PERCEPTUAL WRAPPER WITH 3D-ANCHORED GAUSSIAN NOISE FOR 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) delivers impressive real-time rendering of 3D scenes but remains limited in its ability to represent high-frequency textures, especially under memory constraints and Rate-Distortion Optimization (RDO). Recent methods augment the Gaussian primitives with texture maps, improving the rendering quality given a fixed Gaussian count, but introduce substantial storage overhead. Towards this end, we present a general, model-agnostic *perceptual wrapper* that enhances 3DGS renderings via content- and view-adaptive texture synthesis. Our approach employs a lightweight per-scene network conditioned on the base rendering and 3D-anchored pseudo-random Gaussian noise to synthesize realistic textures, effectively offloading high-frequency appearance details from the underlying 3DGS representation. Guided by Wasserstein Distortion (WD), the network learns to match local feature statistics rather than enforcing pixel-wise fidelity, reducing the oversmoothing associated with traditional reconstruction objectives. Extensive subjective and objective evaluations across RDO and non-RDO 3DGS baselines demonstrate that our perceptual wrapper delivers improved perceptual quality with minimal overhead and substantially reduced storage requirements at comparable perceptual quality levels.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a powerful representation for static 3D scenes, enabling high-quality novel view synthesis at real-time rendering speeds. Its explicit representation, however, faces an inherent trade-off between reconstruction fidelity on the one hand and storage cost and rendering efficiency on the other. Because Gaussian primitives are inherently smooth, representing complex high-frequency textures often requires many primitives, increasing both model size and rendering cost. Consequently, extensive research has explored pruning, structured representations, quantization, and entropy coding across both non-Rate-Distortion Optimized (non-RDO) and Rate-Distortion Optimized (RDO) settings.

Despite significant advances in compact 3DGS, most methods are still optimized using pixel-oriented objectives such as L1 and structural similarity (SSIM). These objectives drive 3DGS models toward accurate pixel-level reconstruction of novel views, even when such fidelity offers limited perceptual benefit. In particular, under stringent storage constraints, these objectives tend to preserve coarse scene structure while sacrificing high-frequency details, producing renderings that may remain competitive in L1 and SSIM yet appear blurry or texture-deficient. This raises a fundamental question: *must every fine-grained texture detail be explicitly encoded by the 3DGS representation?*

Recent work has begun incorporating human perception into 3DGS. Perceptual-GS (Zhou & Ni, 2025), for example, uses human sensitivity to structural edges to guide densification but still relies on conventional L1 and SSIM. Other approaches increase Gaussian expressiveness using additional texture or appearance features (Chao et al., 2025), at the cost of extra parameters. In this work, we explore a new direction: representing coarse scene appearance with 3DGS while synthesizing perceptually plausible high-frequency details after rendering. This idea is inspired by perceptual image and video compression, where texture synthesis improves perceptual quality without enforcing exact pixel-wise reconstruction (Norkin & Birkbeck, 2018; Ameur et al., 2023). To enable perceptually plausible texture synthesis in the image domain, Ballé et al. (2025) introduce an overfitted image compression framework based on Wasserstein Distortion (WD) (Qiu et al., 2024), which prioritizes matching localized feature distributions over exact pixel-wise reconstruction. Under this framework, fine-grained texture details can be synthesized deterministically from a random seed shared between

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

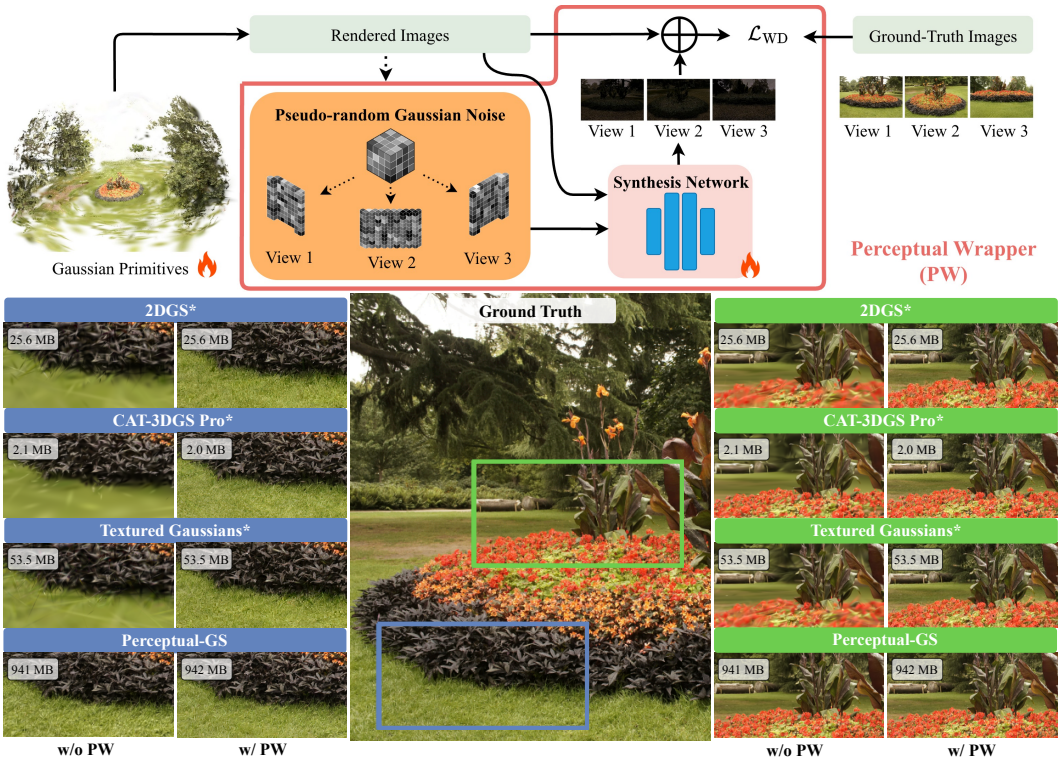


Figure 1: Our perceptual wrapper (PW) conditions a lightweight synthesis network on the base rendering and 3D-anchored noise to synthesize perceptually rich, view- and content-adaptive textures. As a model-agnostic framework, PW improves the perceptual quality of diverse GS representations with negligible storage or transmission overhead. Model sizes are shown in the top-left corner.

the encoder and decoder, a mechanism commonly referred to as *common randomness*. This property is especially appealing for compact 3DGS representations because these fine texture details no longer need to be explicitly encoded in Gaussian primitives.

However, texture synthesis for 3DGS representations presents four key challenges. **First**, 3DGS representations are typically *sparse* and *unstructured*, making direct texture synthesis in the 3DGS domain challenging. **Second**, in contrast to fixed-view image reconstruction, novel view synthesis requires texture synthesis to adapt to arbitrary unseen viewpoints. **Third**, synthesized textures and image details must remain *view-consistent*, ensuring that the same surface point remains coherent across viewpoints. **Fourth**, the diversity of existing 3DGS representations and their training strategies necessitates a generic texture synthesis solution that is agnostic to the underlying 3DGS model.

To address these challenges, we propose a lightweight, model-agnostic *perceptual wrapper* (PW) for texture synthesis in 3DGS systems. PW refines a rendered image using a compact, per-scene optimized convolutional neural network conditioned on both the initial rendering and multi-scale 3D-anchored Gaussian noise. Rather than using fixed image-space noise (Ballé et al., 2025), we anchor the randomness to the 3D scene geometry via a multi-scale 3D hash grid. As a result, observations of the same surface point are associated with consistent random noise features across viewpoints. The synthesis network and the 3DGS representation are jointly optimized using WD, enabling perceptually plausible, view-consistent texture synthesis with negligible storage overhead.

Our contributions are three-fold: (1) we introduce a perceptually optimized 3DGS framework that treats fine-grained, high-frequency visual details as synthesizable signals rather than information that must be explicitly encoded in Gaussian primitives (Fig. 2); (2) we propose a lightweight, model-agnostic *perceptual wrapper* that leverages WD (Sec. 3.1) and multi-scale 3D-anchored common randomness (Sec. 3.4) to achieve content-adaptive and view-consistent texture synthesis (Tables 7 and 8); and (3) through extensive objective and subjective evaluations, we demonstrate consistent perceptual quality improvements across multiple 3DGS backbones (Sec. 4.1), particularly under stringent storage constraints, as illustrated in Fig. 1.

Table 1: Comparison of 3DGS optimization paradigms.

Method	Training Objective	Perceptual Mechanism	Common Randomness	Applicable 3DGS Models
Standard 3DGS				
3DGS / Scaffold-GS	L1 + SSIM	No	No	Non-RDO
CAT-3DGS Pro / HAC++	L1 + SSIM + Rate	No	No	RDO
Texture-Augmented				
GSTex	L1 + SSIM	Local Color Map	No	Non-RDO
Textured Gaussian	L1 + SSIM	Multi-channel Map	No	Non-RDO
Perceptually-Optimized				
Perceptual-GS	L1 + SSIM	Densification	No	Non-RDO
WD-R	L1 + SSIM + WD (+ Rate)	Loss Function	No	RDO and Non-RDO
Ours	**WD (+ Rate)**	**2D Synthesis Network**	**Yes**	**RDO and Non-RDO**

2 Related Work

Compact and Efficient 3DGS. Existing compact 3DGS methods generally fall into two categories: (i) non-RDO and (ii) RDO approaches. Non-RDO methods reduce spatial redundancy either by structuring Gaussian primitives into hierarchical or anchor-based structures (Lu et al., 2024; Ren et al., 2025a; Sun et al., 2024) or by pruning and controlling densification (Mallick et al., 2024; Ren et al., 2025b; Hanson et al., 2025a; Zhang et al., 2025; Ali et al., 2024; Lee et al., 2024; Fan et al., 2024; Papantonakis et al., 2024; Chen et al., 2025b; Fang & Wang, 2024; Girish et al., 2024; Hanson et al., 2025b). In contrast, RDO methods explicitly balance rate and distortion through quantization, codebooks, and entropy models (Zhan et al., 2025; Chen et al., 2025a; Wang et al., 2024b; Liu et al., 2024a;b; Wang et al., 2024a; Chen et al., 2024). Despite their compression gains, both families still rely on standard Gaussian primitives with smoothly varying appearance, which limits their ability to model fine-grained textures within individual primitives. Consequently, reproducing high-frequency texture details requires a large number of Gaussian primitives. Under tight bit budgets, aggressive pruning and quantization significantly reduce the representation capacity available for texture modeling, leading to overly smoothed or blurry reconstructions. By decoupling high-frequency texture synthesis in rendered views from the underlying 3DGS representation, our approach is orthogonal to these methods and can be seamlessly integrated with both non-RDO and RDO frameworks.

Texture-Augmented 3DGS Representations. To overcome the limited expressiveness of standard Gaussian parameterization, recent works augment Gaussian primitives with spatially varying intra-primitive appearance. Explicit approaches associate discrete 2D color or feature patches with individual Gaussians. For example, GStex (Rong et al., 2025) attaches a local colormap to each Gaussian, while Textured Gaussians (Chao et al., 2025) and BBSplat (Svitov et al., 2025) map multi-channel planar textures onto Gaussians to model complex silhouette, color, and opacity variations. Alternatively, continuous implicit formulations such as SVGS (Xu et al., 2024) parameterize intra-primitive appearance using tiny Multi-Layer Perceptrons, movable kernels, or other mechanisms. While these methods can capture finer details with fewer primitives, they increase the attribute dimensionality of each Gaussian, thereby raising storage overhead on a per-Gaussian basis. In contrast, our framework preserves the standard Gaussian representation and synthesizes high-frequency appearance after rasterization, avoiding the need for per-Gaussian texture parameters.

Perceptually-Optimized 3DGS Representations. A parallel line of research improves perceptual quality through perceptually motivated optimization rather than pixel-level fidelity objectives such as L1 and SSIM. Perceptual-GS (Zhou & Ni, 2025) guides densification using perceptual sensitivity maps, GANeRF (Roessle et al., 2023) applies adversarial supervision to NeRFs, while WD-R (Özyılkan et al., 2026) applies Wasserstein Distortion to 3DGS training. Although such objectives often yield sharper textures, they do not introduce an explicit mechanism for synthesizing details beyond those represented by the underlying 3DGS. When the Gaussian count is strictly constrained, the underlying 3DGS representation simply lacks sufficient degrees of freedom to encode fine-grained, high-frequency details. In contrast, our method combines perceptual optimization with an explicit synthesis network and common randomness, as summarized in Table 1.

