LOOK WHERE YOU SPLAT: SPLAT GUIDED FEED-FORWARD 3D GAUSSIAN PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward 3D Gaussian Splatting has recently moved from predicting a Gaussian per pixel to predicting Gaussians from a fixed set of learnable tokens, so the number of Gaussians no longer scales with the input images. These token-based feed-forward models reconstruct compact scenes with as few as 2K Gaussians, yet we find that their tokens do not always attend to where their own Gaussians project. We observe that for 31.2% of the tokens, the cross-attention weights do not align with their splat regions, where their predicted Gaussians project in the input images. A misaligned token aggregates image patch features from outside its splat region, leaves part of the region unattended, or both. We hypothesize that a token predicts its Gaussians more accurately when it attends to image patches within its own splat region. To exploit this insight, we propose Gaussian Splat Attention Bias (GSAB), which aligns each token's attention weights to its own splat region in the input images. Because GSAB only adds an attention bias and an intermediate decoding of existing tokens, it applies to any token-based model, which we verify on two that differ in pose assumption and scale: a pose-free model predicting 2K Gaussians and a posed model predicting 262K. GSAB improves PSNR by up to 0.80 dB on the posed model and by 0.71 dB on the pose-free model.

1 INTRODUCTION

Feed-forward 3D Gaussian Splatting enables fast scene reconstruction and novel view synthesis by predicting 3D Gaussians (Kerbl et al., 2023) from input images in a single forward pass (Charatan et al., 2024; Zhang et al., 2024). Recent methods replace pixel-aligned prediction with a fixed set of learnable Gaussian tokens, which decouples the number of Gaussians from the input images and yields compact scene representations (An et al., 2026; Ren et al., 2026). C3G, for instance, reconstructs 2K Gaussians from two input views, whereas pixel-aligned methods use 131K (Table 1).

Each token cross-attends to image patches, aggregating their features as visual evidence for predicting one or more Gaussians. To reconstruct a particular part of the scene accurately, the token should gather image features that describe that part. These image regions can be identified by projecting the token's predicted Gaussians into each input view, yielding their 2D splat regions. This motivates aligning the token's cross-attention weights with these splat regions. This intuition is supported by query-based detectors, which similarly aggregate image features through queries and benefit from spatial priors that align attention with relevant image regions (Carion et al., 2020; Zhu et al., 2021; Gao et al., 2021; Ye et al., 2023).

However, our analysis reveals that 31.2% of tokens have cross-attention weights that do not align with their splat regions. These tokens aggregate features from outside their splat regions, leave parts of those regions unattended, or both. The former dilutes the visual evidence with content from elsewhere, and the latter leaves the reconstruction underdetermined. Based on this observation, we hypothesize that a Gaussian token predicts its Gaussians more accurately when it attends to image patches within its own splat region. A toy experiment supports this hypothesis (Sec. 3.2), showing that, with the Gaussian geometry, and hence each splat region, held fixed, concentrating attention on the patches inside the region yields more accurate color and opacity.

We therefore propose Gaussian Splat Attention Bias (GSAB), which encourages each token to attend more strongly to its own splat region in the input images. To construct the attention bias, we decode each token into Gaussians at an intermediate block, before the cross-attention of the final block.

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.

GSAB projects these Gaussians into the input views to obtain their splat regions, which determine the attention bias in the cross-attention of the final block. Because GSAB obtains its Gaussians from an intermediate block of the same forward pass and acts only on the attention weights of the existing cross-attention, it can be added to any token-based feed-forward Gaussian prediction model whose tokens cross-attend to input image patches, without an extra pass through its transformer blocks.

We apply GSAB to two token-based models that differ in pose assumption and scale, the pose-free C3G with 2K Gaussians and the posed TokenGS with 262K. On C3G, GSAB improves PSNR over the released checkpoint by 0.71 dB and also improves SSIM and LPIPS (Table 1). On TokenGS, it improves PSNR over our reproduction by up to 0.80 dB (Table 2).

Our contributions are summarized as follows:

- We provide a novel analysis of the relationship between cross-attention alignment and reconstruction accuracy in token-based feed-forward Gaussian prediction, showing that many Gaussian tokens attend outside their splat regions.
- To address this misalignment, we propose Gaussian Splat Attention Bias (GSAB), which aligns each Gaussian token's attention weights to its own splat region in the input images.
- We demonstrate that GSAB effectively improves reconstruction quality, achieving PSNR gains of 0.71 dB on C3G and up to 0.80 dB on TokenGS, showing its generalizability across different pose assumptions and numbers of predicted Gaussians.

2 Related Work

2.1 Feed-Forward 3D Gaussian Splatting

Pixel-based. Early feed-forward 3D Gaussian Splatting methods predict the parameters of pixel-aligned Gaussians from posed images (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025; Zhang et al., 2024). Subsequent work extends this task to joint Gaussian reconstruction and camera estimation (Ye et al., 2025; Huang & Mikolajczyk, 2025a; Jiang et al., 2025; Zhao et al., 2026), with recent 3D vision foundation models (Wang et al., 2024a; 2025). Methods that reduce the redundancy of pixel-aligned predictions (Wang et al., 2024b; Jiang et al., 2025; Moreau et al., 2026; Zhang et al., 2026a;b) still produce Gaussian counts that scale with the input images. This motivates token-based prediction with a fixed Gaussian budget independent of the input images.

Token-based. A fixed set of learnable scene tokens queries image features and each token is decoded into one or more Gaussians (An et al., 2026; Itkin et al., 2026; Ren et al., 2026). Related approaches explore token compression, cross-view fusion, and separate geometry and appearance decoding (Veicht et al., 2026; Li et al., 2026b;c). Interestingly, these token-based methods resemble the DETR (Carion et al., 2020) architecture for object detection, motivating attention guidance toward the Gaussian's splat region.

