# RETROSPLAT: RETROACTIVE REVISION OF STREAMING GAUSSIAN MAPS WITH REPLAY EVIDENCE

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

## ABSTRACT

A streaming Gaussian map begins with partial observations and receives new evidence as the camera moves. RetroSplat uses this evidence to revise Gaussians that have already been published. Each Gaussian belongs to a persistent voxel cell that retains per-frame feature statistics and accumulates evidence for revision. Later depths vote for surface support or free space, and renders into seen cameras yield opacity *replay gradients*. After every six-frame packet, a decoder trained across scenes reads this evidence and revises committed Gaussians without a per-scene optimizer; a causal encoder publishes new cells between revisions. On four RealEstate10K and ACID protocols, the final maps beat the append-only ReCoSplat by 1.0–2.0 dB PSNR. Training without replay costs 1.6–2.8 dB, most on the earliest-observed RealEstate10K content. Doubling the RealEstate10K views from 64 to 128 over the same span changes final-map PSNR by $+0.01$ dB, versus $-0.91$ dB for ReCoSplat. An optional post-stream refit adds 1.85–2.17 dB.

## 1 INTRODUCTION

Feed-forward Gaussian models learn across scenes to reconstruct a renderable 3D scene from images without per-scene fitting. They build on multi-view correspondence (Charatan et al., 2024; Chen et al., 2025a), multi-view image tokens (Zhang et al., 2025), pretrained depth (Xu et al., 2025), and voxel-aligned features (Wang et al., 2026b). These predictors also extend to longer sequences (Chen et al., 2025c) and unposed inputs (Ye et al., 2025; Jiang et al., 2025). In each case, reconstruction starts from one fixed set of views. Continuous capture changes that assumption: the system must publish a usable map before it has seen the full scene. Later images then test predictions that are already part of the map.

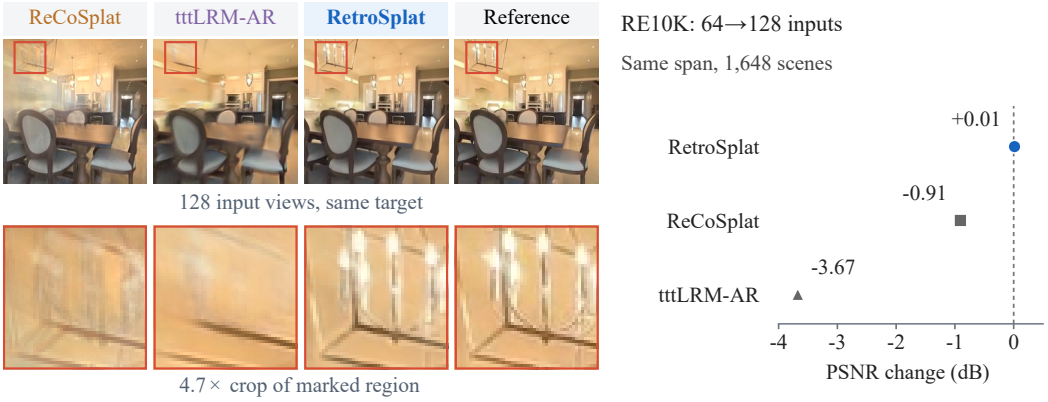



Figure 1: **RetroSplat keeps the thin lantern at 128 inputs.** Left: final maps of one held-out RE10K target (no refit) with 4.7× crops. Right: cohort-mean PSNR change when the same RE10K span is sampled with 128 instead of 64 inputs. Selection in Section G; absolute scores in Table 1.

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.

Among streaming feed-forward systems, ReCoSplat appends each new prediction to the scene (Cheng et al., 2026), and tttLRM-AR re-decodes all Gaussians from a latent state updated per chunk (Wang et al., 2026a). Neither uses the published map's errors on earlier views to correct committed Gaussians. To see where an accumulated map goes wrong, we analyze a variant trained without replay (rendering feedback from earlier views). On 64-view RealEstate10K (RE10K) streams, $32\%$ of opaque pixels in held-out views are composited on average from Gaussians created at least two six-frame groups (packets) after the target's nearest context view. These pixels carry $52\%$ of the squared error. In a separate diagnostic, fitting the final map's opacities with geometry and color fixed raises held-out PSNR by a median $3.4\,\mathrm{dB}$. The fit uses sixteen extra frames, disjoint from context views and targets (Section D). This recoverable error motivates opacity gradients as evidence for map revision.

In RetroSplat, persistent voxel cells store the state from which the Gaussian map is decoded. Each cell retains the feature statistics behind its Gaussian and accumulates evidence for revising it in place. Depths predicted from later frames vote for surface support or free space. Replay renders the map into cameras already seen and records each cell's opacity gradient. Geometric votes require the cell's anchor to project into a later camera; replay also reaches content outside later views. A scene optimizer would descend these gradients (Kerbl et al., 2023; Matsuki et al., 2024; Keetha et al., 2024); we instead feed them to a decoder trained across scenes, as in learned gradient descent (Flynn et al., 2019; Chen et al., 2025b). The decoder revises committed Gaussians at every packet boundary, and between boundaries a causal encoder publishes new cells as frames arrive. No weights change during the stream. The optional post-stream refit optimizes opacities on the context frames (Section 3.6).

Our contributions are:

- **A revisable map state.** Voxel cells store per-frame feature sums and observation counts together with accumulated evidence, and the decoder re-predicts every cell's Gaussian at each packet boundary (Section 3.2).
- **Replay evidence for retroactive revision.** Opacity gradients from rendering the map into earlier cameras are stored beside geometric votes, and the decoder reads both to revise committed Gaussians without a per-scene optimizer (Section 3.3).
- **Causal encoding with per-frame publication.** Every cross-view encoder operator follows a causal camera graph, so each frame is encoded once and new cells appear from the third frame of each packet, before the next revision (Section 3.4).

On the four RE10K and ACID protocols, RetroSplat's final maps beat ReCoSplat by 1.0–2.0 dB PSNR, with better SSIM and LPIPS, and doubling the RE10K views over the same span changes their PSNR by $+0.01\,\mathrm{dB}$ (Fig. 1 and Section 4.2). The optional refit adds 1.85–2.17 dB (Section H). Decoding once more after the last packet, from stored evidence alone, raises PSNR by 0.33–0.57 dB. Training without replay costs 1.6–2.8 dB, and on RE10K the loss is largest for the earliest-observed targets (Section 4.3).

## 2 Related work

**Streaming reconstruction and feature reuse.** Several systems relax assumptions we keep: StreamGS and OF$^3$GS accept unposed streams (Li et al., 2025; Chen et al., 2026), and StreamSplat handles dynamic scenes (Wu et al., 2026). LongSplat updates a global map per frame (Huang et al., 2026). Map conditioning lets new predictions read renders of the accumulated scene (Cheng et al., 2026); SplatVoxel uses rendering error to reweight warped past splats before predicting them again (Wang et al., 2025b). SaLon3R consolidates new and nearby existing Gaussian latents into voxel anchors and refines them before decoding (Guo et al., 2026). Video multi-view stereo reuses features across frames (Düzçeker et al., 2021; Sayed et al., 2022). Our causal camera graph extends this reuse to every cross-view operator.

**Persistent scene memory.** Atlas, NeuralRecon and TransformerFusion maintain feature volumes (Murez et al., 2020; Sun et al., 2021; Božič et al., 2021); causal pointmap models keep recurrent or spatial memory (Wang et al., 2025a; Wang & Agapito, 2025; Wu et al., 2025; Zhuo et al., 2026); and StreamSplat-VACC caches voxel-aligned latent tokens (Song et al., 2026). OnlineX separates an ac-

tive local state from a persistent global state and fuses Gaussian latents with confidence weights (Xia et al., 2026). tttLRM-AR updates fast weights from four-view chunks using a key–value objective (Wang et al., 2026a; Zhang et al., 2026) and decodes all Gaussians for each map without rendering feedback. In RetroSplat, persistent cells hold count-normalized features plus geometric and delayed replay evidence, and a decoder revises committed Gaussians from them (Table 6 contrasts the two designs). Geometric votes follow the surface support and free-space evidence used in depth fusion and space carving (Curless & Levoy, 1996; Newcombe et al., 2011; Kutulakos & Seitz, 2000).

**Rendering feedback and learned refinement.** Gaussian Splatting fits each scene by gradient descent on a rendering loss (Kerbl et al., 2023). Neural and Gaussian SLAM systems optimize a persistent scene model and replay keyframes to preserve earlier observations (Matsuki et al., 2024; Keetha et al., 2024; Sucar et al., 2021). DeepView, G3R and ReSplat learn iterative updates from rendering gradients or residuals over fixed view sets (Flynn et al., 2019; Chen et al., 2025b; Xu et al., 2026). We instead accumulate this feedback in persistent cells as the stream grows. Our optional post-stream refit is closest to methods that optimize feed-forward predictions after inference, such as InstantSplat (Fan et al., 2024), and to learned per-Gaussian optimizers such as Diff3R and Learn2Splat (Liu et al., 2026; Pearl et al., 2026).

# 3 METHOD

## 3.1 FROM INCOMING FRAMES TO MAP OBSERVATIONS

RetroSplat turns a stream of calibrated, posed RGB images of a static scene into one Gaussian map shared by all target cameras. Frames arrive in six-frame *packets*; completing packet $t$ defines *boundary* $t$. At each boundary, the packet's observations are written into persistent voxel cells, which *commits* them to the map; a decoder then predicts the Gaussian of every cell again to form the *boundary map* $G_t$; and replaying $G_t$ into cameras already seen collects evidence for boundary $t+1$ (Fig. 2). Between boundaries, new cells are published frame by frame (Section 3.4). After the last packet, a final decode gives the *final map* $G_{\mathrm{fin}}$, which we evaluate.

The encoder predicts depths and pixel features with a multi-view transformer, plane-sweep matching and a depth network (Xu et al., 2025; Wang et al., 2026b). To give each new observation the context of the existing map, we render the preceding boundary map $G_{t-1}$ into its camera and inject luminance, inverse depth and opacity through zero-initialized convolutions (Cheng et al., 2026; Song et al., 2026). We then unproject the predicted samples and pool them into fine and coarse voxel cells. Section 3.4 restricts these computations to a causal camera graph.

## 3.2 REVISABLE MAP STATE

At both voxel scales, each cell stores a feature sum and an observation count for every source frame that contributed to it. The decoder reads these records to predict the cell's Gaussian at every boundary. A fine cell's *anchor* is the mean position of its samples. Its Gaussian has a mean within $1.5$ fine-cell widths of the anchor along each axis, a covariance, degree-two spherical-harmonic colors and an opacity. The cell's mean features, concatenated with those of the coarse cell containing the anchor, form the decoder input $x_{c,t}$; the coarse features add wider spatial context.

Let $C_t$ be the cells created up to boundary $t$ and $X_t = \{x_{c,t}\}_{c \in C_t}$ their decoder inputs. Each cell also stores geometric evidence $E_t^g$, replay evidence $E_t^p$ and age features $A_t$ derived from its creation packet (Table 9 lists every field).

## 3.3 REVISING THE MAP WITH LATER EVIDENCE

Consider a surface first predicted slightly in front of its true depth. Later packets that see it cast free-space votes against the old anchor, and their coverage opens a gate for the decoder to re-predict every attribute of the cell. If no later camera sees the region, renders into earlier cameras still disagree with their images, and the replay gradients identify the opacity direction that locally reduces the image error. Geometric votes test anchor positions against predicted surfaces; replay tests the composited image against observed RGB.

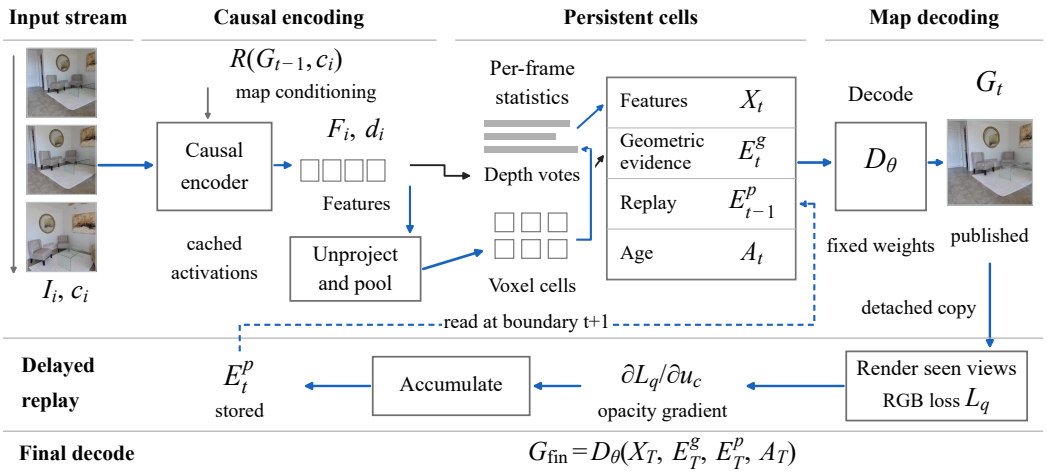



Figure 2: **Overview of RetroSplat.** Each packet updates cell features and geometric evidence; a fixed-weight decoder reads this state and the replay evidence $E_{t-1}^p$ to publish $G_t$. Replaying the detached $G_t$ into seen cameras yields $E_t^p$; after the last packet, a final decode of $E_T^p$ gives $G_{\text{fin}}$.

**Geometric evidence.** Later depth predictions test whether an existing cell agrees with the observed surface. We project its anchor into the current cameras: an anchor more than a margin in front of the surface receives a confidence-weighted free-space vote; one within the margin receives support (Curless & Levoy, 1996; Kutulakos & Seitz, 2000). The counter $v_c$ counts all projections into later cameras' frusta, including occluded anchors.

**Replay evidence.** Replay measures how each cell's opacity affects agreement with observed images. After boundary $t$, we render $G_t$ into the six current cameras and up to six earlier cameras sampled without replacement. For camera $q$, renderer $\mathcal{R}$, pixel $i$ and RGB channel $k$, we compute

$$L_q = \sum_{i,k} [\mathcal{R}(G_t^{\perp}, u, q)_{i,k} - I_{q,i,k}]^2, \qquad \gamma_{c,q} = \partial L_q / \partial u_c. \tag{1}$$

Here $G_t^{\perp}$ denotes detached Gaussian attributes, and $u$ holds temporary opacity logits initialized from the decoded map. We differentiate only with respect to $u$. A positive $\gamma_{c,q}$ means that lowering this cell's opacity would locally reduce the image error. We scale, clip and threshold these gradients (Section B), then accumulate their positive and negative masses and counts in the cell. A camera with a nonzero thresholded gradient contributes one *replay hit*, counted by $n_c^p$. Replay only records these gradients as evidence; it takes no optimizer step.