3 Method

3.1 Preliminary

Wasserstein Distortion. Unlike conventional 3DGS methods that minimize pixel-wise objectives, such as L1 and SSIM, our perceptual wrapper adopts *Wasserstein Distortion (WD)* (Qiu et al.,

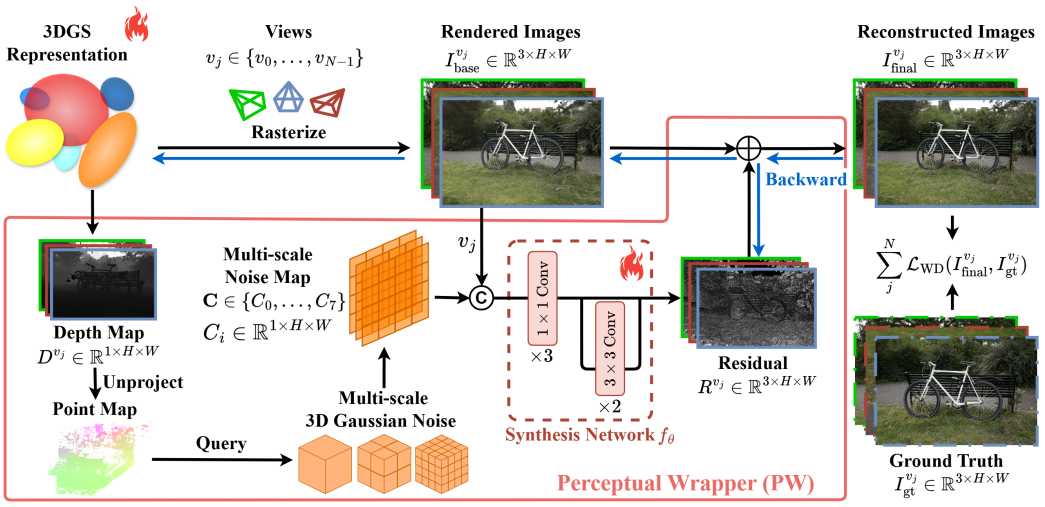


Figure 2: **Overview of the perceptual wrapper (PW) framework**. PW employs a synthesis network f_θ together with 3D-anchored Gaussian noise $\mathbf{C}$ to refine the initial rendering $I^{v_j}_{\text{base}}$ in a content- and view-dependent manner. The full system, including both the PW and the underlying 3DGS representation, is trained end-to-end by minimizing a WD loss $\mathcal{L}_{\text{WD}}$ across training views.

2024) as its primary training objective. WD is an image-to-image perceptual metric constructed from Wasserstein distances between *localized feature distributions*. In a nutshell, the WD between two images I_1, I_2 is computed as $\mathcal{L}_{\text{WD}}(I_1, I_2) = \sum_l \sum_{x,y} W_2^2(F^{(l)}_{I_1}(x,y), F^{(l)}_{I_2}(x,y))$, where $F^{(l)}_{I_1}(x,y), F^{(l)}_{I_2}(x,y)$ denote the local VGG features of these images at layer l centered at spatial location (x,y), and $W_2^2(\cdot,\cdot)$ computes the squared 2-Wasserstein distance between their Gaussian distributions. Rather than enforcing exact point-wise correspondence, WD compares feature statistics within local neighborhoods (of the features) of the reference and reconstructed images. As shown by Ballé et al. (2025), WD aligns more closely with human perceptual judgments than commonly used alternatives such as LPIPS (Zhang et al., 2018) or DISTS (Ding et al., 2022). See Appendix C for more details.

3.2 Method Overview

In this work, we introduce a versatile, model-agnostic perceptual wrapper (PW) framework that integrates perceptual optimization and texture synthesis into both the training and inference stages of per-scene 3DGS representations. Our goal is to overcome the limited ability of 3DGS models to represent highly textured regions under strict storage or transmission bandwidth constraints, where novel-view renderings often suffer from severe blurry artifacts and the loss of high-frequency details. Although texture synthesis under representation-cost constraints has been extensively studied in the image and video domains, its potential for 3DGS remains largely unexplored.

As illustrated in Fig. 2, our PW operates directly on images rendered by 3DGS. This design sidesteps the challenges of modeling and synthesizing textures over sparse, unstructured 3D Gaussians, while remaining fully compatible with existing 3DGS frameworks. PW integrates seamlessly with both non-RDO and RDO-based 3DGS methods, and can be trained end-to-end alongside these models without altering their specific parameterization of Gaussian primitives.

Our framework comprises two key components: (i) a lightweight, model-agnostic synthesis network that produces perceptually plausible high-frequency details conditioned on both the noise input and the initial rendering, and (ii) a multi-scale, 3D-anchored pseudo-random Gaussian noise that facilitates view-consistent texture synthesis. In addition, we replace conventional pixel-fidelity objectives, such as L1 and SSIM, with Wasserstein Distortion (WD), explicitly optimizing the reconstructed appearance toward human perceptual preferences. Together, these components enable the synthesis of visually convincing textures that are not explicitly stored in the underlying 3DGS representation, substantially improving perceptual quality under strict representation-cost constraints.

3.3 TEXTURE SYNTHESIS NETWORK

The synthesis network generates perceptually plausible texture details that are not captured by the underlying 3DGS representation. It operates in a content- and view-adaptive manner by conditioning on both the initial rendered image and pseudo-random Gaussian noise maps that vary with the rendering viewpoint. The network architecture comprises three 1×1 convolutional layers followed by two 3×3 residual convolutional layers.

This lightweight design incurs negligible storage, transmission, and memory overhead (< 0.01 MB) while retaining real-time rendering speeds (Table 2). Given a target view v_j, the input to the network consists of the base rendered image $I^{v_j}_{\text{base}} \in \mathbb{R}^{3\times H\times W}$, where H and W denote the image height and width, and a set of multi-scale pseudo-random Gaussian noise maps $\mathbf{C} = \{C_i\}_{i=0}^{L-1}$, where each $C_i \in \mathbb{R}^{1\times H\times W}$ is obtained from the i-th level of the 3D hash grids, as described in the next section. These multi-resolution 3D hash grids are generated deterministically at both the encoder and decoder using a shared random seed, following the principle of *common randomness*. Consequently, they incur negligible transmission overhead.

The base image $I^{v_j}_{\text{base}}$ and noise maps $\mathbf{C}$ are concatenated and passed through the synthesis network f_θ, with parameters θ, which generates a residual texture signal R^{v_j}: $R^{v_j} = f_\theta(\{I^{v_j}_{\text{base}}, \mathbf{C}\}) \in \mathbb{R}^{3\times H\times W}$. The final output is obtained by adding the synthesized residual R^{v_j} to the base rendering $I^{v_j}_{\text{base}}$: $I^{v_j}_{\text{final}} = I^{v_j}_{\text{base}} + R^{v_j}$.

3.4 3D-ANCHORED PSEUDO-RANDOM GAUSSIAN NOISE

We derive the 2D Gaussian noise maps $\mathbf{C}$ from a 3D hash grid to satisfy a key requirement: the synthesized textures must remain consistent across viewpoints. Ideally, pixels rendered from different views but corresponding to the same 3D surface point should be conditioned on identical or highly correlated noise features to promote cross-view coherence in texture synthesis. A naïve approach that reuses a fixed 2D noise map across all viewpoints fails to satisfy this requirement, as the noise is anchored to image-space coordinates rather than the underlying 3D scene geometry. As the viewpoint changes, the same image-space location generally corresponds to different 3D surface points. This causes the synthesized high-frequency details to produce floating noise patterns and noticeable cross-view inconsistencies (see "CrossViewArtifacts" folder here).

To address this issue, we propose *3D-anchored pseudo-random Gaussian noise*. Inspired by multi-resolution hash-grid representations (Müller et al., 2022; Chen et al., 2024), we construct an L-level ($L = 8$ in our implementation) 3D hash grid, which provides a compact and efficient way to encode spatially varying signals across multiple scales without the memory cost of a dense 3D grid. For each level, the hash table entries are sampled once from a Gaussian distribution using a shared random seed and remain fixed throughout training and inference. Given a rendered view, we first construct a per-pixel 3D point map $P^{v_j} \in \mathbb{R}^{3\times H\times W}$ by unprojecting the rendered depth map into world space. Each 3D point $\mathbf{x}_p \in \mathbb{R}^3$ of a pixel p is normalized by the scene extent s and scaled as $\mathbf{u}_{p,l} = f_l \cdot \mathbf{x}_p / s$, where f_l is the scaling factor of level l. We quantize it to $\mathbf{y}_{p,l} = \lfloor \mathbf{u}_{p,l} \rfloor$, hash the eight neighboring grid vertices $\mathrm{Y}(u_{p,l}) = \{\mathbf{y}_{p,l} + \boldsymbol{\delta} \mid \boldsymbol{\delta} \in \{0,1\}^3\}$ through a hash mapping function $h(\cdot)$ into the indices of the corresponding hash table $\boldsymbol{\xi}_l$, and obtain the noise feature $c_{p,l}$ by trilinear interpolation, $c_{p,l} = \sum_{\mathbf{y}\in \mathrm{Y}(\mathbf{u}_{p,l})} w(\mathbf{u}_{p,l}, \mathbf{y}), \boldsymbol{\xi}_l[h(\mathbf{y})]$. Concatenating all levels gives $\mathbf{c}_p = [c_{p,1}, \ldots, c_{p,L}]$. Since the hash function and random tables are shared across views, the same 3D surface location produces consistent noise features, anchoring synthesized textures to the 3D scene geometry.

Obtaining reliable 3D anchors requires a stable depth estimate for each rendered pixel. To this end, we render a depth map D^{v_j} and unproject it using the camera intrinsics and extrinsics to obtain the corresponding 3D point map. Rather than relying on alpha-blended depth, which is sensitive to opacity values and the spatial distribution of contributing Gaussians, we adopt the median-depth formulation introduced in Huang et al. (2024). For a pixel x, let $\mathcal{G}(x) = \{1, \ldots, N_x\}$ denote the contributing Gaussians, sorted by increasing camera-space depth. For Gaussian i, let z_i, α_i, and $G_i(x)$ denote its depth, opacity, and projected Gaussian weight, respectively. The transmittance before Gaussian i is $T_i(x) = \prod_{j=1}^{i-1}(1 - \alpha_j G_j(x))$, and the median depth is defined as $z_{\text{median}}(x) = \max\{z_i \mid T_i(x) > 0.5,\ i \in \mathcal{G}(x)\}$. This formulation selects the deepest contributing Gaussian before the accumulated transmittance drops below 0.5, providing a more stable surface anchor for querying the 3D hash grids (Table 3). Finally, the 3D point corresponding to each pixel is queried at all L hash-grid levels. The resulting features are concatenated into multi-resolution noise maps $C \in \mathbb{R}^{L\times H\times W}$ and provided to the synthesis network together with the rendered image.

3.5 PER-SCENE OPTIMIZED TRAINING

We adopt a two-stage per-scene optimization procedure to jointly train the 3DGS representation and the synthesis network. Directly optimizing the full pipeline with WD from the outset can destabilize the training while the Gaussian geometry is still evolving through densification and pruning. In the first stage, we disable the synthesis network and warm up the underlying 3DGS representation with its original training objective for $15,000$ iterations, where $\mathcal{L}_{\text{3DGS-D}}$ is the standard 3DGS distortion loss combining L1 and SSIM, $\mathcal{L}_{\text{Reg}}$ denotes the method-specific regularization term, and $\mathcal{L}_{\text{rate}}$ is included only for RDO methods. In the second stage, we activate the synthesis network and jointly optimize it with the 3DGS representation. We replace $\mathcal{L}_{\text{3DGS-D}}$ with $\mathcal{L}_{\text{WD}}$ while retaining the other method-specific terms. Starting from iteration $15,000$, $\mathcal{L}_{\text{WD}}$ is computed between the reconstructed image $I^{v_j}_{\text{final}}$ and the ground truth $I^{v_j}_{\text{gt}}$ and averaged over the training views:

$$\mathcal{L} = \mathcal{L}_{\text{WD}}(I^{v_j}_{\text{final}}, I^{v_j}_{\text{gt}}) \; + \; \mathcal{L}_{\text{Reg}} \; + \; \lambda \mathcal{L}_{\text{rate}}. \tag{1}$$

4 EXPERIMENTS

Baselines. To validate the generality of our PW, we apply it to a diverse set of 3DGS baselines, including both RDO (e.g., CAT-3DGS Pro (Zhan et al., 2025) and HAC++ (Chen et al., 2025a)) as well as to non-RDO (e.g., Pixel-GS (Zhang et al., 2024) and Perceptual-GS (Zhou & Ni, 2025)) methods. Each PW-augmented model is compared with its corresponding baseline without PW. For CAT-3DGS Pro, we adopt an enhanced variant, denoted as CAT-3DGS Pro*, which incorporates the implicit neural representation (INR)-based hyperprior from Tseng et al. (2026). We further compare PW with WD-R (Özyılkan et al., 2026), which improves perceptual quality through a WD-based optimization objective without employing a synthesis network. For a fair comparison, both methods are applied independently to the same CAT-3DGS Pro* backbone. For RDO methods, we evaluate reconstructions decoded from compressed bitstreams at three dataset-average rate points, ensuring that the baseline and PW-augmented models are compared under comparable bit-rate constraints.

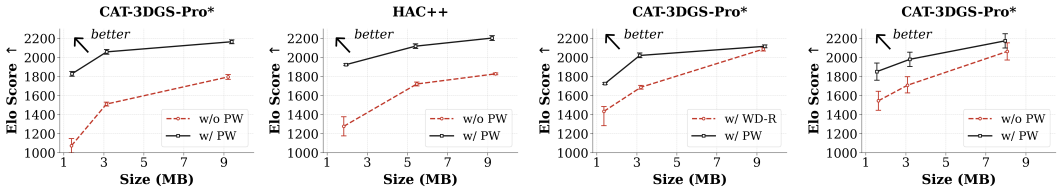
Datasets. We perform both full-reference image quality assessments and no-reference ablation studies on the *Mip-NeRF 360* dataset (Barron et al., 2022), which comprises five outdoor scenes (*bicycle, flowers, garden, stump, treehill*) and four indoor scenes (*bonsai, counter, kitchen, room*). We further assess free-viewpoint navigation quality on the *Tanks and Temples* dataset (Knapitsch et al., 2017), consisting of 4 static scenes (*m60, playground, train, truck*). Specifically, we use the preprocessed version provided by Martin et al. (2025), in which each scene is split into two camera trajectories: one used for training and the other reserved for evaluation.

