ROSE-3D: RENDER-FREE OBJECT-TO-SCENE EMBEDDINGS FOR COMPOSITIONAL 3D REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

3D scene representations are typically computed from the entire scene at a global level, rather than from reusable representations of the objects it contains. We argue that this constrains their capacity to express composition. A scene consists of objects placed in a shared frame, and the same objects recur across scenes; encoding each scene as a whole precludes reuse, making new arrangements of familiar objects as costly as unfamiliar ones. We treat composition as an operation in representation space: a learned operator maps object representations and their placements (translation, rotation, and scale) to a representation of the scene, without assembly or global encoding. To keep an object's content (shape and appearance) separable from its placement, we encode each 3D asset from its mesh and material maps without rendering, binding local surface orientation and appearance to continuous surface coordinates. We show that predicting a scene encoder's features can be satisfied without reading object content: the operator relies largely on layout and barely changes when objects are substituted. We therefore add two complementary objectives: identification, which requires the representation at each location to identify the object placed there, and language supervision, which aligns the composed representation with a description of the scene. With these objectives, the composed representation follows object content: it responds to substituted objects, including objects unseen during composer training, and after a replacement it shifts toward the new object. On question answering and situated next-step navigation, composed representations are competitive with point-map and point-cloud methods and outperform simpler representations built from the same inputs, while representing each new arrangement in milliseconds.

1 INTRODUCTION

Representing a 3D scene so it can be queried, grounded, and acted upon is a key challenge in 3D vision (Hong et al., 2023; Huang et al., 2023). Such a representation involves two choices: what the encoder consumes and what unit it represents. The first choice has been explored extensively, with point clouds (Wu et al., 2025; Wang et al., 2025), posed images, and, more recently, point maps (Mao et al., 2026), each of which observes the scene through projection or sampling. These methods balance fidelity to 3D structure with the ability to use 2D priors. By contrast, the second choice remains underexplored. Typically, the unit of analysis is the entire scene, encoded at a global level: the encoder processes the whole scene and produces a representation that downstream models, often large language models, operate on (Hong et al., 2023; Fu et al., 2024). We argue that it is the second choice, not the first, that determines whether a representation can express composition at all.

In this paper, we refer to a scene as a set of objects within a shared frame, and the same objects may recur across many scenes. Taking the scene as the unit of analysis discards this structure: every appearance of an object is encoded again, and a new arrangement of familiar objects costs as much as an entirely unfamiliar one. Some recent methods alter the unit of analysis. Chat-Scene (Huang et al., 2024) models a scene as a sequence of object-level embeddings, while object-compositional approaches in 3D reconstruction assemble scenes from per-object models (Yang et al., 2021; Wu et al., 2022). Both recover objects from an existing scene and compose them either within a language model's context or in rendering space, so neither yields a scene representation that can be produced

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

from its parts. What remains missing is an operator that maps representations of objects and their placements to the representation of the scene they form. We argue that the obstacle is not the design of such an operator, but the supervision available to train it.

Constructing such an operator means reconsidering both choices at once. Moving the unit of analysis to the object level requires a representation in which object content and placement are separable, so an object encoded once can be placed anywhere, at any orientation and scale, without re-encoding. Scene-level rendered views and point clouds lack this property, since each records the scene as observed rather than as assembled, entangling an object's appearance with its placement in the same values. We therefore encode each 3D asset directly from its mesh and material maps; no views are rendered when encoding assets or composing scenes. Local observations of surface orientation and appearance are bound to continuous surface coordinates in the asset's own frame via Spatial Semantic Pointers (SSPs; Komer et al., 2019), producing a single vector space in which surface coordinates and local attributes remain distinct and can be recombined. We then learn object representations over the resulting tokens through masked feature prediction.

We learn a composition operator over these representations, mapping a set of object representations and their placements to a fixed-size scene representation. We train this operator with a prediction objective to match the representation produced by the encoder for the assembled scene. Agreement alone, however, does not identify what a scene contains. This objective can be satisfied largely from the layout, without reading object content, and the operator converges to a shortcut solution (Geirhos et al., 2020; Wu et al., 2025): it captures the layout while remaining nearly unchanged when objects are replaced. To address this, we pair prediction with two complementary objectives: identification, which requires the representation at each location to identify the object placed there, and language supervision, which aligns the composed representation with a description of the scene and thus requires it as a whole to account for what the scene contains. With these objectives, the composed representation depends on what is present, not only on where objects are.

We validate our approach on two fronts. First, we assess composition directly. Controlled interventions show that replacing or exchanging objects changes the composed representation by a substantial fraction of the change in its inputs, whereas an operator trained with the prediction objective alone remains largely insensitive to them; this holds on held-out scenes, including those with objects absent from every training scene. After an object is replaced, the representation at its location shifts toward the new object, even with the layout unchanged. Second, we evaluate downstream utility on question answering and situated next-step navigation benchmarks built on ScanNet. Composed representations are competitive with point-map and point-cloud methods and outperform simpler representations built from the same objects, and a new arrangement of encoded objects requires a single 4.6 ms operator pass, compared with at least 87 ms for methods that re-encode the scene.

In summary, our contributions are:

1. We introduce a render-free tokenization that represents surface geometry and material appearance as Spatial Semantic Pointers, binding local attributes to continuous surface coordinates, and show that the projection of material descriptors into this space preserves their similarity structure.
2. We show that composition can be achieved in representation space: a learned operator maps reusable object representations and their placements to a scene representation, without assembly or global encoding.
3. We identify why matching a scene encoder is insufficient supervision: the objective can be satisfied largely from the layout. We show that two complementary objectives, identification and language supervision, make the composed representation follow object content, including after replacement.
4. We show that composed representations are competitive with point-map and point-cloud methods on 3D question answering and situated next-step navigation, while representing each new arrangement of encoded objects in milliseconds.

2 Related Work

Vector Symbolic Representations. Vector symbolic architectures represent objects, their attributes, and their relations within a single fixed-width vector while retaining the benefits of distributed rep-

resentations. Smolensky (1990) introduced tensor products to precisely bind roles to fillers, but this approach increases dimensionality through nested structures. Holographic Reduced Representations (Plate, 1995) compress binding into circular convolution, so that binding two D-dimensional vectors yields another D-dimensional vector. Binding and superposition form a closed algebra for constructing, querying, and transforming structures entirely in vector space (Kanerva, 2009; Kleyko et al., 2023), and, being differentiable, can serve as trainable components of neural networks (Ganesan et al., 2021). An open question for perceptual representation learning is whether this algebra remains effective over rich learned features rather than discrete symbols.

Spatial Semantic Pointers (SSPs; Komer et al., 2019) generalize this algebra to continuous domains. Fractional binding raises the Fourier phasors of a base vector to real-valued powers, so that continuous coordinates become vectors and spatial displacement corresponds to binding. This property has been used to represent spatial regions and query relations among entities (Lu et al., 2019) and to model continuous dynamical systems (Voelker et al., 2021). SSPictR (Penzkofer et al., 2025) applies SSPs to segmented visual entities in two dimensions. We use SSPs as the input representation for 3D assets, binding surface geometry and learned material features to continuous surface coordinates in one vector-symbolic interface from which object representations and their composition are learned.

Self-Supervised 3D Representation Learning. Self-supervised pretraining aims to learn 3D representations that generalize without dense annotations. Recent methods vary in the structural biases introduced during learning. Sonata (Wu et al., 2025) shows that point-cloud pretext tasks can be solved from low-level geometric cues and modifies pretraining to remove this shortcut. Stru-Mamba3D (Wang et al., 2025) models structural dependencies among points, instance-aware pretraining (Yang et al., 2026) incorporates object-level scene structure into the learning signal, and E-RayZer (Zhao et al., 2026) uses 3D reconstruction as a spatial pretraining objective.

Joint-embedding predictive architectures (Assran et al., 2023) change what the model predicts: rather than reconstructing observations, it predicts the latent representation of a target region from its context. This approach suits 3D data because effective representations do not require preserving every geometric detail. POMA-3D (Mao et al., 2026) applies joint-embedding prediction to point maps to learn features that are geometrically consistent across views. We perform predictive learning directly over SSP tokens, each of which encodes a continuous surface coordinate, so that the model learns dependencies within and across surface geometry and material appearance over an explicit spatial structure.

Learning Compositional 3D Structure. An effective scene representation must capture both the constituent objects and their layout. Object-centric learning addresses this challenge by factorizing a scene into object-level latent representations. Wiedemer et al. (2024) identifies conditions under which this factorization generalizes to novel compositions. Jung et al. (2024) introduces explicit compositional training signals that facilitate the recombination of independently inferred objects. Karimi Mamaghan et al. (2025) employs visual question answering to assess whether object-centric representations preserve the relations necessary for compositional reasoning.

3D composition depends on placement: each constituent occupies a specific pose within a shared frame, and the layout as a whole conveys additional information. PartGS (Gao et al., 2025) organizes 3D representations around meaningful structural parts. In structured 3D generation, DecompDreamer (Nath et al., 2026) demonstrates that object fidelity and inter-object relationships can conflict. Their composition-aware curriculum separates the learning of global structure from the refinement of local detail. In contrast to previous approaches that infer object structure from observed scenes or organize it for reconstruction and generation, we treat composition as an operation within representation space. Our model maps learned object representations and their placements to a compact representation of the resulting scene. Trained against the representation of the assembled scene, it represents new arrangements of encoded objects without assembly or re-encoding.

