# CoLa3D: Composable Latent 3D Decomposition

**Xudong Cai**[1,3] **Yongcai Wang**[1,2*] **Deying Li**[1] **Chuanxia Zheng**[3*†]

[1]School of Information, Renmin University of China
[2]Hebei Key Laboratory of Real-virtual Integrated Autonomous Systems (RIAS)
[3]PVG, Nanyang Technological University

`https://master-cai.github.io/CoLa3D/`

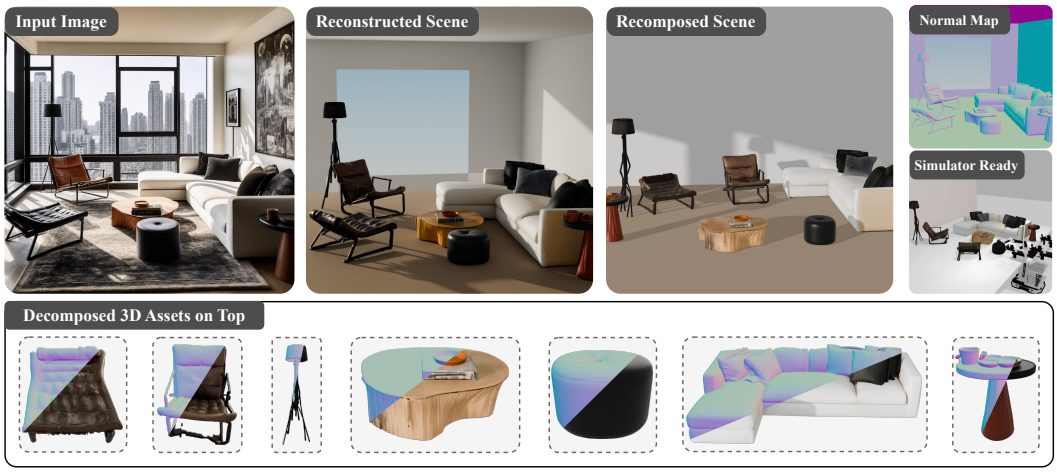


Figure 1: **Composable 3D scene generated by CoLa3D.** Given a single input image (*left*), CoLa3D decomposes its *whole*-scene latent into individual object meshes (*bottom*) within a shared coordinate frame, faithfully reproducing the input layout (*middle left*) while keeping each object physically accessible for downstream tasks such as rearrangement (*middle right*) and physical simulation (*right*).

## Abstract

We introduce CoLa3D, a simple and effective model for composable 3D scene decomposition that, given a single *whole*-scene latent from either a single image or an existing scene mesh as input, directly outputs a set of object meshes with consistent spatial layout that can be assembled back into a coherent scene. CoLa3D introduces a novel paradigm for 3D scene decomposition within a shared *whole*-scene latent, unlike alternatives that perform per-object generation in canonical spaces and then glue the objects back together with fragile pose and scale estimation. The network is a lightweight, flexible, and scalable query-based transformer that accepts various types of queries, including 2D masks, 3D points, and learnable queries, and learns to decompose the scene into objects in a compact latent space. On both synthetic and real-world datasets, CoLa3D achieves state-of-the-art performance on composable 3D scene decomposition, substantially improving global scene-level consistency and object-level quality while running $2\times$ faster and using less than $1\%$ of SAM 3D's training data. Visible depth on par with MoGe2 further confirms that latent decomposition preserves accurate visible scene geometry while producing physically complete, decomposable object meshes.

*Corresponding authors. †Project lead.

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

# 1 Introduction

We consider the problem of *composable 3D scene decomposition* from either an existing 3D mesh or a single image. The output represents the scene as a coherent whole and a collection of *physically plausible 3D instances*, namely object-level meshes together with their spatial arrangement (fig. 1). This is the form in which our physical world is organized: a living room, for example, is not merely a pile of *whole* scene geometry, but a structured arrangement of sofas, coffee tables, lamps, and other objects. Recovering such a composable 3D scene is important for embodied agents that reason jointly about objects and their relationships to interact with them in the physical world [23, 18, 9], as well as for AR/VR systems that require editable and interactive scene representations [43, 73, 78].

Traditional 3D decomposition, also known as *instance segmentation*, operates on raw 3D observations such as point clouds [46, 88, 35, 51, 91, 38], but often struggles when the geometry is noisy or incomplete. Another line of work reconstructs composable 3D scenes from images or RGB-D scans, but typically relies on retrieval-based pipelines [2, 22, 17] or per-scene optimization [90, 77, 73, 74]. These methods are difficult to scale in practice due to the limited coverage of retrieval libraries and the computational cost of test-time optimization. These methods either falter on noisy geometry or rely on costly external pipelines, motivating learning-based composable reconstruction from images.

Recent progress in 3D object generators [84, 86, 75, 63, 27, 60, 76] and 2D segmenters [20, 47, 4] has made composable 3D scene reconstruction from sparse views, or even a single image, increasingly plausible. Recent contributions like MIDI [16], CAST [80], SceneGen [39], and SAM 3D [6] follow a common recipe: they first segment objects in 2D image space using SAM [20], and then condition 3D generators on these masks to reconstruct object assets. While promising, the recomposed scene often suffers from spatial inconsistency and incoherence, as illustrated in fig. 3 and fig. 4. The core difficulty is that object geometry and scene-level scale, position, and pose are not jointly modeled in a shared 3D coordinate frame: object generators usually operate in *canonical* spaces, and the relative layout has to be glued back across separately-generated object latents. Very recent methods [16, 39, 6, 30] mitigate this issue by introducing cross-attention between object latents or by conditioning on global scene features, but they still couple objects only indirectly rather than decomposing a scene into object representations in the same coordinate frame. Moreover, these approaches must generate multiple object latents via diffusion models, making them computationally slow.

In this paper, we introduce a *novel paradigm* for joint object decomposition within a shared scene latent, instead of independent per-object generation in canonical spaces. We do so by introducing ***Co**mposable **La**tent **3D** decomposition* (**CoLa3D**), a feed-forward neural network that predicts a set of *object* latents from a single *whole-scene* latent in the scene coordinate frame, which are then decoded in *parallel* to produce *individual* meshes, as well as their recomposed *whole* scene mesh. Because its decomposition is performed in a *shared* scene coordinate frame, the decoded individual meshes are naturally more spatially consistent with the global scene geometry. Remarkably, its outputs exhibit less interpenetration and appear more physically plausible than MIDI, SceneGen, PartCrafter, and SAM 3D, making them readily usable in physical simulators for downstream tasks.

We instantiate this idea with a flexible query-based decomposition transformer that performs all object decomposition in a single forward pass. Concretely, CoLa3D accepts as input (i) the whole-scene latent, and (ii) promptable queries (*e.g.*, 2D masks, 3D points, or purely learnable queries) that interact with the latent tokens, and produces a set of object latents for a generative decoder to produce per-object meshes. To further improve generalization to novel categories and scenes, we incorporate 3D semantic features from Utonia [87] and 2D semantic features from DINOv3 [52].

To evaluate our model, we create challenging benchmarks on both synthetic (3D-FRONT [11] and MetaScenes [83]) and real-world datasets (Replica [56] and ScanNet++ [81]). CoLa3D achieves state-of-the-art performance on both object asset quality and global scene consistency. More impressively, compared with SAM 3D [6], which was trained on a large-scale dataset with SAM-based priors and substantial policy tuning and computational resources, CoLa3D shows better results using less than $1\%$ of SAM 3D's training data, demonstrating the effectiveness of our latent decomposition paradigm. We hope this work will inspire future research on composable 3D scene generation.

Recent promptable 3D segmenters like Point-SAM [91], Locate 3D [38], and Particulate [25] also build on query-based architectures, but they operate on raw 3D observations. In contrast, CoLa3D operates on a compact generative 3D latent. This allows CoLa3D to extract object-level meshes, rather than segmentation masks, from encoded meshes or single-image generated scene latents.

To summarize, we make the following contributions: (1) We introduce a novel paradigm, CoLa3D, for composable 3D scene decomposition: it obtains object latents from one whole-scene latent in a shared scene coordinate frame and decodes them in parallel into per-object meshes. (2) We design a lightweight query-based decomposition transformer that performs latent-space object extraction in a single forward pass, using 2D mask, 3D point, and learnable queries. (3) We show that CoLa3D achieves state-of-the-art performance on both synthetic and real-world datasets, outperforming strong baselines such as MIDI [16], SceneGen [39], PartCrafter [30], and SAM 3D [6] in both object quality and global scene consistency.

## 2 Related Work

**3D Reconstruction and Generation.** Recovering 3D geometry from images is a long-standing problem in computer vision. Classical reconstruction pipelines such as COLMAP [49, 50] chain multiple stages—structure-from-motion and multi-view stereo—to estimate camera poses and dense geometry, whereas recent feed-forward models [71, 7, 72, 65, 69, 29, 66] unify these stages into an end-to-end network. Beyond reconstruction, generative 3D models synthesize assets from learned priors with point, voxel, implicit, or Gaussian representations [42, 48, 44, 40, 55, 19], and recent foundation 3D generators [86, 75, 76, 59–61, 27] further produce high-quality assets in compact 3D latent spaces [84]. However, these holistic generators do not expose editable instance-level structure. Part-aware 3D generation [33, 5, 13, 79, 58] addresses object decomposition but focuses on single objects rather than scene-level instances. We instead target *scene*-level 3D instance decomposition.

**Scene Decomposition.** Scene decomposition aims to obtain a composable 3D scene made of individual object instances. Traditional 3D decomposition performs instance segmentation on raw 3D observations [51, 21, 91, 38, 25], but they are sensitive to noisy observations and output masks instead of complete object meshes. Recently, generative scene decomposition instead targets object-level geometry together with a coherent global layout. Staged pipelines first segment scene components and then retrieve, reconstruct, or optimize them separately [22, 14, 12, 90, 77, 73, 74, 41, 92, 1, 15, 80, 89]. They are modular but often depend on external segmentation, per-object generation, and layout refinement, which can lead to inconsistent pose, scale, and contact. Joint models couple object geometry, pose, and instance assignment during generation [10, 16, 28, 67, 39, 6, 82, 31], yet objects are still generated in separate latents and need extra pose and scale estimation. PartCrafter [30] jointly denoises per-object latent in a global frame, but object poses are still implicitly modeled through attention. In contrast, our method decomposes a single whole-scene latent into globally aligned object latents in a feed-forward manner, avoiding separate object generation and per-scene optimization. Inter-object spatial relations are thus naturally encoded in the shared latent.

