ViSPLA: Visual Iterative Self-Prompting for Language-Guided 3D Affordance Learning

Hritam Basak
Department of Computer Science
Stony Brook University
Stony Brook, NY, USA
hbasak@cs.stonybrook.edu

Zhaozheng Yin
Department of Computer Science
Stony Brook University
Stony Brook, NY, USA
zyin@cs.stonybrook.edu

Abstract

We address the problem of language-guided 3D affordance prediction, a core capability for embodied agents interacting with unstructured environments. Existing methods often rely on fixed affordance categories or require external expert prompts, limiting their ability to generalize across different objects and interpret multi-step instructions. In this work, we introduce *ViSPLA*, a novel iterative self-prompting framework that leverages the intrinsic geometry of predicted masks for continual refinement. We redefine affordance detection as a language-conditioned segmentation task: given a 3D point cloud and language instruction, our model predicts a sequence of refined affordance masks, each guided by differential geometric feedback including Laplacians, normal derivatives, and curvature fields. This feedback is encoded into visual prompts that drive a multi-stage refinement decoder, enabling the model to self-correct and adapt to complex spatial structures. To further enhance precision and coherence, we introduce Implicit Neural Affordance Fields, which define continuous probabilistic regions over the 3D surface without additional supervision. Additionally, our Spectral Convolutional Self-Prompting module operates in the frequency domain of the point cloud, enabling multi-scale refinement that captures both coarse and fine affordance structures. Extensive experiments demonstrate that *ViSPLA* achieves state-of-the-art results on both seen and unseen objects on two benchmark datasets. Our framework establishes a new paradigm for open-world 3D affordance reasoning by unifying language comprehension with low-level geometric perception through iterative refinement. Project Website

1 Introduction

Affordance, initially conceptualized by Gibson [1], defines the potential action possibilities that objects present to an agent. The evolution of robotic systems toward increasingly unstructured environments necessitates a fundamental paradigm shift in how we conceptualize affordance detection. Formally, we can represent the affordance detection problem as a mapping function $f_\theta : (\mathcal{P}) \mapsto \mathcal{A}$, where $\mathcal{P} \in \mathbb{R}^{N\times 3}$ denotes a point cloud with N points, and $\mathcal{A}$ is the binary affordance mask indicating interactable regions. As shown in Figure 1(a), traditional approaches constrain this mapping to a limited set of predefined K affordance categories $\mathcal{A} = \{a_k\}; k = \{1, 2, .., K\}$, which fundamentally restricts the generalization capability and operational flexibility in dynamic, real-world environments [2]. Although conventional methodologies have predominantly focused on visual modalities, attempting to infer functionality from geometric structures or 2D visual features, such approaches inherently lack the semantic reasoning capabilities essential for complex interaction scenarios. The semantic gap between low-level perceptual features and high-level functional understanding represents a critical

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

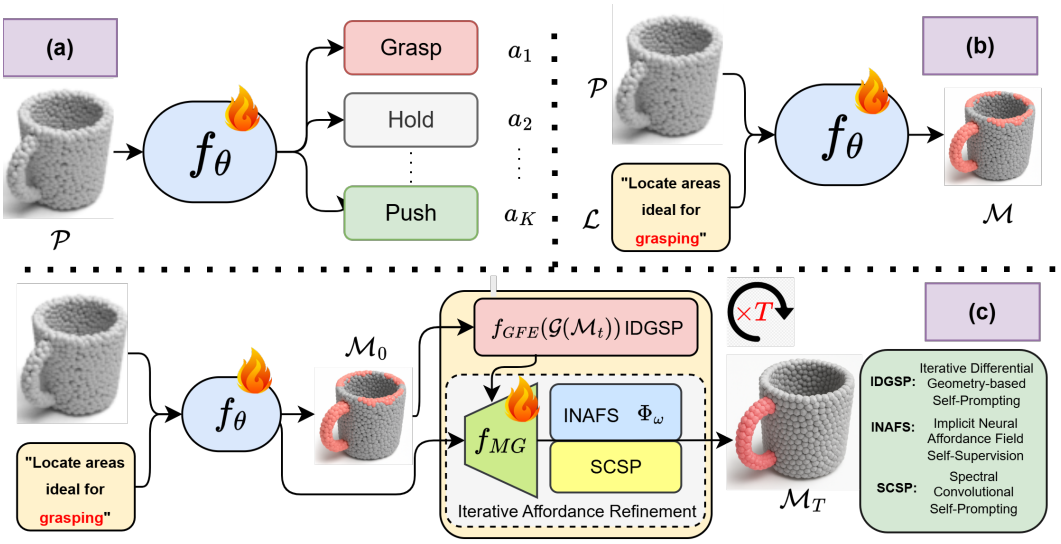

Figure 1: (a) Traditional vision-based methods [3, 4] rely on trainable network f_θ to predict a fixed set of affordances $f_\theta : \mathcal{P} \mapsto \mathcal{A}$; $\mathcal{A} = \{a_1, a_2, ..a_K\}$; (b) Language input along with point-cloud add more flexibility to comprehend complex language instructions and mitigate the problem of open-set affordance prediction [5, 6]: $f_\theta : (\mathcal{P}, \mathcal{L}) \mapsto \mathcal{M}$; (c) We propose refining the initial affordance prediction $\mathcal{M}_0$ for T steps using our proposed IDGSP, and Iterative Affordance Refinement module, consisting of multi-stage refinement decoder f_{MG}, INAFS (Φ_ω) and SCSP. Our solution could be formulated as $f_\theta : (\mathcal{P}, \mathcal{G}(\mathcal{M}_{t-1}), \mathcal{L}) \mapsto \mathcal{M}_t$; $t \in \{1, 2, ..T\}$. Details can be found in section 3.

limitation that inhibits the deployment of autonomous agents in real-world contexts. Language-guided affordance prediction offers a mathematically elegant solution to this complex problem.

