READ: GEOMETRY-GUIDED SOURCE EVIDENCE RETRIEVAL FOR FEED-FORWARD 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward 3D Gaussian splatting enables efficient novel view synthesis, but fine appearance details can be lost when source observations are encoded into Gaussian attributes. We present **Reprojected Evidence Aggregation via Depth (READ)**, a lightweight rendering-time refinement module that uses a frozen Gaussian scene for both base rendering and geometry-guided retrieval of source appearance. READ combines posed source images with rendered RGB, depth, and accumulated alpha, without accessing the reconstruction model's internal features. Geometric reliability and learned source weights guide a bounded residual correction of the base rendering. With only 21K trainable parameters, READ improves novel-view fidelity and visual detail on DL3DV and RealEstate10K with low computational overhead. A single set of weights trained with one reconstruction model transfers across Gaussian reconstruction models, source-view counts, and datasets without retraining.

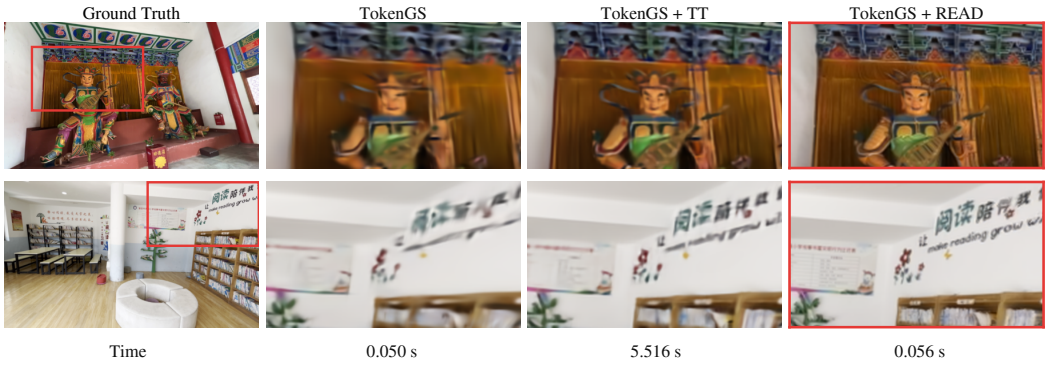

Figure 1: Qualitative comparison on DL3DV with four source views. The first column shows the ground-truth target view and marks the enlarged region. Time is the end-to-end runtime for one target view on an A100 GPU. TT denotes 50-step test-time token tuning.

1 INTRODUCTION

Novel view synthesis (NVS) aims to generate images of a scene from unseen viewpoints using a set of source observations. Neural radiance fields (NeRFs) and 3D Gaussian splatting (3DGS) have advanced this task through differentiable scene representations (Mildenhall et al., 2020; Kerbl et al., 2023). Generalizable radiance-field models condition rendering on one or several source images (Yu et al., 2021; Xu et al., 2024a). Recent feed-forward 3DGS methods instead predict explicit Gaussian scenes directly from sparse source images (Charatan et al., 2024; Chen et al., 2024a; Xu et al., 2024b; Zhang et al., 2024; Xu et al., 2025; Ren et al., 2026). By reusing the predicted scene across target viewpoints, these methods combine cross-scene generalization with efficient rendering.

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.

Preserving fine-grained appearance remains a central challenge. In both dense, image-aligned and compact, token-based representations, source observations are consolidated into Gaussian geometry and appearance before rendering. Subsequent views are therefore synthesized from the information retained in the predicted scene. Fine textures, small markings, and sharp boundaries can become blurred even when the corresponding scene structure has been reconstructed. Importantly, these details may still be clearly visible in the original source images. The source observations thus remain a valuable complement to the stored Gaussian appearance after reconstruction.

Prior work demonstrates the value of accessing source observations during view synthesis. Image-based renderers aggregate source colors and features as part of the rendering process (Wang et al., 2021; Varma T et al., 2023), while reference-based refinement methods transfer details to an initial rendering (Zhou et al., 2023; Bösiger et al., 2024; Yoon et al., 2024). For feed-forward 3DGS, the reconstructed scene provides an explicit geometric guide for this access. Our key insight is to use the reconstructed Gaussian scene and the original source observations as complementary sources of information. The Gaussian scene provides both a base rendering and geometry for locating target-relevant observations, while the source images supply fine appearance details that may not be fully retained by the scene representation.

Using this evidence reliably presents two related challenges. First, correspondence accuracy depends on the reconstructed geometry. Depth errors and occlusions can associate a target pixel with an observation of a different surface, making it essential to assess the geometric support of each retrieved observation. Second, geometrically plausible observations can still differ in their usefulness because of viewing direction and local appearance. Their contributions should therefore adapt to the target view and the available evidence. Effective refinement requires jointly accounting for geometric reliability and the complementary appearance information provided by different source views.

To this end, we propose **Reprojected Evidence Aggregation via Depth (READ)**, a lightweight rendering-time refinement module for feed-forward 3DGS. Target depth and camera geometry locate source colors and shallow features, while source depth and accumulated alpha provide geometric reliability cues. READ combines these aligned observations using geometric reliability and learned source weights to predict a bounded residual that refines the base rendering. Alongside the original source images, READ uses only rendered RGB, depth, accumulated alpha, and camera parameters, without accessing internal features of the reconstruction model. Both the reconstruction model and its Gaussian scene remain fixed. Figure 1 illustrates how READ recovers object details and fine markings that are blurred in the base rendering.

Experiments on DL3DV and RealEstate10K test whether one set of READ weights can improve novel-view quality across reconstruction models and source-view counts. Ablations assess source correspondence, depth consistency, and learned aggregation.