2.2 Prediction-Modulated Attention

Query-based detection and segmentation benefit from spatial priors that focus each query on relevant image regions (Zhu et al., 2021; Meng et al., 2021; Cheng et al., 2022). In particular, intermediate predictions provide spatial priors that guide feature aggregation, improving detection performance (Gao et al., 2021; Liu et al., 2022; Ye et al., 2023; Lin et al., 2023).

For token-based feed-forward 3D Gaussian prediction, the 2D projections of a token's predicted Gaussians (Kerbl et al., 2023) define its splat region in the input images. We observe that a token's cross-attention can be misaligned with its splat region. Motivated by prior prediction-modulated attention, GSAB aligns each token's attention weights to its own splat region.

Concurrent work LocusGS (Li et al., 2026a) also introduces attention biases based on the spatial extent associated with each anchor, which does not precisely reflect the predicted Gaussians' splat regions. In contrast, GSAB derives its attention bias directly from the 2D projections of the predicted Gaussians, targeting their splat regions more precisely than an anchor-based approximation.

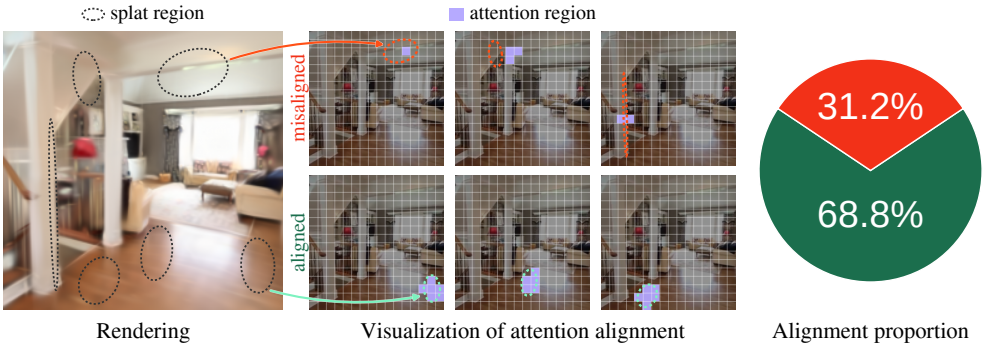


Figure 1: Gaussian tokens often do not attend to their own splat regions. Splat regions of Gaussians predicted by C3G on a RealEstate10K scene (left), cross-attention regions of misaligned and aligned Gaussian tokens over the patch tokens, shown with their splat regions (middle), and the proportion of aligned Gaussian tokens over the RealEstate10K test scenes, where 31.2% are misaligned (right).

3 METHOD

We first introduce the token-based feed-forward Gaussian prediction model (Sec. 3.1), then demonstrate where its tokens attend (Sec. 3.2), and finally present Gaussian Splat Attention Bias (Sec. 3.3).

3.1 PRELIMINARIES: TOKEN-BASED FEED-FORWARD 3D GAUSSIAN SPLATTING

In a token-based feed-forward model (An et al., 2026), a fixed set of learnable scene tokens aggregates image features and is decoded into 3D Gaussians. Given V input images $\{I_v\}_{v=1}^{V}$ are encoded into $M = V \cdot P$ patch tokens $F = \{\mathbf{f}_j\}_{j=1}^{M}$, with P tokens per image. The N learnable Gaussian tokens $T = \{\mathbf{t}_i\}_{i=1}^{N}$ are iteratively updated through L transformer blocks by attending to the image patch tokens and other Gaussian tokens.

The attention from Gaussian token $\mathbf{t}_i$ to patch token $\mathbf{f}_j$ starts from a query $\mathbf{q}_i = \mathbf{W}_Q\,\mathbf{t}_i$ and a key $\mathbf{k}_j = \mathbf{W}_K\,\mathbf{f}_j$, where $\mathbf{W}_Q$ and $\mathbf{W}_K$ are the transformer block's query and key projections. The scaled inner product $\mathbf{q}_i^\top \mathbf{k}_j/\sqrt{d}$ is the *attention logit* of that pair, and the block aggregates the patch tokens in proportion to the *attention weight* a_{ij} that a softmax over the patch tokens obtains from those logits:

$$a_{ij} = \underset{j}{\mathrm{softmax}}\left(\frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}}\right). \tag{1}$$

The Gaussian Head h decodes each token into a Gaussian,

$$G_i = h(\mathbf{t}_i)\,, \tag{2}$$

so that N tokens yield N Gaussians, or $N \cdot K$ Gaussians when the head predicts K per token. Each Gaussian G_i consists of a mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, a covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, an opacity o_i and a color $\mathbf{c}_i$. The Gaussian splat of G_i on a given view's image plane is defined by the 2D mean $\boldsymbol{\mu}_i^{\mathrm{2D}}$ and the 2D covariance $\boldsymbol{\Sigma}_i^{\mathrm{2D}}$, which are computed from the view's camera parameters.

3.2 DO TOKENS LOOK WHERE THEY SPLAT?

Each Gaussian token predicts its output from the image patches it attends to. For example, a token predicting the color of a Gaussian on a wall may miss relevant pixels if its attention covers only part of the splat, or mix in an adjacent window if it attends too broadly. We therefore ask whether a token attends to its predicted Gaussian's splat region and whether this alignment matters for reconstruction.