**Decoding the updated state.** At each boundary, we first update the geometric evidence and write the packet's source statistics. A decoder applied to each cell then combines them with replay evidence from the preceding boundary:

$$G_t = D_\theta(X_t, E_t^g, E_{t-1}^p, A_t). \tag{2}$$

The decoder revises a cell through a revision branch gated by geometric coverage and an opacity head; both read replay evidence. The frozen *base head*, trained with the encoder on six-view inputs, predicts $\hat{g}_c$ from $x_{c,t}$, and the revision branch predicts $\tilde{g}_c$ from the cell features, normalized evidence and age. We blend them as

$$g_c^{\text{mix}} = \hat{g}_c + (1 - e^{-v_c})(\tilde{g}_c - \hat{g}_c). \tag{3}$$

As later cameras cover a cell, the gate $1 - e^{-v_c}$ admits revisions to all its attributes. A cell with $v_c = 0$ keeps the base prediction.

The *opacity head* $h_\theta$ has a zero-initialized output layer and predicts an opacity offset from $\phi_c$, thirteen normalized statistics of observation mass, geometric evidence, age and replay evidence that

also feed the revision branch (Table 8). With mixed opacity $\alpha_c^{\mathrm{mix}}$, clamp margin $\epsilon$ and sigmoid $\sigma$,

$$\ell_c = \mathrm{logit}(\mathrm{clip}(\alpha_c^{\mathrm{mix}}, \epsilon, 1-\epsilon)), \quad \delta_c = \mathrm{clip}(h_\theta(\phi_c), -6, 6),$$
$$\alpha_c^{\mathrm{out}} = \alpha_c^{\mathrm{mix}} + \mathbf{1}[n_c^p > 0][\sigma(\ell_c + \delta_c) - \sigma(\ell_c)]. \tag{4}$$

The head predicts an opacity correction for any cell with a replay hit, even without geometric coverage, from one boundary after the cell's creation. The difference form preserves the original opacity exactly when $\delta_c = 0$.

After decoding $G_t$, replay updates the stored evidence:

$$E_t^p = \mathrm{Accumulate}\big(E_{t-1}^p, \mathrm{Replay}(G_t^{\perp}, u, \mathcal{Q}_t)\big). \tag{5}$$

Here $\mathcal{Q}_t$ is the selected camera set, and $\mathrm{Accumulate}$ adds its statistics to persistent cell counters, initialized to zero for new cells. After the last packet $T$, a *final decode* applies Eq. (2) with $E_T^p$ and returns the final map $G_{\mathrm{fin}}$ without a new frame or render.

### 3.4 Causal encoding and per-frame publication

Each packet starts with a joint encoding of its first three frames; every later frame reads only its two nearest earlier frames in the packet, by camera center. All cross-view operators follow this causal graph, so each frame is encoded once and its activations are cached.

This encoder supports two publication schedules. The *packet path* processes six frames together. The *frame path* publishes new cells from the third arrival of each packet onward and adds their base-head Gaussians to the preceding boundary map; at the next boundary, these provisional cells join the full revision. Because the two paths share weights and boundary updates, their mean final-map PSNR differs by less than $0.01\,\mathrm{dB}$ (Section F).

### 3.5 Training

We train the six-view encoder on RE10K. The streaming stage trains on twelve-packet clips (72 context frames), one clip in four from ScanNet. It updates the encoder's depth and Gaussian regressors, the revision branch, opacity head and conditioning layers, while image features and the base head stay frozen. Held-out targets supervise map renders with MSE plus LPIPS (Zhang et al., 2018). Earlier state is detached, so backpropagation spans one packet, and replay uses context images with the delayed schedule of Eq. (5). The checkpoint is selected on training-validation scenes.

To isolate replay, we also train *RetroSplat (no replay)* with the same pretraining, streaming recipe and update budget. This control omits replay passes, replay inputs and the opacity head but keeps geometric evidence and the revision branch.

### 3.6 Opacity refit after the stream

After the stream, an optional *learned refit* optimizes the final map's opacity logits $u_c$ against context frames; means, covariances, colors and network weights stay fixed. Following Meta-SGD, Diff3R, Learn2Splat and LaCT (Li et al., 2017; Liu et al., 2026; Pearl et al., 2026; Zhang et al., 2026), we learn the step size rather than the update direction. At round $k$, we render sixteen context frames $\mathcal{F}_k$, differentiate their photometric error, and update each cell along its Adam direction $s_c^{(k)}$ with a learned, clipped step size:

$$\bar{\gamma}_c^{(k)} = \frac{1}{|\mathcal{F}_k|} \sum_{q \in \mathcal{F}_k} \frac{\partial \bar{L}_q}{\partial u_c}, \qquad u_c^{(k)} = u_c^{(k-1)} - \eta_0 \exp\big(R_\psi(f_c^{(k)})\big)\, s_c^{(k)}. \tag{6}$$

Here $\bar{L}_q$ is the mean squared RGB error corresponding to $L_q$ in Eq. (1), and $s_c^{(k)}$ is the bias-corrected Adam direction of $\bar{\gamma}_c^{(1)}, \ldots, \bar{\gamma}_c^{(k)}$. The input $f_c^{(k)}$ contains gradient statistics, current opacity and the cell's stored evidence $\phi_c$. A 737-parameter network $R_\psi$ predicts the step size. Its output layer is zero-initialized, so the rule starts as per-cell Adam with base rate $\eta_0 = 0.3$.

We train this network across clips through four refit rounds with first-order gradients and an outer loss of MSE plus $0.05$ LPIPS on held-out views; the streaming network stays frozen. At test time, the refit sees only context frames. We selected twelve rounds on training-validation scenes before test evaluation (Section H; cost in Section 4.5).

# 4 EXPERIMENTS

## 4.1 EVALUATION SETUP

We use the released posed ReCoSplat protocol (Cheng et al., 2026): 1,648 RE10K scenes at 64 and 128 context views and 752 ACID scenes at 32 and 64 views, with eight held-out targets per scene. A stream of $N$ frames gives $N$ context views. The 64-view RE10K list is a subset of the 128-view list over nearly the same span, so the 128-view stream doubles the sampling density, not the coverage. Because RE10K uses separate targets at each length, we compare *cohort means* over all scenes of a protocol. All methods receive the same chronological frames and render full $224^2$ targets.

Table 1: **Final-map quality on RealEstate10K and ACID.** RetroSplat + refit runs twelve rounds of learned opacity updates on context frames after the stream (Section 3.6); all other rows are final maps without post-stream optimization. Bold: best; underline: second best.

| **RealEstate10K** (1,648 scenes) | 64 input views | | | 128 input views | | |
|---|---|---|---|---|---|---|
| Method | PSNR (dB)↑ | SSIM↑ | LPIPS↓ | PSNR (dB)↑ | SSIM↑ | LPIPS↓ |
| ReCoSplat | 25.82 | .884 | .120 | 24.91 | .864 | .142 |
| tttLRM-AR | <u>28.69</u> | **.920** | **.094** | 25.02 | .856 | .161 |
| RetroSplat | 26.86 | <u>.907</u> | <u>.111</u> | <u>26.87</u> | <u>.902</u> | <u>.116</u> |
| *Refit on context frames* | | | | | | |
| RetroSplat + refit | **28.87** | **.920** | <u>.111</u> | **28.72** | **.918** | **.114** |
| **ACID** (752 scenes) | 32 input views | | | 64 input views | | |
| Method | PSNR (dB)↑ | SSIM↑ | LPIPS↓ | PSNR (dB)↑ | SSIM↑ | LPIPS↓ |
| ReCoSplat | 25.77 | .791 | .187 | 25.64 | .790 | .194 |
| tttLRM-AR | <u>29.52</u> | **.888** | **.134** | <u>28.69</u> | <u>.872</u> | <u>.155</u> |
| RetroSplat | 27.61 | .860 | .158 | 27.42 | .856 | .164 |
| *Refit on context frames* | | | | | | |
| RetroSplat + refit | **29.54** | <u>.884</u> | <u>.146</u> | **29.60** | **.886** | **.146** |

**Baselines.** We evaluate the released ReCoSplat and tttLRM-AR models with context poses (Cheng et al., 2026; Wang et al., 2026a). tttLRM-AR reads the same frames upsampled to $536 \times 960$ and runs on two GPUs at 128 views. ReCoSplat (1.11B parameters) and tttLRM-AR (400M) train on DL3DV, ReCoSplat also on ScanNet++, with streams of up to 64 views; neither trains on RE10K. RetroSplat (99.4M) trains on 72-frame clips (Section A, which also reports RGB-only OF$^3$GS).

**Evaluated maps.** A *final map* is one Gaussian set used for every target once the last context view arrives. For RetroSplat, it is the output of the final decode (Section 3.3), with fixed weights, before any refit (Section 3.6). *Online* quality uses the map available as the stream reaches each target. *Retain-$K$* renders the target from the newest boundary map available $K$ source-video frames after that target.

## 4.2 FINAL-MAP QUALITY

RetroSplat beats ReCoSplat on PSNR, SSIM and LPIPS in all four protocols (Table 1). The paired PSNR gain is 1.04–1.96 dB, and it holds on ACID, on which none of the evaluated systems was trained. Before refit, tttLRM-AR leads on RE10K-64 and both ACID protocols; RetroSplat takes the lead at RE10K-128 (Fig. 3).

**Denser sampling of the same span.** When the RE10K stream doubles from 64 to 128 context views over the same span, final-map PSNR changes by $+0.01$ dB for RetroSplat, $-0.91$ dB for ReCoSplat and $-3.67$ dB for tttLRM-AR (Figs. 1 and 8); all three exceed their training stream lengths at 128 views. Without replay this stability is lost; RetroSplat (no replay), trained separately,

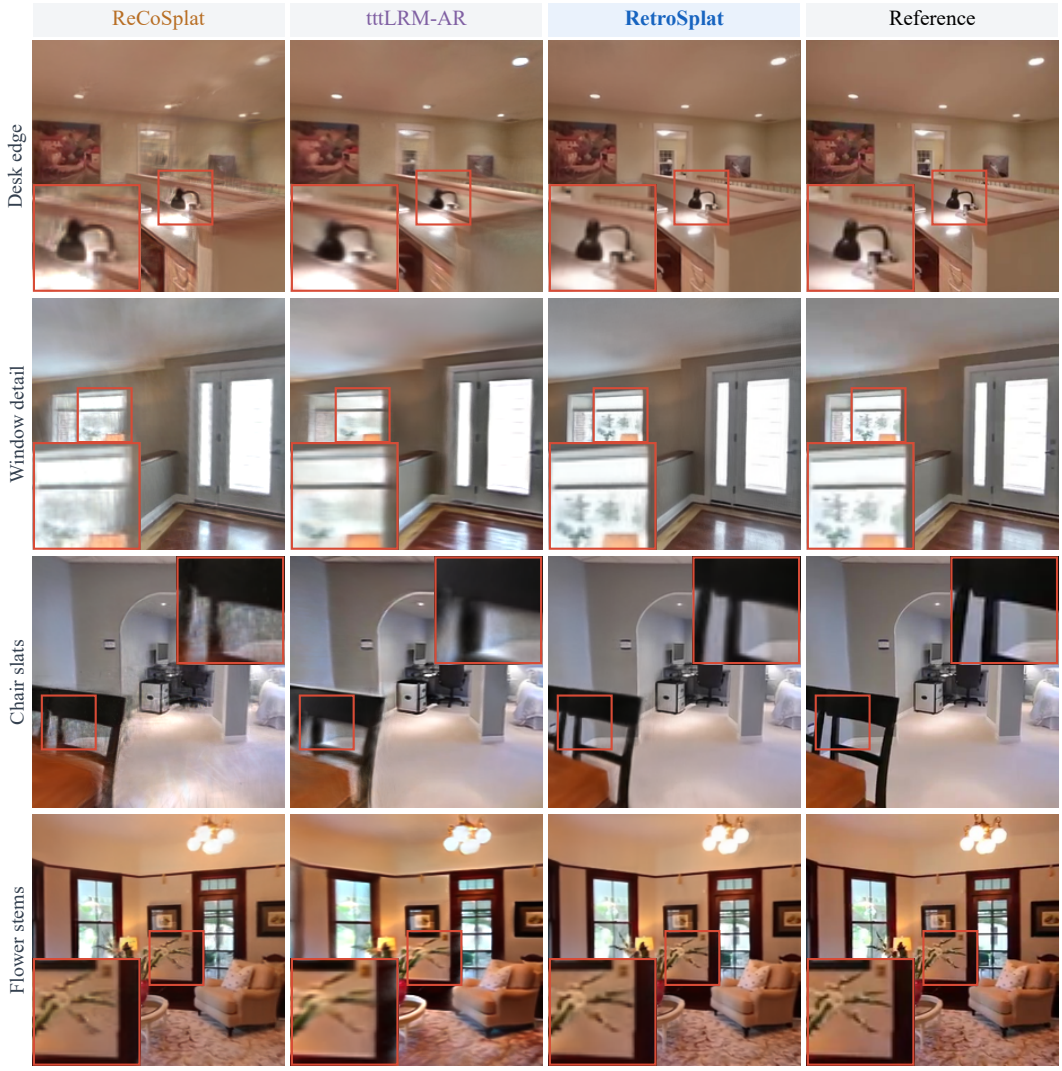



Figure 3: **Selected final maps from** 128**-view RE10K streams, without refit.** RetroSplat preserves desk edges, window detail, chair slats and flower stems blurred by tttLRM-AR. Boxes show 2× crops; selection and additional comparisons are in Section G.

falls from 24.92 to 24.12 dB under the same doubling (Table 5). Thinning RE10K-128 uniformly to 64 frames on 80 scenes changes our map by −0.12 dB, so RetroSplat needs no frame selection; tttLRM-AR gains 3.48 dB from this thinning and then exceeds our unrefitted map (Section I).

**Post-stream refit.** Twelve learned refit rounds on context frames (Section 3.6) raise PSNR over the final map by 1.85–2.17 dB on every protocol and give the best PSNR, SSIM and LPIPS on RE10K-128 and ACID-64. On RE10K-64 the PSNR lead over tttLRM-AR is 0.18 dB; on ACID-32 the two are statistically tied. tttLRM-AR keeps the lower LPIPS on both and the higher SSIM on ACID-32.

