fields reconstruction, 2023. URL https://arxiv.org/abs/2308.15547.

[36] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation, 2024. URL https://arxiv.org/abs/2309.16653.

[37] J. I. E. Urbas. The oblique derivative problem for equations of Monge–Ampère type in two dimensions. *Proceedings of the Centre for Mathematics and its Applications*, 12:171–195, 1987.

[38] C. Villani. *Topics in Optimal Transportation*, volume 58. American Mathematical Society, 2003.

[39] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[40] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612, 05 2004. doi: 10.1109/TIP.2003.819861.

[41] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, June 2024.

[42] Q. Xu, X. Yi, J. Xu, W. Tao, Y.-S. Ong, and H. Zhang. Few-shot nerf by adaptive rendering loss regularization, 2024. URL https://arxiv.org/abs/2410.17839.

[43] Z. Yan, W. F. Low, Y. Chen, and G. H. Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20923–20931, June 2024.

[44] J. Yang, M. Pavone, and Y. Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[45] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction, 2023. URL https://arxiv.org/abs/2309.13101.

[46] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2021. URL https://api.semanticscholar.org/CorpusID:245006364.

[47] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, June 2024.

[48] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes, 2024. URL https://arxiv.org/abs/2404.10772.

[49] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00068. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00068.

[50] Y. Zhang, G. Sun, and C. Chen. Eggs: Exchangeable 2d/3d gaussian splatting for geometry-appearance balanced novel view synthesis, 2025. URL https://arxiv.org/abs/2512.02932.

[51] H. Zhou and Z. Ni. Perceptual-gs: Scene-adaptive perceptual densification for gaussian splatting, 2025. URL https://arxiv.org/abs/2506.12400.

[52] M. Zwicker, H. Pfister, J. Baar, and M. Gross. Ewa volume splatting. *IEEE Visualization*, 01 2001. doi: 10.1109/VISUAL.2001.964490.

A Mathematical Background: Optimal Transport

We provide a self-contained review of the optimal transport concepts used in this paper. For a comprehensive treatment, see [38, 39].

Monge's Problem. Given two probability measures μ and ν on domains Ω and Ω^* respectively, and a cost function $c : \Omega \times \Omega^* \to \mathbb{R}$, Monge's problem seeks a measurable map $T : \Omega \to \Omega^*$ satisfying the measure-preserving condition $T_{\#}\mu = \nu$ for all measurable A) while minimizing the total transport cost:

$$\min_{T:\, T_{\#}\mu=\nu} \int_{\Omega} c(x, T(x))\, d\mu(x). \tag{8}$$

Here $T_{\#}\mu$ denotes the push-forward measure, i.e., $\nu(A) = \mu(T^{-1}(A))$ for all measurable sets $A \subseteq \Omega^*$. In our setting, $\Omega = \Omega^* = [-1, 1] \times [-1, 1]$ is the image domain, $\mu = f\, dx$ is the uniform measure, $\nu = g\, dx$ is the edge-guided importance measure, and $c(x, y) = \|x - y\|^2$ is the squared Euclidean cost.

Brenier's Theorem. Under the quadratic cost $c(x, y) = \frac{1}{2}\|x - y\|^2$ and mild regularity conditions on μ and ν, Brenier [3] proved that the unique solution to Monge's problem is:

$$T = \nabla u, \tag{9}$$

where $u : \Omega \to \mathbb{R}$ is a convex function called the *Brenier potential*. The map $T = \nabla u$ is the unique optimal transport map and is characterized by the convexity of u alone.

The Monge–Ampère Equation. The measure-preserving condition $T_{\#}\mu = \nu$ combined with $T = \nabla u$ yields the Monge–Ampère equation — a fully nonlinear second-order PDE for the Brenier potential:

$$\det D^2 u(x) = \frac{f(x)}{g(\nabla u(x))}, \tag{10}$$

with condition $\nabla u(\Omega) = \Omega^*$. Here $D^2 u$ is the Hessian of u and $\det D^2 u > 0$ everywhere since u is convex. Equation (10) is highly nonlinear and does not admit closed-form solutions in general, motivating numerical solvers.

Neumann Boundary Conditions. On a rectangular domain $\Omega = [-1, 1] \times [-1, 1]$ with $\Omega = \Omega^*$, the boundary condition $\nabla u(\partial\Omega) \subseteq \partial\Omega^*$ simplifies via the obliqueness lemma [37]: boundary points map to boundary points on the same face. Writing the Brenier potential as $u(x) = \frac{1}{2}\|x\|^2 + \varphi(x)$, where φ is the Kantorovich potential, the boundary condition reduces to the homogeneous Neumann condition:

$$\frac{\partial \varphi}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega, \tag{11}$$

where $\mathbf{n}$ is the outward unit normal. This simplification is what makes the FFT-OT solver particularly efficient on rectangular domains.

Fixed-Point Iteration via FFT. Lei and Gu [21] showed that Eq. (10) can be reformulated as a fixed-point problem. Using the algebraic identity $(u_{xx} + u_{yy})^2 = u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f/g \circ \nabla u$, the equation becomes a Poisson equation for u:

$$\Delta u = \sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f/g \circ \nabla u}. \tag{12}$$

Substituting $u = \frac{1}{2}\|x\|^2 + \varphi$ and defining the fixed-point operator $\mathcal{P}$, the iteration $\varphi^{(n+1)} \leftarrow \mathcal{P}[\varphi^{(n)}]$ requires solving a Poisson equation at each step:

$$\mathcal{P}[\varphi^{(n)}] := \Delta^{-1}\left\{\sqrt{[\varphi_{xx}^{(n)} + 1]^2 + [\varphi_{yy}^{(n)} + 1]^2 + 2[\varphi_{xy}^{(n)}]^2 + \frac{2f}{g \circ (\mathrm{Id} + \nabla\varphi^{(n)})}} - 2\right\}. \tag{13}$$

Since the domain is rectangular and the boundary condition is Neumann, Δ^{-1} is computed exactly via the Discrete Cosine Transform (DCT) in $O(N \log N)$ time, where $N = HW$. Convergence is declared when $\|\varphi^{(n+1)} - \varphi^{(n)}\|_2 < \varepsilon$. The final transport map is $T = \mathrm{Id} + \nabla\varphi$.