Misaligned attention. We first ask whether a Gaussian token's prediction and observation align, or, more specifically, whether the token aggregates its visual features from the image region that its predicted Gaussian covers. To answer this, we define two regions over the patch tokens of an input

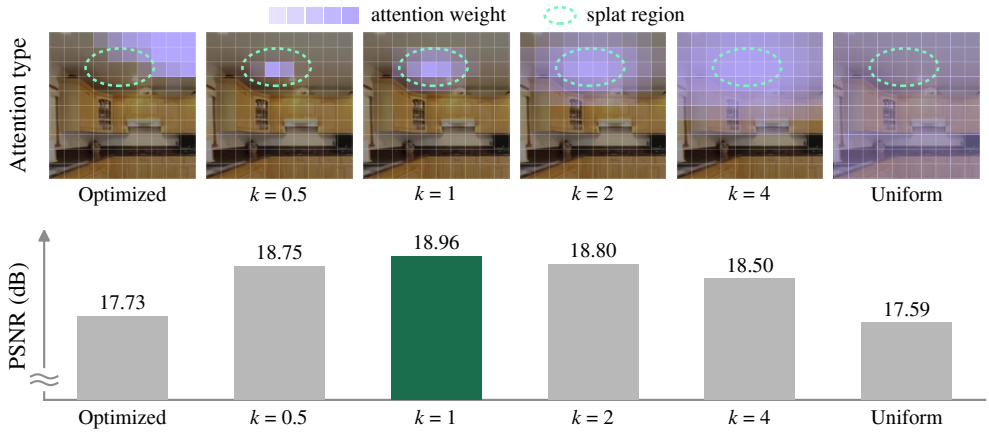


Figure 2: Toy experiment. With the Gaussian geometry fixed, the attention is set to the splat region scaled by k, set *uniform* over all patch tokens, or *optimized* from scratch. Attention matching the splat region ($k = 1$) gives the highest PSNR.

image: the *attention region* A_i, where Gaussian token $\mathbf{t}_i$ places its attention, and the *splat region* S_i, which the predicted Gaussian G_i covers when splatted onto the image.

The attention region A_i (Eq. (3)) is the smallest set of patch tokens that carries 90% of the attention weight a_{ij}, and the splat region S_i (Eq. (4)) is the smallest set that carries 90% of the Gaussian weight w_{ij} of G_i:

$$A_i = \arg\min_{A} \Big\{ |A| \;:\; \sum_{j \in A} a_{ij} \;\geq\; 0.9 \sum_{j} a_{ij} \Big\}, \tag{3}$$

$$S_i = \arg\min_{S} \Big\{ |S| \;:\; \sum_{j \in S} w_{ij} \;\geq\; 0.9 \sum_{j} w_{ij} \Big\}, \tag{4}$$

$$w_{ij} = \exp\Big(-\tfrac{1}{2} \big(\mathbf{x}_j - \boldsymbol{\mu}_i^{\mathrm{2D}}\big)^{\top} \big(\boldsymbol{\Sigma}_i^{\mathrm{2D}}\big)^{-1} \big(\mathbf{x}_j - \boldsymbol{\mu}_i^{\mathrm{2D}}\big) \Big), \tag{5}$$

where the Gaussian weight w_{ij} is the projected Gaussian evaluated at the center position $\mathbf{x}_j$ of patch token $\mathbf{f}_j$ on the input image. Geometrically, S_i approximately consists of the patch tokens inside the ellipse on which the Gaussian weight equals 0.1; for a 2D Gaussian, this ellipse encloses exactly 90% of the mass. Figs. 1 and 2 illustrate S_i by this ellipse.

We define a Gaussian token as aligned when its attention region A_i matches its splat region S_i, with the splat region as the target that the attention should cover. To quantify the match, we compute the precision Prec_i and recall Rec_i of A_i with respect to S_i:

$$\mathrm{Prec}_i = \frac{|A_i \cap S_i|}{|A_i|}, \qquad \mathrm{Rec}_i = \frac{|A_i \cap S_i|}{|S_i|}. \tag{6}$$

Specifically, we define the token $\mathbf{t}_i$ as aligned when both Prec_i and Rec_i are at least a threshold τ, and misaligned otherwise.

We answer our first question by observing C3G (An et al., 2026). We measure alignment with $\tau = 0.4$ for the C3G model trained on RealEstate10K (Zhou et al., 2018) over the RealEstate10K test scenes. We find that 31.2% of the Gaussian tokens are misaligned (Fig. 1). Nearly one out of every three Gaussians is therefore predicted from observations that are either diluted by visual features from elsewhere or capture too little of the region it reconstructs. Additionally, we observe a positive correlation between alignment and reconstruction quality across scenes. Averaged over a scene's Gaussian tokens, Prec_i and Rec_i have Spearman correlations (Spearman, 1904) of 0.61 and 0.32 with PSNR, respectively. Although not conclusive, this suggests that alignment is linked to reconstruction accuracy. We therefore test this directly with our second question: does better alignment between a Gaussian token's prediction and its observation affect the accuracy of the reconstruction?

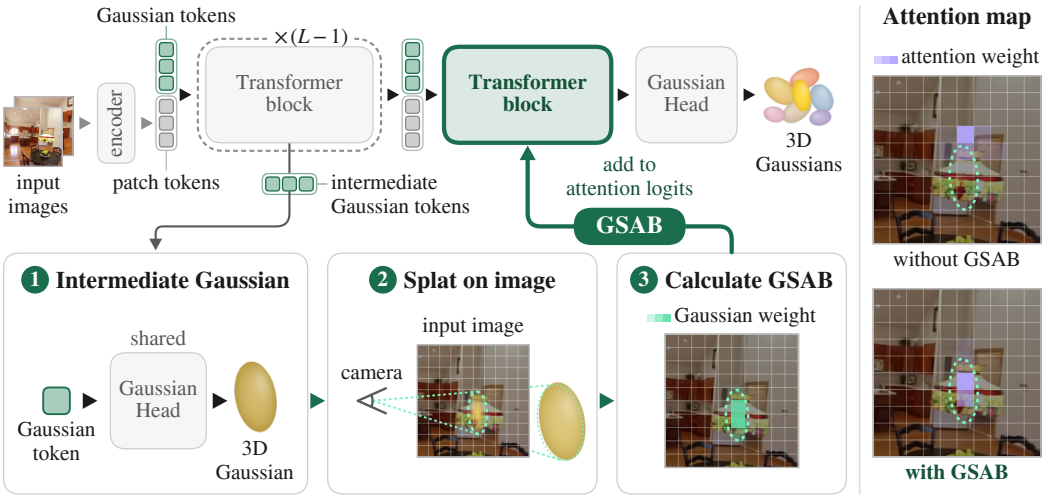


Figure 3: **Overview of GSAB.** The Gaussian weights of each token's intermediate Gaussian, splatted on the input image, are added to the attention logits of the final block as the attention bias. Right: a Gaussian token's attention over patch tokens without and with GSAB.

