NEXEL-SLAM: DECOUPLING GEOMETRY AND APPEARANCE FOR DENSE RGB-D SLAM

Anonymous authors
Paper under double-blind review

ABSTRACT

High appearance complexity need not imply high geometric complexity, yet splatting-based SLAM often ties appearance detail to geometric discretisation. We introduce NeXeL-SLAM, an RGB-D SLAM system that combines explicit surface geometry with spatially varying neural appearance. Building on the nexel representation, we initialise oriented surfels from depth and augment their base colours with a shared neural texture field. Coarse world-space features share information across surfels, while fine surfel-local features attach texture to the evolving geometry. A common differentiable rasteriser supports tracking from depth and base colours, using analytic $\mathrm{SE}(3)$ pose gradients, and mapping with the full neural appearance. New surfels are introduced according to coverage and depth disagreement, while geometry and appearance are jointly refined over keyframes. On Replica, NeXeL-SLAM achieves $37.96\,\mathrm{dB}$ training-view PSNR, $0.59\,\mathrm{cm}$ mean Chamfer distance for the fused reconstruction, and $0.28\,\mathrm{cm}$ all-frame ATE-RMSE. On TUM RGB-D, it achieves $24.31\,\mathrm{dB}$ training-view PSNR, although fused-surface accuracy is less consistent. These results show that an explicit surface map can support detailed appearance alongside accurate reconstruction and tracking, while identifying remaining challenges in fusing real-sensor observations.

1 INTRODUCTION

Dense RGB-D SLAM requires a map that supports camera tracking, surface reconstruction, and high-quality rendering. Gaussian splatting (Kerbl et al., 2023) enables richly textured maps in systems such as MonoGS (Matsuki et al., 2024) and SplaTAM (Keetha et al., 2024), but its volumetric primitives do not explicitly define surfaces. In MonoGS, depth is an alpha-weighted accumulation of Gaussian-centre depths rather than ray–surface intersections. Good agreement with observed images therefore need not imply precise surface geometry.

Explicit surface primitives address this geometric limitation but can still couple texture resolution to geometric discretisation. Triangle Splatting SLAM (Fry et al., 2026), for example, interpolates vertex colours across triangular faces. A detailed poster on a planar wall may therefore require many small triangles despite its simple geometry. This motivates our central question: *can an online SLAM map combine explicit surface geometry with high-fidelity appearance without subdividing geometry solely to represent texture?*

We introduce **NeXeL-SLAM**, which adapts neurally textured surfels to online RGB-D tracking and mapping. Textured splatting methods allow appearance to vary within each primitive (Rong et al., 2025a;b); we build on the nexel representation (Rong et al., 2025b) to construct an evolving surface map from depth. Each nexel combines an oriented surface element and a base colour with a neural appearance residual. The shared field uses coarse world-space features and fine nexel-local features, keeping high-frequency detail attached to the surface as its geometry is refined. New elements are introduced using coverage and depth disagreement, while the field models texture within existing elements.

Tracking and mapping share a differentiable rasteriser but evaluate different appearance models. Tracking aligns depth and base colours with each incoming frame, without querying the neural field. Mapping holds camera poses fixed and jointly refines geometry, base colours, and neural appearance over keyframes. Following analytic pose differentiation in splatting-based SLAM (Matsuki et al.,

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.

2024; Fry et al., 2026), we derive the corresponding $\mathrm{SE}(3)$ pose Jacobians for nexel rendering and evaluate them within the rasteriser.

On Replica, NeXeL-SLAM combines $37.96\,\mathrm{dB}$ training-view PSNR with $0.28\,\mathrm{cm}$ all-frame ATE-RMSE. Its reconstructed surface achieves a mean Chamfer distance of $0.59\,\mathrm{cm}$ under TS-SLAM's TSDF-based protocol, compared with the $0.95\,\mathrm{cm}$ reported for TS-SLAM's TSDF reconstruction (Fry et al., 2026). On TUM RGB-D, the system achieves $24.31\,\mathrm{dB}$ training-view PSNR, although full-range depth and fused reconstruction show more mixed performance. These findings support the combination of explicit geometry and detailed neural appearance, while showing that accurate per-view predictions do not always translate into equally accurate fused surfaces.

Our contributions are as follows:

- An online neurally textured surfel map with depth-initialised geometry and hybrid world-space and nexel-local appearance, allowing texture variation within existing surface elements.
- A direct tracking and mapping formulation that uses base colours for pose estimation and neural textures for map refinement, with analytic $\mathrm{SE}(3)$ pose derivatives in a shared rasteriser.
- Empirical evidence that detailed appearance and accurate surface reconstruction can coexist with sub-centimetre tracking on Replica, together with an analysis of depth and reconstruction limitations on TUM RGB-D.

2 Related Work

Dense RGB-D and neural SLAM. Classical dense RGB-D systems construct geometric maps against which incoming frames are aligned. KinectFusion Newcombe et al. (2011) integrates depth into a volumetric signed-distance representation, while ElasticFusion Whelan et al. (2015) maintains an incrementally updated surfel map. Neural SLAM methods introduce learned scene representations into this tracking-and-mapping process. iMAP Sucar et al. (2021) represents a scene with an implicit network, and NICE-SLAM Zhu et al. (2022) uses hierarchical local features to support detailed reconstruction. ESLAM Johari et al. (2023) and Co-SLAM Wang et al. (2023) explore efficient feature-plane and hybrid encoding representations, respectively, while Point-SLAM Sandström et al. (2023) stores learned features in an adaptively expanded point cloud. These approaches illustrate how the organisation of learned scene information affects online reconstruction. NeXeL-SLAM retains explicit surface primitives for geometry and uses the neural field to model appearance across their support.

Splatting-based SLAM and surface primitives. The differentiable rasterisation of 3D Gaussian Splatting Kerbl et al. (2023) has enabled several approaches to rendering-based SLAM. SplaTAM Keetha et al. (2024) uses a Gaussian map for RGB-D tracking and mapping, with rendered silhouette and depth information guiding map expansion. GS-SLAM Yan et al. (2024) combines adaptive Gaussian expansion with coarse-to-fine pose estimation. Gaussian Splatting SLAM (MonoGS) Matsuki et al. (2024) uses a common Gaussian representation for tracking and mapping and derives analytic camera-pose Jacobians within its rendering pipeline. Photo-SLAM Huang et al. (2024b) instead combines feature-based localisation with Gaussian-based photorealistic mapping. Our system follows the direct, map-based tracking formulation, but evaluates spatially varying neural appearance on explicit surface elements.

