# REMIXGS: SCALABLE FEED-FORWARD GAUSSIAN SPLATTING WITH VIEW-REGISTER MIXING AND INTERLEAVED GAUSSIAN QUERIES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Feed-forward Gaussian Splatting models predict 3D Gaussians from multi-view inputs in a single network pass, replacing iterative per-scene optimisation. The prevalent methods nevertheless scale poorly in viewpoint span, input views, and representation fidelity. Their joint attention involves every patch token of every view, although cross-view alignment is chiefly geometric; their decoders entangle geometry and appearance, allowing geometric errors to be baked into appearance. This paper presents RemixGS, a novel feed-forward Gaussian Splatting method that efficiently reconstructs high-fidelity scenes from large-scale multi-view inputs. Specifically, RemixGS pairs a scalable alternating multi-view encoder with a geometry-texture interleaved Gaussian decoder. The encoder alternates between a local block confined to a single view and a global view-register mixer spanning all views, routing all cross-view interaction through a few per-view registers instead of dense attention across all patch tokens. The decoder carries a fixed set of query tokens through interleaved geometry and texture blocks, where texture reads geometry but not vice versa, preventing geometric errors from being baked into appearance. RemixGS is first trained at a base input and representation scale and then extended to larger-scale inputs and finer representations. Experiments show that our RemixGS consistently surpasses state-of-the-art methods in reconstruction quality as the input scale increases, with substantially lower training and inference cost. Code will be publicly released to facilitate reproducible research.

## 1 INTRODUCTION

Reconstructing 3D scenes from multi-view images and rendering novel views is a central problem in computer vision. The popular 3D Gaussian Splatting (Kerbl et al., 2023) represents a scene with explicit primitives and renders it in real time, but still requires per-scene optimisation. Feed-forward methods replace the slow iterative process with a single network pass, mapping multi-view inputs directly to 3D Gaussians (Charatan et al., 2024; Chen et al., 2024; Zhang et al., 2024). Reconstruction quality is therefore no longer determined by the compute spent on an individual scene but by the model capacity and the training data, and the paradigm is in principle free to improve with both.

Nevertheless, the prevalent feed-forward methods scale poorly in viewpoint span, input views, and representation fidelity. They typically take two to six input views with substantial pairwise overlap and run at low resolution (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025). Real capture instead yields tens to hundreds of images at megapixel resolution, with wider baselines between neighbouring views and smaller co-visible regions (Ling et al., 2024; Ziwen et al., 2025). Most of these methods predict Gaussians on a per-pixel grid (Charatan et al., 2024; Chen et al., 2024), so the representation grows with views and resolution while the scene itself does not change.

The information is carried throughout by the per-view patch tokens in the typical feed-forward architecture (Zhang et al., 2024; Ziwen et al., 2025; Jin et al., 2025; Ren et al., 2026). However, what must align across views is chiefly geometric, since no single view settles where a surface lies. Joint attention still makes every patch token of every view take part, and most of that exchange is redundant. A sound architecture should remove it and retain only the necessary global interaction



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.

between views. The fused multi-view features are then aggregated and decoded into Gaussians, where Gaussians' attributes are usually treated as a whole (An et al., 2026; Ren et al., 2026), leaving geometry and appearance entangled. Under photometric supervision alone, a geometry that departs from the truth may leave a residual that appearance absorbs (Zhang et al., 2020), lowering the loss while the geometric error remains. The architectural design should also keep geometric errors from being baked into appearance.

This paper presents RemixGS, a novel feed-forward Gaussian Splatting method that efficiently reconstructs high-fidelity scenes from large-scale multi-view inputs. Specifically, our RemixGS pairs a scalable alternating multi-view encoder with a geometry-texture interleaved Gaussian decoder. The encoder establishes relations within and across views at patch and structural granularity, respectively, by alternating between a local block confined to a single view and a global view-register mixer spanning all views. The mixer carries all cross-view interaction through a few learned registers held by each view in three steps (Darcet et al., 2024; Wang et al., 2026a). The read step compresses what a view observes into its own registers, the global step is the only one at which two views are related, and the write step returns the result to the patch tokens of the view that produced it. Patch tokens are absent from the global step by construction, so dense attention across the patch tokens of all views is never built. The decoder starts from a fixed set of learned query tokens and carries them through interleaved geometry and texture blocks, and each query token decodes a group of Gaussians (Ren et al., 2026; Itkin et al., 2026; Li et al., 2026c) whose geometric and colour attributes come from the two kinds of block respectively. Texture blocks read the geometry state while geometry blocks never read theirs, so appearance is never in a position to absorb a geometric residual. RemixGS is first trained at a base input and representation scale, and a further training stage extends this base model to larger-scale inputs and a finer scene representation. We compare RemixGS with state-of-the-art methods on inputs that vary in the number of views, in resolution and in viewpoint span. RemixGS consistently surpasses them in reconstruction quality as the input scale increases, leading in PSNR by up to 2.21 dB, while training in at least 30% less time and running inference faster.

The contributions of this paper are summarised as follows:

- We propose a scalable alternating multi-view encoder that models intra-view interactions among patches but routes inter-view interactions through a few per-view registers, cutting redundant dense attention across all patch tokens.
- We design a geometry-texture interleaved Gaussian decoder that predicts geometric and texture attributes in alternating blocks, where texture reads geometry but not vice versa, preventing geometric errors from being baked into appearance.
- We conduct extensive experiments showing that RemixGS can efficiently extend to larger-scale inputs and consistently surpass state-of-the-art methods in reconstruction quality with lower training and inference cost.

## 2 Related Work

### 2.1 Feed-forward Gaussian reconstruction

Early methods place a Gaussian on every input pixel ray, leaving depth as its only free parameter, so the prediction amounts to depth estimation (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025). The number of Gaussians then increases with the number of input pixels. Query-based methods remove this dependence. The scene is carried by a fixed number of query tokens, into which multi-view features are read by attention, and each token decodes a group of Gaussians (Ren et al., 2026; Itkin et al., 2026; Li et al., 2026a). The number of Gaussians is then set by the model. Alongside the images, most feed-forward models take camera poses as an input, usually as a Plücker ray map used as the positional embedding (Zhang et al., 2024; Ziwen et al., 2025; Kang et al., 2026; Ren et al., 2026; Ziwen et al., 2026; Zhang et al., 2026; Cho et al., 2026; Li et al., 2026a). A parallel line (Ye et al., 2025; Li et al., 2026b; Jiang et al., 2025b; Veicht et al., 2026; Li et al., 2026c; Peng et al., 2026) removes that requirement by building on feed-forward geometry models (Wang et al., 2024; 2025a), and some drop pose supervision entirely (Jiang et al., 2025a). As these geometry models improve, accurate poses become cheaper to obtain outside the reconstruction model, so whether pose estimation belongs inside it remains open.

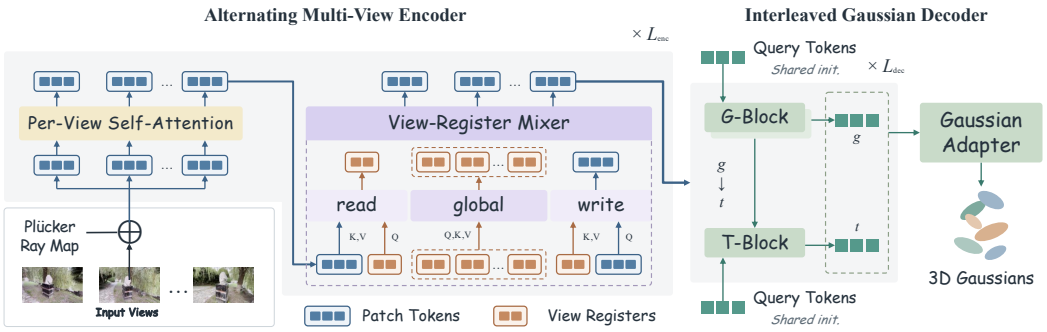


Figure 1: Overview of RemixGS. Each view is patch-embedded together with its Plücker ray map and processed by local attention confined to that view. A view-register mixer carries all cross-view interaction through the $R$ registers of each view, which read the patch tokens of their own view, attend among all $VR$ registers, and write the result back. The decoder starts from $N$ query tokens, carries them through interleaved geometry and texture blocks with one-way guidance from geometry to texture, and emits $G$ Gaussians per token.