3 METHOD

3.1 OVERVIEW

Our goal is to learn representations of 3D assets directly from their meshes and material maps, and then to compose object representations into scene representations without re-encoding each

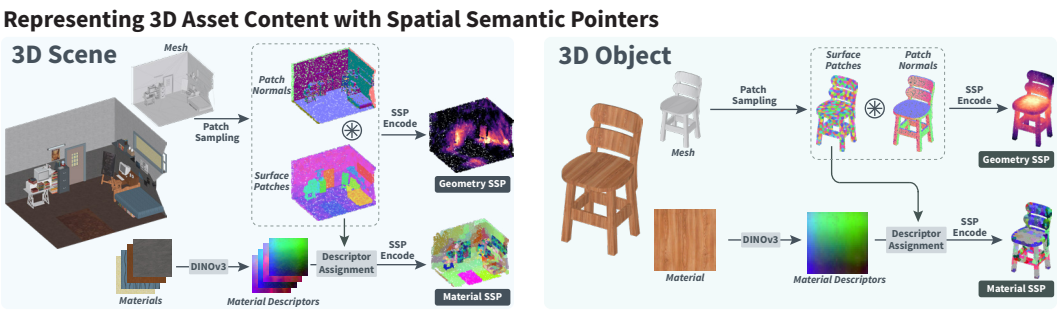

Figure 1: **Tokenization.** Each 3D asset is encoded without rendering. Surface patches carry coordinates and normals, bound by circular convolution (⊛) into geometry tokens; DINOv3 descriptors of the material composite are assigned to patches and bound at the same coordinates into material tokens. The same tokenization applies to individual objects and to composed scenes (Sec. 3.2).

assembled configuration. The method has two parts. First, we represent surface geometry and material appearance as Spatial Semantic Pointer (SSP) tokens and learn asset representations from them through self-supervised prediction (Figure 1). Second, a composition model combines the representations of multiple objects according to their placements (Figure 2).

This design separates the *content* of an object from its *placement* in a particular configuration. The same object representation can be conditioned on different translations, rotations, and scales, and once object representations are available, a new configuration can be represented without assembling the complete 3D scene and passing it through the encoder.

3.2 Representing 3D asset content with Spatial Semantic Pointers

A 3D asset contains several attributes defined over the same spatial domain. Surface geometry associates each surface point with a local orientation. Material appearance associates points on that surface with perceptual features. We preserve this correspondence by binding each local observation to its continuous surface coordinate.

Continuous 3D spatial encoding. The encoding must be continuous in the coordinate, real-valued, direction-symmetric, and multi-scale. An SSP represents a continuous coordinate through the phases of a set of spatial frequencies. For a coordinate $\mathbf{p} \in \mathbb{R}^3$, we define $\phi(\mathbf{p}) = \mathcal{F}^{-1}\big[\exp(\mathrm{i}A\mathbf{p})\big]$, where $A \in \mathbb{R}^{D\times 3}$ is a phase matrix and $\mathcal{F}^{-1}$ is the inverse Fourier transform over the D representation dimensions. Each row $\mathbf{a}_k^\top$ of A specifies a spatial frequency, whose phase at $\mathbf{p}$ is $\mathbf{a}_k^\top \mathbf{p}$. Since the phase varies continuously with $\mathbf{p}$, the map $\phi : \mathbb{R}^3 \to \mathbb{R}^D$ does not require a discrete spatial grid. We use SSPs because binding keeps coordinates and attributes in a single fixed-width space shared across modalities, in which a sum of tokens can still be queried by coordinate (Appendix B).

The frequencies should not privilege a coordinate axis. We therefore construct the directional basis from a regular simplex, whose center-to-vertex directions are equiangular. In three dimensions, this simplex is a tetrahedron. Let $\{\mathbf{u}_j\}_{j=1}^4$ be the unit vectors from its center to its vertices. They satisfy $\sum_j \mathbf{u}_j = \mathbf{0}$ and $\mathbf{u}_j^\top \mathbf{u}_k = -1/3$ for $j \neq k$. The first condition centers the basis, and the second gives equal angular separation between each pair of directions.

A single tetrahedral basis covers only one scale and orientation. We construct scaled and rotated copies using $\mathbf{a}_{s,r,j} = \alpha_s Q_r \mathbf{u}_j$, where α_s is a scale and $Q_r \in SO(3)$ is a rotation. Scaling controls the spatial resolution represented by the Fourier phases. Rotation distributes the basis across orientations. Using n_s scales and n_r rotations gives $K = 4n_s n_r$ frequency vectors, which form the rows of $A_+ \in \mathbb{R}^{K\times 3}$.

Pairing every frequency in A_+ with its negative and adding a zero-frequency component gives a real-valued encoding of dimension $D = 2K + 1 = 8n_s n_r + 1$, continuous in the coordinate across multiple scales and orientations.

Spatially grounded attributes. The map $\phi(\mathbf{p})$ represents where an observation occurs. A separate attribute encoder represents what is observed there. For modality m, let $\mathbf{c}_i^{(m)}$ be the local attribute at surface coordinate $\mathbf{p}_i$, and let ψ_m map that attribute into the common D-dimensional space. The corresponding token is $\mathbf{z}_i^{(m)} = \phi(\mathbf{p}_i) * \psi_m(\mathbf{c}_i^{(m)})$, where $*$ denotes circular-convolution binding.

The coordinate and attribute encoders must use independent phase bases. The coordinate SSP obeys an additive spatial algebra. If a coordinate-valued attribute $\mathbf{c}$ used the same encoder, then $\phi(\mathbf{p}) * \phi(\mathbf{c}) = \phi(\mathbf{p} + \mathbf{c})$, and binding would treat the attribute as a displacement of the surface point. Independent bases preserve the roles of coordinate and attribute while allowing them to interact through the same operation. Appendix B.2 derives the translation identity.

Surface geometry. We represent a mesh as a set of surface patches. Starting from a normalized triangle mesh, we draw area-weighted surface samples. The sampling density therefore depends on surface area rather than mesh tessellation. We select distributed patch centers by farthest-point sampling and assign each surface sample to its nearest center. Patch i is summarized by the mean coordinate $\mathbf{p}_i$ of its assigned samples and their normalized mean normal $\mathbf{n}_i$.

The coordinate and the surface orientation use separate fixed encoders. The coordinate is encoded by ϕ. The normal is encoded as $\psi_{\text{geo}}(\mathbf{n}) = \mathcal{F}^{-1}[\exp(\mathrm{i}B\mathbf{n})]$, where B is sampled independently of the phase matrix A and its frequencies are arranged in conjugate pairs so that the output is real. The geometry token is $\mathbf{z}_i^{\text{geo}} = \phi(\mathbf{p}_i) * \psi_{\text{geo}}(\mathbf{n}_i)$. Each token represents a local surface orientation at a continuous coordinate on the mesh.

Material representation. We derive material features from the asset's native PBR material maps without rendering the asset from multiple views. The available material channels are combined under a fixed Cook–Torrance shading configuration (Cook & Torrance, 1982). A pretrained visual encoder extracts a grid of patch-level descriptors from the resulting material composite.

The descriptors initially occupy the two-dimensional material domain. We map them to the 3D surface through the mesh UV parameterization. Spatial correspondence then associates each valid geometry patch with a perceptual descriptor $\mathbf{d}_i \in \mathbb{R}^{d_v}$. Geometry patches without a sufficiently close material correspondence are marked invalid for the material stream rather than assigned an unrelated descriptor.

A fixed Gaussian projection $R \in \mathbb{R}^{D \times d_v}$ lifts each descriptor into the SSP dimensionality. It is sampled once and shared across all assets, keeping the tokenizer deterministic and independent of any individual asset. Before binding, we normalize the Fourier coefficients of $R\mathbf{d}_i$ to unit magnitude, giving the unitary attribute vector $U(R\mathbf{d}_i)$. Unit Fourier magnitude makes binding norm preserving, and the conjugate spectrum provides its inverse. The material token is $\mathbf{z}_i^{\text{mat}} = \phi(\mathbf{p}_i) * U(R\mathbf{d}_i)$.

The unitary map discards radial magnitude but retains angular structure. For Gaussian R, the expected cosine similarity between two unitary attribute vectors is a strictly increasing function of the cosine similarity between their original descriptors. Appendix C.3 derives this result, and Appendix C.4 evaluates the fixed projection used in our experiments. We combine the geometry and material streams with modality identifiers and validity masks to form the input sequence for the learned encoder.

3.3 Learning 3D Asset Representations

The SSP tokens preserve local surface geometry, material appearance, and continuous surface coordinates. We next learn which properties of these signals can be predicted from the rest of a 3D asset. Geometry and material provide different observations of the same surface. Local shape can inform appearance, appearance can inform shape, and both depend on the surrounding 3D structure. We learn these dependencies through joint-embedding prediction over the aligned token streams.

Multistream spatial prediction. Let $X = [X^{\text{geo}}; X^{\text{mat}}]$ denote the complete SSP token sequence. We mask contiguous regions in 3D rather than selecting independent tokens. Each step uses one of three masking modes. Geometry masking removes only geometry tokens from the region, leaving corresponding material and surrounding context visible. Material masking removes only material tokens, leaving geometry and neighboring material visible. Joint masking removes

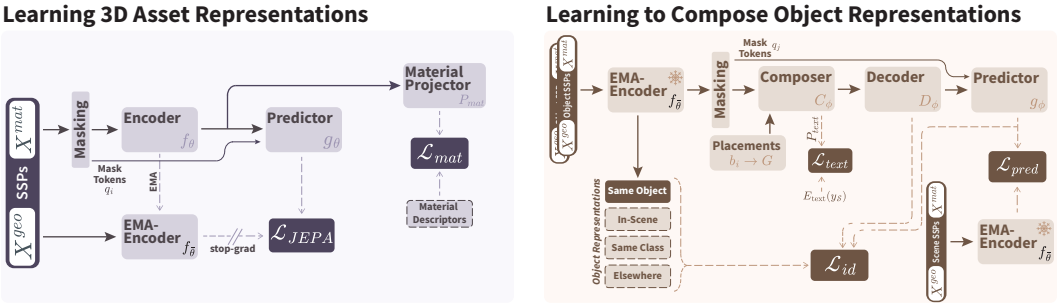

Figure 2: **Left:** An asset encoder f_θ is trained by masked feature prediction against an EMA target $f_{\bar{\theta}}$, with an auxiliary material objective $\mathcal{L}_{\text{mat}}$ (Sec. 3.3). **Right:** A composition operator C_ϕ maps frozen object representations and their placements b_i to a fixed-size scene bank. It is supervised by the frozen encoder's output on the assembled scene ($\mathcal{L}_{\text{pred}}$), by an identification objective $\mathcal{L}_{\text{id}}$ that requires the decoded feature at each location to identify the object placed there among in-scene, same-class, and cross-scene alternatives, and by a contrastive objective $\mathcal{L}_{\text{text}}$ that aligns the pooled bank with a frozen embedding of the scene caption (Sec. 3.4). The decoder, predictor, and projection heads are used only during training. ***Best viewed with zoom-in.***

both streams from the region, so neither modality directly reveals the masked region. The three modes combine cross-modal prediction with prediction from surrounding 3D context.