Surface-based splatting provides an alternative to volumetric Gaussian primitives. 2D Gaussian Splatting Huang et al. (2024a) uses oriented planar Gaussians and ray–splat intersections, and 2DGS-SLAM Zhong et al. (2026) incorporates this representation into RGB-D SLAM. Triangle Splatting SLAM Fry et al. (2026) uses differentiable triangles for tracking and mapping and supports conversion to a connected mesh. These works establish that explicit surface primitives can serve as SLAM maps. Our focus is not the use of surface primitives alone, but their combination with within-primitive neural appearance and an online optimisation procedure for the resulting map.

Neural appearance and textured surface representations. Learned appearance on explicit geometry predates recent splatting methods. DeepSurfels Mihajlovic et al. (2021) introduces learned

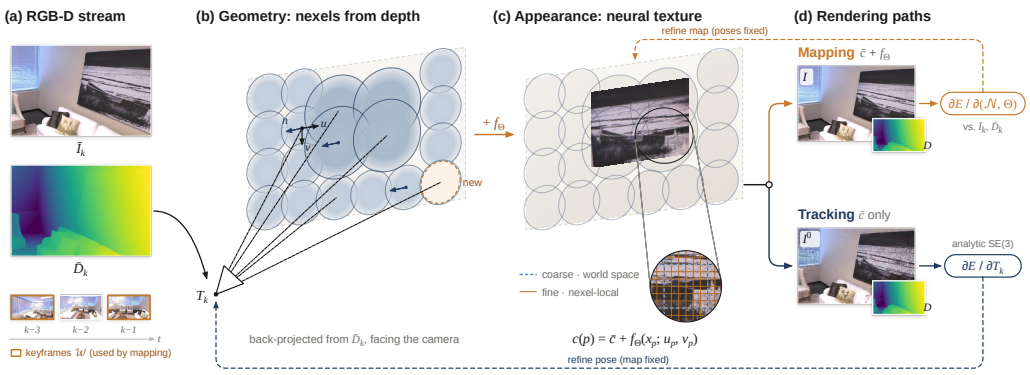


Figure 1: **NeXeL-SLAM overview.** (a–b) RGB-D observations initialise surface elements (nexels) by depth back-projection. (c) A shared neural texture field augments base colours with coarse world-space and fine nexel-local features. (d) Mapping refines geometry and appearance using textured colour and depth, with camera poses fixed. Tracking updates poses from base-colour and depth renderings using analytic $\mathrm{SE}(3)$ gradients, with the map fixed.

online appearance fusion, while SurfelNeRF Gao et al. (2023) incrementally reconstructs neural surfel radiance fields using supplied camera parameters. These methods are relevant precedents for online neural appearance modelling; the distinction in our setting is that camera poses must also be estimated directly against the rendered map. Compact 3DGS Lee et al. (2024) uses a grid-based neural field to encode view-dependent colour compactly, but predicts appearance at the Gaussian level rather than modelling texture variation across each primitive's surface.

Texture-GS Xu et al. (2024) separates texture from Gaussian geometry through a learned texture mapping, while GStex Rong et al. (2025a) attaches explicit textures to individual 2D Gaussians. BBSplat Svitov et al. (2025) similarly uses planar primitives with learnable colour and opacity textures. NeST-Splatting Zhang et al. (2025) represents appearance with a global neural shell texture, allowing fine detail to be modelled without a corresponding increase in primitive density. Most directly related to our representation, Nexels Rong et al. (2025b) combines surfel geometry with a shared world-space neural texture field for novel-view synthesis.

NeXeL-SLAM adapts this principle to a map that is constructed from sequential RGB-D observations and used immediately for pose estimation. Relative to the posed-image optimisation in Nexels, our formulation introduces direct RGB-D tracking, depth-initialised incremental map construction, and joint surface–appearance refinement over keyframes. The hybrid world-space and nexel-local appearance queries address the changing geometry of this online map, while pose differentiation makes the textured representation itself available for direct alignment.

3 Method

NeXeL-SLAM alternates camera tracking and map refinement using a shared nexel representation and differentiable rasteriser (Fig. 1). Tracking estimates poses from depth and base-colour renderings; mapping holds these poses fixed and jointly optimises geometry, base colours, and neural appearance over keyframes.

3.1 Nexels: Neurally-Textured Surfels

Geometry. Following the nexel representation (Rong et al., 2025b), the map is a set of surface elements $\mathcal{N} = \{\mathcal{N}^i\}$. Each nexel is a planar disk with centre $\boldsymbol{\mu}^i \in \mathbb{R}^3$, unit tangent axes $\mathbf{a}_u^i, \mathbf{a}_v^i$, normal $\mathbf{n}^i = \mathbf{a}_u^i \times \mathbf{a}_v^i$, per-axis radii $\mathbf{s}^i = (s_u^i, s_v^i) \in \mathbb{R}^2_{>0}$, opacity $o^i \in [0, 1]$, shape exponents $\boldsymbol{\gamma}^i = (\gamma_u^i, \gamma_v^i)$ with $\gamma \geq 1$, and base colour $\bar{\mathbf{c}}^i \in \mathbb{R}^3$.

Ray–disk kernel. Under the world-to-camera pose $\mathbf{T}_{CW}$, the ray $\mathbf{r}_p$ at pixel p intersects the plane of nexel $\mathcal{N}^i$ at $\mathbf{x}_p^i$. Its normalised tangent coordinates are

$$u_p^i = \frac{\langle \mathbf{x}_p^i - \boldsymbol{\mu}^i, \mathbf{a}_u^i \rangle}{s_u^i}, \qquad v_p^i = \frac{\langle \mathbf{x}_p^i - \boldsymbol{\mu}^i, \mathbf{a}_v^i \rangle}{s_v^i}. \tag{1}$$

The contribution is weighted by a generalised Gaussian kernel,

$$G^i(p) = \exp\Big(-\tfrac{1}{2}\big[(u_p^i)^{2\gamma_u^i} + (v_p^i)^{2\gamma_v^i}\big]\Big), \qquad \alpha^i(p) = o^i\, G^i(p). \tag{2}$$

For $\gamma = 1$, this recovers the planar Gaussian kernel of 2D Gaussian Splatting (Huang et al., 2024a); larger values flatten the interior and sharpen the falloff. The exponents are optimised with the other nexel parameters.

