EVISR-GS: EVIDENCE-GUIDED SUPERVISION AND REFINEMENT FOR SPARSE-VIEW GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse-view Gaussian splatting faces spatially uneven constraints from limited observations. Stronger pixelwise fitting can reinforce ambiguous reconstructions, while regions that could benefit from additional capacity may lack reliable cues for Gaussian placement. We propose EviSR-GS, an evidence-guided framework that combines spatially adaptive supervision with provisional Gaussian refinement, separating capacity allocation from subsequent activity assessment. For supervision, local depth residuals and image–depth gradient mismatch modulate the relative weights of pixelwise RGB reconstruction and patch-level perceptual losses. For refinement, low rendered coverage or disagreement with a fixed depth prior identifies candidate regions. Cross-view depth statistics and current coverage prioritize depth-based Gaussian proposals under a per-event insertion budget. Newly inserted Gaussians are jointly optimized with the existing representation and protected from ordinary pruning during a maturation period. Afterwards, additions with both low opacity and low accumulated rendering visibility undergo soft opacity attenuation. Experiments on LLFF, DTU, and Mip-NeRF 360 demonstrate competitive novel-view synthesis performance. On three-view LLFF, the full framework improves PSNR by 0.69 dB and reduces LPIPS by 0.021 relative to a baseline with both branches disabled. Ablations further support the contributions of both branches and the benefit of spatially varying loss weights over fixed scene-level weights. Code will be made publicly available.

1 INTRODUCTION

Neural Radiance Fields and 3D Gaussian Splatting (3DGS) have advanced novel-view synthesis by reconstructing scene representations from posed images (Mildenhall et al., 2021; Kerbl et al., 2023). With sparse input views, however, reconstruction remains underconstrained: representations that fit the observed images similarly can produce different results at unseen viewpoints. These ambiguities vary across regions, appearing as local blur, boundary artifacts, and incomplete coverage. Existing methods address sparse-view reconstruction through regularization and generalizable representations (Niemeyer et al., 2022; Deng et al., 2022; Yu et al., 2021). Sparse-view 3DGS further incorporates depth and matching priors, together with cross-view consistency, to constrain reconstruction under limited observations (Li et al., 2024; Zheng et al., 2025; Kim et al., 2026; Paliwal et al., 2024; Zhao et al., 2025). Despite these advances, fitting the observed images more closely does not necessarily resolve local reconstruction ambiguities.

This creates two coupled optimization difficulties. Geometry and appearance can compensate for each other, so stronger pixel fitting may reinforce an ambiguous reconstruction (Xiong et al., 2026). Adding Gaussians increases the freedom to fit image residuals, but unreliable placement cues can introduce poorly supported structure. Representation and densification regularization (Zhu et al., 2024; Yin et al., 2024; Xu et al., 2025; Park et al., 2025; Fang et al., 2026; Li et al., 2026; Zhang et al., 2024) and geometric or generative priors (Wu et al., 2025; Wang et al., 2024; Paliwal et al., 2025) address these difficulties. Their local interaction motivates our question: how should available evidence guide supervision, capacity allocation, and the assessment of newly added capacity?

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.

We propose EviSR-GS, an evidence-guided framework that combines spatially adaptive supervision with provisional Gaussian refinement. The framework uses fixed input cues and the evolving representation to guide how local regions are supervised, where new capacity is allocated, and how additions are assessed after optimization. Local depth residuals and image–depth gradient mismatch identify regions where relying exclusively on precise pixel fitting may be less appropriate. EviSR-GS spatial supervision reduces its relative emphasis in these regions while retaining patch perceptual constraints; rendered coverage guides perceptual region construction. This maintains appearance supervision while adapting its spatial precision to input consistency. D^{2}GS uses depth and density to regulate dropout and strengthens distant-region RGB fitting (Song et al., 2025); UGOT matches patch depth distributions through uncertainty-weighted optimal transport (Sun et al., 2024). We use input consistency to balance pixel and patch RGB constraints.

Spatial supervision adjusts the image constraints used to optimize the representation, while our refinement branch allocates additional Gaussians to candidate regions. For candidates identified by low rendered coverage or rendered–prior depth disagreement, weak cross-view support may reflect either limited overlap or unreliable geometry. An informative depth prior may make additional capacity useful despite limited corroboration, whereas a misleading prior may reinforce errors. We therefore allocate provisional Gaussians according to refinement demand under a per-event insertion budget and jointly optimize them with the existing representation. Their lineages are protected from ordinary pruning during a maturation period measured from the original insertion. After maturation, additions with both low opacity and low accumulated rendering visibility undergo soft opacity attenuation. Our refinement branch separates capacity allocation from subsequent activity assessment: refinement demand guides insertion, and activity after joint optimization guides attenuation.

Experiments on LLFF (Mildenhall et al., 2019), DTU (Aanæs et al., 2016), and Mip-NeRF 360 (Barron et al., 2022) demonstrate competitive novel-view synthesis performance. LLFF ablations show improvements from both branches and further gains from spatial weighting over uniform weights. The main contributions of EviSR-GS are:

- EviSR-GS, an evidence-guided framework that coordinates local supervision, provisional capacity allocation, and post-optimization activity assessment for sparse-view Gaussian reconstruction.
- Spatially adaptive supervision that balances pixelwise RGB reconstruction and patch-level perceptual constraints according to local input consistency, with rendered coverage guiding perceptual region construction.
- Budgeted provisional refinement that prioritizes depth-based proposals using refinement demand derived from cross-view depth statistics and rendered coverage, protects inserted lineages from ordinary pruning during maturation, and subsequently adjusts their influence through activity-based opacity attenuation.

2 PRELIMINARIES

Given N posed images $\mathcal{S} = \{(I_i, P_i)\}_{i=1}^{N}$, where P_i denotes the camera parameters associated with I_i, 3DGS represents a scene using M Gaussian primitives (Kerbl et al., 2023):

$$\mathcal{G} = \{g_j\}_{j=1}^{M}, \qquad g_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \alpha_j, \mathbf{c}_j), \tag{1}$$

where $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$, α_j, and $\mathbf{c}_j$ specify position, covariance, opacity, and appearance, respectively. The representation is optimized through differentiable rendering to minimize reconstruction error in the observed views:

$$\begin{aligned} \widehat{I}_i &= \mathcal{R}(\mathcal{G}, P_i), \\ \mathcal{L}_{\text{obs}}(\mathcal{G}) &= \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{img}}(\widehat{I}_i, I_i), \end{aligned} \tag{2}$$

where $\mathcal{R}$ denotes Gaussian splatting and ℓ_{img} is a generic image reconstruction loss. Alongside parameter optimization, 3DGS updates the Gaussian population through density control: cloning and splitting introduce additional primitives, while pruning removes selected primitives.

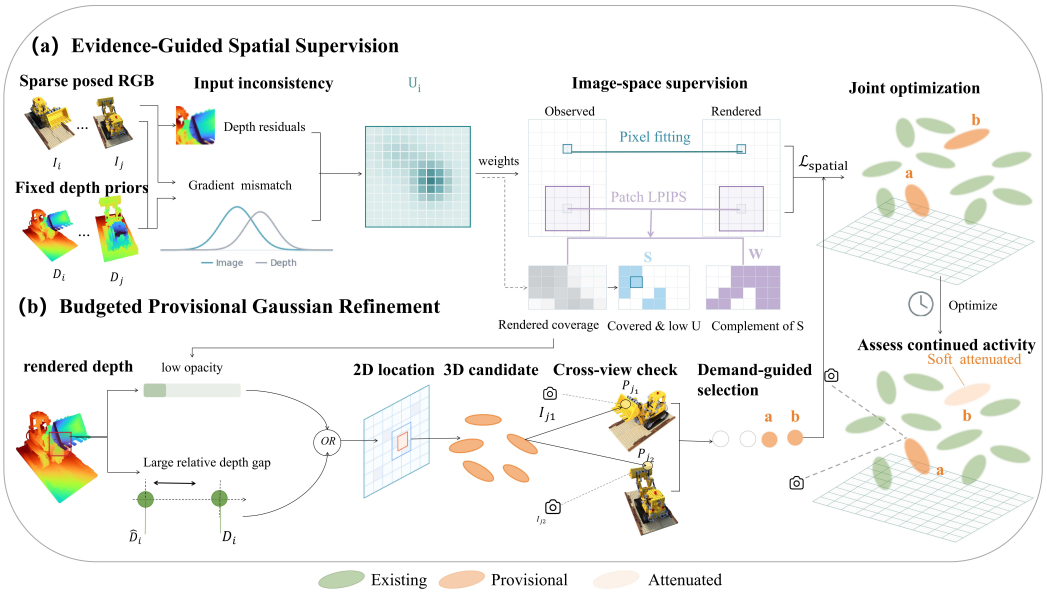

Figure 1: **Evidence-guided supervision and provisional Gaussian refinement.** (a) Fixed RGB–depth evidence and rendered coverage define an inconsistency-aware partition into regions S and W, which determines the pixel and patch supervision used to form the spatial objective. (b) Low-coverage or depth-inconsistent pixels are back-projected into provisional 3D candidates, checked across views, ranked by demand, and selected under a top-B budget. The selected candidates are jointly optimized with existing Gaussians and retained or softly attenuated after maturation according to their activity.