**Query-Based Models.** Query-based models predict object sets through learnable queries and have become a standard formulation for detection and segmentation [3, 93, 85, 8, 24]. Recent promptable 3D models also use points or masks as queries for 3D segmentation [91, 37, 94]. Related object-centric models also learn latent slots to represent scene entities [36, 53, 54]. Our method uses queries as latent extractors: each query interacts with a compact foundation-model scene latent and outputs an object latent that can be decoded into a mesh within the shared scene coordinate frame.

## 3 Method

Given a single image $\boldsymbol{I}^s$ or an existing 3D scene mesh $\mathcal{M}^s$ from AI-generated, scanned, or artist-created sources, our goal is to obtain a composable 3D scene as a set of object meshes $\{\hat{\mathcal{M}}_k^o\}_{k=1}^K$ with their correct spatial layout, where $K$ is the number of objects in the scene. Here, we introduce CoLa3D, a feed-forward neural network that predicts a set of object latents from a single whole-scene latent in a *global latent space*, which are then decoded in parallel to produce per-object meshes. The designed query-based decomposition transformer is efficient, flexible, and scalable. It accepts various types of queries and operates directly on the compact 3D scene latent (as illustrated in Figure 2), instead of segmenting the scene in 2D and lifting each instance separately as in prior methods [16, 39, 6], or performing segmentation on raw 3D observations [51, 21, 91, 38, 25].

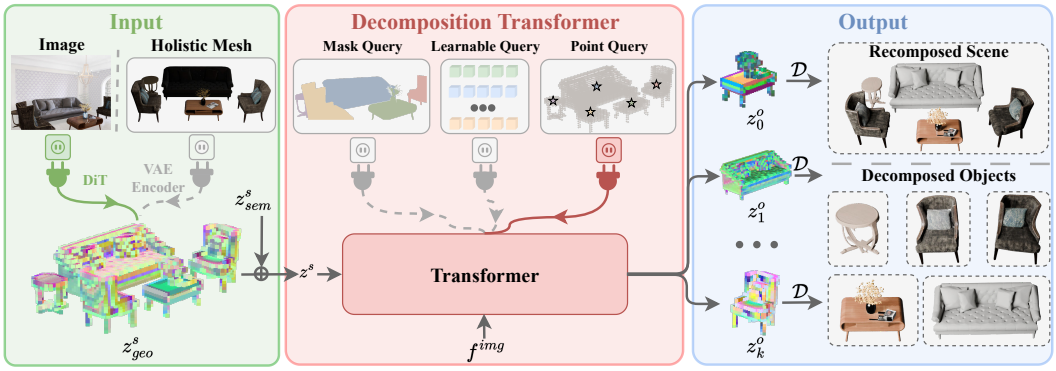


Figure 2: **Overview of CoLa3D.** Starting from an RGB image or a holistic scene mesh, CoLa3D first obtains a structured scene latent $\boldsymbol{z}^s_{\text{geo}}$ with TRELLIS.2 [76] and augments it with 3D semantic features $\boldsymbol{z}^s_{\text{sem}}$ from Utonia [87]. A lightweight decomposition transformer then uses mask, learnable, or point queries to decompose $\boldsymbol{z}^s = \boldsymbol{z}^s_{\text{geo}} + \boldsymbol{z}^s_{\text{sem}}$ through $L$ blocks of self-attention, cross-attention to frozen DINOv3 [52] features $\boldsymbol{f}^{\text{img}}$, and alternating Latent-to-Query/Query-to-Latent cross-attention. Finally, a multi-hot assignment extracts per-instance latents $\{\boldsymbol{z}^o_k\}$, which are decoded in parallel by the shared decoder $\mathcal{D}$ into aligned object meshes and can be recomposed in the global 3D space.

### 3.1 Preliminaries

CoLa3D is built upon TRELLIS.2 [76], a state-of-the-art image-to-3D generator with a compact structured 3D latent space. The proposed latent decomposition paradigm is not specific to TRELLIS.2 and should be applicable to other foundational 3D generators, such as Hunyuan3D [59, 60].

Given a 3D mesh $\mathcal{M}$, it is first represented by O-Voxel $\mathcal{O}(\mathcal{M}) \mapsto \{\boldsymbol{f}_i, \boldsymbol{p}_i\}_{i=1}^{L_{\text{vox}}}$ in a regular 3D grid $\Omega \in \{0, 1, \cdots, N-1\}^3$, where $\boldsymbol{f}_i$ is the feature of the $i$-th active cell and $\boldsymbol{p}_i$ is its 3D coordinate, and $L_{\text{vox}}$ is the number of active cells. Then, the O-Voxel is encoded into a compact latent $\boldsymbol{z} = \mathcal{E}(\mathcal{O}(\mathcal{M}))$ by a sparse-compression encoder, which can be decoded back to a mesh $\hat{\mathcal{M}} = \mathcal{D}(\boldsymbol{z})$ by a corresponding sparse-convolution decoder. Here, $\boldsymbol{z} = \{\hat{\boldsymbol{f}}, \hat{\boldsymbol{p}}\}_{i=1}^{L_{\text{lat}}}$, where $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{p}}$ are the updated feature and coordinate of the $i$-th active cell in the latent space, and $L_{\text{lat}} \ll L_{\text{vox}}$ is the active cell number in the latent space, achieving a highly compact representation of the original mesh. During image-to-3D inference, TRELLIS.2 uses image-conditioned flow models [32, 45] to predict the sparse structure and then generate geometry latents.

We accordingly obtain the scene geometry latent $\boldsymbol{z}^s_{\text{geo}}$ either by encoding $\mathcal{M}^s$ with the O-Voxel encoder, or by generating it from $\boldsymbol{I}^s$ with the image-conditioned flow models. However, TRELLIS.2 is originally designed for object-level generation, and thus we need to adapt it for scene-level generation and decomposition. Since scene-level fine-tuning is not our main contribution, we describe the details in section A.4 and focus on the novel latent decomposition module in the below.

### 3.2 Query-Based 3D Latent Decomposition Transformer

The input is a composite whole-scene latent $\boldsymbol{z}^s = \boldsymbol{z}^s_{\text{geo}} + \boldsymbol{z}^s_{\text{sem}}$, where the geometry component $\boldsymbol{z}^s_{\text{geo}}$ is produced by a foundational 3D generator TRELLIS.2 [76], and the semantic component $\boldsymbol{z}^s_{\text{sem}}$ by a 3D semantic foundation model Utonia [87]; see *Feature Enhancement* below for details. CoLa3D's latent transformer maps this scene latent to a set of object latents:

$$\mathcal{Z} = f_\theta(\boldsymbol{z}^s, \mathcal{Q}), \tag{1}$$

where $\mathcal{Z} = \{\boldsymbol{z}^o_k\}_{k=1}^K$, $\boldsymbol{z}^o_k$ is the latent of the $k$-th object, which can be decoded into an object mesh $\hat{\mathcal{M}}^o_k$ with a shared decoder, and $\mathcal{Q}$ is a set of queries that can be 2D masks, 3D points, or purely learnable queries. Unlike recent query-based 3D segmentation methods [91, 25, 38] that operate on raw point clouds or self-learned 3D features, our input $\boldsymbol{z}^s$ *carries richer geometric and semantic priors while remaining far more compact than raw 3D observations.*

**Query Encoder.** One key ingredient is the set of multi-modal queries $\mathcal{Q}$, which specify objects to be extracted from the scene latent $\boldsymbol{z}^s$. In this work, we consider three query types: 2D masks ($\mathcal{Q}^{\mathrm{M}}$), 3D points ($\mathcal{Q}^{\mathrm{P}}$), and learnable queries ($\mathcal{Q}^{\mathrm{L}}$). Each query produces a $d$-dimensional token $\boldsymbol{q}_k$ via a shared two-step recipe (encode $\rightarrow$ MLP); only the encode step is type-specific. For brevity, $K$ denotes the active query count, which equals $K_{\max}$ in the learnable mode.

For **mask prompts** $\{\mathbf{M}_k\}_{k=1}^K$, where $\mathbf{M}_k \in \{0,1\}^{H\times W}$ is the binary image mask of object $k$, we first use a lightweight CNN with global average pooling to extract a feature vector $\boldsymbol{f}_k^{\text{mask}} \in \mathbb{R}^d$. To disambiguate masks of similar shape, we additionally encode the spatial location of each mask, by applying sinusoidal positional encoding [64] to the center location of the mask, and map the result to a vector $\boldsymbol{f}_k^{\text{pos}} \in \mathbb{R}^d$ via an MLP. The final query token is then given by $\boldsymbol{q}_k = \mathrm{MLP}(\mathrm{LayerNorm}(\boldsymbol{f}_k^{\text{mask}} + \boldsymbol{f}_k^{\text{pos}}))$, yielding the mask query set $\mathcal{Q}^{\mathrm{M}} = \{\boldsymbol{q}_k\}_{k=1}^K$.

**Point prompts** $\{\mathbf{P}_k\}_{k=1}^K$, where $\mathbf{P}_k \in \mathbb{R}^3$ is the 3D coordinate of a prompted active latent cell, are encoded with positional encoding and projected to $\boldsymbol{f}_k^{\text{point}} \in \mathbb{R}^d$ with an MLP. We also augment it with the corresponding latent features at the prompted point, which is then mapped to obtain $\boldsymbol{f}_k^{\text{feat}} \in \mathbb{R}^d$ using another MLP. The query token $\boldsymbol{q}_k$ is finally obtained by adding them together, yielding the point query set $\mathcal{Q}^{\mathrm{P}} = \{\boldsymbol{q}_k\}_{k=1}^K$.

We also consider **learnable queries**, without using any explicit priors. Since the number of objects in the scene is unknown in advance, we maintain a learnable query set $\mathcal{Q}^{\mathrm{L}} = \{\boldsymbol{q}_k\}_{k=1}^{K_{\max}}$, with $K_{\max}$ set larger than the typical object count. Each slot $k$ is assigned a normalized index $k/(K_{\max}-1) \in [0,1]$, encoded with sinusoidal positional encoding, and mapped to $\boldsymbol{q}_k$ through a 2-layer MLP.