Appearance. A shared neural texture field f_Θ, implemented as a multiresolution hash grid and a small MLP, predicts a spatially varying residual to each nexel's base colour. Coarse levels query the world-space hit point $\mathbf{x}_p^i$, sharing appearance across neighbouring nexels. Fine levels query tangent coordinates (u_p^i, v_p^i) within a hash-domain cell reserved for each nexel, keeping fine detail attached to the surface as its geometry is refined. The resulting colour is

$$\mathbf{c}^i(p) = \bar{\mathbf{c}}^i + f_\Theta\big(\mathbf{x}_p^i;\, u_p^i, v_p^i\big). \tag{3}$$

This separation concerns representational capacity: the field can express within-surfel colour variation without subdivision, while photometric mapping still jointly updates geometry and appearance.

Rendering. For each pixel p, we sort the contributing nexels $\mathcal{N}_p$ from front to back and alpha-composite colour, coverage, and depth:

$$\mathbf{C}_p = \sum_{i \in \mathcal{N}_p} \mathbf{c}^i(p)\, \alpha^i(p) \prod_{j<i} \big(1 - \alpha^j(p)\big), \tag{4}$$

$$A_p = \sum_{i \in \mathcal{N}_p} \alpha^i(p) \prod_{j<i} \big(1 - \alpha^j(p)\big), \qquad D_p = \frac{1}{A_p} \sum_{i \in \mathcal{N}_p} z_p^i\, \alpha^i(p) \prod_{j<i} \big(1 - \alpha^j(p)\big), \tag{5}$$

where z_p^i is the camera-frame depth of $\mathbf{x}_p^i$. Thus, D_p is a coverage-normalised average of ray–surface intersection depths. We denote the textured image, coverage, and depth maps by $I(\mathcal{N}, \mathbf{T}_{CW})$, $A(\mathcal{N}, \mathbf{T}_{CW})$, and $D(\mathcal{N}, \mathbf{T}_{CW})$. The base-colour image $I^0(\mathcal{N}, \mathbf{T}_{CW})$ sets $\mathbf{c}^i(p) = \bar{\mathbf{c}}^i$ and requires no neural field queries. Observed colour and depth images are denoted by $\bar{I}$, $\bar{D}$.

3.2 Analytic Pose Jacobian on $\mathrm{SE}(3)$

Following prior splatting-based SLAM systems (Matsuki et al., 2024; Fry et al., 2026), we evaluate analytic pose derivatives inside the rasteriser's backward pass. Here, $\mathbf{C}_p$ denotes the base-colour evaluation used for tracking. For $\mathbf{T}_{CW} \in \mathrm{SE}(3)$ and the six-dimensional increment $\boldsymbol{\tau} = (\boldsymbol{\rho}, \boldsymbol{\theta}) \in \mathfrak{se}(3)$, we use the left-composed update

$$\mathbf{T}_{CW} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}) \circ \mathbf{T}_{CW}, \qquad \mathrm{Exp}(\boldsymbol{\tau}) = \begin{bmatrix} \mathrm{Exp}(\boldsymbol{\theta}^\wedge) & \boldsymbol{\rho} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{6}$$

and write the manifold derivative as

$$\frac{\mathcal{D} f(\mathbf{T}_{CW})}{\mathcal{D} \mathbf{T}_{CW}} \triangleq \lim_{\boldsymbol{\tau} \to 0} \frac{\mathrm{Log}\big(f(\mathrm{Exp}(\boldsymbol{\tau}) \circ \mathbf{T}_{CW}) \circ f(\mathbf{T}_{CW})^{-1}\big)}{\boldsymbol{\tau}}, \tag{7}$$

The local pose Jacobians of a point $\boldsymbol{\mu}_C = \mathbf{T}_{CW} \boldsymbol{\mu}_W$ and a direction $\mathbf{n}_C = \mathbf{R}\, \mathbf{n}_W$ in the camera frame are

$$\frac{\mathcal{D} \boldsymbol{\mu}_C}{\mathcal{D} \mathbf{T}_{CW}} = \big[\, \mathbf{I} \;\; -\boldsymbol{\mu}_C^\times \,\big], \qquad \frac{\mathcal{D} \mathbf{n}_C}{\mathcal{D} \mathbf{T}_{CW}} = \big[\, \mathbf{0} \;\; -\mathbf{n}_C^\times \,\big], \tag{8}$$

where $^\times$ denotes the skew-symmetric matrix.

Pose changes affect the ray–disk intersection, its tangent coordinates, and its camera-frame depth. Applying the chain rule through Eqs. (1)–(5) gives

$$\frac{\partial E}{\partial \boldsymbol{\tau}} = \sum_p \left[\frac{\partial E}{\partial \mathbf{C}_p} \frac{\partial \mathbf{C}_p}{\partial \boldsymbol{\tau}} + \frac{\partial E}{\partial D_p} \frac{\partial D_p}{\partial \boldsymbol{\tau}} \right], \qquad \frac{\partial \mathbf{C}_p}{\partial \boldsymbol{\tau}} = \sum_{i \in \mathcal{N}_p} \frac{\partial \mathbf{C}_p}{\partial (u_p^i, v_p^i, z_p^i)} \frac{\partial (u_p^i, v_p^i, z_p^i)}{\partial \mathbf{x}_p^i} \frac{\mathcal{D}\mathbf{x}_p^i}{\mathcal{D}\mathbf{T}_{CW}}, \tag{9}$$

where $\mathcal{D}\mathbf{x}_p^i / \mathcal{D}\mathbf{T}_{CW}$ follows from Eq. (8) and the intersection geometry. The rasteriser accumulates the resulting colour and depth derivatives.

3.3 Tracking

With the map fixed, we estimate pose $\mathbf{T}_{CW}^k$ of frame k by aligning base-colour and depth renderings with the RGB-D observations:

$$E_{\text{track}}(\mathbf{T}_{CW}) = \lambda_{\text{pho}} \sum_{p \in \mathcal{M}} \left\| I_p^0(\mathcal{N}, \mathbf{T}_{CW}) - \bar{I}_p \right\|_1 + \lambda_{\text{geo}} \sum_{p \in \mathcal{M}} \left| D_p(\mathcal{N}, \mathbf{T}_{CW}) - \bar{D}_p \right|, \tag{10}$$

where $\mathcal{M}$ is the set of pixels selected for alignment.

Tracking with base colours. The photometric term is deliberately evaluated without the neural residual. During online mapping, errors in the texture field can contribute to image disagreement independently of camera motion; evaluating the field at pose-dependent hit points would also require its coordinate derivatives during tracking. Using base colours avoids these queries and derivatives, but sacrifices the within-surfel texture cues available in the full rendering. We therefore combine this photometric term with depth alignment. The two optimisation stages remain connected: mapping updates the geometry and base colours used by subsequent tracking steps.