B FFT-OT Algorithm

Algorithm 1 DCT Poisson Equation Solver

Require: Matrix $\rho(i, j)$ on rectangular grid, $(i, j) \in \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}$
Ensure: Solution u to $\Delta u = \rho$ subject to $\partial u / \partial \mathbf{n} = 0$

1: Add offset c to ρ such that $\int_\Omega \rho = 0$ ▷ compatibility condition
2: $\tilde{\rho} \leftarrow \mathrm{DCT}(\rho)$ ▷ via FFT, $O(N \log N)$
3: Compute $\tilde{u}(m, n)$ for all m, n:

$$\tilde{u}(m, n) = \frac{\tilde{\rho}(m, n)}{2\left[\cos\frac{m\pi}{M} + \cos\frac{n\pi}{N} - 2\right]}$$

4: Set $\tilde{u}(0, 0) \leftarrow 0$ ▷ normalize constant of integration
5: $u \leftarrow \mathrm{IDCT}(\tilde{u})$ ▷ via FFT, $O(N \log N)$
6: **return** u

Algorithm 2 FFT-OT: Screen-Space Transport Map

Require: Source density $f(i,j)$ and edge-guided target density $g(i,j)$ on $\Omega = [-1,1] \times [-1,1]$, $(i,j) \in \{0,\dots,M-1\} \times \{0,\dots,N-1\}$; threshold $\varepsilon > 0$
Ensure: Transport map $T = \mathrm{Id} + \nabla\varphi^*$ such that $T_{\#}f = g$
Initialize $\varphi^{(0)} \leftarrow 0$
$n \leftarrow 0$
repeat
  Copy $\varphi^{(n)}$ to ghost cells from nearest interior cells ▷ enforce Neumann BC
  Compute first-order derivatives $\varphi_x^{(n)}$, $\varphi_y^{(n)}$ and compute $g \circ [\mathrm{Id} + \nabla\varphi^{(n)}]$ via scan conversion
  Compute second-order derivatives $\varphi_{xx}^{(n)}$, $\varphi_{yy}^{(n)}$, $\varphi_{xy}^{(n)}$ via finite differences.
  Compute right-hand side $\rho^{(n)}$:

$$\rho^{(n)} = \sqrt{\left[\varphi_{xx}^{(n)} + 1\right]^2 + \left[\varphi_{yy}^{(n)} + 1\right]^2 + 2\left[\varphi_{xy}^{(n)}\right]^2 + \frac{2f}{g \circ [\mathrm{Id} + \nabla\varphi^{(n)}]}} - 2$$

  Solve $\Delta\varphi^{(n+1)} = \rho^{(n)}$ using Algorithm 1
  $n \leftarrow n + 1$
until $\|\varphi^{(n)} - \varphi^{(n-1)}\|_2 < \varepsilon$
return $T = \mathrm{Id} + \nabla\varphi^{(n)}$

C OT-3DGS Algorithm

Algorithm 3 EDGEGUIDEDDENSITY

Require: Image $I \in \mathbb{R}^{H \times W \times 3}$, $\sigma_{\mathrm{LoG}} = 1.0$, $\sigma_{\mathrm{blur}} = 5.0$, $\rho_0 = 0.2$, $\alpha = 8.0$, $h = 2/\max(H,W)$
Ensure: Normalized density map $\rho \in \mathbb{R}^{H \times W}$
$G \leftarrow 0.299\, I_R + 0.587\, I_G + 0.114\, I_B$ ▷ Convert to luminance
$G' \leftarrow \mathcal{G}_{\sigma_{\mathrm{LoG}}} * G$ ▷ Gaussian smoothing
$\mathbf{M} \leftarrow |\nabla^2 G'|$ ▷ Laplacian response (LoG)
$\mathbf{M} \leftarrow \mathbf{M}/\max(\mathbf{M})$ ▷ Normalize to $[0,1]$
$s \leftarrow \alpha \cdot \mathbf{M}^{1.5}$ ▷ Enhance strong edges
$\rho \leftarrow s + \rho_0$ ▷ Add base density
$\rho \leftarrow \mathcal{G}_{\sigma_{\mathrm{blur}}} * \rho$ ▷ Smooth density field
$\rho \leftarrow \dfrac{4\,\rho}{\sum_{i,j} \rho_{ij} \cdot h^2}$ ▷ Normalize total mass to 4 over $[-1,1]^2$
return ρ

This section provides a detailed description of the OT-3DGS training pipeline, supplementing the high-level overview in the main paper. We present two variants of the method: **Naive OT-3DGS** in Algorithm 4 and **Memory-Efficient OT-3DGS** in Algorithm 5. Both share the same underlying formulation but differ in how the transport maps are applied during training. In both variants, an offline precomputation phase is performed once per camera before training begins. The pipeline is thus divided into two phases: this offline precomputation stage, followed by an online training phase that follows the standard 3DGS optimization loop with an OT-guided, non-uniform photometric loss.

C.1 Phase 1: Offline Precomputation

For each training camera c_k, we compute an *edge-guided density* g_k from the original full-resolution image I_k using EDGEGUIDEDDENSITY in Algorithm 3. Computing at full resolution preserves fine edge detail that would be lost if the density were computed after downsampling; this is the key reason for working at the original image resolution at this stage.

Given g_k, we solve the Monge–Ampère equation $\det \nabla^2\varphi = g/\bar{g}$ for the Brenier potential φ_k using the FFT-OT solver in Algorithm 2. The resulting OT map $T_k = \mathrm{id} - \nabla\varphi_k$ pushes mass from the uniform distribution toward regions of high edge density, concentrating Gaussian placement where the scene contains the most high-frequency detail.

Algorithm 4 OT-3DGS: Optimal Transport Guided 3D Gaussian Splatting Training

Require: Cameras $\mathcal{C} = \{c_k\}$, images $\{I_k\}$, downsampling factor $s = 4$, loss weight $\lambda = 0.2$, iterations $N = 30{,}000$
Ensure: Trained 3D Gaussian scene $\mathcal{G}$

Phase 1: Offline Transport Precomputation
for each camera $c_k \in \mathcal{C}$ **do**
  $g_k \leftarrow \textsc{EdgeGuidedDensity}(I_k)$ ▷ Edge-aware density using Algorithm 3
  $\varphi_k \leftarrow \text{FFT-OT}(g_k)$ ▷ Solve for Brenier potential using Algorithm 2
  $\tilde{I}_k^{\text{gt}} \leftarrow I_k \circ \nabla\varphi_k$ ▷ Transport image to reach target density
  $(\varphi_k,\ \tilde{I}_k^{\text{gt}}) \leftarrow \textsc{Downsample}(\cdot, s)$ ▷ Match training resolution
end for
$\mathcal{G} \leftarrow \textsc{InitFromSfM}()$
Initialize Adam optimizer for Gaussian parameters
Phase 2: Training in Transported Image Space
for $i = 1$ to N **do**
  Sample camera $c_k \sim \mathcal{U}(\mathcal{C})$
  $\hat{I} \leftarrow \textsc{Rasterize}(\mathcal{G}, c_k)$ ▷ Render current estimate
  $\tilde{\hat{I}} \leftarrow \hat{I} \circ \nabla\varphi_k$ ▷ Apply precomputed transport map
  $\mathcal{L} \leftarrow (1-\lambda)\|\tilde{\hat{I}} - \tilde{I}_k^{\text{gt}}\|_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}(\tilde{\hat{I}}, \tilde{I}_k^{\text{gt}})$
  $\mathcal{L}.\textsc{backward}()$
  $\textsc{AdamStep}(\mathcal{G})$
  $\textsc{AdaptiveDensityControl}(\mathcal{G}, i)$
end for
return $\mathcal{G}$

We then apply T_k to the full-resolution image to obtain the transported ground truth $\tilde{I}_k^{\text{gt}} = I_k \circ \nabla\varphi_k$, and downsample both φ_k and $\tilde{I}_k^{\text{gt}}$ by $s = 4\times$ using bilinear and bicubic interpolation respectively, to match the training image resolution. All precomputed quantities are saved to disk; the per-camera cost is $\approx$2 seconds on a single RTX 4090 GPU.