### 4.3 Replay Evidence and Retroactive Revision

Of the three components zeroed at a fixed checkpoint, replay evidence matters most; zeroing it at every decode costs 2.0–2.9 dB (Table 2). A separately trained RetroSplat (no replay), using the same recipe, trails by 1.6–2.8 dB on all four protocols (ACID-32 in Table 13). Adapting a frozen

Table 2: **Ablations.** ΔPSNR (dB), variant minus intact; adaptation compares masked minus read models. 160 / 80 / 80-scene subsets unless marked.

| | RE10K 64 | RE10K 128 | ACID 64 |
|---|---|---|---|
| *Intact RetroSplat (PSNR)* | *26.71* | *26.44* | *27.86* |
| *Zeroed at every decode* | | | |
| Replay evidence | −2.18 | −2.88 | −1.96 |
| Opacity offset | −1.45 | −2.01 | −1.19 |
| Geometric evidence and gate | −1.18 | −1.77 | −0.87 |
| *Full cohorts* | | | |
| Final decode skipped | −0.33 | −0.44 | −0.39 |
| Retrained without replay | −1.94 | −2.76 | −2.13 |
| Adapted: masked − read (2 runs) | −1.37 | −1.62 | – |
| *Per-cell Adam instead of learned updates* | | | |
| Streaming opacity update (80 sc.) | −0.28 | – | – |
| Refit, four rounds (full) | −0.29 | −0.32 | −0.37 |

Table 3: **Where revision acts.** ΔPSNR (dB) for targets in the first, middle and last third of the context frames.

| | Early | Mid | Late |
|---|---|---|---|
| *RetroSplat − no-replay model, full* | | | |
| RE10K-64 | +2.86 | +2.51 | +0.55 |
| RE10K-128 | +4.19 | +3.18 | +1.12 |
| *Zeroed − intact, RE10K-64 (160)* | | | |
| Replay | −2.21 | −2.34 | −2.02 |
| Geometry, gate | −0.01 | −1.23 | −2.37 |
| *Zeroed − intact, RE10K-128 (80)* | | | |
| Replay | −3.26 | −2.64 | −2.68 |
| Geometry, gate | −0.77 | −1.61 | −2.90 |

Table 4: **Online reconstruction and retention on RE10K-64.** PSNR (dB) over 1,648 scenes, before any post-stream refit. Retain-$K$ and per-packet rows use boundary maps; RetroSplat's per-frame scores also include provisional maps.

| | PSNR (dB) ↑ | | | | |
|---|---|---|---|---|---|
| Evaluation state | RetroSplat (no replay) | RetroSplat | ReCoSplat | Window (no map) | RetroSplat − ReCoSplat |
| Online, per packet | 22.17 | 22.09 | 21.89 | – | +0.19 |
| Online, per frame | 23.82 | 23.54 | 21.89 | 27.96 | +1.65 |
| Retain-12 | 25.86 | 25.54 | 23.81 | 33.31 | +1.73 |
| Retain-36 | 28.40 | 28.50 | 26.34 | 18.65 | +2.16 |
| Retain-96 | – | 26.58 | 26.31 | 12.66 | +0.26 |
| Final map | 24.92 | 26.86 | 25.82 | 11.52 | +1.04 |

Window uses only the six latest views, so earlier content leaves its context. ReCoSplat's per-frame entry repeats its packet score. Differences are computed before rounding; – denotes an unmeasured state.

no-replay model to replay inputs through 2,777 decoder parameters adds 1.4–1.6 dB on RE10K; with replay masked, the same adaptation gains 0.01 dB (Section C.2).

**Where revision acts.** Against the no-replay model, early targets gain most (2.86 dB at 64 RE10K views, against 0.55 dB for late ones; Table 3), and the contrast grows at 128 views. Gains over ReCoSplat follow the opposite pattern, from −0.10 dB on early to +2.20 dB on late RE10K-64 targets (Table 17). At a fixed checkpoint, zeroing replay lowers PSNR at every position. Zeroing geometric evidence and its gate leaves early RE10K-64 targets almost unchanged (−0.01 dB) and costs 2.37 dB on late ones. The final decode isolates revision from new input. Reading only the stored state, with no new frame or render, it raises PSNR by 0.33–0.57 dB and improves at least 1,412 of 1,648 RE10K and 695 of 752 ACID scenes per protocol, at 2.1–3.6% of the stream-loop time (Table 16 and Fig. 6).

**Learned updates versus per-cell Adam.** In the streaming control, per-cell Adam updates to opacity offsets replace $h_\theta$, with a tuned learning rate and the same network weights, replay cameras and render budget. At the last boundary the two match in PSNR, with better SSIM and LPIPS for the head. The head gains more from the final evidence (+0.39 vs. +0.10 dB; Section C.3), so after the final decode it leads by 0.28 dB (Table 2); it keeps no optimizer state, and Adam holds 0.7 GiB of offsets and moments. After four post-stream refit rounds at equal budget, learned per-cell step sizes beat per-cell Adam at its calibrated base rate by 0.24–0.37 dB on the four protocols, with lower

Table 5: **Cost of streaming and refit.** 20 RE10K scenes on H100s; inputs $224^2$ except tttLRM-AR ($536\times960$). Views/s: whole pipeline, summed over GPUs; GB: mean peak allocated per GPU.

| | | 64 views | | 128 views | | PSNR↑ (1,648 scenes) | |
|---|---|---|---|---|---|---|---|
| Method | GPUs | Views/s↑ | GB↓ | Views/s↑ | GB↓ | 64 views | 128 views |
| ReCoSplat | 1 | **56.0** | **7.6** | **56.0** | **8.4** | 25.82 | 24.91 |
| tttLRM-AR | 1 / 2 | 7.6 | 51.0 | 12.1 | 53.9 | 28.69 | 25.02 |
| RetroSplat (no replay) | 1 | 43.5 | 12.2 | 33.4 | 23.8 | 24.92 | 24.12 |
| RetroSplat, packet path | 1 | 19.5 | 12.5 | 13.5 | 24.1 | 26.86 | 26.87 |
| RetroSplat + refit (12 rounds) | 1 | 8.3 | 12.5 | 7.4 | 24.1 | **28.87** | **28.72** |

LPIPS. At twelve rounds on the control subsets, Adam at a PSNR-tuned rate trades slightly higher PSNR for worse LPIPS (Section H).

### 4.4 Online quality and retention

Per-frame publication raises RE10K-64 online PSNR by $1.45\,\mathrm{dB}$ over the packet path, which is itself $0.19\,\mathrm{dB}$ above ReCoSplat (Table 4). Replay lowers online and Retain-12 scores ($23.54$ vs. $23.82\,\mathrm{dB}$ online per frame) but slows the later decline. From the best Retain-$K$ state to the final map, RetroSplat declines by $1.6\,\mathrm{dB}$ and the no-replay model by $3.5\,\mathrm{dB}$; ReCoSplat declines least, by $0.5\,\mathrm{dB}$, but its best state stays below our final map. A map-free window over the six latest views scores highest online and lowest at the final map, as earlier regions leave its context (Section B.3).

**Latency and serving.** On one H100 at $64$ views, the frame path publishes each of the last three frames of a packet $56\,\mathrm{ms}$ after admission (median, back-to-back inputs), and the boundary decode and replay pass add $48$ and $145\,\mathrm{ms}$ per packet (Table 11). A trace-driven simulation replays our service times and those of the released tttLRM-AR code under fixed-rate arrivals on 20 RE10K scenes and renders each target from the newest map published by its stream time plus a deadline, so a served map can include later frames. Under these compute constraints, our frame path serves higher PSNR than tttLRM-AR at every simulated rate (5–56 views/s) and deadline (0–1000 ms), for example $27.30$ against $12.97\,\mathrm{dB}$ at $12.7$ views/s and $250\,\mathrm{ms}$. Live runs of the frame path match the simulation within about $1\,\mathrm{dB}$; tttLRM-AR was not run live. Without compute constraints, tttLRM-AR's prefix maps score higher online than our frame path ($27.24$ vs. $24.40\,\mathrm{dB}$; Section B.3).

### 4.5 Computation and memory

Replay is the main cost of revision; it lowers packet-path throughput at $64$ views from $43.5$ to $19.5$ views/s (Table 5). The frame path raises the work per six frames from $299$ to $442\,\mathrm{ms}$ (Table 11). With twelve refit rounds, the full pipeline takes $7.70\,\mathrm{s}$ per RE10K-64 scene on one H100, against $8.42\,\mathrm{s}$ for tttLRM-AR. At $128$ views, our pipeline needs $17.2\,\mathrm{s}$ on one GPU, longer than tttLRM-AR's $10.6\,\mathrm{s}$ on two, but uses less GPU time in total ($17.2$ vs. $21.2$ GPU-seconds). The append-only ReCoSplat is the fastest system and uses the least memory (Section B.2).

## 5 Limitations and conclusion

**Limitations.** We assume known poses and static scenes. The map has no eviction policy, so peak allocated memory grows from $12.5$ to $24.1\,\mathrm{GB}$ when RE10K sampling doubles. Reconstruction quality drops on the doubled-span ACID protocol (Section I). Thin structures remain a failure case (Fig. 9), and RetroSplat trails ReCoSplat on ScanNet and DL3DV (Section E.1).

**Conclusion.** RetroSplat keeps revision evidence beside each committed Gaussian: geometric votes from later cameras and replay gradients from earlier views. A decoder trained across scenes reads this evidence to revise the map. Before refit, the final maps beat ReCoSplat on four RE10K and ACID protocols and hold their PSNR when the same RE10K span is sampled twice as densely. Replay helps preserve earlier content through opacity feedback; extending that feedback to other Gaussian attributes is the next step.

## ETHICS STATEMENT

This work uses existing public scene-reconstruction benchmarks. It involves no human-subject experiments, recruitment of human participants, or collection of new personal data.

## REPRODUCIBILITY STATEMENT

We evaluate fixed released baseline checkpoints. Streaming and learned-refit checkpoints are selected on held-out training-validation scenes; test targets are not used for checkpoint selection. Appendices A–I document the architecture, datasets, training and selection rules, evaluation protocols, implementation details, and additional results. Code, checkpoints, and scripts for reproducing the tables and figures will be released upon publication.

## AI USE STATEMENT

We used large language model assistants to support figure preparation, and manuscript editing. Writing assistance covered grammar, phrasing, readability, consistency of terminology and notation, and checks of the conference template and reference list. Reported measurements come from executed experiments. The authors reviewed the manuscript and take full responsibility for its content.

## REFERENCES

Aljaž Božič, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. TransformerFusion: Monocular RGB scene reconstruction using transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1403–1414, 2021.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467, 2024. doi: 10.1109/CVPR52733.2024.01840.

Ruiyang Chen, Feiran Li, Chu Zhou, Zonglin Li, Zhanyu Ma, and Heng Guo. OF$^3$GS: On-the-fly feed-forward 3D Gaussian splatting from unposed images. *arXiv preprint arXiv:2606.03254*, 2026.

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 *Computer Vision – ECCV 2024*, pp. 370–386, 2025a. doi: 10.1007/978-3-031-72664-4_21.

Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3R: Gradient guided generalizable reconstruction. In *Computer Vision – ECCV 2024*, pp. 305–323, 2025b. doi: 10.1007/978-3-031-72658-3_18.

Ziwen Chen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Fuxin Li, and Zexiang Xu. Long-LRM: Long-sequence large reconstruction model for wide-coverage Gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4349–4359, 2025c. doi: 10.1109/ICCV51701.2025.00414.

Freeman Cheng, Botao Ye, Xueting Li, Junqi You, Fangneng Zhan, and Ming-Hsuan Yang. ReCoSplat: Online feed-forward Gaussian splatting via render-and-compare. *arXiv preprint arXiv:2603.09968*, 2026.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 303–312. ACM, 1996. doi: 10.1145/237170.237269.

Arda Düzçeker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15324–15333, 2021.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, Zhangyang Wang, and Yue Wang. InstantSplat: Sparse-view Gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. DeepView: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2367–2376, 2019.

Jiaxin Guo, Tongfan Guan, Wenzhen Dong, Wenzhao Zheng, Wenting Wang, Yue Wang, Yeung Yam, and Yun-Hui Liu. SaLon3R: Structure-aware long-term feedforward 3D reconstruction from unposed images. *International Journal of Computer Vision*, 134(8):383, 2026. doi: 10.1007/s11263-026-02965-2.

Guichen Huang, Ruoyu Wang, Xiangjun Gao, Che Sun, Yuwei Wu, Shenghua Gao, and Yunde Jia. LongSplat: Online generalizable 3D Gaussian splatting from long sequence images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(6):4994–5002, 2026. doi: 10.1609/aaai.v40i6.42504.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025. doi: 10.1145/3763326.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat track & map 3D Gaussians for dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366, June 2024.

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

Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. doi: 10.1023/A:1008191222954.

Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-Hong Kao, Ying-Cong Chen, and Yan Lu. StreamGS: Online generalizable Gaussian splatting reconstruction for unposed image streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25841–25850, 2025. doi: 10.1109/ICCV51701.2025.02397.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Yueh-Cheng Liu, Jozef Hladký, Matthias Nießner, and Angela Dai. Diff3R: Feed-forward 3D Gaussian splatting with uncertainty-aware differentiable optimization. *arXiv preprint arXiv:2604.01030*, 2026.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, June 2024.

Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3D scene reconstruction from posed images. In *Computer Vision – ECCV 2020*, pp. 414–431, 2020. doi: 10.1007/978-3-030-58571-6_25.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR '11 Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136. IEEE, October 2011. doi: 10.1109/ISMAR.2011.6092378.

Naama Pearl, Stefano Esposito, Haofei Xu, Amit Peleg, Patricia Gschossmann, Lorenzo Porzi, Peter Kontschieder, Gerard Pons-Moll, and Andreas Geiger. Learn2Splat: Extending the horizon of learned 3DGS optimization. *arXiv preprint arXiv:2605.15760*, 2026.

Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. SimpleRecon: 3D reconstruction without 3D convolutions. In *Computer Vision – ECCV 2022*, pp. 1–19. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-19827-4_1.

Changhao Song, Yuxuan Wang, Qibiao Li, Youcheng Cai, and Ligang Liu. StreamSplat: Streaming feed-forward 3D Gaussian splatting. *arXiv preprint arXiv:2608.01659*, 2026.

Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6229–6238, 2021.

Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15598–15607, 2021.

Chen Wang, Hao Tan, Yifan Wang, Zhiqin Chen, Yuheng Liu, Kalyan Sunkavalli, Sai Bi, Lingjie Liu, and Yiwei Hu. tttLRM: Test-time training for long context and autoregressive 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36582–36592, 2026a.

Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, 2025a. doi: 10.1109/CVPR52734.2025.00983.

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Feng Chen, Jia-Wang Bian, Zheng Zhu, Donny Y. Chen, and Bohan Zhuang. VolSplat: Rethinking feed-forward 3D Gaussian splatting with voxel-aligned prediction. In *Computer Vision – ECCV 2026*, pp. 1–19, 2026b. doi: 10.1007/978-3-032-37232-1_1.