Let $\mathcal{C}$ and $\mathcal{T}$ denote the visible context and masked target tokens. An online encoder f_θ observes only $X_{\mathcal{C}}$, while an EMA target encoder $f_{\bar{\theta}}$ observes the complete sequence X. For each $i \in \mathcal{T}$, a predictor g_θ receives the encoded context and a query $\mathbf{q}_i$ that identifies the target coordinate and modality. It does not receive the target token. We write $\widehat{\mathbf{h}}_i = g_\theta(f_\theta(X_{\mathcal{C}}), \mathbf{q}_i)$ and $\mathbf{h}_i = f_{\bar{\theta}}(X)_i$. The representation objective is

$$\mathcal{L}_{\text{JEPA}} = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \left\| \widehat{\mathbf{h}}_i - \text{LN}(\mathbf{h}_i) \right\|_1, \tag{1}$$

where LN denotes layer normalization. The target encoder receives no gradients. Its parameters are updated as an exponential moving average of the online encoder.

Preserving material information. Joint-embedding prediction favors information that can be inferred from spatial and cross-modal context. Some perceptual properties of a material may not be predictable from those signals. We therefore add an objective that keeps the original material descriptor accessible from the learned representation. This objective is applied only to visible material tokens, so it does not reveal masked targets to the online encoder.

For each visible material token $i \in \mathcal{V}_{\text{mat}}$, let $\mathbf{h}_i^{\text{mat}}$ be its online-encoder representation and let $\mathbf{d}_i$ be the perceptual descriptor used to construct that token. A learned linear map P_{mat} projects the encoder feature into the perceptual feature space. We define

$$\mathcal{L}_{\text{mat}} = \frac{1}{|\mathcal{V}_{\text{mat}}|} \sum_{i \in \mathcal{V}_{\text{mat}}} \left[1 - \cos\big(P_{\text{mat}}(\mathbf{h}_i^{\text{mat}}), \mathbf{d}_i\big) \right], \tag{2}$$

$$\mathcal{L}_{\text{repr}} = \mathcal{L}_{\text{JEPA}} + \lambda_{\text{mat}} \mathcal{L}_{\text{mat}}. \tag{3}$$

The predictive objective learns spatial and cross-modal regularities. The material objective preserves access to perceptual appearance information.

We train one encoder on individual objects and composed scenes. A learned level embedding distinguishes the two input types. After pretraining, the predictor and material projection head are discarded. The EMA encoder is retained as the 3D representation model. Its local features on individual objects form the object representations used for composition in Sec. 3.4.

3.4 Learning to Compose Object Representations

Once objects have been encoded, a new configuration of them can be represented without assembling and re-encoding the scene. We learn a composition operator that acts directly in representation space.

It receives the learned representations of the constituent objects together with their placements, and produces a compact representation of the configuration they form.

Composition model. Let $Z_i = \{\mathbf{z}_{ij}\}_{j=1}^{M_i}$ denote the learned representation of object i, where $\mathbf{z}_{ij} \in \mathbb{R}^d$ is a local feature. These features describe the object independently of any particular composition. A placement specifies how the object enters the shared 3D frame.

Let b_i contain the translation, rotation, and scale of object i. A learned placement encoder maps it to feature-wise modulation parameters $(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i) = G(b_i)$, which condition each feature as $\tilde{\mathbf{z}}_{ij} = (1 + \boldsymbol{\gamma}_i) \odot \mathbf{z}_{ij} + \boldsymbol{\beta}_i$. This operation separates object content from its placement. A chair encoded once can be provided to the composer under any number of distinct placements.

Given N objects and their placements, the composition model computes a fixed-size set of L latents $Z_{\mathcal{S}} = C_\phi\big(\{(Z_i, b_i)\}_{i=1}^N\big) \in \mathbb{R}^{L \times d}$, referred to as the composition latent bank, which represents the joint 3D configuration.

Supervision from the assembled scene. We train C_ϕ to align with the representation of the actual assembled scene. The composition model receives only the constituent object representations and their placements. During training, the assembled composition $\mathcal{S}$ is encoded by the frozen representation model, producing a spatial target set $H_{\mathcal{S}}$. The representation model is never updated during composition learning, so every target lives in a fixed space.

The target set may contain a variable number of features per scene, whereas $Z_{\mathcal{S}}$ always contains L latents. A decoder D_ϕ, used only during training, connects these representations. Given a query $\mathbf{q}_j$ specifying a scene location and modality, the decoder retrieves the corresponding feature from $Z_{\mathcal{S}}$.

Masked prediction. We define matching regions through their spatial support. Each constituent feature is transformed into the composition's coordinate frame using its object's placement. Constituent and scene features are then assigned to a shared voxel grid. A match requires both features to belong to the same object and occupy the same spatial cell. Masking a matched region of the constituent input specifies the region to be predicted in the scene representation.

At the masked scene locations $\mathcal{T}$, a training-only predictor g_ϕ maps the decoded feature to the frozen scene feature:

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{T}|} \sum_{j \in \mathcal{T}} \left\| g_\phi\left(D_\phi(Z_{\mathcal{S}}, \mathbf{q}_j)\right) - \text{LN}(\mathbf{h}_j^{\mathcal{S}}) \right\|_1, \tag{4}$$

where $\mathbf{h}_j^{\mathcal{S}} \in H_{\mathcal{S}}$ is the frozen scene feature at query $\mathbf{q}_j$.

Content identification. We additionally require the decoded feature at each location to identify the object that occupies it. For a location $\mathbf{q}_j$ owned by object o, let $\mathbf{a}_j$ be the normalized projection of the decoded feature at $\mathbf{q}_j$, obtained from D_ϕ at visible locations and from g_ϕ at masked ones. Let $\mathbf{t}_o$ be the normalized mean of the constituent features of object o. Negative samples are drawn from three sets F: other objects in the scene, objects of the same category from the training set, and objects from other scenes in the batch. With positives $\mathcal{P}_j$ (the owner and any identical object) and negatives $\mathcal{N}_F(j)$,

$$\ell_F(j) = -\log \frac{\sum_{o \in \mathcal{P}_j} e^{\langle \mathbf{a}_j, \mathbf{t}_o \rangle / \tau}}{\sum_{o \in \mathcal{P}_j} e^{\langle \mathbf{a}_j, \mathbf{t}_o \rangle / \tau} + \sum_{n \in \mathcal{N}_F(j)} e^{\langle \mathbf{a}_j, \mathbf{t}_n \rangle / \tau}}, \qquad \mathcal{L}_{\text{id}} = \sum_F w_F \, \overline{\ell_F}, \tag{5}$$

where $\overline{\ell_F}$ averages over all owned locations on both readout paths.

Language supervision. The prediction and identification objectives supervise the latent bank one location at a time. A description of the composition supervises it as a whole: it names the objects present and how they are arranged, so matching it requires the bank to encode which objects the configuration contains and not only its layout. Each training composition is paired with a natural-language caption $y_{\mathcal{S}}$ (Appendix E.3), embedded by a frozen pretrained sentence encoder E_{text}. Because a caption describes the complete configuration, this objective is computed on the latent bank of the unmasked composition. We pool the bank and project it into the embedding space of the sentence encoder,

$$\mathbf{u}_{\mathcal{S}} = \frac{P_{\text{text}}\big(\frac{1}{L}\sum_{l=1}^{L}\mathbf{z}_{\mathcal{S},l}\big)}{\big\|P_{\text{text}}\big(\frac{1}{L}\sum_{l=1}^{L}\mathbf{z}_{\mathcal{S},l}\big)\big\|_2}, \qquad \mathbf{c}_{\mathcal{S}} = \frac{E_{\text{text}}(y_{\mathcal{S}})}{\|E_{\text{text}}(y_{\mathcal{S}})\|_2}, \tag{6}$$

where P_{text} is a learned linear map. Over a batch of B compositions, a symmetric contrastive objective matches each composition to its own caption:

$$\mathcal{L}_{\text{text}} = -\frac{1}{2B}\sum_{b=1}^{B}\left[\log\frac{e^{\langle \mathbf{u}_b,\mathbf{c}_b\rangle/\tau_t}}{\sum_{b'} e^{\langle \mathbf{u}_b,\mathbf{c}_{b'}\rangle/\tau_t}} + \log\frac{e^{\langle \mathbf{u}_b,\mathbf{c}_b\rangle/\tau_t}}{\sum_{b'} e^{\langle \mathbf{u}_{b'},\mathbf{c}_b\rangle/\tau_t}}\right]. \tag{7}$$

Training objective. The composition model is trained with

$$\mathcal{L}_{\text{comp}} = \mathcal{L}_{\text{pred}} + \lambda_{\text{id}}\,\mathcal{L}_{\text{id}} + \lambda_{\text{text}}\,\mathcal{L}_{\text{text}}. \tag{8}$$

The prediction objective ties the latent bank to the representation of the assembled scene. The identification objective requires each bank location to identify the object placed there. The language objective requires the bank as a whole to encode what the configuration contains. The decoder, predictor, and projection heads are used only during training, and captions are not needed at inference. Only C_ϕ and its output $Z_{\mathcal{S}}$ are retained.