Language-guided affordance detection from 3D point clouds represents a pivotal direction in embodied AI, serving as the critical bridge between perception and manipulation in the physical world. By conditioning the affordance function on natural language instructions, we can formulate a more generalizable mapping: $\mathcal{F}_\theta : (\mathcal{P}, \mathcal{L}) \mapsto \mathcal{M}$, where θ represents the learnable parameters, $\mathcal{L}$ represents a linguistic instruction, and $\mathcal{M} \in \{0, 1\}^N$ is the binary affordance mask (visualized in Figure 1(b)). This formulation opens avenues for handling more diverse and complex scenarios—potentially allowing models to interpret novel affordance types via linguistic cues, handle multi-step tasks through decomposed predictions, and relate instructions to affordances at different levels of granularity. Recent progress in Large Language Models (LLMs) has shown impressive capabilities in sequential reasoning and knowledge grounding [7], but these models are often decoupled from 3D perception. Meanwhile, 3D affordance detection methods typically remain limited to static, single-affordance settings, with little capacity to handle instructions requiring compositional or context-aware reasoning across multiple object parts. This disconnect motivates a more integrated, multimodal approach that unifies linguistic understanding with spatial perception.

To this end, we propose an iterative self-prompting-based 3D affordance detection paradigm that bridges the gap between language understanding and affordance segmentation through geometric feedback-driven refinement, as shown in Figure 1(c). Unlike prior approaches that perform single-pass inference, our method implements a closed-loop system where each predicted affordance mask is used to generate geometric self-prompts that refine subsequent predictions. Mathematically, we formulate this as: $\mathcal{M}_t = f_\theta\big(\mathcal{P}, \mathcal{G}(\mathcal{M}_{t-1}), \mathcal{L}\big)$; $t \in \{1, 2, .., T\}$, where $\mathcal{M}_0 = f_\theta(\mathcal{P}, \mathcal{L})$ is the initial affordance mask predicted from a language-conditioned decoder, and $\mathcal{G}$ denotes the geometric prompt generator that extracts differential features (e.g., curvature, normal derivatives) from $\mathcal{M}_{t-1}$. The final refined mask $\mathcal{M}_T$ integrates both semantic guidance and geometric consistency, enabling robust and generalizable affordance segmentation across varying levels of granularity and complexity.

This approach addresses several critical challenges in the field: **(1)** Existing single-pass inference methods lack the ability to iteratively refine predictions, often leading to suboptimal segmentation, especially on complex geometries; **(2)** most affordance models fail to leverage intrinsic geometric structure for mask refinement, relying instead on language cues alone, which limits localization

accuracy, especially in complex or ambiguous settings; **(3)** the disconnect between high-level language semantics and low-level geometric features, hindering precise and context-aware affordance prediction across multiple scales; and **(4)** the difficulty of achieving fine-grained, geometrically consistent affordance boundaries without dense supervision, particularly in sparse or noisy point clouds. In summary, our contributions are:

- We introduce Visual Iterative Self-Prompting for 3D Affordance Learning (ViSPLA), which leverages geometric features from predicted masks as visual prompts for progressive refinement. Unlike existing single-pass methods, our approach establishes a self-improving cycle that enhances precision across multiple object geometries.
- We propose a novel Differential Geometric Self-Prompting mechanism that extracts mathematical properties (Laplacians, curvatures, normal derivatives) from predicted masks to guide subsequent iterations. This approach enables more accurate affordance localization by incorporating intrinsic geometric cues rather than relying solely on language.
- We develop a Multi-Stage Refinement Decoder that creates dynamic mappings between language tokens and point features. By injecting LLM reasoning into dense point features, our approach bridges high-level semantic understanding with low-level geometric representation.
- We introduce an Implicit Neural Affordance Field technique that learns a smooth, continuous function over the 3D object to refine affordance boundaries and enforce geometric consistency, even without extra supervision. In tandem, our Spectral Convolutional Self-Prompting module analyzes and enhances affordance predictions at multiple structural scales, enabling the model to capture both broad shapes and fine details for robust and accurate segmentation, especially in sparse or noisy scenarios.
- We demonstrate that fine-tuning pre-trained MLLMs through our self-prompting framework yields superior performance on both seen and unseen scenarios.

2 Related Work

2.1 Affordance Detection

Affordance detection aims to identify functionally interactive regions on objects, crucial for enabling robotic agents to manipulate and reason about the physical world. Early works in 2D explored object-level affordances using CNNs [3], later extending to language-conditioned queries [8], but remained limited to coarse spatial reasoning. Subsequent efforts [9, 10] introduced fine-grained part-level detection but were restricted to fixed affordance taxonomies. The emergence of 3D datasets like 3D AffordanceNet [4] and PartNet [11] enabled point cloud-based affordance learning. IAGNet [12] utilized 2D human-object interactions to guide 3D segmentation, while OpenAD [13] advanced open-vocabulary affordance detection using joint text-geometry embeddings. However, these models still rely on static label spaces and do not support complex instruction understanding. Recent methods like LASO [14] incorporate language into 3D affordance prediction, but often assume one-to-one mappings between text and affordance, lacking support for multi-step or compositional reasoning. Chu *et al.* [15] use LLMs for cross-modal object retrieval but cannot produce spatially grounded interaction masks.

In contrast, our work formulates affordance detection as an instruction-conditioned segmentation task that enables open-vocabulary, multi-step reasoning directly over 3D point clouds, overcoming the rigidity of fixed labels and the limitations of prior semantic alignment methods.

2.2 Multimodal Large Language Models

Multimodal Large Language Models (MLLMs) extend the language understanding capabilities of LLMs to the visual and spatial domains [16, 17] by aligning textual tokens with visual and geometric inputs. Initial breakthroughs in 2D MLLMs [18–20] enabled joint reasoning over images and text, yet these models lacked the granularity necessary for fine-grained visual tasks such as segmentation. To mitigate this, models like LLaVA [21] introduced spatial localization, improving regional understanding.

Inspired by this progress, researchers have begun extending MLLMs to the 3D domain. Object-centric MLLMs such as PointLLM [22] and ShapeLLM [23] utilize point-based encoders and

multi-view distillation to encode geometric structure and semantics. These models demonstrate strong performance in 3D captioning and object-level referring expression grounding, but often operate on isolated objects and fail to model complex inter-object spatial dependencies. Scene-level LMMs such as Chat-3D [24], LL3DA [25], and 3D-LLM [26] extend grounding to richer indoor scenes using object identifiers, positional embeddings, and pre-selection modules to facilitate dialogue-driven scene understanding.