3 METHOD

EviSR-GS combines evidence-guided spatial supervision with budgeted provisional Gaussian refinement. The first branch adjusts local appearance constraints; the second allocates new capacity and assesses its activity after maturation. We first define the evidence used by both branches, then describe their optimization rules. Figure 1 summarizes the framework.

3.1 EVIDENCE FOR SUPERVISION AND REFINEMENT

For each training view i, the fixed inputs are reference RGB I_i, external depth D_i, and its validity mask V_i. The evolving state at iteration t includes rendered opacity A_i^t, rendered depth $\widehat{D}_i^t$, and per-Gaussian activity statistics.

Evidence for supervision. Local depth residuals describe departures from a smooth neighborhood, while image–depth gradient mismatch describes disagreement between appearance and depth transitions. We combine them into a soft input-inconsistency indicator:

$$U_i(p) = \mathrm{clip}\big(\beta\widetilde{v}_i(p) + (1-\beta)m_i(p), 0, 1\big), \qquad Q_i(p) = 1 - U_i(p), \tag{3}$$

where $\widetilde{v}_i$ is normalized local depth residual energy, and m_i compares gradient magnitudes of grayscale RGB and normalized depth. We use $\beta = 0.6$. Both indicators are computed from fixed inputs; invalid-depth pixels use $U_i = 0$ and $Q_i = 1$. Indicator computation is detailed in Appendix A.1.

Evidence for refinement. Rendered opacity A_i^t measures image-space coverage, while rendered depth $\widehat{D}_i^t$ provides the current geometry for comparison with D_i.

For a candidate position j, n_j counts training views other than the source view whose prior depth agrees with the projected candidate depth. The statistic $e_j \in [0, 1]$ is the normalized mean relative depth discrepancy over valid cross-view comparisons. For an inserted Gaussian, current opacity α_j^t

and accumulated rendering visibility ν_j^t characterize its activity. Specifically, ν_j^t counts training iterations in which the renderer marks it as visible. Cross-view statistics are detailed in Appendix A.2.

3.2 Evidence-Guided Spatial Supervision

Pixel reconstruction constrains RGB values at individual locations, whereas patch perceptual supervision compares appearance features over local neighborhoods. We use input inconsistency to adjust their relative emphasis: lower inconsistency retains stronger pixel constraints, while higher inconsistency increases the relative weight of patch appearance. The pixel term therefore uses the complementary indicator Q_i:

$$\mathcal{L}_{\mathrm{pix}}^{(i)} = \frac{1}{3|\Omega_i|} \sum_{p \in \Omega_i} Q_i(p) \left\| \widehat{I}_i^t(p) - I_i(p) \right\|_2^2, \tag{4}$$

where Ω_i is the image domain and the factor of three averages over RGB channels.

Input consistency does not indicate whether the current representation covers a region. We therefore also use rendered coverage to organize perceptual supervision. Region s contains covered pixels with $U_i < \tau_u$; region w is its complement within the training image. The coverage mask and dilation are specified in Appendix A.1.

Within each region, the mean masked U_i weights patch-level VGG-LPIPS distances (Zhang et al., 2018). The regional loss $\mathcal{L}_{\mathrm{perc},r}^{(i)}$ averages these weighted distances over patches with non-negligible weights; an empty patch set contributes zero. We combine the regional losses as

$$\mathcal{L}_{\mathrm{patch}}^{(i)} = g_t \left(\lambda_s \mathcal{L}_{\mathrm{perc},s}^{(i)} + \lambda_w \mathcal{L}_{\mathrm{perc},w}^{(i)} \right), \tag{5}$$

with $\lambda_s = 0.5$ and $\lambda_w = 1$, assigning a larger coefficient to region w while retaining input-dependent patch weights. Perceptual evaluation runs every five iterations after iteration 500. To compensate for this interval, $g_t = 5$ when $t > 500$ and $t \bmod 5 = 0$, and zero otherwise.

The complete spatial objective combines both terms:

$$\mathcal{L}_{\mathrm{spatial}}^{(i)} = \lambda_{\mathrm{pix}} \mathcal{L}_{\mathrm{pix}}^{(i)} + \lambda_{\mathrm{patch}} \mathcal{L}_{\mathrm{patch}}^{(i)}, \tag{6}$$

with $\lambda_{\mathrm{pix}} = 2$ and $\lambda_{\mathrm{patch}} = 0.08$. For Full, a view with no valid prior depth uses only the pixel term with unit spatial weights. Appendix A.1 specifies patch construction, retention, loss reduction, and the uniform-weight control.

3.3 Budgeted Provisional Gaussian Refinement

Cross-view support describes the agreement of a proposed Gaussian position with the available depth priors, as measured by the number of agreeing views and their depth discrepancies. Refinement demand uses these statistics together with rendered coverage to prioritize additional capacity. These priorities need not favor the most strongly supported placements: we allocate a limited optimization opportunity according to demand, then assess the activity of inserted Gaussians after joint optimization.

Refinement candidates. We first restrict proposals to locations where the current rendering has low coverage or disagrees with the depth prior. These conditions identify where an addition may change the current reconstruction:

$$\begin{aligned} \delta_i^t(p) &= \frac{\left| \widehat{D}_i^t(p) - D_i(p) \right|}{|D_i(p)| + \epsilon}, \\ \mathcal{D}_i^t(p) &= J_i^t(p)\, \mathbb{I}\left[A_i^t(p) < \tau_\alpha \ \vee \ \delta_i^t(p) > \tau_d \right], \end{aligned} \tag{7}$$

where J_i^t requires validity under V_i and finite positive prior and rendered depths, and $\epsilon > 0$ stabilizes the ratio. The positive rendered-depth requirement excludes completely uncovered pixels. Triggered pixels are preselected by $s_0(p) = 1 - A_i^t(p) + \delta_i^t(p)$ and backprojected using the external depth prior to obtain candidate positions.

Budgeted optimization opportunity. Within the triggered candidate set, we prioritize locations with fewer agreeing views, larger cross-view depth discrepancy, or weaker rendered coverage. Using n_j and e_j from Section 3.1, we define

$$\rho_j = \frac{w_1}{n_j + 1} + w_2 e_j + w_3\big(1 - A_i^t(p_j)\big). \tag{8}$$

We use $(w_1, w_2, w_3) = (0.4, 0.4, 0.2)$, giving $\rho_j \in [0, 1]$. The inverse-support term favors less constrained candidates, while the discrepancy and coverage terms account for cross-view disagreement and the current rendering state. Together, these terms assign priority to an optimization opportunity.

Given a per-event budget $B = 1024$, we select at most B candidates satisfying $\rho_j \geq \tau_\rho$, ranked by $s_0(p_j) + \lambda_\rho \rho_j$. The budget limits the number of additions introduced under this demand-based rule.

Joint optimization and maturation. Selected additions are initialized from external depth and source RGB, then optimized jointly with the existing Gaussians under the same training objective. Because their early opacity and visibility reflect placement and initialization, we allow a maturation period before activity-based assessment. Each addition receives a birth iteration b_j. Ordinary density control excludes tagged lineages with $t - b_j < T_m$ from its pruning mask, and cloning and splitting preserve the lineage birth time. After maturation, the exemption expires; surviving lineages follow ordinary pruning rules and remain eligible for activity-based attenuation.

Activity-based attenuation. After maturation, opacity α_j^t and accumulated rendering visibility ν_j^t provide complementary activity indicators. Opacity controls the Gaussian's contribution during alpha blending, while accumulated visibility records how often the renderer has marked it as visible during training. At scheduled assessments, additions with both low opacity and low accumulated visibility have their opacity multiplied by $\kappa \in (0, 1)$. The conjunction prevents either low opacity or low recorded visibility alone from triggering attenuation. This update reduces the Gaussian's immediate influence while retaining its parameters for subsequent optimization. Ordinary density control continues to govern pruning.

Appendix A.2 specifies cross-view comparisons, selection rules, demand-dependent initialization, and lifecycle rules; Appendix A.3 gives the training schedule.

4 Experiments

We evaluate sparse-view reconstruction across multiple benchmarks and analyze the contributions of both branches, the effects of input-view count, candidate selection, and post-maturation assessment.

4.1 Experimental Setup

Datasets and metrics. We evaluate on LLFF (Mildenhall et al., 2019), DTU (Aanæs et al., 2016), and Mip-NeRF 360 (Barron et al., 2022). The main comparison uses three input views for LLFF and DTU and 24 input views for Mip-NeRF 360. We additionally evaluate LLFF with six and nine input views. We report PSNR, SSIM, and LPIPS (Zhang et al., 2018). Dataset and evaluation details are provided in Appendix B.1.