Pixel selection. Let $\mathcal{V}$ contain pixels with valid depth measurements. The mask $\mathcal{M}$ selects pixels in $\mathcal{V}$ whose coverage A_p exceeds a threshold and whose depth residual is not an outlier relative to the median. With valid-depth statistics enabled, the median is computed over $\mathcal{V}$; if fewer than $\rho|\mathcal{V}|$ pixels remain, we instead retain the best-covered $\rho|\mathcal{V}|$ valid pixels. Both the outlier statistic and fallback size then exclude invalid measurements.

Optimisation. We minimise Eq. (10) by first-order descent on $\boldsymbol{\tau}$ using the analytic gradient from Sec. 3.2. After each step, we apply Eq. (6) and reset $\boldsymbol{\tau}$ to zero. We retain the pose with the lowest objective value across iterations.

Initialisation. Tracking starts from a constant-velocity extrapolation of the two previous pose estimates,

$$\mathbf{T}_{CW}^{k,(0)} = \mathbf{T}_{CW}^{k-1} \left(\mathbf{T}_{CW}^{k-2}\right)^{-1} \mathbf{T}_{CW}^{k-1}, \tag{11}$$

When enabled, a translation cap limits the extrapolation to a multiple of the recent median translation, subject to an absolute ceiling. The previous-pose alternative is $\mathbf{T}_{CW}^{k,(0)} = \mathbf{T}_{CW}^{k-1}$.

Residual weighting. The ratio $\lambda_{\text{pho}}/\lambda_{\text{geo}}$ balances appearance and depth alignment. Since base colours cannot explain all observed appearance variation, we favour the geometric term, accounting for depth reliability. Weights and dataset-specific tracking settings are given in Sec. 4.

3.4 Keyframing and Mapping

We initialise the map from the first depth image using the nexel creation procedure below and register keyframes at a fixed cadence.

Mapping window. The local window $\mathcal{W}$ contains the current frame, the most recent keyframe, and past keyframes with the greatest overlap, estimated by re-projecting a sparse sample of the current depth into each candidate view. Each iteration renders one selected frame from this window; a fixed fraction instead samples a keyframe from the full history, independently of current-view overlap, to supervise regions outside the local window.

Densification and pruning. We add nexels at valid-depth pixels with insufficient coverage A_p or where the rendered surface lies substantially behind the observed depth. Each new nexel is back-projected from its depth measurement, with a camera-facing normal and radii determined by the pixel's projected footprint. We cap insertions per frame to control map growth and prune nexels with low opacity or excessive size. Densification uses no explicit photometric-error threshold.

Mapping objective. With camera poses fixed, we optimise the nexel parameters and field weights Θ using

$$E_{\text{map}} = \sum_{k \in \mathcal{W}} \left[\lambda_{\text{pho}} E^k_{\text{pho}} + \lambda_{\text{geo}} E^k_{\text{geo}} \right] + \lambda_{\text{tex}} E_{\text{tex}} + \lambda_{\text{grid}} E_{\text{grid}}, \tag{12}$$

where the photometric term uses the full textured rendering and combines ℓ_1 and D-SSIM losses (Kerbl et al., 2023), $E^k_{\text{pho}} = (1-\lambda)\|I(\mathcal{N}, \mathbf{T}^k_{CW}) - \bar{I}_k\|_1 + \lambda\, \mathcal{L}_{\text{D-SSIM}}(I(\mathcal{N}, \mathbf{T}^k_{CW}), \bar{I}_k)$. The geometric term is the masked ℓ_1 depth residual from Eq. (10). The auxiliary term E_{tex} compares the coverage-normalised field colour with the observed image. Finally, E_{grid} is a resolution-weighted ℓ_2 penalty on hash-grid features, discouraging fine levels from representing variation that coarser levels can capture.

4 Experimental Results

We evaluate whether NeXeL-SLAM combines high-fidelity appearance, accurate surface reconstruction, and camera tracking on Replica and TUM RGB-D.

4.1 Experimental Setup

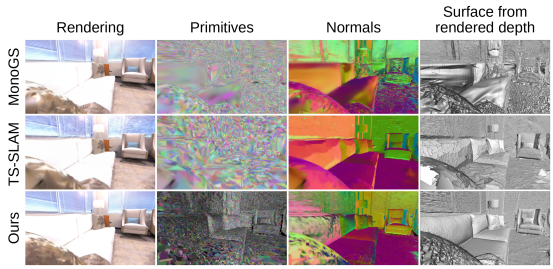
Datasets and baselines. We use the eight Replica sequences (Straub et al., 2019) distributed with NICE-SLAM (Zhu et al., 2022), each containing 2,000 RGB-D frames, and three TUM RGB-D sequences (Sturm et al., 2012): `fr1/desk`, `fr2/xyz`, and the long-office sequence, abbreviated as `fr3/office`. We rerun SplaTAM (Keetha et al., 2024), MonoGS (Matsuki et al., 2024), ESLAM (Johari et al., 2023), and Co-SLAM (Wang et al., 2023) on an NVIDIA A100 with common scoring code; MonoGS uses its single-process configuration. DROID-SLAM (Teed & Deng, 2021) provides an RGB-D tracking reference with a different TUM frame schedule (Appendix A). For surface reconstruction, we additionally rerun TS-SLAM (Fry et al., 2026) and MonoGS on TUM and distinguish rerun results from published Replica values in Table 2.

Metrics. Tracking uses camera-centre ATE-RMSE after rigid alignment without scale fitting, evaluated over all tracked frames. Table 1 reports PSNR and full-range depth ℓ_1 on every fifth input frame at each method's estimated pose, under the valid-reference-depth mask. Methods retain their native depth renderers; NeXeL-SLAM uses Eq. (5) and the full neural appearance for colour evaluation. These scores measure training-view fidelity, not held-out-view quality. Reconstruction is evaluated separately in Sec. 4.3. Dataset scores are means over sequences; per-sequence and keyframe-only tracking results are in Appendix A.

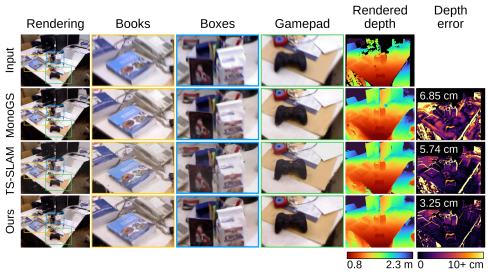
Evaluation settings. The primary comparison uses online maps without offline colour refinement. ESLAM and Co-SLAM receive no reference depth for evaluation-time ray sampling, although all systems use RGB-D observations for SLAM. Alternative rendering settings are reported separately. All primary results use one run per sequence except MonoGS on Replica, which averages three runs. NeXeL-SLAM uses $\lambda_{\text{geo}} = 1$ and $\lambda_{\text{pho}} = 0.5$ in both tracking and mapping. Tracking uses 40 iterations per frame on Replica and 200 on TUM; the velocity cap and valid-pixel statistics of Sec. 3.3 are enabled only on TUM.