Our contributions are summarized as follows:

- We introduce READ, a lightweight rendering-time refinement module that uses a frozen Gaussian scene for both base rendering and geometric correspondence, retrieving source appearance to supplement the result.
- We combine direct depth-based reprojection with a 21K-parameter network for source fusion and residual refinement. READ uses source images and rendered outputs, keeping both alignment and refinement lightweight and independent of reconstruction-specific internal features.
- A single READ model trained with one frozen TokenGS configuration improves novel-view quality across the tested reconstruction models and source-view counts on DL3DV, including a higher-resolution TokenGS configuration, and transfers to RE10K without retraining, with low additional computation.

2 Related Work

2.1 Generalizable Gaussian Reconstruction

Generalizable 3DGS methods predict explicit Gaussian scenes from source images, combining learned reconstruction with efficient rasterization. A central challenge is to infer reliable geometry from sparse observations. pixelSplat (Charatan et al., 2024) learns distributions for locating Gaussian centers, while MVSplat (Chen et al., 2024a) makes cross-view matching explicit through plane-sweep cost volumes. DepthSplat (Xu et al., 2025) further strengthens geometry estimation by combining multi-view matching with pre-trained monocular depth features. GRM and GS-LRM decode dense pixel-aligned Gaussians from multi-view image tokens (Xu et al., 2024b; Zhang et al., 2024), while latentSplat represents scenes with variational feature Gaussians and a lightweight decoder (Wewer et al., 2024).

Recent work also revisits the allocation and refinement of Gaussian primitives. TokenGS uses learnable Gaussian tokens to decouple primitive prediction from image pixels (Ren et al., 2026); GlobalSplat and C3G construct compact scene representations from global or learnable tokens (Itkin et al., 2026; An et al., 2026), while TokenSplat aligns features in token space for joint pose-free reconstruction and camera estimation (Li et al., 2026). iLRM iteratively refines a compact Gaussian representation for high-resolution reconstruction (Kang et al., 2026), and ReSplat uses source-view rendering errors as gradient-free feedback for recurrent Gaussian-parameter updates (Xu et al., 2026). These approaches improve how source observations are consolidated into a renderable scene. READ complements this reconstruction stage by using the resulting geometry to access source appearance during target-view refinement.

2.2 Source-View Aggregation for Neural Rendering

Source-conditioned rendering integrates input-image appearance into view synthesis. pixelNeRF and MuRF condition radiance-field prediction on source features (Yu et al., 2021; Xu et al., 2024a); IBRNet and GNT aggregate source colors and features along target rays (Wang et al., 2021; Varma T et al., 2023). NeuRay models source visibility (Liu et al., 2022), while FWD uses depth to forward-warp features for transformer-based fusion (Cao et al., 2022). READ uses the depth of an already reconstructed Gaussian scene to retrieve target-aligned observations directly. The Gaussian renderer supplies the base image, and a small refinement network learns how to combine the retrieved appearance into a correction.

2.3 Reference-Based Rendering Refinement

Reference-based refinement transfers input-image details to an initial rendering. NeRFLiX learns inter-viewpoint mixing (Zhou et al., 2023), and MaRINeR uses deep feature matching and hierarchical detail transfer (Bösiger et al., 2024). NeRF-oriented methods also combine geometry with learned alignment: GMT uses ray-guided deformable convolutions and transformer aggregation (Yoon et al., 2024), while RoGUENeRF refines geometric reprojection with optical flow (Catley-Chandar et al., 2024). Diffusion-based approaches introduce generative priors for view synthesis or rendering enhancement, including MVSplat360, 3DGS-Enhancer, Difix3D+, and One-Shot Refiner (Chen et al., 2024b; Liu et al., 2024; Wu et al., 2025; Dong et al., 2026).

READ keeps both correspondence estimation and appearance refinement simple: rendered depth directly reprojects source colors and shallow features, and a 21K-parameter convolutional network performs reliability-weighted fusion and residual prediction. This avoids a separate learned alignment stage or generative refinement model while retaining the frozen Gaussian scene as the basis for rendering. The resulting design improves quality with low additional latency and transfers the same learned weights across reconstruction models (Section 4).

3 Method

Figure 2 outlines READ. A pre-trained model predicts a Gaussian scene, and READ retrieves and aggregates source observations to refine each target-view rendering.

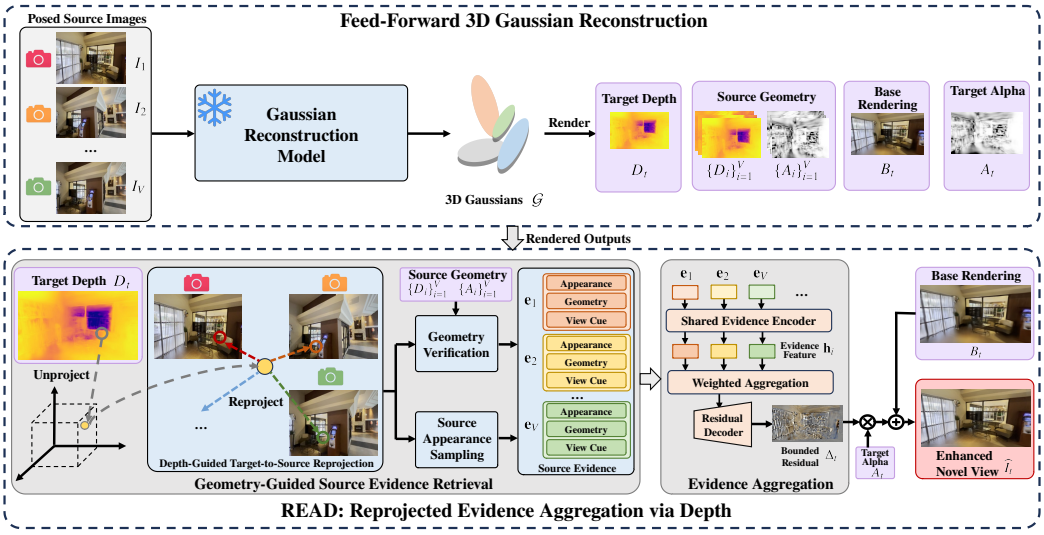

Figure 2: Overview of READ. A Gaussian scene provides rendered depth and alpha for source-view alignment and reliability estimation. READ aggregates aligned source colors and features into a bounded residual, which is scaled by target alpha and added to the base rendering.