Yiming Wang, Lucy Chai, Xuan Luo, Michael Niemeyer, Manuel Lagunas, Stephen Lombardi, Siyu Tang, and Tiancheng Sun. Learning efficient fuse-and-refine for feed-forward 3D Gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 38, pp. 76311–76341, 2025b. doi: 10.52202/085713-2563.

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.

Zike Wu, Qi Yan, Xuanyu Yi, Lele Wang, and Renjie Liao. StreamSplat: Towards online dynamic 3D reconstruction from uncalibrated video streams. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SaiDRQU7Ez.

Chong Xia, Fangfu Liu, Yule Wang, Yize Pang, and Yueqi Duan. OnlineX: Unified online 3D reconstruction and understanding with active-to-stable state evolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 66–76, 2026.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting Gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463, 2025. doi: 10.1109/CVPR52734.2025.01534.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. ReSplat: Learning recurrent Gaussian splatting. In *Computer Vision – ECCV 2026*, pp. 364–382, 2026. doi: 10.1007/978-3-032-37435-6_20.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P4o9akekdf.

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 *Computer Vision – ECCV 2024*, pp. 1–19, 2025. doi: 10.1007/978-3-031-72670-5_1.

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 (CVPR)*, pp. 586–595, June 2018.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Tb9qAxT3xv.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5APgTKsnx8.

# A PROTOCOL DETAILS

**Official protocol.** We use the pinned ReCoSplat release with its unmodified loader, context and target indices, and expected-results CSV. The loader admits 1648 of 1666 RealEstate10K scene keys at both 64 and 128 views; we evaluate this complete set. Every 64-view context list is a subset of its 128-view list. Their endpoints match in 1,635 scenes; in the remaining 13, the 128-view list extends slightly further. ScanNet follows the official conversion, including its 20-pixel border crop and calibration. Upstream LANCZOS resizing and cropping produce $224 \times 224$ inputs and outputs. RetroSplat and ReCoSplat receive identical pixels and ground-truth target cameras. Our encoder retains its native scene gauge and depth bounds (0.5–100 on RealEstate10K, 0.5–15 on ScanNet). Official ReCoSplat weights in their released environment reproduce the posed, calibrated CSV within $0.01\,\mathrm{dB}$ on RealEstate10K-64 and ScanNet-32 and $0.05\,\mathrm{dB}$ on RealEstate10K-128.

**Paired statistics.** Deltas are means of per-scene differences; wins count scenes with a positive difference. Targets are averaged within each scene before averaging over scenes.

**Online and Retain-$K$ protocols.** Online packet queries use the last complete packet of six (RetroSplat) or chunk of eight (ReCoSplat) preceding the target frame. The frame path also publishes new-cell rows between boundaries, alongside the previous boundary rows (Section 3.4); its schedule includes the three-frame bootstrap and overlapping tail re-encoding. These intermediate maps update fresh cells rather than decoding all current statistics. ReCoSplat has no frame path, so its frame entry in Table 4 repeats the packet value. Retain-$K$ admits context indices strictly below the target index plus $K$ and uses the last complete packet, with at least the first bootstrap packet. All queries use the official target lists. A target is early when more than $2/3$ of context frames follow it, late when fewer than $1/3$ follow it, and middle otherwise.

**Causal window control.** The control in Table 4 runs the pretrained six-view predictor (Section B) on frames $i-5, \ldots, i$ after each arrival $i \geq 5$. It retains no map and uses only the six most recent admitted frames, with the first window serving shorter prefixes as for RetroSplat and ReCoSplat. One cold forward with packing takes $55.4\,\mathrm{ms}$ on one H100 ($18.1\,\mathrm{fps}$). Choosing a window by proximity to each target instead would score $31.90\,\mathrm{dB}$ at the final state, compared with $11.52\,\mathrm{dB}$ for the causal window; that target-dependent routing is excluded from the streaming control.

**Compute regimes.** Feed-forward inference uses fixed network weights and no scene-specific optimizer steps. Rasterizer backward passes collect evidence for the next decode: at most twelve views per complete packet (the current six and up to six earlier cameras), or six at cold start. Cost grows with visible scene content (Section 4.5). The optional post-stream refit (Sections 3.6 and H) instead updates final-map opacity logits along per-cell Adam directions scaled by a learned rate, while keeping network weights fixed.

**Training domains.** ReCoSplat reports training on DL3DV (9,894 scenes) and ScanNet++ (968 scenes) (Cheng et al., 2026). Our encoder is trained on RealEstate10K, and the streaming stage draws about one clip in four from ScanNet training scans; ScanNet and ScanNet++ are distinct datasets. ACID is outside both training sets.

## A.1 ADDITIONAL STREAMING BASELINES

Released tttLRM-AR and OF$^3$GS use the same chronological contexts, source tensors and eight held-out targets as ReCoSplat. PSNR, SSIM and LPIPS-VGG use identical implementations, averaging targets within scenes over complete cohorts. These are final-map comparisons; Section B.3 measures publication cost. We did not find runnable official code with suitable weights for StreamGS, LongSplat or StreamSplat-VACC (Li et al., 2025; Huang et al., 2026; Song et al., 2026).

**tttLRM-AR.** The released DL3DV checkpoint (Wang et al., 2026a), trained with up to 64 views, uses four-view fast-weight updates, the official Gaussian decoder and pruning. Every context camera queries the final state. The required target slot contains zero RGB and the first context camera; we capture one Gaussian set for all eight held-out targets. Normalization uses the first context frame

Table 6: **Streaming state and update mechanisms.** Measured costs and sampling comparisons are in Sections B.2 and I.

| | tttLRM-AR | RetroSplat |
|---|---|---|
| Persistent state | fixed-size latent memory: 170M fast weights | voxel cells storing per-frame features and accumulated evidence |
| Revision signal | key–value loss on incoming tokens; one update per four-view chunk | depth support and free-space votes; replay gradients from seen views |
| Map publication | decode all Gaussians from the current latent state | publish new cells per frame; revise committed cells at packet boundaries |
| Publication work | released wrapper rebuilds the whole prefix | incremental encoding and decoding, plus packet-wise replay |
| After the stream | no refit in the released system | optional opacity refit |

Table 7: **RGB-only OF$^3$GS reconstruction.** A similarity fitted only to context cameras places known target cameras in the predicted frame; scores include residual camera error. Posed comparisons use the same cohorts and targets in Table 1.

| Protocol | Scenes | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|
| RE10K-64 | 1,648 | 14.48 | .432 | .507 |
| RE10K-128 | 1,648 | 14.27 | .427 | .518 |
| ACID-32 | 752 | 15.35 | .419 | .534 |
| ACID-64 | 752 | 15.30 | .419 | .537 |

and maximum absolute context-camera translation. Held-out images and cameras enter neither reconstruction nor normalization.

The model expects $536 \times 960$ images. We convert canonical $224^2$ context tensors to RGB8 and resize with PIL LANCZOS, scaling intrinsics along each axis. This changes the aspect ratio to the released shape without adding detail.

The 128-view forward exceeds one H100's 80 GB, so it uses official two-rank sequence parallelism; 32/64 views use one rank. Column-major packing preserves chronological four-view updates and every Gaussian query. Each publication requires a whole-prefix forward (Section B.3).

**OF$^3$GS.** Released unposed inference (Chen et al., 2026) receives context RGB and an empty target-RGB tensor, retaining its context-camera prepass, sequential map pass and native decoder. Canonical $224^2$ sources are converted to RGB8 by released truncation and upsampled with PIL LANCZOS to the native $518^2$ input, preserving square field of view and normalized intrinsics without adding detail. Targets render and score at $224^2$. A global similarity fitted only to context cameras transforms known target cameras into the predicted frame. Let $R_i, C_i$ denote known camera-to-world rotations and centers, and $\widehat{R}_i, \widehat{C}_i$ their predictions. The rotation $Q$ is the closest proper rotation to $N^{-1}\sum_i \widehat{R}_i R_i^\top$. We match the RMS extent of the two context trajectories and their centroids:

$$s = \frac{\sqrt{\sum_i \|\widehat{C}_i - \overline{\widehat{C}}\|^2}}{\sqrt{\sum_i \|C_i - \overline{C}\|^2}}, \qquad t = \overline{\widehat{C}} - sQ\overline{C}. \tag{7}$$

The similarity in Eq. (7) gives $R'_q = QR_q$ and $C'_q = sQC_q + t$ with the target's known intrinsics; Gaussian attributes remain unchanged. No target camera, image or score enters the alignment fit or rule selection. This adapts the published unposed protocol, which also estimates target cameras from target RGB. The model trains on 10,000 DL3DV sequences and 10,000 RealEstate10K trajectories with a pretrained StreamVGGT backbone; we add no training.

Table 7 reports all four protocols, including residual pose error after context alignment; Table 1 gives the posed comparisons on the same cohorts and targets.

**Execution details.** Evaluation uses PyTorch 2.4.0, CUDA 12.1, and gsplat 1.5.0 for tttLRM and 1.4.0 for OF$^3$GS. Both adapters verify a fixed map across all eight target renders; camera round trips validate OF$^3$GS coordinate conversion. Native OF$^3$GS voxelization is nondeterministic across runs.

## B Implementation and training details

**State and scene gauge.** Table 9 lists the retained state. Fine and coarse cells use nearest aggregation at widths $0.001$ and $0.128$ in dataset camera-coordinate units (Section 3.2), without assuming a common physical scale across datasets. Each source frame owns its contributions; re-encoding subtracts the old contribution before adding its replacement. Fine anchors and RGB are observation means; coarse features are queried at the rounded mean anchor. Exact keys use 20 bits per coordinate, and observations outside the fine-key range are dropped before aggregation. The pointwise decoder re-evaluates only cells whose features, evidence or clipped age changed, including every fine cell querying a changed coarse cell. This matches a full decode to floating-point tolerance. Sorting, dependency scans, source payloads and renderer packing grow with scene size.

**Geometric evidence.** Project the anchor of a pre-existing cell $c$ into a frame $i$ of the current packet. Let $z_{ci}$ be the anchor's depth in camera $i$, $d_i$ the depth predicted by the encoder at the pixel where the anchor lands, and $q_i$ the predicted confidence at that pixel; $b_{ci}$ is one when $z_{ci} > 10^{-3}$ and the projection falls inside the image, and zero otherwise. With $\rho_{ci} = z_{ci}/\max(d_i, 10^{-6})$ and $\tau = 0.1$, the accumulated against, support, and in-frustum contributions are

$$a_{ci} = b_{ci} q_i \operatorname{clip}\left(\frac{1-\tau-\rho_{ci}}{\tau}, 0, 1\right), \quad s_{ci} = b_{ci} q_i \operatorname{clip}\left(1 - \frac{|\rho_{ci}-1|}{\tau}, 0, 1\right), \quad v_{ci} = b_{ci}.$$

Against/support counters threshold the packet mean over frames containing the projection at $0.5$. The statistics overlap: in-frustum counts include cells behind the predicted surface. Only anchors present before the packet receive votes. Geometric and replay statistics accumulate across packets, while observation mass follows the current source ledger (Table 8). Age is clipped at eight packets before $T$; other dependency changes can revisit older cells.

**Replay evidence $E^p$ (photometric gradients).** After each commit, we render the detached map into the current packet's cameras and up to six earlier committed frames sampled without replacement using a fixed sampling sequence. Each view sums squared RGB residuals and differentiates only temporary per-cell opacity logits. For gradient $g_{ci}$, form $\widetilde{q}_{ci} = \operatorname{clip}(1000 g_{ci}, -20, 20)$; store $q_{ci} = \widetilde{q}_{ci}$ when $|\widetilde{q}_{ci}| > 10^{-3}$, and zero otherwise. We accumulate positive and negative mass, nonzero-gradient frame count, hit-pass count, and positive and negative hit-frame counts. Repeated replays contribute on each pass; subthreshold mass and counts are discarded. Gradient signs indicate descent directions: positive reduces the logit and negative increases it. The next decode consumes this evidence, including a final decode after the last packet's replay (Sections 3.3 and C.3).

**Learned opacity correction.** The head $h_\theta$ in Eq. (4) is a 13–32–1 ReLU network with a zero-initialized final layer and clamp margin $\epsilon = 10^{-4}$. Its input $\phi_c$ contains normalized evidence; $n_c^p$ counts nonzero-gradient frames after thresholding, and $\alpha_c^{\mathrm{mix}}$ is the gated opacity of Eq. (3). The head activates only after replay evidence arrives, so new cells first receive its correction at a later decode.

**Pretrained predictor interface.** The six-view predictor supplies a 164-channel cell descriptor: 16 fine-grid, 16 coarse-grid and 132 pixel channels (64 upsampled features, 64 Gaussian-regressor features, RGB and confidence). Its GELU decoder combines a 164–256–256–132 fusion MLP with a pixel-feature residual and a 132–38–38 pointwise Gaussian head. For streaming, a frozen copy supplies the reference prediction; the revised branch adds a projection of the thirteen evidence features (Table 8) to the descriptor. The branches predict all Gaussian attributes and blend them with $1 - \exp(-v_c)$. The separate opacity head can act at $v_c = 0$; its offset is applied at each decode without accumulating in the map. The adapter uses sigmoid-bounded mean offsets of $\pm 1.5$ fine-cell widths, softplus scales clamped to $[10^{-10}, 3]$, normalized quaternions, and degree-two spherical harmonics (9 coefficients per color). Anchor RGB supplies the DC color with a learned residual gate.

Table 8: **The thirteen decoder evidence features.** With $T(x) = \log(1+x)/5$ and $D = \max(n_c^p, 1)$: rows 2–6 form $E^g$, row 7 is the age and rows 8–13 form $E^p$. The same vector feeds the revision projection and the opacity head.

| Index | Accumulated quantity | Feature transform |
|---|---|---|
| 1 | Observation mass (pixel count) $m_c$ | $T(m_c)$ |
| 2 | Geometric against mass $a_c$ | $T(a_c)$ |
| 3 | In-frustum count $v_c$ | $T(v_c)$ |
| 4 | Geometric support mass $s_c$ | $T(s_c)$ |
| 5 | Packets with mean against vote above $0.5$ | $T(n_c^a)$ |
| 6 | Packets with mean support vote above $0.5$ | $T(n_c^s)$ |
| 7 | Cell age in packets $A_c$ | $T(\mathrm{clip}(A_c, 0, 8))$ |
| 8 | Positive photometric gradient mass $p_c^+$ | $T(p_c^+)$ |
| 9 | Negative photometric gradient magnitude $p_c^-$ | $T(p_c^-)$ |
| 10 | Nonzero-gradient frame count $n_c^p$ | $T(n_c^p)$ |
| 11 | Replay passes with at least one hit $b_c^p$ | $T(b_c^p)$ |
| 12 | Positive-gradient hit-frame count $n_c^+$ | $n_c^+/D$ |
| 13 | Negative-gradient hit-frame count $n_c^-$ | $n_c^-/D$ |

Table 9: **Persistent cell state and decoded map cache.** Accumulators and decoded rows use fp32; cell keys and observation counts are integers.

| State | Stored content and purpose |
|---|---|
| Fine-cell statistics | Exact cell key and pixel count; sums of 132 pixel features, 16 projected fine features, 3D position (3), and RGB (3): 154 summed channels. |
| Coarse-cell statistics | Exact cell key, pixel count, and sums of 16 projected coarse features. |
| Source ownership | Source frame ID and revision, with its fine/coarse keys, counts, and sums. A replacement removes these contributions before inserting the new revision. |
| Evidence and provenance | Five geometric statistics, creation revision, and six replay-evidence statistics per cell. Creation revision is retained even after the age feature saturates. |
| Decode cache and indices | 40 decoded values per live Gaussian, cell-slot mapping, coarse dependency keys, and change tracking for affected decodes. |
| Replay context | Earlier context images and cameras. Packet mode uses its context pool; the frame path uses an fp16 image buffer with reservoir sampling above 128 frames. |

The cache stores 40 floating-point values per Gaussian: mean (3), covariance (9), color harmonics (27), and opacity (1).