4.2 Rendering and Depth Fidelity

Replica. NeXeL-SLAM achieves the highest training-view PSNR among the evaluated online maps, at $37.96\,\text{dB}$: $2.61\,\text{dB}$ above MonoGS and $3.74\,\text{dB}$ above SplaTAM (Table 1). Its mean depth error is $0.18\,\text{cm}$, compared with $2.21\,\text{cm}$ and $0.71\,\text{cm}$, respectively. Thus, detailed appearance coexists with close agreement to the observed depth in the same explicit surface map.



(a) Map structure from a novel viewpoint (Replica `room0`).



(b) Rendering and depth on TUM `fr1/desk`.

Figure 2: **Qualitative comparisons.** (a) Primitives, normals, and surfaces derived from rendered depth at a novel viewpoint. (b) Colour and depth rendered at each method's estimated pose; annotated values are per-frame mean depth errors against sensor depth.

Table 1: Joint evaluation on Replica and TUM RGB-D. ATE-RMSE and full-range depth ℓ_1 are in centimetres; PSNR is in decibels. ATE uses all tracked frames; rendering is scored every fifth frame at estimated poses. Baselines are reruns; DROID-SLAM uses a different TUM frame schedule. Bold marks the best numerical entry, not significance.

	Replica			TUM RGB-D		
Method	ATE ↓	PSNR ↑	Depth ↓	ATE ↓	PSNR ↑	Depth ↓
SplaTAM	0.386	34.22	0.71	2.092	23.56	**3.49**
MonoGS	0.306	35.35	2.21	1.628	18.99	5.65
ESLAM	0.578	23.65	4.73	2.114	17.90	12.01
Co-SLAM	1.103	28.09	9.56	2.548	20.16	14.33
DROID-SLAM	0.426	–	–	**1.528**	–	–
NeXeL-SLAM	**0.28**	**37.96**	**0.18**	2.02	**24.31**	9.92

TUM RGB-D. NeXeL-SLAM also has the highest PSNR under the primary rendering setting, at $24.31\,\mathrm{dB}$, exceeding SplaTAM by $0.75\,\mathrm{dB}$. Its full-range depth error of $9.92\,\mathrm{cm}$ is lower than those of ESLAM and Co-SLAM but higher than those of MonoGS and SplaTAM. This contrasts with the results restricted to $3\,\mathrm{m}$ in Sec. 4.3, highlighting the importance of reporting the evaluated depth range.

Additional refinement. MonoGS's offline colour refinement raises its PSNR to $39.11\,\mathrm{dB}$ on Replica and $26.48\,\mathrm{dB}$ on TUM, exceeding NeXeL-SLAM in both cases. These results and the reference-depth-guided rendering scores for ESLAM and Co-SLAM are reported separately in Appendix A.

4.3 Geometric Accuracy

Reconstruction. We evaluate reconstructed surfaces against a reference mesh fused from sensor depth at ground-truth poses, following TS-SLAM's Chamfer protocol (Fry et al., 2026). On Replica, NeXeL-SLAM achieves a mean Chamfer distance of $0.59\,\mathrm{cm}$, compared with the published $0.95\,\mathrm{cm}$ for TS-SLAM's TSDF reconstruction, with lower error on every scene (Table 2). On average, 91.6% of the reconstructed surface lies within $1\,\mathrm{cm}$ of the reference. Rendering at ground-truth rather than estimated poses gives a mean depth error of $0.42\,\mathrm{cm}$, providing a further check beyond agreement with the estimated trajectory.

Surface structure and loss weighting. Figure 2a shows the maps from a novel viewpoint near the sofa. MonoGS produces uneven depth-derived surfaces, while TS-SLAM shows faceting and gaps; NeXeL-SLAM better resolves the cushions and creases in this view. Under one evaluation implementation on `room0`, Chamfer distances are $1.09\,\mathrm{cm}$ for MonoGS's online map, $0.94\,\mathrm{cm}$ for TS-SLAM, and $0.68\,\mathrm{cm}$ for NeXeL-SLAM. The MonoGS rerun differs from the published $3.76\,\mathrm{cm}$

Table 2: Geometric accuracy in centimetres (lower is better), following TS-SLAM (Fry et al., 2026). Chamfer compares each reconstruction with a TSDF reference fused from sensor depth at ground-truth poses; predicted TSDF meshes use rendered depth at estimated poses. TUM depth ℓ_1 scoring and fusion use a 3 m cutoff. NeXeL-SLAM is fused and scored on TS-SLAM's keyframes, sharing its reference mesh. Replica scores average eight scenes. †Published values from Fry et al. (2026); all other entries are reruns under one implementation.

	Replica	`fr1/desk`		`fr2/xyz`		`fr3/office`	
Method	Chamf.	Depth	Chamf.	Depth	Chamf.	Depth	Chamf.
MonoGS	4.03†	4.01	**2.68**	2.79	**1.90**	5.26	2.76
TS-SLAM (Delaunay)	1.14†	4.46	4.88	4.16	6.45	4.81	3.56
TS-SLAM (TSDF)	0.95†	4.43	3.24	4.17	5.98	5.00	2.63
NeXeL-SLAM	**0.59**	**3.45**	2.95	**1.95**	7.26	**2.79**	**2.57**

value. Using MonoGS's photometric-dominant loss ratio ($\lambda_{\text{geo}}/\lambda_{\text{pho}} = 0.05$) increases NeXeL-SLAM's error to $1.36\,\text{cm}$, demonstrating the benefit of depth-dominant weighting in this single-scene comparison.

Real sensor data. Within the evaluated $3\,\text{m}$ range, NeXeL-SLAM's depth errors are 3.45, 1.95, and $2.79\,\text{cm}$ on the three TUM sequences, respectively, lower than the MonoGS and TS-SLAM entries in Table 2 (Figure 2b). Fused reconstruction is more mixed. NeXeL-SLAM improves on TS-SLAM's TSDF reconstruction on `fr1/desk` (2.95 vs. $3.24\,\text{cm}$) and is similar on `fr3/office` (2.57 vs. $2.63\,\text{cm}$). MonoGS is more accurate on `fr1/desk`, while both baselines are better on `fr2/xyz`, where NeXeL-SLAM reaches $7.26\,\text{cm}$.