Comparisons and training. We compare with representative sparse-view NeRF and 3DGS-based methods using their reported results and protocols. The NeRF-based baselines include DietNeRF, FreeNeRF, SparseNeRF, and ViP-NeRF (Jain et al., 2021; Yang et al., 2023; Wang et al., 2023; Somraj & Soundararajan, 2023). All experiments use the NexusGS (Zheng et al., 2025) input preprocessing, including its fixed depth maps, camera parameters, validity masks, and scene normalization. We train for 30,000 iterations with spatial-loss coefficients of 2 and 0.08. Refinement uses $\tau_\rho = 0.75$ and a per-event budget of $B = 1024$. Experiments are conducted on NVIDIA RTX 4090 GPUs. Further implementation and evaluation details are provided in Appendices A.1, A.2, and B.1.

Table 1: Sparse-view reconstruction on LLFF and DTU with three input views and Mip-NeRF 360 with 24 input views. Baseline results are taken from the corresponding papers and reported under their original evaluation protocols. Bold and underlined values indicate the best and second-best reported results in each column. '—' indicates results not included in this comparison.

	LLFF (3 views)			DTU (3 views)			Mip-NeRF 360 (24 views)		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-based methods									
Mip-NeRF	14.62	0.351	0.495	8.68	0.571	0.353	21.23	0.613	0.351
DietNeRF	14.94	0.370	0.496	11.85	0.633	0.314	20.21	0.557	0.387
RegNeRF	19.08	0.587	0.336	18.89	0.745	0.190	22.19	0.643	0.335
FreeNeRF	19.63	0.612	0.308	19.92	0.787	0.182	22.78	0.689	0.323
SparseNeRF	19.86	0.624	0.328	19.55	0.769	0.201	22.85	0.693	0.315
DS-NeRF	19.34	0.596	0.295	16.29	0.559	0.451	14.58	0.311	0.692
ViP-NeRF	18.50	0.528	0.321	10.20	0.301	0.363	14.78	0.300	0.774
SimpleNeRF	18.82	0.587	0.274	11.94	0.387	0.286	10.82	0.142	0.880
3DGS-based methods									
3DGS	18.54	0.588	0.272	17.65	0.816	0.146	21.71	0.672	0.248
DNGaussian	19.12	0.591	0.294	18.91	0.790	0.176	18.06	0.423	0.584
FSGS	20.43	0.682	0.248	17.14	0.818	0.162	23.40	0.733	0.238
CoR-GS	20.45	0.712	0.196	19.21	0.853	0.119	23.55	0.727	0.226
UGOT	19.77	0.625	0.273	19.31	0.808	0.160	—	—	—
D^2GS	21.35	0.746	0.179	—	—	—	24.13	<u>0.763</u>	0.221
NexusGS	21.07	0.738	0.177	20.21	0.869	<u>0.102</u>	23.86	0.753	<u>0.206</u>
TWINGS	<u>21.49</u>	<u>0.754</u>	<u>0.167</u>	**21.52**	**0.880**	0.107	<u>24.17</u>	0.762	0.252
EviSR-GS	**21.68**	**0.766**	**0.157**	<u>20.36</u>	<u>0.872</u>	**0.098**	**24.28**	**0.766**	**0.196**

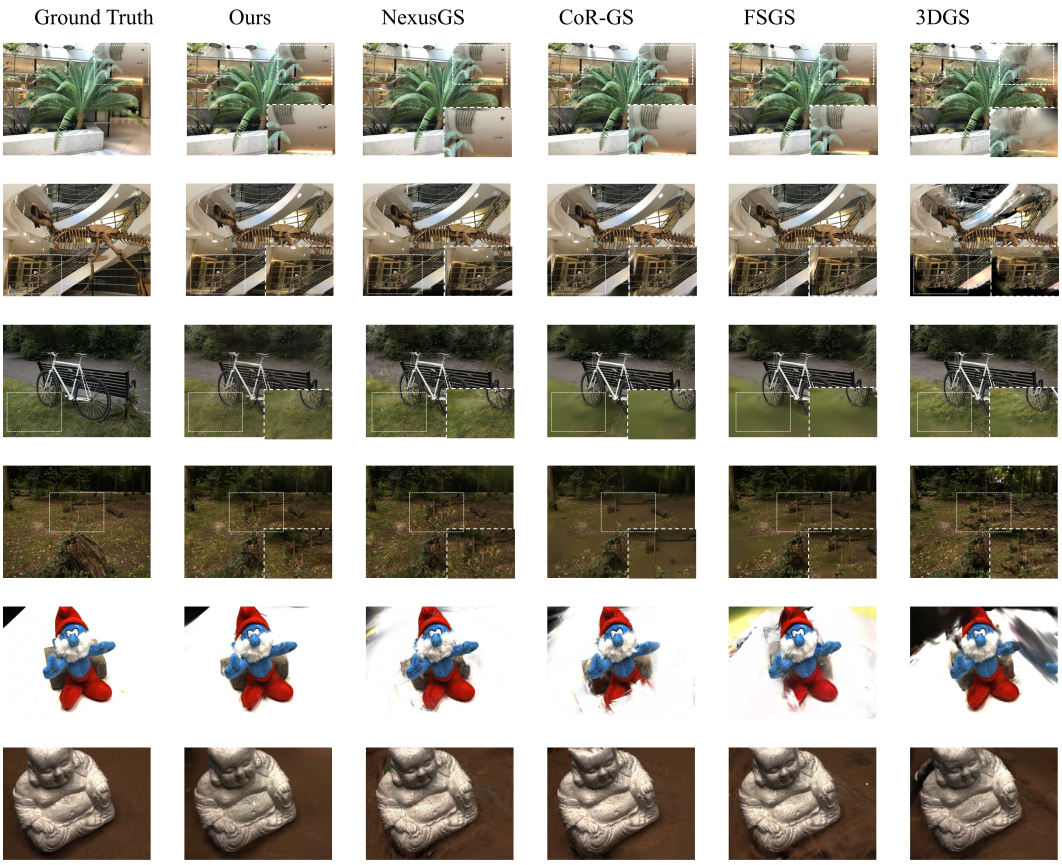

Figure 2: Qualitative comparison on LLFF, DTU, and Mip-NeRF 360. Enlarged regions highlight rendering artifacts, object boundaries, and local details.

Table 2: Ablations of spatial supervision (S), provisional refinement (R), and spatial weighting on LLFF with three input views. Bold indicates the best result in each column.

Configuration	PSNR↑	SSIM↑	LPIPS↓
Base	20.99	0.730	0.178
Base+S	21.50	0.764	0.163
Base+R	21.51	0.745	0.177
Full (uniform weights)	21.54	0.756	0.162
Full	**21.68**	**0.766**	**0.157**

4.2 Sparse-View Reconstruction Results

Table 1 shows that EviSR-GS achieves the lowest reported LPIPS among the listed methods on all three benchmarks, as well as the highest PSNR and SSIM on LLFF and Mip-NeRF 360.

Figure 2 presents visual comparisons across the three benchmarks. In the highlighted regions, EviSR-GS produces fewer spike-like artifacts and less local blur, while preserving clearer object boundaries and finer details.

4.3 Ablation Studies and Analysis

Q1. How do spatial supervision and refinement contribute?
We evaluate spatial supervision (S) and provisional refinement (R) on LLFF with three input views. Base disables both branches, Base+S and Base+R enable one branch each, and Full enables both. Without S, training retains unit-weight RGB squared error and omits patch perceptual supervision. Without R, candidate insertion and post-maturation assessment are disabled. Ordinary densification and pruning remain enabled in all configurations.

To isolate spatial weighting, Full (uniform weights) replaces input-dependent pixel and patch weights with scene-level constants computed before training and held fixed. Loss terms and coefficients, patch construction and sampling, regional masks, patch retention, and refinement settings remain unchanged; constant computation is detailed in Appendix A.1.

Table 2 shows that both branches improve reconstruction over Base. Spatial supervision and refinement provide comparable PSNR gains of 0.51 and 0.52 dB, respectively, while spatial supervision yields larger SSIM and LPIPS improvements. Full achieves the best result across all three metrics: relative to the better single-branch result for each metric, it improves PSNR by 0.17 dB and SSIM by 0.002, and reduces LPIPS by 0.006. These results support the complementary contributions of the two branches.

Uniform weighting improves SSIM and LPIPS over Base+R, showing that the additional appearance constraints are useful even with constant weights. Full further improves PSNR by 0.14 dB and SSIM by 0.010, and reduces LPIPS by 0.005 relative to uniform weighting. This comparison demonstrates the additional benefit of adapting pixel and patch supervision weights to local input consistency under otherwise identical settings.

Q2. Does EviSR-GS remain effective with more input views?
We evaluate LLFF reconstruction with three, six, and nine input views to examine whether EviSR-GS remains effective as observation density increases.

