# MSR-3D: Multi-Mode Semantic Representation for Open-Vocabulary 3D Scene Understanding

**Jianan Fu**[1] **Ting Wu**[2] **Rui Yang**[3] **Jiacheng Sun**[2]
**Xinghong Zhou**[1] **Chenghai Mao**[2] **Pan Jiaxiang**[2] **Xiaomao Li**[1]*
[1]National Key Laboratory of Maritime Unmanned Swarm Intelligence, Shanghai University
[2]Shanghai University [3]Tsinghua University

## Abstract

3DGS-based open-vocabulary 3D scene understanding requires identifying reconstructed 3D instances that correspond to a natural-language query. Existing methods typically compress multi-view observations of an instance into a single semantic descriptor, which is insufficient to represent a complete 3D instance. To retain rich semantic information across diverse viewpoints, we propose **MSR-3D**, a framework built around a Multi-Mode Semantic Representation. Specifically, MSR-3D derives multi-view observations from both 2D segmentation masks and rendered masks of each segmented 3D instance. These observations are organized into groups. Each group is summarized as a semantic mode through a representative feature and associated statistical attributes. Based on the multi-mode semantic representation, we further introduce a lightweight learnable mode-aware matcher for querying. The matcher combines linear scoring with a nonlinear attention-based branch to integrate evidence from valid semantic modes into a three-dimensional score vector for each query–instance pair. Calibrated decision boundaries are then learned in the resulting score space to separate true query–instance matches from semantically related distractors. This structured and decomposable design further improves the interpretability and generalization of MSR-3D. Experiments on open-vocabulary scene-understanding benchmarks show that MSR-3D improves scene-averaged mIoU on LERF-OVS by 6.69 percentage points over the strongest compared baseline and achieves the best overall performance on 3DOVS. The source code is available at our project page `https://sparklingpumpkin.github.io/MSR-3D/`.

## 1 Introduction

Open-vocabulary 3D scene understanding aims to align text queries with objects and regions in 3D scenes. 3D Gaussian Splatting (3DGS) [1] provides a suitable scene representation for this task by combining high rendering quality, fast optimization, and explicit primitives. Existing methods usually accomplish this task through instance-level matching. In this formulation, a reconstructed scene is first decomposed into a set of candidate 3D instances, each represented by a subset of Gaussian primitives. Given a text query, the model then estimates which candidate instances are semantically matched to the query.

As illustrated in Fig. 1, existing instance-level matching paradigms typically compress multi-view observations associated with each 3D instance into a single semantic descriptor [2–6]. The descriptor yields a single query-instance similarity score for each candidate instance. Final selection then relies on thresholding this score [5] or retaining the highest-scoring instance [6].

*Corresponding author.

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

Existing methods often struggle to achieve stable instance-level matching in complex 3D scenes. The visible RGB evidence of a 3D instance varies substantially across viewpoints and is often partially missing in occluded views. Such cross-view inconsistency introduces noise, but also encodes informative cues, including view-dependent visibility, occlusion patterns, observation completeness, and semantic stability. **These cues constitute a key distinction between 3D instances and their 2D counterparts.** Therefore, constructing an effective 3D semantic representation is critical for stable instance-level matching. However, existing methods often overlook this distinction by distilling 2D observations into a single semantic descriptor [2–6], which discards structured cross-view cues important for instance-level matching (Fig. 3). Fully retaining all 2D observations [7] is also undesirable, as it incurs substantial storage and matching costs. Existing multi-descriptor matching [8, 9] retains multiple feature clusters, but the clusters are not sufficiently characterized with auxiliary or instance-level descriptive evidence, and the final query decision still collapses to the single descriptor with the strongest response. Based on the 3D semantic representations, query precision is further limited at two stages. Before the representation is constructed, relying only on semantic features extracted from 2D segmentation masks [2] is prone to introducing occlusion bias into the input evidence, leading to biased aggregated semantics for a 3D instance. After each instance representation is scored against a query, strategies that threshold a single similarity score [5] remain brittle in complex scenes and often admit irrelevant instances, resulting in low precision for text-based queries. Conversely, top-1 selection [2, 6], which keeps only the highest-scoring instance for each query, fails to recover multiple valid instances when the query refers to more than one object.

To address these limitations, we propose **MSR-3D**, a **multi-mode semantic representation for open-vocabulary 3D scene understanding**. Specifically, MSR-3D first constructs a **raw observation pool** that collects multi-view semantic evidence before compression. The pool contains observations from 2D segmentation masks and rendered masks of segmented 3D instances, where the rendered masks add complementary evidence from the reconstructed 3D instance. Subsequently, the **multi-mode semantic representation** is derived from the raw observation pool by adaptively partitioning the observations of each 3D instance into multiple semantic modes. Each mode is summarized by a prototype feature, with prototype-level attributes describing its reliability. Instance-level attributes further summarize the overall evidence of the 3D instance. In the matching stage, the **mode-aware matcher** formulates a three-dimensional score vector from the text query and multi-mode representation. Optimized via supervised learning, these scores are trained to exhibit high separability between positive and negative candidates. An adaptive decision surface then delineates the optimal boundary within this distribution to isolate true matches from distractors. As shown in Fig. 1, MSR-3D integrates evidence from all valid semantic modes to score each instance, rather than relying solely on the strongest descriptor response. MSR-3D therefore balances compression efficiency with matching stability, while its explicit statistical attributes and decomposed scoring design support interpretability and generalization.

**The technical contributions of this paper are summarized as follows:**

- We propose **MSR-3D**, a general semantic representation for 3D instances that compactly preserves multi-view 2D observations as multiple semantic modes.
- We enrich the **raw observation pool** with rendered mask observations from segmented 3D instances, providing more complete semantic evidence for 3D instance representation.
- We design a **mode-aware semantic matcher** that integrates evidence from valid semantic modes. Its decomposed scoring design and flexible decision boundaries improve the stability, interpretability, and generalizability of open-vocabulary semantic querying.

## 2 Related work

### 2.1 2D-guided 3D semantic representations

Early open-vocabulary 3D methods lifted 2D vision-language features into neural fields [10] and point representations [11]. In the 3DGS setting, several methods [12–15] store language or foundation model features in Gaussian scenes. Other methods move toward structured semantic units. PanoGS [16] learns a continuous semantic field and derives panoptic structure. Recent instance-level methods [17–20] construct descriptors from selected views or attach language features to Gaussian object sets. Beyond Averages [7] keeps a full multi-view embedding bag instead of one averaged

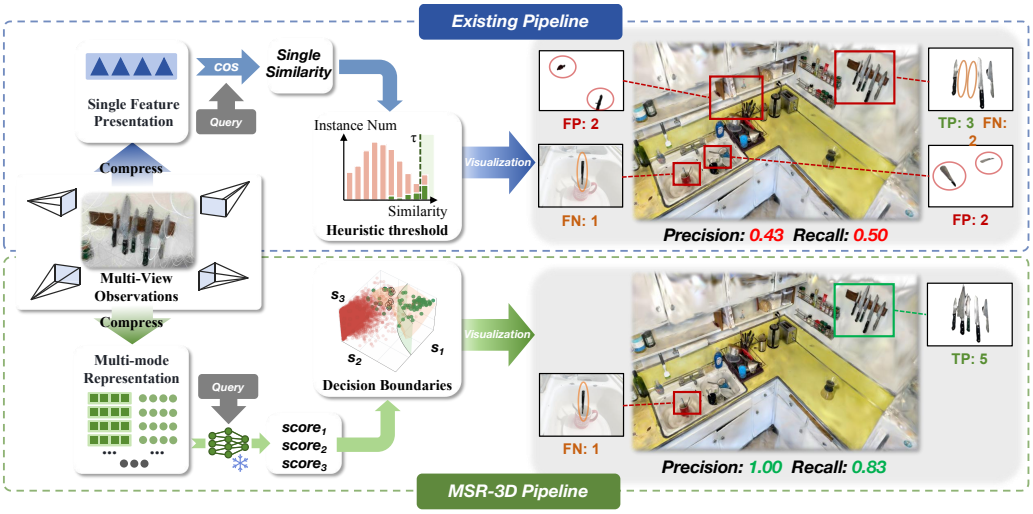


Figure 1: **Comparison of instance-level matching pipelines and outputs.** The left part compares the instance-level matching pipelines, while the right part visualizes the 3D responses to the query "knife" in the scene. The visualization results are obtained from experiments with OpenSplat3D [5].

object descriptor. LAGA [9] keeps multiple feature cluster centers for view-dependent semantics, but its query rule still relies on the strongest responding center. Existing 3D semantic carriers still lack a unified and efficient instance representation that preserves multiple semantic modes.

### 2.2 Multi-view semantic aggregation under inconsistency

Regardless of the carrier, multi-view inconsistency remains the central difficulty. It arises from view-dependent masks, occlusion, alpha compositing, and local crops that miss contextual cues. Existing methods reduce inconsistency before fusion by regularizing 2D supervision [21], aligning cross-view masks [22], or enforcing object-level identity consistency [23]. Context-aware Gaussian propagation further stabilizes semantic features with local neighborhood cues [24]. Other methods make the fusion rule more robust through rendering-weighted aggregation [25], visibility-aware gating and geometric-median aggregation [26], or confidence estimates from observation statistics [16]. At the instance level, related methods select informative views [5], fuse cross-view context [3], or preserve multi-granularity information during semantic Gaussian construction [27, 28]. These methods substantially reduce noise, but they mainly treat inconsistency as an error to suppress. The structured 3D information carried by view-dependent observations therefore remains underused.

### 2.3 Text querying of 3D semantic representations

Given a 3D semantic representation, text querying requires a decision rule that converts query responses into final region or instance selections. LERF [10] and LangSplat [12] follow a relevance-rendering paradigm, where the final output is obtained by thresholding rendered query responses. At the instance level, several methods score candidate objects and apply fixed-threshold selection to scalar responses [4, 5]. Although LAGA [9] retains multiple weighted similarity responses, the final selection still applies the same rule to the strongest response. Other methods make the threshold data-dependent, either through visibility- or aggregation-aware rules [25, 26] or through adaptive thresholding for object-level querying [29]. Another line relies on ranking, retaining only the strongest candidate [6, 30, 18] or expanding from this candidate to nearby high-response instances [2, 24, 3]. Beyond Averages [7] departs from single-descriptor matching by retaining an object-level embedding bag and performing query-conditioned matching over the bag. GOI [31] optimizes a semantic-space hyperplane for each query using target-scene pseudo-masks, whereas our matcher is trained with source-scene supervision and transferred to unseen scenes with fixed parameters. Overall, existing query rules underuse structured 3D instance representations, with most final decisions based on a single similarity response.

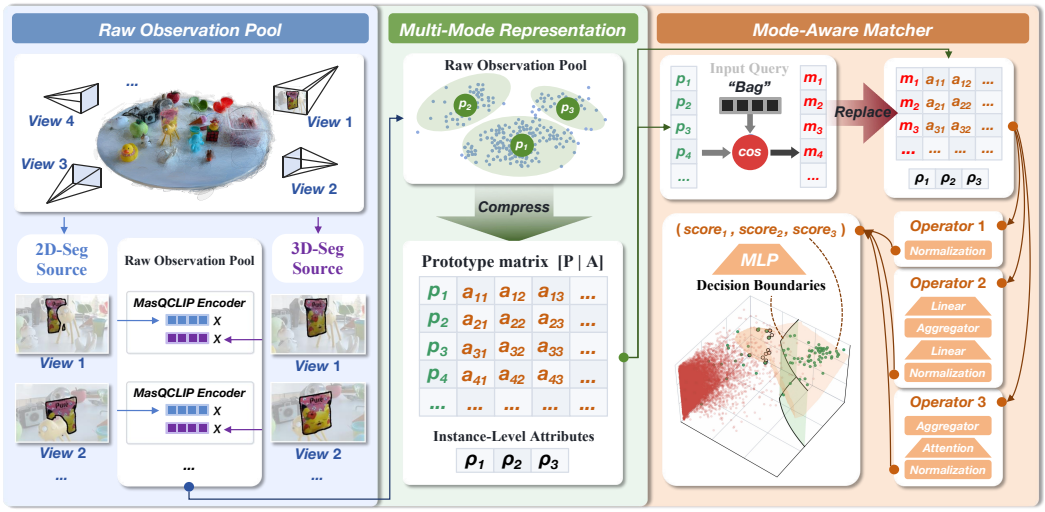


Figure 2: **Overview of MSR-3D.** (a) For each candidate 3D instance, we extract 2D-segmentation and 3D-segmentation observations from posed multi-view images, then organize them into a raw observation pool with geometric quality cues. (b) The raw observation pool is lifted into a compact multi-mode representation. Representative prototypes encode semantic modes, while prototype-level and instance-level statistical attributes summarize their observation patterns. (c) For each query-instance pair, we compute a three-dimensional semantic score, which defines one point in a three-dimensional score space, where decision boundaries separate positives from negatives.