### 2.2 Multi-view feature aggregation

Feed-forward methods fuse views by establishing correspondences, explicitly or implicitly. Explicit matching estimates depth through epipolar attention (Charatan et al., 2024) or cost volumes (Chen et al., 2024; Xu et al., 2025). Implicit matching instead uses cross-attention between two views to regress point maps or Gaussians (Wang et al., 2024; Smart et al., 2024), and combining more than two views requires a separate global alignment. Some methods extend cross-attention to multiple views while within-view self-attention stays separate (Tang et al., 2025; Ye et al., 2025). Recent methods commonly concatenate features from all views into a single sequence and apply global self-attention (Yang et al., 2025; Wang et al., 2025a; 2026b; Zhang et al., 2024; Ren et al., 2026). One line lowers the cost of that sequence, either by shortening it (Shen et al., 2026; Wang & Xu, 2026; Huang et al., 2026) or by replacing attention with state-space layers (Ziwen et al., 2025; Dao & Gu, 2024). A third line routes cross-view traffic through a small set of carriers that also decode, so the channel and the output are sized together (Wang et al., 2025b; Itkin et al., 2026; Kang et al., 2026). Register-based exchange thins the channel rather than the sequence, routing information through per-view registers in a quarter of the global layers, and replacing all of them costs accuracy (Wang et al., 2026a). In each case the width of the cross-view channel is fixed by the input or the output. Mechanisms for extracting the little that is needed from a large input already exist, reading a growing input into a fixed set of learned vectors that later stages access in its place (Jaegle et al., 2021; 2022), or distilling a few targets from a redundant one with a small set of queries or slots (Carion et al., 2020; Locatello et al., 2020).

### 2.3 Token-based Gaussian decoding

The query tokens are learned parameters, fixed in number, from which the Gaussians are decoded. Some methods decode a single Gaussian from each token, which destabilises training (An et al., 2026). Others decode a group of Gaussians from each token, which increases capacity without adding tokens, but a group often spreads across the scene (Li et al., 2026a). Inspired by per-scene optimisation methods, which grow nearby Gaussians as offsets from sparse anchors (Lu et al., 2024), several works give the group an origin of its own, supplied by an extra branch or an external prior (Peng et al., 2026; Cho et al., 2026; Zhang et al., 2026) or by a state stored for each group (Li et al., 2026a). Gaussians have geometric and appearance attributes, usually decoded from the same token for simplicity. Some works predict them in separate decoder streams, to keep texture from masking structural error (Itkin et al., 2026) or to reduce the blur that the entanglement produces (Li et al., 2026c).

# 3 METHODOLOGY

Given $V$ posed input images $\{I_v\}_{v=1}^{V}$, each view and its per-pixel Plücker ray map are patch-embedded into $P$ tokens,

$$x_v = \mathrm{PatchEmbed}_{\mathrm{rgb}}(I_v) + \mathrm{PatchEmbed}_{\mathrm{ray}}(\Pi_v) \in \mathbb{R}^{P\times D}, \tag{1}$$

where $\Pi_v$ is the Plücker map of view $v$. From these $VP$ tokens our feed-forward model RemixGS predicts in a single forward pass a set of 3D Gaussians that can be rendered from arbitrary target viewpoints. Our RemixGS model consists of a scalable alternating multi-view encoder and a geometry-texture interleaved Gaussian decoder. The encoder alternates a per-view local transformer block with a view-register mixer, which moves all cross-view information through the $R$ learned registers of each view, so no dense attention over the $VP$ patch tokens is ever formed. The decoder starts from $N$ learned query tokens and carries them through interleaved geometry and texture blocks. Each query token decodes $G$ Gaussians, whose geometric and colour attributes come from the two kinds of block respectively, so the total $NG$ is set by the model rather than by the view count or the resolution. RemixGS is first trained at a base resolution and scale, and a further training stage extends this base model to larger-scale inputs and a finer scene representation.

## 3.1 SCALABLE ALTERNATING MULTI-VIEW ENCODER

The encoder receives the patch tokens of all $V$ input views, which already carry both the colour of each patch and its ray, and returns the features from which geometry and appearance are decoded. The position of a surface along a known ray is settled only where several views observe it, whereas appearance is largely settled within one view, so what must pass between views is chiefly geometric. Existing models pass the patch tokens themselves, applying joint attention across all $VP$ of them. The cost is $O(V^2P^2D)$ and grows quadratically in both the view count and the resolution, since every patch token takes part in the exchange between views. The encoder is therefore one of the main obstacles to scaling a feed-forward model to more input views and higher resolution. The geometric information that crosses views is far smaller than the patch tokens that would carry it, so the amount exchanged between views can be fixed by the architecture rather than by the view count and the resolution. The closest existing design, VGGT-$\Omega$ (Wang et al., 2026a), gives each view its own registers but uses them in only part of its global layers, so dense attention over the patch tokens still sets the asymptotic cost. Our encoder accordingly establishes relations within a view at patch granularity and relations across views at structural granularity, alternating a local attention block confined to a single view with a view-register mixer that carries all cross-view interaction through $R$ learned registers (Darcet et al., 2024) held by each view. The encoder stacks $L_{\mathrm{enc}}$ layers, each holding this pair.

**Per-View Self Attention.** Let $x \in \mathbb{R}^{BV\times P\times D}$ hold the patch tokens with the views folded into the batch dimension, so that a standard transformer block applied to this tensor attends over the $P$ tokens of one view and over nothing else. Nothing inside a view can be compressed the way the exchange between views can, because the appearance a view reports is defined at the resolution of its own pixels, so this block stays at patch granularity throughout. The views are independent at this stage, and a further view adds one more evaluation of the block without altering the others.

**View-Register Mixer.** Let $r \in \mathbb{R}^{BV\times R\times D}$ hold the registers, initialised from a single learned bank shared across views, and let $\tilde{r} \in \mathbb{R}^{B\times VR\times D}$ denote the same state with the per-view axis unfolded into one sequence. A mixer comprises three pre-norm residual branches, one for each step and each gated by a LayerScale,

$$r \leftarrow r + \mathrm{LS}_{\mathrm{read}}\big(\mathrm{CrossAttn}(r,\, x)\big), \tag{2}$$

$$\tilde{r} \leftarrow \tilde{r} + \mathrm{LS}_{\mathrm{glob}}\big(\mathrm{SelfAttn}(\tilde{r})\big), \tag{3}$$

$$x \leftarrow x + \mathrm{LS}_{\mathrm{write}}\big(\mathrm{CrossAttn}(x,\, r)\big), \tag{4}$$

The read step (Equation 2) compresses what a view observes into its own $R$ vectors, the global step (Equation 3) is the only one at which two views are related, and the write step (Equation 4) returns the result to the patch tokens of the view that produced it. Patch tokens are absent from the global step by construction, so no dense $VP \times VP$ attention is ever built. The mixer uses no view-identity embedding, so the encoder is equivariant to permutations of the input views, no view serves as a reference, and a model trained at one view count is not structurally specialised to it. The mixer

parameters need not be tied across layers and the register state need not be reset at each of them, so the cross-view exchange can be deepened either by adding mixers or by reusing one and carrying the state forward, the second of which costs no parameters.