**Age convention.** Let $r_c$ be a cell's creation packet and $n$ the current packet. Training reads the pre-commit counter and supplies $\mathrm{clip}(n - r_c - 1, 0, 8)$; released evaluation reads the post-commit counter and supplies $\mathrm{clip}(n - r_c, 0, 8)$. New cells have age zero in both. Matching the training convention changes PSNR by $+0.03$ / $+0.04\,\mathrm{dB}$ on the final RealEstate10K-64 / ACID-64 control maps (Section C.1); reported results use the released convention.

**Map capacity.** The cap is 8M cells, increased to 24M for DL3DV-256, which produces about 12.8M cells. Exceeding it raises an error; there is no eviction. Slots with zero observations may be reused, but runtime and memory grow with the retained stream.

### B.1 Training

**Trainable modules.** Streaming training updates the depth regressors and residuals, depth heads and depth upsampler, Gaussian regressor, level projection, revised decoder, history injections, and evidence heads. The reference decoder and other pretrained image features remain frozen. The three history channels are premultiplied luma, normalized inverse depth, and alpha. The map is rendered at $448^2$, area-downsampled to $112^2$, then bilinearly resized to the encoder input.

**Initialization and training.** The predictor is pretrained on RealEstate10K six-view windows with the causal neighbor rule for 150k updates on four H100s, followed by 30k updates at learning rate $2 \times 10^{-5}$. Training clips of the streaming stage contain 12 packets (72 context frames at stride two or three); about one clip in four comes from ScanNet training scans at strides three to eight. After every packet the accumulated map is rendered into two recent and four earlier held-out frames (six recent for the first packet) and penalized with MSE and $0.05\times$LPIPS (Zhang et al., 2018). The map written by earlier packets is detached (truncated backpropagation of one packet). Replay follows the procedure above. AdamW uses learning rate $3 \times 10^{-5}$ for unfrozen pretrained modules and $10^{-4}$ for new modules, with a 40k-update budget on four H100s at $224^2$. Checkpoints are selected on held-out *training* scenes: step 36k for RetroSplat and step 40k for RetroSplat (no replay), which shares the pretraining, clips and update budget but has no replay pass, replay-evidence inputs or opacity head.

### B.2 Cost and timing

**Throughput benchmark.** Tables 5 and 10 use the same 20 official RealEstate10K scenes at 64 and 128 views, with two untimed warm-up streams and three timed repeats per scene. RetroSplat and ReCoSplat each use one uncontended H100. RetroSplat uses eager fp32 image encoding with TF32 matmul and incremental decoding; its largest PSNR difference from full decoding is below 0.01 dB. ReCoSplat uses its released environment, weights, bf16 autocast, chunk size eight and expandable-segments allocator. Input fps divides context frames by whole-stream wall time, including replay and excluding target rendering. CUDA-event traces use one stream per scene; Table 10 reports per-packet replay time and ends before the final decode. That decode adds 0.105 / 0.207 s at 64 / 128 views, yielding 19.5 / 13.5 fps.

Memory is per-scene peak allocated GB; peak reserved memory for RetroSplat is about 38 / 66 GB at 64 / 128 views. Gaussian counts include all live fine cells for RetroSplat and all stored pixel-aligned Gaussians for ReCoSplat. At the 0.001 fine width, RetroSplat retains nearly one cell per input pixel. Its allocated memory grows from 12.5 to 24.1 GB as the stream doubles, compared with 7.6 to 8.4 GB for ReCoSplat.

**Numerical precision.** Repeated fp32 runs differ by less than 0.03 dB on RealEstate10K-64, RealEstate10K-128 and ScanNet-32 because replay rasterizer atomics are nondeterministic. We retain fp32 execution: fp16 encoding with CUDA-graph capture changes PSNR beyond this variation.

**Frame-path service times.** Table 11 uses two timed streams per scene after an untimed warm-up on the same 20 scenes, pooling percentiles over arrivals. Back-to-back inputs measure service time. Packet positions 1–2 render history and buffer inputs; position 3 runs the bootstrap clique, and positions 4–6 each encode one causal row. Merging and decoding new cells publishes a provisional map. Position 6 then publishes the boundary revision and gathers replay evidence without another publication. Incomplete tails re-encode the last six frames. Work per six frames is the median packet sum of frame service times. Whole-stream throughput also includes bootstrap waits and tail re-encoding: 458 / 583 ms per packet, or 12.7 / 10.0 fps at 64 / 128 views.

**Frame-path memory.** The frame path's peak allocated memory (12.7 / 24.2 GB at 64 / 128 views, Table 11) contains three separately measured components: the decoded-row ring (251 MB at both lengths), the decode cache (1.8 / 3.5 GB), and the replay image buffer (19 / 39 MB). The remainder consists of cell statistics, source ownership, and rasterizer temporaries, which were not measured separately.

### B.3 Live execution

**Live runs.** Table 12 runs both paths against a real clock on the 20 RealEstate10K-64 timing scenes, one H100 per path. Frame $j$ arrives at $j/r$. Each item starts once its inputs arrive and the worker is free. *Queue* retains all inputs; *keep-latest* takes the newest frame or six-frame packet and discards older inputs. Official targets become due at their stream time plus the deadline. The same GPU renders them without pre-emption, so queries arriving during an item wait for its publication and

Table 10: **Streaming throughput, memory and map size.** 20 RealEstate10K scenes, one uncontended H100; input: mean ± s.d.; timed loops end before the final decode.

| Model | 64 views | | | | 128 views | | | |
|---|---|---|---|---|---|---|---|---|
| | Input (fps) | Memory (GB) | Gaussians (M) | Replay (ms) | Input (fps) | Memory (GB) | Gaussians (M) | Replay (ms) |
| ReCoSplat, chunk-8 | $56.0 \pm 0.7$ | 7.6 | 3.21 | – | $56.0 \pm 0.9$ | 8.4 | 6.42 | – |
| RetroSplat (no replay) | $43.5 \pm 2.1$ | 12.2 | 3.21 | – | $33.4 \pm 2.5$ | 23.8 | 6.42 | – |
| RetroSplat | $20.1 \pm 1.5$ | 12.5 | 3.21 | 149 | $13.8 \pm 1.3$ | 24.1 | 6.42 | 229 |

Table 11: **Frame-wise publication latency.** Boundary revision and replay are listed separately. RetroSplat on 20 RealEstate10K scenes per length, one H100; back-to-back inputs measure service time, not queueing latency (Section B.2).

| Service time (ms) | RealEstate10K-64 | | | RealEstate10K-128 | | |
|---|---|---|---|---|---|---|
| | p50 | p95 | max | p50 | p95 | max |
| Frames 1–2: history render, then wait for the clique | 5 | 6 | 11 | 6 | 9 | 18 |
| Frame 3: clique (joint bootstrap) to publication | 64 | 67 | 75 | 65 | 71 | 79 |
| Frames 4–6: one row, admission to publication | 56 | 59 | 69 | 58 | 62 | 73 |
| Boundary-decode work after frame 6 | 48 | 80 | 97 | 90 | 151 | 196 |
| Replay pass after frame 6 | 145 | 239 | 370 | 225 | 403 | 661 |
| All arrivals pooled (frame 6 incl. boundary and replay) | 56 | 285 | 529 | 57 | 447 | 927 |
| Work per six frames: frame path | | 442 | | | 562 | |
| Work per six frames: packet path | | 299 | | | 433 | |
| Frame-path input throughput (fps) | | 12.7 | | | 10.0 | |
| Frame-path peak allocated memory (GB) | | 12.7 | | | 24.2 | |

may receive a newer map. The frame path leads within capacity ($2.9\,\mathrm{dB}$ at $12.7\,\mathrm{fps}$ / $100\,\mathrm{ms}$); the packet path leads above it ($5.2$–$5.6\,\mathrm{dB}$ at $20\,\mathrm{fps}$ / $250\,\mathrm{ms}$).

The simulation replays uncontended service-time traces and serves the map available at the nominal query time. Frame-path live scores agree within about $1\,\mathrm{dB}$; packet-path scores are $0.9$–$3.7\,\mathrm{dB}$ higher live because pending queries receive the next publication. Query rendering costs $4$–$6\,\mathrm{ms}$ per $224^2$ target, under $1\%$ of pipeline work. Live-minus-simulated values are scene-paired means; session 2 repeats schedule-sensitive cells on a quieter host. Serving results for ReCoSplat and the 128-view protocol use simulation only.

**tttLRM-AR publication trace.** The released tttLRM-AR wrapper rebuilds the whole prefix for each publication. We measure prefixes $P = 4, 8, \ldots, 64$ on the 20 timing scenes, using one uncontended H100 and released $536\times960$ inputs (Fig. 4). Forward time grows linearly from $0.51\,\mathrm{s}$ at $P = 4$ to $4.20$ / $8.43\,\mathrm{s}$ at $P = 32$ / $64$; peak allocated memory grows from $6.2$ to $51.1\,\mathrm{GB}$. The $P = 64$ maps reproduce full-evaluation PSNR. Without compute constraints, the prefix maps score $27.24\,\mathrm{dB}$ online, exceeding our frame path's $24.40\,\mathrm{dB}$ by $2.84\,\mathrm{dB}$ $[+2.12, +3.71]$ and winning all 20 scenes; their final maps lead by $2.56\,\mathrm{dB}$.

The trace-driven simulation uses either every prefix in order or the largest unpublished complete prefix whenever the worker becomes free. Across $5$–$56\,\mathrm{fps}$ and $0$–$1000\,\mathrm{ms}$ deadlines, our frame path leads by $2.5$–$15.8\,\mathrm{dB}$ under both policies, with every scene-bootstrap interval excluding zero. Our packet path and ReCoSplat also lead throughout. At $12.7\,\mathrm{fps}$ / $250\,\mathrm{ms}$, latest-prefix tttLRM-AR serves $12.97\,\mathrm{dB}$ against $27.30\,\mathrm{dB}$ for our frame path: its first map appears at $0.74\,\mathrm{s}$, $14\%$ of targets have no map at query time, and published maps lag by 29 frames on average. Available-map targets alone score $14.02\,\mathrm{dB}$. Missing maps use the renderer's white background; black changes scores by at most $0.55\,\mathrm{dB}$. Dividing service times by the measured two-GPU speed-up of $1.59$ yields $14.52\,\mathrm{dB}$. These results simulate measured service times; tttLRM-AR was not run live, and its input contains $10.3\times$ as many pixels as ours.

Table 12: **Live reconstruction under fixed-rate arrivals.** Published maps, RealEstate10K-64, 20 timing scenes, one H100 per path; Sim.: trace-driven value of the same cell (Section B.3).

| Rate (fps) | Deadline (ms) | Policy | Path | Served PSNR (dB) Live | Sim. | Live − sim. | Query rendering Render (ms) mean / p95 | Late: render start $>5$ ms after the query time |
|---|---|---|---|---|---|---|---|---|
| *Session 1 (all cells)* | | | | | | | | |
| 5 | 250 | queue | frame | 25.84 | 25.52 | +0.32 | 5.4 / 8.2 | 19% |
| 5 | 250 | queue | packet | 22.07 | 21.18 | +0.90 | 5.2 / 7.3 | 19% |
| 12.7 | 100 | queue | frame | 25.12 | 24.07 | +1.04 | 4.9 / 7.0 | 78% |
| 12.7 | 100 | queue | packet | 22.18 | 19.95 | +2.23 | 4.7 / 7.0 | 54% |
| 12.7 | 250 | queue | frame | 27.09 | 27.30 | −0.20 | 5.5 / 7.4 | 84% |
| 12.7 | 250 | queue | packet | 26.14 | 23.25 | +2.88 | 5.0 / 7.2 | 57% |
| 20 | 250 | queue | frame | 22.65 | 22.69 | −0.04 | 4.7 / 6.5 | 85% |
| 20 | 250 | queue | packet | 27.89 | 24.37 | +3.52 | 5.4 / 7.4 | 80% |
| 30 | 250 | keep-latest | frame | 26.99 | 26.50 | +0.49 | 4.1 / 5.3 | 84% |
| 30 | 250 | keep-latest | packet | 28.59 | 25.27 | +3.32 | 5.1 / 6.9 | 91% |
| *Session 2 (repeat of the schedule-sensitive cells on a quieter host)* | | | | | | | | |
| 12.7 | 100 | queue | frame | 25.10 | 24.07 | +1.02 | 5.0 / 7.3 | 79% |
| 12.7 | 100 | queue | packet | 22.18 | 19.95 | +2.23 | 4.6 / 7.0 | 50% |
| 12.7 | 250 | queue | frame | 27.52 | 27.30 | +0.23 | 5.2 / 7.5 | 79% |
| 12.7 | 250 | queue | packet | 26.14 | 23.25 | +2.88 | 4.9 / 7.1 | 56% |
| 20 | 250 | queue | frame | 22.43 | 22.69 | −0.26 | 4.7 / 6.6 | 89% |
| 20 | 250 | queue | packet | 28.03 | 24.37 | +3.66 | 5.2 / 7.1 | 82% |

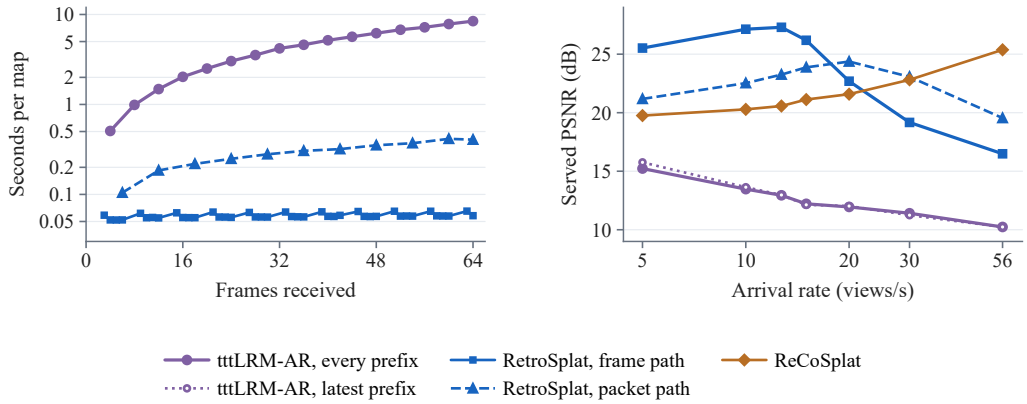



Figure 4: **Prefix reconstruction cost and simulated serving quality.** RealEstate10K-64, 20 timing scenes, one H100; tttLRM-AR at its released $536\times960$ input. Left: work per published map against frames received. Right: served PSNR at a 250 ms deadline against arrival rate, queue policy; dotted: tttLRM-AR publishing the latest complete prefix.