Given V posed source RGB images $\mathcal{I}_s = \{(I_i, K_i, T_i)\}_{i=1}^{V}$, we aim to synthesize a novel RGB image at a target camera (K_t, T_t). Here, $I_i \in \mathbb{R}^{H\times W\times 3}$ denotes the i-th source image, $K_i \in \mathbb{R}^{3\times 3}$ denotes its camera intrinsics, and $T_i \in SE(3)$ denotes its camera-to-world transformation.

3.1 Rendering from a Frozen Gaussian Scene

Rendering the reconstructed scene $\mathcal{G}$ at the target camera produces RGB B_t, depth D_t, and accumulated alpha A_t. Rendering the same scene at source camera i provides D_i and A_i. Accumulated alpha is the per-pixel opacity obtained by front-to-back compositing. READ uses these rendered quantities and the original source images to predict a residual Δ_t:

$$\widehat{I}_t = B_t + A_t \odot \Delta_t, \tag{1}$$

where $\odot$ denotes element-wise multiplication, with A_t broadcast across RGB channels. Target alpha scales the output correction, while the source reliability defined below controls evidence aggregation. Both the pre-trained reconstruction model and the predicted Gaussian parameters remain fixed during READ training and inference.

3.2 Geometry-Guided Source Evidence Retrieval

Target-to-source reprojection. Rendered target depth locates candidate source observations through reprojection. For source and target cameras, write the camera-to-world pose using rotation $R_i \in SO(3)$ and camera center $\mathbf{c}_i \in \mathbb{R}^3$. The following equations use camera-z depth, with depths and translations expressed at the same scene scale. For $\mathbf{u} \in \Omega_t$, where Ω_t is the target image domain, let $\bar{\mathbf{u}} = (u_x, u_y, 1)^\top$ and $d_t = D_t(\mathbf{u})$. We compute

$$\mathbf{X}(\mathbf{u}) = R_t\big(d_t K_t^{-1}\bar{\mathbf{u}}\big) + \mathbf{c}_t, \tag{2}$$

$$\mathbf{x}_i(\mathbf{u}) = R_i^\top(\mathbf{X}(\mathbf{u}) - \mathbf{c}_i)\,, \tag{3}$$

$$\mathbf{v}_i(\mathbf{u}) = \pi(K_i\mathbf{x}_i(\mathbf{u}))\,, \tag{4}$$

where $z_i(\mathbf{u}) = [\mathbf{x}_i(\mathbf{u})]_3$ and $\pi([x, y, z]^\top) = [x/z, y/z]^\top$. The projected location $\mathbf{v}_i$ identifies the corresponding source observation, while z_i provides the expected source-view depth. The mask χ_i excludes projections outside the source camera frustum and samples with invalid depth or accumulated-alpha values.

Source appearance sampling. A shared shallow encoder first maps each source image to a feature map $\mathcal{F}_i = E_{\mathrm{sh}}(I_i)$. For valid projections, bilinear sampling at $\mathbf{v}_i$ retrieves source RGB $C_i = I_i(\mathbf{v}_i)$, shallow source feature $F_i = \mathcal{F}_i(\mathbf{v}_i)$, source depth $\widetilde{d}_i$, and source accumulated alpha $\widetilde{a}_i$. These sampled quantities are defined on the target image grid.

Geometry verification. We measure source-target depth consistency by

$$r_i = \frac{|z_i - \widetilde{d}_i|}{\max(|\widetilde{d}_i|, \epsilon)}, \qquad q_i = \exp\left[-\frac{1}{2}\left(\frac{r_i}{\tau}\right)^2\right], \tag{5}$$

$$g_i = \chi_i \widetilde{a}_i q_i, \tag{6}$$

where $\chi_i \in \{0, 1\}$ is the projection-validity mask, ϵ prevents numerical instability, and τ controls the depth-consistency bandwidth. The geometric reliability g_i combines projection validity, source accumulated alpha, and depth consistency to downweight unreliable source observations.

Per-view evidence. For each source view i, we construct a target-relevant evidence representation $\mathbf{e}_i$ by channel-wise concatenating aligned appearance cues, geometry and visibility terms, and relative viewing direction:

$$\mathbf{e}_i = [C_i,\ C_i - B_t,\ F_i,\ q_i,\ \widetilde{a}_i,\ \chi_i,\ \boldsymbol{\phi}_i], \tag{7}$$

where $[\cdot]$ denotes channel-wise concatenation, and $\boldsymbol{\phi}_i$ is a four-dimensional relative-view descriptor. Let $\mathbf{d}_i$ and $\mathbf{d}_t$ be the unit directions from the reconstructed point toward the source and target camera centers, respectively. We use

$$\boldsymbol{\phi}_i = \left[\mathbf{d}_i^\top \mathbf{d}_t,\ \tfrac{1}{2}(\mathbf{d}_i - \mathbf{d}_t)\right]. \tag{8}$$

The first component measures angular agreement, while the remaining three retain the signed directional difference.

