GenRecon: Bridging Generative Priors for Multi-View 3D Scene Reconstruction

Katharina Schmid[1] **Nicolas von Lützow**[1] **Jozef Hladký**[2] **Angela Dai**[1] **Matthias Nießner**[1]

[1]Technical University of Munich [2]Computing Systems Lab, Huawei Technologies, Switzerland

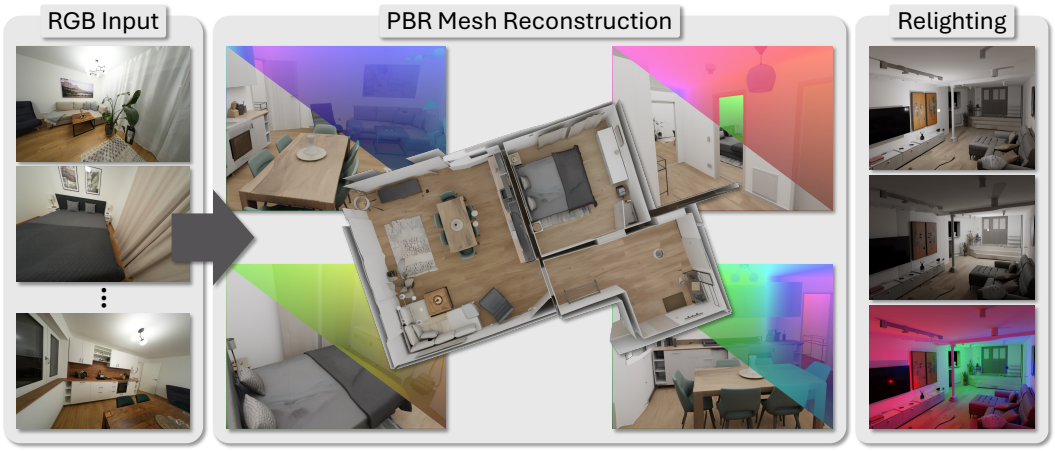

Figure 1: **GenRecon.** Given a sparse set of RGB images of an indoor scene (left), our method reconstructs a complete, high-fidelity PBR mesh (center) by formulating scene reconstruction as conditional 3D generation over overlapping spatial chunks. The recovered mesh with material properties enables realistic relighting and editing in standard rendering pipelines (right).

Abstract

We introduce a new approach to high-fidelity 3D scene reconstruction from multi-view RGB images that tightly couples reconstruction with a strong generative 3D prior. We cast scene reconstruction as conditional 3D generation over a set of spatially-localized, overlapping chunks that together tile the scene, scaling generation to large scene extents. Crucially, we inherit the fidelity and completeness of state-of-the-art generative shape models – we use Trellis.2 as an example – which we generalize to the scene level. To this end, we propose a projection-based conditioning mechanism that lifts posed multi-view image features into a coherent 3D representation aligned with the generative model, independent of view ordering and spatially anchored to the scene, yielding high-fidelity, multi-view consistent generated geometry. This enables lifting the strong object-level prior of Trellis.2 to multi-view, scene-scale generation, producing faithful, editable PBR mesh reconstructions of indoor environments. As a result, we obtain high-fidelity results that outperform cutting-edge reconstruction methods by 16%.

1 Introduction

Reconstructing high-quality 3D scenes from multi-view RGB images is a fundamental problem in computer vision and graphics, underpinning applications ranging from AR/VR and robotics to embodied AI, simulation, and digital content creation. For instance, a robot navigating a cluttered

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

environment, an artist importing a captured environment into a game engine, and an immersive VR experience transporting a user to a distant real-world setting can all be powered by 3D scene reconstructions. The requirements imposed on a reconstruction, however, vary across these settings. For navigation and perception, reconstruction primarily provides the geometric structure needed for downstream tasks, where metric accuracy is prioritized and high surface and visual fidelity is not essential. For content creation and immersive applications, 3D reconstructed scenes must meet a substantially higher fidelity bar, matching the quality of crafted 3D assets with complete, high-fidelity surfaces along with material properties suitable for relighting and editing.

Achieving such high-fidelity 3D scene reconstruction from multi-view images is fundamentally challenging, as it is a highly underconstrained inverse task. From only a set of 2D views, recovering the actual 3D structure at any given location requires many observations from diverse viewpoints. This requires reliable correspondences to be established, a difficult problem due to needing both sophisticated appearance and semantic understanding to handle textureless regions, repetitive patterns, large viewpoint changes, and view-dependent effects. Real scene captures rarely satisfy diverse, accurate correspondences everywhere in a scene, so per-scene optimization-based approaches [1–8] often produce incomplete, noisy, or oversmoothed reconstructions in these underconstrained regions.

Despite these challenges, recent works have made significant progress in reconstruction by incorporating learned priors. Feed-forward scene reconstruction methods [9–17] have transformed the field, recovering geometry directly from images in a single pass and producing remarkably consistent reconstructions that have the potential to power downstream navigation and perception tasks. Unfortunately, their outputs remain ill-suited to the required fidelity needed for content creation scenarios: surfaces remain noisy or oversmoothed in challenging regions, and incomplete in occluded and unobserved areas. At the same time, generative 3D modeling has made rapid strides in producing realistic, coherent, and complete 3D object shapes. Modern generative shape models [18–23] capture powerful priors over high-quality shape geometry, enabling the synthesis of detailed, structurally consistent 3D assets. We observe that these strong generative 3D shape priors offer a powerful opportunity in scene reconstruction: by integrating strong generative shape priors directly into multi-view reconstruction, this enables reconstruction of complete, high-fidelity 3D scene assets.

In this work, we introduce a new approach that tightly couples multi-view 3D reconstruction with a strong generative 3D prior. We formulate scene reconstruction as conditional 3D generation over a set of spatially-localized, overlapping scene chunks, enabling large-scale reconstruction while inheriting the fidelity, completeness, and realism of state-of-the-art generative shape model Trellis.2 [24]. We recast the shape generative prior from Trellis.2 to support multi-view scene chunk generation by formulating a projection-based conditioning pathway that injects posed multi-view image information into the generative model in a spatially-grounded, permutation-invariant manner. This allows precise control over both generated geometry and spatial alignment. By preserving the pretrained prior through parameter-efficient fine-tuning on synthetic scene data, our method produces faithful, editable PBR mesh reconstructions of indoor scenes, significantly narrowing the gap between current reconstruction capabilities and the quality required for content creation scenarios. To summarize, our contributions are:

- We introduce a new approach for reconstructing scene-level PBR meshes from RGB images, by coupling multi-view reconstruction with a strong object-level 3D generative prior. We formulate reconstruction as conditional 3D generation over overlapping spatial scene chunks, casting scene recovery as a single coherent generation process in which all chunks are jointly synthesized under the guidance of the input views.
- To enable this, we extend a single-image, object-level 3D generative prior to a multi-image, pose-controlled setting via a dedicated 3D conditioning pathway. This pathway lifts multi-view image features using explicit camera poses and fuses them in a spatially-grounded, permutation-invariant manner, enforcing strict geometric consistency across views while enabling precise control over the resulting 3D structure.

2 Related Work

Reconstruction without learned priors. Classic multi-view stereo pipelines such as COLMAP [25] reconstruct geometry through feature matching, epipolar verification, and patch-based stereo fusion, but relying solely on photoconsistency, they cannot recover structure in weakly

textured, occluded, or sparsely observed regions. Neural implicit surface methods [1–3, 26] extend this paradigm by representing scenes as continuous signed-distance or density fields optimized via differentiable volume rendering; while they recover smoother surfaces, they still fail in ambiguous regions where triangulation is under-constrained. MonoSDF [4] and NeuSG [5] attempt to mitigate these limitations by augmenting per-scene optimization with monocular depth cues or jointly optimized 3D Gaussian Splatting guidance, yet both remain unable to generate geometry beyond what photoconsistency can constrain. Similarly, Gaussian splatting methods [6–8] optimize explicit anisotropic primitives via differentiable rasterization for fast rendering, but without learned shape priors, they suffer from the same noise and incompleteness in unobserved regions. MeshSplats [27] translates these primitives into disjoint mesh faces for standard ray-tracing pipelines, improving editability but inheriting the underlying reconstruction artifacts.

Reconstruction with learned priors. The shift toward learned priors has produced geometric foundation models [9–12, 28, 29] that regress dense depth maps, pointmaps, or camera parameters directly from images. While these achieve impressive geometric recovery on observed surfaces, their unstructured outputs must be fused into surfaces post hoc, and they lack generative priors to complete occluded regions. Building on this direction, feed-forward volumetric fusion methods [13, 14, 30–34] backproject image features into 3D volumes to directly regress TSDF or occupancy fields, while feed-forward Gaussian splatting methods [15–17, 35–39] localize explicit primitives in a single forward pass. Despite learning strong geometric priors, all of these approaches remain deterministic regressors that cannot generate coherent geometry in unobserved regions, and they produce unstructured depth maps or Gaussian clouds rather than editable meshes. To move beyond deterministic regression, recent methods have explored generative priors. Approaches based on 2D and video diffusion [40–46] synthesize intermediate frames or depth maps that are then reconstructed into 3D via downstream fusion or per-scene optimization, rather than directly generating structured geometry. Native 3D generative models [18–21] come closer to direct 3D output, but largely operate at the object level and are typically restricted to single-view conditioning. MV-SAM3D [47] and ReconViaGen [23] move toward multi-view conditioning, but still operate at object level. Concurrent to our work, Pixal3D [48] adopts a closely related conditioning strategy, back-projecting multi-scale image features into a 3D feature volume to establish explicit pixel-to-3D correspondence and thereby natively supporting pose-controlled single- and multi-view inputs. However, it remains restricted to object-level generation and does not produce PBR-textured geometry. DiffusionGS [22] extends generative completion to scenes yet conditions on a single image and outputs unstructured Gaussian splats. NOVA3R [49] similarly recovers occluded scene geometry from multiple unposed views via a diffusion-based decoder, but produces unstructured point clouds rather than meshes. Compositional methods [47, 48, 50–56] decompose inputs into individual objects, reconstruct each with an off-the-shelf generative model, and assemble them via post-hoc layout optimization. While this paradigm leverages strong object priors, it decouples generation from composition: object boundaries may be inconsistent, occluded geometry is hallucinated independently, and inter-object relations are enforced by optimization rather than emerging from a single coherent generative process.