C.2 Memory-Efficient Transport via Inverse Map

Directly applying the transformation T_k during training is memory-prohibitive. Instead, we use a more efficient "lookup" approach by precomputing the inverse map T_k^{-1} before training begins. This is done via a simple fixed-point scheme:

$$u^{j+1} = y + \nabla\varphi_k(u^j), \tag{14}$$

starting with $u^0 = y$. We run this for K steps on a lower-resolution version of the potential φ_k. The result, $u^K \approx T_k^{-1}$, is saved as a coordinate grid $\mathcal{G}_k$ (taking only $\approx$6 MB per camera). During training, instead of calculating the movement on the fly, the model simply uses this grid to "sample" the correct values via `F.grid_sample`. This process is extremely fast and avoids creating massive temporary buffers in memory.

Number of fixed-point iterations K. Table 2 evaluates the impact of the iteration count K on the *stump* scene from the Mip-NeRF360 [2] dataset. We observe that K=1 is insufficient, leading to visibly blurrier results. The reconstruction quality stabilizes quickly as K increases, with K=10 already achieving converged results that remain unchanged with further iterations. Consequently, we use K=10 in all experiments.

Peak GPU Memory. Table 3 reports the peak training memory across all seven Mip-NeRF360 [2] scenes. Our method remains within $1.6\times$ of standard 3DGS on average. Notably, on *Bonsai* and *Counter*, our method uses *less* memory, as OT-based transport concentrates Gaussians in informative regions, reducing the total number of 3D gaussian splats required.

Algorithm 5 Memory Efficient OT-Guided 3DGS Training Implementation

Require: Training cameras $\mathcal{C} = \{c_k\}$, images $\{I_k\}$, $s = 4$, $\lambda = 0.2$, $\beta = 0.5$, $N = 30{,}000$, $K = 10$, learning rates $\{\eta_\mu, \eta_c, \eta_o, \eta_s, \eta_q\}$
Ensure: Trained 3D Gaussian scene $\mathcal{G}$

Phase 1: Offline Precomputation
for each camera $c_k \in \mathcal{C}$ **do**
  $g_k \leftarrow \textsc{EdgeGuidedDensity}(I_k)$ ▷ Edge-aware density using Algorithm 3
  $\varphi_k^{\text{full}} \leftarrow \text{FFT-OT}(g_k)$ ▷ Brenier potential via FFT-OT using Algorithm 2
  $\tilde{I}_k^{\text{gt,full}} \leftarrow I_k \circ \nabla\varphi_k^{\text{full}}$ ▷ Transport image to reach target density
  $\varphi_k \leftarrow \textsc{Downsample}(\varphi_k^{\text{full}}, s)$; $\tilde{I}_k^{\text{gt}} \leftarrow \textsc{Downsample}(\tilde{I}_k^{\text{gt,full}}, s)$ ▷ Match training resolution
  $u^0 \leftarrow y$
  for $j = 1$ to K **do**
    $u^j \leftarrow y + \nabla\varphi_k(u^{j-1})$ ▷ Fixed-point inversion, $T_k^{-1} \approx u^K$
  end for
  $\mathcal{G}_k \leftarrow u^K$ ▷ Store inverse map as grid (≈ 6 MB per camera)
  $\tilde{I}_k^{\text{gt}} \leftarrow \textsc{GridSample}(\tilde{I}_k^{\text{gt}}, \mathcal{G}_k)$ ▷ Refine transported image using inverse map
end for
Phase 2: Training
$\mathcal{G} \leftarrow \textsc{InitFromSfM}()$
Initialize Adam for μ with η_μ ▷ Decay $1.6 \times 10^{-4} \to 1.6 \times 10^{-6}$
Initialize Adam for $\{c, o, s, q\}$ with $\{\eta_c, \eta_o, \eta_s, \eta_q\}$
for $i = 1$ to N **do**
  $c_k \sim \mathcal{U}(\mathcal{C})$
  $\hat{I} \leftarrow \textsc{Rasterize}(\mathcal{G}, c_k)$ ▷ Render at training resolution
  $\tilde{\hat{I}} \leftarrow \textsc{GridSample}(\hat{I}, \mathcal{G}_k)$ ▷ Apply precomputed transport map
  $\mathcal{L}_{\text{OT}} \leftarrow (1-\lambda)\|\tilde{\hat{I}} - \tilde{I}_k^{\text{gt}}\|_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}(\tilde{\hat{I}}, \tilde{I}_k^{\text{gt}})$
  if use_reg **then**
    $\mathcal{L}_{\text{reg}} \leftarrow \sum_{(i,j)\in\mathcal{N}} w_{ij} \frac{(\hat{I}_i - \hat{I}_j)^2}{\hat{I}_i + \hat{I}_j + \gamma|\hat{I}_i - \hat{I}_j| + \varepsilon}$ ▷ $\gamma = 0.5$, $\varepsilon = 10^{-6}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{OT}} + \beta\,\mathcal{L}_{\text{reg}}$
  else
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{OT}}$
  end if
  $\mathcal{L}.\textsc{backward}()$; $\textsc{AdamStep}(\mathcal{G})$
  if $500 \le i \le 15{,}000$ **and** $i \bmod 100 = 0$ **then**
    $\textsc{DensifyAndPrune}(\mathcal{G})$
  end if
  if $i \bmod 3{,}000 = 0$ **then**
    $\textsc{ResetOpacity}(\mathcal{G})$
  end if
end for
return $\mathcal{G}$

C.3 Metrics

Laplacian Variance (LV). We measure image sharpness using the variance of the discrete Laplacian response, which captures the concentration of high-frequency energy:

$$\mathrm{LV}(\hat{I}) = \frac{1}{N}\sum_{u,v}\Big(\nabla^2\hat{I}(u,v) - \overline{\nabla^2\hat{I}}\Big)^2, \tag{15}$$

where the discrete Laplacian is defined as

$$\nabla^2\hat{I}(u,v) = \hat{I}(u{+}1,v) + \hat{I}(u{-}1,v) + \hat{I}(u,v{+}1) + \hat{I}(u,v{-}1) - 4\hat{I}(u,v), \tag{16}$$

and N denotes the number of pixels. Higher LV indicates stronger high-frequency content and sharper reconstructions.

Table 2: **Ablation on fixed-point iterations K.** Evaluation of OT map approximation on the *stump* scene [2]. We observe that $K = 10$ is sufficient to achieve perceptual quality, as BRISQUE, NIQE, and Laplacian Variance (LV) have converged, indicating stabilized sharpness and structural detail.