The cost of one encoder layer settles what the arrangement buys. Joint attention over all views costs

$$C_{\text{dense}} = O(V^2 P^2 D). \tag{5}$$

For our encoder the three branches contribute a per-view local term, a read and write term, and a global term,

$$C_{\text{ours}} = \underbrace{O(VP^2D)}_{\text{per-view local}} + \underbrace{O(VPRD)}_{\text{read+write}} + \underbrace{O(V^2R^2D)}_{\text{global}}. \tag{6}$$

The dominant term is linear in the view count, as $R \ll P$, and the only quadratic term is the global one, whose coefficient is $R^2$ rather than $P^2$, whereas joint attention is quadratic in the view count throughout. The global term contains no $P$, the read and write terms are linear in it, and only the per-view local term stays quadratic in $P$ as in any vision transformer, so what the design removes on the resolution axis is the $V^2$ multiplier rather than the quadratic itself. Equation (6) contains no term depending on how the cameras are arranged, because two views are related only through the register state and never compared against each other, so the mechanism has the same definition however far apart the cameras stand. Raising $R$ increases the read and write terms linearly and the global term quadratically from a small base, and since the write step is block-diagonal, $R$ also bounds how much of the fused scene a single view can read back. The alternation therefore separates three axes that joint attention ties together, the number of views, the resolution at which each is read, and the number of registers through which they communicate.

### 3.2 Geometry-texture interleaved Gaussian decoder

The decoder turns the encoder features into the parameters of the Gaussians, and every Gaussian carries geometric attributes that place it in space and appearance attributes that give it colour. Most methods predict both kinds along a single path and separate them only at the output, through different linear heads. The only supervision is the photometric loss between the rendered and the ground-truth images, so a geometry that departs from the truth leaves a residual that appearance can absorb, which lowers the loss while the geometric error is never corrected. Predicting the two in separate branches relieves this (Itkin et al., 2026; Li et al., 2026c), but a branch does not remove the compensation on its own. Compensating through appearance stays the shorter route to the photometric objective for as long as the appearance path has the capacity for it, so the quantity to limit is that capacity rather than the connection between the two. Our decoder accordingly interleaves geometry and texture blocks, where each texture block is guided by the current geometry state while geometry never reads texture, mitigating the entanglement between the two.

**Geometry and Texture Blocks.** Both token sets start from the same $N$ query tokens, since the model learns one set of queries, and diverge only because they pass through different blocks. One texture block follows every second of the $L_{\text{dec}}$ geometry blocks, so the texture path holds $L_{\text{dec}}/2$ blocks, deliberately too few to account for the rendering residual on its own. A texture block reads the geometry state current at that point before attending to the encoder features,

$$t \leftarrow \text{TextureBlock}\big(t + W_{\text{guide}}\, g,\ K,\ V\big), \tag{7}$$

where $g$ and $t$ are the geometry and texture tokens and $K$ and $V$ are the keys and values given by the encoder features. The opacity is assigned to the geometry blocks, since it decides whether a primitive exists at all and is the mechanism by which competing depth hypotheses are resolved, and placing it in the texture blocks would amount to letting appearance adjudicate geometry. Concurrent work makes the same separation but assigns opacity to the appearance branch (Li et al., 2026c), which is coherent when the aim is to give appearance capacity and contrary to it when the aim is to withhold it.

**Gaussian Adapter.** The adapter applies one linear map to the geometry tokens and another to the texture tokens, each emitting its attributes for all $G$ Gaussians of a query token at once. Each Gaussian carries a position, an opacity, a scale and a rotation from the geometry map and a colour from the texture map. Positions are expressed in world coordinates through a symmetric exponential map $\text{sign}(x)\,(e^{|x|} - 1)$ with a constant offset on the depth axis. Scales use a clamped exponential with an

upper bound, rotations are normalised quaternions, and opacity is a shifted sigmoid. Colour is predicted as degree-one spherical harmonics and passed to the rasteriser without a bounding activation. The resulting set is rendered by a differentiable Gaussian rasteriser at the target viewpoints.

Appearance is left with too little capacity to absorb a geometric residual. It is carried by $L_{\mathrm{dec}}/2$ blocks and one linear map, and it is confined to colour because opacity is decided in the geometry blocks, so the shortest route from a photometric residual to a lower loss runs through the geometry that produced it. The number of Gaussian primitives is set by $N$ and $G$ alone, since the adapter reads the query tokens and nothing else, so it is the same whatever the view count and the resolution of the input.

### 3.3 Training and adaptation

Training proceeds in two stages. The first trains from scratch at a lightweight base scale, where the two mechanisms of the preceding sections, the fusion of views and the decoding of geometry and appearance from query tokens, are learned in a short schedule. The second continues from those weights and grows the model to larger scales. Both stages minimise the same objective, so growth is a continuation of one run rather than a second run.

**Base stage.** The parameters of the Gaussians are never observed, so the only training signal is the rendered image. At the base scale every parameter is trained from scratch, with no pretrained weights, and the input views are supervised alongside the target views, while validation and test are evaluated on target views only. The objective combines three terms,

$$\mathcal{L} = \mathcal{L}_{\mathrm{rgb}} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}} + \lambda_{\mathrm{v}}\mathcal{L}_{\mathrm{v}}, \qquad \mathcal{L}_{\mathrm{rgb}} = \big\|\hat{I} - I\big\|_2^2, \qquad \mathcal{L}_{\mathrm{s}} = \tfrac{1}{2}\big(1 - \mathrm{SSIM}(\hat{I}, I)\big), \tag{8}$$

$$\mathcal{L}_{\mathrm{v}} = \frac{1}{NG}\sum_i \min\big(\min_v d_i^v,\, \tau\big), \qquad d_i^v = \sum_{c\in\{x,y\}} \mathrm{ReLU}\big(|\hat{u}_{i,c}^v| - 1\big), \tag{9}$$

where $\hat{I}$ and $I$ are the rendered and the ground-truth image and $\hat{u}_i^v$ is the projection of primitive $i$ onto the normalised image plane of view $v$, clamped at $\tau$. We adopt the visibility regulariser of Ren et al. (2026) unchanged: it takes its minimum over views, so a primitive is penalised only when it falls outside every rendered image. No depth or other geometric supervision is used at any point.

**Adaptation stage.** The second stage enlarges the model in short fine-tuning steps. It first adds query tokens, initialised by repeating trained ones with a small perturbation as in Ren et al. (2026), which raises the number of Gaussians and thus yields a finer scene representation. It then raises the input resolution, which adds no parameters since no weight depends on it.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets and Protocol.** We train and evaluate on DL3DV-10K (Ling et al., 2024), whose long camera trajectories make it possible to vary both the number of input views and the viewpoint span within a single scene. Our evaluation sweeps the viewpoint span from a 40-frame span to the full trajectory, and the number of input views from 8 to 96. We report the PSNR, SSIM and LPIPS metrics computed on the held-out target views.

**Baseline Methods.** We compare against two state-of-the-art methods, TokenGS (Ren et al., 2026), retrained under our settings, and QuerySplat (Li et al., 2026c), built on a pretrained geometry model and evaluated from its released weights. Appendix C further compares RemixGS, trained on DL3DV-10K alone, with state-of-the-art methods trained on the RealEstate10K dataset (Zhou et al., 2018).

**Implementation Details.** Our RemixGS sets $D = 1024$, patch size 16, $L_{\mathrm{enc}} = 6$ with a single shared view-register mixer, $R = 16$ registers per view, $L_{\mathrm{dec}} = 8$, $N = 4096$ query tokens and $G = 64$ Gaussians per token, for 262,144 primitives. Colour is degree-one spherical harmonics. The base stage trains for 150K steps at $256^2$ with 24 input and 4 target views, AdamW, a one-cycle schedule with peak learning rate $4 \times 10^{-4}$, and `bf16` mixed precision; the adaptation stage grows $N$ from 1024 to 4096 and raises the resolution to $448 \times 256$ or $512 \times 512$.