# 3 Method

As illustrated in Fig. 2, our method proceeds in three stages. It first enriches the input evidence of each candidate 3D instance with a raw observation pool (Sec. 3.1), then constructs a compact multi-mode semantic representation from this pool (Sec. 3.2), and finally performs open-vocabulary 3D scene understanding with a mode-aware matcher (Sec. 3.3).

Given a candidate-instance set $\mathcal{I} = \{i_1, \ldots, i_M\}$ and a natural-language query $Q$, the goal is to predict the positive instance subset $\hat{\mathcal{Y}}(Q) \subseteq \mathcal{I}$. Here $i$ denotes a candidate 3D instance, $M$ the number of candidate instances in the scene, and $q \in \mathbb{R}^d$ the text feature of the query $Q$, where $d$ is the feature dimension.

## 3.1 Raw observation pool

For each candidate instance $i$, we consider its available views $v \in \{1, \ldots, V^i\}$. In each view, we construct two mask-conditioned observations, each specified by a source-specific mask and the cropped RGB context around the instance (Fig. 2, left). *2D-segmentation observations* are conditioned on masks predicted from the original RGB images by SAM [32]. *3D-segmentation observations* are conditioned on rendered masks obtained by projecting the segmented 3D instance [5] into the corresponding views. Let $\mathcal{M}^i_{v,s}$ denote the source-specific mask in view $v$, where $\mathcal{M}^i_{v,\mathrm{2D}}$ is the mask predicted by the 2D segmentation model and $\mathcal{M}^i_{v,\mathrm{3D}}$ is the rendered mask of the segmented 3D instance. Both observations are encoded from the same cropped RGB context $I^i_v$ by a shared mask-conditioned encoder [33] $E(\cdot,\cdot)$. The encoded semantic feature from source $s$ is:

$$x^i_{v,s} = E(I^i_v, \mathcal{M}^i_{v,s}), \qquad s \in \{\mathrm{2D}, \mathrm{3D}\}. \tag{1}$$

Instead of directly collapsing these view-wise features into a single instance descriptor, MSR-3D retains them as a raw observation pool.

The two sources are complementary. 2D-segmentation observations retain more direct object evidence, but are more sensitive to occlusion and mask quality. 3D-segmentation observations are less affected by real-image occlusion and tend to stabilize semantics across views, but are prone to drift when the 3D reconstruction is inaccurate.

For instance $i$, the raw observation pool is written as $\mathcal{U}^i = \{x_n^i\}_{n=1}^{N^i}$, where $n$ indexes the retained view-source pairs $(v, s)$, and $x_n^i = x_{v,s}^i \in \mathbb{R}^d$.

### 3.2 Multi-mode semantic representation

Given the raw observation pool $\mathcal{U}^i$, we construct a small set of *prototypes* that lift each instance into a compact multi-mode representation (Fig. 2, middle). Each prototype summarizes a subset of $\mathcal{U}^i$ whose features are close in feature space, thereby representing a shared semantic mode. Features within the same subset often correspond to RGB views captured from nearby camera poses [9] (Fig. 3).

Prototypes are constructed by three query-independent selection rules and remain fixed during matching. The prototype $p_k^i$ denotes the $k$-th selected prototype of instance $i$. The index $j_t^i \in \{1, \ldots, N^i\}$ denotes the feature selected at step $t$.

The first group forms a stable reference prototype from the $\kappa$ features with the largest geometric-completeness scores, selected without replacement:

$$j_t^i = \arg\max_j c_j^i, \quad p_1^i = \kappa^{-1} \sum_{t=1}^{\kappa} x_{j_t^i}^i, \quad t = 1, \ldots, \kappa. \tag{2}$$

Here $c_n^i$ is the geometric-completeness score, which combines instance visibility with the relative rendered-mask area in the corresponding view.

The second group selects features that cover underrepresented regions of the raw observation pool:

$$j_t^i = \arg\max_j (\Delta_j - \lambda r_j), \qquad p_t^i = x_{j_t^i}^i. \tag{3}$$

Here $\Delta_j$ is the source-balanced coverage gain: positive increases in raw-pool observations' maximum cosine similarity after adding $x_j^i$ are averaged within each source and then equally across non-empty sources. The redundancy term $r_j = \max_{p \in C_{t-1}} \cos(x_j^i, p)$ is the maximum similarity to the prototype set $C_{t-1}$ selected before step $t$, with penalty strength controlled by $\lambda$.

The third group selects additional features that are far from the selected prototypes:

$$j_t^i = \arg\max_j d_j^{\min}, \qquad p_t^i = x_{j_t^i}^i. \tag{4}$$

Here $d_j^{\min}$ is the distance from $x_j^i$ to its nearest selected prototype.

Fig. 3 illustrates this representation with a PCA projection of the feature distribution for a representative instance. The query feature, MSR-3D prototypes, and the single instance feature used by OpenSplat3D [5] are projected into the same embedding space. MSR-3D covers more latent semantic clusters within the raw observation pool.

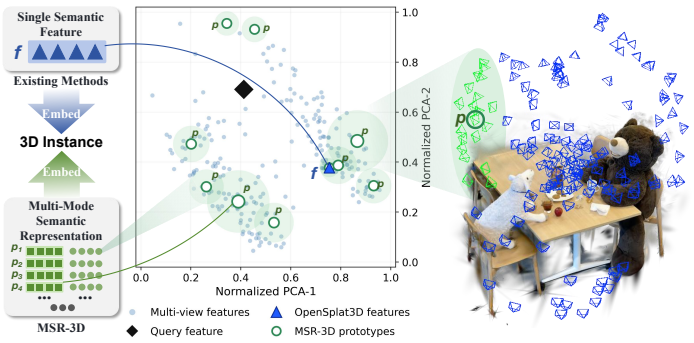


Figure 3: **Visualization of MSR-3D semantic prototypes versus the single semantic feature used by OpenSplat3D.** All plotted elements are PCA projections of features derived from one instance in LERF-OVS dataset. For MSR-3D, green and black dots denote prototype-level and instance-level attributes, respectively.

We finally formalize the lifted **Multi-Mode Semantic Representation** as two matrices, $\mathcal{P}^i = (\mathbf{Z}^i, \mathbf{S}^i)$. Here $\mathbf{Z}^i = \left[\mathbf{P}^i\, \mathbf{A}^i\right]$ is the prototype matrix (Fig. 2, middle), where $\mathbf{P}^i \in \mathbb{R}^{K^i \times d}$ and $\mathbf{A}^i \in \mathbb{R}^{K^i \times L^i}$. The first block $\mathbf{p}_k^i$ stores the $k$-th prototype, while the remaining block $\mathbf{a}_k^i$ stores statistical attributes associated with that prototype, such as coverage, support from the 2D-segmentation and 3D-segmentation sources, and local statistics. These statistical attributes help downstream operators judge the quality and representativeness of each mode. In parallel, $\mathbf{S}^i$ is the instance-level statistical attribute matrix, which retains only a single row of instance summaries, namely $\mathbf{S}^i = \boldsymbol{\rho}^i \in \mathbb{R}^{1\times 3}$. Here $\boldsymbol{\rho}^i$ summarizes instance-level evidence, such as agreement between the 2D-segmentation and 3D-segmentation sources and overall coverage structure. Together, the retained prototype-level statistical attributes and instance-level summaries, selected from a larger candidate set through correlation checks and ablations, expose observation patterns through explicit statistics and allow the downstream matcher to adapt to different instance conditions.

### 3.3 Mode-aware matcher

Based on the multi-mode semantic representation, the matcher computes three calibrated semantic scores and separates positives from negatives with flexible decision boundaries (Fig. 2, right).

Given a query feature $q$, the mode-aware matcher computes its cosine similarity to all prototypes of each candidate instance $i$, yielding $m_k^i(q) = \cos(q, \mathbf{p}_k^i)$ for $k = 1, \ldots, K^i$. The resulting similarity vector is:

$$\boldsymbol{\mu}^i(q) = [m_1^i(q), \ldots, m_{K^i}^i(q)]^\top \in \mathbb{R}^{K^i}. \tag{5}$$

The vector $\boldsymbol{\mu}^i(q)$ stores the similarities between the query and all prototypes of instance $i$. As shown in Fig. 2, the matcher uses $\boldsymbol{\mu}^i(q)$ in place of the prototype feature block $\mathbf{P}^i$, yielding a query-conditioned representation $\mathcal{P}_\mu^i$ without modifying the prototypes. The three score operators are written as:

$$s_b^i(q) = \Phi_{b,\theta_b}(\mathcal{P}_\mu^i\big|_{\Omega_b}), \qquad b \in \{1, 2, 3\}, \tag{6}$$

where $\Phi_{b,\theta_b}$ maps the selected input to a scalar score, with parameters $\theta_b$, and $\Omega_b$ specifies the prototype evidence and statistical attributes included in $\mathcal{P}_\mu^i\big|_{\Omega_b}$ for branch $b$. These three implementations correspond to the three score-operator submodules on the right side of Fig. 2. The three scores differ mainly in the subset they operate on and the semantic role they play.

Specifically, $s_1$ captures a mode from the first group prototype through direct similarity with the query, serving as a geometrically reliable semantic anchor. $s_2$ captures modes from the second group prototypes through two linear modules and an aggregator. It models the dominant separation pattern between positive and negative instances. $s_3$ combines prototype-similarity statistics with attribute-conditioned row attention to produce a bounded nonlinear correction for semantically ambiguous candidates. Binary query-instance labels are initialized by matching rendered masks to ground-truth masks with IoU $> 0.1$ and manually verified and corrected as needed. During training of the $s_2$ head, these labels supervise losses that encourage scores for positive and negative pairs to lie on their respective sides of an initial threshold. For samples close to the threshold, an additional margin constraint is imposed to drive ambiguous pairs further away from the decision boundary. The $s_3$ branch is trained separately with query-level supervision on the benefit of adding an instance to the base selection. The two linear readouts of $s_2$ admit sign-based analysis, whereas $s_3$ models nonlinear interactions between semantic responses and statistical attributes.

Together, the normalized $s_1$ and $s_2$ responses and the bounded $s_3$ correction form a three-dimensional score vector $\mathbf{S}^i(q) = (s_1^i,\, s_2^i,\, s_3^i)$. The matcher combines these scores to define decision boundaries that separate positive and negative instances. An intuitive geometric illustration is provided by the score-space schematic on the left side of the matcher block in Fig. 2. Specifically, the primary MLP learns a residual correction to a base boundary in the $(s_1^i,\, s_2^i)$ score plane, and a shared two-score supplementary classifier adds candidates from a small unselected set. The $s_3$ correction then modifies a continuous margin consistent with this base selection. Additional instances with a positive corrected margin are selected, while every base-selected instance is retained. Rendering and combining the masks of the final selected instances yields the query response.