Better alignment improves reconstruction. To test whether the alignment affects the accuracy of the reconstruction, we design a toy experiment in which we can control the alignment. We run the toy experiment with the officially released C3G checkpoint. We fix the Gaussian geometry predicted by the pretrained checkpoint and freeze the image encoder, ensuring that the splat regions and patch tokens stay identical across settings. We then re-initialize the weights of the Gaussian tokens, the transformer blocks and the Gaussian Head, which learn only the color and opacity of each Gaussian under C3G's reconstruction loss.

With the splat regions fixed, the alignment depends only on the attention, and we compare the reconstruction accuracy when this attention is aligned with the splat region, misaligned with it to a controlled degree, or optimized from scratch. To control the alignment, we replace the cross-attention weights from Gaussian tokens to patch tokens with the normalized Gaussian weight w_{ij} of the token's own splat (Eq. (5)), with $\mathbf{\Sigma}_i^{2\mathrm{D}}$ scaled by k^2 for $k \in \{0.5, 1, 2, 4\}$, and keep them fixed during training. Values of k below or above 1 shrink or enlarge the attention region relative to the splat region, and thereby control the degree of alignment. Learned attention is optimized from scratch without any constraint, and uniform attention, which places the same weight on every patch token, serves as a further reference. Since every setting shares the same geometry and patch tokens, any difference in how accurately the color and opacity are learned comes from this attention alone. We train and evaluate on six RealEstate10K test scenes, chosen arbitrarily before any experiment was run.

Fig. 2 reports PSNR on held-out views after 1K training iterations. The toy experiment shows that attention with $k = 1$ is best on average, 1.23 dB above learned attention, and scaling the splat in either direction lowers the accuracy, by 0.21 dB at $k = 0.5$, 0.16 dB at $k = 2$ and 0.46 dB at $k = 4$. Therefore, within this toy experiment, better alignment raises the accuracy of the reconstruction. As both the observation and the toy experiment support our hypothesis, we propose Gaussian Splat Attention Bias (GSAB) to extend it to token-based feed-forward Gaussian prediction.

3.3 Gaussian Splat Attention Bias

Fig. 3 provides an overview of our proposed Gaussian Splat Attention Bias (GSAB).

Intermediate Gaussian prediction. GSAB predicts Gaussians at an intermediate transformer block and uses their splat regions in the final block's cross-attention. Let $\mathbf{t}_i^{(\ell)}$ be token i after block $\ell < L$; we share the Gaussian head h used for the final prediction to obtain $G_i^{(\ell)} = h(\mathbf{t}_i^{(\ell)})$.

Table 1: Reconstruction quality on RealEstate10K with two input views in the pose-free setting. All methods are evaluated from their released checkpoints at their default resolutions, except those marked with *, which are our reproductions. F^{4}Splat$^-$ uses the full Gaussian budget (ratio 1.0) and F^{4}Splat$^+$ its compact budget (ratio 0.2). Bold marks the best among methods with 2K Gaussians.

Method	#G	PSNR ↑	SSIM ↑	LPIPS ↓
NoPoSplat	131K	25.01	0.841	0.160
SPFSplat	131K	25.87	0.856	0.149
SPFSplatV2	100K	25.91	0.860	0.134
F^{4}Splat$^-$	131K	25.02	0.839	0.151
F^{4}Splat$^+$	26K	24.61	0.824	0.168
C3G	2K	22.53	0.718	0.230
C3G*	2K	21.86	0.689	0.267
C3G* + GSAB	2K	**23.24**	**0.745**	**0.226**

The splat region of the intermediate Gaussian must approximate that of the final Gaussian, since a misplaced intermediate Gaussian would guide attention toward the wrong patches. However, the baseline does not supervise intermediate Gaussian predictions. To address this, we also apply the final prediction's reconstruction loss to the intermediate $G_i^{(\ell)}$; we call this the intermediate reconstruction loss.

Gaussian weight as attention bias. GSAB projects the predicted intermediate Gaussian $G_i^{(\ell)}$ into each input view and calculates its Gaussian weight at every patch center, as in Eq. (5), using the intermediate 2D mean $\boldsymbol{\mu}_{i,v}^{\mathrm{2D},(\ell)}$ and covariance $\boldsymbol{\Sigma}_{i,v}^{\mathrm{2D},(\ell)}$. In the posed setting, projection uses the given camera parameters; for a pose-free baseline, a camera head predicts the input views' camera parameters.

Rather than directly using the Gaussian weight as an attention bias, GSAB adaptively adjusts its strength and spatial spread. In C3G, the logit standard deviation rises from 0.34 at initialization to 3.4 after 25K iterations and 232 after 50K, while the Gaussian weight is bounded by 1. GSAB therefore learns a bias strength s_i to compensate for the growing logit scale. GSAB also learns a spatial spread κ_i to control how tightly attention follows the splat region of the intermediate Gaussian, so that attention can extend beyond that region when the intermediate Gaussian is inaccurate. A two-layer MLP predicts $(s_i, \kappa_i) = \phi(\mathbf{t}_i^{(\ell)})$, with $s_i > 0$ and $\kappa_i \geq 1$. The attention bias, added to the attention logits of Eq. (1), is:

$$B_{ij} = s_i \cdot \exp\left(-\frac{1}{2\kappa_i}\left(\mathbf{x}_j - \boldsymbol{\mu}_i^{\mathrm{2D},(\ell)}\right)^\top \left(\boldsymbol{\Sigma}_i^{\mathrm{2D},(\ell)}\right)^{-1}\left(\mathbf{x}_j - \boldsymbol{\mu}_i^{\mathrm{2D},(\ell)}\right)\right). \tag{7}$$

Importantly, this adaptation preserves the spatial prior provided by the Gaussian weight: the learned strength and spread control only the magnitude and spatial extent of the guidance.