**Feature Enhancement.** To improve the generalization to novel categories and scenes, we augment the geometry latent $\boldsymbol{z}_{\text{geo}}^s$ with 3D semantic features $\boldsymbol{z}_{\text{sem}}^s$ obtained from Utonia [87]. Specifically, we use the input mesh when $\mathcal{M}^s$ is provided; for image input, we first decode the generated scene latent into a proxy mesh. The mesh is then mapped to an O-Voxel representation $\{\boldsymbol{f}_i, \boldsymbol{p}_i\}_{i=1}^{L_{\text{vox}}}$, in a regular 3D grid $\Omega \in \{0,1,\cdots,N-1\}^3$. Then, we feed the $\{\boldsymbol{p}_i\}_{i=1}^{L_{\text{vox}}}$, where $\boldsymbol{p}_i \in \mathbb{R}^3$ is the 3D position of the $i$-th active cell, into Utonia to obtain the corresponding semantic features $\{\boldsymbol{f}_i^{\text{sem}}\}_{i=1}^{L_{\text{lat}}}$. To align the dimensions of the geometry and semantic features, we apply an MLP to map $\{\boldsymbol{f}_i^{\text{sem}}\}_{i=1}^{L_{\text{lat}}}$ to $\boldsymbol{z}_{\text{sem}}^s = \{\hat{\boldsymbol{f}}_i^{\text{sem}}\}_{i=1}^{L_{\text{lat}}}$. After this operation, we obtain scale and dimensionality aligned geometry features $\boldsymbol{z}_{\text{geo}}^s \in \mathbb{R}^{L_{\text{lat}}\times d}$ and semantic features $\boldsymbol{z}_{\text{sem}}^s \in \mathbb{R}^{L_{\text{lat}}\times d}$, where $L_{\text{lat}}$ is the number of latent active cells. The enhanced scene latent is then obtained by summing them together $\boldsymbol{z}^s = \boldsymbol{z}_{\text{geo}}^s + \boldsymbol{z}_{\text{sem}}^s$.

**Attention Blocks.** Following recent query-based segmentors [20, 25, 38, 26], we apply a stack of attention blocks to let queries interact with the scene latent and specialize into instance-level representations. In each block, we first perform self-attention over query tokens $\{\boldsymbol{q}_k\}_{k=1}^K$. Then, updated query tokens $\{\hat{\boldsymbol{q}}_k\}_{k=1}^K$ and latent tokens $\{\boldsymbol{z}_i^s\}_{i=1}^{L_{\text{lat}}}$ cross-attend to the DINOv3 [52] features $\boldsymbol{f}^{\text{img}}$, grounding the 3D representations with 2D semantic cues. Finally, alternating Latent-to-Query and Query-to-Latent cross-attention are performed to flow information between them. We apply 3D RoPE [57, 68] to the query self-attention and bidirectional query–latent cross-attention layers, preserving relative 3D positional cues. After several blocks, the query and latent tokens are refined into instance-aware representations. Note that, the number of queries $K$ is significantly smaller than the number of latent cells $L_{\text{lat}}$ and DINOv3 features $h \times w$, so the query self-attention and other cross-attention operations are efficient and scalable to complex scenes with many objects and high-resolution latents. See section A.1 for block diagram and positional encoding details.

**Instance Prediction.** After obtaining the refined query and latent tokens $\{\hat{\boldsymbol{q}}_k\}_{k=1}^K$ and $\{\hat{\boldsymbol{z}}_i^s\}_{i=1}^{L_{\text{lat}}}$, we predict a soft assignment matrix $\mathbf{S} \in [0,1]^{L_{\text{lat}}\times K}$, whose $(i,k)$-entry $s_{i,k}$ is the probability of latent cell $i$ belonging to object query $k$. Specifically, we first project tokens into a common feature space with a shared lightweight MLP, and then compute $s_{i,k}$ by a temperature-scaled cosine similarity passed through a sigmoid:

$$s_{i,k} = \mathrm{Sigmoid}\left(\tau \cdot \frac{(\boldsymbol{z}_i^s)^\top \hat{\boldsymbol{q}}_k}{\|\boldsymbol{z}_i^s\|_2 \, \|\hat{\boldsymbol{q}}_k\|_2}\right), \tag{2}$$

where $\tau$ is a learnable temperature. For a discrete mask, we threshold $s_{i,k}$ at $0.5$ into a binary assignment $m_{i,k} \in \{0,1\}$, with an argmax fallback for cells left unclaimed. This design allows a latent

cell to be assigned to multiple queries since the latent grid is compact and coarse, where neighboring objects may share cells at contact regions. More inference details are provided in section A.6.

### 3.3 Training

**Training Losses.** We train CoLa3D's latent transformer with a loss: $\mathcal{L} = \mathcal{L}_{\text{mask}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}$, where $\mathcal{L}_{\text{mask}}$ is a mask loss that supervises the predicted masks, and $\mathcal{L}_{\text{reg}}$ is a regularization term that encourages query diversity to prevent query collapse.

Differently from [38, 25], our mask is supervised in the latent space with a multi-hot assignment (data processing is described in section A.2). To calculate the mask loss, we need to match the predicted labels to the ground-truth objects. For mask and point prompts, we directly use the prompt-object correspondence to assign labels. For learnable queries, following [3, 25], we apply Hungarian matching with a mask-to-mask cost, since the queries are unordered. In both cases we obtain an assignment $\pi(k)$ that maps each ground-truth object $k$ to its corresponding query index. Then, the mask loss averages the BCE and Dice terms over all matched objects:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \Big[ \lambda_{\text{bce}} \, \text{BCE}\big(\mathbf{S}_{:,\pi(k)}, \, \boldsymbol{y}_k^*\big) + \lambda_{\text{dice}} \, \text{Dice}\big(\mathbf{S}_{:,\pi(k)}, \, \boldsymbol{y}_k^*\big) \Big], \tag{3}$$

where $\mathbf{S}_{:,\pi(k)} \in [0,1]^{L_{\text{lat}}}$ is the column of $\mathbf{S}$ for the query matched to object $k$, $\boldsymbol{y}_k^* \in \{0,1\}^{L_{\text{lat}}}$ is the multi-hot ground-truth mask of object $k$ over the $L_{\text{lat}}$ active latent cells, $\mathcal{K} = \{k : \boldsymbol{y}_k^* \neq \mathbf{0}\}$ collects the objects with non-empty masks, and $\lambda_{\text{bce}}$, $\lambda_{\text{dice}}$ balance the two terms. The BCE term provides dense cell-wise supervision, while the Dice term (formulated in section A.4) normalizes each matched object mask and reduces the bias of cell-wise BCE toward large objects.

To prevent *query collapse*, where multiple queries specialize to the same object and leave others unused, we apply a regularizer $\mathcal{L}_{\text{reg}}$ penalizing pairwise cosine similarity among refined queries. Denoting by $\tilde{\boldsymbol{q}}_k = \hat{\boldsymbol{q}}_k / \|\hat{\boldsymbol{q}}_k\|_2$ the $\ell_2$-normalized query tokens, the regularizer is defined as

$$\mathcal{L}_{\text{reg}} = \frac{1}{K_{\max}(K_{\max}-1)} \sum_{i=1}^{K_{\max}} \sum_{j=1; j \neq i}^{K_{\max}} \big| \tilde{\boldsymbol{q}}_i^\top \tilde{\boldsymbol{q}}_j \big|. \tag{4}$$

This spreads queries across the feature space and encourages each to attach to a distinct instance, which is especially important for learnable queries that have no explicit disambiguating priors.

**Implementation Details.** By default, CoLa3D's latent transformer uses $K_{\max}$=8 for learnable queries, and employs $L$=6 attention blocks with a hidden dimension of $d$=768. The model was trained using 3D-FRONT dataset [11], where we combine the training splits prepared by MIDI [16] and InstPIFU [34], with a total of around 13k scenes and 82k images. We train the latent transformer for 60k steps with the Adam optimizer, an initial learning rate of $1 \times 10^{-4}$, and a per-GPU batch size of 16. The loss weights are set to $\lambda_{\text{bce}}$=1.0, $\lambda_{\text{dice}}$=1.0, and $\lambda_{\text{reg}}$=0.1. The training takes around 2 days on 4 NVIDIA H100 GPUs. We also finetune the TRELLIS.2 geometry components on the same training set to adapt it for scene-level generation. Since this is not the main contribution of this work, we detail the finetuning process in section A.4.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We assess our model on both synthetic and real-world datasets. For *synthetic* data, we use 3D-FRONT [11] and MetaScenes [83], which provide complete per-object meshes and layouts. For 3D-FRONT, we filter the InstPIFU [34] test split and retain 85 high-quality scenes for evaluation; for MetaScenes, we analogously select 50 high-quality scenes. Each synthetic scene is evaluated from one input image. Furthermore, to further assess the cross-dataset generalization ability, we directly evaluate all models on *real-world* datasets, including Replica [56] and ScanNet++ [81]. For real-world evaluation, we use more challenging scan-derived scenes with reliable ground truth: 8 scenes / 14 images from Replica and 45 scenes / 56 images from ScanNet++.

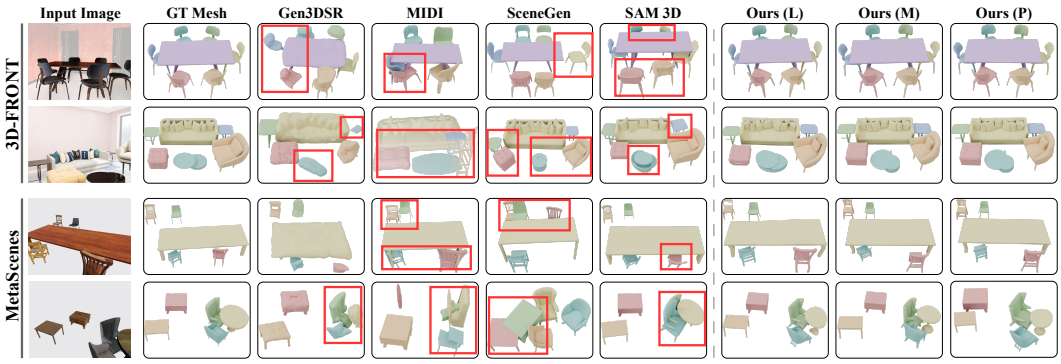


Figure 3: **Qualitative comparisons on synthetic data.** Compared with baselines, CoLa3D produces spatially coherent object meshes with correct relative positions and scales.