4 Experiments

Implementation details. We train on 22,718 scenes composed from 326,084 objects in InternScenes (Zhong et al., 2025), holding out 1,020 evaluation scenes excluded from training of both the encoder and the composition operator; the scene corpus, captions, and question-answer pairs are described in Appendix E. Each object is tokenized into 512 surface patches at $D = 3201$, with material descriptors from DINOv3 ViT-L/16 (Siméoni et al., 2025). The asset encoder is a 24-layer Transformer of width 1024 (307M parameters), trained on individual objects and composed scenes for 150k steps with $\lambda_{\text{mat}} = 0.1$. The composition operator has four layers of width 1024 and $L = 64$ latents on a 4^3 grid (103M parameters), trained for 110k steps with $\lambda_{\text{id}} = \lambda_{\text{text}} = 0.1$ and a frozen MPNet sentence encoder (Song et al., 2020; Reimers & Gurevych, 2019). For question answering, the 64 latents enter Qwen2.5-7B-Instruct (Yang et al., 2024) or LLaVA-OneVision-7B (Li et al., 2024) as soft tokens through a two-layer projector, which is aligned on scene captions with the language model frozen; the model is then adapted with LoRA (Hu et al., 2021) of rank 32 on our question-answer pairs and captions, and finally on each benchmark's training split. Additional details are deferred to Appendix D.

4.1 Comparison with State-of-the-Art Methods

Table 1 compares ROSE-3D with specialist models, 2D and 3D language-model systems, and POMA-3D on the ScanQA, SQA3D, and MSNN benchmarks. ROSE-3D is evaluated on the InternScenes reconstructions of the benchmark scenes, in which objects and their placements are given and whose unlabeled geometry is seen in training, whereas the baselines process the original scans (Appendix E.2); with the same reconstructions and adaptation, it outperforms encoding the assembled scene and pooling the object representations on QA (Appendix A.6). Using Qwen2.5 and LLaVA-OneVision, neither of which receives images, ROSE-3D achieves 24.5 and 23.9 EM@1 on ScanQA, and 52.8 and 53.4 EM@1 on SQA3D, respectively, competitive with POMA-3D, LEO, and the specialist and 2D language-model baselines under these different settings. On MSNN, ROSE-3D is comparable to the strongest methods in both four- and eight-direction accuracy.

LLaVA-3D and Video-3D LLM achieve higher scores on both benchmarks, and Chat-Scene on SQA3D. LLaVA-3D and Video-3D LLM represent scenes with hundreds of tokens derived from posed RGB-D views, processed by a vision encoder aligned with language on large image-text corpora before any 3D training. ROSE-3D instead uses 64 latents composed from object representations learned from meshes and material maps, without rendering views.

ROSE-3D's cost advantage lies in new arrangements of encoded objects. Encoding all 24 objects of a scene takes 0.54 s in total, about 23 ms per object, and each object is encoded only once. Every new arrangement then requires a single 4.6 ms composer pass: 19 times faster than the fastest baseline (87 ms) and at least 39 times faster than view-based methods, which must also render new views. These comparisons include the perception each baseline performs for a new arrangement, and all baseline values are lower bounds (Appendix D.5).

Table 1: Comparison on 3D question answering (ScanQA, SQA3D) and situated next-step navigation (MSNN). New arr.: cost of representing a new arrangement of a 24-object scene on one A100; baseline values are lower bounds that time only each method's public components, and all rows exclude language-model inference (Appendix D.5). †: LoRA-tuned language model.

Method	Modality	ScanQA	SQA3D	MSNN		New arr. (ms) ↓
		EM@1 ↑	EM@1 ↑	4 dir. ↑	8 dir. ↑	
ScanQA (Azuma et al., 2022)	C-PC	21.1	47.2	–	–	87
SQA3D (Ma et al., 2022)	C-PC	–	46.6	–	–	87
3D-VisTA (Zhu et al., 2023)	C-PC	22.4	48.5	39.9	20.1	354
SceneVerse (Jia et al., 2024)	C-PC	22.7	49.9	36.0	19.5	354
FG-CLIP (Xie et al., 2025)	PM	20.9	49.5	39.3	20.4	222
POMA-3D$_{\mathrm{spec}}$ (Mao et al., 2026)	PM	22.3	51.1	**40.4**	21.2	222
LEO (Huang et al., 2023)	C-PC	24.5	50.0	–	–	466
Qwen2.5-VL-7B (Bai et al., 2025)	RGB	23.7	47.8	21.8	2.87	298
LLaVA-OV-7B (Li et al., 2024)	RGB	20.8	47.7	24.0	5.83	180
SplatTalk† (Thai et al., 2025)	RGB (GS)	22.4	47.6	–	–	180
Chat-Scene (Huang et al., 2024)	C-PC + RGB	–	54.6	–	–	352
LLaVA-3D (Zhu et al., 2025)	RGB-D	27.0	55.6	22.9	12.3	182
Video-3D LLM (Zheng et al., 2025)	RGB-D	**30.1**	**58.6**	–	–	182
POMA-3D$^{\dagger}_{\mathrm{llm}}$ (Mao et al., 2026)	PM	21.3	51.6	36.9	**21.4**	222
ROSE-3D† (LLaVA-OV-7B)	SSP	23.9	53.4	40.1	19.9	**4.6**
ROSE-3D† (Qwen2.5-7B)	SSP	24.5	52.8	40.3	21.1	**4.6**

4.2 Impact of Auxiliary Objectives

Material preservation. Masked prediction retains what context can predict, and appearance is only weakly predictable from spatial and cross-modal context. Removing $\mathcal{L}_{\mathrm{mat}}$ accordingly reduces the Spearman correlation between descriptor and encoded-feature similarity from 0.485 to 0.329.

Content identification. The composition operator faces a stronger version of this problem: prediction of scene features can be satisfied largely from the layout of objects, without the object representations. Intervening on the operator's input across 256 held-out scenes at matched training steps (Appendix A), we find that with prediction alone, removing every object barely changes the representation (pooled cosine 0.969 versus 0.529 with $\mathcal{L}_{\mathrm{id}}$), and reassigning content or exchanging two same-category objects produces less than a quarter of the response obtained with identification (0.080 versus 0.364 and 0.068 versus 0.300, relative to the change in the operator's inputs).

Language supervision. Identification acts per location, whereas a caption requires the bank as a whole to account for what the scene contains. Language supervision alone repairs the shortcut at least as effectively: without $\mathcal{L}_{\mathrm{id}}$, removing every object changes the pooled representation, on which the language objective acts directly, almost entirely (cosine 0.185), and the responses to reassignment and same-category replacement reach 0.393 and 0.478. Combining the two gives the strongest response, and identification requires no language annotation.

5 Conclusion and Future Work

We introduced ROSE-3D, which represents 3D scenes by composing reusable object representations in representation space, without rendering or re-encoding the scene. We train the composition operator to identify the object at each location and to match a description of the scene, since agreement with a scene encoder alone can be satisfied largely from the layout. The resulting representation follows object content, including for objects absent from every training scene and after replacement, and is competitive with point-map and point-cloud methods on ScanQA, SQA3D, and MSNN while representing each new arrangement of encoded objects in milliseconds. While our primary focus has been on learning composition, question-answering accuracy remains below that of the strongest 3D language models, whose vision encoders are pretrained on large image-language corpora; aligning our asset encoder with language during pretraining may close this gap. Extending composition to raw scans, through instance segmentation and asset retrieval, and to articulated and deformable objects is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data, including captions and question-answer pairs (see Appendix E). Additionally, we used generative AI tools, specifically Grammarly, ChatGPT and Claude, for sentence restructuring, paraphrasing, and identifying relevant literature. All AI-assisted outputs were reviewed: human evaluators verified the reference question-answer set, and the authors verified all citations against the original publications using Google Scholar. We assume full responsibility for the final content of this work.

ETHICS STATEMENT

Human evaluators, engaged through a professional annotation vendor, answered multiple-choice questions about rendered views of synthetic and reconstructed indoor rooms under written guidelines (Appendix E.4); the task collected no personal information and did not study the evaluators themselves. Our experiments use InternScenes, ScanNet, ScanQA, SQA3D, and MSR3D under their respective licenses and terms of use, with ScanNet and MSR3D accessed through their authorized request processes. The released corpus will follow these upstream terms, and content derived from ScanNet scenes will be released only in forms permitted by the ScanNet Terms of Use. Captions and question-answer pairs generated by language models may contain errors or reflect biases of those models. Generated pairs are filtered by an independent language-model judge, and only the held-out reference set is verified by human evaluators (Appendix E.4).

REPRODUCIBILITY STATEMENT

The method is specified in Section 3, including all training objectives. Appendix D lists architectures, hyperparameters, the adaptation procedure for question answering, the evaluation protocol for each benchmark, and the timing protocol behind the cost comparison. The construction of the spatial semantic pointer tokens, including the material encoding and a proof of its similarity-preservation property, is given in Appendices B and C. Construction of the scene corpus, evaluation splits, captions, and question-answer pairs is described in Appendix E. Because captions and question-answer pairs were generated by stochastic language models, we will release the generated corpus itself, together with code and trained models, upon acceptance. Baseline accuracies are quoted from Mao et al. (2026) and Huang et al. (2024) for Chat-Scene, and baseline costs were measured with public checkpoints following Appendix D.5.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19107–19117. IEEE, 2022.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

Shmuel Friedland, Lek-Heng Lim, and Jinjie Zhang. An elementary and unified proof of grothendieck's inequality. *L'Enseignement Mathématique*, 64(3):327–351, 2019.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.