Human Rating for Subjective Quality Evaluation. We evaluate rate-distortion performance using representation size on the horizontal axis and subjective quality measured by Elo score (Caron & Doucet, 2012) on the vertical axis. For RDO methods, the size reflects the compressed bitstream comprising both the 3DGS representation and the synthesis network. Following the CLIC protocol (CLI, 2025), we conduct pairwise human preference studies on the Mabyduck platform (Mabyduck, 2025), which employs a Bayesian Elo rating system. A higher Elo score indicates a stronger human preference for images that more closely resemble the ground-truth reference. Under the Elo model, equal scores imply a 50% expected win probability, while differences of 100, 200, and 400 points correspond to expected win probabilities of approximately 64%, 76%, and 91%, respectively.

We evaluate both image quality and free-viewpoint navigation quality using full-reference pairwise comparisons. Raters select the reconstruction that appears perceptually closer to the ground-truth reference, which naturally penalizes implausible or inconsistent hallucinations. Image quality is evaluated on rendered test views, whereas navigation quality is evaluated on rendered video sequences to expose temporal and cross-view artifacts.

Overall, we gathered $9,032$ pairwise comparisons from 120 raters, applying pre-screening to remove unreliable responses. We perform controlled within-family comparisons because our goal is to quantify the perceptual improvement introduced by PW rather than attempting to rank heterogeneous 3DGS methods globally. Elo scores are thus meaningful only within each evaluation family. We also report FID (Heusel et al., 2017), DISTS (Ding et al., 2022), and PSNR, with additional metrics and full subjective-study details included in Appendices and the supplementary material.[1]

[1]Complete reconstructed images and videos for all the experiments are here.



(a) RDO methods: CAT-3DGS Pro* and HAC++. (b) Our PW versus WD-R. (c) Navigation quality.

Figure 3: Rate-distortion performance in terms of Elo scores (Error bars correspond to 95% confidence intervals). (a) and (b) illustrate image quality comparisons on *Mip-NeRF 360*. (c) compares navigation quality on *Tanks and Temples*.

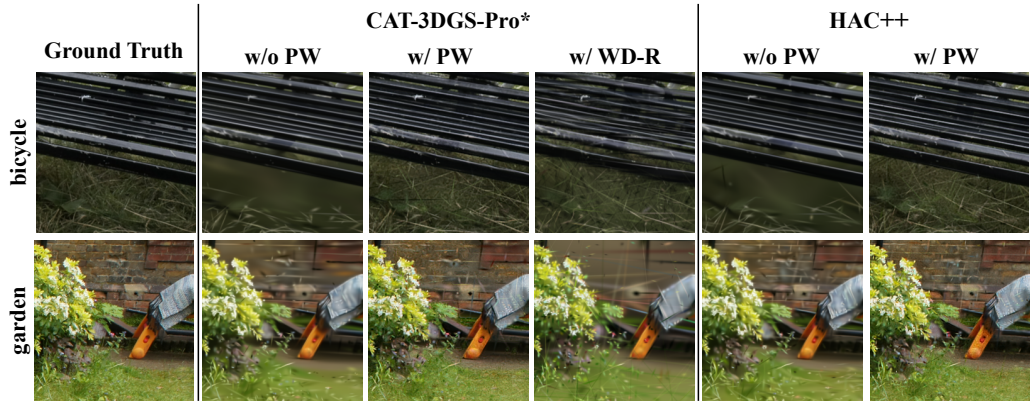


Figure 4: Low-rate qualitative results of perceptual wrapper (PW) integrated into **RDO** methods.

4.1 Subjective and Objective Quality Evaluation

Image Quality Comparisons with RDO 3DGS. Fig. 3a reports the results of our subjective study after integrating PW into RDO 3DGS methods, including CAT-3DGS Pro* (Zhan et al., 2025) and HAC++ (Chen et al., 2025a). Across all evaluated rate points and baseline methods, our approach yields substantial improvements in perceptual quality at comparable bit rates. Notably, the lowest-rate operating point of PW attains an Elo score comparable to the highest-rate point of the corresponding baselines, yielding approximately a $6\times$ reduction in bit rate at similar perceptual quality. Fig. 3b shows that PW outperforms WD-R (Özyılkan et al., 2026), especially in the low bit-rate regime. This indicates that simply replacing fidelity-based objectives with perceptually driven objectives is insufficient to reproduce the gains achieved by PW. Qualitative results in Fig. 4 further confirm that PW better preserves perceptually plausible texture details. As reported in Table 2, PW also retains real-time rendering performance, exceeding 80 FPS across all evaluated settings on an NVIDIA H100 GPU. For complete quantitative results and visuals, refer to Appendix G and H.

Image Quality Comparisons with Non-RDO 3DGS. Table 4 reports results for Perceptual-GS (Zhou & Ni, 2025) and Pixel-GS (Zhang et al., 2024). Integrating PW introduces negligible overhead in model size while consistently improving Elo, FID, and DISTS scores across both methods. Qualitative comparisons in Fig. 5 further demonstrate the perceptual benefits of PW. Notably, PW also yields meaningful gains for Perceptual-GS, despite its existing use of perceptually guided densification. This observation suggests that PW is complementary to sensitivity-guided primitive allocation. The accompanying PSNR reduction is expected and is consistent with the perception-distortion trade-off (Blau & Michaeli, 2018). For complete results, refer to Appendix G and H.

Image Quality Comparisons with Texture-Augmented Gaussians. We evaluate PW against Textured Gaussians (TexGS) (Chao et al., 2025), which augments 2D/3D Gaussian primitives with learnable texture maps. Since the released implementation is built on 2DGS, we denote it as TexGS* and use 2DGS as the reference backbone. Following Kheradmand et al. (2024), we re-train 2DGS to match the number of Gaussians used by TexGS* and denote this variant as 2DGS*. We evaluate 2DGS*, 2DGS* w/ PW, TexGS*, and TexGS* w/ PW under two settings. First, under a comparable model-size budget, including both the texture maps and the PW synthesis network, TexGS* uses 100k Gaussians while 2DGS* uses 208k, because 2DGS* does not incur per-Gaussian texture maps. As shown in Table 5, PW yields consistent gains across all quality metrics except PSNR for both TexGS* and 2DGS*. The PSNR reduction is expected due to perception-distortion trade-off (Blau

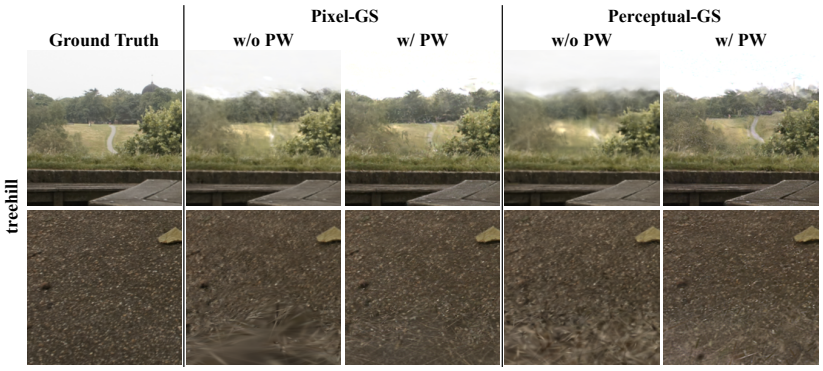


Figure 5: Qualitative results of perceptual wrapper (PW) integrated into **non-RDO** methods.

Table 2: Rendering speed (FPS) ↑ comparisons.

Setting	CAT-3DGS-Pro*	HAC++	TexGS*
w/o PW	269.5	267.7	121.6
w/ PW	133.8	130.5	85.8

Table 3: Depth rendering strategies.

Method	Size (MB) ↓	Elo ↑
Median Depth	**1.40**	**2074**
Alpha-Blended Depth	1.42	1901

Table 4: Non-RDO methods: Pixel-GS and Perceptual-GS w/ and w/o PW.

Method	MB	Elo↑	FID↓	DISTS↓	PSNR↑
Pixel-GS	1264	1857	40.11	0.055	**27.66**
Pixel-GS w/ PW	1275	**2022**	**24.24**	**0.041**	26.14
Perceptual-GS	698	1885	36.29	0.050	**27.77**
Perceptual-GS w/ PW	700	**2024**	**25.27**	**0.044**	25.69

& Michaeli, 2018). Notably, 2DGS* w/ PW achieves the best overall perceptual quality, suggesting that allocating capacity to additional Gaussians and a synthesis network is more effective than storing dedicated texture maps for individual Gaussians. To disentangle this result from differences in Gaussian count, we further compare these competing methods under a fixed number of Gaussians. As reported in Table 6, TexGS* achieves better reconstruction quality than 2DGS* across all quality metrics, reflecting the additional representational capacity provided by its per-Gaussian texture maps. However, 2DGS* w/ PW still exhibits superior perceptual quality compared to TexGS*, as evidenced by its better Elo, FID, and DISTS scores. PW also achieves perceptual improvements for TexGS*. These results show that the perceptual benefits of PW persist even when controlling for Gaussian count. For complete quantitative results and visuals, see Appendix G and H.

Free-Viewpoint Navigation Quality Comparisons. Continuous camera navigation can reveal temporal inconsistencies such as flickering or floating noise patterns (i.e., sticker-like texture artifacts). We therefore evaluate navigation quality using CAT-3DGS Pro* as a representative RDO-based backbone, following the human-rating protocol with reference and reconstructed video sequences rendered along identical camera trajectories. Fig. 3c shows that PW consistently improves perceptual quality during navigation across all 3 rate points, demonstrating that the synthesized details remain coherent. Results for the non-RDO Perceptual-GS (Zhou & Ni, 2025) baseline appear in Appendix D, and video visualizations are provided in the supplementary material.

4.2 More Analyses and Ablation Studies

Alpha-Blended Depth vs. Median Depth. To evaluate the impact of different depth-rendering strategies on perceptual quality during navigation, we conduct no-reference pairwise video comparisons following ITU-R BT.500-15 (ITU-R, 2023), instructing raters to select the reconstruction with fewer perceptual artifacts under matched rendering settings. We use this protocol because navigation artifacts, such as cross-view inconsistency and floating noise, are often more apparent when directly comparing reconstructed sequences than when comparing each sequence against a reference. Table 3 indicates that median depth yields a higher Elo score than alpha-blended depth.

2D- vs. 3D-Anchored Noise. We conduct a subjective study using the same no-reference protocol as above, comparing videos synthesized with fixed 2D noise and our 3D-anchored noise. In Table 7, Elo scores are evaluated separately for each rate (and are not comparable across rates); 3D-anchored noise achieves higher Elo scores than 2D noise. To further assess temporal consistency, we compute tOF (Chu et al., 2020), which quantifies discrepancies between the optical flow fields of rendered and ground-truth frame pairs. tOF is evaluated on *Tanks and Temples*, where camera motion over static scenes provides the required ground-truth reference. In Table 8, 3D-anchored noise attains

Table 5: 2DGS* vs. TexGS* at comparable model size (∼49 MB). PW overhead < 0.01 MB.

Method	#GS	Elo↑	FID↓	DISTS↓	PSNR↑
2DGS*	208k	1807	107.01	0.129	**25.64**
2DGS* w/ PW	208k	**2193**	**53.60**	**0.061**	23.15
TexGS*	100k	1533	119.30	0.141	25.39
TexGS* w/ PW	100k	2004	62.28	0.069	22.48

Table 6: 2DGS* vs. TexGS* with 100k Gaussians each. For TexGS*, size (MB) includes Gaussian parameters + texture maps.

Method	Size	Elo↑	FID↓	DISTS↓	PSNR↑
2DGS*	24	1621	135.80	0.159	24.83
2DGS* w/ PW	24	2100	63.01	0.070	22.47
TexGS*	24+26	1842	119.30	0.141	**25.39**
TexGS* w/ PW	24+26	**2190**	**62.28**	**0.069**	22.48

Table 7: Elo scores ↑ of 2D- vs. 3D-anchored noise using CAT-3DGS Pro* on *Mip-NeRF 360*.

Size (MB)	2D Noise	3D Noise
1.4	1902	**2077**
9.4	1933	**2024**

Table 8: Optical-flow consistency tOF (×10) ↓ on *Tanks and Temples*.

Method	2D Noise	3D Noise
CAT-3DGS Pro*	5.515	**3.804**
Perceptual-GS	3.086	**2.853**

Table 9: Ablation on pseudo-random Gaussian noise.

PW Variants	Size (MB) ↓	Elo score ↑	FID ↓	DISTS ↓	PSNR ↑
Full	**1.40**	**2086**	**37.90**	**0.070**	**23.42**
w/o Gaussian Noise	1.66	1821	79.70	0.112	23.18

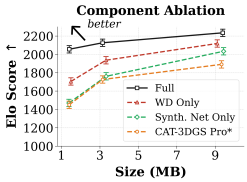


Figure 6: Component analysis.

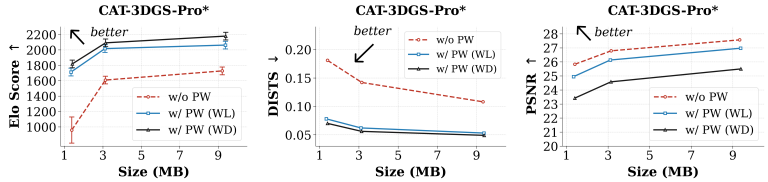


Figure 7: Distortion-perception trade-offs. **WL** denotes a combination of WD, L1 and SSIM losses. **WD** denotes WD loss only.

lower tOF than 2D noise across two representative 3DGS backbones. These results show that 3D-anchored noise more effectively preserves cross-view consistency.