On `fr2/xyz`, fusing NeXeL-SLAM over its own 681 keyframes rather than TS-SLAM's 106 increases Chamfer to $11.33\,\text{cm}$. This is consistent with residual floating surfaces accumulating under fusion: densification fills missing geometry but does not explicitly remove elements that violate observed free space. In the full-range evaluation, pixels beyond $3\,\text{m}$ occupy less than 10% of the image yet contribute 88% and 73% of the depth error on `fr2/xyz` and `fr3/office`, respectively.

4.4 Camera Tracking

On Replica, NeXeL-SLAM achieves $0.28\,\text{cm}$ all-frame ATE-RMSE, compared with $0.306\,\text{cm}$ for MonoGS and $0.386\,\text{cm}$ for SplaTAM. MonoGS has lower error on six scenes; NeXeL-SLAM improves on `office0` and `office4`, with $0.23\,\text{cm}$ versus $0.800\,\text{cm}$ on the latter. NeXeL-SLAM's keyframe-only mean is also $0.28\,\text{cm}$.

On TUM, NeXeL-SLAM's mean ATE-RMSE is $2.02\,\text{cm}$, above MonoGS ($1.628\,\text{cm}$) and the DROID-SLAM reference ($1.528\,\text{cm}$), but below SplaTAM, ESLAM, and Co-SLAM. Its lowest error is $1.00\,\text{cm}$ on `fr2/xyz`, the best among the evaluated methods; its highest is $3.21\,\text{cm}$ on `fr3/office`. The accurate trajectory on `fr2/xyz` does not translate into an equally accurate fused surface, reinforcing the distinction between tracking and reconstruction quality.

4.5 Design Analysis

Loss weighting. We examine the effect of geometric and photometric loss balance within the nexel representation. On Replica `room0`, the final ratio $\lambda_{\text{geo}}/\lambda_{\text{pho}} = 2$ yields a Chamfer distance of $0.68\,\text{cm}$, compared with $1.36\,\text{cm}$ under the photometric-dominant ratio of 0.05 (Table 3). This single-sequence comparison highlights the importance of the optimisation objective alongside the choice of surface representation.

Repeatability. An auxiliary repeatability check on Replica `room1` gives an ATE-RMSE of $0.290 \pm 0.009\,\text{cm}$ and a PSNR of $36.77 \pm 0.06\,\text{dB}$ over three executions of the same configuration. We report the mean and sample standard deviation. These sequence-level statistics are reported separately from the single-run NeXeL-SLAM dataset aggregates.

Table 3: Loss-weight sensitivity on Replica `room0`, using the reconstruction evaluation in Sec. 4.3. Both settings use NeXeL-SLAM; the comparison concerns loss balance rather than different geometric primitives. Chamfer distance is in centimetres.

Setting	$\lambda_{\mathrm{geo}}/\lambda_{\mathrm{pho}}$	Chamfer ↓
Photometric-dominant	0.05	1.36
Final configuration	2	0.68

5 Conclusion

We presented NeXeL-SLAM, an RGB-D SLAM system that combines depth-initialised surface elements with a shared neural texture field, allowing appearance to vary within existing geometric primitives. A common differentiable rasteriser supports tracking from depth and base colours and map refinement with the full neural appearance. On Replica, the system jointly achieves $37.96\,\mathrm{dB}$ training-view PSNR, $0.59\,\mathrm{cm}$ mean Chamfer distance for the fused reconstruction, and $0.28\,\mathrm{cm}$ all-frame ATE-RMSE. These results show that detailed appearance, accurate reconstruction, and sub-centimetre tracking can coexist in an online surface map. On TUM RGB-D, strong rendering and range-limited depth accuracy do not consistently translate into equally accurate fused surfaces. Enforcing free-space consistency during map updates is a promising direction for reducing floating surfaces and improving reconstruction from real RGB-D observations.

AI use statement

We used generative AI tools to assist with manuscript drafting, restructuring, and language editing; identifying and summarising related literature; and preparing LaTeX and bibliographic entries. We also used these tools for research ideation and execution, including refining the research framing, suggesting experimental analyses and consistency checks, and interpreting author-provided results. This included checking consistency between the manuscript, tables, and evaluation descriptions. All research decisions, experiments, technical claims, references, and reported results were reviewed and verified by the authors. The authors take responsibility for the accuracy and integrity of the final manuscript and its contents.

Ethics statement

The reported experiments use the established Replica and TUM RGB-D benchmarks to evaluate camera tracking, rendering, and surface reconstruction. Our work concerns scene modelling rather than identifying or profiling individuals. Potential risks of deployment include revealing private indoor layouts, personal belongings, or information visible in reconstructed textures. Capturing and sharing such maps should therefore involve appropriate permission, data minimisation, and access controls. The benchmark results do not establish suitability for safety-critical navigation: tracking errors and incomplete or inconsistent geometry, including the floating surfaces observed on real RGB-D sequences, require further validation and safeguards before deployment.

Reproducibility statement

Section 3 describes the scene representation, rendering equations, and tracking and mapping objectives. Section 4.1 specifies the datasets, baseline variants, hardware, principal optimisation settings, evaluation metrics, and run counts. Section 4.3 describes the surface-reconstruction evaluation. Appendix A provides per-sequence results, frame-selection differences, available repeat statistics, and additional rendering settings. We distinguish rerun baselines from published reference values and separate the primary rendering evaluation from offline colour refinement and reference-depth-guided sampling. The code will be released publicly upon publication.

References

Nicholas Fry, Eric Dexheimer, Kirill Mazur, Paul H. J. Kelly, and Andrew J. Davison. Triangle splatting slam. In *European Conference on Computer Vision (ECCV)*, pp. 196–214, 2026.

Yiming Gao, Yan-Pei Cao, and Ying Shan. Surfelnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 108–118, 2023.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, 2024a. doi: 10.1145/3641519.3657428.

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21584–21593, 2024b.

Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17408–17419, 2023.

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, 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), 2023. doi: 10.1145/3592433.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728, 2024.

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, 2024.

Marko Mihajlovic, Silvan Weder, Marc Pollefeys, and Martin R. Oswald. Deepsurfels: Learning online appearance fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14524–14535, 2021.

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 *2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 127–136, 2011. doi: 10.1109/ISMAR.2011.6092378.

Victor Rong, Jingxiang Chen, Sherwin Bahmani, Kiriakos Kutulakos, and David Lindell. Gstex: Per-primitive texturing of 2d gaussian splatting for decoupled appearance and geometry modeling. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 3508–3518, 2025a.