Ashwinkumar Ganesan, Hang Gao, Sunil Gandhi, Edward Raff, Tim Oates, James Holt, and Mark McLean. Learning with holographic reduced representations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zhirui Gao, Renjiao Yi, Yuhang Huang, Wei Chen, Chenyang Zhu, and Kai Xu. Self-supervised learning of hybrid part-aware 3d representations of 2d gaussians and superquadrics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9649–9659, 2025.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Uffe Haagerup. A new upper bound for the complex grothendieck constant. *Israel Journal of Mathematics*, 60(2):199–224, 1987.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pp. 289–310. Springer, 2024.

Whie Jung, Jaehoon Yoo, Sungjin Ahn, and Seunghoon Hong. Learning to compose: Improving object centric learning by injecting compositionality. In *International Conference on Learning Representations*, 2024.

Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2):139–159, 2009.

Amir Mohammad Karimi Mamaghan, Samuele Papa, Karl H. Johansson, Stefan Bauer, and Andrea Dittadi. Exploring the effectiveness of object-centric representations in visual question answering: Comparative insights with foundation models. In *International Conference on Learning Representations*, 2025.

Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges. *ACM Computing Surveys*, 55(9):1–52, 2023.

Brent Komer, Terrence C. Stewart, Aaron R. Voelker, and Chris Eliasmith. A neural representation of continuous space using fractional binding. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*. Cognitive Science Society, 2019.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes. *Advances in Neural Information Processing Systems*, 2024.

Thomas Lu, Aaron R. Voelker, Brent Komer, and Chris Eliasmith. Representing spatial relations with fractional binding. In *41st Annual Meeting of the Cognitive Science Society*, Montreal, QC, 2019. Cognitive Science Society.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Ye Mao, Weixun Luo, Ranran Huang, Junpeng Jing, and Krystian Mikolajczyk. POMA-3D: The point map way to 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 7282–7292, 2026.

Utkarsh Nath, Rajeev Goel, Rahul Khurana, Kyle Min, Mark Ollila, Pavan K. Turaga, Varun Jampani, and Tejaswi Gowda. Decompdreamer: A composition-aware curriculum for structured 3d asset generation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL `https://openreview.net/forum?id=3qy4J6QFbn`. J2C Certification.

Anna Penzkofer, Karim Habashy, Chris Eliasmith, and Andreas Bulling. Sspictr: A biologically-plausible image representation. In *ECAI Workshop on Artificial Intelligence and Cognition (AIC)*, pp. 1–13, 2025.

Tony A. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3): 623–641, 1995. doi: 10.1109/72.377968.

Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286, 2019.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223. IEEE, 2023.

W. F. Sheppard. On the application of the theory of error to cases of normal distribution and normal correlation. *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character*, 192:101–167, 1899. doi: 10.1098/rsta.1899.0003.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1–2):159–216, 1990. doi: 10.1016/0004-3702(90)90007-M.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

Anh Thai, Songyou Peng, Kyle Genova, Leonidas Guibas, and Thomas Funkhouser. Splattalk: 3d vqa with gaussian splatting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4712–4721. IEEE, 2025.

Aaron R. Voelker, Peter Blouw, Xuan Choo, Nicole Sandra-Yaffa Dumont, Terrence C. Stewart, and Chris Eliasmith. Simulating and predicting dynamical systems with spatial semantic pointers. *Neural Computation*, 33(8):2033–2067, 2021. doi: 10.1162/neco_a_01410.

Chuxin Wang, Yixin Zha, Wenfei Yang, and Tianzhu Zhang. StruMamba3D: Exploring structural mamba for self-supervised point cloud representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28546–28555, 2025.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *International Conference on Learning Representations*, 2024.

Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *European Conference on Computer Vision*, pp. 197–213. Springer, 2022.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22193–22204, 2025.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Gengshen Zhang, Dawei Leng, and Yuhui Yin. Fg-clip: Fine-grained visual and textual alignment. *arXiv preprint arXiv:2505.05071*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13759–13768. IEEE, 2021.

Bin Yang, Mohamed Abdelsamad, Miao Zhang, and Alexandru Paul Condurache. Towards foundation models for 3d scene understanding: Instance-aware self-supervised learning for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2938–2947, 2026.

Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-RayZer: Self-supervised 3d reconstruction as spatial visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7525–7535, 2026.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8995–9006. IEEE, 2025.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Luo Li, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Xudong Xu, Bo Dai, and Jiangmiao Pang. InternScenes: A large-scale simulatable indoor scene dataset with realistic layouts. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1120.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4295–4305. IEEE, 2025.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2899–2909. IEEE, 2023.

A Composition Analysis

This section examines the central property of the composition operator: its output reflects which objects a scene contains, not only where they are placed. We define the controlled edits used throughout in Appendix A.1, and use them in Appendix A.2 to show how the training objectives shape the representation's response. Appendices A.3 and A.4 extend the analysis to unseen objects and larger scenes. Appendix A.5 then tests whether the representation identifies the correct object after an edit, and Appendix A.6 compares composition with simpler representations built from the same inputs.

A.1 Intervention Protocol

We evaluate how the composed representation responds to controlled changes in its input. Each intervention modifies the object representations provided to the composition operator while placements remain unchanged, and nothing is re-encoded. *Owner permutation* reassigns content among the existing placements: each placement receives the representation of the preceding object in scene order, with rows subsampled or resampled deterministically so that every placement keeps its number of feature rows. *Compatible swap* replaces each object with a different object of the same category. *Two-object swap* exchanges the content of two objects of the same category. *Zero content* replaces all object representations with zeros.

For the first three, the composer's response in a scene is the relative change of its latent bank, $\|Z'_{\mathcal{S}} - Z_{\mathcal{S}}\|_F / \|Z_{\mathcal{S}}\|_F$, over all 64 latents and feature dimensions. Because this quantity lacks an absolute scale, we apply the same intervention to the frozen encoder's stacked object-feature rows X and compute the analogous relative change $\|X' - X\|_F / \|X\|_F$. We report the mean composer response over scenes divided by the mean encoder response over the same scenes. This ratio measures how much of the change in its inputs the operator carries into the scene representation; comparison with re-encoding the edited scene is the subject of Appendix A.5. For zero content, we report the cosine similarity between the latent bank averaged over its 64 latents before and after the intervention, where lower values indicate stronger dependence on content; it admits no encoder reference, since it replaces representations rather than rearranging them.

The objective ablations (Appendix A.2) and the identification analysis (Appendix A.5) use a fixed subset of 256 held-out scenes (Appendix E.2). Within it, compatible swaps apply to the 114 scenes with an available same-category replacement, and two-object swaps to the 227 scenes containing two objects of the same category, of which the objective ablations use the first 100. All other analyses use all 1,020 held-out scenes, normalizing by the encoder's response within each group, and all random choices in constructing the interventions use a fixed seed.

A.2 Objective Ablations

Table 2 presents all combinations of the identification and language objectives added to prediction, each evaluated at 50k steps.

With prediction alone, the operator encodes the layout of a scene but not its content: each intervention changes the representation by less than a tenth as much as it changes the operator's inputs, and removing every object leaves the pooled representation nearly unchanged. The operator satisfies its objective from where objects are, without using what they are.

Either auxiliary objective repairs this shortcut. Identification requires each location of the bank to identify the object placed there, whereas a caption requires the bank as a whole to account for what the scene contains. Each therefore makes the object representations necessary, raising the response to every intervention four- to fivefold and sharply reducing the similarity after content removal.

The two objectives are complementary. Combining them gives the strongest response to every swap intervention, most clearly for compatible swap, where only content distinguishes the objects. Identification contributes this without language annotation, whereas language supervision requires a caption for each training scene. The similarity after content removal stays above zero because placements are preserved: the representation depends primarily on content while still encoding where objects are.

Table 2: Objective ablations. Intervention responses use composers trained for 50k steps and are fractions of the frozen encoder's response to the same intervention; zero content is pooled cosine, where lower indicates stronger dependence on content. ScanQA EM@1 uses the fully trained composers (110k steps) with Qwen2.5-7B

Objective	Owner perm. ↑	Compat. swap ↑	Two-object swap ↑	Zero-content cos. ↓	ScanQA (EM@1) ↑
$\mathcal{L}_{\text{pred}}$	0.080	0.097	0.068	0.969	21.5
$+\,\mathcal{L}_{\text{id}}$	0.364	0.379	0.300	0.529	23.9
$+\,\mathcal{L}_{\text{text}}$	0.393	0.478	0.366	0.185	24.2
$+\,\mathcal{L}_{\text{id}} + \mathcal{L}_{\text{text}}$	**0.416**	**0.531**	**0.375**	**0.188**	**24.5**

A.3 Generalization to Unseen Objects

We examine whether composition extends to objects the composition operator never encountered during training. Of the objects in the held-out scenes, 12,042 appear in no training scene; we compare the 670 held-out scenes containing at least one of them with the 350 containing only objects seen in training. Unseen objects make up about half of the objects in a typical scene of the first group (median 52%), so these scenes are dominated by objects the operator has never composed. Scenes with unseen objects are also larger (median 35 objects against 18), and responses depend on scene size (Appendix A.4), so Table 3 compares the two groups within the size ranges that both populate.

Within each size range, responses to owner permutation and two-object swaps are close between the two groups. The similarity after content removal is somewhat higher with unseen objects, so the representation depends slightly less on their content, but it remains low. Because the operator receives each object only through its encoded representation, it composes objects it has never seen much as it composes familiar ones.

Table 3: Generalization to objects not seen by the composition operator during training, for the final model at 110k steps, compared within ranges of scene size. Compatible swaps are omitted because few scenes with only seen objects admit them.

Objects per scene	Scenes	n	Owner perm. ↑	Two-object ↑	Zero content ↓
10 to 24	With unseen objects	125	0.50	0.52	0.20
	Seen objects only	172	0.51	0.48	0.13
25 to 49	With unseen objects	447	0.46	0.50	0.09
	Seen objects only	105	0.49	0.49	0.00

A.4 Scaling with Object Count

We examine whether composition degrades as scenes contain more objects. Table 4 groups the 1,020 held-out scenes by object count; since exchanging two objects changes a dense scene less than a sparse one, each group is normalized by the encoder's response within it.