In contrast to these approaches, our method leverages a pretrained 3D generative prior to directly synthesize complete, structured mesh geometry conditioned on the input views. By formulating scene reconstruction as a single coherent conditional 3D generation process over overlapping spatial chunks, we bypass per-object decomposition, per-view fusion, and per-scene optimization entirely.

3 Method

We address the problem of reconstructing a complete, high-fidelity 3D scene from a sparse, unordered set of N posed RGB images $\{I_n\}_{n=1}^{N}$ with associated camera intrinsics and extrinsics $\{K_n, T_n\}_{n=1}^{N}$. Our output is a scene-level mesh $\mathcal{M}$ with PBR materials, suitable for direct integration into rendering and authoring pipelines (Figure 2). Our approach tightly couples multi-view reconstruction with a strong generative 3D prior by casting scene reconstruction as the joint generation of a set of overlapping scene chunks that cover the entire scene (Section 3.2). To realize this, we employ the object-level generative prior from Trellis.2 [24], and recast it for scene-level generation by introducing multi-view conditioning that spatially grounds scene chunk generation (Section 3.1).

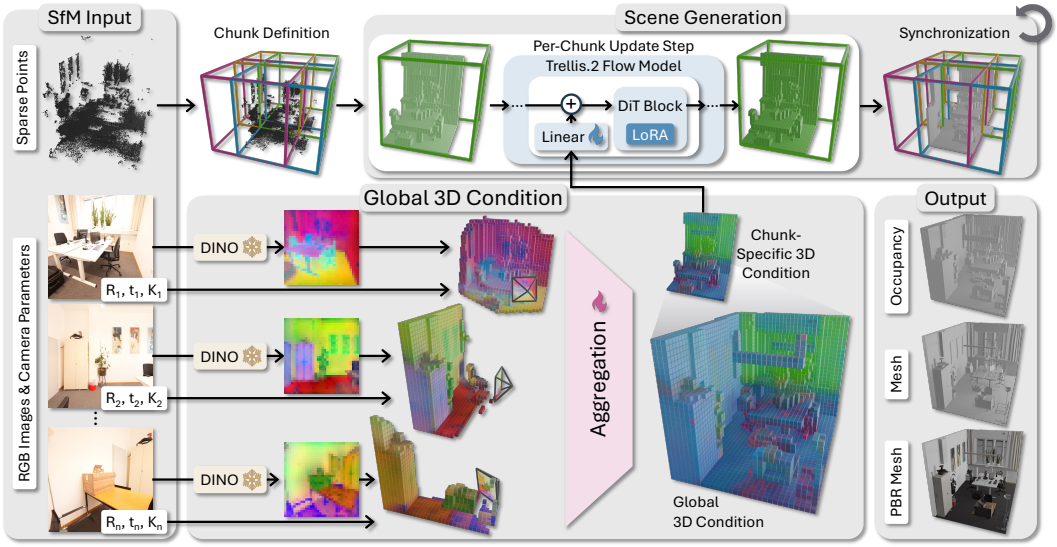

Figure 2: **Pipeline overview.** Given posed RGB images and a sparse point cloud from SfM (left), we define overlapping scene chunks and construct a global 3D conditioning grid by lifting DINOv3 image features into per-view volumes and aggregating them (center). By extending a 3D generative prior with a new spatially-grounded multi-view conditioning pathway, we jointly generate all chunks in a single flow-matching trajectory to recover a complete, pose-aligned scene-level PBR mesh (right).

3.1 Multi-view scene chunk generation

Scene chunks. We define a scene chunk c as a fixed-size 3D volume $V_c = [0, L]^3$ in its own canonical coordinate frame, paired with a translation $t_c \in \mathbb{R}^3$ that places it in the world frame. Each chunk c is associated with a set of input views $\mathcal{V}_c \subseteq \{1, \dots, N\}$ whose cameras observe c. Our generative model Φ takes as input the chunk's canonical volume specification and the posed views $\{(I_n, K_n, T_n^{-1} T_c)\}_{n \in \mathcal{V}_c}$, where T_c denotes the chunk-to-world transform corresponding to t_c, and produces a 3D latent $z^{(c)}$ representing the geometry and appearance within V_c.

Generative prior. We instantiate our generative model Φ from Trellis.2 [24], a state-of-the-art 3D shape generative model that produces high-quality objects by first predicting coarse occupancy, followed by high-fidelity shape and PBR texture. These are parameterized by a flow-matching [57] denoiser operating on the respective latent features. Trellis.2 is designed to take a single unposed RGB image as input through cross-attention; the position, orientation, and scale of generated content are not specified by the input but implicitly determined by the model's training distribution. While this design enables high-quality object generation, the single-image, pose-free conditioning regime is ill-suited for scene reconstruction: capturing large scenes inherently requires multiple views that the model must consume as a coherent set, as well as place generated content in a known coordinate frame so that adjacent chunks compose consistently.

Spatially-grounded multi-view conditioning. We address both gaps with a single design: a 3D conditioning pathway that carries multi-view image evidence into the generative model Φ in a spatially anchored, view-order-invariant form. Given a chunk c with associated views $\mathcal{V}_c$, we encode each image independently, lift the resulting per-view features into 3D grids over the chunk's volume, and aggregate across views in a permutation-invariant fashion to obtain the chunk's 3D conditioning $G^{(c)}$.

We encode each input image with DINOv3 [58], producing a dense 2D feature map F_n for each view to keep this input distribution close to Trellis.2's pretraining. For each view $n \in \mathcal{V}_c$, we then lift F_n into a per-view 3D feature grid $G_n^{(c)}$ defined over the chunk's canonical volume. Each voxel $x \in V_c$ is projected into the view's image plane via $\pi_n(x) = K_n T_n^{-1}(x + t_c)$, and the corresponding feature is retrieved: $G_n^{(c)}(x) = F_n\big(\pi_n(x)\big)$. This projection step spatially grounds the design, tying every conditioning feature to an explicit 3D location in the chunk's coordinate frame.

Finally, the per-view grids $\{G_n^{(c)}\}_{n\in\mathcal{V}_c}$ are aggregated into a single 3D conditioning grid $G^{(c)}$ using an IBRNet-style [59] scheme. The aggregation is permutation-invariant across views and for arbitrary $|\mathcal{V}_c|$, enabling our approach to handle variable numbers of input images without needing a canonical view ordering: For each voxel with per-view features $\{f_i\}_{i=1}^N$, we first compute cross-view statistics $\mu = \frac{1}{N}\sum_i f_i$ and $\sigma^2 = \frac{1}{N}\sum_i f_i^2 - \mu^2$, which serve as global context shared across views. Each view's feature is refined and assigned an aggregation logit by two small MLPs sharing the same input:

$$f_i' = \mathrm{MLP}_{\text{feat}}([f_i, \mu, \sigma^2]), \qquad w_i = \mathrm{MLP}_{\text{weight}}([f_i, \mu, \sigma^2]). \tag{1}$$

The final voxel feature is the mean plus a softmax-weighted residual, where $\mathrm{MLP}_{\text{feat}}$'s final layer is zero-initialized so the module starts training as a cross-view mean:

$$f_{\text{out}} = \mu + \sum_i \alpha_i\, f_i', \qquad \text{where } \alpha_i = \mathrm{softmax}_i(w_i). \tag{2}$$

Conditioning injection. The aggregated 3D condition $G^{(c)}$ is injected into the generative denoiser Φ residually at each block, added voxel-wise through a zero-initialized layer so that initialization preserves the pretrained model's behavior. Because $G^{(c)}$ is defined directly on the chunk's coordinate frame, every conditioning signal carries explicit positional meaning, and view consistency and pose control fall out as direct consequences of the design rather than properties the model must learn.

Training. We train the conditioning pathway together with a low-rank LoRA adapter [60] on the weights of Φ, keeping the remaining Trellis.2 parameters frozen. Training is performed on synthetic scene data, supervising chunk generation against ground-truth chunk latents extracted from the synthetic scenes. Further details are specified in Appendix A.

3.2 Scene reconstruction at test time

At test time, given an unordered set of RGB images $\{I_n\}_{n=1}^N$ of an unseen scene, we produce a scene-level PBR mesh $\mathcal{M}$.

Scene calibration and chunking. Since the input images are unposed at inference time, we first run structure-from-motion (COLMAP [25]) to recover the camera intrinsics $\{K_n\}$, extrinsics $\{T_n\}$, and a sparse point cloud $P \subset \mathbb{R}^3$ of the scene. We apply statistical and radius-based outlier filtering to P and estimate the scene's spatial extent $\mathcal{B} \subset \mathbb{R}^3$ from the filtered points using robust percentile-based bounds. Given $\mathcal{B}$, we partition the scene volume into a set of chunks $\mathcal{C} = \{c_1, \ldots, c_K\}$, each occupying a fixed-size cube V_c in its own canonical frame with a translation $t_c \in \mathbb{R}^3$ placing it in the world frame. Neighboring chunks overlap by a prescribed minimum margin m, providing the regions across which chunks exchange information during joint generation.

Global 3D conditioning. Rather than computing the conditioning grids $G^{(c)}$ independently per chunk, we compute a global conditioning grid G once over the full scene volume and extract per-chunk conditions $G^{(c)}$ as crops. Concretely, we lift each encoded image F_n into a scene-sized voxel grid via the per-view projection of Section 3.1, and aggregate across views to obtain G. For occupancy generation, G is dense at the resolution of the occupancy latents; for shape and texture generation, which operate on higher-resolution sparse latents defined by the predicted occupancy, the lifting and aggregation are also performed on the corresponding sparse high-resolution voxel structure. The per-chunk conditions $G^{(c)}$ are then obtained by cropping G to each V_c.