Table 3 shows that EviSR-GS achieves the highest PSNR and lowest LPIPS at all three view counts among the listed methods, together with the highest SSIM at three and nine views. With six views, it improves PSNR by 0.22 dB over D^2GS and reduces LPIPS by 0.004 relative to DropAnSH-GS, the strongest baselines for the respective metrics. With nine views, it exceeds DropAnSH-GS by 0.05 dB in PSNR and 0.003 in SSIM, while reducing LPIPS by 0.002.

The results suggest that the value of evidence-guided optimization is not confined to the sparsest setting. As additional views strengthen reconstruction constraints, EviSR-GS continues to benefit reconstruction through its local supervision and capacity-allocation design.

Table 3: Reconstruction on LLFF with three, six, and nine input views. Baseline results are taken from the corresponding papers. Bold and underlined values indicate the best and second-best results in each column. '—' indicates results not included in this comparison.

	3 views			6 views			9 views		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-based methods									
Mip-NeRF	14.62	0.351	0.495	22.91	0.756	0.213	24.88	0.826	0.170
DietNeRF	14.94	0.370	0.496	21.75	0.717	0.248	24.28	0.801	0.183
RegNeRF	19.08	0.587	0.336	23.10	0.760	0.206	24.86	0.820	0.161
FreeNeRF	19.63	0.612	0.308	23.73	0.779	0.195	25.13	0.827	0.160
3DGS-based methods									
3DGS	18.54	0.588	0.272	23.74	0.807	0.162	25.44	0.860	0.096
DNGaussian	19.12	0.591	0.294	22.18	0.755	0.198	23.17	0.788	0.180
FSGS	20.43	0.682	0.248	24.09	0.823	0.145	25.31	0.860	0.122
CoR-GS	20.45	0.712	0.196	24.34	0.831	0.122	25.94	0.872	<u>0.088</u>
DropoutGS	19.39	0.632	0.279	24.02	0.816	0.144	25.13	0.869	0.099
DropGaussian	20.33	0.709	0.201	24.58	0.830	0.125	25.85	0.864	0.093
DropAnSH-GS	20.68	0.724	0.194	24.76	**0.837**	<u>0.116</u>	<u>26.24</u>	<u>0.875</u>	<u>0.088</u>
UGOT	19.77	0.625	0.273	—	—	—	—	—	—
D^2GS	<u>21.35</u>	<u>0.746</u>	0.179	<u>24.84</u>	<u>0.834</u>	0.122	—	—	—
NexusGS	21.07	0.738	<u>0.177</u>	24.52	0.814	0.130	25.32	0.834	0.123
EviSR-GS	**21.68**	**0.766**	**0.157**	**25.06**	0.830	**0.112**	**26.29**	**0.878**	**0.086**

Table 4: Candidate selection and post-maturation assessment on LLFF with three input views. Results are mean±standard deviation across three seeds, each averaged over eight scenes. Bold indicates the best mean in each column.

Selection	Assessment	PSNR↑	SSIM↑	LPIPS↓
Support-first	Ordinary	21.4152 ± 0.0361	0.7542 ± 0.0003	0.1629 ± 0.0003
Support-first	Activity-soft	21.4510 ± 0.0018	$\mathbf{0.7584} \pm 0.0003$	$\mathbf{0.1606} \pm 0.0003$
Demand-first	Ordinary	21.4333 ± 0.0056	0.7535 ± 0.0004	0.1646 ± 0.0003
Demand-first	Activity-soft	$\mathbf{21.4936} \pm 0.0341$	0.7582 ± 0.0006	0.1614 ± 0.0004

Q3. How do candidate selection and activity assessment interact?
We compare Support-first and Demand-first selection, each combined with Ordinary pruning or Activity-soft post-maturation attenuation. Support-first prioritizes cross-view depth support, whereas Demand-first uses the demand-guided rule defined in Section 3.3. All variants share the same frozen candidate pools, spatial supervision, per-event insertion budget, initial opacity, and maturation schedule; only candidate selection and post-maturation treatment are varied. Actual insertion counts follow the candidate-shortage rule described in Appendix B.2.

Table 4 shows that Activity-soft improves all three metrics under both selection strategies, with PSNR gains of 0.0358 dB for Support-first and 0.0603 dB for Demand-first. The larger gain under Demand-first suggests that demand-guided selection provides a more favorable candidate set for post-maturation activity assessment in this comparison, while Support-first achieves slightly better final SSIM and LPIPS. These results support the complementary roles of the two stages: candidate selection allocates provisional additions, and activity assessment uses current opacity and accumulated rendering visibility to guide the treatment of mature additions.

4.4 Refinement Quality and Representation Size

Figure 3 compares training-view reconstructions with and without refinement. Full reduces local blur in the LLFF example and protruding boundary artifacts in the DTU example. The demand maps show the spatial priorities for allocation.

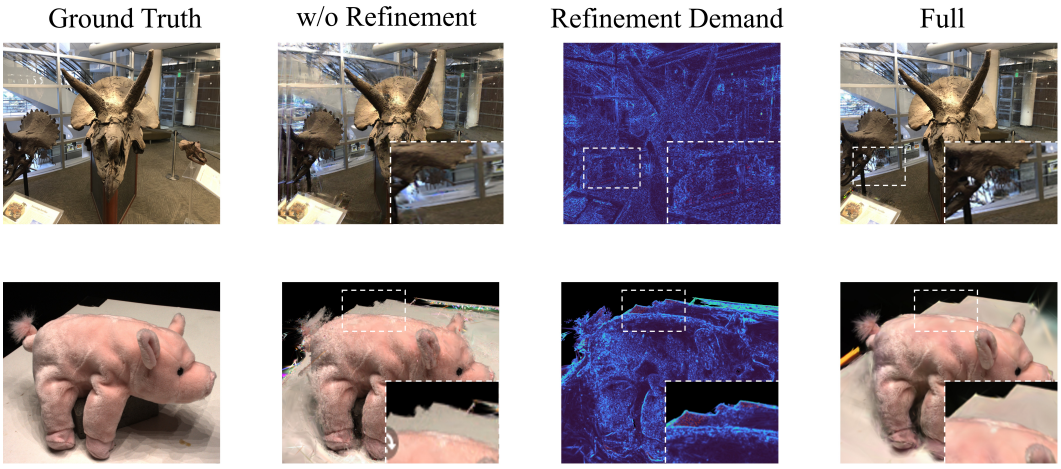

Figure 3: Training-view examples from LLFF (top) and DTU (bottom), each with three input views. Columns show the reference, reconstruction without refinement, refinement demand, and Full. Dashed boxes mark enlarged regions.

Table 5: Allocation and retention statistics for the branch-ablation experiment on LLFF with three input views. Counts are arithmetic means over eight scenes from this experiment. Direct insertions exclude clone/split descendants. Full uses uncertainty-triggered scheduling. The active ratio is the final active seed-tagged count divided by the number of direct insertions.

Configuration	Direct insertions	Mean demand	Final Gaussians	Active ratio
Single-view insertion	47,104	– – –	530,723	75.48%
Cross-view insertion	46,260	0.8162	515,319	52.12%
Full	49,501	0.8208	469,068	35.61%

Table 5 compares three refinement configurations. Single-view insertion adds candidates without cross-view validation, whereas cross-view insertion applies evidence-based selection, initialization, and suppression. Full retains these operations and additionally enables spatial supervision and uncertainty-triggered scheduling. Its larger cumulative insertion count results from a scene-dependent extension of the insertion window described in Appendix A.3, while remaining within the per-event budget. Despite inserting more provisional Gaussians during training, Full finishes with an 11.6% smaller final Gaussian population than single-view insertion, showing that it uses additional provisional capacity while producing a more compact final representation.

5 Conclusion

We presented EviSR-GS, an evidence-guided framework that combines spatially adaptive supervision with provisional Gaussian refinement for sparse-view reconstruction. Fixed input cues and the evolving model state guide local image constraints and budgeted Gaussian allocation, while activity assessed after maturation guides the attenuation of additions. Experiments on LLFF, DTU, and Mip-NeRF 360 demonstrate competitive novel-view synthesis performance. LLFF ablations support the contributions of both branches and the benefit of spatially varying loss weights over fixed scene-level weights. Controlled LLFF comparisons with equal maturation periods show that adding activity-based attenuation to ordinary pruning improves mean reconstruction metrics under both selection strategies. The larger mean gains under demand-first selection suggest the value of considering capacity allocation and subsequent activity assessment jointly.

Limitations and future work. EviSR-GS relies on external depth priors, whose errors can affect both supervision weights and Gaussian placement. Limited valid cross-view comparisons can also weaken the evidence available for allocation. Opacity and accumulated rendering visibility characterize training activity but do not establish geometric correctness. Patch perceptual supervision and cross-view assessment introduce additional computation. Future work will investigate more efficient assessment and reduced reliance on external depth priors.

AI Assistance Disclosure.