From the sparsest to the densest scenes, the responses to owner permutation and compatible swap decrease only slightly, and the response to a two-object swap increases. Although the bank has 64 latents regardless of scene size, it retains its dependence on individual objects even in scenes with 50 or more objects.

A.5 Identification After Edits

The interventions in Appendix A.1 measure how much the composed representation changes when objects are edited. Here we test whether it changes correctly: after objects are replaced or rearranged, does the representation at a location identify the object now placed there?

We edit a scene and compose it twice, once before the edit and once after. Using the identification readout trained with $\mathcal{L}_{\text{id}}$ (Sec. 3.4), applied without further training, we query both banks at the

Table 4: Composer response by scene size for the final model at 110k steps, as fractions of the encoder's response.

Objects per scene	#Scenes	Owner perm.	Compatible	Two-object
Fewer than 10	75	0.505	0.763	0.410
10 to 24	297	0.504	0.702	0.497
25 to 49	552	0.464	0.681	0.500
50 or more	96	0.433	0.674	0.608

surface points of the edited scene, so that only the representation differs between them; the choice of points alone would otherwise change the result by up to 0.06. As in Eq. 5, we compare each readout feature with the templates of the newly placed object and of the object it displaced, and report the fraction of points at which it is more similar to the new one, averaged over scenes (Table 5). As a reference, we train a readout of the same form on the frozen encoder's features and apply it to encodings of the original and edited scenes at the same points. A head trained on the frozen composer in the same way as the encoder's gives the same shifts to within about one point.

The readout is informative but separates objects of the same category only partially: on unedited scenes, it prefers the true object over another object in the scene at 0.74 of points for the composer and 0.79 for the encoder, and over one of the same category at 0.57 and 0.62. These values give an empirical reference for same-category discrimination, and we judge the composer by the shift that re-composition produces and by how close it comes to this reference, relative to re-encoding.

Table 5: Preference for the newly placed object before and after an edit, with all queries at the edited scene's surface points (mean over scenes; higher values indicate a stronger preference for the new object). The composer re-composes the edited scene; the encoder re-encodes it and is read out by a head trained in the same way. Final model, diagnostic subset of held-out scenes.

		Composer		Encoder	
Edit	#Scenes	Before	After	Before	After
Owner permutation	254	0.41	0.43	0.35	0.55
Compatible swap	114	0.28	0.47	0.39	0.51
Two-object swap	227	0.47	0.47	0.44	0.57

All three edits leave the layout unchanged, so a representation that relied on layout would register none of them. A compatible swap changes which objects the scene contains; owner permutations and two-object swaps keep the same objects and change only which object occupies each placement.

When a new object enters the scene, re-composition registers it. The preference for the new object rises from 0.28 to 0.47, with increases in 99% of scenes. On matched objects and query points, the post-edit preference falls 0.094 below the composer's unedited same-category reference and 0.091 below the encoder's, a paired difference of 0.003 (95% interval $[-0.020, 0.025]$ from 2,000 bootstrap resamples of scenes). Replacement thus costs the two systems a similar amount relative to their unedited discrimination, so the composer's lower post-edit value largely reflects its weaker same-category discrimination, and the composed representation carries the content of the objects placed in it without re-encoding the scene.

When objects already present exchange placements, re-encoding registers the change at the cost of a full encoder pass, while re-composition shifts identification only slightly. The composed representation is therefore most sensitive to which objects a scene contains. Supplying the operator with scene-frame patch coordinates does not remedy this. Following the protocol of Appendix A.2, we add them to the model trained with prediction and identification, whose identification objective supervises each location:: at 50k steps, coordinates remove most of the response to content, reducing the shift for compatible swaps from 0.11 to 0.00, consistent with explicit coordinates strengthening the reliance on layout that our objectives counter.

A.6 Composition Controls

To separate the contribution of the representation from that of the input setting and adaptation data, we compare ROSE-3D with two representations built from the same InternScenes reconstructions, with the same token budget, adaptation data, and procedure. *Pooled scene encoding* mean-pools the frozen encoder's features of the assembled scene into the same 4^3 grid of cells as the composer's latent queries, with empty cells set to zero. *Pooled objects* mean-pools each object's mean representation, plus a fixed Fourier embedding of its placement, into the cell containing the object's center. All three are adapted with Qwen2.5-7B under the same pipeline(Table 6).

Table 6: EM@1 of the composition controls with Qwen2.5-7B. All three representations use identical reconstructions, adaptation data, and processing pipelines.

Representation	ScanQA ↑	SQA3D ↑	MSNN 4 dir. ↑	MSNN 8 dir. ↑
Pooled scene encoding	22.6	51.6	39.8	19.6
Pooled objects	24.0	51.8	39.8	21.0
ROSE-3D	24.5	52.8	40.3	21.1

ROSE-3D achieves the highest accuracy on all four metrics. Encoding the assembled scene from the same reconstructions is lowest on ScanQA and SQA3D, so the reconstructed inputs and adaptation data do not account for ROSE-3D's accuracy on their own. Pooling object representations, which keeps objects separate, comes closer, and learned composition improves on it on every metric.

B Spatial Semantic Pointers

B.1 Construction

SSP tokens are computed over surface patches. We scale each asset so that its largest bounding-box extent is 1, draw 40,000 area-weighted surface samples, and select 512 patch centers among them by farthest-point sampling; the normal of each patch is the area-weighted mean of the face normals it contains.

The phase matrix A is built from the four unit vectors of a regular tetrahedron, scaled by $\alpha_s = s/20$ for $s = 1, \ldots, 20$ and rotated by 20 rotations Q_r drawn uniformly from $SO(3)$, giving $K = 4 \times 20 \times 20 = 1600$ directed frequencies. Conjugate pairing and a zero frequency give $D = 2K + 1 = 3201$. Coordinates are divided by a length scale $\ell = 0.008$ in normalized units before the phase is computed. The normal phase matrix B has K rows, unit directions drawn uniformly on the sphere, each scaled by an independent radius drawn uniformly from $[0, 1)$.

B.2 Translation Algebra

The continuous spatial encoding introduced in Sec. 3.2 admits an exact algebra over translations, which constrains how attributes must be encoded. Recall that

$$\phi(\mathbf{p}) = \mathcal{F}^{-1}\left[\exp(\mathrm{i}A\mathbf{p})\right]. \tag{9}$$

For a displacement $\Delta\mathbf{p}$, element-wise multiplication of the corresponding Fourier representations gives

$$\exp(\mathrm{i}A\mathbf{p}) \odot \exp(\mathrm{i}A\Delta\mathbf{p}) = \exp\left(\mathrm{i}A\mathbf{p} + \mathrm{i}A\Delta\mathbf{p}\right) \tag{10}$$

$$= \exp\left(\mathrm{i}A(\mathbf{p} + \Delta\mathbf{p})\right). \tag{11}$$

Since SSP binding corresponds to element-wise multiplication in the Fourier domain,

$$\phi(\mathbf{p}) * \phi(\Delta\mathbf{p}) = \phi(\mathbf{p} + \Delta\mathbf{p}). \tag{12}$$

Thus, continuous translation in the spatial domain corresponds exactly to binding by the SSP of the displacement.

This property also motivates the use of independent phase bases for surface coordinates and modality-specific attributes. If the same spatial encoder ϕ were used for a coordinate-valued attribute $\mathbf{c}$, then

$$\phi(\mathbf{p}) * \phi(\mathbf{c}) = \phi(\mathbf{p} + \mathbf{c}), \tag{13}$$

making the roles of coordinate and attribute indistinguishable under binding. We therefore encode attributes using independently constructed bases ψ_m, as described in Sec. 3.2.

Binding by $\phi(\Delta p)$ could also move an object's tokens, but it acts before the encoder, so every new placement would require re-encoding the object. We therefore apply placements to the learned object representations (Sec. 3.4), which keeps each object's encoding reusable across scenes.

C Material Representation

C.1 Feature Construction

We derive perceptual material features directly from the native PBR material maps rather than from multiview renders of the 3D asset. The base color, metallic, roughness, and normal maps are combined into a 512×512 composite using a fixed Cook–Torrance shading configuration (Cook & Torrance, 1982) under a fixed three-light setting, and DINOv3 ViT-L/16 (Siméoni et al., 2025) processes the composite into a 32×32 grid of descriptors with $d_v = 1024$.

These descriptors initially occupy the two-dimensional material domain. We map them to the 3D surface using the mesh UV parameterization: the surface coordinate of each visual patch is recovered through UV-to-triangle correspondence, with one material sample per triangle, and each geometry patch is associated with its nearest material sample in 3D. A patch receives a descriptor only when a sample lies within 0.05 of it in normalized coordinates; otherwise it is marked invalid for the material stream and its material token is zero, rather than being assigned an unrelated descriptor. This produces a perceptual descriptor $\mathbf{d}_i \in \mathbb{R}^{d_v}$ aligned with each valid patch coordinate $\mathbf{p}_i$.

C.2 Projection and Unitary Encoding

Material descriptors and SSP tokens occupy spaces of different dimensionality. We use a fixed Gaussian matrix $R \in \mathbb{R}^{D \times d_v}$, with entries sampled independently as $R_{ka} \sim \mathcal{N}(0, 1/d_v)$, to lift each descriptor into the SSP space:

$$\widetilde{\mathbf{d}}_i = R\mathbf{d}_i. \tag{14}$$

The matrix is sampled once and shared across all assets. Unlike a learned projection, it introduces no dataset-dependent parameters before SSP binding and maps identical perceptual descriptors consistently across assets.

Before binding the projected descriptor with its coordinate, we normalize its Fourier coefficients to unit magnitude. For $\mathbf{y} = R\mathbf{d}$, we define

$$U(\mathbf{y}) = \mathrm{Re}\left[\mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y})}{|\mathcal{F}(\mathbf{y})|}\right)\right], \tag{15}$$

where the denominator is numerically stabilized for coefficients with near-zero magnitude. Thus, U retains the Fourier phases of $\mathbf{y}$ while setting its Fourier magnitudes to one. For a real input, conjugate symmetry makes the inverse transform real. Under the standard unnormalized-forward and $1/D$-normalized inverse Fourier convention, Parseval's identity gives $\|U(\mathbf{y})\|_2 = 1$.