Joint chunk generation. All chunks are generated jointly by Φ in a single flow-matching trajectory following a MultiDiffusion-style [61] scheme. We maintain a global noisy latent grid z_t covering the full scene volume. At each step t, for each chunk $c \in \mathcal{C}$ we extract its corresponding latent crop $z_t^{(c)}$ and apply the chunk-wise denoiser to obtain a per-chunk prediction $\hat{z}_{t-1}^{(c)}$. The per-chunk predictions are merged into the next global latent z_{t-1} by averaging in overlap regions:

$$z_{t-1}(x) = \frac{1}{\sum_{c\in\mathcal{C}} M_c(x)} \sum_{c\in\mathcal{C}} M_c(x)\, \hat{z}_{t-1}^{(c)}(x), \tag{3}$$

where x indexes a voxel in the global scene grid and $M_c(x) \in \{0, 1\}$ indicates whether x lies within V_c. This aggregation enforces consistency across chunk boundaries throughout the generation

trajectory. For shape and texture generation, we additionally apply a boundary-sensitive variant in which chunk-boundary voxels do not contribute to the aggregation but are still updated by it; we find this improves seam coherence visually. After the final step, the global latent grid z_0 is decoded by the respective Trellis.2 decoders into the final scene mesh $\mathcal{M}$ with PBR materials.

4 Experiments

Datasets. We train on chunks extracted from synthetic indoor scene data. Our primary training dataset is SAGE-10k [62], a set of synthetic indoor scenes with PBR materials and objects generated by Trellis [63]. While SAGE-10k provides a wide variety of single rooms, it does not contain multi-room layouts, windows, or door openings, all of which are important for our model to perform on real-world scenes. To expose our model to these structural elements, we additionally include a subset of scenes from 3D-FRONT [64] for occupancy generation training. See Appendix A for details.

Evaluation. We evaluate on unseen scenes from two datasets: 3D-FRONT [64] and ScanNet++ [65], to assess performance on synthetic and out-of-domain real-world data. Specifically, we evaluate on 50 scenes from ScanNet++ and 25 scenes from 3D-FRONT, using eight input views per scene. Additionally, we assume a set of sparse points and the camera poses to be given. For 3D-FRONT, we use single-room scenes with ground-truth poses, and sample 10k points from backprojected training-view depth maps to obtain points that respect visibility constraints. For ScanNet++, we use the provided COLMAP [25] outputs.

Metrics. We evaluate reconstructed meshes in both 2D and 3D. In 2D, we report geometric errors (MAE, RMSE, AbsRel, SqRel, angular normal error), perceptual/semantic metrics (LPIPS, CLIP), and completeness over valid pixels only; in 3D, we measure alignment and coverage using Chamfer distance, F-score (10 cm), and normal consistency (thresholded at 20 cm), restricted to observed regions. Details are specified in Appendix B.

Baselines. We compare against five reconstruction methods that span dominant paradigms in the literature. 2D Gaussian Splatting (2DGS) [7] performs prior-free per-scene optimization. MonoSDF [4] uses monocular geometric priors to help guide per-scene optimization. Depth Anything 3 (DA3) [12] is a feed-forward monocular depth foundation model; we fuse its predicted depths into 3D meshes using TSDF fusion. FineRecon [14] performs 3D refinement of fused monocular predictions, for which we use DA3 as the underlying depth foundation model. Murre [42] uses diffusion-based depth priors with 3D conditioning. Please refer to Appendix F for additional details of the baselines.

4.1 Reconstruction Results

Table 1: **2D baseline comparisons on real-world data.** We evaluate 2D depth and normal metrics on 50 scenes from ScanNet++.

		2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Depth	MAE (m) ↓	0.2717	0.3451	0.1612	0.1352	**0.1336**	0.1364
	RMSE (m) ↓	0.5074	0.5080	0.3118	0.2901	0.3199	**0.2830**
	AbsRel ↓	0.2437	0.4772	0.2131	0.1775	0.1580	**0.1514**
	SqRel ↓	0.1970	0.6915	0.1634	0.1511	0.1433	**0.1007**
	LPIPS ↓	0.3885	0.1426	0.3413	0.2112	0.3335	**0.1114**
	CLIP score ↑	0.8881	0.9659	0.9216	0.9301	0.9074	**0.9765**
Normal	Angular err. (°) ↓	36.58	22.20	28.81	21.03	25.67	**18.39**
	LPIPS ↓	0.7296	0.4465	0.6837	0.5210	0.6460	**0.3685**
	CLIP score ↑	0.8603	0.9095	0.8967	0.8868	0.8806	**0.9411**
	Completeness ↑	0.6780	0.9496	0.8065	0.8367	0.6819	**0.9864**

Real-world scene reconstruction. In Table 1 and Table 2, we evaluate the performance on ScanNet++ [65]. Despite training only on synthetic data, our method achieves the strongest reconstruction

Table 2: **3D baseline comparisons on real-world data.** We evaluate 3D reconstruction metrics on 50 scenes from ScanNet++.

	2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Chamfer (m) ↓	0.2597	0.1633	0.1291	0.1106	0.1213	**0.0773**
F-score @10cm ↑	0.4685	0.6106	0.6366	0.7004	0.6858	**0.7684**
Normal consistency ↑	0.4794	0.6704	0.6705	0.7467	0.6804	**0.7587**

quality on this entirely unseen real-world dataset across both 2D and 3D metrics, with substantially better perceptual and semantic alignment with the ground-truth laser scans on depth and normals and the highest completeness of all methods evaluated.

Figure 3 qualitatively compares the performance of our method against the baselines on ScanNet++ using 8 input images. While the baselines produce noisy (2DGS, DA3), oversmooth (FineRecon, MonoSDF) surfaces for challenging areas and are incomplete in occluded and unobserved areas (2DGS, DA3, FineRecon, Murre), our approach yields complete and high-fidelity reconstructions.

Synthetic scene reconstruction. Tables 3 and 4 report 2D and 3D metrics on 3D-FRONT [64]. While our occupancy stage was fine-tuned on a small subset of 3D-FRONT scenes, evaluation is performed exclusively on held-out scenes. Our method again achieves the strongest overall performance across both 2D and 3D metrics, indicating that the recovered geometry not only minimizes error but also matches the structural characteristics of the ground truth, avoiding both oversmoothing and high-frequency artifacts.

Table 3: **2D baseline comparisons on synthetic data.** We evaluate 2D depth and normal metrics on 25 scenes from 3D-FRONT.

		2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Depth	MAE (m) ↓	0.4542	0.8134	0.0957	0.1334	**0.0933**	0.1131
	RMSE (m) ↓	0.8475	1.0947	0.3712	0.4024	0.3396	**0.3200**
	AbsRel ↓	0.2191	0.5546	0.0686	0.1121	0.0729	**0.0609**
	SqRel ↓	0.3629	1.2379	0.1259	0.1977	0.1205	**0.0587**
	LPIPS ↓	0.5693	0.2538	0.4622	0.4289	0.3892	**0.1270**
	CLIP score ↑	0.8581	0.9349	0.8819	0.8677	0.8907	**0.9565**
Normal	Angular err. (°) ↓	38.1341	24.6739	16.0993	15.4063	16.9646	**10.7490**
	LPIPS ↓	0.8278	0.4580	0.6256	0.5838	0.6081	**0.2183**
	CLIP score ↑	0.8155	0.8752	0.8842	0.8531	0.8721	**0.9571**
	Completeness ↑	0.3924	0.9002	0.6424	0.5627	0.7484	**0.9871**

Table 4: **3D baseline comparisons on synthetic data.** We evaluate 3D reconstruction metrics on 25 scenes from 3D-FRONT.

	2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Chamfer (m) ↓	0.5924	0.3387	0.2087	0.2227	0.1584	**0.0638**
F-score @10cm ↑	0.1766	0.3463	0.5501	0.5129	0.6014	**0.8655**
Normal consistency ↑	0.3481	0.4264	0.6464	0.6244	0.6624	**0.8199**

4.2 Ablations

We ablate the projection-based 3D conditioning pathway and the number of input views. The ablations are performed on 25 chunks drawn from 25 distinct SAGE-10k [62] scenes not seen during training.

Effectiveness of the 3D contitioning path. We compare three variants: (i) the vanilla pretrained Trellis.2 model; (ii) Trellis.2 with the occupancy stage and geometry stage fine-tuned on our scene data; (iii) our full method. As shown in Table 5 and Figure 4, the vanilla Trellis.2 performs poorly on