Table 1: **Comparisons on synthetic datasets. Prior** column: 🔷 denotes a 2D instance mask, 🔴 a 3D point prompt, and – no prior. We highlight the **Best**, Second-Best, and *Third-Best* results.

| Method | Prior | 3D-FRONT [11] | | | | | | MetaScenes [83] | | | | | | Avg. Time↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Scene | | | Object | | | Scene | | | Object | | | |
| | | CD↓ | F-Score↑ | IoU↑ | CD↓ | F-Score↑ | IoU↑ | CD↓ | F-Score↑ | IoU↑ | CD↓ | F-Score↑ | IoU↑ | |
| Gen3DSR [1] | 🔷 | 0.0230 | 0.7625 | 0.5804 | 0.0793 | 0.5211 | 0.4650 | 0.0140 | 0.854 | 0.654 | 0.0509 | 0.524 | 0.598 | 229.40 |
| PartCrafter [30] | – | 0.0468 | 0.6597 | 0.4470 | 0.1626 | 0.4549 | 0.3633 | 0.0460 | 0.6303 | 0.4315 | 0.1291 | 0.4654 | 0.4320 | 51.04 |
| MIDI [16] | 🔷 | 0.0289 | 0.7014 | 0.4151 | 0.0845 | 0.6512 | 0.5249 | 0.0429 | 0.664 | 0.433 | 0.0697 | 0.617 | 0.522 | 67.45 |
| 3D-Fixer [82] | 🔷 | 0.0194 | 0.7992 | 0.5455 | 0.0654 | 0.5792 | 0.4517 | 0.0212 | 0.7813 | 0.5581 | 0.0784 | 0.5140 | 0.4486 | 98.35 |
| Point-SAM [91] | 🔴 | 0.0170 | 0.8811 | 0.7025 | 0.1310 | 0.5892 | 0.4037 | 0.0175 | 0.901 | 0.699 | 0.1315 | 0.546 | 0.402 | 16.87 |
| SceneGen [39] | 🔷 | 0.0150 | 0.8434 | 0.6107 | 0.0348 | 0.7582 | 0.6684 | 0.0482 | 0.657 | 0.446 | 0.0593 | 0.630 | 0.543 | 49.30 |
| SAM 3D [6] | 🔷 | 0.0104 | 0.8889 | 0.6702 | 0.0420 | 0.7143 | 0.5569 | 0.0267 | 0.827 | 0.585 | 0.0794 | 0.612 | 0.440 | 32.15 |
| **CoLa3D (Free)** | – | **0.0075** | **0.9263** | **0.7474** | 0.0300 | **0.8252** | **0.6933** | 0.0069 | 0.9252 | 0.7566 | **0.0425** | 0.7782 | 0.6632 | *16.64* |
| **CoLa3D (Point)** | 🔴 | 0.0088 | 0.9256 | 0.7460 | **0.0273** | 0.8205 | *0.6908* | **0.0060** | **0.9291** | **0.7689** | *0.0510* | **0.7996** | **0.6982** | **16.60** |
| **CoLa3D (Mask)** | 🔷 | *0.0089* | *0.9166* | *0.7341* | *0.0307* | *0.8144* | 0.6927 | *0.0076* | *0.9206* | *0.7474* | 0.0582 | *0.7187* | *0.6189* | 16.63 |

**Metrics.** For quantitative results, we normalize predicted and ground-truth point clouds and align them with FilterReg before evaluation, following [16, 39]. Then, we sample 20,480 points per object mesh and report Chamfer Distance (CD), F-Score (threshold 0.1), and 3D bounding-box IoU at two levels. At the *scene level*, we assemble all predicted objects and compare the full scene with the ground truth; at the *object level*, we compute the same metrics and average over matched objects. We additionally report per-scene inference time on a single NVIDIA L40S 48 GB GPU, averaging the two datasets in each main table and providing per-dataset timings in Supp. table 5.

**Baselines.** We compare with six strong recent single-image scene generation and decomposition baselines: the per-scene *optimization* method Gen3DSR [1], and *feed-forward* methods including MIDI [16], SceneGen [39], PartCrafter [30], SAM 3D [6], and 3D-Fixer [82]. To further evaluate the merit of our query-based latent decomposition, we replace our query transformer with Point-SAM [91]: given the same point prompts as CoLa3D (Point), Point-SAM segments the latent cells, and each segmented latent-cell group is decoded by the same object decoder. For mask-based methods, we provide ground-truth masks as in prior works [16, 39, 6]; for point-based methods, we randomly sample one point from each ground-truth object. All baselines use their public checkpoints.

### 4.2 Assessing Composable Reconstruction on Synthetic Scenes

We report results on the 3D-FRONT InstPIFU split and MetaScenes in table 1. All three CoLa3D variants surpass the previous *per-object* generation baselines by a large margin at the scene level, while achieving strong object-level performance across all three query types. On 3D-FRONT, CoLa3D (Free) reduces scene CD by 27.9% and improves object F-Score by 8.8% over the best baseline. On MetaScenes, CoLa3D (Point) reduces scene CD by 57.1% and improves object F-Score by 26.9% over the best baseline. The averaged Time column shows that CoLa3D (Free) is also around 2× faster than the feed-forward baselines, due to its joint decomposition in a global latent space.

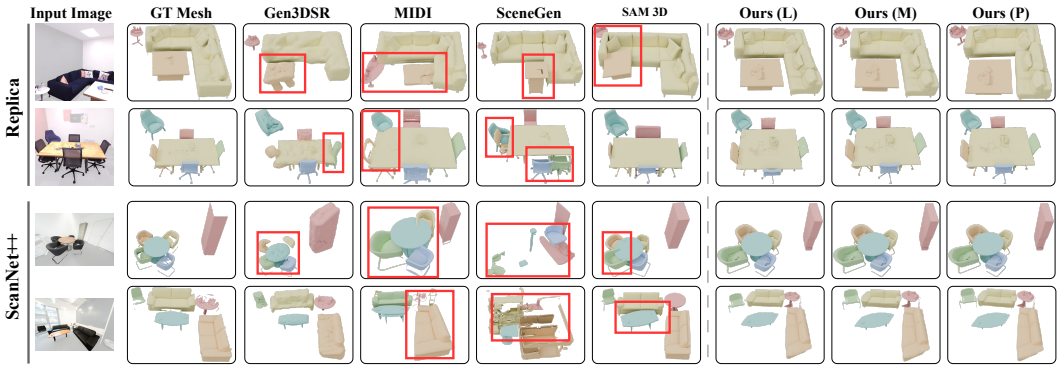


Figure 4: **Qualitative results on real-world scenes.** CoLa3D produces coherent decompositions on Replica and ScanNet++ images, demonstrating robust generalization from synthetic training data.

Table 2: **Zero-shot generalization on real-world scenes. Prior** column: 🔵 denotes a 2D instance mask, 🔴 a 3D point prompt, and – no prior. We highlight the **Best**, Second-Best, and *Third-Best* results. Here, CD is *one-side* Chamfer Distance (GT→Pred).

| Method | Prior | Replica | | | | | | ScanNet++ | | | | | | Avg. Time↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Scene | | | Object | | | Scene | | | Object | | | |
| | | CD↓ | Recall↑ | IoU↑ | CD↓ | Recall↑ | IoU↑ | CD↓ | Recall↑ | IoU↑ | CD↓ | Recall↑ | IoU↑ | |
| Gen3DSR [1] | 🔵 | 0.0070 | 0.837 | 0.612 | 0.0244 | 0.558 | 0.576 | 0.0096 | 0.8050 | *0.6444* | **0.0333** | 0.4949 | 0.4968 | 273.70 |
| PartCrafter [30] | – | 0.0297 | 0.5623 | 0.3643 | 0.1321 | 0.4070 | 0.3998 | 0.0388 | 0.5912 | 0.4448 | 0.1158 | 0.4482 | 0.3659 | 59.57 |
| MIDI [16] | 🔵 | 0.0098 | 0.800 | 0.455 | 0.0307 | 0.617 | 0.475 | 0.0197 | 0.6949 | 0.4832 | 0.0468 | 0.5740 | 0.4607 | 78.85 |
| SceneGen [39] | 🔵 | 0.0105 | 0.768 | 0.477 | *0.0289* | 0.608 | 0.540 | 0.0280 | 0.6594 | 0.4313 | *0.0417* | 0.5530 | 0.3489 | 61.10 |
| SAM 3D [6] | 🔵 | 0.0112 | 0.844 | 0.584 | 0.0356 | 0.599 | 0.491 | 0.0100 | 0.8086 | 0.6338 | 0.0450 | 0.6179 | 0.4225 | 37.05 |
| 3D-Fixer [82] | 🔵 | 0.0069 | *0.8909* | *0.6156* | 0.0333 | 0.5773 | 0.4766 | 0.0207 | 0.7278 | 0.5439 | 0.0378 | 0.5471 | 0.4335 | 107.40 |
| **CoLa3D (Free)** | – | *0.0051* | 0.8898 | 0.5949 | 0.0384 | *0.6575* | *0.5821* | *0.0082* | *0.8310* | 0.6350 | 0.0450 | **0.6434** | **0.4980** | 21.14 |
| **CoLa3D (Point)** | 🔴 | **0.0032** | **0.9371** | 0.6504 | **0.0235** | **0.7457** | **0.6226** | 0.0078 | 0.8423 | **0.6635** | 0.0470 | *0.6278* | *0.4892* | *21.47* |
| **CoLa3D (Mask)** | 🔵 | 0.0034 | 0.9314 | **0.6519** | 0.0340 | 0.6920 | 0.5904 | **0.0076** | **0.8506** | 0.6574 | 0.0456 | 0.6419 | 0.4846 | **20.83** |

The qualitative results in fig. 3 further illustrate the advantages of our latent decomposition paradigm. MIDI and Gen3DSR often generate wrong shapes (first, second, and last rows) when the object is heavily occluded, while SceneGen and SAM 3D produce more accurate shapes but suffer from spatial inconsistency and interpenetration (third and last rows), especially when the two objects are in close contact (last row). CoLa3D shows no such artifacts and produces more coherent layouts and plausible object scales, even under heavy occlusion, suggesting that its joint decomposition in a shared 3D latent space leads to more physically plausible and spatially consistent reconstructions.

### 4.3 Assessing Cross-Dataset Generalization on Real-World Scenes

To further evaluate the generalization ability of CoLa3D, we report results on the real-world Replica and ScanNet++ datasets in table 2. Since real datasets provide only partial object meshes, we report the *one-side* CD (GT→Pred) and Recall in place of bidirectional CD and F-Score. Table 2 shows that CoLa3D achieves the best scene-level results on real-world scans and strong object-level performance. On Replica, CoLa3D (Point) reduces scene CD by 53.6% and improves object Recall by 20.9% over the best baseline. On ScanNet++, CoLa3D (Mask) reduces scene CD by 20.8% and CoLa3D (Free) improves object Recall by 4.1% over the best baseline. CoLa3D (Mask) is the fastest according to the averaged Time column.

As evident in fig. 4, CoLa3D preserves coherent layouts and plausible object scales under zero-shot transfer, even under heavy occlusion. In contrast, Gen3DSR, MIDI, and SceneGen generate wrong shapes (first, third, and last rows). While SAM 3D is trained on 100× more data than ours and uses SAM-based priors, it still suffers from spatial inconsistency and interpenetration (second and last rows), suggesting that the per-object paradigm is not friendly to physical plausibility and spatial consistency. We present qualitative mesh-decomposition results in section B.2.