Disabling Noise Input. To assess the role of pseudo-random Gaussian noise, we remove it and require the synthesis network to generate textures purely from the rendered input. In Table 9, removing the Gaussian noise leads to consistent degradation across all quality metrics, confirming that stochastic noise is essential to texture synthesis.

Component Analysis. Fig. 6 ablates the two core components of our method, i.e., the synthesis network (with 3D noise) and the WD objective, using CAT-3DGS Pro* on *Mip-NeRF 360*. **WD Only** applies the WD loss directly to the 3DGS representation without the synthesis network, while **Synthesis Net Only** trains the synthesis network with 3D noise under L1 and SSIM losses. Our full model consistently achieves the highest Elo score at every rate point, outperforming both ablated variants and indicating that WD and the synthesis network provide complementary perceptual gains.

Distortion-Perception Trade-offs. We analyze how PW behaves as the training objective places increasing emphasis on pixel fidelity. We augment $\mathcal{L}_{\text{WD}}$ in Eq. (1) with empirically weighted L1 and SSIM losses. In Fig. 7, stronger pixel-wise fidelity supervision yields higher PSNR (Distortion), accompanied by modest declines in DISTS and Elo (Perception). That is, when guided by a properly weighted loss, PW achieves more favorable distortion-perception trade-offs than CAT-3DGS-Pro*, suggesting a promising direction for future research.

5 Conclusion

We present a *perceptual wrapper* that introduces texture synthesis and perceptual optimization into existing 3DGS pipelines. To ensure spatially coherent synthesis across viewpoints, we propose multi-scale 3D-anchored pseudo-random Gaussian noise as a form of common randomness. Our method improves the perceptual quality of diverse 3DGS representations while maintaining real-time rendering speeds, offering an effective balance among reconstruction quality, model size, and rendering efficiency. Finally, PW points toward a promising direction for future work on improving 3DGS perception-distortion trade-offs.

AI USE STATEMENT

In this work, we used generative AI tools to implement specific components of the proposed method, including the 3D noise-anchoring modules. We have not used generative AI tools for formulating mathematical claims, assisting in proofs, proposing hypotheses, interpreting results, or generating synthetic datasets; other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools for editing software code for evaluation and figure plotting, editing the paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. The AI-assisted implementations were manually reviewed line-by-line and validated against official metric implementations. AI-edited text was checked for technical accuracy, and surfaced literature was verified by reading the original papers before citing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves a human subjective evaluation to assess perceptual quality. Voluntary participants provided informed consent prior to the study, which followed ITU-R BT.500 guidelines. No personally identifiable information was collected; all data were anonymized and used strictly for research purposes, presenting minimal risk to participants.

All experiments utilize publicly available datasets (Mip-NeRF 360, Tanks and Temples) in accordance with their licenses. These datasets contain static scenes without sensitive or private information, and no new data is released. We strongly encourage the responsible and ethical application of our work.

REPRODUCIBILITY STATEMENT

We will publicly release our source code upon publication, together with the reconstructed images rendered from the representations for all test views, to facilitate direct comparison without retraining.

REFERENCES

7th challenge on learned image compression (CLIC 2025). https://clic2025.compression.cc/, 2025. Last accessed 2026/06/24.

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

Zoubida Ameur, Wassim Hamidouche, Edouard François, Miloš Radosavljević, Daniel Menard, and Claire-Hélène Demarty. Deep-based film grain removal and synthesis. *IEEE Transactions on Image Processing*, 32:5046–5059, 2023. doi: 10.1109/TIP.2023.3308726.

Jona Ballé, Luca Versari, Emilien Dupont, Hyunjik Kim, and Matthias Bauer. Good, cheap, and fast: Overfitted image compression with wasserstein distortion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23259–23268, 2025. doi: 10.1109/CVPR52734.2025.02166.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469, 2022. doi: 10.1109/CVPR52688.2022.00539.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952.

François Caron and Arnaud Doucet. Efficient bayesian inference for generalized bradley–terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012.

Brian Chao, Hung-Yu Tseng, Lorenzo Porzi, Chen Gao, Tuotuo Li, Qinbo Li, Ayush Saraf, Jia-Bin Huang, Johannes Kopf, Gordon Wetzstein, and Changil Kim. Textured gaussians for enhanced 3d scene appearance modeling. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8964–8974. IEEE, 2025.

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

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac++: Towards 100x compression of 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):10210–10226, 2025a. doi: 10.1109/TPAMI.2025.3594066.

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. Dashgaussian: Optimizing 3d gaussian splatting in 200 seconds. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11146–11155, June 2025b.

Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2022. doi: 10.1109/TPAMI.2020.3045810.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European conference on computer vision*, pp. 165–181. Springer, 2024.

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

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21537–21546, 2025a. doi: 10.1109/CVPR52734.2025.02006.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5949–5958, 2025b.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

ITU-R. Recommendation ITU-R BT.500-15: Methodologies for the Subjective Assessment of the Quality of Television Images, 2023. URL `https://www.itu.int/rec/R-REC-BT.500`.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315. IEEE, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024.

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

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728, 2024.

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

Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. Compgs: Efficient 3d scene representation via compressed gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 2936–2944, 2024b.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664, 2024. doi: 10.1109/CVPR52733.2024.01952.

Mabyduck. Mabyduck. `https://www.mabyduck.com`, 2025. Last accessed 2026/06/24.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Pedro Martin, António Rodrigues, João Ascenso, and Maria Paula Queluz. Gs-qa: Comprehensive quality assessment benchmark for gaussian splatting view synthesis. In *2025 17th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–7. IEEE, 2025.

Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Andrey Norkin and Neil Birkbeck. Film grain synthesis for av1 video codec. In *2018 Data Compression Conference*, pp. 3–12, 2018. doi: 10.1109/DCC.2018.00008.

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

Yang Qiu, Aaron B. Wagner, Jona Ballé, and Lucas Theis. Wasserstein distortion: Unifying fidelity and realism. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 2024. doi: 10.1109/CISS59072.2024.10480168.

Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2025a. doi: 10.1109/TPAMI.2025.3568201.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. Fastgs: Training 3d gaussian splatting in 100 seconds. *arXiv preprint arXiv:2511.04283*, 2025b.

Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Ganerf: Leveraging discriminators to optimize neural radiance fields. *ACM Trans. Graph.*, 42(6), nov 2023. ISSN 0730-0301. doi: 10.1145/3618402. URL https://doi.org/10.1145/3618402.

Victor Rong, Jingxiang Chen, Sherwin Bahmani, Kiriakos N. Kutulakos, and David B. Lindell. Gstex: Per-primitive texturing of 2d gaussian splatting for decoupled appearance and geometry modeling. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3508–3518, 2025. doi: 10.1109/WACV61041.2025.00346.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Xiangyu Sun, Joo Chan Lee, Daniel Rho, Jong Hwan Ko, Usman Ali, and Eunbyung Park. F-3dgs: Factorized coordinates and representations for 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7957–7965, 2024.

David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Billboard splatting (bbsplat): Learnable textured primitives for novel view synthesis. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25029–25039. IEEE, 2025.

Yu-Jen Tseng, Chia-Hao Kao, Jing-Zhong Chen, Alessandro Gnutti, Shao-Yuan Lo, Yen-Yu Lin, and Wen-Hsiao Peng. Csgaussian: Progressive rate-distortion compression and segmentation for 3d gaussian splatting. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6883–6892, 2026.

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

Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex Kot, and Bihan Wen. ContextGS : Compact 3d gaussian splatting with anchor level context model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=W2qGSMl2Uu.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Rui Xu, Wenyue Chen, Jiepeng Wang, Yuan Liu, Peng Wang, Cheng Lin, Shiqing Xin, Xin Li, Wenping Wang, and Taku Komura. Svgs: Enhancing gaussian splatting using primitives with spatially varying colors. *arXiv preprint arXiv:2411.18966*, 2024.

Yu-Ting Zhan, He-bi Yang, Cheng-Yuan Ho, Jui-Chiu Chiang, and Wen-Hsiao Peng. Cat-3dgs pro: A new benchmark for efficient 3dgs compression. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, pp. 1367–1371, 2025. doi: 10.23919/EUSIPCO63237.2025.11226320.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Yangming Zhang, Wenqi Jia, Wei Niu, and Miao Yin. Gaussianspa: An "optimizing-sparsifying" simplification framework for compact and high-quality 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26673–26682, 2025. doi: 10.1109/CVPR52734.2025.02484.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *ECCV*, 2024.

Hongbi Zhou and Zhangkai Ni. Perceptual-GS: Scene-adaptive perceptual densification for gaussian splatting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ij0vj0BC72.

Ezgi Özyılkan, Zhiqi Chen, Oren Rippel, Jona Ballé, and Kedar Tatwawadi. Drop-in perceptual optimization for 3d gaussian splatting. 2026. URL https://arxiv.org/abs/2603.23297.

A TABLE OF CONTENTS

This appendix provides additional qualitative and quantitative results that complement the main paper.

Specifically, we include:

- Synthesis network implementation details (Appendix B)
- Illustration of Wasserstein Distortion (Appendix C)
- Free-viewpoint navigation quality comparisons of Perceptual-GS (Appendix D)
- Point cloud analysis (Appendix E)
- Semantic consistency analysis (Appendix F)
- Complete objective metrics (Appendix G).
- Extended qualitative results on Mip-NeRF 360 across scenes and views (Appendix H).
- An ablation study on the kernel sizes used for Wasserstein Distortion (Appendix I).
- Details of human rating study and Elo score computation (Appendix J).

Due to the file size limit, we include only a subset of visualizations. We provide the full set of rendered images and sequences here.

B SYNTHESIS NETWORK ARCHITECTURE

The synthesis network comprises an initial feature processing stage followed by refinement blocks. Specifically, it ingests an 11-channel input formed by concatenating the 3-channel RGB image with 8-channel multi-scale noise maps and passes it through three 1×1 convolutional layers with an intermediate feature dimension of 32, mapping the features down to 3 output channels. These representations are subsequently refined by two 3×3 residual convolutional layers with 3 feature channels throughout. Given the compact footprint of the network ($\sim$6 KB under 32-bit float storage), post-hoc model compression is unnecessary.

C ILLUSTRATION OF WASSERSTEIN DISTORTION

Fig. 8 illustrates the computation of Wasserstein Distortion between two images I_1 and I_2. First, a pre-trained VGG network (Simonyan & Zisserman, 2015) extracts their multi-scale feature representations F_1 and F_2. At each spatial location (x, y) of a feature map, neighboring features are pooled using a spatial weighting function controlled by a parameter σ, forming a localized feature distribution for each image. WD then computes the Wasserstein distance between the corresponding local distributions and aggregates these distances both spatially and across feature maps. In practice, the local feature distributions are approximated as Gaussians, allowing the squared 2-Wasserstein distance W_2^2 to be efficiently computed from their first- and second-order statistics.

In particular, the parameter σ defines the spatial scale of the pooling window (sized $\sigma \times \sigma$), thereby controlling the balance between point-wise fidelity and region-wise statistical matching. For a small σ, WD approaches a point-wise feature distance, similar in spirit to LPIPS, and strongly penalizes local feature mismatches. As σ increases, WD places greater emphasis on matching neighborhood-level feature statistics rather than exact spatial correspondence, allowing perceptually plausible texture re-sampling. Thus, σ provides a controllable trade-off between reconstruction fidelity and perceptual realism. Moreover, σ can vary spatially to enable region-adaptive perceptual optimization (Ballé et al., 2025).

D FREE-VIEWPOINT NAVIGATION QUALITY COMPARISONS OF PERCEPTUAL-GS

Table 10 reports the subjective quality results of Perceptual-GS with and without PW. At comparable model sizes, the PW-augmented variant achieves a higher Elo score, indicating improved perceptual quality during free-viewpoint navigation.

E POINT CLOUD ANALYSIS

The density maps in Fig. 9 show how the Gaussian primitive density distribution changes after incorporating PW. Without PW, Gaussian primitives are densely allocated in texture-rich regions

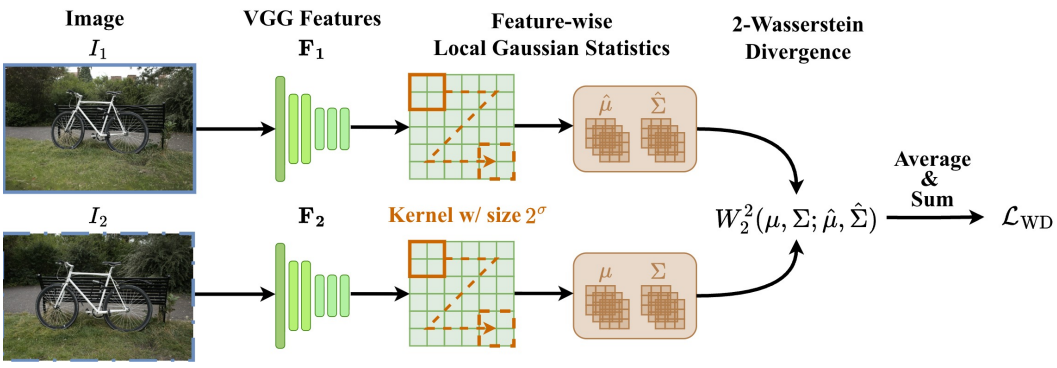


Figure 8: Illustration of Wasserstein Distortion. The distortion is computed as the squared 2-Wasserstein distance between pairs of local Gaussian statistics in the feature space.

Table 10: Navigation quality comparison of Perceptual-GS with and without PW.