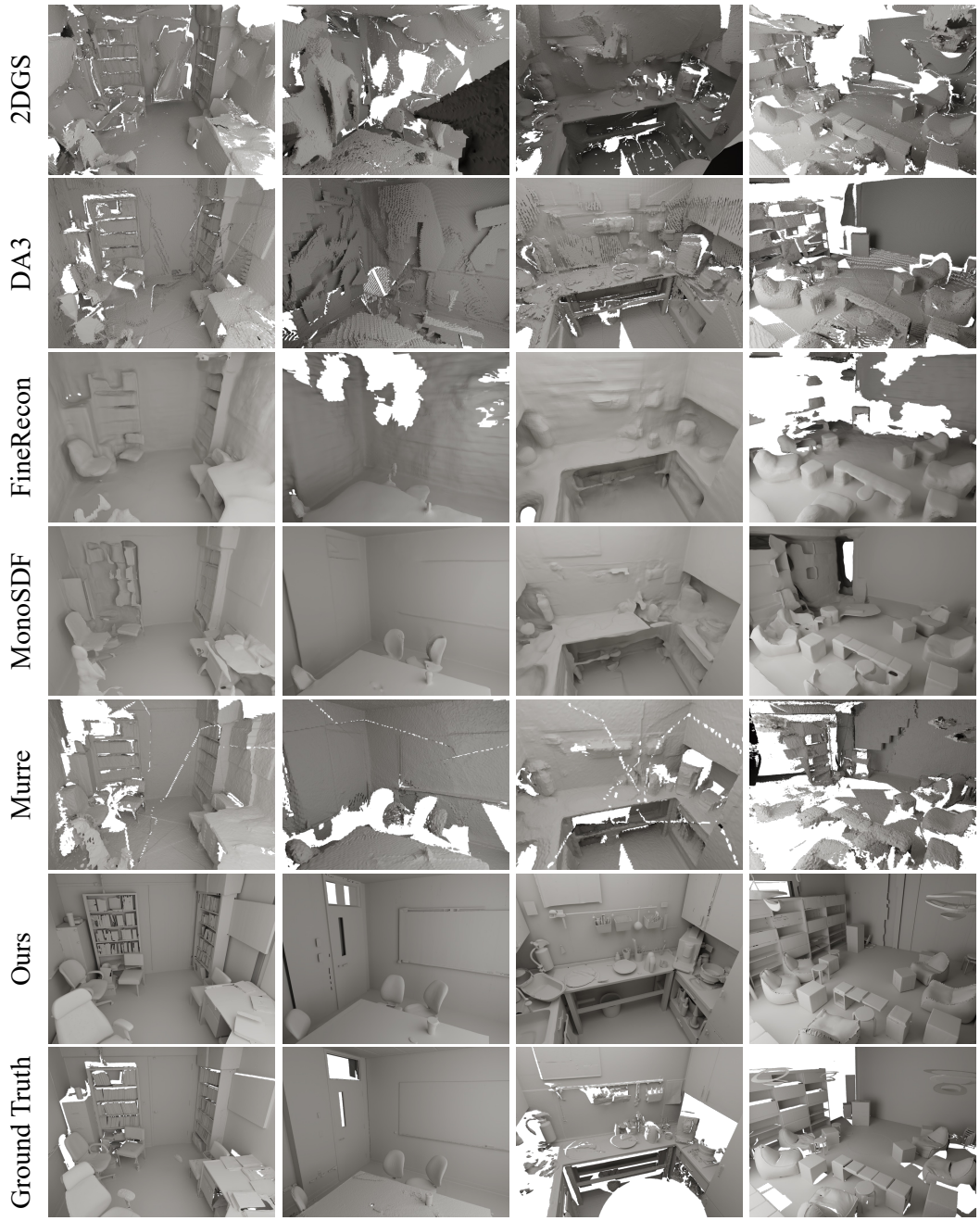

Figure 3: **Real-world comparisons.** Reconstruction performance on four evaluation scenes from ScanNet++. Compared to the baselines our results are more complete and reconstruct finer details.

scene chunks since it was trained for object-level generation with full visibility in the conditioning image. Fine-tuning Trellis.2 on scene data without the 3D condition produces plausible scene fragments but fails to place them in the correct pose. Adding our 3D conditioning pathway resolves this, aligning the generated geometry with the ground-truth pose. This confirms that the 3D condition is what enables pose-controlled chunk generation.

Effect of the number of input views. We further evaluate the sensitivity of our method to the number of conditioning views per chunk. We report results in Table 5 and Figure 4,. Notably, our 3D conditioning enables pose-correct chunk generation from already a single input image. In contrast to

cross-attention conditioning, our 3D conditioning approach achieves progressively better performance as the number of input views increases.

Table 5: **Ablation.** We evaluate the efficacy of our 3D conditioning on 25 scenes from SAGE-10k. For multi-view Trellis.2 and Trellis.2 fine-tuned, we use multidiffusion conditioning.

Method	Views	Chamfer (m) ↓	F-score @10cm ↑	Normal consistency ↑
Trellis.2	1	0.3450	0.2067	0.2341
	2	0.3127	0.2536	0.2791
	4	0.2885	0.2974	0.3166
Trellis.2 fine-tuned	1	0.1945	0.4893	0.5022
	2	0.1978	0.4741	0.4944
	4	0.2000	0.4797	0.4924
Ours	1	0.1345	0.6312	0.6349
	2	0.0812	0.7608	0.7688
	4	0.0420	0.9070	0.8411
	8	**0.0291**	**0.9683**	**0.8804**

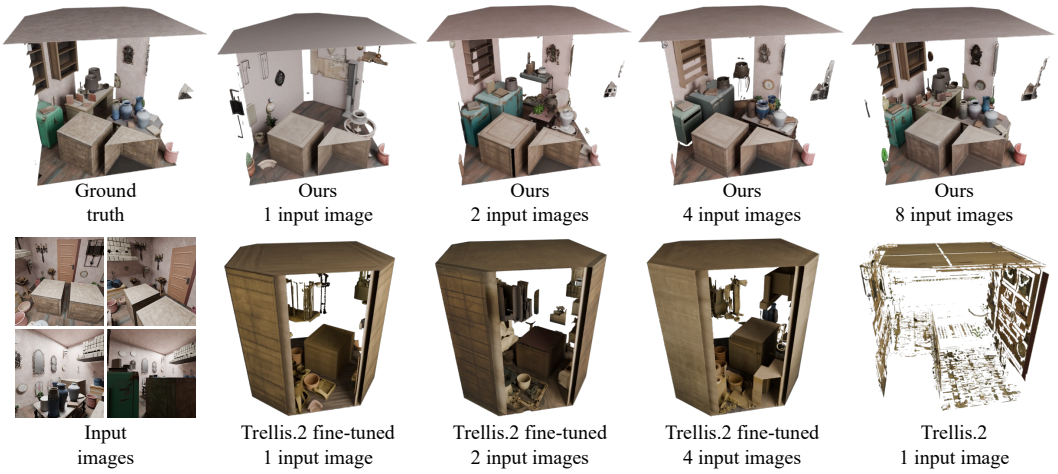

Figure 4: **Ablation results.** Ablation study on SAGE-10k chunk not seen during training. Our projection-based 3D conditioning effectively enables pose-correct chunk generation from as few as a single input image. Reconstruction quality improves with additional views.

4.3 PBR Texture and Relighting

Our method produces scene-level meshes with physically-based material properties (albedo, metallic, roughness), directly importable into standard rendering engines and relightable under arbitrary illumination without any per-scene optimization. Figure 5 shows the predicted material channels alongside the lit reconstruction, and Figure 6 shows relighting on real-world ScanNet++ scenes, with recovered albedo and reflectance responding plausibly in each case. Since our texture stage is fine-tuned exclusively on SAGE-10k, where PBR materials are themselves VLM-predicted rather than ground-truth measured, the recovered textures are visually plausible but do not match dedicated SVBRDF-estimation methods in absolute material accuracy.

4.4 Limitations

While our method achieves strong reconstruction quality across a wide range of indoor scenes, several limitations remain. Reconstructions of non-Lambertian surfaces, such as glass and mirrors, are less reliable, as such materials are underrepresented in SAGE-10k. Our chunk partitioning is currently designed for indoor scene settings with vertical extents up to roughly 5m. Extending beyond this

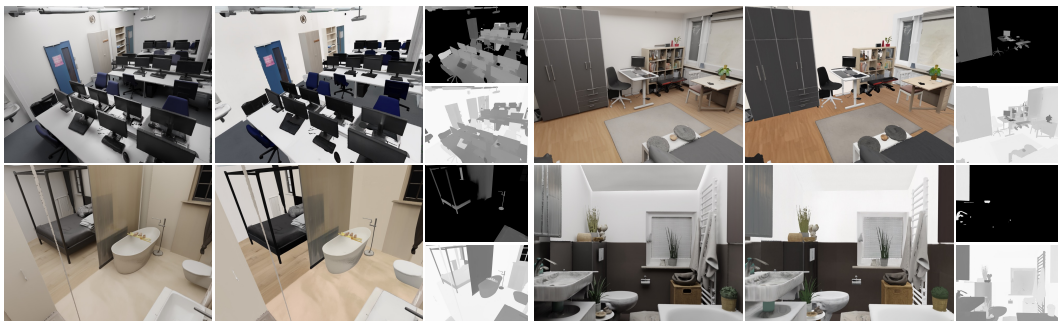

Figure 5: **PBR results.** Qualitative results on four reconstructed scenes from ScanNet++: lit scene (left), albedo (middle), metallic and roughness (right).



Figure 6: **Relighting results.** Varying lighting configurations for scenes reconstructed from ScanNet++. Further visualizations are provided in Appendix E.

would require adaptive chunking based on the full spatial extent. Finally, our incorporation of a strong generative prior may lead to reconstructions occasionally hallucinating content in regions where input views provide weak evidence. In practice, we find quantitatively and qualitatively that the strong scene prior nonetheless yields substantial gains over pure reconstruction baselines, with the prior's tendency to fill in plausible structure improving completeness and surface fidelity far more than occasional hallucinations detract from accuracy.

5 Conclusion

We presented a method for reconstructing scene-level PBR meshes from posed multi-view images by lifting an object-level 3D generative prior to scene scale. Our key idea is to formulate scene reconstruction as the joint generation of overlapping spatial chunks, conditioned on the input views through a projection-based 3D conditioning pathway that injects multi-view image features into the generative model in a spatially grounded manner. Combined with parameter-efficient LoRA fine-tuning on synthetic indoor data, this enables faithful, editable PBR mesh reconstructions that inherit the fidelity and completeness of modern object-level generative priors while extending naturally to scene scale. We see this work as a step toward closing the gap between current reconstruction methods and the quality required for downstream graphics, simulation, and embodied AI pipelines.

Acknowledgements

Katharina Schmid is a doctoral fellow at the Munich Data Science Institute (MDSI), and Nicolas von Lützow was supported by the MDSI focus topic *Understanding Existing Structures in Building Planning* (USP). Angela Dai was supported by the ERC Starting Grant SpatialSem (101076253). This work was further supported by the ERC Consolidator Grant *Gen3D* (101171131), by compute resources from the Jülich Supercomputing Center under project *MeshFoundation*, and by the Computing Systems Lab, part of the Huawei Technologies Switzerland AG.

References

[1] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction, 2023. URL https://arxiv.org/abs/2306.03092.

[2] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023. URL https://arxiv.org/abs/2106.10689.

[3] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces, 2021. URL https://arxiv.org/abs/2106.12052.

[4] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction, 2022. URL https://arxiv.org/abs/2206.00665.

[5] Hanlin Chen, Chen Li, Yunsong Wang, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance, 2025. URL https://arxiv.org/abs/2312.00846.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. URL https://arxiv.org/abs/2308.04079.

[7] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields, 2025. URL https://arxiv.org/abs/2403.17888.

[8] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction, 2025. URL https://arxiv.org/abs/2406.06521.

[9] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. URL https://arxiv.org/abs/2312.14132.

[10] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. URL https://arxiv.org/abs/2406.09756.

[11] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. URL https://arxiv.org/abs/2503.11651.

[12] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025. URL https://arxiv.org/abs/2511.10647.

[13] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video, 2021. URL https://arxiv.org/abs/2104.00681.

[14] Noah Stier, Anurag Ranjan, Alex Colburn, Yajie Yan, Liang Yang, Fangchang Ma, and Baptiste Angles. Finerecon: Depth-aware feed-forward network for detailed 3d reconstruction, 2023. URL https://arxiv.org/abs/2304.01480.