3.3 Geometry-Aware Evidence Aggregation

The retrieved evidence from different source views is aggregated according to both learned appearance preference and geometric reliability. All source views share the same shallow encoder and evidence network, with no view-index embeddings. A shared, target-conditioned evidence encoder produces $\mathbf{h}_i = f_{\mathrm{enc}}(\mathbf{e}_i, B_t, g_i)$. The base rendering, already present in the RGB difference in $\mathbf{e}_i$, also produces a learned multiplicative modulation of the encoded feature. Geometric reliability gates this feature before and after spatial mixing. A shared scoring head predicts a source preference β_i from $\mathbf{h}_i$, which is combined with g_i to determine the aggregation weights. Let $\mathcal{V}(\mathbf{u}) = \{i \mid \chi_i = 1,\ g_i > \eta\}$ denote supported source views at target pixel $\mathbf{u}$, with numerical threshold $\eta = 10^{-8}$. We define

$$w_i = \begin{cases} \dfrac{\exp(\beta_i) g_i}{\sum_{j \in \mathcal{V}(\mathbf{u})} \exp(\beta_j) g_j}, & i \in \mathcal{V}(\mathbf{u}), \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

$$\mathbf{h} = \sum_{i=1}^{V} w_i \mathbf{h}_i. \tag{10}$$

When $\mathcal{V}(\mathbf{u})$ is empty, all source weights are zero, yielding $\mathbf{h}(\mathbf{u}) = \mathbf{0}$. Shared per-view processing and weighted summation make the aggregated feature invariant to joint reordering of source images and their associated camera and geometry inputs for a fixed Gaussian scene.

Bounded residual decoding. The aggregated target feature is decoded into a bounded RGB residual:

$$\Delta_t = \rho \tanh\left(\frac{h_{\mathrm{out}}(h_{\mathrm{dec}}(\mathbf{h}))}{\rho}\right), \tag{11}$$

where h_{dec} and h_{out} are the residual decoder and RGB prediction head, respectively, and ρ bounds the residual magnitude. The target alpha scales the residual before it is added to the base rendering according to Eq. 1. We zero-initialize the output head so that READ initially reproduces the base rendering. Appendix A.1 details the encoder configuration and training settings.

Table 1: Four-view results on DL3DV. † denotes published results; all other results use our common evaluation protocol, with the best scores in bold. Time is the end-to-end runtime for one target view; #GS denotes Gaussian count.

Method	PSNR↑	SSIM↑	LPIPS↓	Time (s)↓	#GS
MVSplat†	21.630	0.721	0.2330	0.146	458K
DepthSplat	22.481	0.759	0.1819	0.125	458K
DepthSplat + READ	23.473	**0.777**	**0.1817**	0.153	458K
TokenGS	22.792	0.732	0.3306	**0.050**	262K
TokenGS + TT	23.567	0.761	0.3071	5.516	262K
TokenGS + READ	**23.930**	0.771	0.2493	0.056	262K

3.4 Training and Base-Model Integration

We train READ after its base Gaussian reconstruction model has converged, optimizing READ with the photometric objective

$$\mathcal{L}_{\mathrm{READ}} = \|\widehat{I}_t - I_t\|_1 + \lambda_{\mathrm{ssim}} \frac{1 - \mathrm{SSIM}(\widehat{I}_t, I_t)}{2}, \tag{12}$$

where $\lambda_{\mathrm{ssim}} = 0.2$. The target image I_t provides supervision but is not an input to READ. At inference, the base rendering B_t supplies target-view appearance. READ consumes rendered RGB, depth, and accumulated alpha from the reconstruction model. Training and evaluation configurations are described in Section 4.

4 Experiments

4.1 Experimental Setup

Training. We train READ for 10,000 optimization steps on DL3DV-10K (Ling et al., 2024) using a pre-trained TokenGS model (Ren et al., 2026) with 1,024 Gaussian tokens, four source views, and 256×256 images. Only READ is optimized, using AdamW with a learning rate of 2×10^{-4}. All full-model evaluations reuse the same READ architecture and learned weights. The main DL3DV evaluation uses the higher-resolution TokenGS model with 4,096 tokens at 448×256; both models are evaluated without additional training. Component variants follow the same training and evaluation settings. Additional implementation settings, evaluation conventions, and timing procedures are provided in Appendix A; component-ablation definitions are given in Appendix B.1.

Evaluation protocols. Image resolutions are reported as width $\times$ height. We evaluate on DL3DV and RealEstate10K (RE10K) (Zhou et al., 2018) at 448×256 and 256×256, respectively. The main DL3DV evaluation uses four source views on 140 scenes, with additional tests using two, six, and eight source views. RE10K is used for cross-dataset evaluation with two source views. For local results, targets coinciding with source cameras are excluded. We report PSNR, SSIM, and LPIPS averaged within each scene and then across scenes.

Comparisons. We integrate READ with TokenGS, DepthSplat (Xu et al., 2025), and ReSplat (Xu et al., 2026). Each Base/+READ pair shares its source inputs, cameras, and Gaussian scene. Published MVSplat (Chen et al., 2024a) results follow their original protocols and are marked with †. We also compare with TokenGS test-time token tuning (TT), which adapts scene tokens to the source images for 50 optimization steps.

4.2 Main Results on DL3DV

On four-view DL3DV, READ improves TokenGS by 1.14 dB in PSNR while adding 6 ms of end-to-end latency (Table 1). SSIM and LPIPS also improve. The same READ weights improve DepthSplat's PSNR and SSIM while LPIPS remains essentially unchanged, demonstrating transfer from token-based to pixel-aligned Gaussian reconstruction. READ also outperforms TokenGS + TT in



Figure 3: Qualitative results on DL3DV with four source views. Red boxes in the first column indicate the enlarged regions.

Table 2: Generalization to two and six source views on DL3DV. † denotes published results; all other results use our common evaluation protocol, with the best quality scores at each view count in bold.

	2 views				6 views			
Method	PSNR↑	SSIM↑	LPIPS↓	#GS	PSNR↑	SSIM↑	LPIPS↓	#GS
MVSplat†	17.540	0.529	0.402	229K	22.930	0.775	0.193	688K
DepthSplat	19.036	0.599	**0.309**	229K	23.488	0.801	0.152	688K
DepthSplat + READ	19.551	0.609	0.313	229K	25.042	**0.829**	**0.146**	688K
TokenGS	19.890	0.625	0.428	262K	23.696	0.760	0.309	262K
TokenGS + TT	18.834	0.585	0.433	262K	24.847	0.796	0.283	262K
TokenGS + READ	**20.239**	**0.636**	0.383	262K	**25.534**	0.818	0.206	262K

the four-view setting. Figure 3 shows that TT retains blur in local details, whereas READ recovers clearer object boundaries and textures by accessing source appearance at the target view.