During the preparation of this manuscript, the authors used ChatGPT as an assistance tool for language polishing, organization of technical descriptions, drafting figure captions, identifying and organizing potentially relevant references, and inspecting and summarizing implementation details. The core research idea, problem formulation, methodological contributions, and scientific decisions were developed by the authors independently. All references, code, experiments, numerical results, interpretations, and final manuscript contents were independently checked and verified by the authors. No AI system was used to generate experimental data or replace author verification.

References

Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2): 153–168, 2016.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Shuangkang Fang et al. Dropping anchor and spherical harmonics for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5885–5894, 2021.

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.

Hyeseong Kim et al. Twings: Thin plate splines warp-aligned initialization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jiashu Li et al. Herogs: Hierarchical guidance for robust 3d gaussian splatting under sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 38(4):1–14, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5480–5490, 2022.

Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In *European Conference on Computer Vision*, pp. 19–37, 2024.

Avinash Paliwal, Xilong Zhou, Wei Ye, Jinhui Xiong, Rakesh Ranjan, and Nima Khademi Kalantari. Ri3d: Few-shot gaussian splatting with repair and inpainting diffusion priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25094–25103, 2025.

Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21600–21609, 2025.

Nagabhushan Somraj and Rajiv Soundararajan. Vip-nerf: Visibility prior for sparse input neural radiance fields. In *ACM SIGGRAPH*, pp. 1–11, 2023.

Meixi Song, Xin Lin, Dizhe Zhang, Haodong Li, Xiangtai Li, Bo Du, and Lu Qi. D^2GS: Depth-and-density guided gaussian splatting for stable and accurate sparse-view reconstruction. *arXiv preprint arXiv:2510.08566*, 2025. URL https://arxiv.org/abs/2510.08566.

Wei Sun, Qi Zhang, Yanzhao Zhou, Qixiang Ye, Jianbin Jiao, and Yuan Li. Uncertainty-guided optimal transport in depth supervised sparse-view 3D gaussian. *arXiv preprint arXiv:2405.19657*, 2024. URL https://arxiv.org/abs/2405.19657.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9065–9076, 2023.

Qisen Wang, Yifan Zhao, Jiawei Ma, and Jia Li. How to use diffusion priors under sparse views? In *Advances in Neural Information Processing Systems*, volume 37, pp. 30394–30424, 2024.

Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yanning Zhang. Sparse2dgs: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11307–11316, 2025.

Kaiqiang Xiong, Rui Peng, Jiahao Wu, Zhanke Wang, Jie Liang, Xiaoyun Zheng, Feng Gao, and Ronggang Wang. Intrinsic geometry-appearance consistency optimization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40918–40928, 2026.

Yexing Xu, Longguang Wang, Minglin Chen, Sheng Ao, Li Li, and Yulan Guo. Dropoutgs: Dropping out gaussians for better sparse-view rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 701–710, 2025.

Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8254–8263, 2023.

R. Yin, V. Yugay, Y. Li, et al. Fewviewgs: Gaussian splatting with few view matching and multi-stage training. *Advances in Neural Information Processing Systems*, 37:127204–127225, 2024.

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

Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21424–21433, 2024.

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

Chen Zhao, Xuan Wang, Tong Zhang, Saqib Javed, and Mathieu Salzmann. Self-ensembling gaussian splatting for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4940–4950, 2025.

Yulong Zheng, Zicheng Jiang, Shengfeng He, Yandu Sun, Junyu Dong, Huaidong Zhang, and Yong Du. Nexusgs: Sparse view synthesis with epipolar depth priors in 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26800–26809, 2025.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, 2024.

APPENDIX

The appendix provides implementation details in Appendix A and supplementary experimental protocols and results in Appendix B.

A IMPLEMENTATION DETAILS

A.1 SPATIAL SUPERVISION

Input indicators. For a zero-padded 3×3 neighborhood $\mathcal{N}_3(p)$, let $c_i(p) = \max(1, \sum_{q \in \mathcal{N}_3(p)} V_i(q))$. We compute the local depth statistic as

$$\begin{aligned} \mu_i(p) &= \frac{\sum_{q \in \mathcal{N}_3(p)} V_i(q) D_i(q)}{c_i(p)}, \\ v_i(p) &= \frac{\sum_{q \in \mathcal{N}_3(p)} V_i(q) (D_i(q) - \mu_i(q))^2}{c_i(p)}. \end{aligned} \tag{9}$$

Each neighbor q contributes a squared residual relative to its own local mean $\mu_i(q)$. We obtain $\widetilde{v}_i$ by dividing by the maximum over valid pixels plus 10^{-8}, clipping to $[0, 1]$, and masking invalid pixels. The statistic is set to zero when this maximum is at most 10^{-8}.

Reference RGB is averaged across channels, and valid depth is min–max normalized with 10^{-6} added to its range. Both inputs are masked by V_i before applying reflection-padded 3×3 Sobel filters. With gradient magnitude $g = \sqrt{g_x^2 + g_y^2 + 10^{-6}}$, the mismatch is

$$m_i = V_i \left[1 - \exp\left(- \frac{|g_i^I - g_i^D|}{g_i^I + g_i^D + 10^{-6}} \right) \right]. \tag{10}$$

Equation 3 combines these statistics with $\beta = 0.6$ to obtain U_i and $Q_i = 1 - U_i$. Both indicators use fixed reference RGB and depth inputs. They are cached per view and refreshed after at least five global iterations. Invalid-depth pixels use $U_i = 0, Q_i = 1$.

Regions and patches. Threshold rendered opacity at $\tau_c = 0.1$ and dilate the resulting coverage mask with a 3×3 kernel. Region s contains covered pixels with $U_i < \tau_u = 0.3$, including covered invalid-depth pixels. Region w is its complement within the entire training image. If s is empty, use the whole image for s and an empty w.

The regional perceptual calculations use patch sides of 48 and 64, with maximum image sides of 256 and 512 for s and w, respectively. Resizing preserves aspect ratio, using bilinear interpolation for RGB and nearest-neighbor interpolation for masks and indicators. We mask both RGB images and U_i, then enumerate non-overlapping grid patches, retaining partial boundary patches at their actual size. Masked RGB patches are passed directly to the frozen repository-local `lpipsPyTorch` VGG-LPIPS implementation (version 0.1). Reference RGB images are stored in $[0, 1]$, whereas rendered patches are passed in the renderer's native RGB range. Neither `normalize=True` nor an explicit $2x - 1$ transformation is used. The wrapper applies its fixed channel-wise z-score normalization internally.

Partial boundary patches are passed directly to the LPIPS wrapper without padding or resizing. The rendered patch, reference patch, and corresponding mask use the same spatial crop.

Let $\overline{U}_{i,r}(P)$ denote the mean masked indicator over the entire patch, including masked-out pixels. Retain patches with $\overline{U}_{i,r}(P) \geq 10^{-6}$ in $\mathcal{P}_{i,r}$. The regional loss is

$$\mathcal{L}^{(i)}_{\text{perc},r} = \frac{1}{|\mathcal{P}_{i,r}|} \sum_{P \in \mathcal{P}_{i,r}} \overline{U}_{i,r}(P)\, \ell_{\text{LPIPS}}\big(X_{i,r}[P], Y_{i,r}[P]\big), \tag{11}$$

where X and Y are masked rendered and reference RGB. An empty patch set contributes zero. Each region is reduced by its retained patch count, without normalizing by the weight sum or imposing a fixed patch-count ratio between regions.

Objective and fallback. The complete Full spatial objective is

$$\mathcal{L}^{(i)}_{\text{spatial}} = 2\mathcal{L}^{(i)}_{\text{pix}} + 0.08\, g_t\big(0.5\mathcal{L}^{(i)}_{\text{perc},s} + \mathcal{L}^{(i)}_{\text{perc},w}\big), \tag{12}$$

where $g_t = 5$ when $t > 500$ and $t \bmod 5 = 0$, and zero otherwise. The pixel term is RGB MSE weighted by Q_i at valid-depth pixels and by one elsewhere, divided by $3|\Omega_i|$. No L1, D-SSIM, or depth loss is added to this objective. If a view has no depth prior or its validity mask is empty, the perceptual branch is skipped and pixel weights are one. The DTU background penalty is optimized in a separate step.

Uniform-weight control. We compute the scene constant $\overline{U}$ once by averaging the original indicator over all valid finite-depth pixels across training views, and set $\overline{Q} = 1 - \overline{U}$. Valid pixels use $\overline{Q}$, invalid pixels retain unit weight, and retained patches use $\overline{U}$. The original indicators still determine regional masks, patch retention, and adaptive scheduling. Loss coefficients and refinement settings match Full. This control requires depth priors for all training views and valid finite-depth pixels in the scene.

A.2 Provisional Gaussian Refinement

Candidate construction and cross-view statistics. Among pixels satisfying Eq. 7, we retain at most $4B$ candidates with the highest s_0 scores and backproject them using the external depth prior and camera calibration. Cross-view comparisons exclude the source view. When the number of available depth-prior views exceeds the configured limit, views are selected by a fixed stride through their stored order.