	Reference-Based			No-Reference		
K	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑
1	23.03	0.594	0.321	20.66	2.62	604
5	25.93	0.747	0.238	17.50	2.31	829
10	25.95	0.747	0.237	**17.39**	2.33	**833**
20	**26.00**	**0.749**	**0.235**	17.57	**2.32**	827

Table 3: **Quantitative comparison with 3DGS [18] on peak GPU memory usage.** Peak training memory in MB across Mip-NeRF360 [2] scenes. The ratio column denotes memory overhead relative to standard 3DGS [18] at $1.00\times$. Our method maintains low memory overhead, with an average ratio of $1.15\times$, by using precomputed inverse maps to avoid dense transport during training. All experiments are conducted on a single RTX 4090 GPU.

Scene	3DGS (MB)	Ours (MB)	Ratio
Bicycle	3848	4276	$1.11\times$
Bonsai	1251	**974**	$\mathbf{0.78}\times$
Counter	1137	**872**	$\mathbf{0.77}\times$
Garden	2505	3012	$1.20\times$
Kitchen	**1022**	1273	$1.25\times$
Room	**683**	1013	$1.48\times$
Stump	2818	3783	$1.34\times$
Mean	1895	2172	$1.15\times$

C.4 Phase 2: Training Loop

Implementation Details. The training loop closely follows standard 3DGS [18], with one modification to the photometric loss. At each iteration a camera c_k is sampled uniformly, the current Gaussian scene $\mathcal{G}$ is rasterized to produce a rendered image $\hat{I}$, and the OT map is applied via a single `F.grid_sample` call: $\tilde{\hat{I}} = \texttt{grid_sample}(\hat{I}, \mathcal{G}_k)$. The loss is computed between $\tilde{\hat{I}}$ and the precomputed transported ground truth $\tilde{I}_k^{\text{gt}}$:

$$\mathcal{L}_{\text{OT}} = (1-\lambda)\,\|\tilde{\hat{I}} - \tilde{I}_k^{\text{gt}}\|_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}(\tilde{\hat{I}},\, \tilde{I}_k^{\text{gt}}), \quad \lambda = 0.2.$$

where densification and opacity reset follow the original 3DGS [18] schedule: densify every 100 iterations from step 500 to 15,000; reset opacity every 3,000 iterations.

Performance Across Checkpoints. Table 4 compares OT-3DGS and standard 3DGS [18] across training checkpoints on the *Train Station* and *Tree* scenes from NeRF On-the-Go [32]. OT-3DGS consistently achieves lower BRISQUE and NIQE throughout training, indicating improved perceptual quality, and yields substantially higher Laplacian Variance, reflecting sharper reconstructions. At 30k iterations on *Train Station*, OT-3DGS significantly reduces perceptual distortion and improves structural sharpness by $4.6\times$ in Laplacian Variance. On *Tree*, the improvement is even more pronounced, with further reductions in BRISQUE and NIQE and a $6.3\times$ increase in Laplacian Variance, demonstrating consistently sharper and more perceptually faithful results.

C.5 Optional Relative Difference Prior (RDP) Regularization

An optional regularization term $\mathcal{L}_{\text{reg}}$ from Section 3 can be added to the loss $\mathcal{L} = \mathcal{L}_{\text{OT}} + \beta\,\mathcal{L}_{\text{reg}}$. Table 5 shows that β=0.0 yields the best perceptual results. As β increases, the reconstructions become smoother but increasingly over-regularized, leading to the loss of high-frequency details and reduced sharpness. Therefore, we set β=0.0 by default to maintains better pecetual quality.

Table 4: **Training iterations comparison.** Standard 3DGS vs. OT-3DGS across training on the *Train Station* and *Tree* scenes from NeRF On-the-Go [32]. We observe consistent improvements in BRISQUE and NIQE as the number of training iterations increases, with OT-3DGS maintaining better perceptual quality and sharper structural details throughout training compared to the 3DGS [18].

			Reference-Based			No-Reference			
Scene	Method	Iter	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑	#Gaussians
Train Station	3DGS	1k	13.76	0.549	**0.565**	84.96	**9.67**	**3.8**	86,881
	OT	1k	**14.03**	**0.551**	0.570	**83.45**	9.71	3.7	88,139
	3DGS	5k	**16.29**	**0.604**	**0.503**	65.96	6.25	16.4	242,155
	OT	5k	15.98	0.597	0.504	**60.38**	**5.57**	**32.7**	237,078
	3DGS	10k	16.14	**0.612**	0.475	56.67	5.02	35.6	499,376
	OT	10k	**16.58**	0.611	**0.467**	**48.44**	**4.25**	**88.5**	453,451
	3DGS	15k	16.57	**0.616**	0.458	49.38	4.41	58.7	707,899
	OT	15k	**16.64**	0.605	**0.455**	**39.25**	**3.72**	**185.4**	631,035
	3DGS	20k	**16.74**	**0.615**	**0.448**	44.49	4.07	81.6	707,899
	OT	20k	16.66	0.595	0.451	**32.88**	**3.44**	**302.8**	631,035
	3DGS	30k	**16.61**	**0.609**	**0.446**	39.93	3.86	105.7	707,899
	OT	30k	16.58	0.581	0.452	**27.88**	**3.29**	**486.9**	631,035
Tree	3DGS	1k	13.04	0.253	**0.726**	**80.92**	8.73	9.1	83,377
	OT	1k	**13.14**	**0.256**	**0.726**	82.56	**8.69**	**10.8**	84,003
	3DGS	5k	13.90	0.356	0.593	30.64	4.35	300	697,992
	OT	5k	**14.14**	**0.368**	**0.575**	**25.80**	**4.34**	**963**	831,495
	3DGS	10k	14.07	0.381	0.518	23.55	3.34	489	1,653,484
	OT	10k	**14.44**	**0.387**	**0.499**	**23.41**	**3.19**	**2063**	1,838,583
	3DGS	15k	14.06	**0.388**	0.484	21.88	2.82	659	2,059,447
	OT	15k	**14.30**	0.378	**0.471**	**20.91**	**2.54**	**3198**	2,241,575
	3DGS	20k	14.01	**0.384**	0.473	20.76	2.53	801	2,059,447
	OT	20k	**14.08**	0.359	**0.465**	**18.74**	**2.21**	**4484**	2,241,575
	3DGS	30k	**13.92**	**0.372**	0.469	20.92	2.35	1013	2,059,447
	OT	30k	13.78	0.334	**0.468**	**16.91**	**2.10**	**6412**	2,241,575

Table 5: **Regularization weight β.** Ablation on β for the *Statue* scene from RobustNeRF [33], where the best perceptual quality and structural sharpness are achieved at β=0. Lower BRISQUE and NIQE indicate better perceptual quality, while higher Laplacian Variance (LV) reflects greater sharpness.

	Reference-Based			No-Reference		
β	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑
0.0	19.15	0.671	0.282	**21.66**	**2.75**	**1587**
0.1	19.50	0.691	0.267	21.81	2.76	1303
0.5	**19.92**	0.717	**0.257**	23.72	2.93	797
1.0	19.59	**0.720**	0.268	24.72	3.08	527

C.6 Choice on Edge-Guided Density Function

The OT transport map T_k depends on the density function g_k used for the target measure. We compare four configurations on *Patio High* from NeRF On-the-Go [32]: **Uniform** ($g_k \equiv 1$) as an isotropic baseline, **Sobel** gradient magnitude (σ=5, base 0.2), **LoG (ours)** absolute Laplacian-of-Gaussian (σ=1), and **Laplacian Variance** (9×9 neighborhood). As shown in Table 6, LoG achieves the best performance across all no-reference metrics (BRISQUE, NIQE, and LV). Sobel improves over Uniform but tends to over-densify broad regions, while Laplacian Variance yields weaker

performance in capturing fine structural details. Our framework can accommodate task-specific density functions. In practice, σ=1 LoG provides the most consistent results in preserving structural edges and high-frequency detail, and is therefore used as the default.