Table 1: Comparison on DL3DV-10K over three viewpoint spans. Rows marked $^{512}$ are evaluated at $512 \times 512$ resolution, the others at $448 \times 256$.

| | | 40 frames | | | 180 frames | | | Full trajectory | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Method | $V$=8 | 16 | 24 | 16 | 32 | 64 | 32 | 64 | 96 |
| PSNR↑ | TokenGS (Ren et al., 2026) | 22.45 | 23.11 | 23.47 | 18.95 | 19.50 | 19.65 | 18.44 | 18.83 | 18.83 |
| | **RemixGS (ours)** | **24.42** | **25.21** | **25.68** | **21.07** | **21.57** | **21.83** | **20.37** | **20.77** | **20.81** |
| | QuerySplat$^{512}$ (Li et al., 2026c) | 23.32 | 23.43 | 23.55 | **20.80** | 20.79 | 20.69 | **20.01** | 20.09 | 20.00 |
| | **RemixGS$^{512}$ (ours)** | **23.81** | **24.61** | **25.02** | 20.53 | **21.03** | **21.20** | 19.84 | **20.23** | **20.34** |
| SSIM↑ | TokenGS (Ren et al., 2026) | 0.683 | 0.721 | 0.733 | 0.540 | 0.564 | 0.568 | 0.516 | 0.531 | 0.532 |
| | **RemixGS (ours)** | **0.771** | **0.801** | **0.810** | **0.651** | **0.671** | **0.677** | **0.621** | **0.636** | **0.638** |
| | QuerySplat$^{512}$ (Li et al., 2026c) | 0.723 | 0.729 | 0.729 | 0.641 | 0.641 | 0.634 | 0.614 | 0.613 | 0.612 |
| | **RemixGS$^{512}$ (ours)** | **0.749** | **0.774** | **0.783** | **0.644** | **0.661** | **0.662** | **0.622** | **0.630** | **0.635** |
| LPIPS↓ | TokenGS (Ren et al., 2026) | 0.384 | 0.357 | 0.348 | 0.520 | 0.502 | 0.497 | 0.548 | 0.533 | 0.531 |
| | **RemixGS (ours)** | **0.303** | **0.282** | **0.272** | **0.421** | **0.412** | **0.405** | **0.451** | **0.442** | **0.440** |
| | QuerySplat$^{512}$ (Li et al., 2026c) | **0.311** | **0.307** | **0.308** | **0.411** | **0.411** | **0.416** | **0.440** | **0.440** | **0.440** |
| | **RemixGS$^{512}$ (ours)** | 0.353 | 0.333 | 0.326 | 0.472 | 0.462 | 0.459 | 0.500 | 0.493 | 0.489 |

### 4.2 Experimental Results

**Quantitative Comparison.** We evaluate each model on every setting in Table 1; within a setting and a resolution every method sees the same scenes and the same target views. QuerySplat builds on VGGT-Ω (Wang et al., 2026a) and degrades sharply under an aspect-ratio mismatch; for a fair comparison, Table 1 compares QuerySplat and RemixGS at $512 \times 512$, the resolution QuerySplat is trained at.

With the viewpoint span fixed and input views added, the methods change by very different amounts. With $448 \times 256$ input, from 8 to 24 views at the 40-frame span RemixGS gains 1.26 dB of PSNR and TokenGS 1.02 dB; from 16 to 64 views at the 180-frame span, 0.76 and 0.70 dB; from 32 to 96 views on the full trajectory, 0.44 and 0.39 dB. With $512 \times 512$ input, RemixGS$^{512}$ gains 1.21, 0.67 and 0.50 dB at the three spans, and QuerySplat$^{512}$ 0.23, $-0.11$ and $-0.01$ dB.

QuerySplat reads its geometry from the pretrained VGGT-Ω (Wang et al., 2026a), so how it scales with the view count is bound to that model; TokenGS concatenates the features of all views into one sequence under global self-attention, and as views are added the redundant comparisons in that sequence grow faster than the informative ones; RemixGS carries every cross-view interaction through the $R$ registers each view holds, so the cross-view state grows with the number of views.

With the number of input views fixed and the viewpoint span widened, the PSNR of every method falls. Taking 16 views as an example, from the 40- to the 180-frame span, RemixGS falls 4.14 dB and TokenGS 4.16 dB with $448 \times 256$ input, and RemixGS$^{512}$ 4.08 dB and QuerySplat$^{512}$ 2.63 dB with $512 \times 512$ input. Taking 64 views, from the 180-frame span to the full trajectory, RemixGS and TokenGS fall 1.06 and 0.82 dB, and RemixGS$^{512}$ and QuerySplat$^{512}$ 0.97 and 0.60 dB.

With $448 \times 256$ input, RemixGS leads TokenGS in every setting, by 1.93 to 2.21 dB of PSNR, 0.077 to 0.111 of SSIM and 0.075 to 0.099 of LPIPS, with the same 262,144 primitives. With $512 \times 512$ input, RemixGS$^{512}$ differs from QuerySplat$^{512}$ in PSNR by $+0.49$ to $+1.47$ dB at the 40-frame span, $-0.27$ to $+0.51$ dB at the 180-frame span and $-0.17$ to $+0.34$ dB on the full trajectory, and trails it on LPIPS in every setting by 0.018 to 0.061, with half of QuerySplat's 524,288 primitives. RemixGS is trained without a perceptual loss, whereas the released QuerySplat model adds an LPIPS term at weight 0.05. Along the view count the difference between the methods comes from the fusion mechanism.

**Qualitative Comparison.** Figures 2 and 3 show four target views of two scenes each, from the 180-frame setting with 32 input views. At $448 \times 256$, TokenGS smears the slats of the canopy in Scene 1 into streaks and doubles its curved edge, whereas RemixGS keeps the slats apart and the window frames on the stone-clad wall regular. In Scene 2, TokenGS reduces the open wardrobe and




Figure 2: Qualitative comparison on DL3DV-10K at $448 \times 256$: two scenes from the evaluation set, four target views each. The top row is the ground truth reference.




Figure 3: Qualitative comparison on DL3DV-10K at $512 \times 512$: two scenes from the evaluation set, four target views each. The top row is the ground truth reference.

the kitchenette by the door to blur, and RemixGS retains both. At $512 \times 512$, QuerySplat blurs the steps before the entrance of Scene 1 until their edges can no longer be told apart and smears most of the goods on the shelves of Scene 2, whereas RemixGS keeps the edges of the steps distinct and most of the goods sharp. Overall, RemixGS reconstructs regions of complex geometry and texture more faithfully than both baselines. More qualitative results are provided in Appendix D.

### 4.3 Ablation and Analysis

**Efficiency Analysis.** Figure 4 reports peak memory, the latency of a single forward pass, total compute and the share of it spent on attention at $448 \times 256$ resolution, over the input view counts of Table 1. TokenGS concatenates the features of every view into one sequence and attends over it jointly, and QuerySplat reads its geometry from a pretrained geometry model. Our encoder relates two views only through the registers each view holds, so the dominant term of Equation (6) is linear in the view count, and the rightmost panel gives its share of total compute. Neither the parameter count nor the primitive count moves with the view count, 222 M parameters and 262,144 primitives for TokenGS, 1118 M and 524,288 for QuerySplat, and 279 M and 262,144 for RemixGS. QuerySplat's count excludes that pretrained geometry model. At an equal view count and an equal number of training steps, training takes 63.6 hours against 97.3 for TokenGS, both on $4\times$A800 GPUs. QuerySplat's published base stage of 300K iterations at a global batch of 64 is about 15 times our training compute, at least 300 hours on the same $4\times$A800 GPUs.

**Register Ablation.** In Section 3.1, each view exchanges information with the others only through its $R$ registers, and since the write step is block-diagonal, $R$ also bounds how much of the fused scene a single view reads back. Table 2 varies $R$ and reports PSNR at several input view counts. Raising $R$ to 32 or 64 changes PSNR by at most 0.10 dB in any setting, so we keep $R$=16.

**Query Token Ablation.** Changing the number of query tokens is the one move that adds parameters, and the added tokens start as copies of the trained ones under a small perturbation. Taking $N$ from 1024 to 4096 raises PSNR at every setting of Table 3 by 0.30 to 0.92 dB, and 8192 adds 0.23 to 0.35 dB more, both gains growing with the input view count. We keep $N$=4096 so that RemixGS matches TokenGS in primitive count in Table 1.