[15] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction, 2024. URL https://arxiv.org/abs/2312.12337.

[16] 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, 2024. URL https://arxiv.org/abs/2403.14627.

[17] 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, 2025. URL https://arxiv.org/abs/2505.23716.

[18] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models, 2024. URL https://arxiv.org/abs/2404.07191.

[19] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization, 2023. URL https://arxiv.org/abs/2306.16928.

[20] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image, 2024. URL https://arxiv.org/abs/2309.03453.

[21] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo, Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu, Changrong Hu, Shaoxiong Yang, Song Zhang, Yang Liu, Tianyu Huang, Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He, Tian Liu, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. URL https://arxiv.org/abs/2501.12202.

[22] Ziwei Liao, Mohamed Sayed, Steven L. Waslander, Sara Vicente, Daniyar Turmukhambetov, and Michael Firman. Complete gaussian splats from a single image with denoising diffusion models, 2025. URL https://arxiv.org/abs/2508.21542.

[23] Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation, 2025. URL https://arxiv.org/abs/2510.23306.

[24] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation, 2025. URL https://arxiv.org/abs/2512.14692.

[25] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016. IEEE. doi: 10.1109/cvpr.2016.445.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL https://arxiv.org/abs/2003.08934.

[27] Rafał Tobiasz, Grzegorz Wilczyński, Marcin Mazur, Sławomir Tadeja, Weronika Smolak-Dyżewska, and Przemysław Spurek. Meshsplats: Mesh-based rendering with gaussian splatting initialization, 2026. URL https://arxiv.org/abs/2502.07754.

[28] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state, 2025. URL https://arxiv.org/abs/2501.12387.

[29] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting, 2026. URL https://arxiv.org/abs/2510.10726.

[30] Aljaž Božič, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers, 2021. URL https://arxiv.org/abs/2107.02191.

[31] Noah Stier, Alexander Rich, Pradeep Sen, and Tobias Höllerer. Vortx: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion, 2021. URL https://arxiv.org/abs/2112.00236.

[32] Huiyu Gao, Wei Mao, and Miaomiao Liu. Visfusion: Visibility-aware online 3d scene reconstruction from videos, 2023. URL https://arxiv.org/abs/2304.10687.

[33] Youngju Na, Woo Jae Kim, Kyu Beom Han, Suhyeon Ha, and Sung eui Yoon. Uforecon: Generalizable sparse-view surface reconstruction from arbitrary and unfavorable sets, 2024. URL https://arxiv.org/abs/2403.05086.

[34] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions, 2022. URL https://arxiv.org/abs/2208.14743.

[35] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth, 2025. URL https://arxiv.org/abs/2410.13862.

[36] 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, 2024. URL https://arxiv.org/abs/2410.24207.

[37] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting, 2025. URL https://arxiv.org/abs/2410.22128.

[38] Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting, 2025. URL https://arxiv.org/abs/2511.07321.

[39] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat++: Generalizable 3d gaussian splatting for efficient indoor scene reconstruction, 2025. URL https://arxiv.org/abs/2503.22986.

[40] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models, 2024. URL https://arxiv.org/abs/2405.10314.

[41] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors, 2023. URL https://arxiv.org/abs/2312.02981.

[42] Haoyu Guo, He Zhu, Sida Peng, Haotong Lin, Yunzhi Yan, Tao Xie, Wenguan Wang, Xiaowei Zhou, and Hujun Bao. Multi-view reconstruction via sfm-guided monocular depth estimation, 2025. URL https://arxiv.org/abs/2503.14483.

[43] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos, 2024. URL https://arxiv.org/abs/2409.02095.

[44] Tian-Xing Xu, Xiangjun Gao, Wenbo Hu, Xiaoyu Li, Song-Hai Zhang, and Ying Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors, 2025. URL https://arxiv.org/abs/2504.01016.

[45] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model, 2025. URL https://arxiv.org/abs/2408.16767.

[46] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis, 2024. URL https://arxiv.org/abs/2409.02048.

[47] Baicheng Li, Dong Wu, Jun Li, Shunkai Zhou, Zecui Zeng, Lusong Li, and Hongbin Zha. Mv-sam3d: Adaptive multi-view fusion for layout-aware 3d generation, 2026. URL https://arxiv.org/abs/2603.11633.

[48] Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3d: Pixel-aligned 3d generation from images, 2026. URL https://arxiv.org/abs/2605.10922.

[49] Weirong Chen, Chuanxia Zheng, Ganlin Zhang, Andrea Vedaldi, and Daniel Cremers. Nova3r: Non-pixel-aligned visual transformer for amodal 3d reconstruction, 2026. URL https://arxiv.org/abs/2603.04179.

[50] Qihang Zhang, Chaoyang Wang, Aliaksandr Siarohin, Peiye Zhuang, Yinghao Xu, Ceyuan Yang, Dahua Lin, Bolei Zhou, Sergey Tulyakov, and Hsin-Ying Lee. Scenewiz3d: Towards text-guided 3d scene composition, 2023. URL https://arxiv.org/abs/2312.08885.

[51] Xiaoyu Zhou, Xingjian Ran, Yajiao Xiong, Jinlin He, Zhiwei Lin, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Gala3d: Towards text-to-3d complex scene generation via layout-guided generative gaussian splatting, 2024. URL https://arxiv.org/abs/2402.07207.

[52] Xiao-Lei Li, Haodong Li, Hao-Xiang Chen, Tai-Jiang Mu, and Shi-Min Hu. Discene: Object decoupling and interaction modeling for complex scene generation. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687589. URL https://doi.org/10.1145/3680528.3687589.

[53] Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance, 2024. URL https://arxiv.org/abs/2403.12409.

[54] Haonan Han, Rui Yang, Huan Liao, Jiankai Xing, Zunnan Xu, Xiaoming Yu, Junwei Zha, Xiu Li, and Wanhua Li. Reparo: Compositional 3d assets generation with differentiable 3d layout alignment, 2025. URL https://arxiv.org/abs/2405.18525.

[55] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Wei Yang, Lan Xu, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image, 2025. URL https://arxiv.org/abs/2502.12894.

[56] Edoardo Alberto Dominici, Jozef Hladky, Floor Verhoeven, Lukas Radl, Thomas Deixelberger, Stefan Ainetter, Philipp Drescher, Stefan Hauswiesner, Arno Coomans, Giacomo Nazzaro, Konstantinos Vardis, and Markus Steinberger. Dreamanywhere: Object-centric panoramic 3d scene generation, 2025. URL https://arxiv.org/abs/2506.20367.

[57] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[58] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

[59] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

[60] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[61] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation, 2023. URL https://arxiv.org/abs/2302.08113.

[62] Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. Sage: Scalable agentic 3d scene generation for embodied ai, 2026. URL https://arxiv.org/abs/2602.10116.

[63] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[64] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Qixun Zeng, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, Yi Liu, Peng Liu, Lin Ma, Le Weng, Xiaohang Hu, Xin Ma, Qian Qian, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*, 2020.

[65] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[66] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[67] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[68] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[70] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[71] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation, 2025. URL https://arxiv.org/abs/2412.03558.

[72] Lu Ling, Yunhao Ge, Yichen Sheng, and Aniket Bera. I-scene: 3d instance models are implicit generalizable spatial learners, 2026. URL https://arxiv.org/abs/2512.13683.

[73] Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, Richard Newcombe, and Jakob Engel. Shaper: Robust conditional 3d shape generation from casual captures, 2026. URL https://arxiv.org/abs/2601.11514.

[74] Zador Pataki, Paul-Edouard Sarlin, Johannes L. Schönberger, and Marc Pollefeys. Mp-sfm: Monocular surface priors for robust structure-from-motion, 2025. URL https://arxiv.org/abs/2504.20040.

[75] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments, 2017. URL https://arxiv.org/abs/1709.06158.

[76] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[77] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021.

[78] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

A Experimental Setup

Implementation Details. We build on Trellis.2 [24] at resolution 512. Consequently, for the occupancy generation, we use a dense latent grid of resolution 16, and for the shape and texture generation, a sparse latent grid of resolution 32 per chunk. We apply LoRA adapters [60] to all attention layers with rank 8 and scaling 16. The 3D condition is injected into each of the 30 DiT blocks through a per-block zero-initialized projection that is residually added to the block's current representation. The aggregation network consists of two parallel MLPs sharing the same per-view input: a three-layer feature MLP and a two-layer scalar MLP that produces per-view aggregation logits, which are softmaxed across views to form attention-style aggregation weights.

Training Details. We train with a global batch size of 32. We use AdamW [66] with a learning rate of 1×10^{-4} for newly introduced layers and 3×10^{-5} for LoRA parameters. The learning rate decay is set to 0.01 for new parameters and 0.0 for LoRA parameters. During training, the number of conditioning views per chunk is uniformly sampled from 1 to 16. We train with classifier-free guidance [67], dropping the condition with a probability of 0.1.

Inference Details. The chunk size is estimated based on the sparse point cloud from SfM. Specifically, we choose the chunk size as $1.11 \times$ scene height. Chunks are placed such that neighboring chunks overlap by a prescribed minimum margin $m = 0.25$. For shape and texture generation, we apply a boundary-sensitive variant of the voxel feature synchronization. Specifically, the boundary-sensitive variant modifies the MultiDiffusion [61] overlap aggregation by excluding voxels in the outermost $b = 1$ rows of each chunk's x/y faces from contributing to the cross-chunk mean, since these boundary regions are systematically biased by the network's lack of receptive-field context beyond the chunk extent. Every voxel, including boundaries, still receives the resulting interior-only average wherever at least one chunk's interior covers it, so seams are healed by trustworthy interior predictions while voxels with no interior coverage fall back to their own chunk's estimate. Following the original Trellis.2 implementation, we use 12 flow-matching steps for each model at inference time.

Dataset Curation. We use a subset of 5,000 rooms from SAGE-10k [62]. For each room, we randomly place light sources, sample random camera poses with a per-room sampled focal length, and render the resulting RGB images. For each room, we extract three randomly-rotated training chunks, yielding 15,000 chunk-level training examples in total. Additionally, we include a subset of scenes from 3D-FRONT [64]: 1,200 houses, 3 chunks per house, used only for the occupancy stage.