Table 6: **Edge-guided density function ablation.** Comparison of density functions g_k on the *Patio High* scene from NeRF On-the-Go [32], where lower BRISQUE/NIQE and higher LV (Laplacian Variance) scores indicate better perceptual quality and structural sharpness.

	Reference-Based			No-Reference		
Density	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑
Uniform	14.37	0.391	0.550	36.31	4.06	182
Sobel	14.23	0.383	0.540	25.32	3.18	**329**
LoG ($\sigma = 0.5$)	14.28	0.386	0.541	24.16	3.17	309
LoG ($\sigma = 1.0$)	**14.44**	**0.395**	**0.528**	**22.79**	**3.04**	316
LoG ($\sigma = 2.0$)	14.36	0.389	0.536	23.51	3.11	312
LoG ($\sigma = 3.0$)	14.18	0.378	0.552	25.84	3.29	286
LoG ($\sigma = 8.0$)	14.02	0.369	0.579	32.76	3.78	224
Lap. Var.	13.83	0.361	0.563	27.48	3.35	307

C.7 Grid-Sampled OT vs. Naive OT

We compare standard 3DGS and our OT variants (Naive OT and grid-sample OT) in Table 7. Grid-sample OT achieves comparable perceptual quality to our Naive OT, improving BRISQUE and NIQE relative to standard 3DGS, with only a slightly lower structural sharpness. Importantly, it significantly reduces memory usage, enabling more efficient and scalable training.

Table 7: **Grid-sample vs. Naive OT.** Comparison of standard 3DGS and our OT variants (Naive OT and grid-sample OT) on *Corner* and *Patio High* [32]. Naive OT achieves the best perceptual quality, while grid-sample OT remains comparable (lower BRISQUE/NIQE than 3DGS) with similar structural sharpness (LV), and offers significantly reduced memory usage.

		Reference-Based			No-Reference		
Scene	Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑
Corner	Standard 3DGS	19.76	**0.619**	0.392	29.88	3.41	323
	Ours (Naive OT)	**19.83**	0.611	**0.362**	13.23	**2.51**	**1228**
	Ours (grid-sampled OT)	19.32	0.571	0.400	**12.92**	**2.51**	1215
Patio High	Standard 3DGS	**15.06**	**0.420**	0.518	28.32	3.62	123
	Ours (Naive OT)	14.15	0.381	0.531	17.10	2.76	**559**
	Ours (grid-sampled OT)	14.64	0.397	**0.515**	**16.50**	**2.73**	537

D Quantitative Results

Table 8: **Quantitative comparison with 3DGS [18] on the RobustNeRF dataset.** Our method consistently achieves better perceptual quality, as indicated by lower BRISQUE and NIQE and higher Laplacian Variance (LV), while maintaining comparable peak GPU memory usage to 3DGS.

		Reference-Based			*No-Reference*			*Memory Usage*
Scene	Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑	Peak (MB)↓
Android	3DGS (Retrained)	**24.24**	**0.809**	**0.267**	28.31	3.87	124	**1294**
	Ours w/o reg	23.36	0.757	0.305	**8.91**	**2.70**	**566**	1987
	Ours w/ reg	23.92	0.782	0.286	14.39	2.91	299	1901
Crab 1	3DGS (Retrained)	**22.15**	**0.752**	**0.361**	44.12	4.45	63	**371**
	Ours w/o reg	21.32	0.715	0.382	**29.11**	**3.40**	**326**	490
	Ours w/ reg	22.07	0.733	0.370	32.65	3.52	164	546
Crab 2	3DGS (Retrained)	**23.90**	**0.788**	**0.371**	57.64	5.58	28	**315**
	Ours w/o reg	23.63	0.772	0.377	**42.69**	**4.39**	**137**	406
	Ours w/ reg	23.78	0.779	0.371	45.96	4.60	60	460
Statue	3DGS (Retrained)	**19.95**	**0.744**	**0.250**	24.32	3.20	397	**1370**
	Ours w/o reg	19.26	0.674	0.278	**23.01**	**2.77**	**1538**	1949
	Ours w/ reg	19.70	0.714	0.260	23.44	2.92	802	1782
Yoda	3DGS (Retrained)	**24.41**	**0.778**	0.375	52.96	4.75	37	**288**
	Ours w/o reg	23.91	0.754	0.383	**34.81**	**3.79**	**341**	378
	Ours w/ reg	24.38	0.767	**0.372**	37.76	3.80	111	417

Table 9: **Quantitative comparison with 3DGS [18] on the NeRF On-the-Go [32] dataset.** Our method consistently achieves improved perceptual quality, as indicated by lower BRISQUE and NIQE and higher Laplacian Variance (LV), while maintaining comparable peak GPU memory usage to 3DGS.