Table 1: Scene-averaged results (%) on LERF-OVS and 3DOVS.

| Method | LERF-OVS (Full) | | | LERF-OVS (Unseen) | | | 3DOVS | | |
|---|---|---|---|---|---|---|---|---|---|
| | $mIoU$ | $mAcc$ | $mF1$ | $mIoU$ | $mAcc$ | $mF1$ | $mIoU$ | $mAcc$ | $mF1$ |
| OpenGaussian [2]$^\star$ | 55.08 | 75.20 | 62.82 | 57.83 | 76.20 | 64.67 | 73.14 | 85.48 | 77.11 |
| Dr.Splat [4]$^\star$ | 48.79 | 71.63 | 54.58 | 51.13 | 71.41 | 59.80 | 75.40 | 86.90 | 76.84 |
| LAGA [9]$^\star$ | 54.74 | 73.61 | 62.02 | 57.91 | 78.21 | 65.63 | 75.56 | 88.57 | 77.52 |
| OpenInsGaussian [3]$^\star$ | 54.63 | 74.01 | 61.25 | 56.57 | 74.53 | 63.10 | 73.85 | 87.14 | 78.28 |
| Segment-then-Splat [6]$^\dagger$ | 50.74 | 64.88 | 58.93 | 56.29 | 71.75 | 66.19 | 76.12 | 90.48 | 81.11 |
| CAGS [24]$^\star$ | 55.01 | 72.82 | 61.19 | 56.57 | 74.53 | 63.10 | 73.93 | 87.14 | 78.00 |
| VALA [26] | 54.38 | 75.79 | 52.92 | 56.96 | 77.31 | 59.55 | 76.39 | 88.57 | 79.10 |
| OpenSplat3D [5] | 58.01 | 75.20 | 61.07 | 59.69 | 76.20 | 65.68 | 79.41 | 93.33 | 80.52 |
| Ours | 64.70 | 85.52 | 72.63 | 63.15 | 83.21 | 72.83 | 80.98 | 95.00 | 83.02 |

*Note.* Red and yellow highlights mark the best and second-best values, respectively. † denotes single-select evaluation, and ⋆ denotes a locally tuned variant beyond the native scoring or selection rule.

# 4 Experiments

## 4.1 Experimental setup

**Datasets.** Following prior work [2, 5, 6], we evaluate MSR-3D on two widely used indoor open-vocabulary benchmarks, LERF-OVS and 3DOVS [34], using both quantitative and qualitative results. Our evaluation includes within-dataset comparison and fixed-parameter cross-dataset transfer. Table 2 reports results for all 67 queries across Figurines, Ramen, Teatime, and Waldo Kitchen. Supervision is restricted to the 18 query strings from *waldo_kitchen*. Of the 49 queries from the remaining three scenes, 45 are exact-string-unseen with respect to the 18 supervised query strings from Waldo Kitchen. All 61 evaluation queries across the 10 scenes of 3DOVS are exact-string-unseen relative to the same 18 supervised query strings. We evaluate them with the same fixed learned parameters, without 3DOVS supervision or target-specific tuning. This protocol evaluates model performance on the supervised scene and generalization to other scenes and datasets. The benchmark covers diverse conditions, including clear or ambiguous 2D evidence, well-observed or occluded instances, and substantial variation in object size, contextual dependence, and segmentation quality.

**Metrics.** Following the OpenGaussian [2] protocol, we report $mIoU$ and $mAcc$, where $mAcc$ denotes Acc@0.25 on rendered query responses. For each query, rendered-mask IoU is averaged over its annotated evaluation frames, and Acc@0.25 is determined from the resulting query mean IoU. These metrics mainly reflect visible target regions. We also report $mF1$ to evaluate agreement between predicted and ground-truth instance sets across the scene.

**Implementation details.** We compare MSR-3D with recent 3DGS-based baselines under a unified protocol. To isolate the effects of representation and matching, all methods are evaluated on the same OpenSplat3D [5] reconstruction and instance segmentation. The scoring and selection procedures of the compared methods are ported from their released implementations to this shared representation, with light tuning only when direct transfer is clearly unstable. This shared representation provides a strong instance prior for the baselines, making the comparison conservative for MSR-3D. All experiments are conducted on four NVIDIA RTX 4090 GPUs, each with 24GB memory.

## 4.2 Comparative experiments

**Overall comparison.** We aggregate results by averaging query-level metrics within each scene and then averaging the scenes equally. Table 1 reports all 67 LERF-OVS queries, including the supervised Waldo scene, the 45 exact-string-unseen queries from the other three scenes, and all 61 3DOVS queries. MSR-3D improves $mIoU$ over the strongest compared baseline by 6.69 and 3.46 percentage points on the 67- and 45-query LERF-OVS sets, respectively. On 3DOVS, it attains the highest $mIoU$, $mAcc$, and $mF1$.

**Per-scene comparison.** We then compare MSR-3D with representative baselines across all 67 queries in the four LERF-OVS scenes in Table 2. Per-scene metrics average the query-level results within each scene. MSR-3D improves $mIoU$ on the supervised Waldo Kitchen scene by 17.10

Table 2: Quantitative results (%) on the four scenes of the LERF-OVS dataset.

| Method | Figurines | | | Ramen | | | Teatime | | | Waldo Kitchen | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $mIoU$ | $mAcc$ | $mF1$ | $mIoU$ | $mAcc$ | $mF1$ | $mIoU$ | $mAcc$ | $mF1$ | $mIoU$ | $mAcc$ | $mF1$ |
| OpenGaussian [2]$^\star$ | 58.62 | 85.71 | 67.91 | 48.30 | 64.29 | 53.33 | 64.34 | 78.57 | 74.76 | 49.08 | 72.22 | 55.28 |
| Dr.Splat [4]$^\star$ | 48.97 | 71.43 | 62.22 | 46.98 | 71.43 | 56.43 | 53.82 | 71.43 | 60.20 | 45.37 | 72.22 | 39.46 |
| LAGA [9]$^\star$ | 53.92 | 76.19 | 61.18 | 53.18 | 71.43 | 69.52 | 66.05 | 85.71 | 73.57 | 45.79 | 61.11 | 43.78 |
| OpenInsGaussian [3]$^\star$ | 55.14 | 80.95 | 64.36 | 48.44 | 64.29 | 51.90 | 64.34 | 78.57 | 74.76 | 50.61 | 72.22 | 53.98 |
| Segment-then-Splat [6]$^\dagger$ | 66.71 | 80.95 | 72.22 | 35.60 | 50.00 | 47.62 | 60.54 | 78.57 | 72.62 | 40.11 | 50.00 | 43.25 |
| CAGS [24]$^\star$ | 53.99 | 76.19 | 63.56 | 48.44 | 64.29 | 51.90 | 64.34 | 78.57 | 74.76 | 53.27 | 72.22 | 54.54 |
| VALA [26] | 58.50 | 80.95 | 68.57 | 47.65 | 71.43 | 47.54 | 63.59 | 78.57 | 56.54 | 47.77 | 72.22 | 39.03 |
| OpenSplat3D [5] | 65.71 | 85.71 | 76.03 | 45.58 | 64.29 | 52.62 | 67.67 | 78.57 | 70.00 | 53.08 | 72.22 | 45.63 |
| Ours | 68.52 | 90.48 | 75.49 | 48.87 | 71.43 | 69.05 | 71.06 | 85.71 | 77.62 | 70.37 | 94.44 | 68.38 |

percentage points over the strongest baseline and achieves competitive performance when transferred to the other scenes.

**Comparison across candidate quality.** Table 3 compares MSR-3D with OpenSplat3D direct matching across upstream candidate-quality bins. For each query, candidate quality is the highest recorded mask IoU between an annotated positive candidate instance and the query ground truth, with queries lacking such candidates reported separately. On LERF-OVS, MSR-3D improves response IoU by 14.72 and 6.16 percentage points in the two intermediate bins, indicating more effective use of imperfect candidates. On 3DOVS, 47 of 61 queries fall in the highest-quality bin, where MSR-3D achieves 89.52% response IoU versus 87.95% for OpenSplat3D, while only two queries have quality below 0.50. This distribution is consistent with the smaller overall gain on 3DOVS. Bin definitions and aggregation details are given in Appendix C.3.2.

Table 3: Response IoU (%) stratified by upstream candidate quality.

| Candidate quality / status | LERF-OVS | | | 3DOVS | | |
|---|---|---|---|---|---|---|
| | Queries | MSR-3D | OpenSplat3D | Queries | MSR-3D | OpenSplat3D |
| $[0, 0.25)$ | 2 | 25.39 | 25.39 | 1 | 11.17 | 11.17 |
| $[0.25, 0.50)$ | 14 | **50.67** | 35.95 | 1 | **47.18** | 31.76 |
| $[0.50, 0.75)$ | 18 | **56.13** | 49.97 | 12 | **56.70** | 56.55 |
| $[0.75, 1]$ | 31 | **86.62** | 82.11 | 47 | **89.52** | 87.95 |
| No positive correspondence | 2 | **1.86** | 0.92 | 0 | – | – |

Fig. 4 illustrates how different matching rules fail in 3D scenes. Single-score thresholding introduces false positives when semantically related but incorrect instances receive high similarity scores, whereas top-1 single-select methods often miss valid instances in multi-target queries, such as "knife". Existing multi-select baselines inherit this limitation because their decisions are driven by a single aggregated semantic score. In contrast, MSR-3D combines multi-source semantic evidence, multi-mode instance representations, and an adaptive decision boundary, leading to more balanced control of false positives and false negatives. This supports the central claim that robust instance-level matching requires structured evidence and calibrated decision boundaries rather than a single similarity score.

### 4.3 Ablation studies

#### 4.3.1 Observation pool ablation

We ablate the raw observation pool on LERF-OVS using all 67 queries. In Table 4, "2D Seg" and "3D Seg" indicate whether observations from the corresponding source are retained. For each single-source setting, we retrain the downstream matcher on the retained source while keeping the semantic frontends and $s_1$ fixed; the dual-source row reports the original full model as a reference. The two single-source settings yield similar $mF1$, whereas the full dual-source model achieves a higher $mF1$. These results help substantiate the contribution of dual-source evidence to representation stability.

#### 4.3.2 Prototype-matrix ablation

We ablate the components of the lifted prototype representation to assess whether structured statistical attributes improve matching beyond prototype similarity alone. As shown in Table 5, prototype




Figure 4: **Qualitative comparison between our method and representative open-vocabulary 3DGS scene-understanding methods.** Each row corresponds to one text query. The leftmost column shows the reference image and the positive target region, and the remaining columns visualize the predictions of different methods. Green contours denote true positives, orange contours denote false negatives, and red contours denote false positives. Precision and recall for each query are reported below the corresponding panel.

similarity provides the main semantic signal. The results suggest that combining prototype-level and instance-level statistical attributes supports better generalization than using either alone. This indicates that the statistical attributes expose observation quality and support patterns that are not captured by semantic similarity alone.

Table 4: Ablation results (%) of the raw observation pool on LERF-OVS.

| 2D Seg | 3D Seg | $mIoU$ | $mAcc$ | $mF1$ |
|---|---|---|---|---|
| ✓ | – | 64.52 | **88.69** | 71.59 |
| – | ✓ | 61.03 | 81.55 | 71.10 |
| ✓ | ✓ | **64.70** | 85.52 | **72.63** |

*Abbr.* 2D/3D indicate whether observations from the corresponding segmentation source are retained.