Inference-time Compute and Memory. We analyse the compute and memory resources of our method at inference time. The experiments are run on a single GPU with 80 GB VRAM using 8 input views unless stated otherwise. The times are synchronized and cover the pipeline only (exluding checkpoint loading and mesh writing). Memory refers to device-level peak, i.e., what the card must supply. Table 6 shows the effect of the scene size. Runtime and peak VRAM scale approximately linearly with the number of occupied voxels, which predicts cost better than chunk count. Typical medium scenes (60k voxels) run 2 min / 24 GB, so scenes up to this size fit on a 24 GB consumer GPU. Table 7 shows the effect of the number of input views. The runtime across different numbers of input views is essentially flat because the generation iterates over chunks, not images; views only enter through the conditioning tensor. VRAM grows mildly.

Table 6: **Runtime and peak VRAM vs. scene size.** We report runtime and peak GPU memory on 16 scenes from ScanNet++.

Chunks	Occupied voxels	Runtime (min:s)	Peak VRAM (GB)
2	12,016	0:33	6.5
4	19,312	0:53	8.5
8	60,831	2:06	23.8
12	67,682	2:27	23.4
29	126,672	5:52	45.0
22	145,052	6:38	61.2

Table 7: **Runtime and peak VRAM vs. number of input views.** We report runtime and peak GPU memory for a 6-chunk and a 12-chunk scene from ScanNet++.

	6-chunk scene		12-chunk scene	
Input views	Runtime (min:s)	VRAM (GB)	Runtime (min:s)	VRAM (GB)
1	1:00	10.5	2:04	20.9
4	1:10	12.4	2:43	29.9
8	1:27	12.6	2:42	30.5
32	1:23	13.3	2:34	26.4
128	1:24	24.2	2:43	26.8

B Evaluation Details

We evaluate reconstruction quality both on unseen 2D test views and directly on the extracted 3D meshes.

In the 2D setting, we render depth and normal maps from the reconstructed meshes at held-out viewpoints. For depth, we report the mean absolute error (MAE) and root mean squared error (RMSE) in meters, together with their relative counterparts AbsRel and SqRel, measuring depth accuracy. For normals, we report the angular normal error in degrees to assess view-space surface orientations. We additionally measure perceptual and semantic agreement with the ground-truth views using LPIPS [68] with an AlexNet backbone [69] and CLIP score [70] on both depth and normal renderings. Finally, we report prediction completeness. All metrics are computed only over pixels where ground-truth views are defined, excluding empty or unobserved regions and avoiding ambiguities caused by windows and mirrors.

In the 3D setting, we evaluate the Chamfer distance to measure average 3D alignment and the F-score to measure areas within a reasonable tolerance. Additionally, we evaluate normal consistency between nearest-neighbor surface points to measure local surface orientation.

Masking. For 2D evaluation, views are rendered at 876×584 resolution on ScanNet++ and at 512×512 on 3D-FRONT. Pixel-wise depth and normal metrics are computed only over the intersection of valid ground-truth and predicted pixels, such that they measure reconstruction quality where both meshes produce a hit. Missing predictions are instead captured by the separate completeness metric. For perceptual and image-level metrics such as LPIPS, FID/KID, and CLIP score, we apply the GT mask to the predicted images to avoid punishing results in regions that are undefined for the GT laser scan.

For 3D evaluation, metrics are computed inside a per-scene observation envelope rather than over the full extent of the predicted mesh. On ScanNet++, this envelope is derived from the full-scanner fusion at 1.1 cm voxel resolution and dilated by 15 cm, yielding a permissive region around observed surfaces that gates evaluation without acting as a tight accuracy threshold. Predicted meshes are clipped to this envelope before sampling, removing geometry outside the scanned region. For 3D-FRONT, we instead use the ground-truth mesh axis-aligned bounding box inflated by 20%.

2D Metrics. We aggregate image-space metrics over all valid pixels pooled across all test frames. Let $d_{f,i}$ and $n_{f,i}$ denote the predicted depth and view-space unit normal at pixel i in frame f, with $d^*_{f,i}$ and $n^*_{f,i}$ denoting the corresponding ground truth. The set $\mathcal{I}$ contains all frame-pixel pairs where

both predicted and ground-truth depth are valid; depth and normal errors are averaged over $\mathcal{I}$.

$$\text{MAE:} \quad \frac{1}{|\mathcal{I}|} \sum_{(f,i)\in\mathcal{I}} \left| d_{f,i} - d^*_{f,i} \right|, \tag{4}$$

$$\text{RMSE:} \quad \sqrt{\frac{1}{|\mathcal{I}|} \sum_{(f,i)\in\mathcal{I}} \left(d_{f,i} - d^*_{f,i} \right)^2}, \tag{5}$$

$$\text{AbsRel:} \quad \frac{1}{|\mathcal{I}|} \sum_{(f,i)\in\mathcal{I}} \frac{\left| d_{f,i} - d^*_{f,i} \right|}{d^*_{f,i}}, \tag{6}$$

$$\text{SqRel:} \quad \frac{1}{|\mathcal{I}|} \sum_{(f,i)\in\mathcal{I}} \frac{\left(d_{f,i} - d^*_{f,i} \right)^2}{d^*_{f,i}}, \tag{7}$$

$$\text{Normal error:} \quad \frac{1}{|\mathcal{I}|} \sum_{(f,i)\in\mathcal{I}} \arccos\left(n_{f,i}^\top n^*_{f,i} \right) \cdot \frac{180}{\pi}. \tag{8}$$

For LPIPS and CLIP score, we compare saved ground-truth and predicted renderings per held-out test view and average the resulting scores over all views. Let $\mathcal{F}$ denote the set of evaluated test frames, m the evaluated modality, i.e., depth or normal, and I_f^m and $I_f^{m,*}$ the predicted and ground-truth rendered images for modality m at frame f.

$$\text{LPIPS}_m: \quad \frac{1}{|\mathcal{F}|} \sum_{f\in\mathcal{F}} \text{LPIPS}\left(I_f^m, I_f^{m,*} \right), \tag{9}$$

$$\text{CLIP}_m: \quad \frac{1}{|\mathcal{F}|} \sum_{f\in\mathcal{F}} \frac{\phi_{\text{CLIP}}(I_f^m)^\top \phi_{\text{CLIP}}(I_f^{m,*})}{\left\| \phi_{\text{CLIP}}(I_f^m) \right\|_2 \left\| \phi_{\text{CLIP}}(I_f^{m,*}) \right\|_2}. \tag{10}$$

3D Metrics. For 3D evaluation, we uniformly sample 200k points from both the predicted and ground-truth meshes and compute nearest-neighbor distances in both directions.

Chamfer distance is reported as the mean of the predicted-to-ground-truth and ground-truth-to-predicted nearest-neighbor distances in meters.

For F-score@10cm, we compute precision as the fraction of predicted samples within 0.1 m of the ground-truth surface and recall as the fraction of ground-truth samples within 0.1 m of the predicted surface, and then report their mean.

Normal consistency is computed symmetrically over the same nearest-neighbor correspondences as the average absolute normal dot product in both directions. To avoid rewarding geometrically unrelated but similarly oriented surfaces, normal agreement is set to zero for nearest-neighbor pairs farther than 0.2 m.

C Additional Quantitative Results

Robustness Analysis. To assess the robustness of our evaluation, we provide the per-scene standard deviations for our 2D (see Table 8) and 3D evaluation (see Table 9) on ScanNet++. Notably, GenRecon has the lowest per-scene standard deviation among all methods. Since all methods are evaluated on identical scenes, the per-scene deviations mainly reflect variation in scene difficulty. Therefore, in Table 10, we additionally provide the per-scene wins for the five best-performing methods.

Observed vs. Unobserved Region Analysis. Table 11 analyzes the reconstruction performance of our method in regions that were observed versus regions that were not observed by input views. A point is classified as observed if it lies in the frustum of at least one input camera and has an unoccluded line of sight from that camera (occlusion tested against the mesh); this mask is computed independently for the GT and predicted meshes, each against its own geometry. The mask is applied only to the query set, while nearest neighbors are always searched in the full opposite set. Concretely,

Table 8: **2D robustness analysis on real-world data.** We evaluate 2D depth and normal metrics on 50 scenes from ScanNet++.

		2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Depth	MAE (m) ↓	0.2717 ± 0.1384	0.3451 ± 0.4604	0.1612 ± 0.0978	0.1352 ± 0.1008	**0.1336 ± 0.0874**	0.1364 ± 0.0879
	RMSE (m) ↓	0.5074 ± 0.2323	0.5080 ± 0.5059	0.3118 ± 0.1699	0.2901 ± 0.1825	0.3199 ± 0.2134	**0.2830 ± 0.1753**
	AbsRel ↓	0.2437 ± 0.1072	0.4772 ± 0.6970	0.2131 ± 0.1302	0.1775 ± 0.1262	0.1580 ± 0.1249	**0.1514 ± 0.0933**
	SqRel ↓	0.1970 ± 0.1286	0.6915 ± 1.5304	0.1634 ± 0.1650	0.1511 ± 0.1599	0.1433 ± 0.1774	**0.1007 ± 0.1107**
	LPIPS ↓	0.3885 ± 0.0937	0.1426 ± 0.1159	0.3413 ± 0.1163	0.2112 ± 0.1123	0.3335 ± 0.0919	**0.1114 ± 0.0633**
	CLIP score ↑	0.8881 ± 0.0357	0.9659 ± 0.0265	0.9216 ± 0.0179	0.9301 ± 0.0346	0.9074 ± 0.0251	**0.9765 ± 0.0107**
Normal	Angular err. (°) ↓	36.58 ± 4.78	22.20 ± 8.93	28.81 ± 6.18	21.03 ± 4.88	25.67 ± 6.05	**18.39 ± 5.84**
	LPIPS ↓	0.7296 ± 0.0775	0.4465 ± 0.1353	0.6837 ± 0.0754	0.5210 ± 0.0861	0.6460 ± 0.0853	**0.3685 ± 0.0788**
	CLIP score ↑	0.8603 ± 0.0331	0.9095 ± 0.0312	0.8967 ± 0.0138	0.8868 ± 0.0207	0.8806 ± 0.0235	**0.9411 ± 0.0141**
	Completeness ↑	0.6780 ± 0.1927	0.9496 ± 0.1142	0.8065 ± 0.1156	0.8367 ± 0.1095	0.6819 ± 0.1542	**0.9864 ± 0.0216**

Table 9: **3D robustness analysis on real-world data.** We evaluate 3D reconstruction metrics on 50 scenes from ScanNet++.