This normalization serves the circular-convolution binding operation. Multiplication by a unit-magnitude Fourier spectrum preserves the norm of the bound representation, and its inverse is obtained by conjugating that spectrum. The material attribute vector is therefore $U(R\mathbf{d}_i)$, which is bound to the coordinate SSP as described in Sec. 3.2.

The complete transformation $U \circ R$ is nonlinear and does not preserve arbitrary Euclidean distances. The following subsection characterizes the angular structure retained by this transformation.

C.3 Why Similarity Survives Unitary Encoding

The unitary map U discards the magnitude of every Fourier coefficient and keeps only its phase. A natural concern is that this destroys the similarity structure of the material descriptors. It does not. For two descriptors with cosine similarity ρ, the expected inner product of their unitary encodings is a strictly increasing function of ρ. Similar descriptors remain similar, and the ordering of similarities is preserved in expectation over the projection. We state this precisely below and verify it empirically for the particular projection used in our experiments in Sec. C.4.

Intuition. Because R is Gaussian, every Fourier coefficient of $R\mathbf{d}$ is a Gaussian variable. For two descriptors $\mathbf{d}$ and $\mathbf{d}'$, the coefficients at the same frequency are correlated Gaussians whose correlation equals the cosine similarity ρ of the descriptors. Unitarization compares only their phases. The expected cosine of the phase difference between two correlated Gaussians is a classical quantity: for real Gaussians it is Grothendieck's function $G_{\mathbb{R}}(\rho) = \frac{2}{\pi}\arcsin\rho$ (Sheppard, 1899), and for complex Gaussians it is Haagerup's function $G_{\mathbb{C}}(\rho) = \frac{\pi}{4}\rho\,{}_2F_1(\frac{1}{2},\frac{1}{2};2;\rho^2)$ (Haagerup, 1987; Friedland et al., 2019). Both are strictly increasing on $[-1,1]$. The unitary encoding inherits this monotonicity.

Proposition 1 (Material phase kernel)**.** *Let $\mathbf{d},\mathbf{d}' \in \mathbb{R}^{d_v}$ be nonzero descriptors with cosine similarity ρ, let $R_{ka} \sim \mathcal{N}(0, 1/d_v)$ independently, and let $D = 2m+1$ be odd. Then*

$$\mathbb{E}_R\,\langle U(R\mathbf{d}), U(R\mathbf{d}')\rangle = \frac{1}{D}\,G_{\mathbb{R}}(\rho) + \frac{D-1}{D}\,G_{\mathbb{C}}(\rho) =: G_D(\rho), \tag{16}$$

and G_D is continuous and strictly increasing on $[-1,1]$.

Proof sketch. Write $\mathbf{a} = R\mathbf{d}$ and $\mathbf{b} = R\mathbf{d}'$ with Fourier coefficients $\widehat{a}_k$ and $\widehat{b}_k$. Since R is isotropic Gaussian and the Fourier basis is orthogonal, coefficients at distinct frequencies are independent, and at each frequency k the pair $(\widehat{a}_k, \widehat{b}_k)$ is a correlated Gaussian pair with correlation ρ. The DC coefficient is real; for odd D the remaining coefficients form m conjugate pairs of proper complex Gaussians.

By Parseval's identity, the inner product of the unitarized vectors decomposes over frequencies as

$$\langle U(\mathbf{a}), U(\mathbf{b})\rangle = \frac{1}{D}\,\mathrm{sgn}(\widehat{a}_0)\,\mathrm{sgn}(\widehat{b}_0) + \frac{2}{D}\sum_{k=1}^{m}\cos\Big(\arg\widehat{a}_k - \arg\widehat{b}_k\Big), \tag{17}$$

where the factor of two accounts for each conjugate pair. Taking expectations, the DC term contributes $G_{\mathbb{R}}(\rho)/D$ by Grothendieck's identity and each of the m complex terms contributes $2G_{\mathbb{C}}(\rho)/D$ by Haagerup's identity. Summing gives Eq. 16. Monotonicity of G_D follows from that of its two components. □

Remarks. The result is a monotone random-kernel statement, not an isometry. Since $U(c\mathbf{a}) = U(\mathbf{a})$ for $c > 0$, the unitary map cannot preserve Euclidean distances; what it preserves is the ordering of angular similarities. Because Eq. 17 is an average of $m+1$ independent bounded terms, the inner product concentrates around $G_D(\rho)$ at rate $O(1/\sqrt{D})$ by Hoeffding's inequality. At $D = 3201$ this guarantee is loose, and the proposition concerns expectation over random projections whereas our tokenizer uses a single fixed draw. The empirical evaluation below therefore characterizes the projection actually used.

Correspondence with the implementation. All experiments use one fixed Gaussian projection with $d_v = 1024$ and $D = 3201$. Fourier magnitudes are lower-bounded by 10^{-8} for numerical stability; no coefficient reached this threshold on the evaluated descriptors.

C.4 Empirical Preservation of Material-Feature Similarity

Proposition 1 predicts that similarity ordering survives unitary encoding in expectation. We check that it survives for the fixed projection used throughout. Using 100,000 DINO descriptors from 3,127 assets, we compute Pearson and Spearman correlations between original and transformed cosine similarities over 1,000,000 cross-asset pairs, and exact cross-asset nearest-neighbor overlap on a fixed subset of 25,000 descriptors.

Table 7: Preservation of DINO material-feature similarity. Correlations compare transformed and original cosine similarities; kNN reports cross-asset neighborhood overlap.

Transformation	Pearson	Spearman	kNN@10	kNN@50
$R\mathbf{d}$	0.9960	0.9943	0.9309	0.9385
$U(R\mathbf{d})$	0.9920	0.9903	0.8788	0.8991

The linear projection alone preserves similarity almost exactly. Unitarization costs a further 0.004 in correlation and roughly five points of neighbor overlap, leaving Spearman correlation at 0.9903 and top-50 overlap at 89.9%. The transformation is not an isometry, as the proposition states, but the angular and neighborhood structure of the descriptors survives it.

Encoded features. Section 4.2 asks how much of this similarity survives encoding. We sample 1,000,000 pairs of distinct assets and, for each pair, compute the cosine similarity of their unitarized DINOv3 descriptors. We then encode each descriptor alone as a single material token and compute the cosine similarity of the encoder's final-layer outputs, before P_{mat}. The Spearman correlation between the two similarities is 0.485 with $\mathcal{L}_{\mathrm{mat}}$ and 0.329 without it. A perfect correlation is not expected, since the encoder is trained to predict masked context rather than to reproduce the descriptors.

D Implementation Details

Tokenization is described in Appendices B and C.

D.1 Asset Encoder

The encoder f_θ is a 24-layer Transformer with width 1024, 16 heads, and an MLP ratio of 4 (307M parameters). The predictor g_θ has 12 layers of width 384 (22M parameters). The target encoder $f_{\bar{\theta}}$ is an exponential moving average of f_θ with momentum increased linearly from 0.996 to 1.0.

Targets are selected by spatial block masking in normalized coordinates, using up to six axis-aligned boxes, four short and two long, with aspect ratios between 0.5 and 2. At each step, targets are drawn from both streams jointly with probability 0.5, in which case geometry and material at the same location are masked together, or from the geometry or material stream alone with probability 0.25 each, in which case the other stream remains visible as context.

Each step processes a batch of individual assets and a batch of composed scenes, whose losses are weighted 0.75 and 0.25. The material projection P_{mat} is linear, and $\lambda_{\mathrm{mat}} = 0.1$. We train the encoder for 150k steps, and its EMA $f_{\bar{\theta}}$ serves as the frozen representation model for composition learning.

Each 3201-dimensional token is mapped to the encoder width by a linear layer shared across streams and summed with a Fourier embedding of its surface coordinate, a learned stream embedding, and the level embedding of Sec. 3.3.

D.2 Composition Operator

Architecture. The composition operator C_ϕ has four blocks of width 1024 with 16 heads and MLP width 4096, and $L = 64$ learned latent queries. Each block applies cross-attention from the latents to the conditioned object features, followed by self-attention among the latents. Each latent query carries a bias from the Fourier encoding (16 frequencies) of one of the 4^3 voxel centers of the unit cube. The placement encoder G maps a 12-dimensional placement vector, consisting of translation, a six-dimensional rotation, and log-scale, through a two-layer MLP ($12 \to 1024 \to 2048$, SiLU) to $(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i)$, with the final layer initialized to zero so that conditioning begins as the identity. Each object thus enters as its encoder features in its own frame, conditioned only on its placement; patch coordinates in the scene frame are not supplied. The composition operator and placement encoder together comprise 103M parameters. The training-only decoder D_ϕ is a single cross-attention block of width 1024 with 16 heads, with queries formed from an 8-frequency Fourier encoding of the query location and a learned stream embedding (14M parameters). The predictor g_ϕ has six blocks of width 512 with 8 heads (20M parameters).

Targets and masking. The assembled scene is tokenized with 128 surface patches per scene element, in scene coordinates with length scale $\ell = 0.004$, whereas each object's constituent features come from its 512-patch encoding at $\ell = 0.008$. Prediction targets are the parameter-free layer-normalized scene features of the frozen encoder, detached from the computation graph. Constituent and scene features are assigned to a 16^3 voxel lattice, and a cell is matched for an object when it

contains both that object's constituent features and its scene features. Masking selects 70% of objects and, within each, 50% of its matched cells, with at least four matched cells masked and at least 30% of constituent features left visible.