Method	Size (MB)	Elo↑
Perceptual-GS	698	1936
Perceptual-GS w/ PW	700	**2050**

such as grass, whereas with PW, the density in these regions is substantially reduced. These results indicate that PW offloads high-frequency texture reconstruction to the synthesis network, reducing the need to explicitly model fine-scale detail and enabling a sparser set of primitives at comparable perceptual quality.

F Semantic Consistency Analysis

Wasserstein Distortion permits local texture re-sampling, which improves perceptual quality but may also introduce hallucinated details that are inconsistent with the reference, particularly in structured or semantically meaningful regions. Since no single existing metric reliably captures all such failure modes, we complement our perceptual evaluation with a semantic-consistency analysis.

Specifically, we use SegFormer-B5 (Xie et al., 2021) pretrained on ADE20K with 150 semantic classes to segment both the ground-truth test images and the corresponding rendered images. We then measure the mean Intersection-over-Union (mIoU) between their predicted semantic masks. For each test image, we retain only pixels whose predicted class confidence exceeds 0.7, and consider a ground-truth semantic class valid only if it contains at least $1{,}000$ pixels. For each valid class, we compute the IoU between its segmentation masks in the ground-truth and rendered images, and average the results across classes and images.

We compare CAT-3DGS Pro* with and without PW at three rate points. As shown in Table 11, the PW-augmented models achieve consistently higher semantic mIoU, indicating that the perceptual improvements introduced by PW do not come at the expense of semantic consistency and, in fact, better preserve the semantic content of the reference.

G Complete Objective Metrics

This section reports Elo scores, LPIPS, DISTS, FID, CMMD (Jayasumana et al., 2024), PSNR, SSIM, and model size (MB) for all evaluated settings. We provide per-scene numbers as well as dataset averages for both RDO and non-RDO methods. Note that Elo scores are computed from pairwise comparisons across configurations (operating points) rather than per scene; therefore, per-scene Elo scores are not available and are denoted as "–".

Basically, FID and CMMD (Jayasumana et al., 2024) measure the discrepancy between the feature distributions of rendered and ground-truth images, using Inception-v3 and CLIP ViT-L/14-336 features, respectively. To obtain sufficient samples for distribution-level estimation, we adopt the patch-based protocol of HiFiC (Mentzer et al., 2020) with 256×256 patches. Both metrics are computed per scene and averaged across scenes.

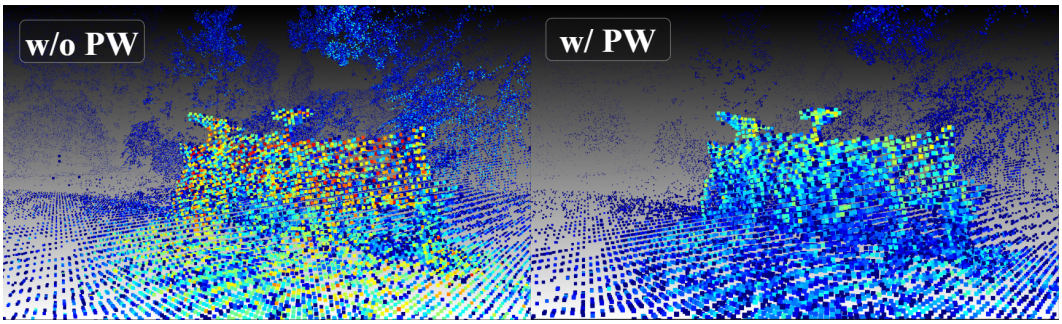


Figure 9: Gaussian primitive density visualization w/ and w/o perceptual wrapper (PW). In the density maps, color encodes local primitive density, ranging from blue (sparse) to red (dense).

Table 11: Semantic segmentation consistency mIoU using CAT-3DGS Pro* on MipNeRF 360.

Rate	Orig. mIoU ↑	**w/ PW mIoU** ↑	Improvements
low	0.4487	**0.7425**	0.2938
mid	0.5861	**0.8203**	0.2342
high	0.6791	**0.8599**	0.1808

For RDO methods (i.e., CAT-3DGS Pro* and HAC++), we report results at the same dataset-average rate points as in the main paper. Table 12 and Table 13 summarize the results at three rate points. In addition, Table 14 reports the quantitative result of the WD-R (Özyılkan et al., 2026) implementation using CAT-3DGS Pro* as the backbone. The Elo scores are obtained from comparisons against the same set of CAT-3DGS Pro* w/ PW models as in Table 12. Therefore, the results, except for Elo scores, are shared across the two tables.

For non-RDO methods (i.e., Pixel-GS and Perceptual-GS), the full results are provided in Table 15 and Table 16.

Finally, the results for Textured Gaussians* and 2DGS* variants are provided in Table 17.

Table 12: Per-scene objective metrics on the Mip-NeRF 360 dataset for CAT-3DGS Pro* and its variant augmented with our perceptual wrapper (PW).

Method	Metric	Type	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
CAT-3DGS Pro* (High rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2164**
		w/o PW	–	–	–	–	–	–	–	–	–	1795
	LPIPS	w/ PW	0.253	0.128	0.180	0.315	0.160	0.131	0.183	0.272	0.292	**0.213**
		w/o PW	0.329	0.256	0.277	0.415	0.194	0.173	0.298	0.328	0.426	0.299
	DISTS	w/ PW	0.045	0.026	0.034	0.067	0.025	0.039	0.060	0.080	0.069	**0.049**
		w/o PW	0.107	0.086	0.076	0.142	0.065	0.058	0.105	0.148	0.183	0.108
	FID	w/ PW	20.84	21.51	43.78	33.38	19.68	21.62	29.41	26.42	29.29	**27.32**
		w/o PW	47.90	47.51	79.18	101.25	45.71	34.56	57.28	57.62	109.40	64.49
	CMMD	w/ PW	0.153	0.137	0.343	0.373	0.119	0.090	0.188	0.614	0.374	**0.266**
		w/o PW	0.444	0.496	0.715	1.196	0.219	0.217	0.522	0.911	1.104	0.647
	PSNR	w/ PW	22.90	30.13	27.60	19.12	25.01	28.89	29.69	24.84	21.37	25.50
		w/o PW	24.96	32.45	29.26	21.36	27.31	31.06	31.78	26.67	23.31	**27.57**
	SSIM	w/ PW	0.565	0.860	0.818	0.473	0.718	0.847	0.829	0.619	0.495	0.691
		w/o PW	0.874	0.972	0.945	0.767	0.941	0.966	0.966	0.872	0.802	**0.901**
	Size (MB)	w/ PW	15.87	4.81	5.71	13.58	13.08	5.63	3.92	8.86	13.06	9.39
		w/o PW	15.84	4.54	4.71	13.79	15.35	5.26	3.16	8.60	12.74	9.33
CAT-3DGS Pro* (Mid rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2058**
		w/o PW	–	–	–	–	–	–	–	–	–	1510
	LPIPS	w/ PW	0.288	0.157	0.207	0.338	0.216	0.159	0.195	0.304	0.317	**0.242**
		w/o PW	0.388	0.293	0.319	0.447	0.279	0.223	0.350	0.403	0.482	0.354
	DISTS	w/ PW	0.052	0.035	0.041	0.074	0.034	0.046	0.063	0.086	0.075	**0.056**
		w/o PW	0.145	0.102	0.096	0.167	0.107	0.078	0.137	0.210	0.236	0.142
	FID	w/ PW	23.73	25.93	49.07	36.89	24.70	26.42	31.76	27.16	31.81	**30.83**
		w/o PW	68.81	57.68	92.98	118.07	70.74	47.20	77.68	95.35	151.69	86.69
	CMMD	w/ PW	0.172	0.191	0.420	0.474	0.279	0.127	0.255	0.815	0.429	**0.351**
		w/o PW	0.784	0.620	0.980	1.445	0.527	0.342	0.753	1.290	1.390	0.903
	PSNR	w/ PW	22.09	28.82	26.67	18.78	23.62	27.50	28.87	23.92	20.93	**24.58**
		w/o PW	24.76	30.98	28.26	21.22	26.41	29.58	30.55	26.15	23.22	26.79
	SSIM	w/ PW	0.497	0.831	0.782	0.385	0.611	0.814	0.802	0.553	0.455	**0.636**
		w/o PW	0.859	0.960	0.928	0.757	0.922	0.951	0.951	0.848	0.791	0.885
	Size (MB)	w/ PW	4.70	1.99	2.24	4.74	4.77	2.22	1.51	2.52	3.60	3.14
		w/o PW	4.51	1.90	2.03	4.94	5.77	2.04	1.20	2.39	3.54	3.15
CAT-3DGS Pro* (Low rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**1827**
		w/o PW	–	–	–	–	–	–	–	–	–	1073
	LPIPS	w/ PW	0.332	0.196	0.258	0.371	0.277	0.195	0.227	0.353	0.355	**0.285**
		w/o PW	0.461	0.342	0.371	0.492	0.371	0.285	0.388	0.480	0.547	0.415
	DISTS	w/ PW	0.065	0.046	0.056	0.088	0.048	0.058	0.075	0.103	0.090	**0.070**
		w/o PW	0.193	0.125	0.126	0.200	0.158	0.107	0.168	0.268	0.286	0.181
	FID	w/ PW	27.17	32.95	62.19	42.68	31.46	32.93	39.15	33.38	39.20	**37.90**
		w/o PW	99.21	71.69	110.73	142.30	112.32	64.47	91.73	136.58	195.17	113.80
	CMMD	w/ PW	0.235	0.286	0.666	0.636	0.457	0.209	0.412	0.954	0.650	**0.501**
		w/o PW	1.359	0.776	1.296	1.711	1.184	0.514	0.934	1.590	1.662	1.225
	PSNR	w/ PW	21.21	27.16	25.29	18.15	22.33	26.08	27.63	22.62	20.34	**23.42**
		w/o PW	24.03	29.65	27.16	20.76	25.31	28.09	29.55	25.11	22.82	25.83
	SSIM	w/ PW	0.421	0.780	0.723	0.335	0.500	0.760	0.781	0.458	0.393	0.572
		w/o PW	0.808	0.941	0.900	0.726	0.889	0.929	0.934	0.797	0.756	**0.853**
	Size (MB)	w/ PW	1.73	1.12	1.23	2.05	2.23	1.13	0.78	1.04	1.34	1.40
		w/o PW	1.71	1.05	1.06	2.14	2.69	0.99	0.66	1.02	1.28	1.40

Table 13: Per-scene objective metrics on the Mip-NeRF 360 dataset for HAC++ and its variant augmented with our perceptual wrapper (PW).

Method	Metric	Type	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
HAC++ (High rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2203**
		w/o PW	–	–	–	–	–	–	–	–	–	1827
	LPIPS	w/ PW	0.245	0.122	0.169	0.326	0.158	0.124	0.165	0.257	0.290	**0.206**
		w/o PW	0.284	0.189	0.201	0.390	0.173	0.134	0.214	0.282	0.374	0.249
	DISTS	w/ PW	0.044	0.026	0.032	0.070	0.025	0.038	0.053	0.075	0.070	**0.048**
		w/o PW	0.107	0.081	0.071	0.145	0.070	0.057	0.102	0.145	0.191	0.108
	FID	w/ PW	21.14	20.33	40.49	35.18	19.51	21.15	25.59	25.47	29.69	**26.50**
		w/o PW	47.38	45.75	75.16	105.46	49.32	33.26	53.56	56.71	117.66	64.92
	CMMD	w/ PW	0.145	0.166	0.229	0.425	0.191	0.095	0.142	0.411	0.348	**0.239**
		w/o PW	0.438	0.455	0.629	1.258	0.247	0.201	0.477	0.912	1.113	0.637
	PSNR	w/ PW	23.19	30.53	27.85	19.05	25.22	29.40	30.19	25.07	21.34	25.76
		w/o PW	25.02	32.67	29.41	21.25	27.20	31.36	31.77	26.62	23.24	**27.62**
	SSIM	w/ PW	0.588	0.858	0.822	0.425	0.725	0.865	0.837	0.642	0.498	0.696
		w/o PW	0.734	0.946	0.911	0.569	0.835	0.925	0.921	0.760	0.639	**0.804**
	Size (MB)	w/ PW	14.12	6.10	6.10	11.71	12.01	6.07	4.37	10.60	11.87	9.22
		w/o PW	14.53	5.88	5.36	11.95	14.42	5.91	3.83	10.61	12.06	9.39
HAC++ (Mid rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2117**
		w/o PW	–	–	–	–	–	–	–	–	–	1721
	LPIPS	w/ PW	0.262	0.132	0.206	0.335	0.186	0.135	0.178	0.272	0.312	**0.224**
		w/o PW	0.315	0.200	0.218	0.405	0.214	0.151	0.234	0.303	0.401	0.271
	DISTS	w/ PW	0.045	0.028	0.045	0.073	0.029	0.040	0.057	0.076	0.077	**0.052**
		w/o PW	0.128	0.087	0.081	0.158	0.096	0.064	0.115	0.167	0.216	0.124
	FID	w/ PW	21.80	22.00	53.09	36.13	22.15	22.73	27.98	26.02	33.27	**29.46**
		w/o PW	57.98	49.24	83.03	114.26	65.44	38.55	62.25	73.40	147.33	76.83
	CMMD	w/ PW	0.166	0.184	0.304	0.484	0.333	0.100	0.172	0.493	0.430	**0.296**
		w/o PW	0.577	0.501	0.776	1.412	0.394	0.240	0.550	1.048	1.257	0.751
	PSNR	w/ PW	22.72	29.82	26.83	18.88	24.43	28.67	29.76	24.66	21.09	25.21
		w/o PW	24.83	32.05	29.01	21.16	26.78	30.68	31.37	26.53	23.07	**27.28**
	SSIM	w/ PW	0.548	0.846	0.794	0.408	0.668	0.847	0.820	0.611	0.464	0.667
		w/o PW	0.714	0.939	0.901	0.561	0.813	0.914	0.912	0.752	0.630	**0.793**
	Size (MB)	w/ PW	7.93	3.84	3.73	6.92	6.86	3.62	2.74	6.23	6.76	5.40
		w/o PW	8.25	3.64	3.17	7.20	8.26	3.38	2.32	6.34	6.72	5.47
HAC++ (Low rate)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**1924**
		w/o PW	–	–	–	–	–	–	–	–	–	1277
	LPIPS	w/ PW	0.314	0.170	0.235	0.372	0.269	0.178	0.210	0.325	0.358	**0.270**
		w/o PW	0.399	0.250	0.299	0.460	0.345	0.231	0.294	0.385	0.479	0.349
	DISTS	w/ PW	0.059	0.038	0.051	0.086	0.048	0.055	0.068	0.095	0.099	**0.067**
		w/o PW	0.182	0.114	0.131	0.197	0.171	0.103	0.160	0.233	0.274	0.174
	FID	w/ PW	26.77	28.37	57.24	44.88	31.96	30.90	35.01	29.50	46.01	**36.74**
		w/o PW	91.63	66.19	116.69	145.75	131.02	61.50	88.09	115.54	195.03	112.38
	CMMD	w/ PW	0.270	0.267	0.520	0.690	0.410	0.183	0.271	0.741	0.677	**0.448**
		w/o PW	1.242	0.674	1.317	1.783	1.307	0.460	0.842	1.444	1.637	1.190
	PSNR	w/ PW	21.56	28.14	26.01	18.24	22.74	26.75	28.75	23.40	20.51	24.01
		w/o PW	24.14	30.13	27.43	20.73	25.29	28.59	30.15	25.62	22.83	**26.10**
	SSIM	w/ PW	0.450	0.803	0.742	0.347	0.522	0.785	0.805	0.512	0.394	0.596
		w/o PW	0.640	0.907	0.849	0.516	0.716	0.870	0.884	0.692	0.576	**0.739**
	Size (MB)	w/ PW	2.76	1.56	1.43	2.47	2.32	1.30	1.10	2.25	2.29	1.94
		w/o PW	2.59	1.45	1.17	2.46	2.46	1.19	0.94	2.12	2.10	1.83