# C Controlled revision experiments

## C.1 Replay pathways with fixed weights

The fixed-weight controls in Table 2 use 160 RealEstate10K-64, 80 RealEstate10K-128 and 80 ACID-64 scenes, selected by a fixed pseudorandom ordering within each protocol. They share the RetroSplat checkpoint, context and target RGB and indices, replay cameras and packet schedule. Targets are averaged within scenes before paired differences.

Table 13: **Separately trained replay recipes.** Final-map PSNR (dB), with the same pretraining, clips and update budget. Fixed-checkpoint interventions appear in Table 2; matched adaptation in Table 14.

| Model | RE10K-64 (1,648 scenes) | RE10K-128 (1,648 scenes) | ACID-32 (752 scenes) | ACID-64 (752 scenes) |
|---|---|---|---|---|
| RetroSplat (no replay) | 24.92 | 24.12 | 25.98 | 25.29 |
| RetroSplat | **26.86** | **26.87** | **27.61** | **27.42** |
| Difference | +1.94 | +2.76 | +1.63 | +2.13 |

**Interventions.** *Replay evidence off* zeros the six replay columns at every decode in both pathways; replay still runs. *Opacity offset off* disables the explicit correction while retaining replay in the general branch. *Geometry and gate off* zeros the five geometric columns, preserving creation stamps and observation mass; the zero in-frustum count also closes the full-attribute gate. Each variant conditions later encoding and uses its own final decode (an identity for replay-off). These are closed-loop interventions, not additive pathway effects. With the opacity offset disabled, the general branch exceeds replay-off by 0.73 / 0.87 / 0.77 dB on RealEstate10K-64 / 128 and ACID-64. All three removals lower SSIM and raise LPIPS; geometry-and-gate removal gives worse LPIPS than replay removal on both RealEstate10K cohorts, despite a smaller PSNR loss. The age control subtracts one from the revision argument while preserving zero age for new cells; reported results retain the released age convention.

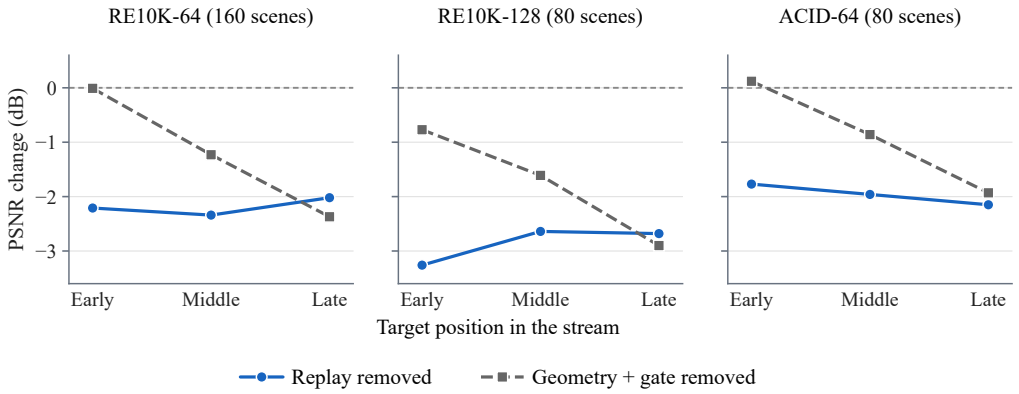



Figure 5: **Evidence contributions by target position.** Target-weighted PSNR change after removing each pathway at every decode, relative to the intact checkpoint. Early targets have more than $2/3$ of context frames after their source-frame index, late targets fewer than $1/3$, and middle targets the rest.

**Target position.** Replay removal lowers PSNR in every position group on all three protocols (Fig. 5). The final decode helps middle targets most, roughly doubling their replay-off effect relative to the last boundary map. Removing geometry and its gate affects late targets most; early-target changes are −0.01 / −0.77 / +0.12 dB on RealEstate10K-64 / 128 and ACID-64.

### C.2 Learning to read replay evidence

Both arms start from RetroSplat (no replay). We widen its evidence projection from 7 to 13 inputs with six zero columns and add an opacity head with a zero output layer, preserving the initial function. All 511 parent tensors (3,750,480 parameters) stay frozen. The six trainable tensors contain 2,777 parameters: the $13 \to 164$ projection and opacity head. R reads replay; M zeros all six replay columns during training and evaluation. Masking $n_c^p$ also disables the head, so M effectively adapts 1,312 projection parameters. The contrast tests replay access and head activation together.

Table 14: **Matched adaptation to replay evidence.** Final-map PSNR averaged over two matched runs; R reads replay and M zeros it during training and evaluation. Wins count scenes with positive R-minus-M PSNR in each run.

| Arm | RE10K-64 (160 scenes) | RE10K-128 (80 scenes) |
|---|---|---|
| Parent: RetroSplat (no replay) | 24.70 | 23.62 |
| M: replay zeroed | 24.71 | 23.63 |
| R: replay read | 26.08 | 25.25 |
| R − M | +1.37 | +1.62 |
| Wins | 160/160 | 80/80 |

Both arms share training data, clips, minibatch order, initialization and replay after every commit. AdamW uses learning rate $10^{-4}$, 500 warm-up updates and weight decay 0.01 on the projection only, on two H100 GPUs. We evaluate update 6,000, with two matched repetitions per arm, using the packet path and final decode (an identity for the parent and M).

R gains 1.36–1.39 / 1.60–1.64 dB over M on RealEstate10K-64 / 128, improving every scene in both runs; the two-run means are +1.37 / +1.62 dB (Table 14), with SSIM changes +.0272 / +.0468 and LPIPS changes −.0151 / −.0284. M gains only 0.01 dB over its parent. Masking R's replay columns at inference costs 1.32–1.35 / 1.54–1.58 dB, returning it near M and the parent. Jointly trained RetroSplat (36k updates) scores a further 0.63 / 1.19 dB above mean R; this is a full-training reference, not a controlled training increment.

### C.3 Direct optimization and the final decode

This control uses the first 80 of the 160 RealEstate10K-64 control scenes. We fix the network, RGB and indices, renderer and general revision branch, replacing only the explicit opacity head with persistent Adam offsets. The eight official targets are excluded from fitting and learning-rate selection.

**Same-base direct optimizer.** Offsets and Adam moments are keyed by live cell identity and reset when a key disappears or its slot is reused. Offsets are reapplied at boundary and final decodes. Both arms use the same replay cameras—the current six-frame packet and up to six earlier frames—and 126 render/backward pairs over eleven packets, each on its own current map. Adam takes one step per packet with a shared step counter. Its update modifies the current boundary map, one boundary before the learned decoder reads the same evidence; there is no final Adam step. Four training-validation scenes (128 context and 64 held-out frames each) select learning rate 0.3 from $\{0.3, 1, 3, 10\}$. A below-grid check at 0.1 gives 25.223 dB, below 25.711 at the selected rate.

RetroSplat exceeds head-off by 1.43 dB (80/80 scenes) and Adam by 0.28 dB (62/80), +.015 SSIM and −.020 LPIPS. The last boundary maps tie in PSNR (−0.00 dB); finalization adds +.39 versus +.10 dB. Adam needs 0.7 GiB of offset and moment state (11.74 versus 11.04 GiB peak allocated) and about 2% more stream time.

**Perceptual direct optimizer.** We also optimize $N_{\text{pix}}(\text{MSE}+\lambda\,\text{LPIPS})$ per replayed frame, where $N_{\text{pix}} = 3HW$ and LPIPS uses the same VGG network as training and evaluation. An extra VGG and rasterizer backward pass sends perceptual gradients only to Adam offsets. Final-map PSNR

Table 15: **Learned revision and direct opacity optimization.** Same base network, 80 RealEstate10K-64 scenes; final maps. Adam replaces only the explicit opacity head, with one step per packet. Time excludes target rendering.

| Method | PSNR↑ | SSIM↑ | LPIPS↓ | s/scene↓ | ΔPSNR: RetroSplat − row |
|---|---|---|---|---|---|
| RetroSplat, opacity head off | 25.19 | .878 | .138 | 3.50 | +1.43 |
| Head off + Adam (MSE) | 26.34 | .891 | .134 | 3.58 | +0.28 |
| Head off + Adam (MSE+LPIPS) | 25.60 | .878 | .118 | 4.94 | +1.02 |
| RetroSplat | 26.62 | .905 | .115 | 3.53 | – |

on the same four validation scenes selects $(\lambda, \mathrm{lr}) = (.05, .3)$ from $\{.05, .2, 1\} \times \{.3, 1\}$; this pair also has the grid's best LPIPS. Against squared-error Adam, it trades $-.74\,\mathrm{dB}$ for $-.016$ LPIPS. RetroSplat remains ahead by $1.02\,\mathrm{dB}$ ($80/80$ scenes), $+.027$ SSIM and $-.004$ LPIPS. Paired re-timing gives $4.84$ versus $3.59\,\mathrm{s}$ ($+35\%$), including $1.21\,\mathrm{s}$ of perceptual backward work; Table 15 retains original-session timings. These extra passes exceed the squared-error arm's compute budget.

Since the selected weight lies at the grid edge, we check $\lambda = .0125$ and $.003125$ at learning rate $.3$ on validation and test scenes. Test PSNR / SSIM / LPIPS are 26.02 / .885 / .117 and 26.27 / .889 / .119, respectively. Both settings remain below RetroSplat on all three metrics.

**Final decode.** One decoder call after the final replay consumes the new evidence without another render or backward pass, preserving the packet counter, ages, features and geometric evidence. Opacity changes in 57–95% of cells; changes to other attributes are smaller (median anchor displacement about $10^{-5}$ units, covariance change $1.4\%$, color harmonics $0.6\%$). On eight diagnostic scenes, an opacity-only decode recovers $+.43$ of the $+.46\,\mathrm{dB}$ gain, with unchanged peak memory. Across all nine protocols, finalization adds .29–.63 dB and improves 75–99% of scenes ($6/8$ on ScanNet-32; Table 16), leaving cross-system ordering unchanged. Uncontended re-timing on the

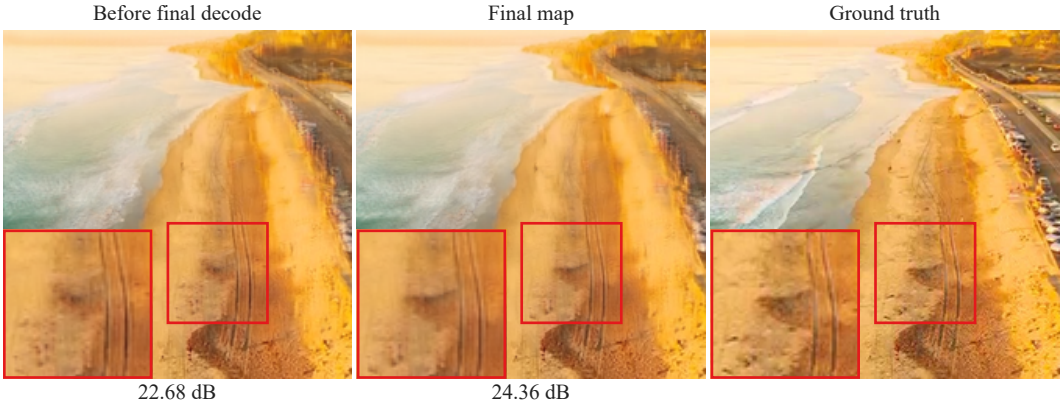



Figure 6: **The final decode revises a committed map with stored replay evidence.** An ACID scene after all 64 inputs; both columns use the same target camera, images and weights. Insets enlarge the red box $1.5\times$; PSNR on the full $224^2$ target.

20 scenes of Table 5, with two warm-ups and three timed passes, gives $0.105 \pm 0.001$ s at 64 views and $0.207 \pm 0.001$ s at 128 views ($\pm$: s.d. over per-scene means), or $3.1\%$ and $2.1\%$ of the stream loop.

## D SINGLE-MAP DIAGNOSIS

The provenance diagnosis uses 300 RealEstate10K-64 scenes and 2,400 targets on RetroSplat (no replay). For pixels with rendered alpha above $0.5$, we bucket the compositing-weighted mean creation packet $\bar{k}_p$ by $\mathrm{round}(\bar{k}_p) - k^\star$, where $k^\star$ is the packet of the target's pose-nearest context camera. Buckets at least two packets later cover $32\%$ of pixels and $52\%$ of squared error; $64\%$ of covered pixels mix creation packets with weighted standard deviation above one packet. The diagnosis uses no ground-truth depth.

Fitting one view-independent opacity per cell on 160 final RealEstate10K-64 maps of RetroSplat (no replay), using sixteen other in-span frames while fixing geometry and color, raises held-out PSNR by a median $3.4\,\mathrm{dB}$; 148 of the 160 scenes gain more than $2\,\mathrm{dB}$. The change is sparse: the fitted opacity exceeds the decoded opacity by more than $0.2$ for $0.8\%$ of cells and falls below it by more than $0.2$ for $0.1\%$. The sixteen fitting frames provide additional supervision, disjoint from context and evaluation frames. These results identify a substantial composition error that opacity alone can repair. This diagnostic uses a different cohort and fitting setup from Section C.3.