Table 3: **Comparisons on MetaScenes-hard. Prior** column: denotes a 2D instance mask, a 3D point prompt, and – no prior. We highlight the **Best**, Second-Best, and *Third-Best* results.

| Method | Prior | Scene | | | Object | | |
|---|---|---|---|---|---|---|---|
| | | CD↓ | F-Score↑ | IoU↑ | CD↓ | F-Score↑ | IoU↑ |
| MIDI [16] | | 0.0236 | 0.7235 | 0.4309 | 0.0766 | 0.5643 | 0.4816 |
| SceneGen [39] | | 0.0414 | 0.6449 | 0.4305 | 0.0651 | 0.5949 | *0.5249* |
| SAM 3D [6] | | 0.0112 | 0.8748 | 0.6339 | 0.0698 | 0.6090 | 0.4668 |
| 3D-Fixer [82] | | 0.0184 | 0.8058 | 0.5506 | 0.0720 | 0.5238 | 0.4611 |
| **CoLa3D (Free)** | – | **0.0093** | **0.8921** | **0.7229** | 0.0826 | *0.6242* | 0.5227 |
| **CoLa3D (Point)** | | *0.0106* | 0.8910 | *0.7112* | **0.0600** | **0.6858** | **0.5745** |
| **CoLa3D (Mask)** | | 0.0095 | *0.8870* | 0.7180 | *0.0653* | 0.6666 | 0.5525 |

### 4.4 Assessing Scalability to Denser Scenes

We further evaluate the scalability of CoLa3D to denser and more complex scenes by extending both the training data and query capacity. To this end, we augment our original training set with ARSG-110K [82], a large-scale scene dataset introduced by 3D-Fixer that contains over 110k scenes and 3M annotated images with object-level geometry and layout annotations. After data cleaning, we retain 100k ARSG scenes and 870k images, with at most 20 objects per scene. Using the combined training set, we retrain all three CoLa3D variants for 120k steps with $K_{\max}$=20, keeping the remaining training settings unchanged.

To assess reconstruction quality beyond the original 2–8-object evaluation setting, we construct *MetaScenes-hard*, a separate test set of 50 MetaScenes scenes containing 9–20 visible objects per scene. We retain the other filtering criteria used for the main benchmark to ensure reliable reference geometry. We report CD, F-Score, and bounding-box IoU at both the scene and object levels, measuring the fidelity of the assembled reconstruction and its constituent objects.

Table 3 shows that CoLa3D retains its scene-level advantage on MetaScenes-hard across all three query types. Every variant outperforms MIDI, SceneGen, SAM 3D, and 3D-Fixer on scene CD, F-Score, and IoU. CoLa3D (Free) achieves the best scene-level results without instance prompts, reducing CD by 17.0% over the strongest baseline SAM 3D. Its scene IoU also increases from 0.6339 to 0.7229 relative to the best baseline. At the object level, CoLa3D (Point) achieves the best CD, F-Score, and IoU. It reduces CD by 7.8% over SceneGen and improves F-Score by 12.6% over SAM 3D, the strongest respective baselines. These results indicate that CoLa3D maintains its scene-level performance advantage over the compared baselines in scenes with more objects.

### 4.5 Assessing Visible Depth Accuracy

An ideal composable real-to-simulation 3D generation and decomposition should not only produce high-quality object meshes, but also preserve the overall scene geometry and layout. To evaluate this aspect, we assess the visible-region depth accuracy of reconstructed meshes on 3D-FRONT. To this end, we render 512×512 depth maps from reconstructed meshes and compare them against ground-truth depth. In addition to the 3D reconstruction baselines, we include MoGe2 [70], a state-of-the-art monocular depth estimator, as a strong reference. We report scale-and-shift aligned absolute relative error (AbsRel), RMSE, and $\delta_1$ (the fraction of valid pixels with relative depth ratio below 1.25) on visible foreground pixels, together with completion, which measures the fraction of ground-truth foreground pixels covered by valid rendered depth.

Table 4 (a) shows that CoLa3D preserves strong visible-region geometry after latent decomposition. Among 3D reconstruction baselines, CoLa3D reduces AbsRel by 14.9% and RMSE by 10.3% over the best baseline, while also achieving the highest completion score. Compared with the MoGe2 reference, CoLa3D produces competitive visible-region depth while additionally yielding decomposable meshes. Qualitative depth comparisons are provided in fig. 9.

Table 4: **Additional results and ablations on 3D-FRONT. Left:** visible-region depth accuracy. MoGe2 is shown as a reference and excluded from ranking. **Right:** ablation study on the InstPIFU split. We highlight the **Best**, Second-Best, and *Third-Best* results among 3D methods.

**(a) Visible-region depth accuracy.**

| Method | Prior | AbsRel↓ | $\delta_1$ ↑ | RMSE↓ | Completion↑ |
|---|---|---|---|---|---|
| MoGe2 [70] | – | 0.0423 | 0.9862 | 0.1802 | 0.9888 |
| SAM 3D [6] | 🔷 | 0.0937 | *0.9245* | 0.3444 | 0.8352 |
| SceneGen [39] | 🔷 | 0.1282 | 0.8780 | 0.4234 | 0.8494 |
| Gen3DSR [1] | 🔷 | 0.1320 | 0.8636 | 0.4171 | 0.7619 |
| MIDI [16] | 🔷 | 0.1738 | 0.7778 | 0.5447 | 0.8152 |
| **CoLa3D (Mask)** | 🔷 | **0.0797** | 0.9352 | **0.3090** | *0.9175* |
| **CoLa3D (Free)** | – | 0.0844 | **0.9370** | 0.3243 | 0.9182 |
| **CoLa3D (Point)** | 🔴 | *0.0895* | 0.9244 | *0.3251* | **0.9337** |

**(b) Ablation study.**

| Model Variant | Scene | | | Object | | |
|---|---|---|---|---|---|---|
| | CD↓ | F-Score↑ | IoU↑ | CD↓ | F-Score↑ | IoU↑ |
| Baseline ($\mathcal{Q}_{\text{learnable}}$) | 0.0072 | **0.9300** | *0.7603* | 0.0318 | *0.8127* | 0.6912 |
| w/o DINOv3 features | 0.0077 | 0.9282 | 0.7552 | *0.0324* | 0.8024 | 0.6715 |
| w/o Utonia features | 0.0080 | 0.9268 | 0.7477 | 0.0492 | 0.7671 | 0.6478 |
| w/o diversity loss | *0.0073* | *0.9286* | 0.7551 | 0.0353 | 0.8031 | 0.6695 |
| w/o multi-hot | 0.0078 | 0.9214 | 0.7469 | 0.0394 | 0.8015 | 0.6568 |
| w/ $\mathcal{Q}_{\text{mask}}$ | **0.0067** | 0.9289 | 0.7626 | 0.0336 | 0.8151 | *0.6880* |
| w/ $\mathcal{Q}_{\text{point}}$ | 0.0075 | 0.9274 | **0.7655** | **0.0270** | **0.8216** | **0.7051** |

### 4.6 Ablation Study

We ablate CoLa3D along two axes: decomposition design choices and object-query type. All variants are trained solely on the 3D-FRONT InstPIFU train split using a single H100 GPU, with all other training hyperparameters held identical. The learnable-query variant serves as the prompt-free baseline, and each row in table 4 (b) changes one component or query modality.

**Importance of Feature Augmentation** Feature augmentation is important because latent decomposition must distinguish object extent inside a shared scene representation, not only preserve the global scene shape. Removing DINOv3 features drops object IoU by $0.0197$, while removing Utonia increases object CD by $54.7\%$ over the learnable-query baseline.

**Ablations on Loss Design** The multi-hot mask supervision and query diversity loss are significant for improving decomposition quality. We investigate them by removing the diversity loss and replacing multi-hot supervision with single-hot labels. The former variant drops object IoU by $3.1\%$, while the latter reduces object IoU by $5.0\%$.

**Ablations on Query Type** Our query-based decomposition design is flexible to various query types. The mask variant achieves the best scene-level performance (scene CD reduced by $6.9\%$), while the point variant achieves the best object-level performance (object CD reduced by $15.1\%$ over learnable queries), suggesting that explicit prompts further improve decomposition quality.

## 5 Conclusions

We present CoLa3D, a novel paradigm for Composable Latent 3D decomposition. Our approach departs from the common recipe of segmenting objects in 2D and reconstructing each one separately, by performing decomposition directly in a *whole*-scene latent space. It achieves state-of-the-art results in both object quality and scene-level consistency across various datasets, in a significantly faster manner. Our simple, flexible, and scalable query-based transformer also supports various query types and latent sources. It also achieves visible-region depth accuracy on par with strong depth estimators such as MoGe2, which is another benefit over per-object generation methods.

**Limitations and future work.** The coarse latent grid can absorb thin or tightly clustered objects into one cell; a hierarchical latent is a natural remedy. Training data is also a bottleneck: available scene datasets are small in scale, limited in quality, and sparsely populated with objects. Scaling to larger, higher-quality, and more densely populated data is an important next step.


## Acknowledgments and Disclosure of Funding

We thank Ruijie Zhu for insightful discussions that helped refine this work, and Yingqi Wang for valuable assistance with the robotic demonstration. Yongcai Wang is supported in part by the National Natural Science Foundation of China under Grant No. 61972404, the Public Computing Cloud, Renmin University of China, and the Blockchain Lab, School of Information, Renmin University of China. Deying Li is supported in part by the National Natural Science Foundation of China under Grant No. 12071478. Chuanxia Zheng is supported by NTU SUG-NAP and National Research Foundation, Singapore, under its NRF Fellowship Award NRF-NRFF17-2025-0009.

## References


[1] Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision (3DV)*, 2025. URL https://arxiv.org/abs/2404.03421.

[2] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *CVPR*, June 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, et al. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r35clVtGzw.

[5] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. In *CVPR*, 2025.

[6] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *CVPR*, pages 7220–7232, 2026.

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, pages 370–386. Springer, 2024.

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, June 2022.

[9] Nikita Chernyadev, Nicholas Backshall, Xiao Ma, Yunfan Lu, Younggyo Seo, and Stephen James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. In *Conference on Robot Learning (CoRL)*, pages 4201–4217, 2024.

[10] Manuel Dahnert, Angela Dai, Norman Müller, and Matthias Nießner. Coherent 3d scene diffusion from a single rgb image. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2412.10294.

[11] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *ICCV*, pages 10933–10942, 2021.

[12] Daoyi Gao, Dávid Rozenberszki, Stefan Leutenegger, and Angela Dai. Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[13] Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. MeshArt: Generating articulated meshes with structure-guided transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 618–627, 2025.

[14] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4022–4031, 2022.

[15] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 25367–25377, 2025.

[16] 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. In *CVPR*, pages 23646–23657, 2025.