4 Experiments

4.1 Experimental Setup

Baselines. We apply GSAB to C3G (An et al., 2026), which predicts 2K Gaussians, one per Gaussian token, and to TokenGS (Ren et al., 2026), which predicts 262K Gaussians, 64 per Gaussian token. C3G + GSAB and TokenGS + GSAB are trained with the training settings of their baselines, with the changes in Appendix A.2, and we compare each with the released or reported results of its baseline as well as with our reproduction of the baseline under the same training settings. For a fair comparison, we select competing methods whose training and evaluation settings are consistent with those of C3G or TokenGS in the number of input views. Appendix B explains how the competing methods are selected. In the pose-free setting, we evaluate the officially released checkpoints of NoPoSplat (Ye et al., 2025), SPFSplat (Huang & Mikolajczyk, 2025a), SPFSplatV2 (Huang & Mikolajczyk, 2025b) and F^{4}Splat (Kim et al., 2026). In the posed setting, we list the published



Figure 4: Qualitative comparison on RealEstate10K. The number of Gaussians is at the bottom right.

results of MVSplat (Chen et al., 2024) and DepthSplat (Xu et al., 2025). Additionally, in our experience, TokenGS results also vary with the random seed. We therefore train our reproduction of TokenGS and TokenGS + GSAB with three seeds each and report the mean and standard deviation.

Datasets. C3G is trained and evaluated on RealEstate10K (Zhou et al., 2018), and TokenGS on DL3DV (Ling et al., 2024). RealEstate10K is used at 224×224 with 2 input views, and DL3DV with 4 input views during training. As in C3G (An et al., 2026) and NoPoSplat (Ye et al., 2025), each target camera pose is optimized for 100 iterations with the Gaussians frozen before the metrics are computed. On DL3DV, we evaluate on the DL3DV benchmark with 2, 4 and 6 input views.

Implementation details. We train C3G and C3G + GSAB for 450k iterations with a batch size of 8 per GPU on 8 H100 GPUs. TokenGS is trained on the same 8 H100 GPUs in two stages, 150k iterations at 256×256 with 1,024 Gaussian tokens and then 10k iterations at 256×448 with 4,096 Gaussian tokens, both with a batch size of 8 per GPU. The intermediate Gaussians for GSAB are decoded at block $\ell = L/2$, block 1 of 2 for C3G and block 6 of 12 for TokenGS. The intermediate Gaussian prediction is trained with the same reconstruction loss as the final Gaussian prediction. We follow the reconstruction loss from each baseline, a combination of MSE and LPIPS for C3G and of MSE, SSIM and a visibility term for TokenGS. Further implementation details are provided in Appendix A.

4.2 Results and Analysis

Pose-free setting. As shown in Table 1, C3G + GSAB improves over the released C3G checkpoint by 0.71 dB PSNR, reaching 23.24 dB with the same 2K Gaussians, and also improves SSIM and LPIPS. NoPoSplat, SPFSplat, SPFSplatV2 and F^{4}Splat$^-$ predict 100K to 131K Gaussians, about 50 to 64 times as many as C3G, and reach a higher PSNR of 25.01 to 25.91 dB. A more compact version, F^{4}Splat$^+$, uses 26K Gaussians, about 13 times as many as C3G + GSAB, and reaches 24.61 dB. GSAB narrows the gap between these methods and C3G without adding Gaussians. Appendix C.2 repeats this comparison with every method evaluated at 224×224. Figure 4 provides qualitative comparisons, where C3G + GSAB shows sharper structures and more accurate colors than C3G with the same 2K Gaussians.

Posed setting. As shown in Table 2, GSAB improves TokenGS by an average of 0.46, 0.66 and 0.80 dB PSNR with 2, 4 and 6 input views, and improves SSIM and LPIPS in every setting. The gains exceed the standard deviation over the three seeds in every setting. DepthSplat predicts one Gaussian per pixel, so its number of Gaussians grows in proportion to the number of input views, from 229K to 688K, whereas TokenGS + GSAB keeps 262K Gaussians in every setting. With this fixed number of Gaussians, TokenGS + GSAB improves from 19.46 to 23.87 dB PSNR as the input views increase from 2 to 6. With 6 input views, TokenGS + GSAB exceeds DepthSplat with 4 input views while using 43% fewer Gaussians. Although DepthSplat keeps a higher SSIM and a lower

Table 2: Reconstruction quality on DL3DV in the posed setting at 256×448. All methods are trained with 4 input views, except DepthSplat, which randomly samples from 2 to 6. Unmarked results are reported by TokenGS (Ren et al., 2026), and * marks our reproduction, with the mean and standard deviation over 3 seeds. Bold marks the better of TokenGS* and TokenGS* + GSAB.