The four-view-trained READ model improves all three TokenGS metrics at two and six views and improves DepthSplat PSNR and SSIM, with a small LPIPS increase only at two views (Table 2). TT helps at six views but degrades the sparse two-view base.

At eight source views, READ improves all three metrics for TokenGS, DepthSplat, and ReSplat (Table 3). The gain on ReSplat shows that source observations remain useful even after recurrent refinement of the Gaussian scene. These results extend READ to additional reconstruction architectures and view counts. The same refinement weights operate on compact token-based scenes, pixel-aligned Gaussian predictions, and recurrently refined scenes. Across these representations, rendered geometry supplies correspondence while the original images supply appearance. This separation lets READ exploit improvements in the base reconstruction without requiring access to its internal features or a new refinement network for each architecture. Appendix C reports results with the TokenGS configuration used to train READ, and Appendix D examines variation across target views.

4.3 Cross-Dataset Transfer to RE10K

READ transfers to RE10K with both the reconstruction model and refinement weights trained on DL3DV. It improves all three TokenGS metrics (Table 4), increasing PSNR and SSIM while narrowing the LPIPS gap to the pixel-aligned baselines. Figure 4 shows clearer object boundaries and local textures, illustrating the benefit of source-guided refinement on the new dataset.

Table 3: Eight-view novel-view synthesis on DL3DV.

Method	PSNR↑	SSIM↑	LPIPS↓
DepthSplat	24.354	0.835	0.131
DepthSplat + READ	26.818	0.878	0.114
TokenGS	24.600	0.789	0.280
TokenGS + READ	27.505	0.869	0.157
ReSplat	27.962	0.888	0.116
ReSplat + READ	**28.914**	**0.905**	**0.097**

Table 4: Novel-view synthesis on RE10K with two source views.

Method	PSNR↑	SSIM↑	LPIPS↓
MVSplat	26.281	0.867	0.130
DepthSplat	27.422	0.888	**0.114**
TokenGS	27.481	0.886	0.169
TokenGS + READ	**28.336**	**0.895**	0.136

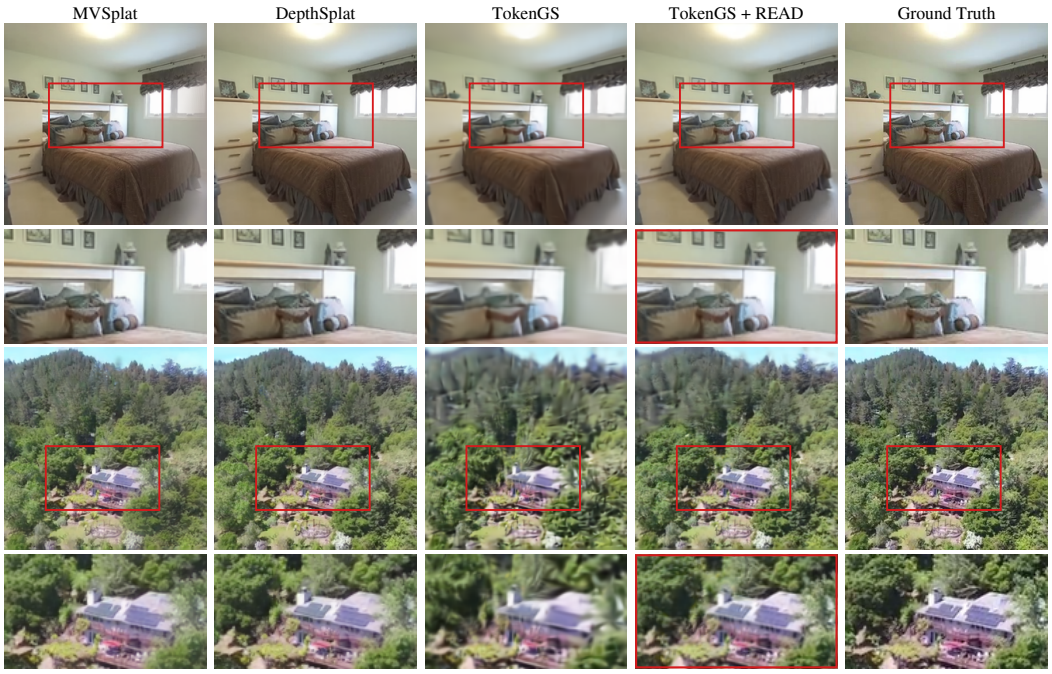

Figure 4: Qualitative results on RE10K with two source views. Each detail row enlarges the boxed region above.

4.4 Ablation Studies

Full READ improves all three metrics over the frozen base (Table 5). The base-image-only CNN provides little improvement, supporting the value of source evidence. Replacing depth consistency with a constant in the evidence and reliability computation largely removes the gain, and shuffling source cameras degrades performance below the base. These controls demonstrate the importance of locating and selecting geometrically supported observations. Uniform and fixed reliability weighting also improve quality with trained evidence encoders and residual decoders. Relative to uniform weighting, learned source preferences further improve all three metrics, adding about 0.3 dB in PSNR.

Together, these results support a division of labor between retrieval and fusion. Geometric correspondence makes source detail available at the target pixel, while learned weighting selects among observations with different appearance and reliability. The strong uniform-weight variant highlights the value of the retrieved evidence; the additional gain from learned preferences shows the benefit of adapting its combination to the target view. Both are implemented within the same small refinement network. Appendix B details the variants.