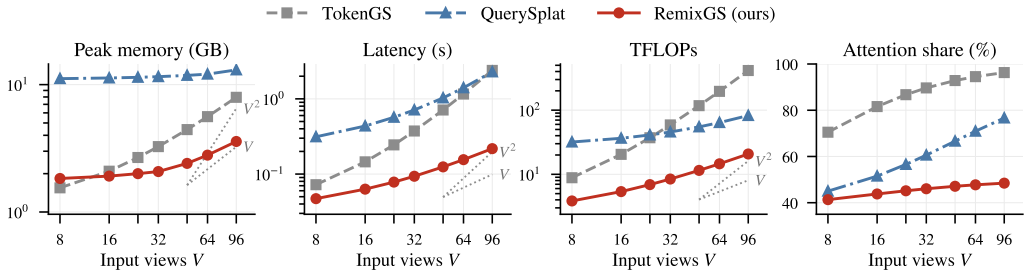


Figure 4: Efficiency at $448 \times 256$ resolution over the input view counts of Table 1. FLOPs and the attention share are counted from the model definitions; attention counts the $QK^\top$ and $AV$ products of Equation (6) alone. Memory and latency are a single forward pass on one GPU. QuerySplat is counted without VGGT-Ω (Wang et al., 2026a), the geometry model it builds on, while its memory and latency include it.

Table 2: Register ablation on DL3DV-10K at $256 \times 256$, in PSNR (dB).

| | 40 frames | | 180 frames | | Full | |
|---|---|---|---|---|---|---|
| $R$ | $V$=8 | 24 | 16 | 64 | 32 | 96 |
| 16 | 23.59 | 25.89 | 20.29 | 21.40 | 19.71 | 20.33 |
| 32 | 23.68 | 25.98 | 20.37 | 21.49 | 19.78 | 20.40 |
| 64 | 23.64 | 25.99 | 20.35 | 21.49 | 19.79 | 20.40 |

Table 3: Query-token ablation on DL3DV-10K at $256 \times 256$, in PSNR (dB).

| | 40 frames | | 180 frames | | Full | |
|---|---|---|---|---|---|---|
| $N$ | $V$=8 | 24 | 16 | 64 | 32 | 96 |
| 1024 | 23.16 | 24.97 | 19.99 | 20.82 | 19.38 | 19.89 |
| 4096 | 23.59 | 25.89 | 20.29 | 21.40 | 19.71 | 20.33 |
| 8192 | 23.82 | 26.24 | 20.54 | 21.70 | 19.95 | 20.60 |

**Interleaved Decoder Ablation.** Table 4 compares the interleaved decoder with a non-interleaved one, in which a single path of geometry blocks predicts every attribute. With 24 input views interleaving raises PSNR by 0.99 to 1.63 dB, and with 8 input views by 1.07 dB on the 40-frame span, while on the 180-frame span and the full trajectory the two models are within 0.1 dB. The gain grows with the number of input views, as more views constrain the geometry that guides the texture blocks.

Table 4: Interleaved decoder ablation on DL3DV-10K at $256 \times 256$, in PSNR (dB).

| | 40 frames | | 180 frames | | Full | |
|---|---|---|---|---|---|---|
| Interleaved | $V$=8 | 24 | 8 | 24 | 8 | 24 |
| × | 22.34 | 23.51 | 17.61 | 19.43 | 16.48 | 18.31 |
| ✓ | 23.41 | 25.14 | 17.59 | 20.60 | 16.39 | 19.30 |

## 5 Conclusion

We present RemixGS, a feed-forward 3D Gaussian reconstruction model with two components. Our alternating multi-view encoder passes all cross-view information through a few registers held by each view, so dense attention over the patch tokens is never formed. Our geometry-texture interleaved decoder lets geometry guide texture but not vice versa, turning a fixed set of query tokens into a constant number of Gaussian primitives while mitigating the entanglement of geometry and appearance. With this design, the cost of the model grows close to linearly in the number of views, and only its per-view term stays quadratic in the resolution, as in any vision transformer. Experiments show that our RemixGS consistently surpasses state-of-the-art methods in reconstruction quality as the input scale increases, with lower training and inference cost. Since the number of query tokens is set by the model rather than by the input, it remains an open axis of growth: raising it from 4096 to 8192 still improves PSNR by 0.23 to 0.35 dB.

**Limitations.** These query tokens carry no spatial locality: nothing ties one to a region of the scene, so added tokens densify the scene uniformly rather than where detail is needed. Giving the query set spatial structure is therefore our next step. The scale the architecture is built for also remains to be reached, limited so far by available data and compute budget rather than by its own cost. RemixGS also relies on camera poses from COLMAP or a geometry model.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods, translation, and interpreting results. We have not used generative AI tools to propose or refine hypotheses, to design or provide feedback on the research methodology or experiments, or to clean or reformat datasets, and the remaining tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools to help draft and edit parts of the paper, prepare figures, identify and summarise related literature, and format references. The authors directed and verified all AI-assisted work: we reviewed and tested all AI-written code, checked every reported number against the raw evaluation outputs, and verified every reference against its published version. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work uses only the publicly available DL3DV-10K and RealEstate10K datasets under their terms of use, and involves no human subjects and no new data collection. Both consist of real-world videos, and RealEstate10K is drawn from public online videos of property interiors. Like any 3D reconstruction method, ours can reconstruct private spaces from captured images, and it should be applied only with the consent of those who own or appear in the captured scenes.

### REPRODUCIBILITY STATEMENT

Section 4.1 gives the datasets, the evaluation protocol, and the architecture and training hyperparameters of RemixGS, and Section 3.3 describes its two training stages. The appendix gives the source of the camera poses and the full protocol of the zero-shot RealEstate10K evaluation, including the scenes available to us. We will release the code and model weights upon acceptance.

## 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (ECCV)*, 2024.

In Cho, Jeonghwan Cho, Mijin Yoo, Gim Hee Lee, and Seon Joo Kim. ATSplat: Compact feed-forward 3D Gaussian splatting with adaptive token expansion. *arXiv preprint arXiv:2607.20417*, 2026.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.

David Huang, Guile Wu, Chengjie Huang, Bingbing Liu, and Dongfeng Bai. TurboVGGT: Fast visual geometry reconstruction with adaptive alternating attention. *arXiv preprint arXiv:2605.14315*, 2026.

Roni Itkin, Noam Issachar, Yehonatan Keypur, Xingyu Chen, Anpei Chen, and Sagie Benaim. GlobalSplat: Efficient feed-forward 3D Gaussian splatting via global scene tokens. In *European Conference on Computer Vision (ECCV)*, 2026.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A self-supervised large view synthesis model. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025b.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *International Conference on Learning Representations (ICLR)*, 2025.

Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. iLRM: An iterative large 3D reconstruction model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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):1–14, 2023.

Wenyu Li, Sidun Liu, Tongrui Hu, Peng Qiao, and Yong Dou. LocusGS: Spatially grounded tokens for feed-forward 3D Gaussian splatting. *arXiv preprint arXiv:2608.12825*, 2026a.

Yihui Li, Chengxin Lv, Zichen Tang, Hongyu Yang, and Di Huang. TokenSplat: Token-aligned 3D Gaussian splatting for feed-forward pose-free reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Yinglong Li, Donghui Shen, Xiaoyu Zhang, Zhichao Ye, Hongyu Wu, Aimin Hao, Guofeng Zhang, and Haomin Liu. QuerySplat: Decoupling geometry and appearance representations in 3DGS prediction. *arXiv preprint arXiv:2608.01186*, 2026c.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Chensheng Peng, Quentin Herau, Jiezhi Yang, Yichen Xie, Yihan Hu, Wenzhao Zheng, Matthew Strong, Masayoshi Tomizuka, and Wei Zhan. UniQueR: Unified query-based feedforward 3D reconstruction. In *European Conference on Computer Vision (ECCV)*, 2026.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. TokenGS: Decoupling 3D Gaussian prediction from pixels with learnable tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Fast visual geometry transformer. In *International Conference on Learning Representations (ICLR)*, 2026.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot Gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Alexander Veicht, Sunghwan Hong, Dániel Baráth, and Marc Pollefeys. ZipSplat: Fewer Gaussians, better splats. *arXiv preprint arXiv:2606.05102*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-$\omega$. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Weijie Wang, Donny Y. Chen, Zeyu Zhang, Duochao Shi, Akide Liu, and Bohan Zhuang. ZPressor: Bottleneck-aware compression for scalable feed-forward 3DGS. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations (ICLR)*, 2026b.