Table 5: Ablation results (%) of the multi-mode semantic representation.

| Proto sim. | Proto attr. | Inst. attr. | $mIoU$ | $mAcc$ | $mF1$ |
|---|---|---|---|---|---|
| ✓ | – | – | 59.99 | 81.55 | 72.31 |
| ✓ | ✓ | – | 60.44 | 81.55 | 69.30 |
| ✓ | – | ✓ | 60.20 | 80.95 | 67.92 |
| ✓ | ✓ | ✓ | **64.70** | **85.52** | **72.63** |

*Abbr.* sim.=similarity, Proto=prototype, Inst.=instance, attr.=attributes.

### 4.3.3 Multi-dimensional semantic score ablation

We ablate the three calibrated semantic scores to verify that they provide complementary evidence rather than redundant similarity estimates. Because each score is computed from a different subset of prototype evidence, this experiment also serves as an indirect ablation of the number of prototypes used for matching. Table 6 shows that $s_1$ obtains higher $mIoU$ but lower $mF1$. This gap is consistent with the metric distinction described above. In contrast, $s_2$ provides a complementary instance-level view and already yields a higher $mF1$ when used alone, while combining $s_1$ and $s_2$ further improves

the query results. $s_3$ is not designed to operate independently. Together with the other scores, $s_3$ supplements positive instances with higher semantic ambiguity that tend to receive low responses from both $s_1$ and $s_2$.

Table 6: Ablation results (%) of the three-score composition.

| $s_1$ | $s_2$ | $s_3$ | $mIoU$ | $mAcc$ | $mF1$ |
|---|---|---|---|---|---|
| ✓ | – | – | 59.69 | 77.98 | 59.39 |
| – | ✓ | – | 53.23 | 74.60 | 64.32 |
| ✓ | ✓ | – | 63.49 | 84.13 | 72.61 |
| ✓ | – | ✓ | 60.74 | 82.14 | 62.43 |
| – | ✓ | ✓ | 53.71 | 77.38 | 61.88 |
| ✓ | ✓ | ✓ | **64.70** | **85.52** | **72.63** |

*Abbr.* $s_1$, $s_2$, and $s_3$ denote the three calibrated semantic scores used by the matcher.

## 5 Conclusion

This paper presents **MSR-3D**, a **Multi-Mode Semantic Representation** for 3D scene understanding under open vocabulary queries. MSR-3D preserves multiple semantic modes for each candidate instance. It organizes observations collected across viewpoints and from both 2D segmentation and rendered 3D segmentation sources into compact prototypes with auxiliary statistical attributes. Based on this representation, the matcher integrates evidence from all valid semantic modes to distinguish matched instances from distractors. Experiments on LERF-OVS and 3DOVS validate the effectiveness of MSR-3D. MSR-3D achieves the best overall performance and improves scene-averaged $mIoU$ on LERF-OVS by 6.69 percentage points over the strongest compared baseline. These results indicate that MSR-3D provides a compact, interpretable, and generalizable interface for open-vocabulary 3D instance matching. More broadly, this work suggests that multi-mode instance representations are worth further exploration for 3D semantic reasoning beyond instance matching.

## Acknowledgments and Disclosure of Funding

This work was supported by the National Science Fund for Distinguished Young Scholars of China (Grant No. 62225308). We thank the anonymous reviewers for their constructive comments and suggestions, which helped improve the manuscript. The authors declare no competing interests.

## References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 7 2023.

[2] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. OpenGaussian: Towards point-level 3D Gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024.

[3] Tianyu Huang, Runnan Chen, Dongting Hu, Fengming Huang, Mingming Gong, and Tongliang Liu. OpenInsGaussian: Open-vocabulary instance Gaussian segmentation with context-aware cross-view fusion. *arXiv preprint arXiv:2510.18253*, 2025.

[4] Jun-Seong Kim, GeonU Kim, Yu-Ji Kim, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. Splat: Directly referring 3D Gaussian splatting via direct language embedding registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14137–14146, 2025.

[5] Jens Piekenbrinck, Christian Schmidt, Alexander Hermans, Narunas Vaskevicius, Timm Linder, and Bastian Leibe. OpenSplat3D: Open-vocabulary 3D instance segmentation using Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 5285–5294, 2025.

[6] Yiren Lu, Yunlai Zhou, Yiran Qiao, Chaoda Song, Tuo Liang, Jing Ma, Huan Wang, and Yu Yin. Segment then Splat: Unified 3D open-vocabulary segmentation via Gaussian splatting. In *Advances in Neural Information Processing Systems*, 2025.

[7] Abdalla Arafa and Didier Stricker. Beyond averages: Open-vocabulary 3D scene understanding with Gaussian splatting and bag of embeddings. *arXiv preprint arXiv:2509.12938*, 2025.

[8] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. OVIR-3D: Open-vocabulary 3D instance retrieval without training on 3D data. In *7th Annual Conference on Robot Learning*, 2023.

[9] Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Tackling view-dependent semantics in 3D language Gaussian splatting. In *Proceedings of the International Conference on Machine Learning*, 2025.

[10] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[11] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023.

[12] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D language Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[13] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3DGS: Supercharging 3D Gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[14] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3D Gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024.

[15] Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. LangSplatV2: High-dimensional 3D language Gaussian splatting with 450+ FPS. *Advances in Neural Information Processing Systems*, 2025.

[16] Hongjia Zhai, Hai Li, Zhenzhe Li, Xiaokun Pan, Yijia He, and Guofeng Zhang. PanoGS: Gaussian-based panoptic segmentation for 3D open vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14114–14124, 2025.

[17] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. ObjectGS: Object-aware scene reconstruction and scene understanding via Gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8350–8360, 2025.

[18] Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. InstanceGaussian: Appearance-semantic joint Gaussian representation for 3D instance-level perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14078–14088, 2025.

[19] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Hendrik P. A. Lensch, Nassir Navab, and Federico Tombari. SuperGSeg: Open-vocabulary 3D segmentation with structured Super-Gaussians. In *Proceedings of the International Conference on 3D Vision*, 2026.

[20] Elena Alegret, Kunyi Li, Sen Wang, Siyun Liang, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. GALA: Guided attention with language alignment for open vocabulary Gaussian splatting. In *Proceedings of the International Conference on 3D Vision*, 2026.

[21] Yuning Peng, Haiping Wang, Yuan Liu, Chenglu Wen, Zhen Dong, and Bisheng Yang. GAGS: Granularity-aware feature distillation for language Gaussian splatting. *arXiv preprint arXiv:2412.13654*, 2024.

[22] Lei Tian, Xiaomin Li, Liqian Ma, Hao Yin, Zirui Zheng, Hefei Huang, Taiqing Li, Huchuan Lu, and Xu Jia. CCL-LGS: Contrastive codebook learning for 3D language Gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9855–9864, 2025.

[23] SungMin Jang and Wonjun Kim. Identity-aware language Gaussian splatting for open-vocabulary 3D semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20467–20476, 2025.

[24] Wei Sun, Yuan Li, and Jianbin Jiao. CAGS: Open-vocabulary 3D scene understanding with context-aware Gaussian splatting. *Image and Vision Computing*, 165:105830, 2026.

[25] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: An efficient approach for language Gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024.

[26] Sen Wang, Kunyi Li, Siyun Liang, Elena Alegret, Jing Ma, Nassir Navab, and Stefano Gasperini. Visibility-aware language aggregation for open-vocabulary segmentation in 3D Gaussian splatting. In *Proceedings of the International Conference on 3D Vision*, 2026.

[27] Yen-Jen Chiou, Wei-Tse Cheng, and Yuan-Fu Yang. ProFuse: Efficient cross-view context fusion for open-vocabulary 3D Gaussian splatting. *arXiv preprint arXiv:2601.04754*, 2026.

[28] Jingyu Zhao, Jiahui Wang, Yinan Deng, and Yufeng Yue. OpenMIGS: Multi-granularity information-preserving open-vocabulary 3D Gaussian splatting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21143–21150, 2025.

[29] Dianyi Yang, Xihan Wang, Yu Gao, Shiyang Liu, Bohan Ren, Yufeng Yue, and Yi Yang. OpenGS-Fusion: Open-vocabulary dense mapping with hybrid 3D Gaussian splatting for refined object-level understanding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21135–21142, 2025.

[30] Minchao Jiang, Shunyu Jia, Jiaming Gu, Xiaoyuan Lu, Guangming Zhu, Anqi Dong, and Liang Zhang. VoteSplat: Hough voting Gaussian splatting for 3D scene understanding. *arXiv preprint arXiv:2506.22799*, 2025.

[31] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. GOI: Find 3D Gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the ACM International Conference on Multimedia*, pages 5328–5337, 2024.

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3992–4003, 2023.

[33] Xin Xu, Tianyi Xiong, Zheng Ding, and Zhuowen Tu. MasQCLIP for open-vocabulary universal image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 887–898, 2023.

[34] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3D open-vocabulary segmentation. In *Advances in Neural Information Processing Systems*, 2023.

# A Technical appendices and supplementary material

The appendices are organized as follows. Appendix B provides method details that are compressed in the main text, including raw-pool construction, prototype construction, statistical attributes, the staged matcher, and method limitations. Appendix C collects supplementary experimental material, including setup details, native-setting baseline results, and qualitative failure cases. Appendix D discusses broader impacts and potential misuse.

# B Supplementary method details

## B.1 Raw observation pool

### B.1.1 Role of the raw pool in the overall pipeline

The raw observation pool is the earliest and most irreversible failure point for instance-level matching. If an instance is represented only by a single crop or a single aggregated view, then occlusion, missing context, and rendering artifacts are already written into the representation before the matcher even starts. The method therefore treats raw semantic collection as a first-class stage rather than a passive input assumption.

### B.1.2 Mask-conditioned crop construction

Following the middle-crop practice in OpenSplat3D [5], each observation is encoded from a mask-conditioned crop. We compute the bounding box of $\mathcal{M}^i_{v,s}$, enlarge it by 10% on each side, and clip it to the image boundary. This avoids whole-image distractors while retaining limited context for MasQCLIP beyond the tight mask box. For rendered masks from segmented 3D instances, we additionally remove obvious rendering artifacts and smooth mask boundaries before semantic encoding.

### B.1.3 Geometric quality scores

For the reference prototype, we follow the view-selection score used in OpenSplat3D [5]. For observation $x^i_n$, let $V^i_n$ denote the number of Gaussian primitives of instance $i$ that are visible in the corresponding view, $L^i$ denote the number of Gaussian primitives assigned to instance $i$, $A^i_n$ denote the rendered binary-mask area of the instance in that view, and $A^i_{\max} = \max_m A^i_m$. The geometric-completeness score is

$$c^i_n = \begin{cases} \dfrac{V^i_n}{L^i} \dfrac{A^i_n}{A^i_{\max}}, & L^i > 0 \text{ and } A^i_{\max} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The first factor measures how much of the segmented instance is visible in the view, while the second factor measures the relative projected area of the rendered instance mask.

## B.2 Multi-mode semantic representation

### B.2.1 Prototype selection

The representation is organized into three complementary prototype groups. We use the same notation as in the main text: $p^i_k$ denotes the $k$-th selected prototype of instance $i$, and $j^i_t \in \{1, \ldots, N^i\}$ denotes the feature selected at step $t$.

**First group.** The first group provides a stable reference prototype. It averages the $\kappa$ features with the highest geometric-completeness scores, selected without replacement:

$$j^i_t = \arg\max_j c^i_j, \quad p^i_1 = \kappa^{-1} \sum_{t=1}^{\kappa} x^i_{j^i_t}, \quad t = 1, \ldots, \kappa. \tag{8}$$

Here $\kappa = 5$ in our implementation, and $c^i_j$ is the continuous geometric-completeness score defined in Appendix B.1.3. This prototype serves as the input to the reliable similarity score $s_1$.