Table 5: Component ablations on DL3DV; camera shuffling is applied only at inference.

Variant	PSNR↑	SSIM↑	LPIPS↓
Frozen TokenGS base	22.792	0.732	0.331
Full READ	**23.930**	**0.771**	**0.249**
Base-image-only CNN	22.783	0.731	0.330
Without depth consistency	22.896	0.709	0.318
Uniform source weighting	23.625	0.764	0.262
Fixed reliability weighting	23.618	0.764	0.263
Shuffle source cameras (inference)	21.867	0.688	0.365

Table 6: Time (s) is the end-to-end runtime for one target view on an A100 GPU. Overheads are relative to TokenGS; TT includes one-time scene adaptation.

#Views	TokenGS	+READ	READ overhead	+TT	TT overhead
2	0.039	0.044	+0.005	4.574	+4.534
4	0.050	0.056	+0.006	5.516	+5.466
6	0.064	0.072	+0.008	6.423	+6.359

4.5 Efficiency

READ adds 5–8 ms of single-target end-to-end latency to TokenGS across two to six source views (Table 6), and 28 ms to the four-view DepthSplat path (Table 1). These increments include additional rendering, evidence retrieval, and residual prediction. READ uses a feed-forward correction at each target view, with no test-time optimization. Direct reprojection supplies the alignment, allowing the small convolutional network to focus on combining evidence and predicting the correction. In the four-view TokenGS setting, this design raises PSNR from 22.792 to 23.930 dB while increasing latency from 50 to 56 ms. The quality gain therefore comes with a small increment to the complete rendering path, including the cost of retrieving source evidence. Appendix A.3 details the timing procedure.

4.6 Limitations

READ depends on source-view coverage and the accuracy of the reconstructed geometry. Incorrect depth can misalign source observations, and systematic geometry errors may remain consistent across rendered views. Regions unseen in all source images offer no directly retrievable detail, while reflections and exposure differences complicate appearance transfer. READ also requires access to source observations and camera information at rendering time, introducing storage and memory requirements beyond the Gaussian scene alone.

5 Conclusion

We presented READ, a lightweight module that refines Gaussian renderings using geometrically aligned source observations. Depth-based reliability and learned source weighting guide a bounded, alpha-gated residual correction. One model trained with TokenGS improves multiple Gaussian reconstruction models on DL3DV and transfers to a higher-resolution TokenGS configuration and to RE10K. These results show that source observations can complement stored Gaussian appearance and improve novel-view quality with low additional computation.

AI USE STATEMENT

Generative AI was used to assist in outlining, drafting, and language refinement for this work. All final arguments, analysis, factual verification, and editorial decisions remain solely the responsibility of the author.

REPRODUCIBILITY STATEMENT

The method section describes READ's source-evidence construction, geometric reliability, aggregation, and training objective. Appendix A provides the architecture, training settings, pre-trained TokenGS checkpoints, evaluation conventions, and timing procedures. Appendix B specifies the component variants and correspondence tests. Additional evaluations are provided in Appendices C and D. All full-model evaluations use the same learned READ weights, with each base/+READ pair sharing an identical Gaussian scene.

REFERENCES

Honggyu An, Jaewoo Jung, Mungyeom Kim, Chaehyun Kim, Minkyeong Jeon, Jisang Han, Kazumi Fukuda, Takuya Narihira, Hyunah Ko, Junsu Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. Learning compact 3D representations from feed-forward novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 363–373, 2026.

Lukas Bösiger, Mihai Dusmanu, Marc Pollefeys, and Zuria Bauer. MaRINeR: Enhancing novel views by matching rendered images with nearby references. In *European Conference on Computer Vision*, pp. 76–94. Springer, 2024. doi: 10.1007/978-3-031-72673-6_5.

Ang Cao, Chris Rockwell, and Justin Johnson. FWD: Real-time novel view synthesis with forward warping and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15713–15724, 2022.

Sibi Catley-Chandar, Richard Shaw, Gregory Slabaugh, and Eduardo Pérez-Pellitero. RoGUENeRF: A robust geometry-consistent universal enhancer for NeRF. In *European Conference on Computer Vision*, pp. 54–71. Springer, 2024. doi: 10.1007/978-3-031-73254-6_4.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386. Springer, 2024a.

Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. MVSplat360: Feed-forward 360 scene synthesis from sparse views. In *Advances in Neural Information Processing Systems*, volume 37, pp. 107064–107086, 2024b.

Yitong Dong, Qi Zhang, Minchao Jiang, Zhiqiang Wu, Qingnan Fan, Ying Feng, Huaqi Zhang, Hujun Bao, and Guofeng Zhang. One-shot refiner: Boosting feed-forward novel view synthesis via one-step diffusion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(5): 3659–3667, 2026. doi: 10.1609/aaai.v40i5.37365.

Roni Itkin, Noam Issachar, Yehonatan Keypur, Xingyu Chen, Anpei Chen, and Sagie Benaim. GlobalSplat: Efficient feed-forward 3D gaussian splatting via global scene tokens. *arXiv preprint arXiv:2604.15284*, 2026. Accepted to ECCV 2026.

Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. iLRM: An iterative large 3D reconstruction model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37332–37342, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, July 2023. doi: 10.1145/3592433.

Yihui Li, Chengxin Lv, Zichen Tang, Hongyu Yang, and Di Huang. TokenSplat: Token-aligned 3D gaussian splatting for feed-forward pose-free reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40886–40895, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Xi Liu, Chaoyi Zhou, and Siyu Huang. 3DGS-Enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *Advances in Neural Information Processing Systems*, volume 37, pp. 133305–133327, 2024.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7824–7833, 2022.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. TokenGS: Decoupling 3D gaussian prediction from pixels with learnable tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15365–15375, 2026.

Mukund Varma T, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that NeRF needs? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xE-LtsE-xx.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2021.

Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentSplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European Conference on Computer Vision*, pp. 456–473. Springer, 2024. doi: 10.1007/978-3-031-73021-4_27.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. DIFIX3D+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26024–26035, 2025.

Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. MuRF: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20041–20050, 2024a.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16453–16463, 2025.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. ReSplat: Learning recurrent gaussian splatting. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2510.08575.

Yinghao Xu, Zifan Shi, Yifan Wang, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2024b. doi: 10.1007/978-3-031-72633-0_1.

Youngho Yoon, Hyun-Kurl Jang, and Kuk-Jin Yoon. GMT: Enhancing generalizable neural rendering via geometry-driven multi-reference texture transfer. In *European Conference on Computer Vision*, pp. 274–292. Springer, 2024. doi: 10.1007/978-3-031-73001-6_16.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024. doi: 10.1007/978-3-031-72670-5_1.

Kun Zhou, Wenbo Li, Yi Wang, Tao Hu, Nianjuan Jiang, Xiaoguang Han, and Jiangbo Lu. NeRFLiX: High-quality neural view synthesis by learning a degradation-driven inter-viewpoint MiXer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12363–12374, 2023.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):65:1–65:12, August 2018. doi: 10.1145/3197517.3201323.

APPENDIX

A IMPLEMENTATION AND EVALUATION DETAILS

A.1 NETWORK AND TRAINING

Table 7 summarizes READ's layers and settings. The shallow source encoder, per-view evidence encoder, and source-scoring head share their parameters across source views. The evidence representation has 21 channels: source RGB, RGB–base difference, eight shallow feature channels, depth consistency, source accumulated alpha, projection validity, and the four-dimensional relative-view descriptor. The RGB output head is zero-initialized, so READ initially reproduces the base rendering.

The target-aligned evidence $\mathbf{e}_i \in \mathbb{R}^{21\times H\times W}$ is encoded as

$$\begin{aligned} \mathbf{m}_t &= 2\sigma(P_b(B_t)), & \mathbf{z}_i &= \mathrm{GELU}(P_e(\mathbf{e}_i)) \odot \mathbf{m}_t \odot g_i, \\ \mathbf{h}_i &= \mathrm{GELU}(P_s(\mathbf{z}_i)) \odot g_i, & \beta_i &= P_\beta(\mathbf{h}_i). \end{aligned} \tag{13}$$

Here σ is the sigmoid function; $\mathbf{m}_t$, $\mathbf{z}_i$, and $\mathbf{h}_i$ have 24 channels, while β_i is a single-channel logit. The modulation $\mathbf{m}_t$ is shared across source views, and the scalar reliability g_i is broadcast across feature channels. All convolutions use unit stride and preserve spatial resolution. Only P_b has a bias.

Table 7: READ architecture and numerical settings.

Component	Configuration
Shallow source encoder	$3\to8$ convolution, GELU, $8\to8$ convolution, GELU; 3×3 kernels
Evidence projection P_e	$21\to24$ convolution, 3×3 kernel, GELU
Base-rendering modulator P_b	$3\to24$ convolution, 1×1 kernel, bias, followed by $2\sigma(\cdot)$
Per-view spatial encoder P_s	$24\to24$ convolution, 3×3 kernel, GELU; reliability gating as in Eq. 13
Source-scoring head P_β	$24\to1$ convolution, 1×1 kernel, no activation
Residual decoder	Two $24\to24$ convolution–GELU layers with 3×3 kernels; zero-initialized $24\to3$ RGB output head with a 1×1 kernel
Numerical settings	$\tau=0.02$, $\rho=0.25$, $\epsilon=10^{-3}$, support threshold $\eta=10^{-8}$
Trainable parameters	21,072

For numerical stability, source lanes with $g_i \le \eta$ are excluded from aggregation. We compute cross-view weights from $\beta_i + \log\max(g_i, \eta)$, mask unsupported lanes, and renormalize the remaining weights. An empty support set produces a zero aggregated feature.

READ is trained on DL3DV-10K for 10,000 optimization steps with four source views at 256×256. We use AdamW in FP32 with a learning rate of 2×10^{-4}, a 200-step warm-up, and a cosine schedule configured for a 150,000-step horizon. Training uses two GPUs with 16 samples per GPU. Only READ is optimized; the pre-trained reconstruction model and its Gaussian outputs remain fixed. All evaluations of the full READ model use the same architecture and the weights obtained after 10,000 steps. Independently trained component variants are described in Appendix B.

A.2 RECONSTRUCTION MODELS AND EVALUATION

Image resolutions are reported as width × height. READ is trained with the released TokenGS checkpoint `dl3dv_base.safetensors`, which uses 1,024 Gaussian tokens and emits 65,536 Gaussians. The main DL3DV TokenGS evaluation instead uses `dl3dv_4v.safetensors`, the released checkpoint fine-tuned with 4,096 tokens at 448×256, which emits 262,144 Gaussians. Each token produces 64 Gaussians in both configurations. The same READ weights are also evaluated with released DepthSplat and ReSplat models. ReSplat uses its native eight-view setting; the eight-view TokenGS and DepthSplat evaluations extend beyond their training view-count ranges. On RE10K, TokenGS and READ retain their DL3DV-trained weights.

The target and source depths D_t and D_i are expressed as camera-z depth in the same scene coordinates as the camera translations. Following the TokenGS preprocessing, camera translations are

scaled by 0.15 for DL3DV and 0.25 for RE10K. Accumulated alpha is obtained from each model's native front-to-back Gaussian rasterization. Nonfinite samples, projections outside the viewing frustum, and source-depth samples behind the near plane are marked invalid before computing depth consistency.

All local base/+READ comparisons use identical source inputs, cameras, and Gaussian scenes. We exclude target cameras that coincide with source views and average PSNR, SSIM, and LPIPS first within each scene and then across scenes. The four-view DL3DV evaluation contains 140 test scenes and 6,440 novel target views. Published reference values marked with † in the main paper retain their original evaluation protocols.