Let $\mathcal{V}_j$ contain the valid comparisons for candidate j. A comparison requires a finite in-image projection, camera-space depth $z_{jk} > 0.01$, and a valid finite positive prior depth sampled by nearest neighbor. We compute

$$\begin{aligned} d_{jk} &= \frac{|z_{jk} - D_k(p_{jk})|}{|D_k(p_{jk})| + \epsilon}, \\ n_j &= \sum_{k \in \mathcal{V}_j} \mathbb{I}[d_{jk} \leq \tau_d], \\ e_j &= \text{clip}\left(\frac{\sum_{k \in \mathcal{V}_j} d_{jk}}{(|\mathcal{V}_j| + \epsilon)(\tau_d + \epsilon)}, 0, 1\right), \end{aligned} \tag{13}$$

where $\epsilon = 10^{-8}$. For $\mathcal{V}_j = \varnothing$, we set $n_j = 0$ and $e_j = 1$. No additional occlusion mask is applied.

Selection and initialization. Candidates passing the demand threshold are selected using the ranking in Section 3.3. If fewer than B candidates qualify, all are selected without replenishment. The selected indices are passed directly to insertion, without an additional confidence-threshold test. Each source pixel contributes at most one candidate per insertion event.

Source RGB initializes the zeroth-order spherical harmonic coefficients; higher orders are zero. Rotations are initialized to identity, and isotropic scales are derived from nearest-neighbor distances among existing and newly inserted points. The full method initializes opacity as

$$\alpha_j^{\text{init}} = 0.05 + 0.15\rho_j. \tag{14}$$

Lineage and activity bookkeeping. Each direct insertion receives a lineage identifier and birth iteration b_j, both inherited through cloning and splitting. Maturation age is therefore measured from the original insertion. Rendering visibility is counted per Gaussian instance: direct additions and clone/split children start at zero, and each positive renderer visibility flag increments the count by one. Pruning applies the same surviving-point mask to parameters, lineage identifiers, birth times, and visibility counts. Immature tagged lineages are excluded from ordinary pruning; splitting may still replace a parent with children of the same lineage. Ordinary opacity resets also apply to additions and preserve their identifiers, birth times, and activity counts.

Ordinary pruning also applies the surviving-point mask to Adam's first and second moments. Parameter expansion preserves existing moments and appends zero moments for direct insertions and clone/split children, while retaining the existing Adam step counters.

Post-maturation attenuation. At each scheduled assessment, current opacity and accumulated rendering visibility determine the set of tagged instances to attenuate:

$$\mathcal{M}_t = \left\{ j : \; j \text{ is seed-tagged}, \; t - b_j \geq T_m, \; \alpha_j^t < \tau_o, \; \nu_j^t < \tau_v \right\}. \tag{15}$$

For $j \in \mathcal{M}_t$, we apply

$$\alpha_j^t \leftarrow \mathrm{clip}\big(\kappa \alpha_j^t, 10^{-6}, 1 - 10^{-6}\big), \tag{16}$$

with $\tau_o = 0.03$, $\tau_v = 10$, and $\kappa = 0.1$. The update is stored through the inverse sigmoid, without an additional visibility or texture protection gate. A surviving instance may be attenuated at multiple assessments. Attenuation does not delete parameters; ordinary pruning continues after maturation. The training schedule is specified in Appendix A.3.

A.3 Training Procedure and Hyperparameters

Training procedure. The EviSR-GS implementation builds on NexusGS (Zheng et al., 2025). We train for $T = 30{,}000$ iterations using fixed external depth priors. At each iteration, one training view is sampled without replacement; the view list is replenished once exhausted. Optimizer settings, parameter-group learning rates, ordinary density-control parameters, and opacity-reset settings are provided in Appendix B.1.

Algorithm 1 summarizes the execution order. Rendering provides the RGB image, depth, opacity, and per-instance visibility flags. After accumulating visibility and retrieving the input indicators, we compute the spatial objective described in Appendix A.1. Ordinary density control, candidate insertion, and post-maturation assessment follow backpropagation and precede the main optimizer step. Newly inserted Gaussians participate in the objective from subsequent iterations. Learning-rate updates and scheduled opacity resets follow the main optimizer step.

Input-dependent scheduling. The initial insertion window spans iterations 500–$5{,}000$, with insertion events every 100 iterations and a maturation period of 800 iterations. Before iteration 500, we collect the mean inconsistency over valid-depth pixels in each sampled training view. At iteration 499, if the average of these values is at least 0.2, the insertion start and end are changed to $2{,}000$ and $8{,}000$, respectively, and the maturation period is increased to $3{,}000$. Otherwise, the initial settings remain unchanged. This rule is applied separately to each scene.

DTU background update. For DTU, a separate optimizer step precedes the main rendering and minimizes the mean squared rendered opacity over a background mask, with coefficient one. The mask identifies pixels whose maximum reference RGB channel is below $30/255$, or $15/255$ for scan110, and retains locations satisfying the background condition across the preceding 49 rows in the same column. Reference RGB values within this mask are set to zero. Gradients from this separate update are cleared before the main rendering and reconstruction update.

Hyperparameters. Table 6 lists the supervision and refinement hyperparameters shared across the three benchmarks. The insertion schedule and maturation period are initial values; the input-dependent adjustment described above is applied separately to each scene. Settings specific to the controlled comparisons are provided in Appendix B.2.

Algorithm 1 Evidence-guided supervision and refinement

Require: Training images and cameras, fixed depth priors, initial Gaussians, and training settings
Initialize optimizer, lineage records, and activity statistics
for $t = 1, \ldots, T$ **do**
  Sample a training view
  Apply the separate DTU background update when applicable
  Render RGB, depth, opacity, and visibility
  Accumulate per-instance rendering visibility
  Retrieve input indicators and update the input-dependent schedule when applicable
  Compute and backpropagate the spatial objective from Appendix A.1
  Update density-control statistics and apply scheduled densification and pruning
  Generate, select, and insert candidates when scheduled
  Assess mature tagged instances and apply scheduled opacity attenuation
  if $t < T$ **then**
    Apply the main Adam update and clear gradients
  end if
  Update learning rates and apply scheduled opacity reset
end for
return the Gaussian representation

B Supplementary Experiments

B.1 Experimental Settings and Evaluation Protocols

Input depth and preprocessing. Our external depth maps are obtained from the NexusGS (Zheng et al., 2025) input preprocessing pipeline. We use the corresponding depth maps, camera parameters, validity masks, and scene normalization without additional depth estimation or depth supervision. The depth representation follows the same coordinate convention as the NexusGS initialization.

Datasets and evaluation. We evaluate on LLFF (Mildenhall et al., 2019), DTU (Aanæs et al., 2016), and Mip-NeRF 360 (Barron et al., 2022). The main comparison uses three training views for LLFF and DTU and 24 for Mip-NeRF 360. Additional LLFF experiments use six and nine views.

The LLFF experiments cover eight scenes: fern, flower, fortress, horns, leaves, orchids, room, and trex, using the `images_8` image set. Reconstruction quality is evaluated on test views using PSNR, SSIM, and LPIPS (Zhang et al., 2018). For evaluation, rendered and reference PNG images are converted to tensors in $[0, 1]$ using `to_tensor` and passed directly to the same repository-local `lpipsPyTorch` VGG-LPIPS implementation (version 0.1). No additional `normalize=True` flag or explicit $2x - 1$ transformation is used; the wrapper applies its fixed channel-wise z-score normalization internally. The local refinement examples show renderings from training viewpoints.

Baseline results. Baseline values are taken from the corresponding papers under their reported evaluation protocols; our results are obtained from our own runs. The six-view LLFF and 24-view Mip-NeRF 360 results for D^{2}GS (Song et al., 2025) are taken from Appendix E, Table 8 of the source paper. Because the external values are reproduced from their source papers, their exact evaluation protocols may differ; Table 1 should therefore be read as a comparison of reported results rather than a strictly controlled cross-method re-evaluation.

Optimization. We use the same training hyperparameters across LLFF, DTU, and Mip-NeRF 360. Adam uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon_{\text{Adam}} = 10^{-15}$. Positions, zeroth- and higher-order spherical harmonic coefficients, opacity, scale, and rotation are optimized in separate parameter groups. Table 7 lists the effective optimizer learning rates. The position learning rate follows an exponential schedule over $30{,}000$ iterations, multiplied by the scene-specific spatial scale s_{scene}. The other listed learning rates remain constant.

Ordinary density control. These updates are skipped at evaluation iterations $5{,}000$, $10{,}000$, $15{,}000$, $20{,}000$, and $25{,}000$. The densification gradient threshold is 5×10^{-4}, and the opacity-pruning threshold is 0.005. Gaussians with a training-view contribution score or texture score of

Table 6: Supervision and refinement hyperparameters shared across LLFF, DTU, and Mip-NeRF 360. Schedule values are specified before input-dependent adjustment.