#Input views	Method	PSNR ↑	SSIM ↑	LPIPS ↓	#G
2	MVSplat	17.54	0.529	0.402	229K
	DepthSplat	19.31	0.615	0.310	229K
	TokenGS	19.35	0.609	0.440	262K
	TokenGS*	19.00±0.23	0.580±0.010	0.456±0.009	262K
	TokenGS* + GSAB	**19.46**±0.05	**0.603**±0.003	**0.433**±0.004	262K
4	MVSplat	21.63	0.721	0.233	458K
	DepthSplat	23.12	0.780	0.178	458K
	TokenGS	23.02	0.747	0.326	262K
	TokenGS*	22.36±0.30	0.713±0.013	0.353±0.013	262K
	TokenGS*+ GSAB	**23.02**±0.16	**0.742**±0.007	**0.319**±0.008	262K
6	MVSplat	22.93	0.775	0.193	688K
	DepthSplat	24.19	0.823	0.147	688K
	TokenGS	23.69	0.766	0.311	262K
	TokenGS*	23.07±0.30	0.735±0.011	0.336±0.012	262K
	TokenGS*+ GSAB	**23.87**±0.19	**0.767**±0.008	**0.299**±0.009	262K

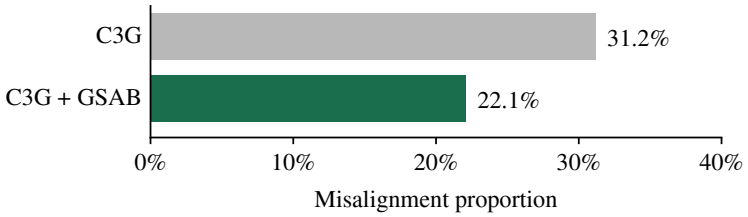


Figure 5: GSAB reduces the share of misaligned Gaussian tokens in C3G, measured as in Fig. 1.

LPIPS in all settings, this gap is already present for TokenGS, and GSAB reduces it. Qualitative comparisons are given in Appendix C.3.

Attention alignment. We next examine whether GSAB aligns the attention of each Gaussian token with its splat region as intended. As in Sec. 3.2, a token is misaligned when the precision or the recall of its attention region A_i with respect to its splat region S_i falls below $\tau = 0.4$ (Eqs. (3), (4) and (6)). Figure 5 shows that GSAB reduces the share of misaligned tokens in C3G from 31.2% to 22.1%. Together with the PSNR gain in Table 1, this indicates that the improvements in alignment and reconstruction quality coincide.

Intermediate Gaussian prediction. GSAB builds its attention bias from the Gaussians decoded at the intermediate block, so it relies on their splat regions matching those of the final Gaussians. Figure 6 draws the splat regions of the Gaussians predicted by 10 sampled Gaussian tokens at the intermediate and the final block, and shows that the two mostly overlap, which makes the intermediate Gaussians a reliable basis for the attention bias. Comparing the crops rendered from the intermediate and the final Gaussians further shows that the final block refines the colors of the predicted Gaussians for a more accurate reconstruction.

Ablation study. Table 3 compares C3G when the intermediate reconstruction loss and GSAB are added one at a time, after 200k training iterations. The intermediate reconstruction loss alone does not improve any of the three metrics, as PSNR decreases from 21.64 to 21.50 dB. The intermediate reconstruction loss is therefore a prerequisite of GSAB rather than one of its contributions. Adding GSAB raises PSNR to 22.33 dB and improves SSIM and LPIPS, showing that the gain comes from GSAB. An ablation of the camera condition in the pose-free setting is given in Appendix C.1.

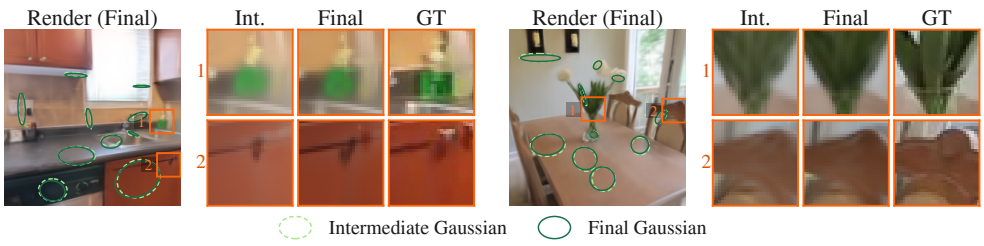


Figure 6: Splat regions of Gaussians predicted from the same token at the intermediate (dashed) and final (solid) block overlap for 10 sampled tokens of C3G + GSAB. Orange boxes mark 30×30-pixel crops that highlight the color change from the intermediate (Int.) to the final (Final) render.

Table 3: Ablation of the intermediate reconstruction loss and GSAB on RealEstate10K.

Baseline	Intermediate recon. loss	GSAB	PSNR ↑	SSIM ↑	LPIPS ↓
C3G	✘	✘	21.64	0.674	0.260
	✔	✘	21.50	0.667	0.271
	✔	✔	**22.33**	**0.704**	**0.256**

5 Conclusion

Based on our observation that the Gaussian tokens of token-based feed-forward Gaussian prediction models often do not attend to their own splat regions, we hypothesize that a token predicts its Gaussians more accurately when it attends to image patches within its own splat region. Supported by a toy experiment, we propose GSAB, which improves the alignment between the attention of each token and its splat region using Gaussians decoded at an intermediate block. GSAB projects these intermediate Gaussians into the input views, and their splat regions determine the attention bias added to the cross-attention between Gaussian tokens and patch tokens in the final block. Because GSAB only requires an additional decoding by the Gaussian head at the intermediate block, it can be applied to a wide range of token-based feed-forward Gaussian prediction models. Empirically, GSAB increases the alignment and improves reconstruction quality on a pose-free model with 2K Gaussians, and also improves reconstruction quality on a posed model with 262K Gaussians. Therefore, GSAB makes compact scene reconstructions with 2K Gaussians more accurate without making them larger, and likewise improves token-based feed-forward Gaussian prediction models that predict 262K Gaussians.