Table 16: **Contribution and cost of the final decode.** Last boundary map (B) against final map (F) of RetroSplat; wins: scenes with higher final-map PSNR, of $n$; extra time: seconds per scene and share of the stream loop ($^{\dagger}$: uncontended re-timing).

| | | PSNR↑ | | | | SSIM↑ | | LPIPS↓ | | Extra time | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Protocol | $n$ | B | F | ΔPSNR | Wins | B | F | B | F | s | % |
| RE10K-64 | 1,648 | 26.53 | 26.86 | +0.33 | 1,412 | .902 | .907 | .114 | .111 | 0.105 | 3.0 |
| RE10K-128 | 1,648 | 26.43 | 26.87 | +0.44 | 1,527 | .896 | .902 | .120 | .116 | $0.207^{\dagger}$ | $2.1^{\dagger}$ |
| ACID-32 | 752 | 27.04 | 27.61 | +0.57 | 729 | .849 | .860 | .164 | .158 | 0.053 | 3.6 |
| ACID-64 | 752 | 27.03 | 27.42 | +0.39 | 695 | .848 | .856 | .168 | .164 | 0.105 | 3.0 |
| ScanNet-32 | 8 | 23.50 | 23.79 | +0.29 | 6 | .771 | .778 | .267 | .261 | 0.052 | 2.9 |
| DL3DV-32 | 140 | 20.61 | 21.24 | +0.63 | 138 | .689 | .709 | .317 | .303 | 0.052 | 3.1 |
| DL3DV-64 | 140 | 21.59 | 21.99 | +0.40 | 131 | .732 | .745 | .282 | .273 | 0.107 | 2.5 |
| DL3DV-128 | 140 | 21.86 | 22.22 | +0.36 | 137 | .748 | .759 | .268 | .259 | 0.207 | 1.6 |
| DL3DV-256 | 140 | 21.60 | 21.95 | +0.35 | 126 | .739 | .749 | .277 | .269 | 0.413 | 1.0 |

Table 17: **Final-map quality by target position.** Target-weighted PSNR differences of RetroSplat's final map (and of its learned refit where named), 1,648 RealEstate10K scenes per length; early / late: more than two-thirds / fewer than one-third of context frames follow the target.

| Protocol | Comparison | Early | Middle | Late |
|---|---|---|---|---|
| RE10K-64 | RetroSplat − ReCoSplat | −0.10 | +0.96 | +2.20 |
| | RetroSplat − RetroSplat (no replay) | +2.86 | +2.51 | +0.55 |
| | RetroSplat − tttLRM-AR | −2.77 | −1.73 | −1.01 |
| | RetroSplat + learned refit − tttLRM-AR | −0.19 | +0.45 | +0.29 |
| | *tttLRM-AR (PSNR)* | *28.76* | *28.67* | *28.63* |
| RE10K-128 | RetroSplat − ReCoSplat | +0.23 | +1.43 | +3.95 |
| | RetroSplat − RetroSplat (no replay) | +4.19 | +3.18 | +1.12 |
| | RetroSplat − tttLRM-AR | +0.06 | +1.35 | +3.89 |
| | RetroSplat + learned refit − tttLRM-AR | +2.31 | +3.83 | +4.89 |
| | *tttLRM-AR (PSNR)* | *26.08* | *25.07* | *24.01* |

Table 17 splits the full-cohort final-map differences by target position (target-weighted means). Gains over ReCoSplat grow toward late targets, whereas gains over RetroSplat (no replay) are largest on early targets, which receive the most later evidence.

# E Cross-domain evaluation

## E.1 ScanNet and DL3DV

Table 18 evaluates the unchanged checkpoint at $224^2$: eight ScanNet scans with 32 context views, and 140 DL3DV scenes at each of 32, 64, 128 and 256 views, with eight targets per scene. DL3DV uses pose translation scale 0.5 and near/far bounds 0.5/100, calibrated on training scenes; ReCoSplat retains its native normalization and bounds. Frame indices and pixels match, while coordinate tensors remain method-specific. Training includes ScanNet for RetroSplat and DL3DV for ReCoSplat (Section A).

The separately trained replay recipe gains 1.24 dB over no replay on ScanNet and 2.5–3.3 dB across DL3DV lengths. All three metrics improve on every paired scene: eight ScanNet scans and $140/140$ DL3DV scenes per length. The final maps still trail ReCoSplat on ScanNet and at every DL3DV length (Table 18).

**Fine-tuning on DL3DV.** Adding 1,938 DL3DV training scenes to the original mixture, excluding every benchmark scene, we continue training for 10k updates to a fixed final checkpoint at 46k. Fine-tuning improves all three mean metrics, with PSNR gains of 1.28 / .97 / .60 / .24 dB at 32 / 64 / 128 / 256 views. The remaining deficits to ReCoSplat are 1.06 / .61 / .49 / .22 dB. PSNR stays

Table 18: **Cross-domain evaluation.** Final maps on ScanNet (8 scans) and DL3DV (140 scenes per length); ΔPSNR is the paired mean difference. Bold marks the best score.

| | | ReCoSplat | | | RetroSplat | | | RetroSplat − ReCoSplat |
|---|---|---|---|---|---|---|---|---|
| Dataset | Views | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | ΔPSNR |
| ScanNet | 32 | **24.06** | **.788** | **.224** | 23.79 | .778 | .261 | −0.27 |
| DL3DV | 32 | **23.57** | **.795** | **.160** | 21.24 | .709 | .303 | −2.33 |
| DL3DV | 64 | **23.57** | **.797** | **.163** | 21.99 | .745 | .273 | −1.58 |
| DL3DV | 128 | **23.32** | **.792** | **.171** | 22.22 | .759 | .259 | −1.10 |
| DL3DV | 256 | **22.41** | **.766** | **.196** | 21.95 | .749 | .269 | −0.46 |

lower and LPIPS higher at every length; SSIM is similar at 128 and 256 views. Other domains were not evaluated.

## F Incremental execution checks

With the same weights, causal neighbors and conditioning map, frame-wise and packet-wise execution agree on 40 RealEstate10K scenes per length: RetroSplat (no replay) differs by at most $6.6 \times 10^{-4}$ in mean relative depth error and $2.4 \times 10^{-3}$ in relative $L_2$ feature error; map PSNR differs by less than $0.01\,\mathrm{dB}$ on average and at most $0.04\,\mathrm{dB}$ per scene. Disabling TF32 reduces encoder residuals to $10^{-6}$ in a two-scene probe. For RetroSplat on the 20 timing scenes, frame-wise minus packet-wise PSNR is $-0.007$ / $-0.003\,\mathrm{dB}$ at 64 / 128 views (maximum absolute difference $0.024\,\mathrm{dB}$). The frame path publishes fresh cells between boundaries and stores fp16 replay images, with reservoir sampling above 128 frames; Section B.2 reports its publication cost.

## G Additional visual comparisons

**Rendering and selection.** All panels are unchanged $224^2$ archived renders; insets enlarge the same pixels and crops across methods. tttLRM-AR and OF$^3$GS use their released input sizes; OF$^3$GS uses context-camera alignment (Section A.1). The 64-view examples in Figs. 6 and 9 come from the first three scenes per dataset in a fixed pseudorandom ordering, independent of quality; all official targets were rendered. For 128 views, we rendered the twelve scenes closest to the cohort medians from 152 candidates: RetroSplat-minus-tttLRM-AR scene PSNR lies within $\pm 0.5\,\mathrm{dB}$ of $+1.90\,\mathrm{dB}$ at 128 views and within $\pm 0.75\,\mathrm{dB}$ of $-1.62\,\mathrm{dB}$ at 64 views. Figures 3 and 7 select individual targets with large PSNR gaps and clear visual differences from this pool, using four distinct scenes each. Eligibility uses eight-target scene means; these selected improvements do not replace the full-cohort results in Table 1. The teaser's $7.25\,\mathrm{dB}$ target gain is $2.09\,\mathrm{dB}$ over its scene. Exact selection and crop records accompany the figure sources.

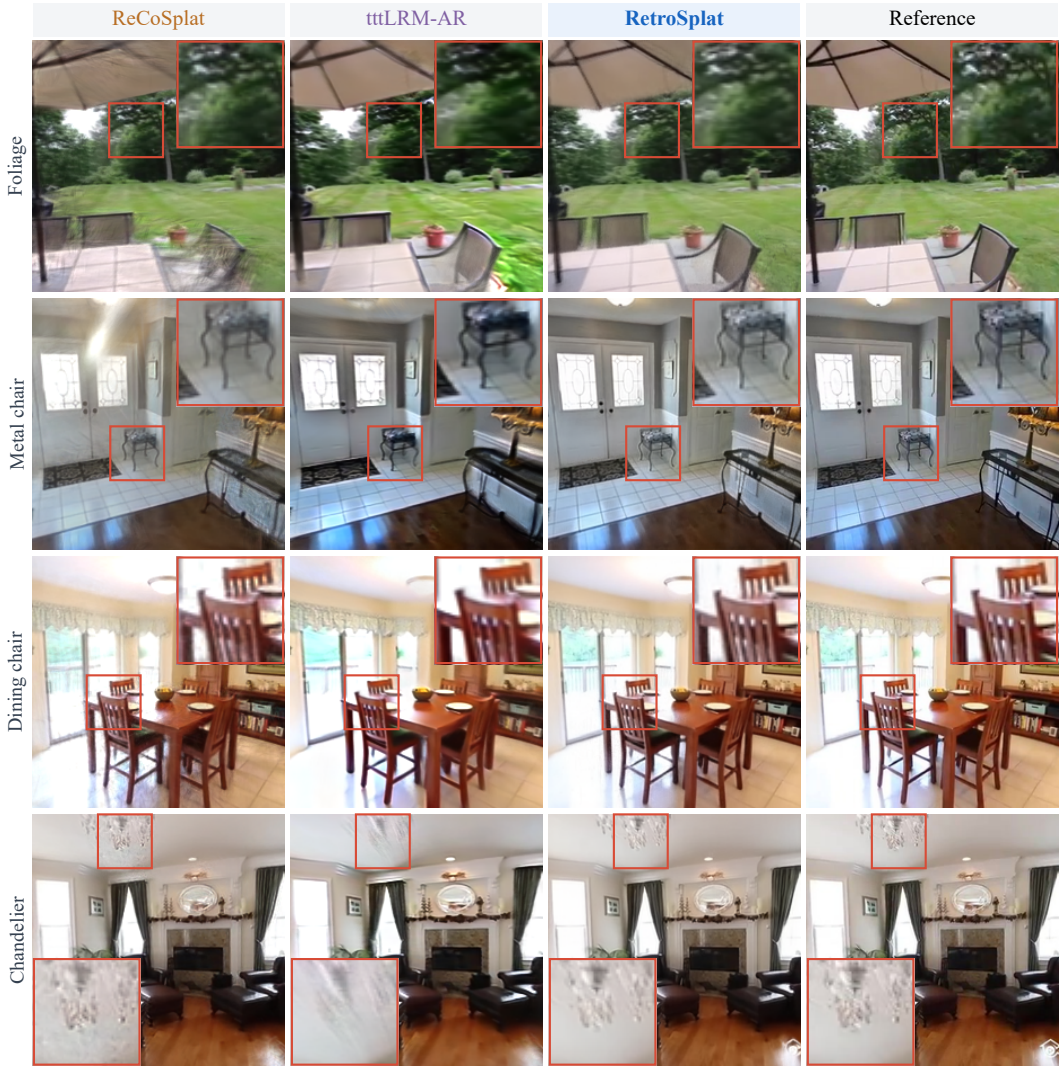



Figure 7: **Additional final-map comparisons at** 128 **inputs, without refit.** Four RE10K scenes complement Fig. 3: foliage, a metal chair, dining-chair slats and chandelier pendants. Red boxes are enlarged 2×, with the same crop and inset position across each row. These examples are selected for visible differences against tttLRM-AR (Section G).

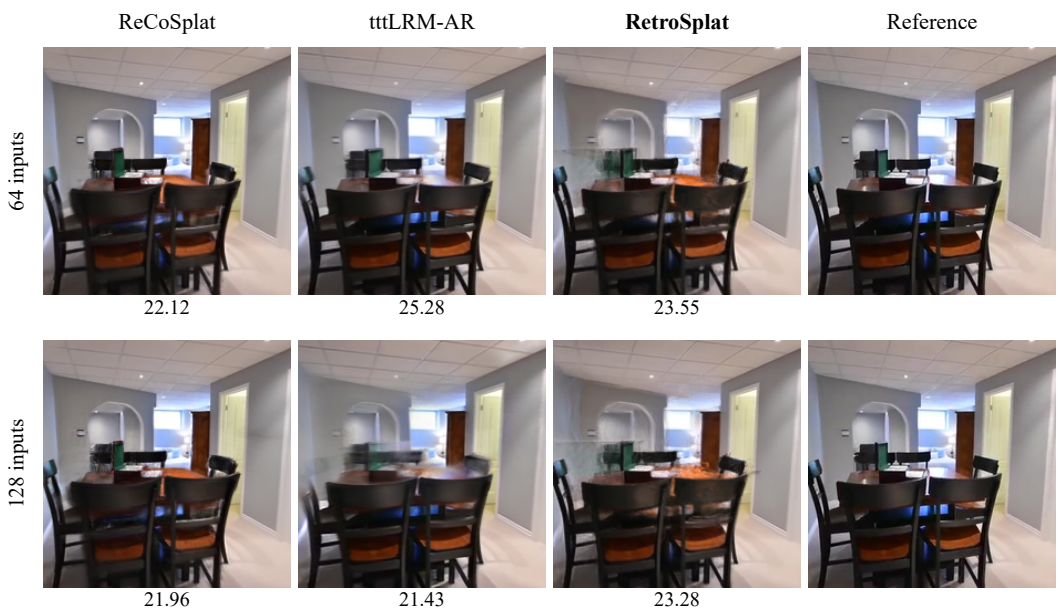



Figure 8: **Changing input density at a fixed camera.** The same dining-room view with 64 and 128 inputs from one scene span, without refit. RetroSplat retains floaters near the green box; its target PSNR decreases despite the stable cohort mean in Table 1. Numbers are target PSNR in dB.

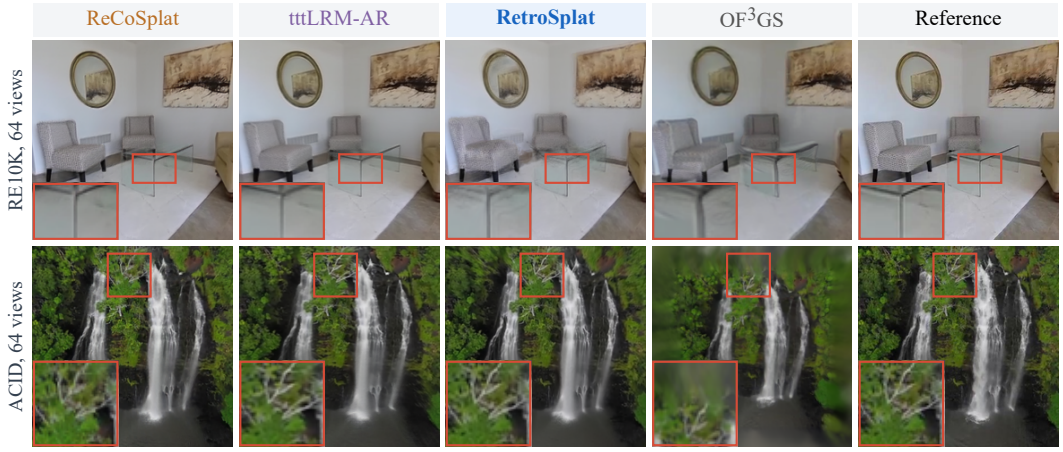



Figure 9: **A thin-structure failure and an ACID comparison.** Final maps from 64 input views, without refit. RetroSplat misses glass-table edges preserved by tttLRM-AR (top); the bottom row shows an ACID waterfall. Boxes are enlarged 2×; OF$^3$GS uses context-camera alignment.