However, despite these advances, existing 3D MLLMs predominantly focus on global grounding and object identification, lacking the capacity for localized, affordance-specific segmentation or functional reasoning over object parts. 3D-AffordanceLLM [6] takes a step forward by introducing an <AFF> token and a custom decoder to generate affordance masks from natural language queries. Unlike [6], which performs single-pass instruction-to-mask mapping with no feedback loop, our method introduces an Iterative Self-Prompting mechanism that progressively refines predictions by leveraging prior affordance masks as feedback prompts. Moreover, 3D-AffordanceLLM lacks any geometric introspection; in contrast, our Differential Geometric Self-Prompting explicitly uses curvature, Laplacian, and boundary topology cues for precise localization, going beyond language-only guidance. While 3D-AffordanceLLM relies on a fixed decoder architecture, our Multi-scale Visual-Language Integration Module dynamically aligns instructions with geometric features at varying resolutions, enhancing affordance prediction across objects of diverse scale and complexity. Finally, unlike their static segmentation output, our Affordance Dictionary Adaptive Fusion fuses temporal and functional context across steps, enabling robust multi-stage affordance reasoning. Together, these advances allow our model to achieve superior open-world generalization and performance in sequential, instruction-grounded 3D affordance tasks.

3 Proposed Method

We propose *ViSPLA*, a novel iterative self-prompting framework for language-guided 3D affordance detection that incorporates differential geometric feedback for progressive mask refinement. Unlike previous methods that rely on single-pass inference, our approach employs a recurrent self-prompting mechanism that leverages the intrinsic geometric properties of predicted affordance masks to guide subsequent refinements.

3.1 Probelm Formulation

Following the paradigm reformation introduced by 3D-AffordanceLLM [6], we formulate affordance detection as an Instruction Reasoning Affordance Segmentation (IRAS) task. Given a natural language instruction $\mathcal{L}$ and a point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ containing N points, our goal is to predict a binary affordance mask $\mathcal{M} \in \{0,1\}^N$ indicating regions suitable for the specified interaction. While existing approaches [6] model this as a direct mapping $f_\theta : (\mathcal{P}, \mathcal{L}) \mapsto \mathcal{M}$, we introduce an iterative refinement process, as already described in section 1:

$$\mathcal{M}_t = f_\theta\big(\mathcal{P}, \mathcal{G}(\mathcal{M}_{t-1}), \mathcal{L}\big);\ t \in \{1, 2, ..., T\} \tag{1}$$

where $\mathcal{M}_0 = f_\theta(\mathcal{P}, \mathcal{L})$ is the initial affordance prediction and $\mathcal{G}$ is our proposed geometric prompt generator that extracts meaningful differential features from previous mask predictions. *ViSPLA* consists of three main components: **(1)** a language-guided affordance detection backbone based on 3D-AffordanceLLM, **(2)** a differential geometry-based self-prompting module, and **(3)** an iterative affordance refinement module, consisting of a multi-stage refinement decoder, implicit neural field supervision, and spectral convolutional self-prompting. The overall workflow is shown in Figure 2.

3.2 Preliminaries: Language-guided Affordance Detection Backbone

We build upon the 3D-AffordanceLLM [6] architecture, adopting it as our backbone, which comprises a pre-trained point encoder f_{PE}, a point cloud backbone f_{PB}, a projection module f_{proj}, a large language model f_{LLM}, and an affordance decoder f_{AFD}. Given an input point cloud $\mathcal{P}$ and a natural language instruction $\mathcal{L}$, the system proceeds as follows: the point encoder first extracts geometric features $X = f_{\text{PE}}(\mathcal{P}) \in \mathbb{R}^{m\times c}$, where m denotes the number of keypoints and c is the feature dimension. These features are projected into the token space via f_{proj}, yielding $Y = f_{\text{proj}}(X) \in \mathbb{R}^{m\times d}$, where d matches the dimensionality of the LLM token embeddings. The projected point tokens are concatenated with the instruction tokens and passed into the LLM f_{LLM}, which processes them

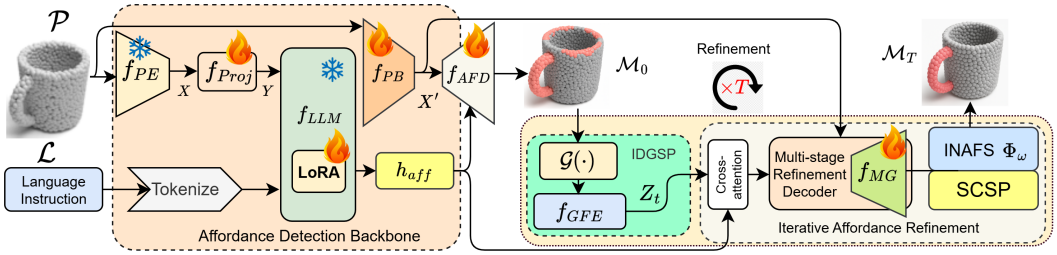

Figure 2: Overview of the *ViSPLA* framework: given a point cloud $\mathcal{P}$ and a language instruction $\mathcal{L}$, first we extract geometric features $X = f_{PE}(\mathcal{P})$, project them to $Y = f_{proj}(X)$, and pass them along with language tokens to the frozen LLM with trainable LoRA layer. Next, dense point-cloud features $X' = f_{PB}(\mathcal{P})$ and affordance tokens from LLM $h_{aff} = f_{LLM}(Y, \mathcal{L})$ are extracted and passed to the affordance decoder f_{AFD} to produce an initial mask $\mathcal{M}_0$. The iterative affordance refinement module then refines the mask via T steps. At each iteration t, geometric descriptors $Z_t = f_{GFE}(\mathcal{G}(\mathcal{M}_{t-1}))$ (e.g., Laplacian, curvature) are computed (subsection 3.3) and injected as visual prompts, along with X' and h_{aff} to multi-stage refinement decoder f_{MG} (subsubsection 3.4.1) to produce the refined affordance prediction $\mathcal{M}_t$, after processing them through INAFS (subsubsection 3.4.2) and SCSP (subsubsection 3.4.3). The process converges after T steps, enabling precise, language-guided 3D affordance segmentation $\mathcal{M}_T$ through closed-loop geometric feedback.

to generate a response sequence that includes a special affordance token <AFF>. The hidden representation corresponding to <AFF>, denoted as h_{aff}, is then extracted. Meanwhile, the point cloud backbone f_{PB} computes dense point-wise features $X' = f_{\text{PB}}(\mathcal{P}) \in \mathbb{R}^{N \times c'}$. Finally, the affordance decoder f_{AFD} fuses h_{aff} with X' to produce the initial affordance mask $\mathcal{M}_0 = f_{\text{AFD}}(h_{\text{aff}}, X')$.