[17] Zhening Huang, Xiaoyang Wu, Fangcheng Zhong, Hengshuang Zhao, Matthias Nießner, and Joan Lasenby. Litereality: Graphics-ready 3d scene reconstruction from rgb-d scans. In *NeurIPS*, 2025.

[18] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, pages 287–318, 2022.

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):1–14, 2023.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[21] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL `https://arxiv.org/abs/2311.14405`.

[22] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *ECCV*, pages 260–277. Springer, 2020.

[23] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conf. Robot Learn.*, pages 80–93. PMLR, 2023.

[24] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050, June 2023.

[25] Ruining Li, Yuxin Yao, Chuanxia Zheng, Christian Rupprecht, Joan Lasenby, Shangzhe Wu, and Andrea Vedaldi. Particulate: Feed-forward 3d object articulation. In *CVPR*, 2026.

[26] Ruining Li, Yuxin Yao, Matt Zhou, Chuanxia Zheng, Christian Rupprecht, Joan Lasenby, Shangzhe Wu, and Andrea Vedaldi. Instruct-particulate: Scaling feed-forward 3d object articulation with kinematic control. In *NeurIPS*, 2026.

[27] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE TPAMI*, 2025.

[28] Zhiqi Li, Wenhuan Li, Tengfei Wang, Zhenwei Wang, Junta Wu, Haoyuan Wang, Yunhan Yang, Zehuan Huang, Yang Li, Peidong Liu, and Chunchao Guo. Moca: Mixture-of-components attention for scalable compositional 3d generation. *arXiv preprint arXiv:2512.07628*, 2025. URL `https://arxiv.org/abs/2512.07628`.

[29] 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. In *ICLR*, 2026.

[30] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. In *NeurIPS*, 2025.

[31] Lu Ling, Yunhao Ge, Yichen Sheng, and Aniket Bera. I-scene: 3d instance models are implicit generalizable spatial learners. *arXiv preprint arXiv:2512.13683*, 2025. URL `https://arxiv.org/abs/2512.13683`.

[32] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

[33] Anran Liu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Zhiyang Dou, Hao-Xiang Guo, Ping Luo, and Wenping Wang. Part123: part-aware 3d reconstruction from a single-view image. In *ACM SIGGRAPH*, pages 1–12, 2024.

[34] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision (ECCV)*, 2022.

[35] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *CVPR*, pages 21736–21746, 2023.

[36] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL `https://arxiv.org/abs/2006.15055`.

[37] Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zhuo Chen, Yanwen Guo, and Chunchao Guo. P 3-sam: Native 3d part segmentation. In *ECCV*, pages 195–211. Springer, 2026.

[38] Paul Mcvay, Sergio Arnaud, Ada Martin, Arjun Majumdar, Krishna Murthy Jatavallabhula, Phillip Thomas, Ruslan Partsey, Daniel Dugas, Abha Gejji, Alexander Sax, Vincent-Pierre Berges, Mikael Henaff, Ayush Jain, Ang Cao, Ishita Prasad, Mrinal Kalakrishnan, Michael Rabbat, Nicolas Ballas, Mido Assran, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. LOCATE 3D: Real-world object localization via self-supervised learning in 3D. In *ICML*, pages 43476–43502, 2025.

[39] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. In *Int. Conf. 3D Vision*. IEEE, 2026.

[40] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[41] Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. In *CVPR*, pages 6022–6033, 2025.

[42] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[43] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021.

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019.

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.

[46] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[47] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

[48] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, pages 4209–4219, 2024.

[49] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

[50] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pages 501–518, 2016. doi: 10.1007/978-3-319-46487-9_31.

[51] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, United Kingdom, May 29 - June 2, 2023*, pages 8216–8223. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160588. URL https://doi.org/10.1109/ICRA48891.2023.10160588.

[52] 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.

[53] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2110.11405.

[54] Krishnakant Singh, Simone Schaub-Meyer, and Stefan Roth. Glass: Guided latent slot diffusion for object-centric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2407.17929.

[55] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 34: 19313–19325, 2021.

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

[57] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024. doi: 10.1016/j.neucom.2023.127063.

[58] Jiaxiang Tang, Ruijie Lu, Zhaoshuo Li, Zekun Hao, Xuan Li, Fangyin Wei, Shuran Song, Gang Zeng, Ming-Yu Liu, and Tsung-Yi Lin. Efficient part-level 3d object generation via dual volume packing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=qbVbZWxUib.

[59] Tencent Hunyuan3D Team. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation, 2024.

[60] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

[61] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. URL https://arxiv.org/abs/2506.15442.

[62] Tencent Hunyuan3D Team. Hunyuan3d 3.1, 2026. URL https://3d.hunyuan.tencent.com/.

[63] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[65] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025.

[66] Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-omega. In *CVPR*, 2026.

[67] Ling Wang, Hao-Xiang Guo, Xinzhou Wang, Fuchun Sun, Kai Sun, Pengkun Liu, Hang Xiao, Zhong Wang, Guangyuan Fu, Eric Li, Yang Liu, and Yikai Wang. Scenetransporter: Optimal transport-guided compositional latent diffusion for single-image structured 3d scene generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xjCkwPhQWq.

[68] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. doi: 10.48550/arXiv.2409.12191. URL https://arxiv.org/abs/2409.12191.

[69] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, June 2025.

[70] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. URL https://arxiv.org/abs/2507.02546.

[71] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024.

[72] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DTQIjngDta.

[73] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *ECCV*, pages 197–213. Springer, 2022.

[74] Tianhao Wu, Chuanxia Zheng, Qianyi Wu, and Tat-Jen Cham. Clusteringsdf: Self-organized neural implicit surfaces for 3d decomposition. In *ECCV*, pages 255–272. Springer, 2024.

[75] 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. In *CVPR*, June 2025.

[76] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. In *CVPR*, 2026.

[77] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *ICCV*, pages 13779–13788, October 2021.

[78] Yuanbo Yang, Yifei Yang, Hanlei Guo, Ruijie Xiong, Yue Wang, and Yiyi Liao. Urbangiraffe: Representing urban scenes as compositional generative neural feature fields. In *IEEE International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.00844.

[79] Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. In *ACM SIGGRAPH Asia Conference Papers*, 2025. URL https://arxiv.org/abs/2507.06165.

[80] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM TOG*, 44(4):1–19, 2025.

[81] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: a high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

[82] Ze-Xin Yin, Liu Liu, Xinjie Wang, Wei Sui, Zhizhong Su, Jian Yang, and Jin Xie. 3d-fixer: Coarse-to-fine in-place completion for 3d scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2604.04406.

[83] Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Song-Chun Zhu, Tengyu Liu, and Siyuan Huang. Metascenes: Towards automated replica creation for real-world 3d scans. In *CVPR*, 2025.

[84] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM TOG*, 42(4):1–16, 2023.

[85] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=3mRwyG5one.

[86] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM TOG*, 43(4):1–20, 2024.

[87] Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, Zehao Huang, Naiyan Wang, and Hengshuang Zhao. Utonia: Toward one encoder for all point clouds, 2026. URL https://arxiv.org/abs/2603.03283.

[88] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16259–16268, 2021.

[89] Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. Depr: Depth guided single-view scene reconstruction with instance-level diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[90] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, pages 15838–15847, October 2021.

[91] Yuchen Zhou, Jiayuan Gu, Tung Yen Chiang, Fanbo Xiang, and Hao Su. Point-SAM: Promptable 3d segmentation model for point clouds. In *ICLR*, 2025.

[92] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. ObjectGS: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8350–8360, 2025. doi: 10.1109/ICCV51701.2025.00782.

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

[94] Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. Partsam: A scalable promptable part segmentation model trained on native 3d data. In *ICLR*, volume 2026, pages 132469–132493, 2026.

# CoLa3D: Composable Latent 3D Decomposition

*Supplementary Material*

NeurIPS 2026 Submission

## A Additional Method Details

### A.1 Attention Block Details

This section expands the *Attention Blocks* component of section 3.2. As shown in fig. 5, each block maintains two information streams: object queries, which represent instance hypotheses, and 3D scene-latent tokens, which preserve the geometric support of the reconstructed scene. The block first applies self-attention over the query tokens so that different object hypotheses can exchange context before being grounded in visual evidence. The queries and latent tokens then both cross-attend to the frozen DINOv3 image features, using image features as keys and values. This step injects 2D semantic cues into both streams while keeping the 3D latent tokens anchored to the shared scene coordinate frame.

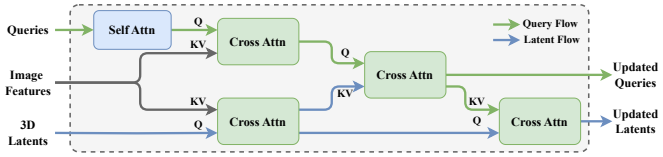


Figure 5: **Architecture details of the attention block used in the Query-Based 3D Latent Decomposition Transformer.**

After image-feature grounding, the block exchanges information between the two streams through bidirectional query–latent attention. The query stream first attends to the latent-conditioned evidence, allowing each query to refine the spatial extent of its object hypothesis. The latent stream then attends back to the updated queries, making each 3D latent token more instance-aware before mask prediction. Stacking these blocks gradually turns generic queries into object-specific tokens and enriches the scene latent with instance-level cues, which is the representation used by the final assignment head in section 3.2.

**Positional Encoding.** We use 3D rotary position embeddings (RoPE) [57] in the query self-attention and bidirectional query–latent cross-attention layers. For an attention query/key feature $\boldsymbol{x} \in \mathbb{R}^{D_{\text{head}}}$ with 3D position $(p_x, p_y, p_z)$, we partition its 2D feature pairs across the three spatial axes. Let $\boldsymbol{x}_\alpha$ denote the feature pairs assigned to axis $\alpha \in \{x, y, z\}$, and let $P_\alpha$ be the number of pairs assigned to that axis. For each axis, the $i$-th 2D feature pair in $\boldsymbol{x}_\alpha$ is rotated by

$$\theta_{i,p_\alpha} = \frac{p_\alpha}{\lambda^{i/P_\alpha}}, \quad i \in \{0, \dots, P_\alpha - 1\}, \tag{5}$$

where $\lambda$ is the RoPE frequency base. The rotation is applied independently for each axis,

$$\begin{pmatrix} x'_{\alpha,2i} \\ x'_{\alpha,2i+1} \end{pmatrix} = \begin{pmatrix} \cos\theta_{i,p_\alpha} & -\sin\theta_{i,p_\alpha} \\ \sin\theta_{i,p_\alpha} & \cos\theta_{i,p_\alpha} \end{pmatrix} \begin{pmatrix} x_{\alpha,2i} \\ x_{\alpha,2i+1} \end{pmatrix}. \tag{6}$$

The final head feature concatenates the axis-wise rotated pairs, allowing attention to encode relative displacement along all three spatial axes. In our implementation, the attention head dimension is $D_{\text{head}} = 64$, which gives $D_{\text{head}}/2 = 32$ complex feature pairs. We allocate 10 frequency pairs to each of the $x$, $y$, and $z$ axes and use identity-phase padding for the remaining two pairs.