Parameter	Value
Training	
Training iterations T	30,000
Training views per iteration	1
Spatial supervision	
Indicator mixture coefficient β	0.6
Pixel / patch loss coefficients	2 / 0.08
Regional coefficients λ_s / λ_w	0.5 / 1
Coverage threshold τ_c	0.1
Inconsistency threshold τ_u	0.3
Coverage-mask dilation kernel	3×3
Patch side lengths for s / w	48 / 64
Maximum image sides for s / w	256 / 512
Patch-retention threshold	10^{-6}
Perceptual evaluation interval	5, for $t > 500$
Perceptual multiplier when active	5
Candidate allocation	
Coverage trigger τ_α	0.3
Relative depth threshold τ_d	0.05
Per-event insertion budget B	1024
Maximum preselected candidates $4B$	4,096
Maximum comparison views, excluding source	2
Demand weights (w_1, w_2, w_3)	$(0.4, 0.4, 0.2)$
Demand threshold τ_ρ	0.75
Demand ranking coefficient λ_ρ	1
Initial insertion start / end	500 / 5,000
Insertion interval	100
Input-dependent scheduling threshold	0.2
Maturation and assessment	
Initial maturation period T_m	800
Assessment start / interval	2,500 / 500
Current-opacity threshold τ_o	0.03
Accumulated-visibility threshold τ_v	10
Opacity attenuation factor κ	0.1

at least 0.5 are excluded from the ordinary pruning mask. Immature tagged lineages additionally receive the protection described in Appendix A.2. The separate depth-pruning routine is disabled. Ordinary density control remains enabled when provisional refinement is disabled.

Opacity reset. Opacity resets occur at iterations $2001 + 3000k$, for nonnegative integers k within training. Each reset caps opacity at 0.05 and clears the first and second Adam moments for opacity while retaining the optimizer step counter. Lineage identifiers, birth iterations, and accumulated visibility remain unchanged. Post-maturation attenuation applies only to eligible tagged instances and retains their Adam moments.

Random seeds and aggregation. The LLFF branch and uniform-weight ablations use random seed 1. The controlled refinement comparison uses paired seeds 0, 1, and 2 across all four configurations. For this comparison, we first average each metric over the eight LLFF scenes within each run, then report the mean and standard deviation across the three seeds. Experimental controls are detailed in Appendix B.2. Dataset-specific configurations, view splits, training and evaluation scripts, and environment specifications will be released with the code.

Table 7: Optimization and ordinary density-control settings shared across LLFF, Mip-NeRF 360 and DTU. Learning rates are effective optimizer values. Supervision and provisional refinement settings are listed in Table 6.

Parameter	Value
Optimization	
Training iterations	30,000
Adam β_1 / β_2	0.9 / 0.999
Adam ϵ	10^{-15}
Initial position learning rate	$2.4 \times 10^{-4} s_{\text{scene}}$
Final position learning rate	$1.6 \times 10^{-6} s_{\text{scene}}$
Position learning-rate schedule length	30,000
SH zeroth-order learning rate	1.25×10^{-3}
SH higher-order learning rate	6.25×10^{-5}
Opacity learning rate	0.025
Scale learning rate	0.03
Rotation learning rate	0.001
Ordinary density control	
Active iteration range	$500 < t < 30{,}000$
Densification interval	100
Gradient threshold	5×10^{-4}
Opacity-pruning threshold	0.005
Clone/split scale threshold relative to scene extent	0.01
Children per split	4
Contribution / texture protection thresholds	0.5 / 0.5
Separate depth pruning	Disabled
Opacity reset	
First reset / interval	2,001 / 3,000
Opacity cap	0.05
Adam first and second moments	Cleared
Adam step counter	Retained

B.2 Controlled Comparisons

Branch ablations. We evaluate spatial supervision (S) and provisional refinement (R) on LLFF with three input views. Base disables both branches, Base+S and Base+R enable one branch each, and Full enables both.

Disabling S retains the RGB squared-error term with unit pixel weights and removes patch perceptual supervision. Disabling R removes external candidate insertion and post-maturation activity assessment. Ordinary densification and pruning remain enabled in all configurations.

Uniform-weight control. Full (uniform weights) retains the loss terms and coefficients, patch construction and sampling, regional-mask and patch-retention rules, and refinement settings of Full. Only the input-dependent pixel and patch loss weights are replaced by fixed scene-level constants.

The constants are computed from the training inputs before optimization, as specified in Appendix A.1, and remain fixed throughout training. The original input indicators continue to determine regional masks, patch retention, and adaptive scheduling. This comparison examines spatial weighting while retaining both supervision terms.

Detailed component ablations. The supplementary component study distinguishes the following operations. Observation conditioning denotes input-dependent pixel weighting. Deficiency discovery denotes candidate generation and insertion at locations triggered by low coverage or rendered–prior depth discrepancy. Cross-view assessment introduces cross-view depth statistics, and demand guidance uses these statistics together with current coverage to prioritize candidates. Adaptive scheduling denotes the input-dependent refinement schedule in Appendix A.3.

In the progressive study, demand-dependent initialization and lifecycle rules are added after candidate generation and demand-guided selection. Full additionally enables patch perceptual supervision and adaptive scheduling.

Each component-removal configuration independently starts from Full. Removing provisional refinement disables external candidate insertion and its subsequent lifecycle treatment, while retaining ordinary densification and pruning.

Removing demand guidance disables the demand threshold and ranks candidates using the source-view score s_0 instead of the demand-augmented score. When cross-view comparisons are available, candidates with agreeing depth support are retained; if none satisfy this condition, candidates with valid cross-view comparisons are retained instead.

Removing patch perceptual supervision disables the patch LPIPS loss while retaining input-dependent pixel weighting. Removing observation conditioning disables input-dependent pixel weighting while retaining pixel reconstruction and patch LPIPS supervision.

Candidate selection and post-maturation assessment. The Q3 comparison combines two candidate-selection strategies with two post-maturation treatments. Support-first disables demand thresholding and demand-augmented ranking, and prioritizes candidates with stronger cross-view depth support. Demand-first follows the demand-based priority described in Section 3.3.

For each scene and seed, the four configurations share the same frozen candidate pool and match actual insertion counts. They also share spatial supervision, fixed initial opacity, insertion and maturation schedules, and ordinary density-control settings. The controlled comparison therefore replaces the full method's demand-dependent opacity initialization with a common fixed value.

Both treatments protect additions from ordinary pruning for the same maturation period. After this period, Ordinary applies ordinary density control without additional activity-based attenuation. Activity-soft additionally attenuates mature additions with both low current opacity and low accumulated rendering visibility. Ordinary pruning remains enabled after maturation in both treatments.

The two Demand-first configurations thus provide an equal-waiting-period comparison of ordinary pruning and activity-based attenuation. The four configurations use paired random seeds 0, 1, and 2. Metric aggregation follows Appendix B.1.

B.3 Additional Ablation Results

We supplement the main ablations with a detailed component study, demand-threshold sensitivity, an analysis of selection–assessment interaction, and a comparison with opacity-only attenuation.

Progressive integration and component removal. The progressive study first examines refinement components under input-weighted pixel supervision, then adds patch perceptual supervision and adaptive scheduling to complete EviSR-GS. This order separates the refinement configurations from the final integration of both branches. The removal study independently disables each component from Full. Configuration definitions are provided in Appendix B.2.

Table 8 supports distinct but complementary contributions from supervision and refinement. Removing patch perceptual supervision produces the largest degradation among the tested removals: PSNR decreases by 0.2705 dB and LPIPS increases by 0.0210. Removing observation conditioning causes smaller changes of 0.1450 dB and 0.0040, respectively. These results support the contribution of patch appearance constraints beyond pixel reconstruction, with input-dependent pixel weighting providing an additional benefit.

Removing provisional refinement reduces PSNR by 0.1780 dB and increases LPIPS by 0.0060, supporting its contribution alongside appearance supervision. Demand guidance and adaptive scheduling have smaller removal effects, with PSNR decreasing by 0.0389 dB and 0.0198 dB, respectively.

The progressive study shows that intermediate additions do not uniformly improve all metrics. Completing EviSR-GS with patch perceptual supervision and adaptive scheduling raises PSNR from 21.4261 to 21.6820 dB and reduces LPIPS from 0.1784 to 0.1570. The complete framework achieves the best results across all three metrics in both panels.

Table 8: Detailed component analysis on LLFF with three input views. (a) Progressive integration; each intermediate row extends the preceding configuration. Full additionally enables patch perceptual supervision and adaptive scheduling. (b) Independent component removal from Full. Bold indicates the best result in each panel.

Configuration	PSNR↑	SSIM↑	LPIPS↓
(a) Progressive integration			
Observation conditioning	21.3445	0.7417	0.1780
+ Deficiency discovery	21.4169	0.7426	0.1782
+ Cross-view assessment	21.3440	0.7416	0.1776
+ Demand-guided allocation	21.4502	0.7431	0.1782
+ Initialization and lifecycle rules	21.4261	0.7429	0.1784
Full framework	**21.6820**	**0.7660**	**0.1570**
(b) Component removal			
Full framework	**21.6820**	**0.7660**	**0.1570**
w/o observation conditioning	21.5370	0.7621	0.1610
w/o provisional refinement	21.5040	0.7641	0.1630
w/o demand guidance	21.6431	0.7643	0.1580
w/o patch perceptual supervision	21.4115	0.7426	0.1780
w/o adaptive scheduling	21.6622	0.7644	0.1590

Table 9: Allocation statistics for the demand-threshold sensitivity experiment on LLFF with three input views. All reported quantities are arithmetic means over eight scenes. Direct insertions exclude clone and split descendants.