Victor Rong, Jan Held, Victor Chu, Daniel Rebain, Marc Van Droogenbroeck, Kiriakos N. Kutulakos, Andrea Tagliasacchi, and David B. Lindell. Nexels: Neurally-textured surfels for real-time novel view synthesis with sparse geometries. *arXiv preprint arXiv:2512.13796*, 2025b. doi: 10.48550/arXiv.2512.13796.

Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18433–18444, 2023.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. doi: 10.48550/arXiv.1906.05797.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

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.

David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Billboard splatting (bbsplat): Learnable textured primitives for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25029–25039, 2025.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13293–13302, 2023.

Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems*, 2015. doi: 10.15607/RSS.2015.XI.001.

Tian-Xing Xu, Wenbo Hu, Yu-Kun Lai, Ying Shan, and Song-Hai Zhang. Texture-gs: Disentangling the geometry and texture for 3d gaussian splatting editing. In *Computer Vision – ECCV 2024*, pp. 37–53, 2024. doi: 10.1007/978-3-031-72698-9_3.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604, 2024.

Xin Zhang, Anpei Chen, Jincheng Xiong, Pinxuan Dai, Yujun Shen, and Weiwei Xu. Neural shell texture splatting: More details and fewer primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25229–25238, 2025.

Xingguang Zhong, Yue Pan, Liren Jin, Marija Popović, Jens Behley, and Cyrill Stachniss. Globally consistent rgb-d slam with 2d gaussian splatting. *IEEE Transactions on Robotics*, 2026. doi: 10.1109/TRO.2026.3697159.

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12786–12796, 2022.

A Appendix

This appendix provides per-sequence tracking, rendering, and reconstruction results, together with additional rendering settings and qualitative comparisons. We distinguish shared-evaluator reruns from published reference values throughout.

A.1 Per-sequence Tracking

Tables 4 and 5 report camera-centre ATE-RMSE after rigid alignment without scale fitting. All-frame results are compared in the upper blocks, with NeXeL-SLAM's keyframe-only scores listed separately. SplaTAM trajectories are rescored using the shared evaluator rather than its inherited mean-error metric on world-to-camera translation vectors.

TUM frame selection. The rerun baseline loaders thin associated frames to at most 32 frames per second, whereas DROID-SLAM processes all associated frames. On `fr2/xyz`, the resulting streams contain 3,397 and 3,669 frames, respectively. DROID-SLAM therefore serves as an additional tracking reference rather than an identical-input comparison.

Table 4: Replica tracking: ATE-RMSE [cm] after rigid alignment without scale fitting. R and O denote room and office. The upper block uses all tracked frames; NeXeL-SLAM's keyframe-only results are shown separately. Bold marks the lowest numerical error within the all-frame block.

Method	R0	R1	R2	O0	O1	O2	O3	O4	Mean
SplaTAM	0.372	0.245	0.267	0.461	0.290	0.403	0.439	0.612	0.386
MonoGS	**0.250**	**0.206**	**0.201**	0.379	**0.220**	**0.255**	**0.140**	0.800	0.306
ESLAM	0.693	0.626	0.537	0.461	0.471	0.546	0.681	0.607	0.578
Co-SLAM	0.584	1.982	0.993	0.559	0.538	1.997	1.335	0.839	1.103
DROID-SLAM	0.456	0.353	0.407	0.308	0.281	0.485	0.643	0.472	0.426
NeXeL-SLAM (all frames)	0.273	0.300	0.221	**0.301**	0.348	0.271	0.267	**0.231**	**0.277**
NeXeL-SLAM (keyframes)	0.275	0.301	0.231	0.301	0.368	0.270	0.266	0.220	0.279

Table 5: TUM RGB-D tracking: ATE-RMSE [cm]. The upper block uses all tracked frames; NeXeL-SLAM's keyframe-only results are shown separately. DROID-SLAM processes all associated frames rather than the thinned stream used by the other rerun baselines. Bold marks the lowest numerical error within the all-frame block.

Method	fr1/desk	fr2/xyz	fr3/office	Mean
SplaTAM	1.911	1.539	2.825	2.092
MonoGS	**1.407**	1.757	1.720	1.628
ESLAM	2.476	1.086	2.778	2.114
Co-SLAM	2.880	1.820	2.945	2.548
DROID-SLAM	1.600	1.356	**1.628**	**1.528**
NeXeL-SLAM (all frames)	1.829	**1.005**	3.211	2.015
NeXeL-SLAM (keyframes)	1.935	1.005	3.196	2.046

Run-to-run variation. MonoGS was run three times on each Replica scene. Its mean ATE-RMSE is $0.306 \pm 0.002\,\mathrm{cm}$, with the standard deviation computed over three run-level eight-scene means, not across individual scenes. The other rerun baseline entries use one run per sequence.

A.2 Per-sequence Appearance and Depth

Tables 6 and 7 report NeXeL-SLAM's training-view PSNR and full-range depth ℓ_1. Colour uses the full neural texture field and depth uses the coverage-normalised rendering in Eq. (5). Both are evaluated at estimated poses on every fifth input frame, under the valid-reference-depth mask. These are observed-view measurements rather than held-out-view results.

Table 6: Per-sequence NeXeL-SLAM appearance and full-range depth on Replica, evaluated at estimated poses on every fifth input frame. PSNR is in decibels and depth ℓ_1 is in centimetres. R and O denote room and office.

Metric	R0	R1	R2	O0	O1	O2	O3	O4	Mean
PSNR ↑	34.78	36.83	38.38	41.22	41.85	36.00	36.32	38.33	37.96
Depth ℓ_1 ↓	0.19	0.16	0.20	0.16	0.13	0.17	0.22	0.20	0.18

Qualitative comparison. Figure 3 compares renderings on Replica `room0`. Each method is rendered at its own estimated pose, with sensor depth used as the reference for the depth-error visualisation.

Table 7: Per-sequence NeXeL-SLAM appearance and full-range depth on TUM RGB-D, evaluated at estimated poses on every fifth input frame. PSNR is in decibels and depth ℓ_1 is in centimetres. These depth scores are distinct from the 3 m-limited comparison in Table 2.