Zipeng Wang and Dan Xu. FlashVGGT: Efficient and scalable visual geometry transformers with compressed descriptor attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting Gaussian splatting and depth. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3D Gaussian splats from sparse unposed images. In *International Conference on Learning Representations (ICLR)*, 2025.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

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 (ECCV)*, 2024.

Xiaoxue Zhang, Xiaoxu Zheng, Yixuan Yin, Tiao Zhao, Kaihua Tang, Michael Bi Mi, Zhan Xu, and Dave Zhenyu Chen. AnchorSplat: Feed-forward 3D Gaussian splatting with 3D geometric priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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):1–12, 2018.

Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-LRM: Long-sequence large reconstruction model for wide-coverage Gaussian splats. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chen Ziwen, Hao Tan, Peng Wang, Zexiang Xu, and Li Fuxin. Long-LRM++: Preserving fine details in feed-forward wide-coverage reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026.

# APPENDIX

## A CAMERA POSE DEPENDENCE AND DECOUPLING

RemixGS reads camera poses only as the Plücker ray maps at the input of its encoder, so it places no requirement on where the poses come from. They can be the ground-truth poses a dataset provides, which are themselves estimated offline by structure-from-motion such as COLMAP (Schönberger & Frahm, 2016), or they can be predicted by a feed-forward geometry model such as VGGT (Wang et al., 2025a), $\pi^3$ (Wang et al., 2026b), Depth Anything 3 (Lin et al., 2026) or VGGT-$\Omega$ (Wang et al., 2026a), kept frozen during both training and inference. Only the poses are taken from the geometry model, so replacing it with another changes the input RemixGS receives and leaves its architecture unchanged. Methods built on the intermediate features of a geometry model, such as QuerySplat (Li et al., 2026c), C3G (An et al., 2026) and AnySplat (Jiang et al., 2025b), tie their architecture to that model instead, and moving to another requires modifying the network and retraining it from scratch. We keep RemixGS decoupled from any particular geometry model, and a version trained to accept the poses of several such models is left for future work.

## B SHARING THE VIEW-REGISTER MIXER ACROSS ENCODER LAYERS

The encoder of Section 3.1 stacks $L_{\mathrm{enc}}$ layers, each a per-view block followed by a view-register mixer, and the mixer admits two forms. In the shared form, one set of mixer parameters serves all $L_{\mathrm{enc}}$ layers and the register state is carried from one layer to the next, so the registers accumulate the cross-view state over the depth of the encoder. In the independent form, each layer has a mixer of its own. The two forms perform the same read, global and write steps and therefore the same computation; they differ only in parameters, which the independent form multiplies by $L_{\mathrm{enc}}$. In our experiments the independent form did not improve on the shared one, and the shared form reached a higher training PSNR. That one mixer serves every layer as well as separate ones suggests that multi-view fusion follows a nearly fixed pattern across depth: each layer applies the same exchange between views to a register state that the per-view blocks keep refining, rather than a fusion specific to that layer. The extra parameters thus bring no benefit in our setting, and RemixGS uses a single shared view-register mixer, so the parameter cost of the cross-view exchange does not grow with the depth of the encoder.

## C ZERO-SHOT EVALUATION ON REALESTATE10K

RealEstate10K consists of walkthrough videos of houses collected from YouTube, with camera poses recovered by SLAM and bundle adjustment. Its scenes and camera trajectories differ from those of DL3DV-10K, so the evaluation measures how RemixGS transfers to a dataset it has never seen. We evaluate the $256 \times 256$ RemixGS of the ablations, trained on DL3DV-10K alone, on the RealEstate10K test set (Zhou et al., 2018) without any training or fine-tuning on it. We follow the protocol of C3G (An et al., 2026) and GlobalSplat (Itkin et al., 2026): the input views are resized to $256 \times 256$ with Lanczos filtering, the ground truth is resized and centre-cropped to $256 \times 256$, the model renders at $256 \times 256$, and the metrics are averaged within each scene and then across scenes. The context views are sampled uniformly between the two endpoints of each evaluation index, and the target views are taken from the index unchanged. Our copy of the test split holds 5,345 of the 5,601 indexed scenes, so the scene set is not identical to that of the other methods. RemixGS was trained with bilinear downsampling, whereas this protocol uses Lanczos, which also affects the result. Table 5 lists methods trained on RealEstate10K, as reported by GlobalSplat.

With 12, 24 and 36 input views, RemixGS reaches 27.91, 28.40 and 28.51 dB of PSNR, and its PSNR and SSIM rise with the view count, whereas those of LVSM and GlobalSplat-32K fall and those of ZPressor stay nearly flat. In PSNR it trails only GlobalSplat-32K, ZPressor-6 and, at 12 views, LVSM, and at 36 views it is level with ZPressor-6, although all of them are trained on RealEstate10K. In LPIPS it trails the strongest of these methods; RemixGS is trained without a perceptual loss.

Table 5: Zero-shot evaluation on RealEstate10K at $256 \times 256$. Our RemixGS is trained on DL3DV-10K only; the other methods are trained on RealEstate10K, and their numbers are taken from GlobalSplat. LVSM is not Gaussian-based.

| | 12 views | | | 24 views | | | 36 views | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| LVSM (Jin et al., 2025) | 28.65 | 0.898 | 0.095 | 27.24 | 0.874 | 0.112 | 26.38 | 0.855 | 0.126 |
| NoPoSplat (Ye et al., 2025) | 21.26 | 0.667 | 0.200 | 21.24 | 0.664 | 0.200 | 21.19 | 0.663 | 0.200 |
| AnySplat (Jiang et al., 2025b) | 23.06 | 0.807 | 0.215 | 24.11 | 0.838 | 0.198 | 24.20 | 0.842 | 0.192 |
| DepthSplat (Xu et al., 2025) | 21.35 | 0.809 | 0.190 | 19.66 | 0.743 | 0.239 | 18.84 | 0.704 | 0.268 |
| ZPressor-6 (Wang et al., 2025b) | 28.46 | 0.910 | 0.098 | 28.51 | 0.911 | 0.097 | 28.50 | 0.911 | 0.097 |
| ZPressor-3 (Wang et al., 2025b) | 23.63 | 0.846 | 0.157 | 23.65 | 0.846 | 0.157 | 23.65 | 0.846 | 0.157 |
| C3G (An et al., 2026) | 23.61 | 0.740 | 0.203 | 23.80 | 0.747 | 0.198 | 23.81 | 0.747 | 0.199 |
| GlobalSplat-32K (Itkin et al., 2026) | 29.54 | 0.903 | 0.121 | 29.48 | 0.901 | 0.122 | 29.39 | 0.899 | 0.126 |
| **RemixGS (ours)** | 27.91 | 0.885 | 0.170 | 28.40 | 0.895 | 0.160 | 28.51 | 0.897 | 0.157 |

## D Additional Qualitative Results

Figures 5 to 8 show further scenes in the setting of Figures 2 and 3, the 180-frame span with 32 input views, and none of these scenes appears in those figures. Figures 5 and 6 compare RemixGS with TokenGS on six scenes at $448 \times 256$, and Figures 7 and 8 compare RemixGS$^{512}$ with QuerySplat$^{512}$ on four scenes at $512 \times 512$. Each scene shows four target views, with the ground truth in the top row.