**Second group.** The second group preserves dominant and previously undercovered semantic modes by greedily improving coverage of the raw observation pool. With selected prototypes $C_{t-1}$, we define $\mathrm{cover}(x_j^i) = \max_{p \in C_{t-1}} \cos(x_j^i, p)$ and $r_j = \max_{p \in C_{t-1}} \cos(x_j^i, p)$. Let $\mathcal{S}$ denote the set of non-empty observation sources and $\mathcal{U}_s^i$ denote features from source $s$. For candidate feature $x_j^i$,

$$\Delta_j = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{1}{|\mathcal{U}_s^i|} \sum_{x_\ell^i \in \mathcal{U}_s^i} \max\big(\cos(x_\ell^i, x_j^i) - \mathrm{cover}(x_\ell^i), 0\big). \tag{9}$$

The selected prototype is

$$j_t^i = \arg\max_j (\Delta_j - \lambda r_j), \qquad p_t^i = x_{j_t^i}^i. \tag{10}$$

Here $C_{t-1}$ is the selected prototype set before step $t$, $\mathrm{cover}(x_\ell^i)$ is the current best similarity coverage of feature $x_\ell^i$, $\Delta_j$ is the source-balanced residual coverage gain of candidate feature $x_j^i$, $r_j$ penalizes redundancy with existing prototypes, and $\lambda$ controls this penalty. The residual coverage term allows modes not yet well represented by $C_{t-1}$ to contribute to later selections. Selection stops once the minimum budget is met and the gain is small, or when the features are sufficiently covered. The number of prototypes in this group is usually 2–6.

**Third group.** The third group retains diverse features for ambiguity handling. It is selected by farthest-first traversal: The traversal is initialized by the feature with the largest summed similarity to the raw observation pool, $j_t^i = \arg\max_j \sum_\ell \cos(x_j^i, x_\ell^i)$ for $t = 1$, with $p_t^i = x_{j_t^i}^i$. Subsequent features are selected by

$$j_t^i = \arg\max_j d_j^{\min}, \qquad p_t^i = x_{j_t^i}^i. \tag{11}$$

Here $d_j^{\min} = \min_{p \in C_{t-1}} \big(1 - \cos(x_j^i, p)\big)$. We keep three prototypes from the 2D segmentation source because it usually carries richer and cleaner image evidence without rendering artifacts, while one representative rendered 3D prototype is sufficient to provide complementary 3D-source evidence.

### B.2.2 PCA visualization details

Fig. 3 visualizes raw observations, selected prototypes, the query feature, and the single OpenSplat3D descriptor after projecting them with the same PCA transform fitted on the displayed feature set. Because PCA is only a two-dimensional projection of the original feature space, the translucent green regions extended from the prototypes are schematic and illustrate prototype coverage rather than exact metric neighborhoods.

### B.2.3 Statistical attribute selection

The retained attributes are selected from a larger rule-based candidate set. We first considered statistics expected to help distinguish positive from negative query-instance pairs, then removed weak contributors through linear-correlation checks and fine-grained ablations.

### B.2.4 Prototype-level statistical attributes

For a prototype $p_k$, members are assigned by nearest-prototype similarity: $a(x) = \arg\max_k \cos(x, p_k)$, with member set $\mathcal{A}_k = \{x \in \mathcal{U}^i \mid a(x) = k\}$. Let $\mathrm{src}(x) \in \mathcal{S}$ denote the observation source of $x$, where $\mathcal{S} = \{\mathrm{2D}, \mathrm{3D}\}$. For each second-group prototype, we compute:

- **Support ratio**. This measures how many observations are assigned to prototype $p_k$: $r_k^{\mathrm{sup}} = |\mathcal{A}_k|/|\mathcal{U}^i|$. Here $\mathcal{A}_k$ is the member set of $p_k$, and $\mathcal{U}^i$ is the raw observation pool of instance $i$. The prototype with the largest support ratio represents the dominant semantic mode.
- **Source support**. This measures whether prototype $p_k$ is supported by the 2D-segmentation source, the 3D-segmentation source, or both: $r_k^{\mathrm{src}} = |\{s : \exists x \in \mathcal{A}_k, \mathrm{src}(x) = s\}|/|\mathcal{S}|$. This attribute distinguishes prototypes supported by both observation sources from those supported by only one source.
- **Member compactness**. This measures the average feature agreement between prototype $p_k$ and its members: $c_k = |\mathcal{A}_k|^{-1} \sum_{x \in \mathcal{A}_k} \cos(x, p_k)$. A larger value indicates that the assigned observations form a more compact mode.

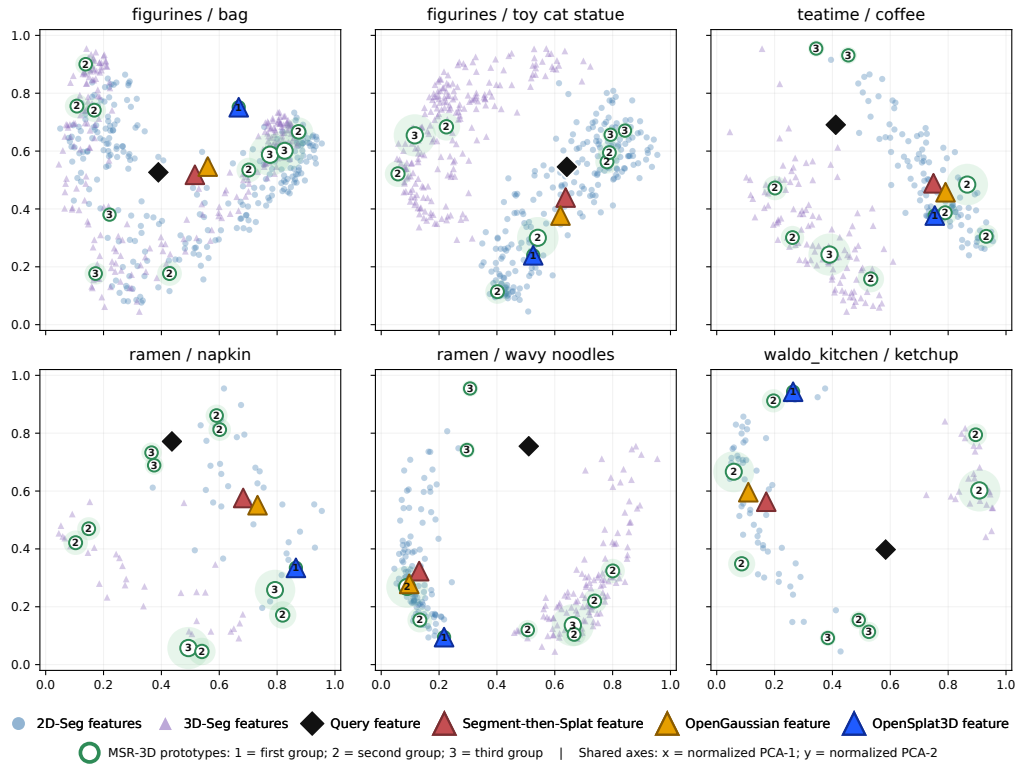


Figure 5: **Additional PCA projections of semantic features.** This figure extends the feature projection experiment in Fig. 3 to more scenes and instances.

- **Prototype separation**. This measures how far prototype $p_k$ is from the closest other prototype: $d_k = 1 - \max_{j \neq k} \cos(p_k, p_j)$. Here $p_j$ ranges over other prototypes of the same instance. A larger value indicates that the prototype captures more distinct semantic evidence.
- **Source composition**. This measures the composition of a specific source within the prototype members: $\pi_k(s) = |\{x \in \mathcal{A}_k : \mathrm{src}(x) = s\}|/|\mathcal{A}_k|$. This attribute records the source balance of each prototype.

### B.2.5 Instance-level statistical attributes

Instance-level attributes summarize the prototype set:

- **Prototype count**. The symbol $K_2$ denotes the number of retained second-group prototypes and reflects the semantic complexity retained for the instance.
- **Dominant support**. This measures the support ratio of the largest mode: $r^{\mathrm{dom}} = \max_k r_k^{\mathrm{sup}}$. Here $r_k^{\mathrm{sup}}$ is the support ratio of prototype $p_k$. A larger value means the instance is dominated by one mode.
- **Source agreement**. This measures support weighted agreement between the 2D-segmentation and 3D-segmentation sources: $a^{\mathrm{src}} = \sum_k r_k^{\mathrm{sup}} r_k^{\mathrm{src}}$. Here $r_k^{\mathrm{src}}$ is the source support of prototype $p_k$. This attribute increases when major modes are supported by both observation sources.

## B.3 Matcher and score calibration

### B.3.1 Concrete definitions of the three score heads

**Score $s_1$: geometry aware branch.** The first score is simply $s_1(Q, i) = \cos(q, p_1^i)$. This branch has no trainable backbone beyond the later calibration and boundary modules.

**Score $s_2$: comprehensive aware branch.** The comprehensive aware branch is decomposed into three stages.

**Linear layer 1.** The input of this stage is the second group prototype set and its prototype level attributes. For each prototype $p_k^i$, we form a 7 dimensional row descriptor from its query similarity and attributes:

$$z_k^i = [\cos(q, p_k^i),\, r_k^{\mathrm{sup}},\, r_k^{\mathrm{src}},\, c_k,\, d_k,\, \pi_k(\mathrm{2D}),\, \pi_k(\mathrm{3D})]^\top. \tag{12}$$

Linear layer 1 is shared across prototype rows: $\ell_{2,k}^i = w_{\mathrm{row}}^\top z_k^i + b_{\mathrm{row}}$. It outputs one row score $\ell_{2,k}^i$ per prototype. Because no row level labels are used, this layer is optimized jointly with the final instance readout through the staged $s_2$ objectives described below. The learned weight pattern is summarized in Table 7. The following summaries report signs and orders of magnitude in the original input coordinates, without feature-wise standardization. Coefficient magnitudes are therefore not directly comparable as feature importance.

Table 7: Sign and order of magnitude of weights in the linear scorer for $s_2$.

| Feature | Sign and scale |
|---|---|
| $\cos(q, p_k^i)$ | positive, $10^0$ scale |
| Support ratio $r_k^{\mathrm{sup}}$ | positive, $10^{-1}$ scale |
| Source support $r_k^{\mathrm{src}}$ | negative, $10^{-1}$ scale |
| Member compactness $c_k$ | negative, $10^{-1}$ scale |
| Prototype separation $d_k$ | negative, $10^{-1}$ scale |
| 2D source composition $\pi_k(\mathrm{2D})$ | positive, $10^{-3}$ scale |
| Rendered 3D source composition $\pi_k(\mathrm{3D})$ | positive, $10^{-2}$ scale |
| Bias | positive, $10^{-1}$ scale |

Query similarity and support have positive coefficients, while compactness and separation have negative coefficients. The source-composition coefficients are small and positive. These signs describe the fitted combination, not the quality of individual attributes in isolation.

**Aggregator.** The input to this stage is the row score set $\{\ell_{2,k}^i\}_{k=1}^{K_2}$, where $K_2 \in [2, 6]$. We aggregate it into three summary statistics:

$$\begin{aligned} \ell_{\max}^i &= \max_k \ell_{2,k}^i, \\ \ell_{\mathrm{top2}}^i &= \mathrm{mean}(\mathrm{top2}(\{\ell_{2,k}^i\}_{k=1}^{K_2})), \\ \ell_{\mathrm{mean}}^i &= \frac{1}{K_2}\sum_{k=1}^{K_2} \ell_{2,k}^i, \\ \boldsymbol{\alpha}_2^i &= [\ell_{\max}^i, \ell_{\mathrm{top2}}^i, \ell_{\mathrm{mean}}^i]^\top. \end{aligned} \tag{13}$$

The output $\boldsymbol{\alpha}_2^i$ summarizes the strongest, top grouped, and average prototype responses of the instance.