3.3 Iterative Differential Geometry-based Self-Prompting

Building upon the initial prediction $\mathcal{M}_0$, we introduce our core contribution: *Iterative Differential Geometry-Based Self-Prompting (IDGSP)*. This module leverages geometric feedback derived from prior affordance masks to progressively refine segmentation outputs. At each iteration t, we compute a set of differential geometric descriptors from the previous mask $\mathcal{M}_{t-1}$:

$$\mathcal{G}(\mathcal{M}_{t-1}) = \left\{\nabla^2 \mathcal{M}_{t-1},\ \nabla \mathcal{M}_{t-1} \cdot \mathbf{n},\ \mathcal{H}(\mathcal{M}_{t-1}),\ \kappa_1(\mathcal{M}_{t-1}),\ \kappa_2(\mathcal{M}_{t-1})\right\} \tag{2}$$

where $\nabla^2 \mathcal{M}_{t-1}$ denotes the Laplacian of the mask capturing local curvature variation, $\nabla \mathcal{M}_{t-1} \cdot \mathbf{n}$ represents the normal derivative quantifying alignment with surface normals, $\mathcal{H}(\mathcal{M}_{t-1})$ is the mean curvature of mask boundaries, and κ_1, κ_2 are the principal curvatures. These geometric signals encode essential boundary-aware and topological properties that reflect the physical plausibility of affordance regions. The extracted descriptors are transformed into a dense per-point representation $\mathcal{Z}_t \in \mathbb{R}^{N \times d}$ using a learnable geometric feature extractor f_{GFE}: $\mathcal{Z}_t = f_{\text{GFE}}\left(\mathcal{G}(\mathcal{M}_{t-1})\right)$. This geometry-driven prompt $\mathcal{Z}_t$ is then injected into the multi-stage refinement decoder (subsubsection 3.4.1) to guide the refinement process.

3.4 Iterative Affordance Refinement

3.4.1 Multi-Stage Refinement Decoder

To operationalize the geometric self-prompting mechanism, we employ a multi-stage refinement decoder that iteratively updates the affordance mask using both language and geometry-informed cues. At each iteration t, the geometric features $\mathcal{Z}_t$ are fused with the initial LLM embedding h_{aff} using a cross-attention mechanism:

$$h_{\text{aff}}^{(t)} = \text{CrossAttn}(h_{\text{aff}}, \mathcal{Z}_t) \tag{3}$$

The refined embedding $h_{\text{aff}}^{(t)}$ is then combined with dense point cloud features X' via a mask generation module f_{MG} to produce the updated affordance mask:

$$\mathcal{M}_t = f_{\text{MG}}(h_{\text{aff}}^{(t)}, X') \tag{4}$$

3.4.2 Implicit Neural Affordance Field Self-Supervision

To complement the discrete iterative refinement process with a smooth, continuous representation, we incorporate a regularization strategy based on implicit neural fields to enhance boundary precision and geometric consistency without relying on additional labels. This component learns a continuous implicit function $\Phi_\omega : \mathbb{R}^3 \times \mathbb{R}^d \to [0, 1]$, parameterized by ω, which maps any 3D point $\mathbf{x} \in \mathbb{R}^3$ and its corresponding feature vector to a scalar-valued affordance probability.

The function Φ_ω is trained via energy minimization loss $\mathcal{L}_{INAFS} = \mathcal{E}(\Phi_\omega)$ over the 3D spatial domain Ω, incorporating geometric priors and alignment with the current mask predictions. The energy term is defined as:

$$\mathcal{E}(\Phi_\omega) = \int_\Omega \|\nabla \Phi_\omega(\mathbf{x})\|^2 d\mathbf{x} + \lambda_1 \int_{\partial\Omega} \left(\Phi_\omega(\mathbf{x}) - \mathcal{M}(\mathbf{x})\right)^2 d\mathbf{x} + \lambda_2 \int_\Omega \left(|\Delta\Phi_\omega(\mathbf{x})| - \beta\|\kappa(\mathbf{x})\|\right)^2 d\mathbf{x} \tag{5}$$

Here, the first term encourages spatial smoothness by minimizing the gradient norm of the implicit field. The second term enforces fidelity to the current predicted mask $\mathcal{M}(\mathbf{x})$ at the boundary $\partial\Omega$, ensuring consistency with previously inferred affordances. The third term aligns the second-order variation of the field, measured by the Laplacian $\Delta\Phi_\omega$, with the Gaussian curvature $\kappa(\mathbf{x})$ (where $\kappa = \kappa_1 \cdot \kappa_2$), thereby promoting geometric conformity with intrinsic surface structures. The weighting parameters λ_1, λ_2, and scaling constant β balance the contributions of fidelity and curvature alignment. After optimization, the final affordance mask is extracted by thresholding the implicit field at 0.5:

$$\mathcal{M}_{\text{refined}} = \{\mathbf{x} \in \mathcal{P} \mid \Phi_\omega(\mathbf{x}) > 0.5\} \tag{6}$$

This implicit representation allows the model to refine coarse predictions into geometrically consistent and semantically plausible affordance regions, even in the absence of explicit supervision.

3.4.3 Spectral Convolutional Self-Prompting

To complement spatial refinement with a frequency-aware perspective, we introduce *Spectral Convolutional Self-Prompting (SCSP)*, which enables the model to capture affordance structures at multiple scales in the spectral domain of point cloud. We treat the 3D point cloud as a discrete manifold encoded by the normalized Laplacian operator $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{A}$ is the affinity matrix derived from local geometric similarity, and $\mathbf{D}$ is the corresponding degree matrix. Given the predicted affordance mask $\mathcal{M} \in \mathbb{R}^N$, we project it to spectral domain via eigen-decomposition:

$$\hat{\mathcal{M}} = \sum_{i=1}^{N} \alpha_i \mathbf{u}_i, \quad \text{where} \quad \alpha_i = \langle \mathcal{M}, \mathbf{u}_i \rangle \tag{7}$$

Here, $\{\mathbf{u}_i\}_{i=1}^N$ are the eigenvectors of $\mathbf{L}$, and $\{\alpha_i\}$ are the corresponding spectral coefficients. Refinement is performed by applying a learnable spectral filter $g(\lambda_i)$, parameterized over the eigenvalues $\{\lambda_i\}$, yielding the updated mask in the spectral domain:

$$\hat{\mathcal{M}}_{t+1} = \sum_{i=1}^{N} g(\lambda_i)\alpha_i^{(t)}\mathbf{u}_i \tag{8}$$

By operating in the spectral domain, SCSP provides a principled, resolution-aware mechanism for affordance enhancement without explicit hierarchical supervision. The entire refinement process is performed iteratively for T steps, allowing the model to progressively improve affordance localization by incorporating differential geometric feedback.