Objectives. The identification objective is applied equally on the visible and masked readout paths, with temperature $\tau = 0.07$ and $\lambda_{\text{id}} = 0.1$. Its set weights w_F are 1.0 for in-scene negatives, 0.25 for same-category negatives (up to 16 per location), and 0.05 for cross-scene negatives. The language objective uses a frozen `all-mpnet-base-v2` encoder (Song et al., 2020; Reimers & Gurevych, 2019) with 768-dimensional embeddings, a linear projection P_{text} from 1024 to 768 dimensions, temperature $\tau_t = 0.07$, and $\lambda_{\text{text}} = 0.1$; caption embeddings are precomputed once.

D.3 Optimization

Table 8 summarizes training. Both stages use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.04, gradient clipping at 1.0, and BF16 mixed precision on 8 NVIDIA B200 GPUs. No EMA is used during composition learning.

Table 8: Training hyperparameters.

	Asset encoder	Composition operator
Steps	150k	110k
Effective batch	512 objects, up to 32 scenes	64 scenes
Peak learning rate	3×10^{-4}	3×10^{-4}
Schedule	5% warmup, cosine decay, 5% cooldown to 10^{-6}	constant for 100k, linear decay over 10k
Training time	406 GPU-hours	104 GPU-hours

D.4 Question Answering

The 64 latents of $Z_{\mathcal{S}}$ are mapped by a two-layer projector to the hidden width of the language model (3584 for both backbones) and inserted as soft tokens before the question. We use Qwen2.5-7B-Instruct (Yang et al., 2024) and LLaVA-OneVision-7B (Li et al., 2024); no images are supplied to either model. For SQA3D and MSNN, the agent's situation is given only as text preceding the question; no pose is supplied to the scene representation.

Adaptation proceeds in three stages. The projector is first aligned for two epochs on scene captions with the language model frozen. The projector and a LoRA adapter on the language model (rank 32, $\alpha = 64$) are then trained on our question-answer pairs and captions for up to ten epochs, stopping once validation loss improves by less than 0.5% for two consecutive epochs after a minimum of three, and finally on the ScanQA training split for two epochs. These stages use learning rates of 3×10^{-4} for the projector and 10^{-4} for LoRA, with cosine decay and a 3% linear warmup in the mixed stage, and select the checkpoint with the lowest validation loss. For SQA3D and MSNN, the model is instead fine-tuned from the selected mixed-stage checkpoint for two epochs with learning rates of 10^{-3} for the projector and 2×10^{-4} for LoRA under cosine decay. All stages use an effective batch of 192, and the composition operator is frozen throughout.

Answers are scored with each benchmark's official exact-match protocol against all reference answers. MSNN is evaluated on the ScanNet scenes of MSR3D, with answers given as one of four or eight directions.

D.5 Cost Measurement

All timings use a single NVIDIA A100-SXM4-80GB GPU with resident models and inputs, three warmup runs, twenty measured runs, and CUDA synchronization; we report medians. We use a 24-object InternScenes scene reconstructed from Matterport3D. For all methods, language-model inference, data loading, and host-to-device transfer are excluded.

For our method, encoding the 24 objects once takes 0.54 s, comprising surface and farthest-point sampling, material compositing and DINOv3 descriptors, SSP encoding, and one batched encoder forward pass (0.2 s). A new arrangement of the same objects then requires only a composer forward pass (4.6 ms). For view-based methods, 32 views are rendered with nvdiffrast at 512×512 (26 ms) and encoded in a single batched forward pass by each method's public vision encoder at its stated resolution: FG-CLIP ViT-B/16 at 224 for POMA-3D and FG-CLIP, SigLIP SO400M/14 at 384 for LLaVA-OneVision, SplatTalk, LLaVA-3D, and Video-3D LLM, and the Qwen2.5-VL vision tower at 224. For point-cloud methods, Mask3D (Schult et al., 2023) with its ScanNet200 checkpoint generates instance proposals at 2 cm voxels (352 ms), followed by PointNet++ (Qi et al., 2017) over the proposed objects; Chat-Scene is bounded below by Mask3D alone, and LEO additionally renders 32 views and encodes them with OpenCLIP ConvNeXt-Base. The ScanQA and SQA3D baselines are timed with a VoteNet (Qi et al., 2019) backbone on 40,000 points (87 ms). PointNet++ and VoteNet use the original CUDA sampling and grouping kernels. Baseline values time only these public components and exclude any further 3D encoding, projection, or reasoning stages, so they are lower bounds.

E Dataset

This section details the scene corpus, evaluation splits, and the two language corpora used for training and adaptation. The corpus, tokens, captions, and question-answer pairs will be made publicly available upon acceptance.

E.1 Scene Corpus

Scenes are sourced from InternScenes (Zhong et al., 2025), which provides indoor environments as collections of placed object assets, each with an oriented bounding box. Each asset consists of a textured mesh with physically based rendering (PBR) material maps. The corpus comprises three scene types. *Real* scenes are reconstructions of ScanNet (Dai et al., 2017) rooms, where scanned objects are replaced by matched CAD assets at their original placements; 1,513 ScanNet scenes are included. *Generated* scenes are created by the InternScenes layout generator using the same asset library. *Synthetic* scenes consist of a small set of procedurally authored rooms.

The object library comprises 374,000 assets with valid material maps, of which 338,000 are placed in at least one scene. The corpus contains 23,738 scenes; 22,718 are allocated for training and 1,020 are reserved for evaluation. As objects recur across scenes, each placed asset appears in roughly two training scenes.

E.2 Evaluation Splits

Held-out scenes. The 1,020 evaluation scenes are excluded from training of both the asset encoder and the composition operator. Objects within these scenes may also appear in training scenes, so this split evaluates generalization to novel arrangements of familiar objects.

Unseen objects. Of the objects in the evaluation scenes, 12,042 never appear in any training scene. The 670 evaluation scenes containing at least one of them form the unseen-object evaluation set. Because the asset encoder is trained on the asset library, these objects are unseen by the composition operator but not necessarily by the encoder.

Diagnostic subset. The objective ablations and the identification analysis (Appendix A.5) use a fixed subset of 256 evaluation scenes, the first 256 in a deterministic hash ordering of scene keys.

External benchmarks. ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), and MSNN (Linghu et al., 2024) are based on ScanNet scenes, all of which are included in the real subset of the corpus. Because ROSE-3D operates on objects and their placements, we evaluate on the InternScenes reconstructions of the benchmark scenes, which provide both; the baselines in Table 1 process the original scans. Benchmark coordinates are mapped using ScanNet's axis-alignment transform, followed by the per-scene normalization applied during corpus construction. The mapping is validated for each scene by confirming that SQA3D situation positions are within the room

and that specified facing directions correspond to the named landmark objects. We evaluate on the ScanQA validation split, the SQA3D test split, and the MSNN validation split, and use each benchmark's training split exclusively for adaptation. No caption or question-answer pair in our corpora is generated for a scene in any benchmark's evaluation split. The reconstructions of all benchmark scenes, including those in evaluation splits, belong to the unlabeled training corpus of the asset encoder and composition operator; none is among the 1,020 held-out scenes.

Answer preservation. Of the 4,675 ScanQA validation questions, 3,904 (84%) refer only to objects with a matched CAD asset in the reconstruction; the remaining 771 refer to at least one object absent from it. ROSE-3D answers the first group at 24.3 EM@1 and the second at 23.1 (Qwen2.5-7B), so objects missing from the reconstruction slightly lower accuracy rather than raise it.

E.3 Scene Captions

Scene captions serve two roles: they supervise the composition operator through the language objective (Sec. 3.4), and they align the question-answering projector. A single caption is generated per scene using GPT-5.4, based on rendered views of the scene and a structured record of its contents. This record enumerates each object's category and oriented bounding box, along with spatial facts derived from these boxes, such as nearest neighbors, support relations, groups of proximate same-category objects, and coarse region labels. The model is instructed to describe only objects present in the record, to express spatial relations between objects rather than using coordinates, and to organize descriptions by region. Captions are validated for comprehensive coverage of the object inventory, exclusion of objects not present in the record, and inclusion of relational content. The corpus contains 22,520 captions. Of these, 35 describe held-out evaluation scenes; they are used only to adapt the language model and never enter the composition operator's objective. The 233 training scenes without a caption receive the prediction and identification objectives but are excluded from the language objective.

E.4 Question-Answer Pairs

Human-verified set. A reference set of question-answer pairs is constructed for 200 scenes. Candidate questions and answer options are generated by Gemini 3.1 Pro using four rendered views of each scene (three angled and one top-down) and the scene's object inventory. These are presented to human evaluators in multiple-choice format with four to nine options, including an *Unsure* option for ambiguous cases. Evaluators adhere to written guidelines addressing the counting of partially occluded objects, interpretation of situated viewpoints, distance judgments from the top-down view, and visibility across views. Questions concerning color, material, or on-screen text are intentionally excluded. Pairs for which evaluators disagreed or selected Unsure are discarded. Of approximately 2,500 candidates, 1,691 pairs across 194 scenes are retained. No question from this set is used in any training or alignment stage.

Generation at scale. The human-verified set serves as a reference bank for generating question-answer pairs for the remaining scenes using Gemini 3.1 Pro, based on the same four rendered views and the scene's object inventory. Prompts are designed to elicit questions with answers verifiable from the views. Each candidate pair is independently evaluated by GPT-5.4, using the same views and inventory, and retained only if accepted. Answers are categorized as numeric, yes/no, single-token, list, or free-form.

Corpus. After removing all ScanQA validation and test scenes, the corpus comprises 199,340 pairs across 22,496 scenes. It includes 8,537 pairs for 957 held-out evaluation scenes, which, like the held-out captions, are used only to adapt the language model. Together with the captions, the corpus is divided by a deterministic 5% scene-level split into a training partition (189,308 pairs across 21,369 scenes) and a validation partition used for convergence (10,032 pairs across 1,127 scenes). Answers are distributed as follows: 69% single-token, 13% numeric, 12% free-form, 5% yes/no, and 0.3% list.