A.3 Timing Measurements

We measure single-target end-to-end latency on an A100 80GB PCIe GPU with batch size one. For a READ-augmented model, the measured computation includes Gaussian reconstruction, all required target and source renders, source-evidence retrieval, and residual prediction. Data loading, host-to-device transfer, and metric computation are excluded. TokenGS inference uses BF16, DepthSplat uses its native FP32 rasterizer, and READ runs in FP32.

For TokenGS test-time token tuning (TT), the measurements in Table 6 include reconstruction, 50 scene-adaptation steps, and one target render. TT adaptation is performed once per scene and can be amortized over multiple targets, whereas READ refines each target view during rendering.

B Component Ablations and Correspondence Tests

B.1 Independently Trained Component Variants

Each component variant in Table 5 is independently trained for 10,000 steps with a zero-initialized RGB output head. The reconstruction model, optimizer, batch size, and learning-rate schedule match full READ. Training uses the 1,024-token TokenGS model at 256×256, and evaluation uses the 4,096-token model at 448×256. The network structure is retained when selected inputs are disabled or source weights are replaced.

Base-image-only CNN. Sampled source RGB, shallow source features, geometric cues, and relative-view cues are withheld from the per-view evidence. The base rendering remains available through the target-conditioning path. This variant evaluates refinement without aligned source appearance.

Without depth consistency. We set $q_i = 1$ in both the evidence representation and the computation of g_i. Projection validity and source accumulated alpha remain active, so $g_i = \chi_i \widetilde{a}_i$. This removes depth-consistency weighting while retaining the other geometric validity checks.

Uniform source weighting. For each target pixel, we assign equal weights to the same set $\mathcal{V}(\mathbf{u})$ of source views with nonzero geometric reliability. Thus, $w_i = 1/|\mathcal{V}(\mathbf{u})|$ within a nonempty set, and unsupported views receive zero weight. An empty set yields zero aggregated features. The evidence encoder and residual decoder are still trained.

Fixed reliability weighting. We replace learned source preferences with normalized geometric reliability, using $w_i = g_i / \sum_{j \in \mathcal{V}(\mathbf{u})} g_j$ over the supported views. The weights are applied to the encoded per-view features, and the evidence encoder and residual decoder remain trainable. This variant tests the contribution of learned source preferences while retaining learned evidence encoding and residual prediction.

B.2 Inference-Time Correspondence Tests

We separately perturb the inputs used for source retrieval while keeping the full READ weights and the Gaussian scene fixed. Table 8 reports permutations of source RGB, source-camera assignments, and target depth. These tests change the appearance supplied for retrieval, the image–camera correspondence, or the 3D points used for reprojection, respectively.

Table 8: Effect of correspondence perturbations on DL3DV.

Setting	PSNR↑	SSIM↑	LPIPS↓
TokenGS	22.792	0.732	0.331
Full READ	23.930	0.771	0.249
Shuffle source RGB	17.781	0.499	0.459
Shuffle source cameras	21.867	0.688	0.365
Shuffle target depth	22.905	0.735	0.321

All three perturbations reduce performance relative to full READ, supporting its dependence on source appearance and the geometry used to locate that appearance. Source RGB and camera permutations also reduce quality below the base rendering, showing that incorrect evidence can produce harmful residual corrections.

C Evaluation with the Training-Time TokenGS Model

We also evaluate READ on the DL3DV test scenes using the same TokenGS configuration that was used to train it: 1,024 Gaussian tokens, 65,536 Gaussians, and 256×256 images. The 140 test scenes and 6,440 novel target views match the main four-view evaluation. Table 9 shows improvements in all three metrics with this reconstruction model.

Table 9: DL3DV test results with the 1,024-token TokenGS model at 256×256.

Method	PSNR↑	SSIM↑	LPIPS↓
TokenGS	20.709	0.623	0.444
TokenGS + READ	**22.131**	**0.701**	**0.302**

Together with Table 1, these results show that one set of READ weights is effective with both TokenGS configurations. Refinement remains beneficial when the reconstruction checkpoint, Gaussian token budget, and image resolution change together.

D PERFORMANCE ACROSS TARGET VIEWS

To examine how refinement varies along a sequence, we use the seven DL3DV test scenes whose prescribed source-view indices are $[0, 17, 28, 49]$. At each of the 50 target indices, we average the paired TokenGS and TokenGS + READ PSNR values across scenes using the main four-view evaluation outputs.

Figure 5 shows the largest gains at target cameras that coincide with source views. Gains also remain positive on average over the 46 novel targets per scene, with a mean PSNR improvement of 1.37 dB. This analysis characterizes how the refinement magnitude varies along the evaluated sequences.

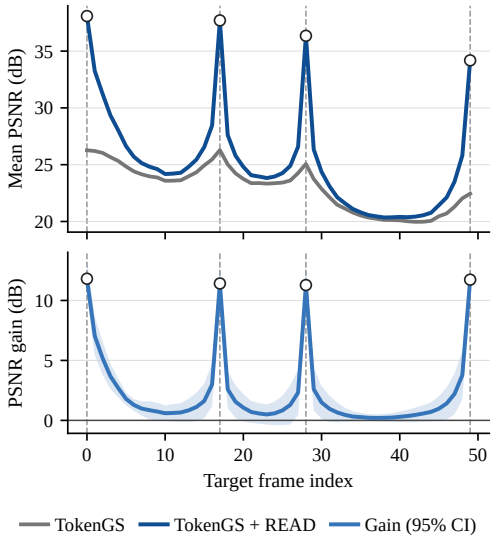

Figure 5: PSNR across seven DL3DV sequences. Dashed lines and open markers indicate source-view targets. The lower panel shows the mean paired READ gain; shading denotes a pointwise two-sided 95% Student-t confidence interval across the seven scenes.