		Reference-Based			*No-Reference*			*Memory Usage*
Scene	Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑	Peak (MB)↓
Arcdetriomphe	3DGS (Retrained)	**20.35**	**0.796**	**0.199**	12.64	4.07	1012	561
	Ours w/o reg	19.40	0.765	0.218	**11.74**	**3.90**	**1821**	589
	Ours w/ reg	19.71	0.774	0.217	11.82	4.08	1035	**518**
Corner	3DGS (Retrained)	**20.44**	**0.6792**	**0.3258**	29.88	3.41	323	1330
	Ours w/o reg	19.87	0.6195	0.3577	**12.92**	**2.51**	**1215**	1252
	Ours w/ reg	20.34	0.6507	0.3426	19.26	2.75	470	**1168**
Drone	3DGS (Retrained)	15.04	**0.4818**	0.5316	23.07	3.51	194	**327**
	Ours w/o reg	15.08	0.4480	0.5353	**13.79**	**2.63**	**889**	384
	Ours w/ reg	**15.38**	0.4717	**0.5281**	19.35	2.95	266	440
Fountain	3DGS (Retrained)	**13.93**	**0.2878**	0.5813	26.72	2.90	308	568
	Ours w/o reg	13.33	0.2530	**0.5780**	**22.32**	**2.39**	**1366**	583
	Ours w/ reg	13.52	0.2695	0.5875	28.55	2.95	365	**558**
Mountain	3DGS (Retrained)	**14.01**	**0.3582**	**0.5389**	16.93	2.55	535	568
	Ours w/o reg	13.74	0.3340	0.5407	**15.40**	**2.03**	**2046**	629
	Ours w/ reg	13.65	0.3431	0.5651	18.80	2.54	603	**537**
Patio	3DGS (Retrained)	**12.50**	**0.3874**	**0.4995**	22.98	3.60	579	1049
	Ours w/o reg	11.74	0.2813	0.5404	**20.39**	**3.51**	**4265**	900
	Ours w/ reg	12.34	0.3545	0.5299	26.85	3.76	757	**851**
Patio High	3DGS (Retrained)	15.06	0.4196	0.5177	28.32	3.62	123	**757**
	Ours w/o reg	14.64	0.3975	0.5154	**17.10**	**2.73**	**537**	857
	Ours w/ reg	**15.57**	**0.4324**	**0.4922**	21.41	2.99	218	768
Spot	3DGS (Retrained)	15.38	0.4256	0.5528	**16.00**	3.10	249	1100
	Ours w/o reg	15.39	0.4209	**0.5313**	16.16	**2.87**	**1854**	**997**
	Ours w/ reg	**15.48**	**0.4259**	0.5502	21.03	3.26	421	1007
Statue	3DGS (Retrained)	**14.54**	**0.5587**	0.4914	16.74	2.97	258	812
	Ours w/o reg	14.22	0.5171	**0.4877**	**10.62**	**2.47**	**1492**	721
	Ours w/ reg	14.47	0.5439	0.4967	17.14	3.03	254	**599**
Train	3DGS (Retrained)	**18.66**	**0.6321**	**0.3634**	43.29	4.12	203	**1034**
	Ours w/o reg	17.63	0.5713	0.3908	**30.68**	**3.06**	**911**	1122
	Ours w/ reg	18.19	0.6031	0.3823	36.29	3.46	349	1085
Train Station	3DGS (Retrained)	16.85	**0.6114**	**0.4434**	39.13	3.85	106	561
	Ours w/o reg	16.52	0.5844	0.4508	**28.19**	**3.27**	**474**	530
	Ours w/ reg	**16.87**	0.6035	0.4453	33.89	3.38	142	**505**
Tree	3DGS (Retrained)	14.26	**0.3787**	**0.4578**	**17.15**	2.32	1023	1631
	Ours w/o reg	13.41	0.3280	0.4778	20.27	**2.09**	**6136**	1726
	Ours w/ reg	**14.52**	0.3768	0.4825	23.12	2.63	1183	**1299**

Table 10: **Quantitative comparison with 3DGS [18]on the Mip-NeRF360 [2] dataset (Part I).** Our method demonstrates strong perceptual quality, as reflected by BRISQUE, NIQE, and Laplacian Variance (LV). Notably, for prior methods, the trends in these perceptual metrics are consistent with reference-based metrics, indicating that no-reference metrics alone can reliably rank method performance as well. Peak GPU memory usage is also reported.

		Reference-Based			*No-Reference*			*Memory Usage*
Scene	Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑	Peak (MB)↓
Bicycle	Plenoxels†	21.912	0.496	0.506	55.26	6.04	45	–
	iNGP†	22.193	0.491	0.487	25.86	4.21	568	–
	iNGP-big†	22.171	0.512	0.446	27.27	4.03	635	–
	MipNeRF360†	24.370	0.685	0.301	28.03	2.55	699	–
	3DGS†	25.246	**0.771**	**0.205**	17.45	2.05	1073	–
	3DGS (retrained)	**25.280**	0.767	0.208	17.05	2.09	1106	3848
	Ours (w/o reg)	23.070	0.709	0.253	**16.64**	**2.03**	**2003**	4276
	Ours (w/ reg)	23.700	0.725	0.251	19.89	2.29	1012	**3657**
Bonsai	Plenoxels†	24.669	0.814	0.398	57.90	7.12	14	–
	iNGP†	30.337	0.890	0.227	32.31	5.12	135	–
	iNGP-big†	30.685	0.906	0.205	32.08	4.80	123	–
	MipNeRF360†	**33.460**	0.941	0.176	31.12	4.18	90	–
	3DGS†	32.780	0.938	0.129	26.81	3.71	348	–
	3DGS (retrained)	32.780	**0.956**	**0.095**	25.86	3.67	356	1251
	Ours (w/o reg)	29.920	0.930	0.120	**25.68**	**3.42**	**909**	974
	Ours (w/ reg)	31.240	0.944	0.110	29.27	3.70	425	**958**
Counter	Plenoxels†	23.624	0.759	0.441	62.43	7.52	12	–
	iNGP†	26.439	0.798	0.342	18.83	6.12	152	–
	iNGP-big†	26.691	0.817	0.306	16.11	5.76	156	–
	MipNeRF360†	**29.550**	0.894	0.204	25.13	3.74	93	–
	3DGS†	28.700	0.905	0.204	15.68	2.83	412	–
	3DGS (retrained)	29.430	**0.920**	**0.112**	14.81	2.77	425	1137
	Ours (w/o reg)	25.110	0.822	0.164	**14.46**	**2.58**	430	**872**
	Ours (w/ reg)	28.820	0.905	0.121	16.05	2.86	**460**	880
Flowers	Plenoxels†	20.130	0.432	0.506	50.28	6.21	38	–
	iNGP†	20.360	0.450	0.481	22.52	4.75	545	–
	iNGP-big†	20.670	0.486	0.441	25.58	4.38	663	–
	MipNeRF360†	**21.670**	0.584	0.346	27.44	2.87	774	–
	3DGS†	21.540	0.605	0.337	17.49	2.26	970	–
	3DGS (retrained)	21.230	**0.632**	**0.302**	20.89	2.25	1084	3358
	Ours (w/o reg)	20.900	0.621	0.307	**16.17**	**2.13**	**1856**	3397
	Ours (w/ reg)	21.320	0.628	0.311	19.19	2.38	1011	**2944**
Garden	Plenoxels†	23.495	0.606	0.386	47.43	5.01	102	–
	iNGP†	24.599	0.649	0.312	29.10	3.33	557	–
	iNGP-big†	25.069	0.701	0.257	26.45	3.03	682	–
	MipNeRF360†	26.980	0.813	0.170	18.29	2.17	862	–
	3DGS†	27.410	0.868	**0.103**	11.65	1.99	1125	–
	3DGS (retrained)	**27.500**	**0.868**	0.106	11.27	2.09	1150	**2505**
	Ours (w/o reg)	25.650	0.828	0.148	**11.22**	**1.98**	**1984**	3012
	Ours (w/ reg)	26.300	0.837	0.144	13.72	2.25	1153	2672

Table 11: **Quantitative comparison with 3DGS [18]on the Mip-NeRF360 [2] dataset (Part II).** Our method demonstrates strong perceptual quality, as reflected by BRISQUE, NIQE, and Laplacian Variance (LV). Notably, for prior methods, the trends in these perceptual metrics are consistent with reference-based metrics, indicating that no-reference metrics alone can reliably rank method performance as well. Peak GPU memory usage is also reported.