Table 14: Per-scene objective metrics on the Mip-NeRF 360 dataset for WD-R using CAT-3DGS Pro* as the backbone.

Method	Metric	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
CAT-3DGS Pro* (w/ WD-R, High rate)	Elo score	–	–	–	–	–	–	–	–	–	2087
	LPIPS	0.254	0.159	0.192	0.323	0.166	0.130	0.198	0.229	0.328	0.220
	DISTS	0.060	0.048	0.048	0.085	0.038	0.046	0.075	0.087	0.107	0.066
	FID	41.97	53.35	23.89	59.79	24.51	45.06	20.58	32.93	64.55	40.74
	CMMD	0.093	0.050	0.050	0.076	0.120	0.212	0.018	0.113	0.179	0.101
	PSNR	24.32	31.26	28.59	20.15	26.06	30.17	30.72	26.73	22.05	26.67
	SSIM	0.683	0.927	0.881	0.507	0.784	0.904	0.900	0.749	0.572	0.768
	Size (MB)	15.41	4.65	6.96	14.42	7.62	4.06	4.52	11.43	14.82	9.32
CAT-3DGS Pro* (w/ WD-R, Mid rate)	Elo score	–	–	–	–	–	–	–	–	–	1686
	LPIPS	0.331	0.222	0.248	0.370	0.304	0.184	0.251	0.302	0.390	0.289
	DISTS	0.097	0.080	0.074	0.116	0.100	0.067	0.107	0.126	0.156	0.102
	FID	59.31	69.31	36.72	83.50	59.84	72.39	72.25	72.80	127.89	72.67
	CMMD	0.170	0.100	0.174	0.187	0.249	0.304	0.097	0.212	0.263	0.195
	PSNR	23.34	29.26	26.82	19.70	24.03	28.20	29.73	25.60	21.92	25.40
	SSIM	0.601	0.893	0.838	0.462	0.647	0.861	0.878	0.684	0.540	0.711
	Size (MB)	4.57	1.62	2.42	4.94	2.49	1.39	1.37	3.93	6.07	3.20
CAT-3DGS Pro* (w/ WD-R, Low rate)	Elo score	–	–	–	–	–	–	–	–	–	1434
	LPIPS	0.428	0.300	0.287	0.443	0.429	0.258	0.296	0.426	0.459	0.370
	DISTS	0.156	0.118	0.095	0.167	0.172	0.100	0.136	0.203	0.218	0.152
	FID	80.89	86.61	59.13	103.43	147.56	140.83	157.10	163.68	236.32	130.62
	CMMD	0.450	0.237	0.231	0.479	0.575	0.772	0.646	0.811	0.757	0.551
	PSNR	22.01	26.72	25.68	18.95	22.16	26.01	28.25	23.67	21.65	23.90
	SSIM	0.496	0.821	0.803	0.394	0.491	0.799	0.852	0.554	0.506	0.635
	Size (MB)	1.78	0.77	1.11	1.93	1.01	0.74	0.75	1.33	2.77	1.35

Table 15: Per-scene objective metrics on the Mip-NeRF 360 dataset for Perceptual-GS and its variant augmented with our perceptual wrapper (PW).

Method	Metric	Type	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
Perceptual-GS	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2024**
		w/o PW	–	–	–	–	–	–	–	–	–	1885
	LPIPS	w/ PW	0.259	0.104	0.148	0.298	0.125	0.118	0.147	0.248	0.313	0.196
		w/o PW	0.189	0.157	0.160	0.277	0.114	0.110	0.173	0.199	0.295	**0.186**
	DISTS	w/ PW	0.050	0.020	0.031	0.065	0.028	0.024	0.040	0.063	0.074	**0.044**
		w/o PW	0.046	0.034	0.046	0.069	0.029	0.027	0.054	0.057	0.086	0.050
	FID	w/ PW	21.46	19.51	36.01	30.89	16.54	20.06	23.63	22.99	36.38	**25.27**
		w/o PW	24.41	36.52	54.45	49.38	24.90	25.36	37.35	24.22	50.05	36.29
	CMMD	w/ PW	0.131	0.138	0.103	0.271	0.142	0.077	0.083	0.374	0.383	**0.189**
		w/o PW	0.122	0.311	0.313	0.380	0.097	0.131	0.314	0.529	0.444	0.294
	PSNR	w/ PW	22.37	30.55	27.96	19.32	25.92	29.36	30.52	24.81	20.43	25.69
		w/o PW	25.50	32.62	29.40	21.47	27.53	31.80	32.21	27.10	22.34	**27.77**
	SSIM	w/ PW	0.519	0.902	0.858	0.437	0.785	0.872	0.878	0.631	0.440	0.703
		w/o PW	0.783	0.953	0.922	0.629	0.862	0.935	0.935	0.797	0.640	**0.828**
	Size (MB)	w/ PW	1009	383	373	942	883	402	425	977	905	700
		w/o PW	1010	379	372	941	886	402	428	968	902	698

Table 16: Per-scene objective metrics on the Mip-NeRF 360 dataset for Pixel-GS and its variant augmented with our perceptual wrapper (PW).

Method	Metric	Type	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
Pixel-GS	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	**2022**
		w/o PW	–	–	–	–	–	–	–	–	–	1857
	LPIPS	w/ PW	0.210	0.111	0.154	0.264	0.125	0.119	0.157	0.208	0.277	**0.181**
		w/o PW	0.206	0.168	0.167	0.286	0.111	0.110	0.188	0.205	0.301	0.194
	DISTS	w/ PW	0.042	0.021	0.034	0.059	0.029	0.025	0.041	0.056	0.062	**0.041**
		w/o PW	0.051	0.039	0.048	0.088	0.030	0.028	0.061	0.060	0.092	0.055
	FID	w/ PW	19.66	19.56	36.86	28.27	16.67	20.33	24.61	20.94	31.25	**24.24**
		w/o PW	29.67	39.34	56.74	59.65	25.59	24.95	42.92	25.66	56.50	40.11
	CMMD	w/ PW	0.163	0.175	0.181	0.234	0.088	0.074	0.103	0.462	0.392	**0.208**
		w/o PW	0.164	0.349	0.360	0.424	0.087	0.121	0.396	0.541	0.589	0.337
	PSNR	w/ PW	23.93	30.47	27.63	20.34	25.87	29.25	30.10	26.20	21.47	26.14
		w/o PW	25.15	32.63	29.25	21.75	27.40	31.70	31.67	27.01	22.37	**27.66**
	SSIM	w/ PW	0.669	0.892	0.853	0.525	0.787	0.867	0.868	0.726	0.542	0.748
		w/o PW	0.764	0.950	0.920	0.628	0.862	0.935	0.928	0.786	0.639	**0.824**
	Size (MB)	w/ PW	2075	489	590	1702	2031	725	584	1547	1729	1275
		w/o PW	2075	488	589	1697	2015	717	592	1496	1706	1264

Table 17: Per-scene objective metrics on the Mip-NeRF 360 dataset for 2DGS and Textured Gaussians, with and without our perceptual wrapper (PW).

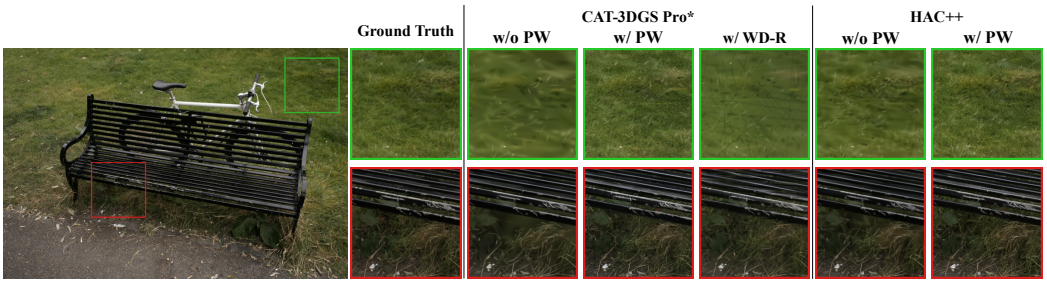
Method	Metric	Type	bicycle	bonsai	counter	flowers	garden	kitchen	room	stump	treehill	Avg.
2DGS* (#GS: 208k)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	–
		w/o PW	–	–	–	–	–	–	–	–	–	–
	LPIPS	w/ PW	0.189	0.088	0.102	0.249	0.118	0.082	0.073	0.225	0.243	**0.152**
		w/o PW	0.347	0.154	0.152	0.514	0.197	0.104	0.112	0.324	0.500	0.267
	DISTS	w/ PW	0.058	0.046	0.039	0.088	0.035	0.041	0.050	0.108	0.085	**0.061**
		w/o PW	0.138	0.096	0.071	0.197	0.108	0.060	0.085	0.169	0.236	0.129
	FID	w/ PW	32.96	72.24	87.55	55.03	33.79	52.95	49.39	49.26	49.21	**53.60**
		w/o PW	79.92	118.65	141.82	154.36	80.87	69.46	83.78	87.03	147.23	107.01
	CMMD	w/ PW	0.293	0.656	0.378	0.993	0.202	0.204	0.242	0.811	0.736	**0.502**
		w/o PW	0.731	0.834	0.652	2.177	0.575	0.183	0.436	1.187	1.468	0.916
	PSNR	w/ PW	20.85	26.30	25.20	17.21	22.83	25.32	28.91	21.73	20.00	23.15
		w/o PW	23.29	28.80	27.43	19.85	25.62	28.58	30.89	24.00	22.33	**25.64**
	SSIM	w/ PW	0.419	0.740	0.730	0.297	0.565	0.747	0.826	0.446	0.378	0.572
		w/o PW	0.636	0.899	0.868	0.475	0.777	0.898	0.916	0.650	0.577	**0.744**
	Size (MB)	w/ PW	88.04	19.90	19.46	53.53	78.27	28.97	23.25	75.11	56.34	49.21
		w/o PW	88.03	19.90	19.45	53.52	78.26	28.96	23.24	75.10	56.33	49.20
TexGS* (#GS: 100k)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	–
		w/o PW	–	–	–	–	–	–	–	–	–	–
	LPIPS	w/ PW	0.199	0.109	0.122	0.259	0.132	0.097	0.087	0.238	0.256	**0.166**
		w/o PW	0.374	0.187	0.188	0.549	0.238	0.129	0.134	0.357	0.526	0.298
	DISTS	w/ PW	0.063	0.059	0.049	0.097	0.041	0.050	0.060	0.113	0.090	**0.069**
		w/o PW	0.143	0.112	0.088	0.210	0.120	0.073	0.096	0.179	0.249	0.141
	FID	w/ PW	36.07	90.51	104.79	60.87	37.38	63.69	60.29	52.74	54.22	**62.28**
		w/o PW	80.47	131.15	165.59	168.77	88.87	80.18	92.73	99.01	166.90	119.30
	CMMD	w/ PW	0.448	0.978	0.750	1.389	0.252	0.302	0.443	0.815	0.806	**0.687**
		w/o PW	0.770	0.909	0.888	2.287	0.709	0.204	0.463	1.232	1.575	1.004
	PSNR	w/ PW	20.52	24.91	24.28	16.92	22.34	24.29	27.79	21.40	19.85	22.48
		w/o PW	23.31	28.12	27.00	19.80	25.40	28.01	30.60	23.88	22.40	**25.39**
	SSIM	w/ PW	0.393	0.699	0.679	0.275	0.521	0.704	0.785	0.416	0.362	0.537
		w/o PW	0.627	0.880	0.849	0.461	0.754	0.880	0.907	0.638	0.571	**0.730**
	Size (MB)	w/ PW	88.04	19.90	19.46	53.53	78.27	28.97	23.25	75.11	56.34	49.21
		w/o PW	88.03	19.90	19.45	53.52	78.26	28.96	23.24	75.10	56.33	49.20
2DGS* (#GS: 100k)	Elo score	w/ PW	–	–	–	–	–	–	–	–	–	–
		w/o PW	–	–	–	–	–	–	–	–	–	–
	LPIPS	w/ PW	0.201	0.122	0.124	0.263	0.135	0.098	0.086	0.242	0.254	**0.170**
		w/o PW	0.405	0.219	0.208	0.574	0.269	0.148	0.155	0.387	0.547	0.324
	DISTS	w/ PW	0.062	0.065	0.049	0.099	0.043	0.051	0.061	0.113	0.089	**0.070**
		w/o PW	0.163	0.129	0.098	0.225	0.149	0.083	0.109	0.207	0.267	0.159
	FID	w/ PW	35.80	92.33	108.31	62.41	38.68	64.17	60.76	51.65	53.01	**63.01**
		w/o PW	95.97	147.69	180.55	180.23	111.50	93.10	107.08	113.78	192.29	135.80
	CMMD	w/ PW	0.417	0.918	0.641	1.238	0.309	0.305	0.373	0.825	0.807	**0.648**
		w/o PW	1.049	1.186	1.155	2.535	0.990	0.278	0.679	1.420	1.698	1.221
	PSNR	w/ PW	20.53	24.77	24.41	16.88	22.25	24.29	28.05	21.27	19.79	22.47
		w/o PW	22.87	26.95	26.30	19.56	24.90	27.12	29.79	23.58	22.22	**24.81**
	SSIM	w/ PW	0.396	0.675	0.700	0.273	0.520	0.701	0.808	0.408	0.364	0.538
		w/o PW	0.602	0.857	0.830	0.442	0.726	0.863	0.894	0.618	0.561	**0.710**
	Size (MB)	w/ PW	42.23	9.55	9.34	25.68	37.55	13.90	11.15	36.03	27.03	23.61
		w/o PW	42.23	9.54	9.33	25.67	37.54	13.89	11.15	36.02	27.02	23.60