3.5 Overall Learning Strategy

To effectively address data scarcity and ensure robust affordance understanding, we adopt a multi-stage training strategy inspired by 3D-AffordanceLLM [6]. The pre-trained backbone is frozen, and we train the proposed self-prompting modules, including IDGSP, INAFS, and SCSP, to refine affordance masks using geometric and spectral cues. Our overall loss combines multitask objectives:

$$\mathcal{L} = \lambda_{\text{txt}}\mathcal{L}_{\text{txt}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{IDGSP}}\mathcal{L}_{\text{IDGSP}} + \lambda_{\text{INAFS}}\mathcal{L}_{\text{INAFS}} + \lambda_{\text{SCSP}}\mathcal{L}_{\text{SCSP}} \tag{9}$$

Here $\mathcal{L}_{\text{txt}}$ is autoregressive CE loss for LLM response generation, $\mathcal{L}_{\text{mask}}$ is BCE + Dice loss for initial affordance mask prediction, both used in the affordance backbone (following [6]). $\mathcal{L}_{\text{IDGSP}}$ penalizes

inconsistencies between iterative mask refinements and encourages smoothness, $\mathcal{L}_{\text{INAFS}}$ denotes energy-based regularization over the implicit affordance field (described in subsubsection 3.4.2), $\mathcal{L}_{\text{SCSP}}$ performs spectral consistency and spatial regularization using total variation TV. Specifically, $\mathcal{L}_{\text{IDGSP}}$ and $\mathcal{L}_{\text{SCSP}}$ are formulated as:

$$\mathcal{L}_{\text{IDGSP}} = \sum_{t=1}^{T} \lambda_t \|\mathcal{M}_t - \mathcal{M}_{t-1}\|_{W_{2,2}}^2 + \alpha \|\nabla^4 \mathcal{M}_T\|_2^2, \text{ and} \tag{10}$$

$$\mathcal{L}_{\text{SCSP}} = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_k \|W_k(\hat{\mathcal{M}}_t - \hat{\mathcal{M}}_{t-1})\|_F^2 + \tau \text{TV}(\mathcal{M}_T) \tag{11}$$

where in Equation 10, $\|\cdot\|_{W_{2,2}}^2$ is the Sobolev $W^{2,2}$ norm measuring the difference between consecutive masks, capturing both value differences and derivatives (geometric properties), λ_t is iteration-specific weight, ∇^4 is biharmonic operator, and $\alpha|\nabla^4 \mathcal{M}_T|_2^2$ is Tikhonov regularization term ensuring the final mask has smooth boundaries with controlled curvature. The first term in Equation 11 penalizes changes in the spectral components of the mask across iterations, enforcing frequency-consistent refinement, whereas the second term ensures that the final predicted mask $\mathcal{M}_T$ is spatially smooth, reducing over-segmentation and promoting contiguous affordance regions. $\sum_{t=1}^{T}$ is summation over all iterations of the self-prompting process (from 1 to T), $\sum_{k=1}^{K}$ is summation over K different frequency bands or scales of analysis, W_k is diagonal matrix that isolates the k-th frequency band, γ_k is scale-dependent weight coefficients for each frequency band k, $\hat{\mathcal{M}}_t$ is spectral decompositions of affordance masks at iteration t, $|\cdot|_F^2$ is Frobenius norm squared, measuring differences in frequency components, τ denotes weight parameter balancing spectral consistency and spatial coherence, and $\text{TV}(\mathcal{M}_T)$ is total variation regularizer promoting spatial smoothness in the final mask.

This multi-stage optimization pipeline enables the model to progressively refine affordance predictions by integrating linguistic reasoning with geometric and spectral feedback, leading to improved generalization and mask accuracy in open-world settings.

4 Experiments and Results

4.1 Dataset Description

Following previous works [5, 14], we conduct evaluations on two complementary 3D affordance datasets: PIAD [12] and LASO [14], each designed to test different aspects of generalization. PIAD serves as a complementary benchmark, comprising $7,012$ point clouds from the same object categories as LASO but introduces a stricter generalization setting—entire object instances are withheld from training, requiring the model to predict affordances on previously unseen geometries. As PIAD lacks textual annotations, we augment it with language instructions by sampling prompts from LASO's question pool, ensuring semantic alignment with each target affordance type. This design enables evaluation of our model's robustness in both instruction-conditioned and shape-driven generalization scenarios. LASO, on the other hand, contains $19,751$ language-guided point cloud pairs spanning $8,434$ unique object instances across 23 object categories and 17 affordance types. It supports both Seen and Unseen splits, where the Unseen configuration deliberately excludes specific affordance-object combinations during training to assess zero-shot generalization.

4.2 Implementation Details

Following 3D-AffordanceLLM [6], we utilize Phi-3.5-mini-instruct [27] as our base LLM with LoRA [28] fine-tuning. For 3D processing, we adopt Point-BERT [29] pre-trained with ULIP-2 [30] as our point encoder (f_{PE}) and Point Transformer [31] as our point backbone (f_{PB}). The feature dimension d is set to 512 for both language and point features. The projector layer (f_{proj}) is implemented as a simple linear layer

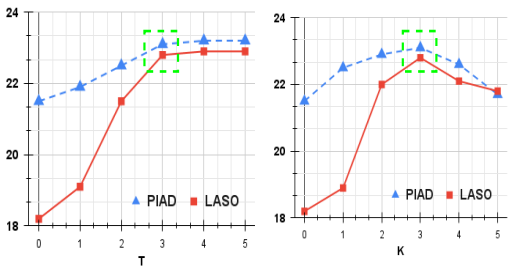

Figure 3: Performance analysis (aIoU on "seen" setting) with varying T and K values.

Table 1: Qualitative comparison of our proposed method on the PIAD (left) and LASO (right) datasets. The best and second-best results are highlighted in red and blue, respectively. LASO* indicates reported results of LASO [14] in GEAL [5]. † denotes our reproduced results of [6].