**Linear layer 2.** The input of this stage is the row summary from the Aggregator together with instance level attributes. We concatenate them as

$$\mathbf{v}^i = [K_2, r^{\mathrm{dom}}, a^{\mathrm{src}}]^\top, \qquad \mathbf{y}_2^i = \begin{bmatrix} \boldsymbol{\alpha}_2^i \\ \mathbf{v}^i \end{bmatrix}. \tag{14}$$

Linear layer 2 maps $\mathbf{y}_2^i$ to the raw comprehensive score: $s_2^{\mathrm{raw}}(Q, i) = \mathbf{w}_2^\top \mathbf{y}_2^i + b_2$. The $s_2$ head is trained with staged query-level objectives rather than a single one-shot loss. The first objective is a query-level ranking loss. Let $\mathcal{Y}_Q^{\mathrm{full}}$ denote full-object positives, $\mathcal{Y}_Q^{\mathrm{part}}$ part-only positives, $\mathcal{Y}_Q^{\mathrm{band}}$ positives used for the band consistency term, $\mathcal{Y}_Q^+$ all positives, and $\mathcal{N}_Q$ negatives for query $Q$. For a full-object query, the loss is

$$\mathcal{L}_{\mathrm{full}} = \log\left(\exp(\max_{p\in\mathcal{Y}_Q^{\mathrm{full}}} s_p/T) + \sum_{n\in\mathcal{N}_Q}\exp(s_n/T)\right) - \max_{p\in\mathcal{Y}_Q^{\mathrm{full}}} s_p/T. \tag{15}$$

For a part-only query, the loss is

$$\mathcal{L}_{\text{part}} = \log\left(\sum_{p\in\mathcal{Y}_Q^{\text{part}}} \exp(s_p/T) + \sum_{n\in\mathcal{N}_Q} \exp(s_n/T)\right) - \log \sum_{p\in\mathcal{Y}_Q^{\text{part}}} \exp(s_p/T). \tag{16}$$

The ranking objective further includes a full-query hard-margin term and a positive-band consistency term,

$$\begin{aligned} \mathcal{L}_{\text{hard}} &= \text{softplus}\left(-(\max_{p\in\mathcal{Y}_Q^{\text{full}}} s_p - \max_{n\in\mathcal{N}_Q} s_n - \xi_f)\right), \\ \mathcal{L}_{\text{band}} &= \frac{1}{|\mathcal{Y}_Q^{\text{band}}|} \sum_{p\in\mathcal{Y}_Q^{\text{band}}} [\max_{p'\in\mathcal{Y}_Q^{\text{band}}} s_{p'} - s_p - \delta_{\text{band}}]_+. \end{aligned} \tag{17}$$

The score-refinement objective fixes the operating point to $\tau = 0.85$ and optimizes calibrated scores with

$$\mathcal{L}_{\text{ref}} = \lambda_+\mathcal{L}_+ + \lambda_-\mathcal{L}_- + \lambda_{\text{full}}\mathcal{L}_{\text{hard}} + \lambda_{\text{band}}\mathcal{L}_{\text{band}}, \tag{18}$$

where

$$\begin{aligned} \mathcal{L}_+ &= \frac{1}{|\mathcal{Y}_Q^+|} \sum_{p\in\mathcal{Y}_Q^+} [\tau + \delta_+ - s_2(Q,p)]_+, \\ \mathcal{L}_- &= \frac{1}{|\mathcal{N}_{K_h}|} \sum_{n\in\mathcal{N}_{K_h}} [s_2(Q,n) - (\tau - \delta_-)]_+. \end{aligned} \tag{19}$$

Here $s_p$ and $s_n$ denote raw $s_2$ scores for positive and negative instances, $T$ is the temperature, and $\mathcal{N}_{K_h}$ denotes the top-$K_h$ negatives under the calibrated $s_2$ values. We use $(\lambda_+, \lambda_-, \lambda_{\text{full}}, \lambda_{\text{band}}) = (1.0, 1.0, 0.3, 0.2)$, $(\delta_+, \delta_-, \delta_{\text{band}}) = (0.02, 0.03, 0.05)$, and $\xi_f = 0.05$. The output of the readout stage is the raw comprehensive score $s_2^{\text{raw}}$. Query-level normalization and TauNet calibration are described below. The learned weight pattern is summarized in Table 8.

Table 8: Sign and order of magnitude of weights in the linear aggregator for $s_2$.

| Feature | Sign and scale |
|---|---|
| $\ell_{\max}$ | positive, $10^0$ scale |
| $\ell_{\text{top2}}$ | positive, $10^0$ scale |
| $\ell_{\text{mean}}$ | positive, $10^0$ scale |
| Prototype count $K_2$ | positive, $10^{-1}$ scale |
| Dominant support $r^{\text{dom}}$ | negative, $10^{-1}$ scale |
| Source agreement $a^{\text{src}}$ | positive, $10^0$ scale |
| Instance bias | positive, $10^{-1}$ scale |
| Global bias | positive, $10^{-2}$ scale |

The maximum, top-two mean, and mean row scores have positive coefficients. Source agreement is also positive, while dominant support is negative. These signs describe the joint readout rather than the independent importance of each attribute.

**Score $s_3$: ambiguity aware branch.** The ambiguity aware branch combines an instance summary with attention over valid prototype rows. The summary $\mathbf{x}_Q^i \in \mathbb{R}^{20}$ contains the maximum, top-two mean, mean, minimum, and standard deviation of prototype similarities, their within-query ranks and differences from the corresponding query maxima, and the two normalized score margins together with their hyperbolic tangents and product. For each valid row, $\mathbf{u}_k^i \in \mathbb{R}^{16}$ combines the within-instance centered similarity, six standardized prototype attributes, and interactions of centered similarity with those attributes and three instance attributes. The prototype attributes are support, source support, compactness, separation, and the two source proportions; the instance attributes are prototype count, dominant support, and source agreement. Let $\bar{m}^i$ be the mean similarity over valid rows and $\sigma_m$ the source-fit standard deviation of row similarities. With learned vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the attention and bounded score are

$$\begin{aligned} \alpha_k^i &= \text{softmax}_k\left(2\tanh(\boldsymbol{\gamma}^\top \mathbf{u}_k^i)\right), \\ s_3(Q,i) &= 2\tanh\left(\boldsymbol{\beta}^\top \mathbf{x}_Q^i + \frac{\sum_k \alpha_k^i m_k^i(q) - \bar{m}^i}{\sigma_m}\right). \end{aligned} \tag{20}$$

All sums and attention normalization use valid rows only. The instance summary is standardized using fixed source-fit statistics. Row attributes use fixed source-fit statistics, while centered row similarities are divided by $\sigma_m$. Statistics weight queries equally, instances equally within a query, and valid rows equally within an instance for row statistics. The resulting score lies in $(-2, 2)$; its coefficients do not represent independent linear effects on raw attributes.

With upstream modules fixed, we train this branch on the 18 source-scene queries using an equally query-weighted listwise cross-entropy objective over non-base additions and abstention. Targets are determined by Pareto improvements in native rendered-mask metrics relative to the base prediction, with a fixed zero abstention logit. The branch therefore learns when an additional instance is beneficial rather than treating every correspondence-positive instance as a required addition.

### B.3.2 Operator calibration

The $s_2$ scores are centered and rescaled within each query. TauNet then takes their distribution as input and predicts a query specific operating point without transforming the scores themselves. The $s_3$ branch instead uses fixed source-fit feature statistics and its bounded output directly, without query-wise min-max normalization.

**Query level centering and rescaling.** The input is the raw $s_2$ score vector over all candidate instances of a query. Let $\bar{s}_2^i = s_2^{\mathrm{raw}}(Q, i)$. The branch is centered at the query minimum and rescaled by the query range:

$$\tilde{s}_2^i = \frac{\bar{s}_2^i - \min_{j \in \mathcal{C}_Q} \bar{s}_2^j}{\max_{j \in \mathcal{C}_Q} \bar{s}_2^j - \min_{j \in \mathcal{C}_Q} \bar{s}_2^j + \varepsilon}, \qquad s_2(Q, i) = (\tilde{s}_2^i)^{\gamma_c}, \quad \gamma_c = 2.0. \tag{21}$$

This operation has no trainable parameters. Its output is $s_2(Q, i) \in [0, 1]$, preserving the within-query ordering. The $s_1$ response is min-max normalized over its full instance feature bank. Its margin around 0.92 and the $s_2$ margin around the TauNet threshold are each divided by a robust within-query scale and passed through $\tanh$ before boundary evaluation.

**TauNet.** For the comprehensive branch, TauNet predicts a query specific operating point from the calibrated $s_2$ vector $\{s_2(Q, i) : i \in \mathcal{C}_Q\}$. The input to TauNet is a 16 dimensional query summary

$$\begin{aligned} \mathbf{c}_Q = [&t_1, t_2, t_3, t_5, t_1 - t_2, t_1 - t_3, t_1 - t_5, \\ &n_{.8}, n_{.9}, n_{.95}, \mu_Q, \sigma_Q, \Delta_Q, H_Q, N_Q, \bar{t}_3]^\top. \end{aligned} \tag{22}$$

Here $t_m$ is the $m$-th largest calibrated $s_2$ value, $n_\rho$ counts candidates with score above $\rho$, $\mu_Q$, $\sigma_Q$, $\Delta_Q$, and $H_Q$ denote the mean, standard deviation, range, and entropy of the calibrated score distribution, $N_Q = |\mathcal{C}_Q|$, and $\bar{t}_3$ is the mean of the top three scores. TauNet is defined as

$$\begin{aligned} h_Q &= \mathrm{ReLU}(W_\tau^{(1)} \mathbf{c}_Q + b_\tau^{(1)}), \\ z_Q &= W_\tau^{(2)} h_Q + b_\tau^{(2)}, \\ \tau_Q &= \tau_{\min} + (\tau_{\max} - \tau_{\min}) \,\mathrm{sigmoid}(z_Q), \end{aligned} \tag{23}$$

where the hidden dimension is 32, $\tau_{\min} = 0.8$, and $\tau_{\max} = 1.0$. During score refinement, $\tau_Q$ is fixed to 0.85. During TauNet calibration, the earlier Score 2 head is frozen and only TauNet is trained.

Let $\mathcal{Y}_Q^+$ denote the positive instances and $\mathcal{N}_Q^{K_n}$ the top $K_n$ negative instances under the calibrated $s_2$ values. TauNet is trained with a threshold shaping loss:

$$\mathcal{L}_\tau = \lambda_+ \frac{1}{|\mathcal{Y}_Q^+|} \sum_{p \in \mathcal{Y}_Q^+} \big[\tau_Q + \delta_+ - s_2(Q, p)\big]_+ + \lambda_- \frac{1}{|\mathcal{N}_Q^{K_n}|} \sum_{n \in \mathcal{N}_Q^{K_n}} \big[s_2(Q, n) - (\tau_Q - \delta_-)\big]_+, \tag{24}$$

where $[x]_+ = \max(x, 0)$. The output $\tau_Q$ is therefore encouraged to stay below positives and above the strongest negatives, with small margins $\delta_+$ and $\delta_-$.

### B.3.3 Decision boundaries

The first two scores form a base selection through a primary boundary and a shared supplementary classifier. The nonlinear third score then revisits all base-unselected instances while preserving the base selection.