Limitations. With only two input views and 2K Gaussians, C3G + GSAB reconstructs fine structures and textures inaccurately (Fig. 7). In such a compact reconstruction, thin, elongated structures are often represented by chains of shorter Gaussians that can be disconnected or misplaced. Likewise, fine textures are reconstructed with Gaussians that are too large and too sparsely spaced for them, so individual Gaussians appear as visible blobs and lower the fidelity of the rendering. Although GSAB improves the overall reconstruction quality, it does not fully overcome these limitations, which likely stem from the sparse input views and the small number of Gaussians.

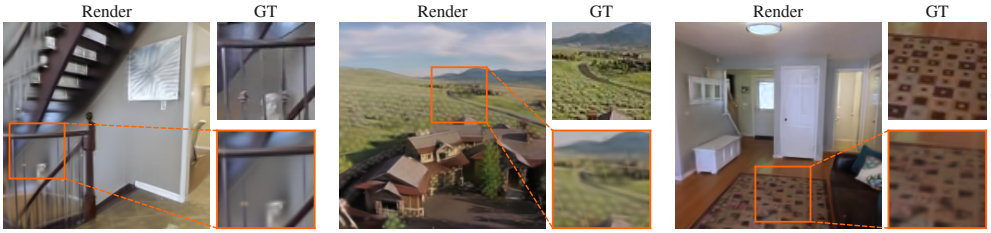


Figure 7: Failure case examples of C3G + GSAB.

AI USE STATEMENT

We used generative AI tools for research ideation and execution, namely refining hypotheses, designing experiments, implementing methods, preparing datasets and interpreting results. We did not use them to generate synthetic data or to prove mathematical claims. Additionally, we used generative AI tools to find related work, create figures, and draft and polish the paper. We have reviewed all AI-assisted work. All AI-drafted or AI-polished text was reviewed by at least two authors, and we verified that the AI-generated code implements the method described in this paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work improves feed-forward 3D reconstruction and uses only the public RealEstate10K and DL3DV datasets, following their terms of use. As with any method that reconstructs real scenes from images, it could be applied to private spaces without consent, and we intend it for research on 3D reconstruction only.

REPRODUCIBILITY STATEMENT

Section 4.1 describes the baselines, datasets, evaluation protocol and training settings, and Appendix A gives the remaining implementation details needed to reproduce our results. We will release our code and trained models 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 363–373, 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)*, pp. 213–229, 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (ECCV)*, pp. 370–386, 2024.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1299, 2022.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3621–3630, 2021.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3D gaussian splatting from sparse views. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 27947–27957, 2025a.

Ranran Huang and Krystian Mikolajczyk. SPFSplatV2: Efficient self-supervised pose-free 3D gaussian splatting from sparse views. *arXiv preprint arXiv:2509.17246*, 2025b.

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.

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 (TOG)*, 44(6):1–16, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

Injae Kim, Chaehyeon Kim, Minseong Bae, Minseok Joo, and Hyunwoo J. Kim. F4Splat: Feed-forward predictive densification for feed-forward 3D gaussian splatting. In *European Conference on Computer Vision (ECCV)*, pp. 1–19, 2026.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40886–40895, 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.

Yutong Lin, Yuhui Yuan, Zheng Zhang, Chen Li, Nanning Zheng, and Han Hu. DETR does not need multi-scale or locality design. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 6545–6554, 2023.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169, 2024.

Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations (ICLR)*, 2022.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3651–3660, 2021.

Arthur Moreau, Richard Shaw, Michal Nazarczuk, Jisu Shin, Thomas Tanay, Zhensong Zhang, Songcen Xu, and Eduardo Pérez-Pellitero. Off The Grid: Detection of primitives for feed-forward 3D gaussian splatting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11756–11766, 2026.

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

Charles Edward Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024a.

Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. FreeSplat: Generalizable 3D gaussian splatting towards free view synthesis of indoor scenes. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 107326–107349, 2024b.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting gaussian splatting and depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463, 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)*, volume 2025, pp. 54009–54033, 2025.

Mingqiao Ye, Lei Ke, Siyuan Li, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Cascade-DETR: Delving into high-quality universal object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 6704–6714, 2023.

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)*, pp. 1–19, 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18924–18933, 2026a.

Zicheng Zhang, Xiangting Meng, Ke Wu, and Wenchao Ding. SparseSplat: Towards applicable feed-forward 3D gaussian splatting with pixel-unaligned prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5049–5058, 2026b.

Jia-Chen Zhao, Beiqi Chen, Xinyang Chen, Guangcong Wang, and Liqiang Nie. StructSplat: Generalizable 3D gaussian splatting from uncalibrated sparse views. *arXiv preprint arXiv:2606.28321*, 2026.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

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 International Conference on Computer Vision (ICCV)*, pp. 4349–4359, 2025.

A Implementation Details

A.1 Gaussian Splat Attention Bias

Bias strength and spatial spread. A two-layer MLP ϕ with a GELU activation predicts the bias strength $s_i = \mathrm{softplus}(\cdot)$ and the spatial spread $\kappa_i = \mathrm{softplus}(\cdot) + 1$ of each token from $\mathbf{t}_i^{(\ell)}$. Its last layer is initialized to zero, so every token starts from $s_i = \log 2$ and $\kappa_i = 1 + \log 2$. ϕ is trained through the final reconstruction loss.

Camera head. Following the training settings of the baseline, GSAB is trained with the ground-truth cameras. For pose-free evaluation, we fine-tune VGGT's pretrained camera head for 10k iterations on the input views of the training set, with the trained model frozen. The predicted camera parameters of the input views replace the ground truth in GSAB, and the target view is rendered with the mean predicted field of view of the input views, so that GSAB and the rendering both use the predicted cameras.

A.2 Training

Losses. C3G uses MSE and LPIPS (weight 0.05), and TokenGS uses MSE, SSIM (weight 0.2) and a visibility term. In C3G + GSAB and TokenGS + GSAB, the intermediate reconstruction loss applies the same loss to the renders of the intermediate Gaussians and is added with equal weight to the reconstruction loss of the final Gaussians.