H Extended Qualitative Results on Mip-NeRF 360

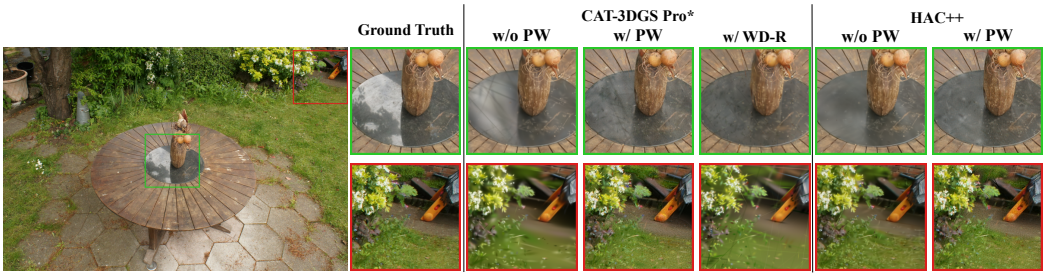
This section provides extended qualitative comparisons on the Mip-NeRF 360 dataset across all scenes.

Fig. 10 and Fig. 11 present qualitative comparisons of RDO 3DGS methods (i.e., CAT-3DGS Pro* and HAC++) across all scenes and rate points. For each scene, we compare the baseline reconstruction with the variant incorporating our perceptual wrapper (PW) at comparable bit rates. Fig. 12 presents qualitative comparisons of non-RDO 3DGS methods (i.e., Pixel-GS and Perceptual-GS). Fig. 13 presents qualitative comparisons with texture-augmented methods at comparable model sizes, namely Textured Gaussians* (TexGS*) (Chao et al., 2025) and 2DGS* (Huang et al., 2024).

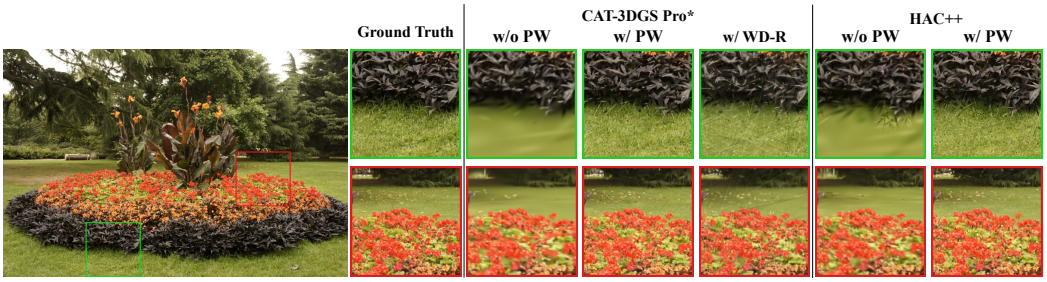
Complete reconstructed images and videos for all the experiments are here.



(a) Visualization of the *bicycle* scene

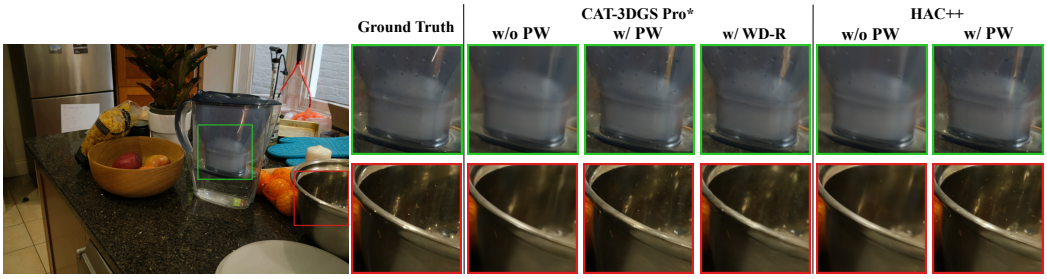


(b) Visualization of the *bonsai* scene

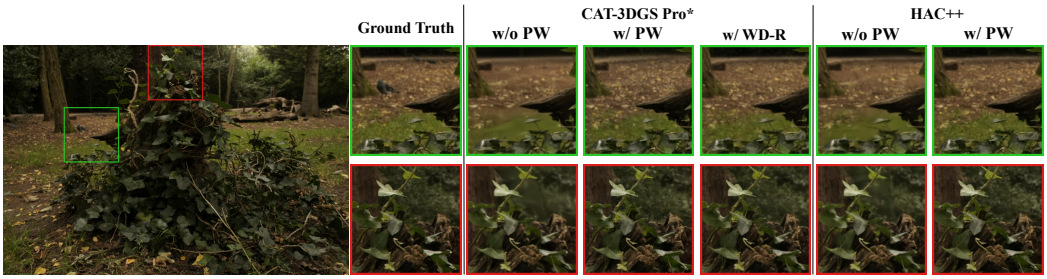


(c) Visualization of the *flowers* scene

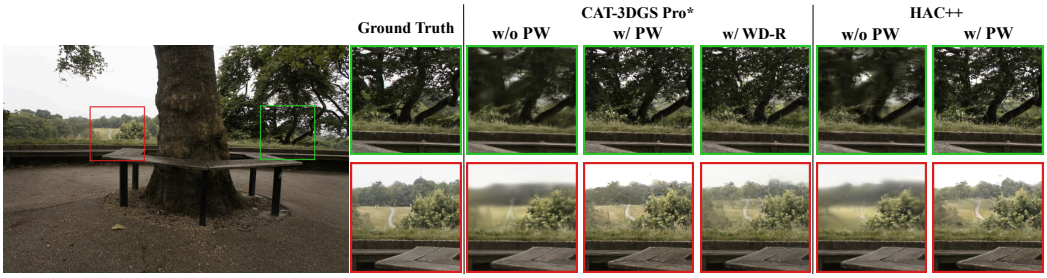
Figure 10: Visualization of the RDO methods at the lowest-rate point on the Mip-NeRF 360 dataset.



(a) Visualization of the *counter* scene

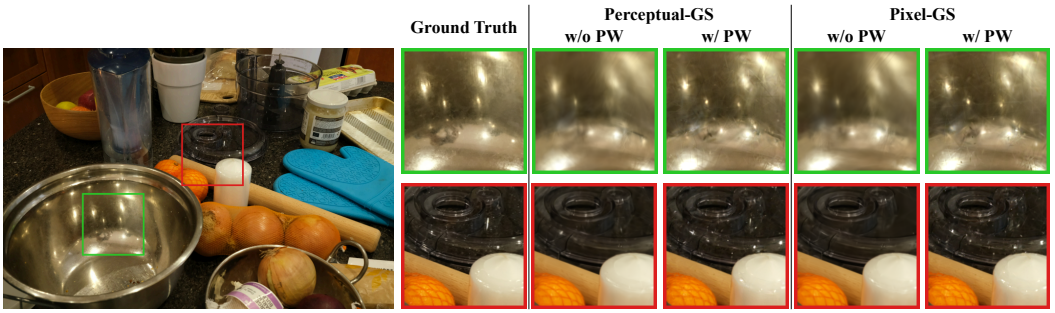


(b) Visualization of the *stump* scene

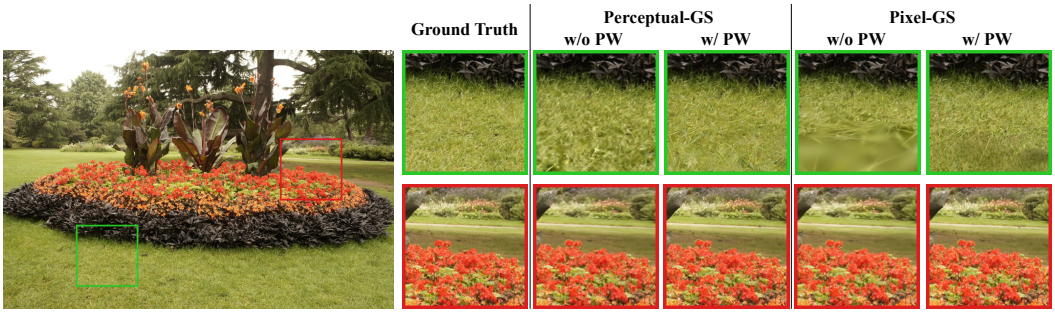


(c) Visualization of the *treehill* scene

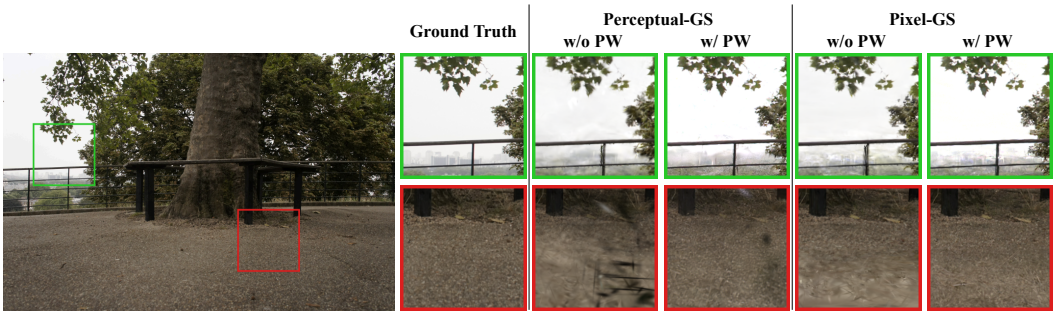
Figure 11: Visualization of the RDO methods at the high-rate point on the Mip-NeRF 360 dataset.



(a) Visualization of the *counter* scene



(b) Visualization of the *flowers* scene

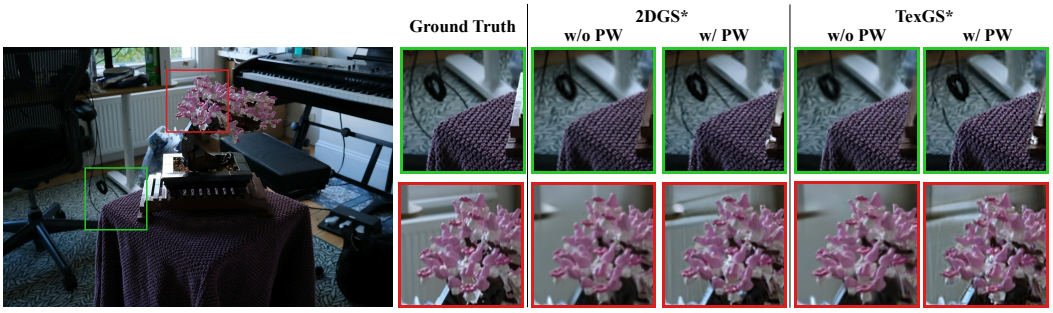


(c) Visualization of the *treehill* scene

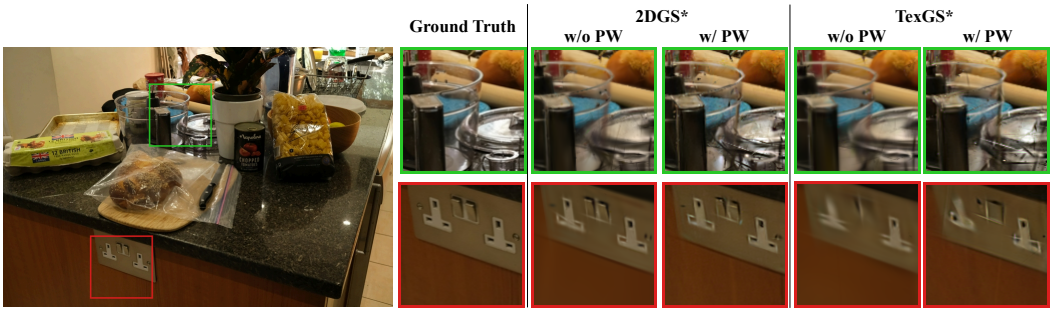
Figure 12: Visualization of the non-RDO methods on the Mip-NeRF 360 dataset.