**Primary boundary.** Let $\omega_1^i$ and $\omega_2^i$ denote the centered and rescaled margins of $s_1(Q,i)$ and $s_2(Q,i)$, respectively. The base score follows a recall oriented union rule, $B_{\text{base}}(Q,i) = \max(\omega_1^i, \omega_2^i)$. We then build a small feature vector

$$\phi_{12}(Q,i) = [\omega_1^i,\, \omega_2^i,\, \omega_1^i\omega_2^i,\, \omega_1^i + \omega_2^i,\, \omega_1^i - \omega_2^i,\, B_{\text{base}}(Q,i)]^\top. \tag{25}$$

The residual boundary is

$$\begin{aligned} \Delta_\psi(Q,i) &= W_2^{\text{m}}\,\text{ReLU}(W_1^{\text{m}}\phi_{12}(Q,i) + b_1^{\text{m}}) + b_2^{\text{m}}, \\ B_{12}(Q,i) &= B_{\text{base}}(Q,i) + \Delta_\psi(Q,i). \end{aligned} \tag{26}$$

The hidden dimension is 8. A probability form is obtained by $p_{12}(Q,i) = \text{sigmoid}(\beta B_{12}(Q,i))$, where $\beta = 8$. Let $y_Q^i \in \{0,1\}$ be the positive instance label. The total training objective for the primary boundary is the weighted sum of classification, soft set IoU, pairwise ranking, and residual regularization terms:

$$\mathcal{L}_{12} = \lambda_{\text{cls}}\,\text{BCE}_w(p_{12}, y_Q) + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}} + \lambda_{\text{res}}\frac{1}{|\mathcal{C}_Q|}\sum_{i\in\mathcal{C}_Q}\Delta_\psi(Q,i)^2, \tag{27}$$

$$\mathcal{L}_{\text{iou}} = 1 - \frac{\sum_{i\in\mathcal{C}_Q} p_{12}(Q,i)y_Q^i}{\sum_{i\in\mathcal{C}_Q} p_{12}(Q,i) + \sum_{i\in\mathcal{C}_Q} y_Q^i - \sum_{i\in\mathcal{C}_Q} p_{12}(Q,i)y_Q^i + \varepsilon}, \tag{28}$$

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{Y}_Q^+||\mathcal{N}_Q|}\sum_{p\in\mathcal{Y}_Q^+}\sum_{n\in\mathcal{N}_Q}\big[\delta_{\text{r}} - \big(B_{12}(Q,p) - B_{12}(Q,n)\big)\big]_+. \tag{29}$$

Here $\text{BCE}_w$ is the weighted binary cross entropy over candidate instances, $\mathcal{N}_Q$ denotes the negative instances used for ranking, and the $\lambda$ terms are scalar loss weights. The last term penalizes large residual corrections so that the learned boundary remains close to the base $s_1, s_2$ boundary. The output of the primary boundary is the main selected set $\mathcal{S}_{12}(Q) = \{i \in \mathcal{C}_Q : B_{12}(Q,i) > 0\}$.

**Two-score supplementary classifier.** In this subsection, $\mathcal{C}_Q$ denotes the candidate instances evaluated for query $Q$. Before excluding primary-selected instances, we retain the three highest $s_2$ responses below the TauNet threshold:

$$\mathcal{T}_Q = \text{TopK}_{i\in\mathcal{C}_Q:\, s_2(Q,i)<\tau_Q}(s_2(Q,i), 3), \qquad \mathcal{F}_Q = \mathcal{T}_Q \setminus \mathcal{S}_{12}(Q). \tag{30}$$

The shared classifier takes the two normalized margins as input. With fixed source-fit standardization, it applies an MLP with six hidden units and a sigmoid output:

$$\mathbf{t}^i = \text{Std}([\omega_1^i, \omega_2^i]^\top), \qquad p_{\text{add}}(Q,i) = \text{sigmoid}\big(\mathbf{w}_{\text{s}}^\top\,\text{ReLU}(W_{\text{s}}\mathbf{t}^i + \mathbf{b}_{\text{s}}) + b_{\text{s}}\big). \tag{31}$$

It adds at most the highest-probability candidate meeting its fixed selection threshold. The union with $\mathcal{S}_{12}(Q)$ forms the base set $\mathcal{B}_Q$. If the primary residual boundary selects nothing, its uncorrected union rule supplies the primary set.

**Third-score correction and selection.** Let $H(Q,i)$ be the bounded continuous margin obtained from the primary boundary and the supplementary classifier, with $H(Q,i) > 0$ exactly for $i \in \mathcal{B}_Q$. For the supplementary classifier's highest-ranked candidate, it takes the larger of the bounded primary margin and the $\tanh$-transformed centered logit margin, with sign aligned to base membership; other instances retain the primary margin. We compute $s_3(Q,i)$ for every candidate instance and form

$$R(Q,i) = H(Q,i) + s_3(Q,i). \tag{32}$$

All base-selected instances are protected. Among the remaining instances, the decoder adds only the highest positive corrected margin:

$$\begin{aligned} i^\star &= \arg\max_{i\in\mathcal{C}_Q\setminus\mathcal{B}_Q} R(Q,i), \\ \hat{\mathcal{Y}}_Q &= \begin{cases} \mathcal{B}_Q \cup \{i^\star\}, & R(Q,i^\star) > 0, \\ \mathcal{B}_Q, & \text{otherwise.} \end{cases} \end{aligned} \tag{33}$$

If no unselected candidate remains, the decoder returns $\mathcal{B}_Q$. Ties are resolved by instance identifier. Thus, the third score supplements the base set without removing an existing selection or admitting multiple additional instances.

### B.4 Clarification of representation properties

The conclusion describes MSR-3D as compact, interpretable, and generalizable. We clarify these three properties here to avoid treating them as informal claims.

**Compactness of the representation.** The compactness of MSR-3D comes from both its representation design and its attribute selection process:

- **Representation design**. MSR-3D does not retain the full observation bag for each instance. As described in Sec. 3.2 and Appendix B, raw observations are compressed into a small prototype set and auxiliary statistical attributes while preserving several semantic modes.
- **Ablation evidence**. Table 5 shows that adding prototype-level and instance-level attributes improves the selection metrics over prototype similarity alone. Appendix B.2.3 further explains that the retained attributes were selected from a larger candidate set after removing weak contributors, and that the final attributes are weakly linearly correlated with each other.

**Interpretability of the representation.** The interpretability of MSR-3D comes from three sources:

- **Representation components**. Prototypes summarize semantic modes, so each retained prototype corresponds to a concrete subset of observations rather than an opaque latent token.
- **Statistical attributes**. The prototype-level and instance-level attributes are explicitly defined in Appendices B.2.4 and B.2.5. These attributes expose why one prototype is treated as more reliable or more distinctive than another.
- **Matcher structure**. The two linear $s_2$ readouts permit sign-based analysis of semantic responses and statistical attributes. The nonlinear $s_3$ branch combines explicit summary statistics and attribute-conditioned attention, as described in Appendix B.3; its coefficients are not interpreted as independent attribute effects.

**Generalizability of the representation.** The claim refers to the benchmark protocol rather than an unrestricted deployment guarantee. The evidence comes from three aspects:

- **Fixed statistical construction**. MSR-3D uses fixed prototype construction and explicitly defined statistical attributes, as described in Sec. 3.2 and Appendices B.2.3, B.2.4, and B.2.5. It therefore does not optimize a new representation for each target scene.
- **Cross scene and cross dataset evaluation**. Sec. 4.2 evaluates fixed representation, calibration, and decision modules across LERF-OVS scenes and the 3DOVS benchmark, with the supervised Waldo scene identified separately. Tables 2 and 1 show that the gains are not limited to the training scene or a single benchmark.
- **Calibrated score space separation**. Sec. 3.3 and Appendix B.3.3 define a matcher that maps each query and instance to calibrated semantic scores and learns compact boundaries for positive negative separation. This design separates instances in the learned score space instead of thresholding a single similarity response.

### B.5 Method limitations

MSR-3D assumes that a 3DGS scene and candidate 3D instances are already available. MSR-3D remains constrained by upstream instance segmentation and does not recover missing target instances or separate targets from severely merged candidates. Severe reconstruction errors and fragmented masks also remain uncorrected before the raw observation pool is built. The representation also depends on frozen 2D vision-language features; if the base model poorly aligns a fine-grained category with the query text, the downstream matcher has limited semantic evidence to recover the target.

The compact multi-mode representation is a deliberate tradeoff. It preserves several representative semantic modes and their statistical attributes, but it does not retain the full multi-view observation bag. Rare observations or extremely view-specific cues remain susceptible to loss during prototype selection. In addition, the calibrated score operators and decision boundaries are evaluated under the

benchmark protocols used in this paper. Further validation is needed under substantial shifts in scene type, segmentation quality, or camera coverage.

# C Additional experimental analysis

## C.1 Supplementary experimental setup details

**Benchmark scale.** LERF-OVS contains four scenes, namely *figurines*, *ramen*, *teatime*, and *waldo_kitchen*, with 67 natural-language queries in total. Our 3DOVS evaluation covers 10 scenes with 61 scene-query pairs.

**Ground-truth matching and supervision.** We initialize the query-instance correspondence table by treating a rendered mask and a ground-truth mask with $\mathrm{IoU} > 0.1$ as a positive match, then manually verify and correct the correspondences as needed. The lower correspondence threshold accommodates valid instances whose masks cover only a small fraction of a multi-object query mask. These correspondences define the ground-truth instance sets for $mF1$, whereas Acc@0.25 is computed separately from the query mean rendered-mask IoU. Scene-level $mIoU$, $mAcc$, and $mF1$ average the corresponding query-level metrics within each scene. Dataset-level values then average the scene-level metrics with equal scene weights.

The 18 source-scene queries used for fitting and validation are covered by 22 query–frame ground-truth masks. For the 14 Waldo fitting queries, the strict query–instance correspondence labels comprise 4,592 pairs, with 25 positives and 4,567 negatives. These pairs are generated automatically rather than annotated individually from scratch. These counts do not describe the separate action-benefit supervision used for $s_3$. Manual verification took approximately 5 minutes for the training labels and 30 minutes for the evaluation labels; these figures measure verification time rather than full annotation cost. Evaluation labels are used only for assessment and never for matcher fitting or model selection. The frozen matcher requires no additional query masks or query–instance labels on target scenes.

**Implementation details.** Unless otherwise stated, we use SAM to generate 2D instance masks and MasQCLIP [33] to extract instance-aware semantic features. A † mark denotes single-select evaluation, while methods without † use their default multi-instance rule. A $\star$ mark denotes that we further tuned the ported selection parameters after the native rule transfer because the untuned quality was too low on the shared MSR-3D scenes.

**Compute resources.** The main GPU memory cost comes from the upstream 3DGS reconstruction, mask rendering, SAM mask generation, and MasQCLIP feature extraction. These preprocessing steps are run on the same 24GB RTX 4090 GPUs and follow the memory profile of the corresponding upstream pipelines. MSR-3D itself adds compact MLPs, the linear $s_2$ scorer, and a nonlinear attention-based $s_3$ branch. Once the instance features are cached, matching operates on at most a few prototypes and scalar attributes per instance, so the additional model memory of the matcher is well below 1GB and is also compatible with CPU evaluation. We did not keep a unified wall-clock log for every preprocessing and evaluation run, so we do not report a separate end-to-end inference-time table.

## C.2 Licenses and existing assets

We use existing datasets, pretrained models, and research code only for benchmark evaluation. The LERF code repository is released under the MIT license, while the LERF-OVS benchmark files used in our evaluation come from the public LERF and expanded LERF releases used by prior work. We did not find a standalone license file for these benchmark files or their re-annotated labels. The 3DOVS benchmark is downloaded from public 3D-OVS and LangSplat data links, whose repositories provide dataset download links but no standalone dataset license file. We use LERF-OVS and 3DOVS for academic benchmark evaluation. SAM is released under the Apache 2.0 license. MasQCLIP is released under the Creative Commons Attribution-NonCommercial 4.0 International license. OpenAI CLIP is released under the MIT license. The 3DGS reference implementation, OpenSplat3D, OpenGaussian, Dr.Splat, and LangSplat are released under the Gaussian-Splatting License, which permits non-commercial research and evaluation use. Segment then Splat is released

under the MIT license. For LaGa, we did not find a standalone license file in the public repository or the local copy, so we use it only for academic comparison and do not redistribute its code. The release includes processed benchmark assets with source attribution and component-specific license information.