	2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Chamfer (m) ↓	0.2597 ± 0.1568	0.1633 ± 0.1815	0.1291 ± 0.0721	0.1106 ± 0.0676	0.1213 ± 0.0574	**0.0773 ± 0.0309**
F-score @10cm ↑	0.4685 ± 0.1287	0.6106 ± 0.3102	0.6366 ± 0.1980	0.7004 ± 0.1675	0.6858 ± 0.1345	**0.7684 ± 0.0838**
Normal consistency ↑	0.4794 ± 0.0709	0.6704 ± 0.2126	0.6705 ± 0.1007	0.7467 ± 0.0922	0.6804 ± 0.0931	**0.7587 ± 0.0760**

per bucket (observed/unobserved), we compute *masked prediction → full GT (precision)* and *masked GT → full prediction (recall)*; both directions enter the bidirectional Chamfer distance, the F-score, and the normal consistency. GT structure unseen by all input views is therefore part of the unobserved GT query set, so unrecovered geometry is directly penalized on the recall side, and wrong completions are penalized by their distance to the true GT surface. We make two observations: (1) The prior does not hurt observed regions. GenRecon achieves the best observed-region results on all three metrics. (2) The prior helps most in unobserved regions, widening GenRecon's margin substantially. Notably, a much larger fraction of GenRecon's predicted surface lies in unobserved regions than for any baseline. This means it genuinely completes unseen geometry and does so accurately.

Comparison with Additional Reconstruction Baselines. Following our DA3 setup, we additionally evaluate against VGGT [11] (pose-free and with ground truth cameras) and AnySplat [17] (with GT cameras, meshing both its rendered depth and its depth head). Table 12 and Table 13 show that GenRecon remains best.

Comparison with Object-Wise 3D Generative Baselines. We show a comparison against object-wise scene generation baselines on 10 ScanNet++ scenes. To match the baselines' setting, we exclude structural elements (walls, floors, ceilings, windows) and small clutter, and evaluate one-sided metrics using GT-mesh samples as queries, since the GT meshes are incomplete. All baselines receive GT instance masks from ScanNet++ annotations. For single-view baselines, we compare against MIDI [71] and I-Scene [72]. We retain the 10 instances with the largest visible masks, as both degrade on small masks. Table 14 shows that GenRecon outperforms both by a large margin. We attribute this largely to boundary-truncated objects, which are common in ScanNet++ but not present in these methods' training data. Table 15 shows a comparison with the multi-view baseline ShapeR [73]. We limit the objects per scene to the 15 largest instances. ShapeR receives the same 8 images and GT cameras, plus GT instance masks and depth estimates from Depth Anything v3 [12], which GenRecon does not use. GenRecon clearly outperforms ShapeR despite requiring neither masks nor

Table 10: **Per-scene performance comparison.** We evaluate on 50 ScanNet++ scenes and determine the wins based on Chamfer Distance.

Method	Wins	Win rate	2nd place
Ours	32	64%	9
FineRecon	6	12%	17
VGGT (GT cameras)	6	12%	5
DA3	4	8%	9
Murre	2	4%	10

Table 11: **Reconstruction analysis of observed vs. unobserved regions.** We evaluate 3D reconstruction metrics separately on observed and unobserved regions of 25 scenes from ScanNet++.

		2DGS	MonoSDF	DA3	FineRecon	Murre	Ours
Observed	Chamfer (m) ↓	0.1505	0.1254	0.0752	0.0673	0.0754	**0.0549**
	F-score @10cm ↑	0.5687	0.6524	0.7512	0.8285	0.7921	**0.8843**
	Normal consistency ↑	0.5385	0.7378	0.7690	0.8340	0.7654	**0.8444**
Unobserved	Chamfer (m) ↓	0.2221	0.1307	0.1676	0.1219	0.1603	**0.0735**
	F-score @10cm ↑	0.3467	0.5601	0.4707	0.5819	0.4745	**0.7137**
	Normal consistency ↑	0.3561	0.5802	0.5388	0.6257	0.5024	**0.7215**
Coverage	In-frustum fraction	1.0000	0.9287	1.0000	0.9504	1.0000	0.9469
	Observed fraction	0.8919	0.7129	0.8056	0.8030	0.8622	0.4193

Table 12: **2D baseline comparisons against additional reconstruction baselines on real-world data.** We evaluate 2D depth and normal metrics on 50 scenes from ScanNet++.

		VGGT	AnySplat	DA3	Ours
Depth	MAE (m) ↓	0.1769	0.2422	0.1612	**0.1364**
	RMSE (m) ↓	0.3283	0.4000	0.3118	**0.2830**
	AbsRel ↓	0.2102	0.2743	0.2131	**0.1514**
	SqRel ↓	0.1596	0.2020	0.1634	**0.1007**
	LPIPS ↓	0.3319	0.3752	0.3413	**0.1114**
	CLIP score ↑	0.9210	0.9085	0.9216	**0.9765**
Normal	Angular err. (°) ↓	29.97	36.63	28.81	**18.39**
	LPIPS ↓	0.6890	0.7354	0.6837	**0.3685**
	CLIP score ↑	0.8829	0.8680	0.8967	**0.9411**
	Completeness ↑	0.8074	0.7841	0.8065	**0.9864**

depth, while additionally reconstructing structural elements and producing relightable PBR materials. These results empirically support our claim: a single coherent scene-level generation outperforms per-object composition in realistic conditions.

SfM Dependence. In the main evaluation, all methods (ours and all baselines) use the dense-capture COLMAP outputs provided by ScanNet++, so comparisons are consistent. To test dependence on this, we rerun GenRecon with poses and sparse points estimated from the 8 input views alone. Since classical COLMAP is unreliable at 8 views, we use MP-SfM [74] (SuperPoint+MASt3R matching, fixed intrinsics), which registered all views in the 10 scenes drawn from the ScanNet++ validation set. Table 16 shows nearly identical performance, demonstrating that GenRecon is robust to realistic camera pose estimates obtained from sparse-view SfM.

Matterport3D Results. Beyond ScanNet++, we evaluate on Matterport3D [75] as an additional real-world dataset. We evaluate on 25 diverse rooms from 20 houses. As Matterport3D's FoV is much narrower than ScanNet++'s DSLR captures, we use 16 input views per scene. All methods receive the same views, poses, and GT meshes, and since Matterport3D provides no SfM, baselines requiring it receive a pseudo-SfM model synthesized from GT (favoring, e.g., Murre's sparse-depth

Table 13: **3D baseline comparisons against additional reconstruction baselines on real-world data.** We evaluate 3D reconstruction metrics on 50 scenes from ScanNet++.

	VGGT	AnySplat	DA3	Ours
Chamfer (m) ↓	0.1294	0.1720	0.1291	**0.0773**
F-score @10cm ↑	0.6420	0.5084	0.6366	**0.7684**
Normal consistency ↑	0.6527	0.5514	0.6705	**0.7587**

Table 14: **Single-view baseline comparison.** We evaluate one-sided 3D reconstruction metrics on 10 scenes from ScanNet++, only considering the 10 objects with the largest visibility masks.

	MIDI (1 image)	I-Scene (1 image)	Ours (8 images)
Completeness L1 (m) ↓	0.2274	0.1606	**0.0410**
R-score @10cm ↑	0.4325	0.5928	**0.9482**
Normal consistency (20cm cap) ↑	0.3854	0.5008	**0.7821**

Table 15: **Multi-view baseline comparison.** We evaluate one-sided 3D reconstruction metrics on 10 scenes from ScanNet++ only considering the 10 objects with the largest visibility masks.

	ShapeR (DA3)	Ours
Completeness L1 (m) ↓	0.0621	**0.0435**
R-score @10cm ↑	0.8367	**0.9342**
Normal consistency (20cm cap) ↑	0.6879	**0.7584**

conditioning). Since each camera rig carries its own calibration, we align principal points by cropping for the baselines; this is resampling-free with a median cost of 1.5% of pixels, all at the image edges. Because the Matterport3D meshes are incomplete, we report one-sided metrics using GT-mesh samples as queries. Table 17 shows that GenRecon leads on all metrics on this second real-world dataset, despite the domain shift to narrow-FoV panorama-rig captures.

Object-Level Transfer on DTU. To investigate the transferability of our approach, we evaluate on DTU [76]. GenRecon is applied zero-shot, despite DTU differing from our indoor-scene training distribution. We compare against Trellis.2, extended to multiple views in two naive ways: stochastic, which conditions each denoising step on a single input image, cycling through the views round-robin across steps; and multidiffusion, which runs the model once per input image at every step and averages the velocity predictions before applying the usual CFG (not to be confused with our chunk-level MultiDiffusion). We evaluate on the standard 15-scan DTU subset against the structured-light GT, restricted to DTU's observed volume above the table. At matched view budgets, both methods see identical inputs. GenRecon is scored directly in the DTU frame; Trellis.2 has no metric scale and is granted an oracle alignment to the GT. Table 18 demonstrates that the projection-based conditioning transfers well beyond the intended domain: zero-shot GenRecon outperforms Trellis.2 for every view count, and accuracy continues to improve with the number of input views.

PBR Material Quality. To investigate the PBR quality, we evaluate 10 held-out SAGE-10k scenes, which include full GT albedo, roughness, and metallic maps. We render the material passes of the predicted and GT meshes from 10 held-out views per scene and compare per-pixel (see Table 19).

D Additional Ablation Studies

Number of Input Images and 3D Conditioning.

Chunk Size. The chunk size c is not a free parameter: it must roughly match the room height, which we estimate from the SfM points. GenRecon is trained mainly on SAGE-10k, whose rooms

Table 16: **Pose-source ablation.** We evaluate 3D reconstruction metrics on 10 scenes from ScanNet++ using camera poses from full-scene SfM vs. SfM on the 8 input views only.