Type	Method	aIoU ↑	AUC ↑	SIM ↑	MAE ↓
Seen	MBDF [33]	9.3	74.9	0.415	0.143
	PMF [34]	10.1	75.1	0.425	0.141
	FRCNN [35]	12.0	76.1	0.429	0.136
	ILN [36]	11.5	75.8	0.427	0.137
	PFusion [37]	12.3	77.5	0.432	0.135
	XMF [38]	12.9	78.2	0.441	0.127
	IAGNet [12]	20.5	84.9	0.545	0.098
	LASO [14]	19.7	84.2	0.590	0.096
	3DAffLLM† [6]	21.5	82.6	0.643	0.104
	GEAL [5]	22.5	85.0	0.601	0.092
	Ours	**23.1**	**85.8**	**0.664**	**0.089**
Unseen	MBDF [33]	4.2	58.2	0.325	0.213
	PMF [34]	4.7	60.3	0.330	0.211
	FRCNN [35]	5.1	61.9	0.332	0.195
	ILN [36]	4.7	59.7	0.325	0.207
	PFusion [37]	5.3	61.9	0.33	0.193
	XMF [38]	5.7	62.6	0.342	0.186
	IAGNet [12]	8.0	71.8	0.352	0.127
	LASO [14]	8.0	69.2	0.386	0.118
	3DAffLLM† [6]	7.4	71.0	0.413	0.115
	GEAL [5]	8.7	72.5	0.390	0.102
	Ours	**9.2**	**73.1**	**0.431**	**0.099**

Type	Method	aIoU ↑	AUC ↑	SIM ↑	MAE ↓
Seen	ReferTrans [39]	13.7	79.8	0.497	0.124
	ReLA [40]	15.2	78.9	0.532	0.118
	3D-SPS [41]	11.4	76.2	0.433	0.138
	IAGNet [12]	17.8	82.3	0.561	0.109
	LASO [14]	20.8	87.3	0.629	0.093
	LASO* [14]	19.7	85.2	0.600	0.097
	3DAffLLM† [6]	18.2	84.9	0.622	0.104
	GEAL [5]	22.0	86.7	0.634	0.092
	Ours	**22.8**	**87.3**	**0.651**	**0.090**
Unseen	ReferTrans [39]	10.2	69.1	0.432	0.145
	ReLA [40]	10.7	69.7	0.429	0.144
	3D-SPS [41]	7.9	68.8	0.402	0.158
	IAGNet [12]	12.9	77.8	0.443	0.129
	LASO [14]	14.6	80.2	0.507	0.119
	LASO* [14]	15.6	79.9	0.549	0.108
	3DAffLLM† [6]	15.3	78.7	0.542	0.124
	GEAL [5]	16.7	80.9	0.567	0.106
	Ours	**17.1**	**81.5**	**0.571**	**0.103**

mapping point features to match the LLM token dimension. For the Affordance Decoder, we follow the architecture from LISA [32] but adapted for 3D data. For our iterative self-prompting mechanism, we set the number of refinement iterations $T = 3$ (as performance plateaus beyond this point while computational cost rises sharply (see Figure 3)), with weight parameters $\lambda_t = 0.8^t$ to gradually reduce consistency constraints. In the IDGSP loss, we set $\alpha = 0.1$ for the Tikhonov regularization term. For INAFS, we use $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, and $\beta = 0.05$. The SCSP module uses $K = 3$ frequency bands (following validation in Figure 3) with weights $\gamma_1 = 1.0$, $\gamma_2 = 0.7$, $\gamma_3 = 0.4$, and $\tau = 0.2$ for the total variation term. We use AdamW optimizer with an initial learning rate of 4×10^{-5} with cosine scheduling and warm-up ratio of 0.03. All experiments are done on four NVIDIA V100 GPU with a batch size of 16, training for 20 epochs in $\sim 12hr$.

4.3 Findings and Comparison with SoTA

Our proposed framework achieves consistent and substantial performance improvements across the PIAD benchmark, as shown in Table 1, setting a new state-of-the-art for language-guided 3D affordance detection. To ensure a fair and consistent comparison, we reproduced the results of 3D-AffordanceLLM [6] under our evaluation protocol and dataset setup, accounting for differences from the original implementation. In the seen configuration, our method achieves 23.1% aIoU, 85.8% AUC, and 0.664 SIM, outperforming the prior best (GEAL) by relative 2.66%, 0.92%, and 10.48%, respectively. In the unseen split—designed to evaluate zero-shot generalization to novel affordance-object pairs—we obtain 9.2% aIoU, 73.1% AUC, and 0.431 SIM, again surpassing GEAL by 0.5% in aIoU, 0.8% in AUC, and 4.1% in SIM. This improvement reflects our framework's ability to preserve spatial and structural coherence in predicted masks, particularly in ambiguous or underrepresented regions. Earlier fusion-based approaches like [33–38] exhibit significantly inferior performance due to their generic multimodal architectures that fail to model the specialized nature of affordance relationships. These methods are unable

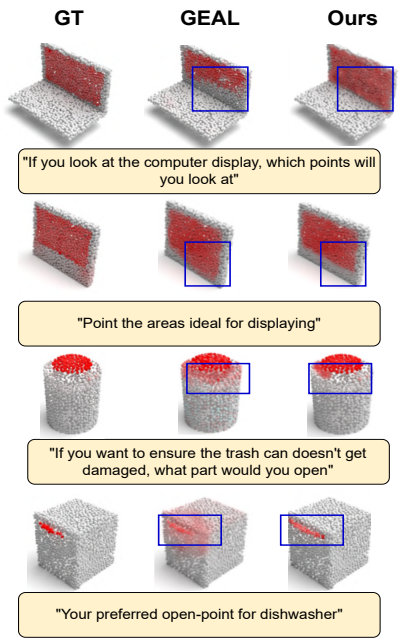

Figure 4: Qualitative comparison of our affordance segmentation results with GEAL [5].

to bridge the geometric-semantic gap, resulting in
substantial performance degradation (relative aIoU dropping by more than 50% compared to our method). Unlike prior methods such as [14, 6, 5] that operate in a single-pass decoding mode and rely heavily on textual embeddings, our method incorporates closed-loop refinement with geometric feedback, allowing it to resolve fine-grained boundaries in a context-aware manner.