**Internal learned parameters.** MSR-3D contains only lightweight learned scoring parameters. Appendix B.3 summarizes the signs and orders of magnitude of the two linear $s_2$ readouts in their original input coordinates. The nonlinear $s_3$ branch and the compact TauNet and decision-boundary MLPs are also described there. The public release includes representation construction, query matching, evaluation code, and pretrained parameters.

### C.3 Additional quantitative results

#### C.3.1 Native-setting context and unified-protocol results

Table 9 lists reported native-setting results alongside unified-protocol LERF-OVS results. Native-setting results are included only for context because the underlying segmentation backbone, training protocol, and evaluation setup differ across papers. We omit native-setting 3DOVS results from this table because ten-scene 3DOVS results are not consistently available in prior work; most papers report the five-scene split, which is not aligned with the ten-scene protocol used in our unified evaluation.

**Important note.** The native-setting numbers in this table are not directly comparable. Differences in the segmentation model, candidate set, rendered-view protocol, and dataset split can dominate the reported scores. Native-setting numbers should therefore be treated only as contextual references rather than evidence of comparable segmentation quality under a shared protocol.

Table 9: Native-setting context and unified-protocol LERF-OVS results (%).

(a) LERF-OVS scene-level results reported by the corresponding papers.

| | Figurines | | Ramen | | Teatime | | Waldo Kitchen | |
|---|---|---|---|---|---|---|---|---|
| Method | $mIoU$ | $mAcc$ | $mIoU$ | $mAcc$ | $mIoU$ | $mAcc$ | $mIoU$ | $mAcc$ |
| OpenGaussian [2] | 39.29 | 55.36 | 31.01 | 42.25 | 60.44 | 76.27 | 22.70 | 31.82 |
| SuperGSeg [19] | 43.68 | 60.71 | 18.07 | 23.94 | 55.31 | 77.97 | 26.71 | 45.45 |
| Dr.Splat [4] | 54.42 | 80.36 | 24.33 | 35.21 | 57.35 | 77.97 | 37.05 | 63.64 |
| OpenSplat3D [5] | 60.71 | 85.71 | 49.20 | 76.06 | 73.27 | 88.14 | 55.63 | 77.27 |
| CAGS [24] | 60.85 | 82.14 | 36.29 | 46.48 | 68.40 | 86.44 | 37.62 | 63.64 |
| VoteSplat [30] | 68.62 | 85.71 | 39.24 | 61.97 | 66.71 | 88.14 | 25.84 | 33.68 |
| LAGA [9] | 64.10 | – | 55.60 | – | 70.90 | – | 65.60 | – |
| Segment-then-Splat [6] | 49.83 | – | 54.38 | – | 63.47 | – | 40.71 | – |
| VALA [26] | 60.38 | 89.29 | 45.41 | 67.61 | 70.61 | 88.14 | 55.71 | 86.36 |

(b) LERF-OVS dataset-level results under original-paper and unified settings.

| Method | Original Paper | | Unified Protocol | |
|---|---|---|---|---|
| | $mIoU$ | $mAcc$ | $mIoU$ | $mAcc$ |
| OpenGaussian [2] | 38.36 | 51.43 | 55.08 | 75.20 |
| SuperGSeg [19] | 35.94 | 52.02 | – | – |
| Occam's LGS [25] | 61.30 | 82.50 | – | – |
| Dr.Splat [4] | 43.29 | 64.30 | 48.79 | 71.63 |
| InstanceGaussian [18] | 45.30 | 58.44 | – | – |
| OpenSplat3D [5] | 59.70 | 81.79 | 58.01 | 75.20 |
| VoteSplat [30] | 50.10 | 67.38 | – | – |
| CAGS [24] | 50.79 | 69.62 | 55.01 | 72.82 |
| OpenInsGaussian [3] | 42.62 | 62.11 | 54.63 | 74.01 |
| LAGA [9] | 64.00 | – | 54.74 | 73.61 |
| Segment-then-Splat [6] | 52.10 | – | 50.74 | 64.88 |
| LangSplatV2 [15] | 59.90 | 84.10 | – | – |
| VALA [26] | 58.02 | 82.85 | 54.38 | 75.79 |
| Ours | – | – | 64.70 | 85.52 |

*Note.* Original-paper values retain their reported native settings. Unified-protocol values cover all 67 queries and average the four LERF-OVS scenes equally; – indicates an unavailable value.

#### C.3.2 Candidate-quality evaluation details

Table 3 uses the same frozen matcher and all 67 LERF-OVS and 61 3DOVS queries as the main comparison. Candidate quality is the maximum recorded query-instance IoU among correspondence-

positive candidates, rather than the final prediction IoU. Positive correspondences exist for 65 LERF-OVS queries and all 61 3DOVS queries; the remaining two LERF-OVS queries are reported separately. Each entry averages rendered response IoU over annotated frames within each query, then over queries within each scene, and finally equally over scenes represented in the bin. Empty bins are marked with –. The two lowest-quality bins on 3DOVS contain one query each and do not establish a reliable improvement pattern.

### C.3.3 Representation and matching configurations

We compare four representation and matching configurations on all 67 LERF-OVS queries, averaging metrics equally across the four scenes, including the supervised Waldo scene. A single descriptor uses the mean instance feature, whereas the multi-descriptor threshold baseline uses the maximum cosine similarity over valid prototypes without statistical attributes. For the single-descriptor refit, statistical attributes are removed and the score frontends remain fixed; the primary boundary, supplementary classifier, and third-score branch are refitted with their original architectures. The first two modules are fitted on 14 Waldo queries, with model and policy selection on all 18 Waldo queries; the third-score branch is fitted on all 18. The selected policy disables additions by the supplementary classifier. The similarity thresholds are selected on the same 18 source queries by rendered-mask $mIoU$, with instance $F1$ as the secondary criterion. This is not end-to-end retraining, and the overlapping source queries do not form an independent validation set. The full multi-mode model is kept unchanged.

Table 10: Representation and matching ablation on LERF-OVS (%).

| Representation | Matching rule | $mIoU$ | $mAcc$ | $mF1$ |
|---|---|---|---|---|
| Single descriptor | Refitted matcher | 59.46 | 81.15 | 69.51 |
| Single descriptor | Similarity threshold | 45.94 | 62.70 | 49.18 |
| Multiple descriptors | Max-similarity threshold | 42.79 | 62.90 | 48.75 |
| Multi-mode representation | Full matcher | 64.70 | 85.52 | 72.63 |

Table 10 shows that refitting the downstream matcher improves all three metrics over thresholding a single descriptor, while maximum similarity over multiple descriptors is insufficient to improve all metrics. The full configuration performs best. Because representation and matching configurations differ across rows, these comparisons support their combined contribution rather than isolate the effect of representation alone.

### C.3.4 Selection rules with fixed semantic scores

We next hold the candidate instances and all three semantic scores fixed and compare how they are converted into an instance set, using the same LERF-OVS evaluation above. An MLP with eight hidden units maps the three scores to a scalar. It is trained on 14 Waldo queries with class-balanced binary cross-entropy; the training epoch is selected on all 18 Waldo queries by instance-set IoU and F1. We compare selecting its highest-scoring instance with selecting all instances whose scalar exceeds a shared threshold. Both rules use the same fitted MLP. A separate axis-aligned rule selects the intersection of enabled score-threshold conditions. The scalar threshold and the joint threshold configuration are selected on the 18 source queries by rendered-mask $mIoU$, with instance $F1$ as the secondary criterion. These queries include the fitting set, so selection is not independently validated. The joint search uses a finite grid allowing each score condition to be disabled; the selected rule uses only $s_1$ and $s_2$. The full matcher retains its frozen selection procedure without refitting.

Table 11: Selection-rule ablation with fixed scores on LERF-OVS (%).

| Selection rule | $mIoU$ | $mAcc$ | $mF1$ |
|---|---|---|---|
| Scalar MLP, top-ranked instance | 54.80 | 72.82 | 67.21 |
| Scalar MLP, shared threshold | 59.69 | 82.94 | 67.98 |
| Joint score thresholds | 59.71 | 81.15 | 64.72 |
| Full matcher | 64.70 | 85.52 | 72.63 |

Table 11 shows that the full matcher outperforms these simplified selection rules on all three metrics. Unlike score-channel removal, this comparison leaves score generation unchanged. The alternatives

differ in architecture, training objective, and set selection, so the results compare complete decision procedures rather than attribute the gains solely to boundary shape.

### C.4 Failure cases

The remaining errors mainly arise from limitations in the input observations and the evaluation setting:

- **Mask or reconstruction error**. Severe mask errors or inaccurate reconstruction produce incomplete or shifted observations in both 2D segmentation and rendered 3D segmentation sources.
- **2D semantic model limitation**. MSR-3D uses semantic features extracted by a 2D vision language model. If the base model fails to recognize a fine grained category or align it with the query text, the downstream representation has limited semantic evidence to recover the target.
- **Semantically close distractors**. Some queries share visual evidence with instances outside the target set, especially for small objects, repeated objects, or objects with similar local context. In these cases, the matcher receives limited evidence for separating the target from semantically close distractors.
- **Frame limited annotation**. Some annotated evaluation frames show only part of the relevant 3D instance set. Predicted instances outside the visible or annotated regions can therefore be penalized by frame-based metrics.

These cases show that the proposed matching stage still depends on the quality of the underlying instance masks and the completeness of the available observations.

## D Broader impacts and potential misuse

This work studies the instance-level matching setting in open-vocabulary 3D scene understanding. Positive uses include robotic assistance, embodied navigation, and interactive scene retrieval. At the same time, the same capability could be misused in surveillance-oriented pipelines that localize or retrieve sensitive objects in reconstructed environments. In addition, the current study is evaluated mainly on indoor benchmarks, so its behavior under broader demographic, environmental, or deployment variation is not established. We therefore present the proposed method as a research method for controlled benchmark evaluation rather than a deployment-ready system.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The main claims are stated in the abstract and introduction and are supported by the quantitative, qualitative, and ablation results in Sec. 4.2, Fig. 4, and Tables 4–6.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The method limitations, scope constraints, and failure modes are discussed in Appendix B.5, Appendix C.4, and the broader-impact discussion above.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A].

Justification: This paper does not present formal theoretical results requiring theorem statements or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The paper and appendix specify the method and evaluation protocol. The public release provides code, pretrained weights, benchmark data, and instructions for reproducing the main results.

Guidelines:

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

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

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

   Answer: [Yes].

   Justification: Code, pretrained weights, and reproduction instructions are available at `https://github.com/SparklingPumpkin/MSR-3D`, with processed benchmark data at `https://huggingface.co/datasets/KaiLai/MSR-3D-Data`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: Datasets, metrics, baselines, and key implementation details are summarized in Sec. 4.2 and Appendix C.1.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No].

   Justification: We report deterministic benchmark metrics on fixed query sets, but we do not currently provide repeated-run error bars, confidence intervals, or significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: Hardware, memory usage, and the scope of wall-clock accounting are discussed in Appendix C.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: We have read the NeurIPS Code of Ethics and are not aware of deviations from it in this research setting.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: Potential positive uses, misuse risks, and deployment caveats are discussed in Appendix D.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A].

    Justification: This submission does not release a high-risk generative or foundation model that requires controlled-access safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes].

    Justification: Licenses and usage terms for the main datasets, pretrained models, and external code used in our experiments are summarized in Appendix C.2. The release includes source attribution and component-specific license information.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes].

    Justification: The released code, pretrained weights, and processed benchmark data are accompanied by installation, data-preparation, and evaluation instructions, input specifications, and source and license documentation.

    Guidelines:

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

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

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

    Answer: [N/A].

    Justification: This work does not use crowdsourcing or involve human-subject data collection.

    Guidelines:

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

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

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

    Answer: [N/A].

    Justification: This work does not involve human-subject research requiring IRB or equivalent approval.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A].

    Justification: LLMs are not an important, original, or non-standard component of the core method in this research.

    Guidelines:

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