The 3D latent tokens use their integer sparse-grid coordinates as RoPE positions. For query tokens, however, there is no native 3D coordinate because they are 1D object-hypothesis tokens. Following the M-RoPE design in Qwen2-VL [68], we lift the query index to a virtual diagonal 3D coordinate, *i.e.* query $k$ is assigned $(k, k, k)$, such as $(0, 0, 0)$, $(1, 1, 1)$, and $(2, 2, 2)$. This simple diagonal extension gives query–query and query–latent attention a consistent 3D positional frame, while the latent stream still uses the actual scene geometry coordinates.

### A.2 Training Data Processing

We derive multi-hot part supervision from the multi-object O-Voxel representation before encoding. Each object is represented by its active O-Voxel cells with a contiguous instance index, and cells from all objects are merged by high-resolution spatial coordinates instead of being forced into a single object label. When multiple objects share a coordinate, their instance labels are accumulated into an object-channel multi-hot vector, while geometric attributes are merged by averaging vertex offsets and OR-aggregating intersection indicators; single-object coordinates naturally reduce to one-hot labels. Because mask prediction is performed after the O-Voxel encoder, we project these full-resolution multi-hot labels to the latent sparse grid using the encoder downsampling ratio, which is $16\times$ for the $1024 \to 64$ grid compression in our implementation, and use channel-wise OR to aggregate labels within each latent cell. The resulting target allows one active latent token to supervise multiple object channels at contact or overlapping regions; the predicted per-query sigmoid masks are then matched to these object masks and trained with multi-label BCE and per-object Dice losses as described in section 3.3.

### A.3 Evaluation Data Processing

We construct evaluation sets from both synthetic datasets, 3D-FRONT [11] and MetaScenes [83], and real-world scan datasets, Replica [56] and ScanNet++ v2 [81]. For the main benchmarks, we retain scenes or views with 2–8 visible objects and exclude samples with severe object collisions, unreliable segmentations, or low-quality ground-truth meshes, so that the metrics reflect reconstruction quality rather than invalid evaluation geometry. For MetaScenes-hard, the separate 50-scene test set used for the scalability evaluation in section 4.4, we retain scenes with 9–20 visible objects while applying the other filtering criteria unchanged.

**3D-FRONT.** 3D-FRONT [11] is a large-scale synthetic indoor-scene dataset of professionally designed furnished rooms, with semantic layouts and textured 3D furniture models. Its complete object meshes and layouts make it suitable for evaluating scene-level geometry and object-level decomposition. We start from the 3D-FRONT test split provided by InstPIFU [34] and apply the common filtering protocol above, resulting in 85 high-quality scenes.

**MetaScenes.** MetaScenes [83] is a large-scale, simulatable 3D scene dataset built for automated replica creation from real-world indoor scans. It combines scan-derived layouts with clean object assets and physically plausible scene optimization, providing synthetic scenes with more realistic arrangements. We randomly sample camera views from the full scene set, render input images, filter the rendered samples with the same protocol as 3D-FRONT, and randomly select 50 high-quality scenes for evaluation.

**Replica.** Replica [56] contains photorealistic indoor scene reconstructions with dense geometry, high-resolution textures, and semantic/instance annotations. We use the processed and rendered Replica data released by DP-Recon [41]. After filtering for reliable scan geometry, segmentation quality, and the common object-count range, the Replica evaluation set contains 8 scenes and 14 images.

**ScanNet++ v2.** ScanNet++ [81] is a high-fidelity indoor-scene dataset with accurate scan geometry, high-resolution imagery, RGB-D captures, and semantic annotations. Its complex real-world scenes provide a more challenging setting for assessing model performance under realistic scan conditions. Applying the same real-scan filtering protocol yields 45 scenes and 56 images.

### A.4 Training Details

**TRELLIS.2 Scene-Level Finetuning.** The original TRELLIS.2 [76] is trained on a large-scale dataset of 3D assets, but with a focus on *single-object* generation. To adapt it for scene-level meshes $\mathcal{M}^s$, we finetune its three geometry components—the sparse-structure Flow Transformer, the O-Voxel Flow Transformer, and the O-Voxel decoder—on our curated *scene*-level datasets, following the same data processing and training losses as TRELLIS.2. We use the 1024-resolution SLAT DiT setting for this scene-level finetuning. We find that TRELLIS.2 can already produce promising scene

generation after finetuning on a relatively small-scale 3D scene dataset, demonstrating the strong geometry generalizability of the foundation model.

**Dice Loss Formulation.** The Dice term used in the mask loss $\mathcal{L}_{\text{mask}}$ is computed per matched query as

$$\text{Dice}\big(\mathbf{S}_{:,\pi(k)},\, \boldsymbol{y}_k^*\big) = 1 - \frac{2\,\langle \mathbf{S}_{:,\pi(k)},\, \boldsymbol{y}_k^* \rangle + \epsilon}{\|\mathbf{S}_{:,\pi(k)}\|_1 + \|\boldsymbol{y}_k^*\|_1 + \epsilon}, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, $\| \cdot \|_1$ the $\ell_1$ norm, and $\epsilon$ a small constant for numerical stability in degenerate zero-norm cases. Computing Dice *per query* (macro averaging) gives small and large objects equal weight, which is essential in our sparse latent space where a per-cell BCE alone is dominated by large objects that contain many more active latent cells.

### A.5 Inference Time Details

Table 5: **Detailed per-dataset inference time.** We report the average per-scene inference time in seconds.

| Method | Prior | 3D-FRONT [11] | MetaScenes [83] | Replica [56] | ScanNet++ [81] | Avg. |
|---|---|---|---|---|---|---|
| Gen3DSR [1] | 🔵 | 215.90 | 242.90 | 282.60 | 264.80 | 251.55 |
| PartCrafter [30] | – | 49.17 | 52.91 | 62.79 | 56.34 | 55.30 |
| MIDI [16] | 🔵 | 64.20 | 70.70 | 82.50 | 75.20 | 73.15 |
| SceneGen [39] | 🔵 | 48.60 | 50.00 | 59.70 | 62.50 | 55.20 |
| SAM 3D [6] | 🔵 | 30.90 | 33.40 | 39.10 | 35.00 | 34.60 |
| 3D-Fixer [82] | 🔵 | 93.20 | 103.50 | 120.90 | 93.90 | 102.88 |
| Point-SAM [91] | 🔴 | 20.18 | *13.55* | 23.91 | 19.22 | 19.22 |
| **CoLa3D (Free)** | – | 19.87 | 13.40 | 23.43 | 18.85 | 18.89 |
| **CoLa3D (Point)** | 🔴 | *20.12* | **13.08** | *23.76* | *19.17* | *19.03* |
| **CoLa3D (Mask)** | 🔵 | **19.55** | 13.71 | **23.19** | **18.47** | **18.73** |

table 5 provides the per-dataset timing details behind the averaged Time columns in the main quantitative tables. All timings are measured on one NVIDIA L40S 48 GB GPU with 8 vCPUs (AMD EPYC 7R13) and 64 GB RAM. We exclude texture-related inference and processing time from all timing reports.

### A.6 Inference Details

For Learnable queries, the initial multi-hot assignment may still contain redundant queries or spatially incoherent predictions, so we apply a lightweight two-step refinement at inference time. We first merge redundant queries that describe the same object via greedy non-maximum suppression, fusing any pair whose mask intersection-over-union exceeds $0.5$. We then extract the 6-connected components of each instance over the sparse latent grid and reassign small fragments to neighboring instances by majority vote, removing isolated noise while preserving compact small objects.

Using the refined discrete mask, we carve the per-instance latent out of the scene latent by gathering the cells assigned to each query,

$$\boldsymbol{z}_k^o = \{(\hat{\boldsymbol{f}}_i,\, \hat{\boldsymbol{p}}_i) \mid m_{i,k} = 1\}, \tag{8}$$

and decode each $\boldsymbol{z}_k^o$ in parallel with the shared TRELLIS.2 decoder $\mathcal{D}$ to obtain the per-object meshes $\{\hat{\mathcal{M}}_k^o\}_{k=1}^{\hat{K}}$, where $\hat{K}$ is the predicted number of objects. Since all instances are carved from a single scene latent in a shared coordinate system, the decoded meshes are naturally aligned and faithfully preserve the global spatial relationships of the original scene, without any post-hoc registration or layout regression as required by other methods [16, 39, 6].

## B Additional Experimental Results

### B.1 Geometric Interpenetration

We quantify object interpenetration on 3D-FRONT and MetaScenes, where complete ground-truth object meshes are available. We report two complementary metrics on the reconstructed scenes. *Collision rate* measures the percentage of object pairs with non-zero mesh penetration. *Normalized interpenetration volume* measures the pairwise intersection volume normalized by object volume, computed by voxelizing each scene on a $256^3$ grid.

Table 6: **Geometric interpenetration on synthetic datasets.** We report collision rate (CR) and normalized interpenetration volume (NIV), both in percent; lower is better. **Prior** column: denotes a 2D instance mask, a 3D point prompt, and – no prior. We highlight the **Best**, Second-Best, and *Third-Best* results.

| Method | Prior | 3D-FRONT | | MetaScenes | |
|---|---|---|---|---|---|
| | | CR↓ | NIV↓ | CR↓ | NIV↓ |
| Gen3DSR [1] | | 1.9 | *0.12* | **1.1** | **0.03** |
| MIDI [16] | | 8.8 | 0.34 | 5.9 | 0.32 |
| SceneGen [39] | | 5.1 | 0.35 | 17.6 | 1.11 |
| SAM 3D [6] | | 3.6 | *0.12* | 10.1 | 0.91 |
| PartCrafter [30] | – | 2.8 | 1.18 | *2.8* | 0.12 |
| **CoLa3D (Free)** | – | **1.8** | **0.06** | 2.3 | *0.13* |
| **CoLa3D (Point)** | | 3.4 | 0.19 | 4.5 | 0.19 |
| **CoLa3D (Mask)** | | *2.4* | 0.11 | 4.3 | 0.22 |

Table 6 shows that CoLa3D (Free) achieves the lowest collision rate and normalized interpenetration volume on 3D-FRONT, at 1.8% and 0.06%, respectively. On MetaScenes, it ranks second in collision rate at 2.3% and third in normalized interpenetration volume at 0.13%. All three CoLa3D variants achieve lower values on both metrics than MIDI and SceneGen across the two datasets.