Metric	fr1/desk	fr2/xyz	fr3/office	Mean
PSNR ↑	23.10	26.24	23.58	24.31
Depth ℓ_1 ↓	3.67	16.07	10.04	9.92

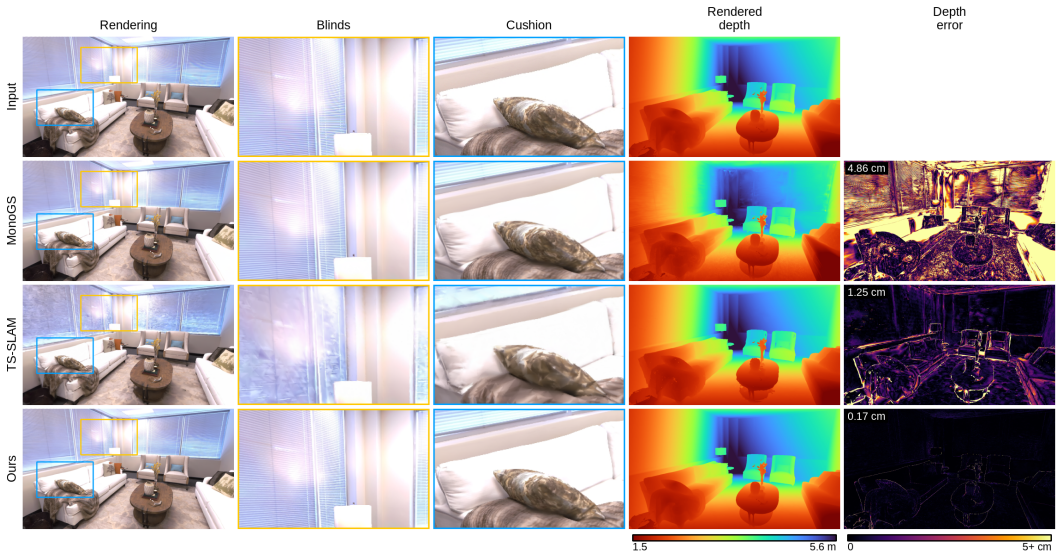


Figure 3: **Qualitative comparison on Replica `room0`.** Each method is rendered at its own estimated pose. Depth-error maps compare the rendered depth with the sensor observations.

A.3 Per-sequence Geometric Accuracy

Table 8 expands the Replica reconstruction results in Table 2, following the Chamfer protocol in Sec. 4.3. NeXeL-SLAM's TSDF reconstruction is compared with a reference fused from sensor depth at ground-truth poses. Published baseline values are marked separately from our measurements.

In addition to Chamfer distance, we report two surface-coverage measures. *Accuracy* is the percentage of the reconstructed surface within $1\,\mathrm{cm}$ of the reference, while *completion* is the percentage of the reference surface within $1\,\mathrm{cm}$ of the reconstruction. Their means are 91.6% and 91.8%, respectively.

We also evaluate rendered depth at ground-truth rather than estimated poses. This yields a mean error of $0.42\,\mathrm{cm}$, compared with $0.18\,\mathrm{cm}$ at estimated poses. The ground-truth-pose evaluation avoids using the estimated trajectory for rendering, while still evaluating the map constructed by the SLAM system.

Table 8: Per-sequence geometric accuracy on Replica. Chamfer distance and depth ℓ_1 are in centimetres; accuracy and completion are percentages within $1\,\mathrm{cm}$ of the opposite surface. Estimated-pose and ground-truth-pose depth evaluations are reported separately. †Published in Fry et al. (2026); these values are distinguished from the reruns discussed in Sec. 4.3. Bold marks the lowest reported Chamfer distance in each column. MonoGS's online map scores 1.09 on `room0` under our implementation (3.76 reported).

Method / metric	R0	R1	R2	O0	O1	O2	O3	O4	Mean
Chamfer distance ↓									
MonoGS†	3.76	4.39	4.63	2.93	4.78	4.18	4.36	3.26	4.03
MonoGS-2D†	1.93	1.16	1.54	1.56	0.70	1.49	1.37	1.15	1.36
TS-SLAM (Delaunay)†	1.16	1.01	1.09	0.77	0.69	1.37	2.00	1.02	1.14
TS-SLAM (TSDF)†	1.00	0.77	1.01	0.66	0.56	1.12	1.69	0.78	0.95
NeXeL-SLAM	**0.68**	**0.60**	**0.51**	**0.51**	**0.47**	**0.58**	**0.72**	**0.63**	**0.59**
NeXeL-SLAM, further metrics									
Depth, est. pose ↓	0.19	0.16	0.20	0.16	0.13	0.17	0.22	0.20	0.18
Depth, GT pose ↓	0.62	0.47	0.29	0.34	0.38	0.32	0.62	0.31	0.42
Acc. $< 1\,\mathrm{cm}$ (%) ↑	86.2	91.7	95.8	97.0	99.1	92.2	83.1	87.9	91.6
Comp. $< 1\,\mathrm{cm}$ (%) ↑	86.2	92.0	95.8	97.3	99.1	92.8	83.5	88.1	91.8

A.4 Additional Rendering Settings

Table 9 separates the primary rendering setup from two alternatives. MonoGS applies an additional offline colour-refinement pass. ESLAM and Co-SLAM instead use reference depth to guide evaluation-time ray sampling. This supplies information to the renderer beyond the primary evaluation setting; it does not change the fact that all evaluated systems use RGB-D observations during SLAM. These alternative scores are not mixed into Table 1.

A.5 Published Surface-based Rendering Reference

Table 10 retains TS-SLAM's published Replica PSNR and depth error as context for the explicit-surface representation. These particular rendering values are quoted rather than rerun, unlike the TUM reconstruction comparisons in Sec. 4.3.

The depth protocols differ: NeXeL-SLAM evaluates coverage-normalised depth on every fifth input frame, whereas the published TS-SLAM evaluation uses keyframes and excludes pixels without rendered depth. This table therefore provides a published rendering reference, not a matched-protocol comparison of reconstructed surfaces. The latter is addressed separately in Sec. 4.3.

Table 9: Mean PSNR [dB] under the primary and alternative rendering settings. Offline colour refinement adds optimisation, while reference-depth-guided sampling supplies additional evaluation-time information. Results from these settings are not pooled into a single ranking.

Method	Setting	Replica	TUM RGB-D
MonoGS	Online map	35.35	18.99
	After offline colour refinement	39.11	26.48
ESLAM	Without reference-depth guidance	23.65	17.90
	With reference-depth guidance	30.56	21.67
Co-SLAM	Without reference-depth guidance	28.09	20.16
	With reference-depth guidance	30.44	20.22

Table 10: Published Replica rendering reference. TS-SLAM's PSNR and depth values are quoted from Fry et al. (2026), not obtained from a shared rendering evaluation. Frame selection and depth masking differ from the NeXeL-SLAM evaluation, so no best-value highlighting is applied.

Method / source	PSNR [dB] ↑	Depth ℓ_1 [cm] ↓
TS-SLAM (as published)	31.33	0.68
NeXeL-SLAM (this evaluation)	37.96	0.18