Similar trends are observed on the LASO dataset, where our model achieves 22.8% aIoU, 87.3% AUC, and 0.651 SIM on seen objects, and 17.1% aIoU, 81.5% AUC, and 0.571 SIM in the more challenging unseen setting. Notably, in the unseen split, our framework outperforms the best baseline (GEAL) by 2.4% aIoU, 0.9% AUC, and 0.8% SIM relative. Traditional baselines such as [12, 26, 14] suffer huge degradation in performance when transitioning from seen to unseen due to their rigid, non-adaptive architectures. Even stronger models like [6, 5] exhibit sharp performance drops (e.g., GEAL: 22.0→16.7 aIoU on PIAD), revealing their limited ability to transfer learned affordance priors to unfamiliar topologies.

4.4 Ablation Study

The ablation study in Table 2 demonstrates the incremental contribution of each component in our self-prompting framework. When comparing the baseline (row 1) to the full model (row 4), we observe consistent performance improvements across all metrics and datasets, with particularly significant gains in the unseen settings.

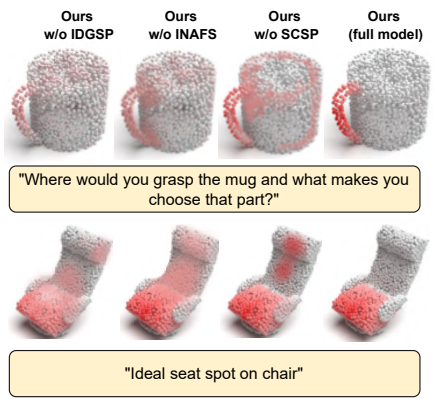

Figure 5: Qualitative visualization of ablation experiments.

(1) The *Spectral Convolutional Self-Prompting (SCSP)* module provides the initial performance lift (+0.6/+1.3 aIoU on PIAD/LASO seen), confirming the effectiveness of frequency-domain processing for capturing multi-scale affordance patterns. By operating in the spectral domain, SCSP enables the model to modulate signal components at different structural frequencies, processing both coarse affordance regions and fine boundary details simultaneously (shown in rows 1, 2 of Figure 4). **(2)** Adding *Implicit Neural Affordance Field Self-Supervision (INAFS)* yields further improvements (+0.8/+0.8 aIoU), particularly in structural similarity metrics. This suggests that the continuous implicit field representation enhances boundary precision and produces geometrically coherent affordance regions. The implicit field's ability to capture smooth transitions and model topological relationships proves especially beneficial for complex object geometries, as evident in row 3 of Figure 4. **(3)** The most substantial gains come from incorporating *Iterative Differential Geometry-Based Self-Prompting (IDGSP)*, which provides a significant boost on LASO seen (+2.5 aIoU) and notable improvements across unseen scenarios. This demonstrates that leveraging geometric features (Laplacians, curvatures, normal derivatives) as visual prompts enables the model to progressively refine affordance boundaries through geometric feedback, particularly crucial for distinguishing fine-grained functional regions (row 4 of Figure 4). The synergistic effect of all three components is most pronounced in generalization scenarios, where the relative improvements are larger for unseen settings than seen settings. This confirms our hypothesis that geometric self-prompting enhances the model's ability to adapt to novel object-affordance relationships by leveraging intrinsic geometric cues rather than relying solely on seen training examples. Qualitative visualization is provided in Figure 5. Additional findings can be found in the supplementary file.

4.5 Cross-dataset Generalization

Table 3 highlights the cross-dataset generalization performance when models trained on LASO are evaluated on PIAD. Our full model outperforms prior state-of-the-art (GEAL) across all metrics, achieving 19.7%/12.5% aIoU, 84.5%/75.2% AUC, and 0.610/0.465 SIM in seen/unseen splits—marking up to +1.3% aIoU and +0.025 SIM improvements.

Table 2: Ablation study of different components. The best results are in **bold**.

Type	IDGSP	INAFS	SCSP	PIAD aIoU	PIAD AUC	PIAD SIM	PIAD MAE	LASO aIoU	LASO AUC	LASO SIM	LASO MAE
Seen	✗	✗	✗	21.5	82.6	0.643	0.104	18.2	84.9	0.622	0.104
	✗	✗	✓	22.1	83.5	0.650	0.099	19.5	85.4	0.631	0.099
	✗	✓	✓	22.9	84.2	0.657	0.093	20.3	86.1	0.643	0.094
	✓	✓	✓	**23.1**	**85.8**	**0.664**	**0.089**	**22.8**	**87.3**	**0.651**	**0.090**
Unseen	✗	✗	✗	7.4	71.0	0.413	0.115	15.3	78.7	0.542	0.124
	✗	✗	✓	8.0	72.1	0.420	0.109	16.0	79.3	0.558	0.116
	✗	✓	✓	8.5	72.5	0.429	0.105	16.5	80.7	0.566	0.110
	✓	✓	✓	**9.2**	**73.1**	**0.431**	**0.099**	**17.1**	**81.5**	**0.571**	**0.103**

Table 3: Cross-dataset generalization (LASO→PIAD)

Method	Seen aIoU	Seen AUC	Seen SIM	Unseen aIoU	Unseen AUC	Unseen SIM
3DAffLLM [6]	17.6	82.4	0.57	10.8	72.5	0.425
GEAL [5]	18.4	83.2	0.59	11.6	73.8	0.44
Ours w/o SCSP	18.9	83.6	0.595	11.8	74	0.445
Ours (Full Model)	**19.7**	**84.5**	**0.61**	**12.5**	**75.2**	**0.465**

The ablation variant without SCSP shows a clear drop (-0.8% seen, -0.7% unseen aIoU), confirming SCSP's role in learning transferable, multi-scale affordance cues. Even without it, our model still outperforms [5, 6], validating the strength of our differential geometry-based self-prompting. All methods exhibit performance degradation in unseen scenarios, but our model maintains the smallest drop, indicating greater robustness to distribution shifts—enabled by shape-aware reasoning rather than dataset-specific memorization.

5 Conclusion and Future Works

We presented *ViSPLA*, a geometry-aware iterative framework for language-guided 3D affordance detection. By combining differential geometric self-prompting, implicit neural fields, and spectral refinement, our model progressively improves affordance segmentation beyond the constraints of single-pass or fixed-label paradigms. Experimental results on LASO and PIAD benchmarks show strong generalization, particularly in zero-shot and cross-dataset settings. Despite its strengths, *ViSPLA* incurs additional computation due to iterative refinement and may face challenges with highly deformable or articulated objects. Future work will explore hybrid prompting strategies that combine geometric and learned latent cues, as well as adaptive iteration control for real-time efficiency. Extending to dynamic scenes and scene-level affordance reasoning is another promising direction. Together, these developments will move us closer to robust, generalizable affordance understanding in complex real-world environments—paving the way for more capable and adaptable embodied agents.