		Reference-Based			*No-Reference*			*Memory Usage*
Scene	Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓	LV↑	Peak (MB)↓
Kitchen	Plenoxels†	23.420	0.648	0.447	57.59	7.24	12	–
	iNGP†	28.548	0.818	0.254	**27.05**	5.38	310	–
	iNGP-big†	29.479	0.858	0.195	35.16	4.96	263	–
	MipNeRF360†	32.230	0.920	0.127	40.84	4.04	163	–
	3DGS†	30.317	0.922	0.129	34.49	3.12	805	–
	3DGS (retrained)	**32.260**	**0.948**	**0.067**	32.06	3.09	826	**1022**
	Ours (w/o reg)	27.890	0.883	0.104	30.73	**2.80**	**2588**	1273
	Ours (w/ reg)	30.210	0.914	0.088	37.54	3.08	1251	1133
Room	Plenoxels†	27.594	0.842	0.419	63.64	7.35	14	–
	iNGP†	29.269	0.855	0.301	40.96	5.12	73	–
	iNGP-big†	29.690	0.871	0.261	36.63	4.61	70	–
	MipNeRF360†	31.630	0.913	0.211	34.85	4.37	61	–
	3DGS†	30.632	0.914	0.220	23.68	3.50	241	–
	3DGS (retrained)	**32.230**	**0.943**	**0.112**	21.95	3.38	249	**683**
	Ours (w/o reg)	30.500	0.922	0.135	**18.84**	**3.33**	**535**	1013
	Ours (w/ reg)	31.040	0.929	0.129	21.91	3.46	274	982
Stump	Plenoxels†	20.661	0.523	0.503	54.18	6.56	26	–
	iNGP†	23.626	0.574	0.450	33.15	5.46	330	–
	iNGP-big†	23.466	0.594	0.421	33.38	5.25	375	–
	MipNeRF360†	26.400	0.744	0.261	31.49	3.22	348	–
	3DGS†	26.550	**0.775**	**0.210**	23.42	2.69	430	–
	3DGS (retrained)	**26.640**	0.770	0.216	22.51	2.64	454	**2818**
	Ours (w/o reg)	25.950	0.747	0.237	**17.39**	**2.33**	**833**	3783
	Ours (w/ reg)	26.290	0.756	0.232	21.79	2.59	514	3533
Treehill	Plenoxels†	22.280	0.510	0.495	54.60	6.14	32	–
	iNGP†	22.380	0.518	0.489	16.16	4.42	232	–
	iNGP-big†	22.390	0.542	0.450	15.07	4.17	296	–
	MipNeRF360†	**22.960**	0.631	0.347	20.23	2.46	442	–
	3DGS†	22.510	0.638	**0.317**	17.72	2.11	652	–
	3DGS (retrained)	22.730	**0.639**	0.319	19.81	2.23	637	3621
	Ours (w/o reg)	22.330	0.626	0.336	**14.37**	**2.07**	**1066**	3883
	Ours (w/ reg)	22.800	0.633	0.339	20.37	2.30	541	**3329**

E Qualitative Results

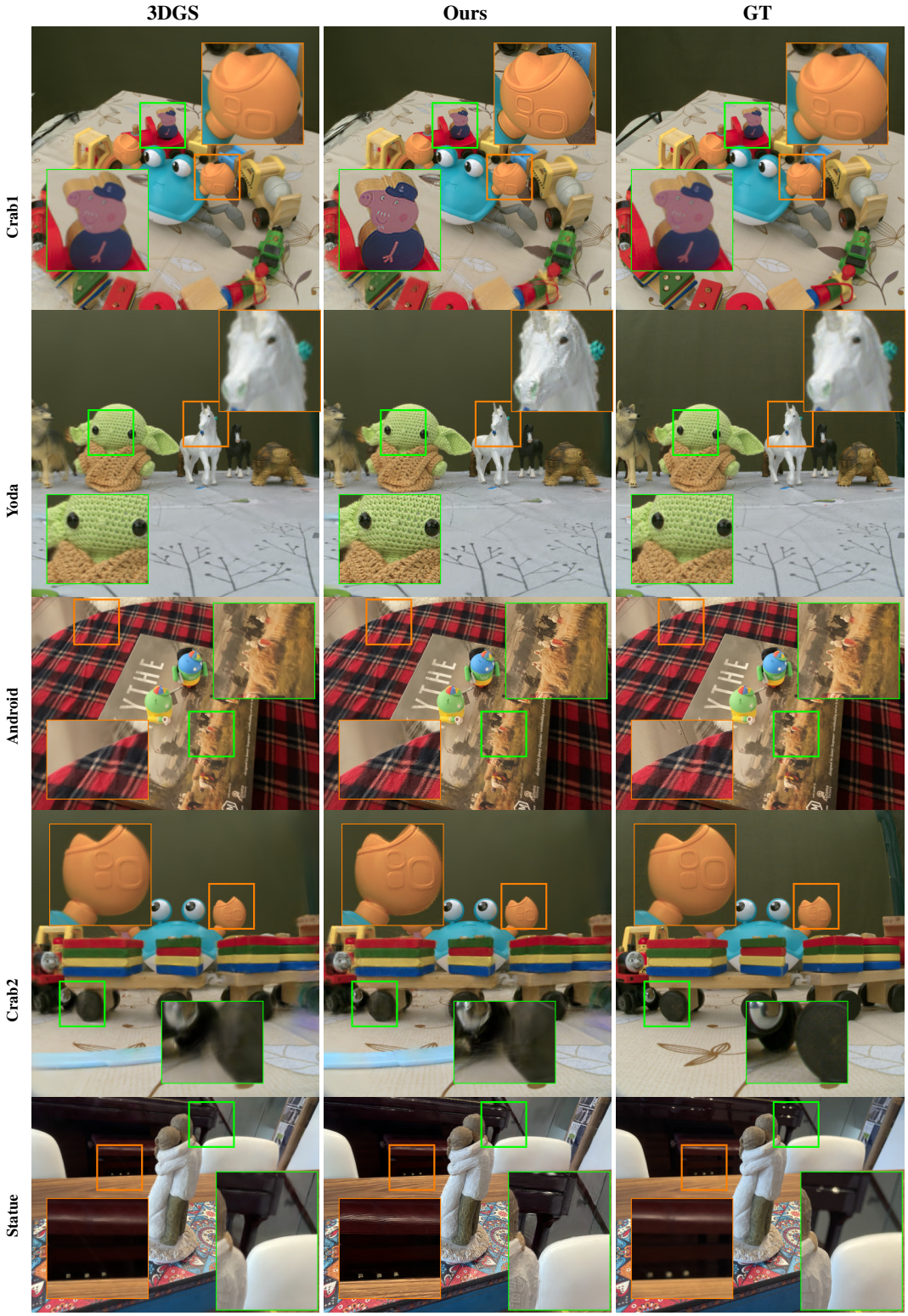


Figure 7: **Visual comparison with 3DGS [18] on RobustNeRF [33] scenes.** Our method achieves higher perceptual quality, with sharper edges, clearer fine structures, and improved reflectance in zoomed-in object regions, while 3DGS exhibits noticeable blurring and over-smoothed details.



Figure 8: **Visual comparison with 3DGS [18] on the NeRF On-the-Go [32] dataset.** Our method with non-uniform supervision better retains high-frequency structures which occupy small pixel regions, such as sharp edges and small text, while 3DGS often struggles under uniform supervision. Here, the ground truth (GT) refers to standard $4\times$ downsampled high-resolution images.