Supervision views. Both models supervise the input views as well as four target views per training scene. For TokenGS, this option is available in the official code but is neither enabled by default nor described in the paper, and we enable it because it narrows the gap between our reproduction and the reported results. Our reproduction of TokenGS remains below its reported results, likely due to differences in the training data and in training details not specified in the paper.

Optimization. C3G uses AdamW ($\beta = (0.9, 0.95)$, weight decay 0.05) with learning rates of 10^{-4} for the decoder, the Gaussian head and ϕ and 10^{-6} for the VGGT backbone, a 1,000-iteration warm-up followed by cosine decay, full precision and gradient clipping at 0.5. TokenGS is trained with AdamW and a one-cycle schedule at a peak learning rate of $4{\times}10^{-4}$, bf16 precision and gradient clipping at 1.0.

Gradient gating. In C3G training, any Gaussian whose position-gradient norm exceeds 0.005 has its entire gradient set to zero. This keeps rare Gaussians predicted near a camera, whose gradients become unusually large, from corrupting the shared Gaussian head, which otherwise leads to a training collapse in which all Gaussians end up behind the camera. The gate is applied to the final Gaussians of our C3G reproduction and to both the intermediate and the final Gaussians of C3G + GSAB.

A.3 Datasets

RealEstate10K. Frames are resized and center-cropped from 360×640 to 224×224. Since RealEstate10K is distributed as YouTube video identifiers, only a shrinking subset of its scenes remains downloadable, and our copy holds 66,033 training scenes. We follow the view sampler in the official code of C3G (An et al., 2026), which draws two input views and four target views per scene, with the gap between input frames widened from 25 frames to a uniform draw over 45 to 90 frames during the first 37.5k iterations.

DL3DV. From the 10,581 scenes of DL3DV-ALL-960P, we hold out the 140 benchmark scenes and undistort the rest with the preprocessing script in the official code of Long-LRM (Ziwen et al., 2025), skipping the 17 portrait-oriented scenes as that script does. The benchmark scenes are left untouched, and evaluation uses their original frames. We follow the view sampler in the official code of TokenGS (Ren et al., 2026) rather than the one described in its paper, which draws a 10–40-frame span, takes its two endpoints and random frames in between as input views, and draws the target views from the same span.

Table 4: Ablation of the camera condition on RealEstate10K after 200k iterations. FoV denotes the field of view used for rendering in training and evaluation. *Pred* uses a VGGT camera head fine-tuned for 10k iterations on each frozen backbone, *GT* uses the camera parameters provided by the dataset, and the target-view extrinsics are *GT* in every row.

Method	GSAB camera	Train render FoV	Eval render FoV	PSNR ↑	SSIM ↑	LPIPS ↓
C3G	N/A	GT	GT	21.64	0.674	0.260
	N/A	GT	Pred	21.96	0.686	**0.253**
C3G + GSAB	Pred	GT	GT	21.93	0.688	0.264
	Pred	GT	Pred	**22.33**	**0.704**	0.256
	GT	GT	GT	22.22	0.699	0.262

Table 5: Reconstruction quality on RealEstate10K with two input views in the pose-free setting, with every method evaluated at 224×224. Our reproductions are marked with *, and methods whose default resolution is 256×256 with †.

Method	#G	PSNR ↑	SSIM ↑	LPIPS ↓
NoPoSplat†	100K	20.26	0.643	0.233
SPFSplat†	100K	25.19	0.840	0.148
SPFSplatV2	100K	25.91	0.860	0.134
F^{4}Splat$^{-\dagger}$	100K	24.88	0.840	0.143
F^{4}Splat$^{+\dagger}$	20K	23.96	0.809	0.188
C3G	2K	22.53	0.718	0.230
C3G*	2K	21.86	0.689	0.267
C3G* + GSAB	2K	23.24	0.745	0.226

B Selection of Competing Methods

We include a competing method only if it is trained and evaluated with the same number of input views as the baseline it is compared with, on the same evaluation scenes. The evaluation resolution may differ, and Appendix C.2 repeats the pose-free comparison with every method at 224×224. Methods trained with a different number of input views are excluded, since their results are not directly comparable, and are discussed in Sec. 2 instead. The one exception is DepthSplat (Xu et al., 2025), whose DL3DV model is fine-tuned from a RealEstate10K model with two to six input views sampled at random and is evaluated with two, four and six input views. We regard this as comparable to training with four input views and include DepthSplat in Table 2.

C More Results

C.1 Camera Condition

Since GSAB uses predicted cameras in the pose-free setting, its gain could instead come from rendering with the predicted field of view. Table 4 separates these two effects. When rendering with the ground-truth field of view, GSAB improves C3G from 21.64 dB to 21.93 dB with predicted cameras and to 22.22 dB with ground-truth cameras, and when rendering with the predicted field of view, it improves C3G from 21.96 dB to 22.33 dB. Thus, GSAB improves C3G regardless of the rendering field of view.

C.2 Pose-Free Setting at 224×224

Table 1 evaluates each method at its own default resolution. Table 5 repeats that comparison with every method evaluated at 224×224.



Figure 8: Qualitative comparison on DL3DV in the posed setting with 4 input views, for three benchmark scenes. DepthSplat predicts 458K Gaussians, and TokenGS and TokenGS + GSAB predict 262K.

C.3 Qualitative Results in the Posed Setting

Figure 8 compares TokenGS + GSAB with TokenGS and DepthSplat on three DL3DV benchmark scenes with 4 input views, where the TokenGS renderings come from its official checkpoint trained with 4 input views. DepthSplat predicts 458K Gaussians, which helps it reconstruct fine details, whereas TokenGS and TokenGS + GSAB predict 262K. At the same number of Gaussians, TokenGS + GSAB reconstructs fine details with better fidelity than TokenGS.