# H LEARNED REFIT

**Procedure.** Refit changes only final-map opacity; network weights, means, covariances and color harmonics stay fixed. The logits are $u_c = \mathrm{logit}(\mathrm{clip}(\alpha_c, 10^{-4}, 1-10^{-4})) + \delta_c$, with $\delta_c$ initialized to zero and clipped to $[-8, 8]$. Round $k$ of $R$ renders sixteen context positions $\lfloor (jR+k-1)n/(16R) \rfloor$, $j = 0, \ldots, 15$, from $n$ frames. Each frame supplies one RGB-MSE render/backward pass; $\bar{\gamma}_c^{(k)}$ averages its gradients. Adam uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and bias correction. The step size is $\eta_0 \exp(\mathrm{clip}(R_\psi(f_c^{(k)}), -4, 2))$, with $\eta_0 = 0.3$.

**Features and training.** The 21 detached features are the thirteen evidence features in Table 8; Adam direction; nonzero-gradient fraction; gradient sign agreement (positive minus negative counts over their sum); $|\bar{\gamma}_c^{(k)}|$ and the root second moment, mapped by $(\log_{10}(x + 10^{-12}) + 12)/8$; $\tanh(\delta_c/2)$; opacity; and $k/R$. The 21–32–1 ReLU network has 737 parameters and a zero-initialized output, initially reproducing per-cell Adam. Training differentiates four refit rounds only through the step sizes, with gradients, directions and features held constant. Eight held-out targets, never context frames, define $\mathcal{L} = \mathcal{L}_4 + 0.3 \cdot \frac{1}{3} \sum_{k<4} \mathcal{L}_k$, where each $\mathcal{L}_k$ averages MSE plus $0.05$ LPIPS. Clips follow streaming training: 72 RealEstate10K context frames, with one clip in four from ScanNet training scans, excluding validation/test rooms. We train 2,200 steps on four H100s, one clip per GPU (8,800 clips), using AdamW at $10^{-3}$, no weight decay, 100 warm-up steps, cosine decay to $0.1$ of peak and gradient clipping at 1.

**Validation and depth.** On 14 RealEstate10K training-validation scenes with 64-frame prefixes, the zero-initialized network selects the base rate and inner LPIPS weight from $\{0.3, 1\} \times \{0, 0.01\}$ by PSNR, allowing at most $0.002$ LPIPS increase. This selects $\eta_0 = 0.3$ and RGB-only loss (28.41 dB, LPIPS .103); rate 1 reaches 28.89 dB but LPIPS .110. The same scenes select step 2,200 over 1,100 (28.764 versus 28.759 dB), preferring the later step within 0.02 dB. We then select among 4, 8, 12 and 16 rounds by PSNR subject to total time at most 7.58 s (90% of tttLRM-AR's time) and LPIPS at most $0.005$ above four rounds. Total combines median stream/final-decode time on the 20 timing scenes with median validation refit time; twelve rounds meet both limits (Table 19). Deployment times appear in Table 5.

Table 19: **Refit depth on the validation scenes.** 14 RealEstate10K training-validation scenes, 64-frame prefixes, median seconds per scene. Total: stream, final decode and refit. Limits: (a) total at most 7.58 s; (b) LPIPS at most .106.

| | | Adam | Learned refit | | | | |
|---|---|---|---|---|---|---|---|
| Rounds | Passes | PSNR↑ | PSNR↑ | LPIPS↓ | Refit s | Total s | Setting |
| 0 | 0 | – | 27.37 | .108 | 0 | 3.24 | final map |
| 4 | 64 | 28.41 | 28.76 | .101 | 1.41 | 4.65 | trained setting |
| 8 | 128 | 28.89 | 29.34 | .102 | 2.81 | 6.05 | – |
| 12 | 192 | 29.23 | 29.69 | .105 | 4.22 | 7.46 | selected |
| 16 | 256 | 29.49 | 29.94 | .107 | 5.62 | 8.85 | – |

**Full-cohort results.** The four-round learned and Adam refits use the same final maps, context views and backward-pass budgets (Table 20). Four learned rounds gain 1.21–1.47 dB, improving at least 97% of scenes in each protocol. Paired gains over Adam are $0.29$ $[0.28, 0.30]$, $0.32$ $[0.31, 0.34]$, $0.24$ $[0.22, 0.25]$ and $0.37$ dB $[0.34, 0.40]$ on RealEstate10K-64 / 128 and ACID-32 / 64. Both take 1.47 / 2.57 s at 64 / 128 views, with refit peak memory 3.2 / 5.8 GB including map and weights. Recomputed maps agree with Table 1 within $0.015$ dB per scene.

Twelve rounds add 1.85–2.17 dB over the final map and beat tttLRM-AR on all three metrics at RealEstate10K-128 and ACID-64. On RealEstate10K-64, the PSNR gain is $0.18$ dB $[0.08, 0.28]$, with equal SSIM (.920) and worse LPIPS (.111 versus .094). On ACID-32, PSNR differs by $0.02$ dB $[-0.14, 0.17]$, with lower SSIM (.884 versus .888) and worse LPIPS (.146 versus .134).

**Perceptual optimization control.** This untrained control uses 40 Adam steps at rate 0.3, MSE plus 0.1 LPIPS, and sixteen evenly spaced context frames (640 render/backward passes). It was

Table 20: **Post-stream opacity optimization on the full evaluation cohorts.** Final-map PSNR / LPIPS, full cohorts. Passes: context renders with a backward pass. s: seconds added per RealEstate10K-64 scene on the 20 timing scenes ([†]in its own full-cohort run).

| | | | RE10K-64 | RE10K-128 | ACID-32 | ACID-64 |
|---|---|---|---|---|---|---|
| Map | Passes | s | PSNR↑ / LPIPS↓ | | | |
| RetroSplat final map | – | – | 26.86 / .111 | 26.87 / .116 | 27.61 / .158 | 27.42 / .164 |
| + per-cell Adam, 4 rounds | 64 | 1.47 | 27.92 / .107 | 27.76 / .112 | 28.67 / .151 | 28.52 / .157 |
| + learned refit, 4 rounds | 64 | 1.47 | 28.21 / .106 | 28.09 / .109 | 28.91 / .149 | 28.89 / .151 |
| + learned refit, 12 rounds | 192 | 4.39 | 28.87 / .111 | 28.72 / .114 | 29.54 / .146 | 29.60 / .146 |
| + perceptual Adam, 40 steps | 640 | 16.3[†] | 28.10 / .073 | 28.02 / .076 | 28.68 / .092 | 28.60 / .092 |
| Learned minus Adam, ΔPSNR | | | +0.29 | +0.32 | +0.24 | +0.37 |
| tttLRM-AR (reference) | | | 28.69 / .094 | 25.02 / .161 | 29.52 / .134 | 28.69 / .155 |

selected from three candidates on 160 RealEstate10K-64 control scenes, a subset of the test cohort, by PSNR subject to at most 0.005 LPIPS increase. It yields the lowest LPIPS in Table 20, with lower PSNR than learned refit and 16.3 s per scene in its full-cohort run.

Table 21: **Step-size controls for twelve-round refit.** Twelve refit rounds of the same final map, control cohorts; Adam rows: per-cell Adam at base rate $\eta_0$. Δ: learned minus row, 95% interval (scenes where the learned rates have higher PSNR).

| Step size | PSNR↑ | SSIM↑ | LPIPS↓ | ΔPSNR (dB) | ΔLPIPS |
|---|---|---|---|---|---|
| *RealEstate10K-64* (160 scenes) | | | | | |
| Final map, no refit | 26.71 | .906 | .113 | – | – |
| Learned (ours) | 28.81 | .921 | .113 | – | – |
| Evidence shuffled | 28.87 | .913 | .127 | −0.06 [−0.09, −0.02] (49) | −0.015 [−0.017, −0.013] |
| Scalar schedule | 28.70 | .908 | .133 | +0.12 [+0.07, +0.17] (96) | −0.020 [−0.023, −0.018] |
| Adam, $\eta_0 = 0.3$ | 28.51 | .918 | .113 | +0.30 [+0.27, +0.33] (154) | −0.001 [−0.001, +0.000] |
| Adam, $\eta_0 = 1.0$ | 28.90 | .915 | .124 | −0.08 [−0.12, −0.05] (52) | −0.012 [−0.013, −0.010] |
| *ACID-32* (80 scenes) | | | | | |
| Final map, no refit | 27.98 | .863 | .158 | – | – |
| Learned (ours) | 29.92 | .886 | .144 | – | – |
| Evidence shuffled | 30.04 | .884 | .146 | −0.12 [−0.16, −0.08] (20) | −0.003 [−0.004, −0.001] |
| Scalar schedule | 29.96 | .881 | .148 | −0.03 [−0.08, +0.02] (37) | −0.004 [−0.006, −0.003] |
| Adam, $\eta_0 = 0.3$ | 29.68 | .883 | .146 | +0.25 [+0.20, +0.29] (72) | −0.003 [−0.003, −0.002] |
| Adam, $\eta_0 = 1.0$ | 30.02 | .884 | .145 | −0.09 [−0.13, −0.06] (23) | −0.002 [−0.003, −0.000] |

**Rate controls.** Table 21 uses the same final maps, context frames and twelve rounds on 160 RealEstate10K-64 and 80 ACID-32 scenes. The scalar schedule uses each round's median learned rate; evidence shuffling permutes only the thirteen stored-evidence features, leaving gradient and opacity features intact. Adam at $\eta_0 = 1.0$ exceeds learned rates by 0.08 / 0.09 dB, with LPIPS higher by 0.012 / 0.0015. Shuffling evidence raises PSNR by 0.06 / 0.12 dB and LPIPS by 0.015 / 0.003: it does not establish an independent PSNR benefit from stored evidence. Learned rates give the highest SSIM on both cohorts. At four or twelve rounds, final-round per-scene medians reach the clip ceiling in 82–98% of scenes across protocols; each round's per-cell first percentiles remain 4–10 times below the median.

# I SAMPLING DENSITY AND DOUBLED-SPAN STREAMS

**RealEstate10K thinning.** On the first 80 RealEstate10K-64 control scenes, thinning 128 inputs to 64 at positions $\lfloor i \cdot 127/63 + 0.5 \rfloor$ preserves both endpoints and targets. RetroSplat changes from 26.72 to 26.60 dB; tttLRM-AR improves from 25.21 to 28.69 dB. At 32 inputs they reach 26.37 / 29.07 dB. The separate 80-scene RealEstate10K-128 cohort confirms the pattern ($-0.18$ / $+3.44$ dB). Full-stream twelve-round refit reaches 28.66 dB: $-0.03$ dB $[-0.46, +0.40]$ versus 64-input tttLRM-AR, with LPIPS $+0.024$ $[0.016, 0.032]$; versus its 32-input map, the differences are $-0.41$ dB $[-0.86, +0.04]$ and $+0.029$ LPIPS.

**ACID-128 protocol.** We double the official ACID-64 span from its first context frame; clip length admits 107 of 752 scenes. We sample 128 inputs uniformly in camera-center arc length and eight non-context targets, one per eighth of the span, using fixed per-scene sampling. Median spans are 542 / 271 frames at ACID-128 / 64, with mean context gaps 4.49 / 4.53. Thinning to 64 uses the endpoint-preserving rule and the same targets. This protocol was fixed before evaluation. ACID-64 references use the same scenes but different official targets.

Table 22: **Reconstruction quality on doubled-span ACID streams.** 107 ACID scenes. $\Delta$: paired PSNR difference from RetroSplat's unrefitted ACID-128 map (scenes where the row is higher).

| | | ACID-64 | ACID-128 | | | $64 \to 128$ | |
|---|---|---|---|---|---|---|---|
| Method | Frames | PSNR↑ | PSNR↑ | SSIM↑ | LPIPS↓ | ΔPSNR | Δ vs RetroSplat |
| RetroSplat | 128 | 28.31 | 26.74 | .840 | .174 | −1.57 | – |
| + learned refit, 12 rounds | 128 | 30.70 | 28.71 | .874 | .155 | −1.99 | +1.97 (107) |
| + learned refit, 4 rounds | 128 | 29.89 | 28.01 | .864 | .161 | −1.88 | +1.27 (107) |
| tttLRM-AR | 128 | 29.21 | 25.58 | .807 | .219 | −3.63 | −1.16 (32) |
| tttLRM-AR, thinned | 64 | 29.21 | 28.68 | .881 | .141 | −0.53 | +1.94 (95) |
| ReCoSplat | 128 | 26.65 | 24.86 | .782 | .196 | −1.79 | −1.88 (10) |
| RetroSplat, thinned | 64 | – | 26.99 | .855 | .160 | – | +0.25 (70) |
| + learned refit, 12 rounds | 64 | – | 29.04 | .883 | .144 | – | +2.29 (107) |
| + learned refit, 4 rounds | 64 | – | 28.35 | .876 | .148 | – | +1.61 (106) |

**Density and span.** At 128 inputs, RetroSplat exceeds tttLRM-AR by 1.16 dB $[0.80, 1.52]$ and ReCoSplat by 1.88 dB (Table 22); twelve-round refit extends the former gap to 3.13 dB $[2.75, 3.51]$, improving 103 of 107 scenes. Thinning boosts tttLRM-AR by 3.10 dB $[2.91, 3.30]$. On the same thinned inputs, refitted RetroSplat leads by 0.35 dB $[0.05, 0.65]$, with LPIPS $+0.003$ $[-0.003, +0.009]$; using all 128 inputs instead gives $+0.02$ dB $[-0.30, +0.34]$ and worse LPIPS ($+0.014$ $[0.008, 0.021]$) against that thinned baseline.

Across spans at their native densities, unrefitted RetroSplat loses 2.07 dB $[1.74, 2.38]$ less than tttLRM-AR. A separate post hoc comparison fixes 64 inputs: twelve-round RetroSplat loses 1.66 dB $[1.38, 1.94]$, versus 0.53 dB for tttLRM-AR, retaining a 0.35 dB lead from its higher starting point. Targets differ across spans. Lower sensitivity to sampling density does not imply stable quality over a wider scene at fixed input count.