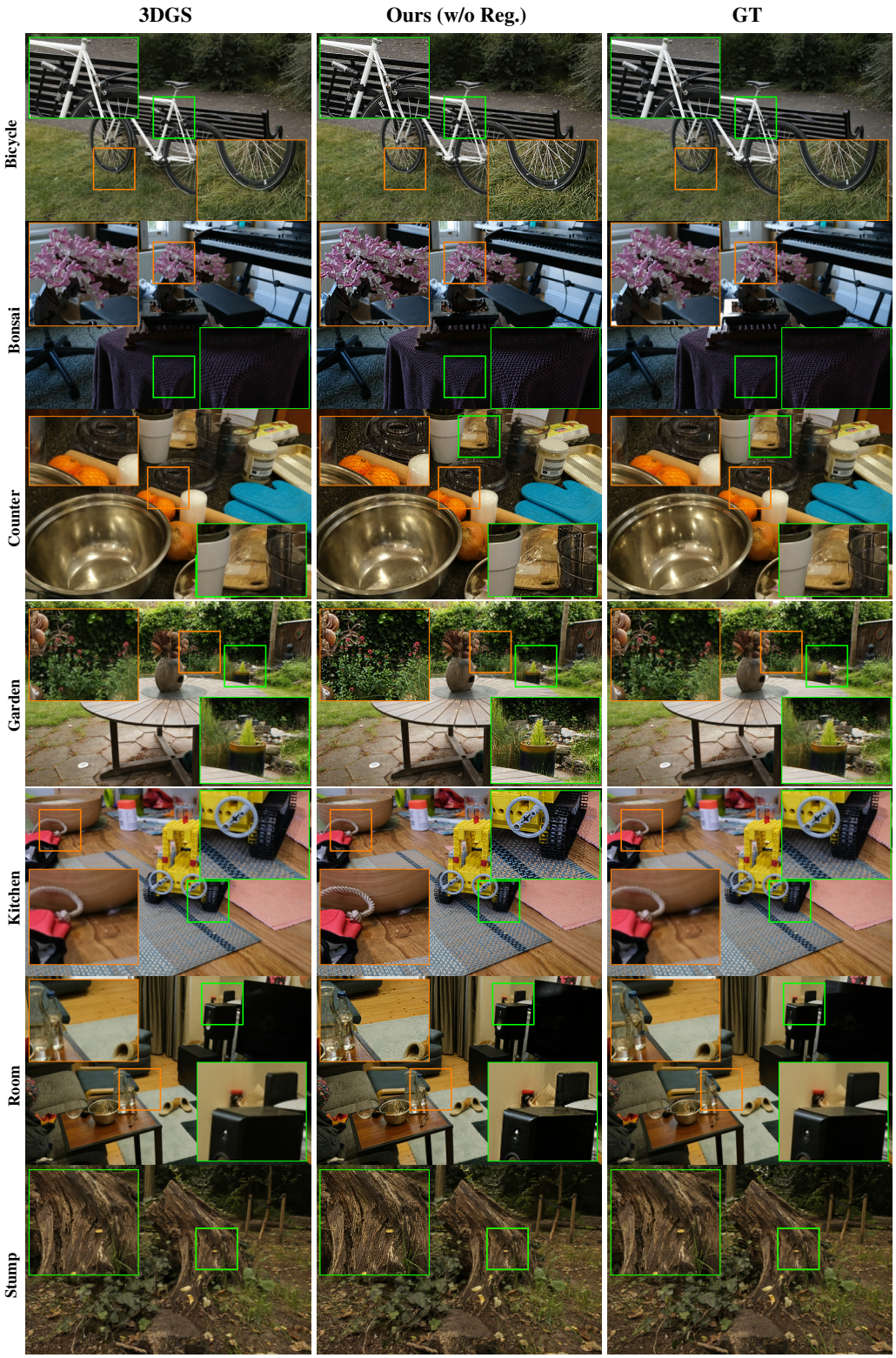


Figure 9: **Qualitative comparison with 3DGS [18] on the Mip-NeRF 360 [2] dataset.** Our method achieves higher perceptual quality with sharper structural details through geometry-aware, non-uniform supervision, improving representation of structures occupying small pixel regions compared to 3DGS [18]. Here, GT denotes standard $4\times$ downsampled high-resolution images.

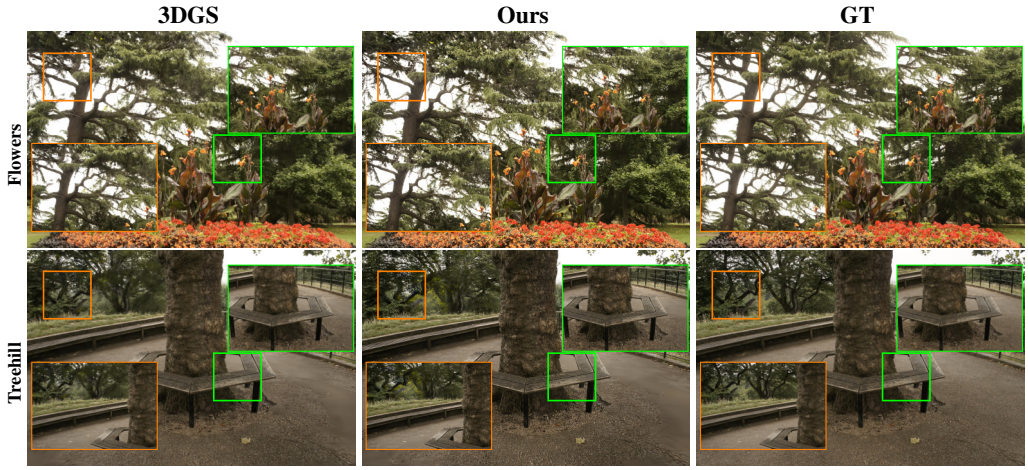


Figure 10: **Additional qualitative comparisons with 3DGS [18] on the Mip-NeRF 360 [2] dataset.** Our method preserves sharper structural details through geometry-aware, non-uniform supervision, particularly in regions containing fine structures. Here, GT denotes standard $4\times$ downsampled high-resolution images.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

Answer: [Yes]

Justification: Our claim made in the abstract and introduction accurately reflects the paper's contributions and scope. We futher demonstrate it in our methodology and experiment sections.

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: Yes, we discuss the limitation of our method in the main paper.

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 include theoretical results.

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: We provide all the information needed to reproduce the main experimental results in the main paper, supplementary materials.

   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: Our code will be released upon acceptance.

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: We provide experimental details in the main paper and supplementary material.

   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: This paper does not report error bars.

   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: We provide experimental details, including the computer resources needed to reproduce the experimental results in the main paper and supplementary material.

   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: This work conforms with 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: We provide discussion on the potential societal impacts of our method in the main paper.

    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 poses no such risks.

    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: We have cited the original owners of all assets used in this paper properly.

    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: That the paper does not release new assets.

    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 nor 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 crowdsourcing nor research with human subjects.

    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 development in this research does not involve LLMs as any important, original, or non-standard components.

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.