### B.2 Decomposition from Existing Meshes

Beyond image input, the same query transformer also accepts encoded scene meshes. fig. 6 shows three mesh-decomposition pathways: two HunYuan-3D 3.1 [62] meshes generated from text and image prompts, respectively, and an existing holistic scene mesh. In each case, CoLa3D consumes the holistic mesh, renders a proxy image, and decomposes the encoded scene latent into aligned object meshes. The results show that the same mesh-to-mesh decomposition pipeline applies cleanly to generated and existing scene meshes.

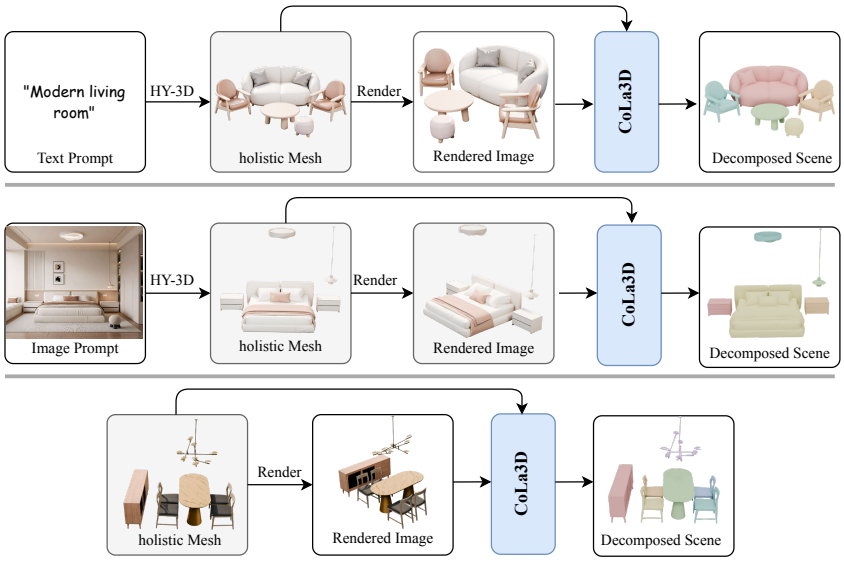


Figure 6: **Decomposition from holistic scene meshes.** CoLa3D produces coherent object-level meshes across generated and existing holistic scene meshes.

### B.3 Additional Synthetic Visualizations

We provide more qualitative comparisons on the 3D-FRONT InstPIFU split and MetaScenes, using the same layout as the main-paper synthetic visualization in fig. 3. These examples further show how CoLa3D preserves object alignment and relative scale across diverse synthetic indoor scenes.

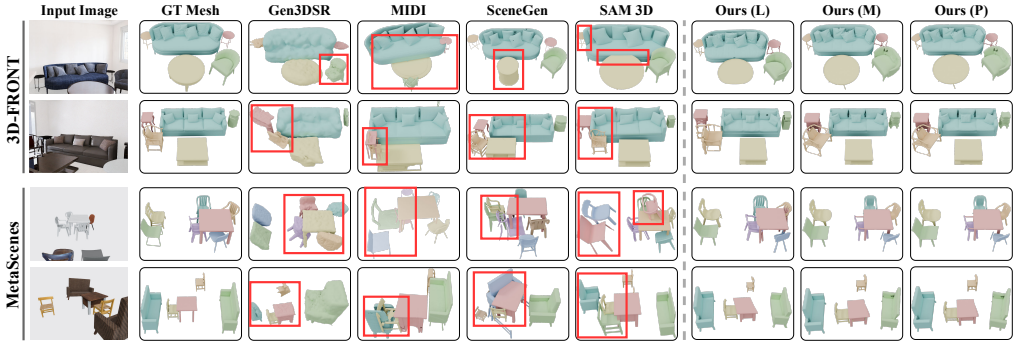


Figure 7: **Additional qualitative comparisons on synthetic scenes.** We show more results on the 3D-FRONT InstPIFU split and MetaScenes following the same layout as fig. 3.

### B.4 Additional Real-World Visualizations

We provide more zero-shot qualitative comparisons on Replica and ScanNet++, using the same layout as the main-paper real-world visualization in fig. 4. These examples illustrate the behavior of CoLa3D under real image observations with partial scans and cluttered layouts.

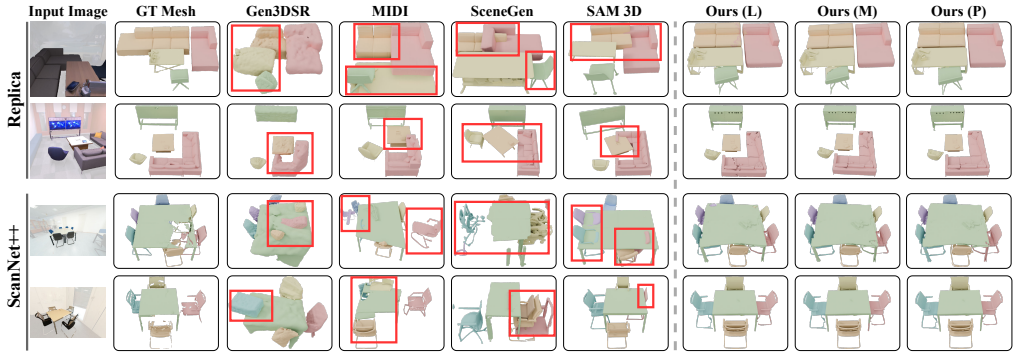


Figure 8: **Additional qualitative comparisons on real-world scenes.** We show more zero-shot results on Replica and ScanNet++ following the same layout as fig. 4.

### B.5 Qualitative Depth Comparison

We provide qualitative depth visualizations on 3D-FRONT to complement the quantitative results in table 4.

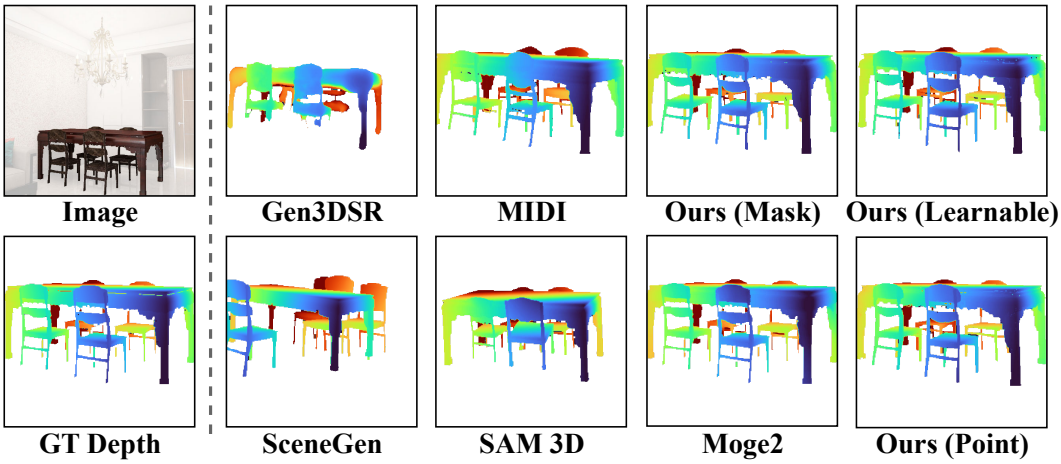


Figure 9: **Qualitative depth comparison on 3D-FRONT.** We visualize the input image, ground-truth depth, rendered depth maps from Gen3DSR, MIDI, SceneGen, SAM 3D, and CoLa3D with mask, learnable, and point queries, and the MoGe2 prediction as a monocular-depth reference. CoLa3D yields visible-region depth on par with MoGe2 while additionally producing decomposable per-object geometry.

## C Broader Impacts

CoLa3D turns a single image or a holistic scene mesh into an editable, object-level 3D scene in a single forward pass. On the positive side, this lowers the barrier to producing simulation-ready 3D environments and can benefit downstream applications such as robotics and embodied-AI training, physically plausible simulation, AR/VR experiences, digital twins, and 3D content creation for games, film, and design. By decomposing scenes directly in a shared latent space, our method also removes the need for per-object reconstruction and post-hoc layout regression, which makes such tools more accessible to researchers and practitioners with limited compute.

On the negative side, stronger composable 3D reconstruction tools may be misused to fabricate realistic but fake environments, or to reconstruct private interior spaces from casually captured images without consent, raising potential privacy, copyright, and disinformation concerns. The model also inherits biases from its training data: scene datasets are dominated by certain categories of indoor furniture and room layouts, so generation and decomposition quality may be uneven across cultures, building styles, and underrepresented object categories. We encourage downstream users to respect dataset licenses, obtain consent before reconstructing private spaces, and follow applicable privacy and safety requirements when integrating such 3D reconstruction tools into user-facing products.

## NeurIPS Paper Checklist

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 state the paper's task, method, scope, and empirical claims, and the experiments in section 4 evaluate the claimed synthetic and real-world settings.

   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 paper discusses limitations and future work in section 5, including failure modes from coarse latent grids and limited training data diversity.

   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 present theoretical results, theorems, or formal proofs.

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: The method, training losses, inference procedure, datasets, baselines, metrics, and evaluation protocol are described in sections A, 3 and 4.

   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: Code and pretrained checkpoints are publicly available through `https://github.com/Master-cai/CoLa3D`, together with instructions for setup, data preparation, and evaluation.

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: The experimental setup in section 4.1 specifies datasets, splits, baselines, prompts, metrics, and evaluation details, while sections A and 3.3 provide training and inference details.

   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: We report aggregate metrics on fixed benchmark splits, but the paper does not include error bars, confidence intervals, or statistical significance tests.

   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: The training compute is reported in section 3.3, and average per-scene inference times are reported in tables 1 and 2.

   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 authors have reviewed the NeurIPS Code of Ethics and believe the research conforms to it; the work uses public or properly licensed research assets and does not involve human subjects or private personal data.

   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 paper discusses potential positive uses, potential misuse risks, dataset bias, and responsible-use considerations of composable 3D scene generation in the supplementary material (section C).

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: The paper does not release high-risk scraped datasets or broadly deployable generative models; the released materials are intended to support research reproducibility.

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: Existing datasets, models, and baseline implementations are credited through citations in the main text, and we use them only for research purposes consistent with their licenses and terms of use.

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: [Yes]

    Justification: Code and pretrained checkpoints are released through `https://github.com/Master-cai/CoLa3D`, with accompanying documentation for setup and usage.

    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 research does not involve crowdsourcing experiments or 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 research does not involve crowdsourcing experiments or human subjects, so IRB approval or equivalent review is not applicable.

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 core method development does not use LLMs as an important, original, or non-standard methodological component.

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.