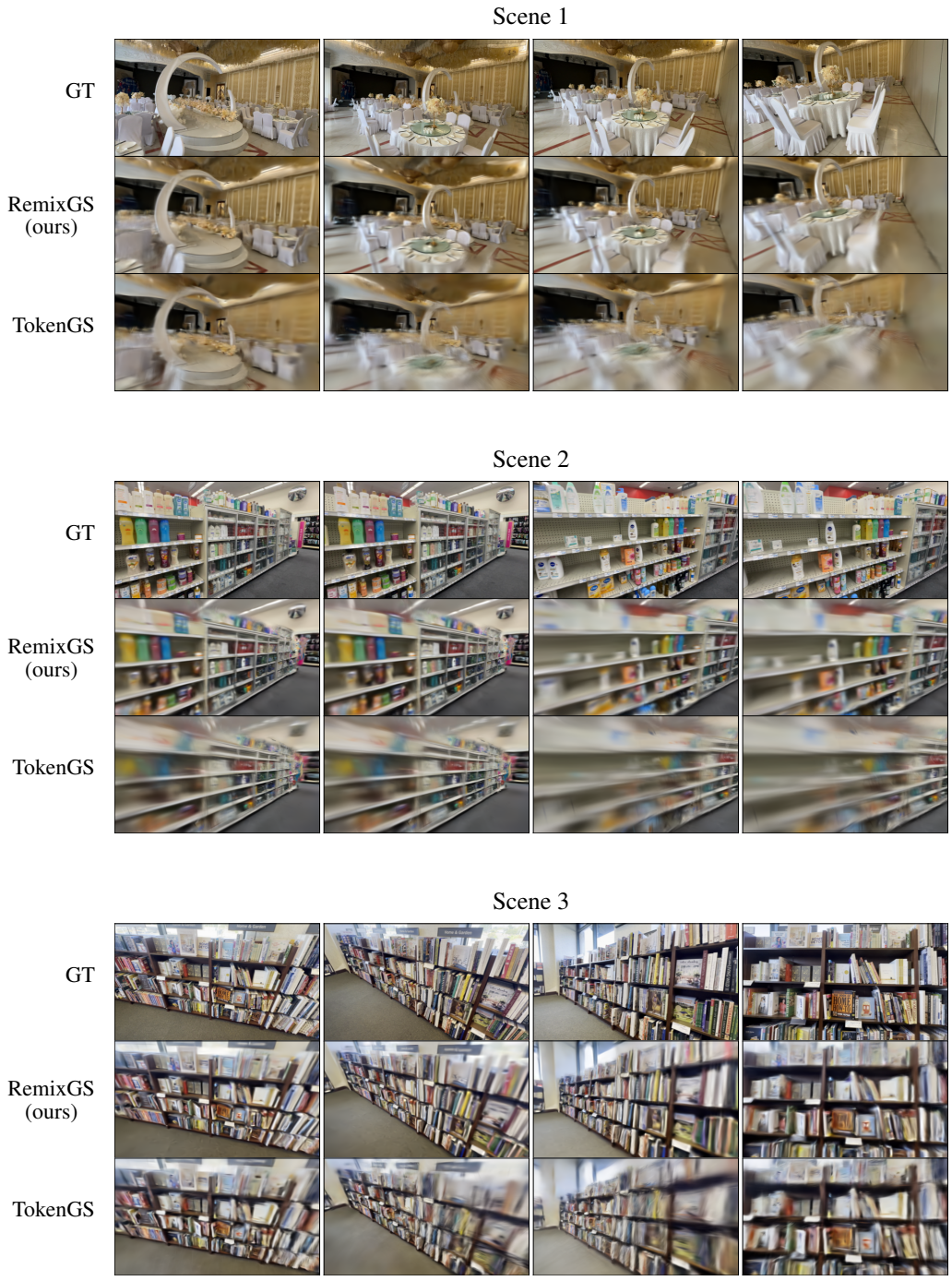


Figure 5: Additional qualitative comparison on DL3DV-10K at $448 \times 256$, four target views per scene. The top row of each scene is the ground truth reference.

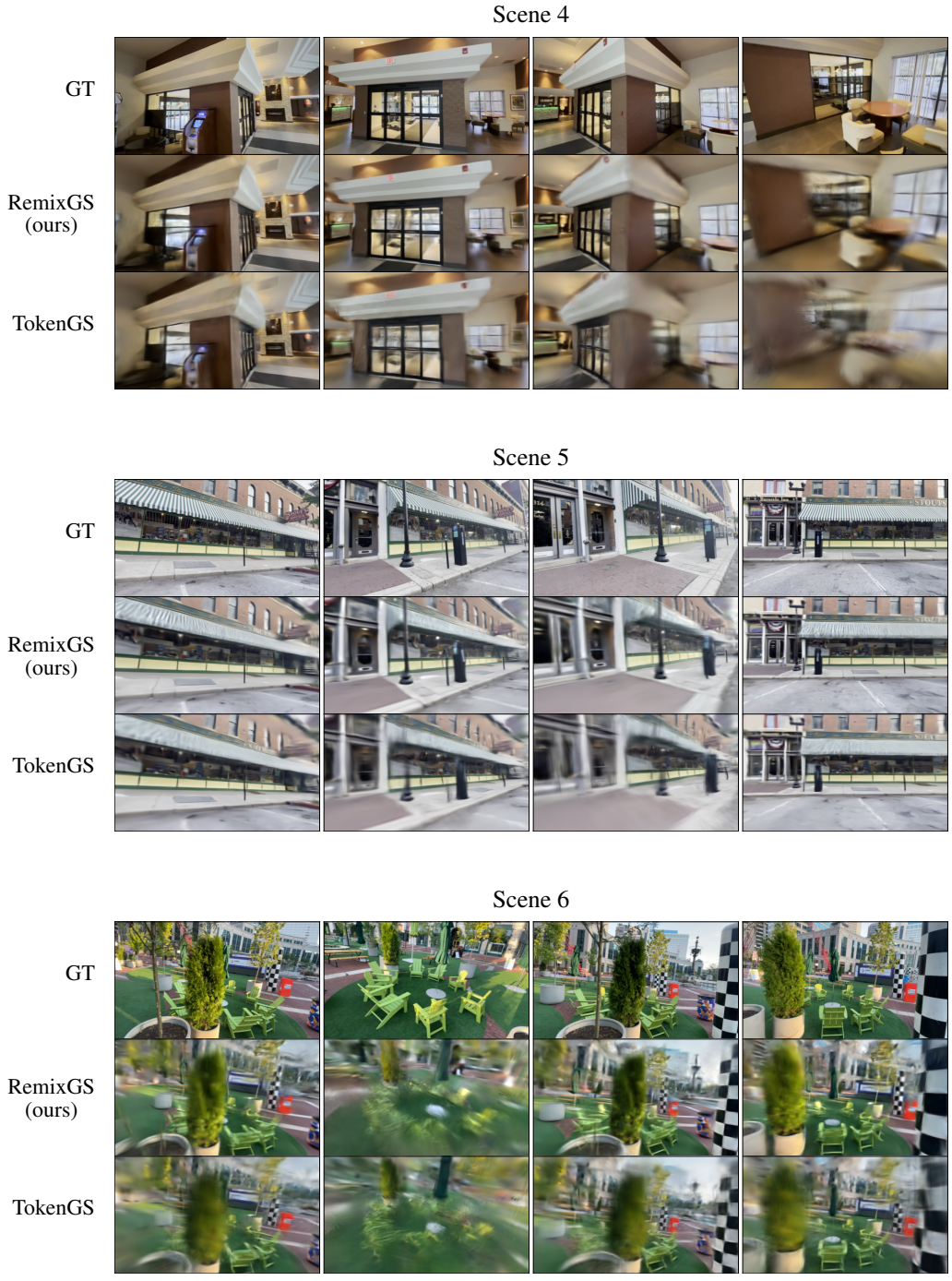


Figure 6: Additional qualitative comparison on DL3DV-10K at $448 \times 256$, four target views per scene. The top row of each scene is the ground truth reference.