τ_ρ	PSNR↑	SSIM↑	LPIPS↓	Mean direct insertions	Mean selected demand
0.50	21.6132	0.7628	0.1602	50,740	0.8155
0.60	21.6032	0.7630	0.1607	50,755	0.8191
0.75 (default)	**21.6820**	**0.7660**	**0.1570**	50,760	0.8240
0.90	21.5934	0.7547	0.1596	2,915	0.9130

Sensitivity to the demand threshold. We evaluate four values of the demand threshold τ_ρ on LLFF with three input views. The threshold controls candidate eligibility, whereas the per-event budget limits the number of direct insertions. Table 9 reports reconstruction quality, the number of direct insertions, and the mean demand score of selected candidates.

The default threshold $\tau_\rho = 0.75$ achieves the best reported reconstruction metrics among the tested settings. Compared with $\tau_\rho = 0.50$ and 0.60, it improves PSNR by 0.0688 dB and 0.0788 dB, respectively, while also achieving higher SSIM and lower LPIPS.

The mean direct insertion counts remain nearly unchanged for $\tau_\rho = 0.50$, 0.60, and 0.75, consistent with the per-event budget being the main limiting factor in this range. Thus, changing candidate eligibility within this range does not substantially alter the number of actual direct insertions.

Increasing the threshold to $\tau_\rho = 0.90$ produces a more restrictive allocation regime. The mean number of direct insertions decreases from $50{,}760$ to $2{,}915$, corresponding to a reduction of approximately 94.3%. This reduction is accompanied by a PSNR decrease of 0.0886 dB, an SSIM decrease of 0.0113, and an LPIPS increase of 0.0026 relative to the default. The mean selected demand increases from 0.8240 to 0.9130, indicating that the retained candidates have higher demand scores.

These results reveal a tunable trade-off between reconstruction quality and provisional allocation. The setting $\tau_\rho = 0.75$ provides the best reconstruction quality among the tested configurations, whereas $\tau_\rho = 0.90$ substantially restricts provisional insertions while producing lower reconstruction quality. The more restrictive setting may nevertheless be preferable when reducing provisional allocation is prioritized. Direct insertion counts characterize the provisional allocation process only; they are not interpreted as the final Gaussian population or as a proportional reduction in computational cost. We therefore use $\tau_\rho = 0.75$ as the default threshold for the quality-oriented experiments.

Table 10: Changes in the reported means from Ordinary to Activity-soft in Table 4. Each change is computed as Activity-soft minus Ordinary. Positive PSNR and SSIM changes and negative LPIPS changes indicate improvement. The last row reports the difference between these changes (Demand-first minus Support-first).

Selection	ΔPSNR (dB)	ΔSSIM	ΔLPIPS
Support-first	$+0.0358$	$+0.0042$	-0.0023
Demand-first	$+0.0603$	$+0.0047$	-0.0032
Demand-first minus Support-first	$+0.0245$	$+0.0005$	-0.0009

Table 11: Ordinary and Opacity-soft on LLFF with three input views. Metrics are averaged over eight scenes, with one run per scene. The final row counts scenes where Opacity-soft improves the corresponding metric over Ordinary.

Treatment	PSNR↑	SSIM↑	LPIPS↓
Ordinary	21.4743	0.754021	0.163753
Opacity-soft	21.4884	0.754662	0.163110
Scenes improved	$4/8$	$6/8$	$6/8$

Selection–assessment interaction. For each scene and seed, we generated a record-only candidate pool after validity checking and old-score preselection, but before strategy-specific ranking and final eligibility filtering. Candidate geometry and diagnostic fields, including support, demand, risk, and reprojection statistics, were frozen and reused during replay without recomputation. Support-first ranked candidates by decreasing cross-view support, followed by reprojection error and candidate ID as tie-breakers, whereas Demand-first ranked candidates by decreasing frozen demand score and candidate ID. At each event, each strategy inserted the first $\min(K, N)$ candidates, where K is the fixed per-event budget and N is the number of available frozen candidates. If fewer than K candidates were available, all available candidates were inserted without replenishment. Because Support-first and Demand-first replayed the same frozen candidate pool, both strategies resulted in a mean of $46{,}260$ direct insertions per scene, averaged over eight scenes.

Table 10 summarizes the changes from Ordinary to Activity-soft under each selection strategy, computed from the reported means in Table 4. Both assessment treatments use the same maturation period, as specified in Appendix B.2.

Activity-soft improves the reported mean PSNR, SSIM, and LPIPS under both selection strategies. The improvements are larger under Demand-first: PSNR increases by 0.0603 dB versus 0.0358 dB, SSIM increases by 0.0047 versus 0.0042, and LPIPS decreases by 0.0032 versus 0.0023. With the maturation period held fixed, these mean differences suggest that demand-guided candidate selection benefits more from post-maturation activity assessment than support-first selection in this comparison.

Opacity-only attenuation. In a separate comparison, we evaluate Ordinary against Opacity-soft on the eight LLFF scenes with three input views. Opacity-soft attenuates mature additions based on low current opacity, without requiring low accumulated rendering visibility. Table 11 reports scene-averaged results from one run per scene and the number of scenes improved by Opacity-soft.

Opacity-soft yields small mean changes: PSNR increases by 0.0141 dB, SSIM increases by approximately 0.00064, and LPIPS decreases by approximately 0.00064. SSIM and LPIPS improve in six of eight scenes, while PSNR improves in four. This comparison describes the observed behavior of the opacity-only alternative and complements the joint activity assessment studied in Q3. It does not directly isolate the additional effect of the visibility criterion.

B.4 Additional Qualitative Results

Figure 4 illustrates local reconstruction differences at a held-out test viewpoint. Observation support varies across the scene, providing spatial context for the highlighted regions. The baseline exhibits blurred skeletal details in region A and less distinct railing structures in region B. The full framework

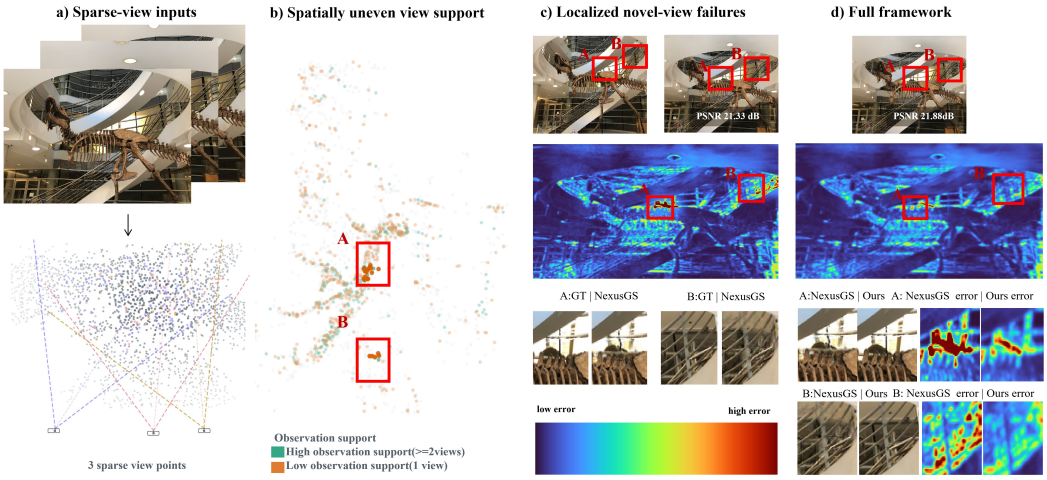

Figure 4: Observation support and test-view reconstruction on the LLFF trex scene with three training views. (a) Sparse input views and camera configuration. (b) Observation-support visualization, distinguishing points supported by one training view from those supported by at least two views. (c) Reference images, NexusGS reconstructions, and corresponding pixel-wise RGB error maps with enlarged regions. (d) Reconstructions from EviSR-GS with corresponding error maps and local comparisons. Regions A and B highlight skeletal structures and background railings, respectively. Error maps are computed with respect to the reference images using the same color scale across all comparisons; lower values indicate smaller reconstruction errors. The displayed test-view PSNR improves from 21.33 dB for NexusGS to 21.88 dB for EviSR-GS.

produces clearer local structures, with reduced errors in the corresponding visualizations. PSNR increases by 0.55 dB for the displayed test view.

These examples complement the aggregate metrics by showing where the full framework improves reconstruction beyond the training viewpoints. The component contributions are examined separately through the controlled comparisons in Q1 and Q3.