(a) Visualization of the *bicycle* scene



(b) Visualization of the *bonsai* scene



(c) Visualization of the *counter* scene

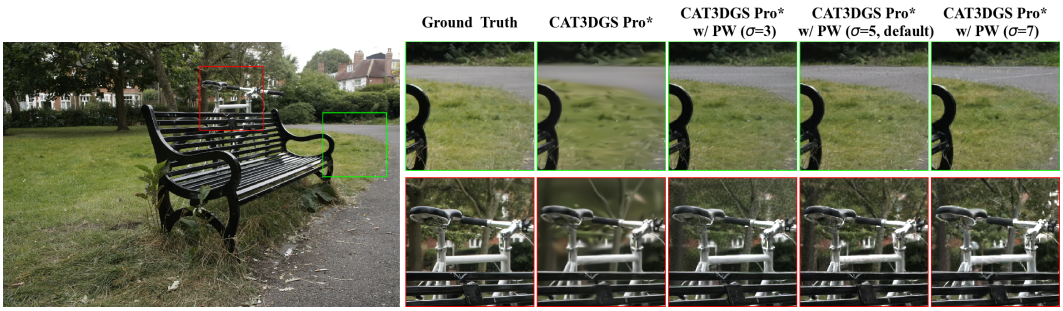
Figure 13: Visualization of the TexGS* on the Mip-NeRF 360 dataset.

I Kernel Sizes for Wasserstein Distortion

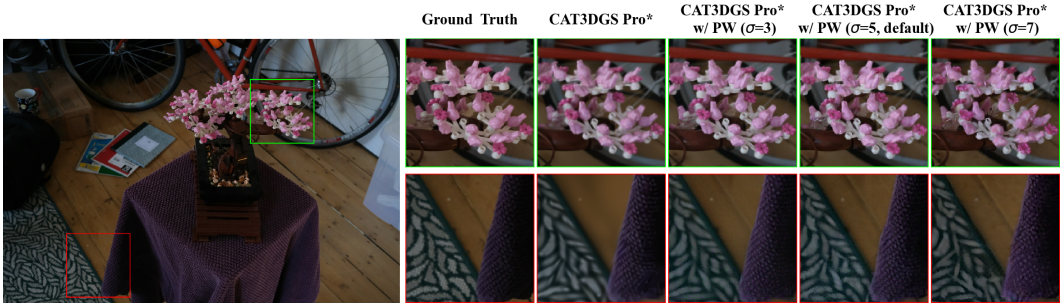
The behavior of Wasserstein Distortion (WD) is governed by its spatial kernel size σ. A small σ emphasizes fine-scale and local deviations, encouraging the model to preserve strict structural geometry. However, this can also result in spotty patterns in complex regions such as tree leaves. Conversely, a large σ prioritizes statistical matching over broader regions, allowing for more flexible texture resampling, but it can also lead to distorted structures. To determine the σ for our datasets, which span both indoor and outdoor scenes, we evaluated 3 distinct settings: $\sigma \in \{8,\ 32,\ 128\}$.

As shown in Fig. 14, our default configuration ($\sigma = 32$) achieves the best perceptual trade-off across varying scene types, and we adopt this setting throughout the main experiments.

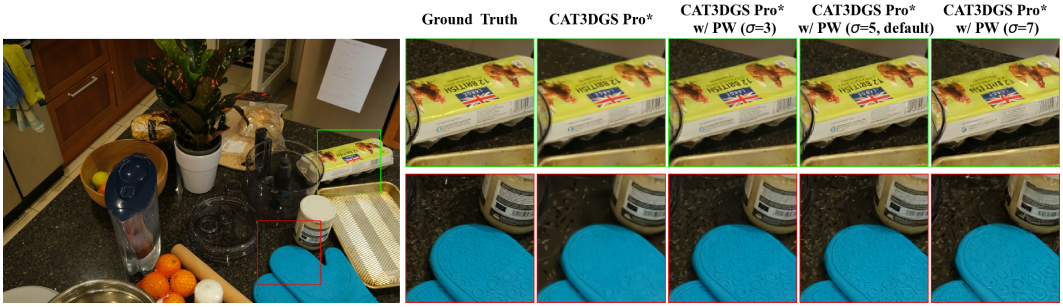
Beyond the fixed kernel size σ used in our main experiments, we further investigated spatially adaptive σ values guided by saliency maps following Ballé et al. (2025). By assigning smaller σ to salient regions, this variant improves the quality of text and structured regions. However, the remaining regions show the same performance as the fixed-σ method, resulting in negligible overall improvement, consistent with Özyılkan et al. (2026).



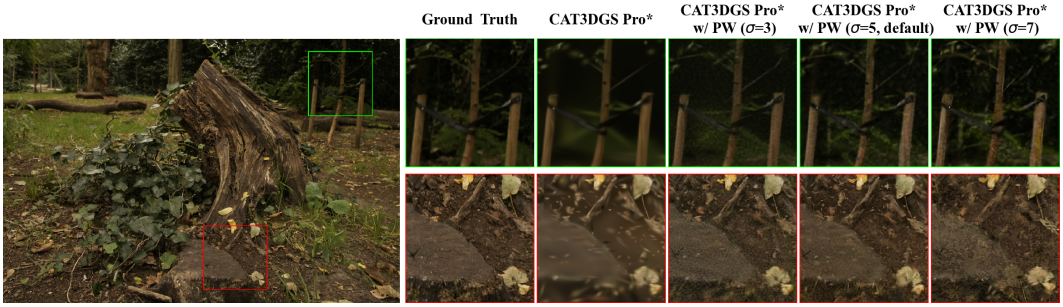
(a) Visualization of the *bicycle* scene



(b) Visualization of the *bonsai* scene



(c) Visualization of the *counter* scene



(d) Visualization of the *stump* scene

Figure 14: Visualization of different kernel sizes (σ).

J Details of Human Rating Study and Elo Computation

Raters are first introduced to the objectives of the test and are given instructions on how to use the Mabyduck platform, as shown in Fig. 15. Fig. 16 illustrates the Mabyduck platform interface during a test session, in which the reference image is displayed on the left and the reconstructions are displayed on the right.

After collecting the human ratings, Elo scores are computed via the Bradley-Terry model (Bradley & Terry, 1952), where the probability that method i is preferred over method j is

$$P(i \text{ beats } j) = \frac{\lambda_i}{\lambda_i + \lambda_j}, \tag{2}$$

with λ_i and λ_j denoting the underlying strength parameters for methods i and j. These parameters are estimated via Bayesian inference with Gibbs sampling (Caron & Doucet, 2012). The resulting strength parameters are then converted to interpretable Elo scores:

$$\text{Elo}_i = E_0 + s \times \log(\lambda_i), \tag{3}$$

where E_0 and s are the baseline rating and scale factor, respectively (set to 2000 and 400 on the Mabyduck platform). Comparing the Elo scores of two models allows us to predict the winning probability:

$$P(i \text{ beats } j) = \frac{1}{1 + 10^{(\text{Elo}_j - \text{Elo}_i)/s}}. \tag{4}$$

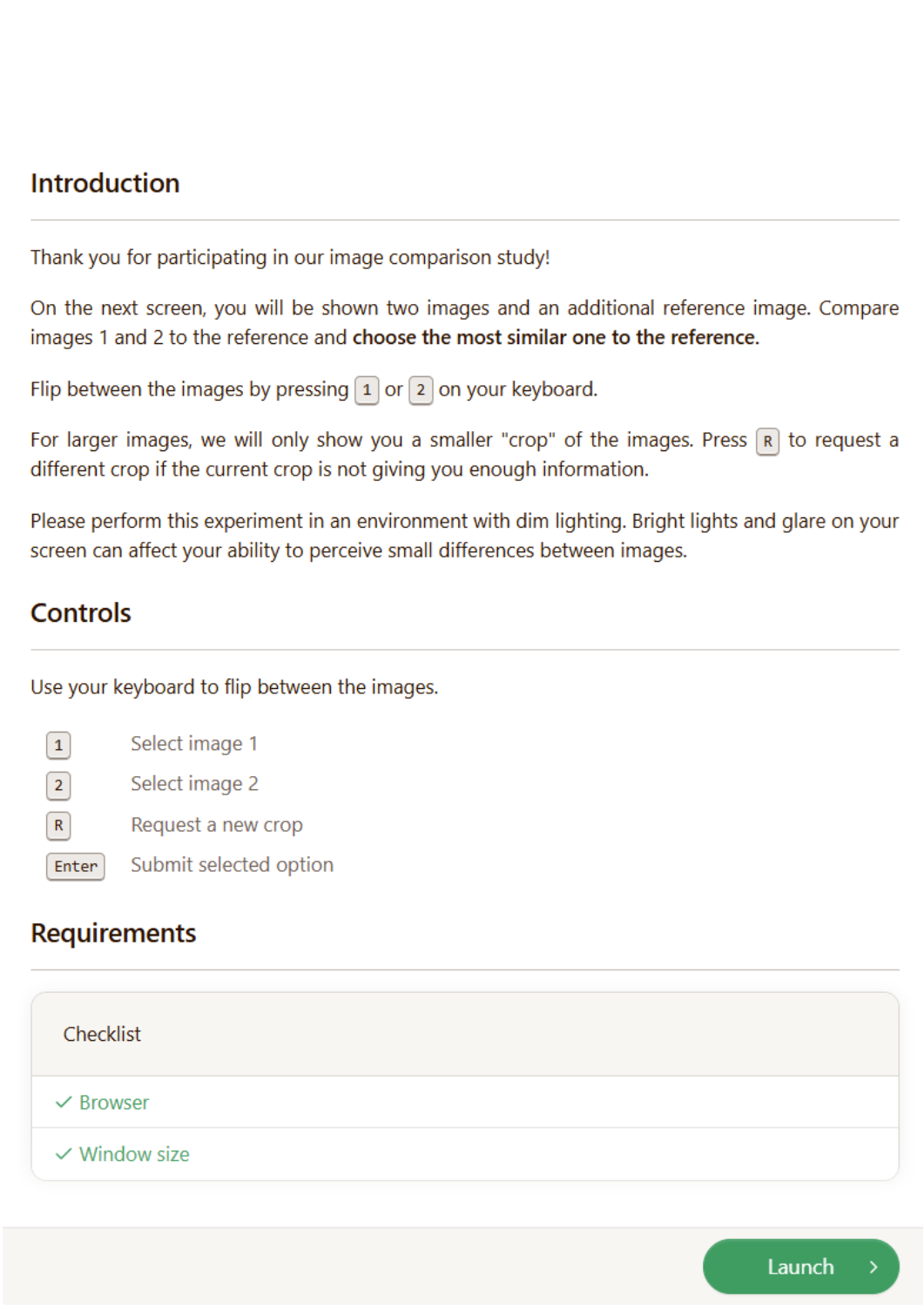


Figure 15: Instructions screen for our perceptual image comparison study, outlining the test objective, and providing guidance on using keyboard controls to navigate, crop, and submit selections.

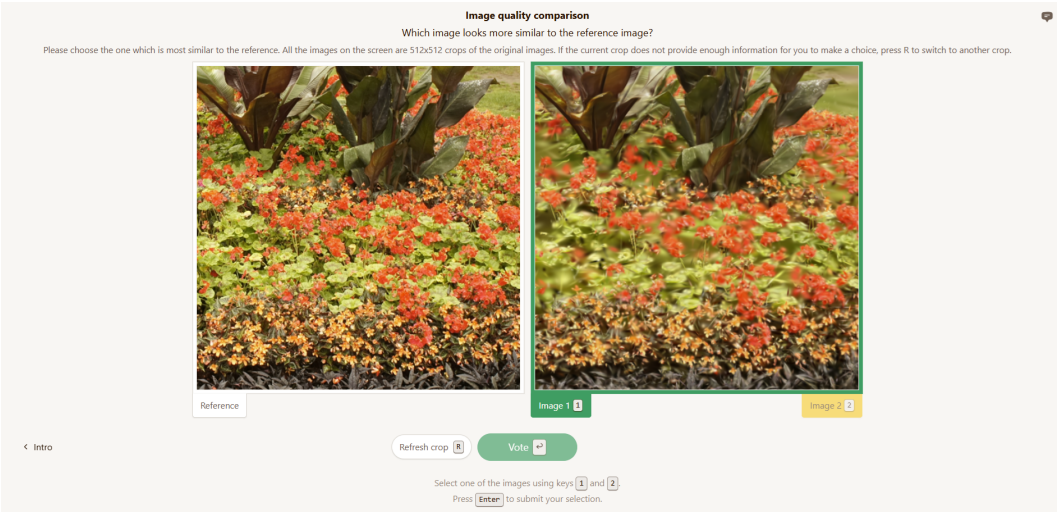


Figure 16: Interactive comparison interface in our study, showing a reference image (left) alongside two reconstructions (right) for raters to evaluate and select the one most similar to the reference.