	Ours (full-scene SfM poses)	Ours (poses from 8 inputs only)
Chamfer (m) ↓	0.0720	**0.0708**
F-score @10cm ↑	**0.7826**	0.7574
Normal consistency ↑	**0.7957**	0.7922

Table 17: **3D baseline comparisons on Matterport3D.** We evaluate one-directional (GT → prediction) 3D reconstruction metrics on 25 scenes from Matterport3D.

	DA3	FineRecon	Murre	Ours
Completeness (m) ↓	0.2265	0.1654	0.2126	**0.0977**
Recall @10cm ↑	0.5532	0.6375	0.5696	**0.7274**
Normal consistency ↑	0.5642	0.6476	0.5690	**0.6995**

Table 18: **3D reconstruction on DTU.** We evaluate 3D reconstruction metrics on 15 scans from DTU for different numbers of input views. For TRELLIS.2 with multiple views, we compare stochastic and multidiffusion conditioning.

Method	Views	Acc. (mm) ↓	Comp. (mm) ↓	Overall (mm) ↓	F@4mm ↑	F@8mm ↑	Normal cons. ↑
TRELLIS.2	1	11.90	12.89	12.39	0.176	0.332	0.307
	2 (stochastic)	11.66	13.21	12.43	0.174	0.332	0.291
	2 (multidiffusion)	11.44	12.44	11.94	0.196	0.366	0.308
	4 (stochastic)	12.45	13.18	12.82	0.168	0.312	0.273
	4 (multidiffusion)	11.98	12.65	12.31	0.191	0.347	0.300
GenRecon	1	7.33	11.72	9.52	0.291	0.486	0.445
	2	6.81	8.85	7.83	0.422	0.632	0.544
	4	5.36	7.42	6.39	0.529	0.711	0.623
	8	4.96	5.72	5.34	0.590	0.810	0.677
	16	4.39	5.61	5.00	0.585	0.814	**0.702**
	32	**4.30**	**5.54**	**4.92**	**0.603**	**0.819**	0.699

have a fixed height of $2.7\,\mathrm{m}$; thus, the training data covers $c \in [2.7, 3.0]\,\mathrm{m}$. Since real-world room heights vary, we assess robustness outside this range by binning 50 ScanNet++ scenes according to their estimated chunk size (see Table 20). We find that performance is best for chunk sizes at or below the training range and degrades gracefully beyond this range. Each chunk holds a fixed latent-voxel grid, so larger chunks imply a coarser effective resolution, limiting applicability to room heights below $5\,\mathrm{m}$. Note that bins contain different scenes, so scene difficulty is not fully controlled.

Overlap Margin and MultiDiffusion. Table 21 provides an ablation on MultiDiffusion and the overlap margin. We vary the margin from 0 to 8 latent voxels (0–50% of the 16-voxel chunk; $m = 0.25$ in the paper corresponds to 4 voxels). Note that our boundary-sensitive denoising (BS) requires a margin of at least 2 voxels. Compared to independent chunk generation (overlap margin 0), MultiDiffusion improves all metrics and avoids visible seams at chunk boundaries. Accuracy improves monotonically up to a margin of 4 voxels (25%) and saturates beyond, supporting our default choice. Unlike the fixed-margin variants, which hold the overlap constant and let the chunk row overshoot the scene, our adaptive scheme fits the minimal number of chunks to the scene extent and absorbs the slack as extra overlap, floored at 25% to guarantee a sufficient blending seam, yielding the best overall results.

Table 19: **PBR material evaluation on SAGE-10k.** We evaluate generated albedo, roughness, and metallic maps on 10 held-out scenes with 10 held-out images each.

		Generated
Albedo	PSNR ↑	21.91
	SSIM ↑	0.699
	LPIPS ↓	0.347
	RMSE ↓	0.084
Roughness	RMSE ↓	0.076
	SSIM ↑	0.953
Metallic	RMSE ↓	0.096
	SSIM ↑	0.910

Table 20: **Effect of chunk size.** We evaluate 3D reconstruction metrics on 50 scenes from ScanNet++, grouped by chunk size.

Chunk size (m)	<2.7	2.7–3.0	3.0–3.3	3.3–3.7	3.7–4.0	≥4.0
# Scenes	5	14	11	9	7	4
Chamfer (m) ↓	**0.049**	0.073	0.074	0.090	0.082	0.102
F-score @10cm ↑	**0.859**	0.787	0.780	0.729	0.752	0.676
Normal consistency ↑	**0.805**	0.760	0.778	0.752	0.754	0.666

Table 21: **Overlap-margin ablation.** We evaluate 3D reconstruction metrics on 25 scenes from ScanNet++ for different overlap margins, with and without BS.

Overlap margin	0	1	2	2	4	8	Ours (adaptive ≥4)
BS	–	–	–	✓	✓	✓	✓
Chamfer (m) ↓	0.0800	0.0732	0.0700	0.0700	0.0667	0.0671	**0.0664**
F-score @10cm ↑	0.7530	0.7635	0.7639	0.7640	0.7879	0.7806	**0.7908**
Normal consistency ↑	0.7553	0.7761	0.7797	0.7800	0.7882	0.7900	**0.7915**

Camera-Chunk Association. No explicit camera-to-chunk matching is required in our design: we lift all input views into a single global 3D condition grid spanning the scene, which is then partitioned into chunk-specific conditions. Effectively, every camera whose frustum intersects a chunk contributes to it. To assess whether chunks can end up with little or no view coverage, we count, for each chunk, how many of the 8 cameras observe it (see Table 22). A camera is considered to observe a chunk if at least X% of the chunk's sparse-structure voxels project into the image frustum. This test does not account for occlusion. The results show that on ScanNet++, with its wide-FoV DSLR captures, 8 input images suffice for every chunk to receive 3D conditioning from multiple views in nearly all cases, ruling out pure hallucination.

Table 22: **Camera coverage per chunk.** Number of chunks observed by n of the 8 input cameras on 25 scenes (155 chunks) from ScanNet++, for different visibility thresholds.

Criterion	0	1	2	3	4	5	6	7	8	Mean
≥1%	0	1	1	9	13	31	25	36	39	6.14
≥5%	0	1	1	14	20	38	34	33	14	5.56
≥10%	0	1	9	18	25	38	41	23	0	4.97

3D-FRONT fine-Tuning. We train a variant whose occupancy generation model is fine-tuned on SAGE-10k only, with an identical number of training steps and all other stages unchanged. Table 23 shows that adding 3D-FRONT provides consistent improvement. We hypothesize this stems from SAGE-10k containing only scenes without door openings, windows, or multi-room layouts, structural elements that 3D-FRONT introduces and that are common in ScanNet++.

E Additional Qualitative Results

Large Scene Generation As visualized in Figure 7, our method yields high-fidelity results on large 3D indoor scenes from ScanNet++.

Relighting Additional relighting results are provided in Figure 8.

F Baseline Implementation Details

We evaluate all baselines in the same posed multi-view setting as our method. Whenever possible, we use the exact released configurations and only make changes required to adapt the methods to

Table 23: **3D-FRONT ablation.** We evaluate 3D reconstruction metrics on 25 scenes from ScanNet++ with and without 3D-FRONT in the training data.

	Ours	Ours w/o 3D-FRONT
Chamfer (m) ↓	**0.0664**	0.0771
F-score @10cm ↑	**0.7908**	0.7497
Normal consistency ↑	**0.7915**	0.7651



Figure 7: **Large Scene Generation.** Top-down view (left) and multiple close-ups (right).

our evaluation protocol. All such changes fix issues and make the methods applicable under our sparse-view evaluation; none of them weaken the baselines.

2D Gaussian Splatting [7]. We use the official 2D Gaussian Splatting implementation and optimize scenes independently. The hyperparameters follow the authors' Tanks-and-Temples large-scene preset, adjusting only the TSDF fusion to keep the top 50 clusters by triangle count, rather than just one, to avoid removing correct geometry from unconnected regions.

Depth-Anything-3 [12]. We run the pretrained *DA3NESTED-GIANT-LARGE-1.1* model to return RGB-D predictions jointly for all input images in a single forward pass. During TSDF fusion, we use a voxel size of 0.01 m, truncation distance 0.04 m, and maximum depth 4.5 m. We again keep the top 50 clusters of the resulting mesh. The predicted depths are also reused as input to FineRecon.

Murre [42]. We use the pre-trained checkpoint and default setup to generate depth estimates. The following TSDF fusion is mostly standard, but we lower the fusion consistency threshold from 3 to 2, since the default threshold frequently removes too much geometry when only a small number of views overlap. Sparse-depth conditioning is performed using the full available SfM point cloud, independent of the number of RGB frames used for reconstruction. For the synthetic input points, we apply view-based visibility masking to mimic the SIFT visibility tracks available for SfM outputs.

MonoSDF [4]. We use the official MonoSDF implementation with Omnidata [77] depth and normal supervision. Specifically, we use their ScanNet [78] configuration and optimize for the full 200k iterations.

FineRecon [14]. We use the official weights, which were trained on ScanNet. The released inference pipeline requires scene bounds, which we compute from the ground-truth mesh axis-aligned bounding box, enlarged by 20%.



Figure 8: **Additional relighting results.** Varying lighting configurations for scenes reconstructed from ScanNet++.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the main contributions and scope. We introduce a new approach to high-fidelity 3D scene reconstruction from multi-

view RGB images that tightly couples reconstruction with a strong generative 3D prior and demonstrate the performance on synthetic and real-world datasets.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed explicitly in Section 4.4, where we elaborate on limitations with regards to non-Lambertian surfaces, the chunk definition as well as potential hallucinations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All results described in the paper can be reproduced by following the mentioned implementation detilas. The datasets used for training and evaluation are publically available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: We plan to open-source our data preprocessing and training code upon acceptance. An anonymized version of the code can be provided upon request.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All information about data splits, hyperparameters and optimizer types are provided in the paper or appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Reporting formal statistical significance is not standard practice in the field, and the referenced baseline papers do not include such measures either.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the compute resources used for training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to all concerns mentioned in the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impact is discussed in Section 5

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no use of data or models with a high risk for missuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We build upon the model Trellis.2 [24], which is correctly credited and the license is respected. The datasets [64**?** , 65] used for training and evaluation are credited as well.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not include research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The method in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.