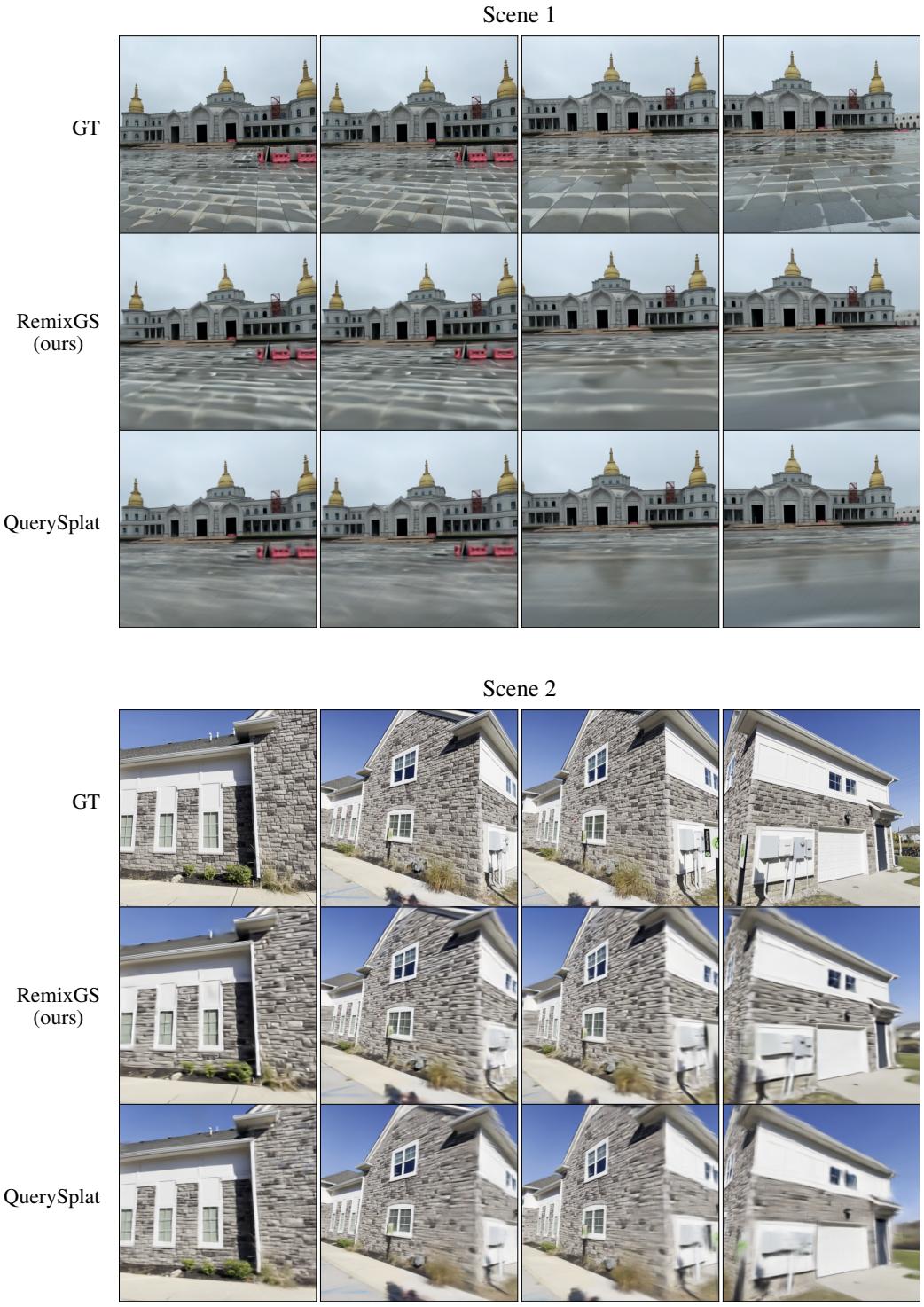


Figure 7: Additional qualitative comparison on DL3DV-10K at $512 \times 512$, four target views per scene. The top row of each scene is the ground truth reference.

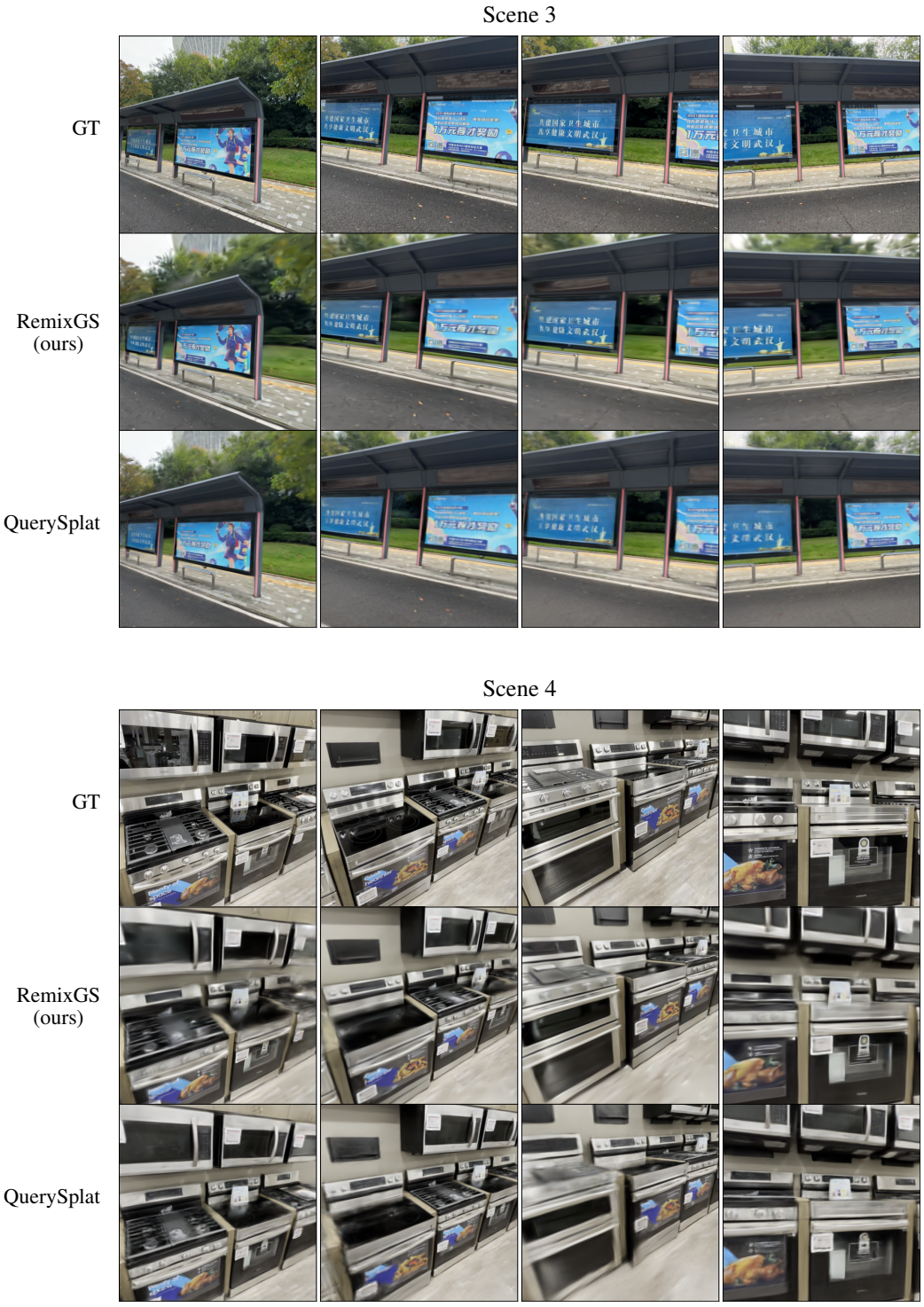


Figure 8: Additional qualitative comparison on DL3DV-10K at $512 \times 512$, four target views per scene. The top row of each scene is the ground truth reference.