Acknowledgements

The authors have been supported by the following NSF grants: IIS-2331769, CMMI-2246673, and ECCS-2025929.

References

[1] Gibson, J. J. *The ecological approach to visual perception: classic edition*. Psychology press, 2014.

[2] Yu, C., H. Wang, Y. Shi, et al. Seqafford: Sequential 3d affordance reasoning via multimodal large language model. *arXiv preprint arXiv:2412.01550*, 2024.

[3] Do, T.-T., A. Nguyen, I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018.

[4] Deng, S., X. Xu, C. Wu, et al. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787. 2021.

[5] Lu, D., L. Kong, T. Huang, et al. Geal: Generalizable 3d affordance learning with cross-modal consistency. *arXiv preprint arXiv:2412.09511*, 2024.

[6] Chu, H., X. Deng, Q. Lv, et al. 3d-affordancellm: Harnessing large language models for open-vocabulary affordance detection in 3d worlds. *arXiv preprint arXiv:2502.20041*, 2025.

[7] Liu, A., B. Feng, B. Xue, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[8] Lu, L., W. Zhai, H. Luo, et al. Phrase-based affordance detection via cyclic bilateral interaction. *IEEE Transactions on Artificial Intelligence*, 4(5):1186–1198, 2022.

[9] Chen, J., D. Gao, K. Q. Lin, et al. Affordance grounding from demonstration video to target image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6799–6808. 2023.

[10] Li, G., V. Jampani, D. Sun, et al. Locate: Localize and transfer object parts for weakly supervised affordance grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10922–10931. 2023.

[11] Mo, K., S. Zhu, A. X. Chang, et al. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918. 2019.

[12] Yang, Y., W. Zhai, H. Luo, et al. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10905–10915. 2023.

[13] Nguyen, T., M. N. Vu, A. Vuong, et al. Open-vocabulary affordance detection in 3d point clouds. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5692–5698. IEEE, 2023.

[14] Li, Y., N. Zhao, J. Xiao, et al. Laso: Language-guided affordance segmentation on 3d object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14251–14260. 2024.

[15] Chu, M., X. Zhang. Iris: Interactive responsive intelligent segmentation for 3d affordance analysis. *arXiv e-prints*, pages arXiv–2409, 2024.

[16] Basak, H., Z. Yin. Semidavil: Semi-supervised domain adaptation with vision-language guidance for semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9816–9828. 2025.

[17] Basak, H., H. Tabatabaee, S. Gayaka, et al. Enhancing single image to 3d generation using gaussian splatting and hybrid diffusion priors. *arXiv preprint arXiv:2410.09467*, 2024.

[18] Alayrac, J.-B., J. Donahue, P. Luc, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[19] Li, J., D. Li, S. Savarese, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[20] Liu, S., Z. Zeng, T. Ren, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[21] Liu, H., C. Li, Q. Wu, et al. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[22] Xu, R., X. Wang, T. Wang, et al. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.

[23] Qi, Z., R. Dong, S. Zhang, et al. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer, 2024.

[24] Wang, Z., H. Huang, Y. Zhao, et al. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

[25] Chen, S., X. Chen, C. Zhang, et al. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438. 2024.

[26] Hong, Y., H. Zhen, P. Chen, et al. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[27] Abdin, M., J. Aneja, H. Awadalla, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[28] Hu, E. J., Y. Shen, P. Wallis, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[29] Yu, X., L. Tang, Y. Rao, et al. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322. 2022.

[30] Xue, L., N. Yu, S. Zhang, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27091–27101. 2024.

[31] Zhao, H., L. Jiang, J. Jia, et al. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268. 2021.

[32] Lai, X., Z. Tian, Y. Chen, et al. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589. 2024.

[33] Tan, X., X. Chen, G. Zhang, et al. Mbdf-net: Multi-branch deep fusion network for 3d object detection. In *Proceedings of the 1st International Workshop on Multimedia Computing for Urban Data*, pages 9–17. 2021.

[34] Zhuang, Z., R. Li, K. Jia, et al. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16280–16290. 2021.

[35] Xu, X., S. Dong, T. Xu, et al. Fusionrcnn: Lidar-camera fusion for two-stage 3d object detection. *Remote Sensing*, 15(7):1839, 2023.

[36] Chen, H., Z. Wei, Y. Xu, et al. Imlovenet: Misaligned image-supported registration network for low-overlap point cloud pairs. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–9. 2022.

[37] Xu, D., D. Anguelov, A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253. 2018.

[38] Aiello, E., D. Valsesia, E. Magli. Cross-modal learning for image-guided point cloud shape completion. *Advances in Neural Information Processing Systems*, 35:37349–37362, 2022.

[39] Li, M., L. Sigal. Referring transformer: A one-step approach to multi-task visual grounding. *Advances in neural information processing systems*, 34:19652–19664, 2021.

[40] Liu, C., H. Ding, X. Jiang. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23592–23601. 2023.

[41] Luo, J., J. Fu, X. Kong, et al. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16454–16463. 2022.

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 articulated in both the abstract and introduction precisely encapsulate the technical contributions and scope of this work. Each claim is substantiated in the main body through formal mathematical exposition and comprehensive empirical evaluation.

   Guidelines:

   - The answer NA 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 NA 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: A few limitations of our work are included in the Conclusion and Future Works section and detailed elaborations are provided in Supplementary File. We briefly discuss the possible limitations and how our future work could address them.

   Guidelines:

   - The answer NA 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: [NA]

Justification: We do not include any theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the experimental details are provided in sufficient detail, ensuring reproducibility.

   Guidelines:

   - The answer NA 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: [No]

Justification: Although the datasets used in the paper are open-sourced, we do not release the code for this work. However, sufficient details are provided in the paper to ensure reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental settings are provided in section 4.2: Implementation Details.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: No statistical analysis is provided.

   Guidelines:

   - The answer NA 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: Details are provided in section 4.2: Implementation Details.

   Guidelines:

   - The answer NA 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 reviewed and accepted the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: There are many robotics applications that are discussed, however, no negative societal impact could be found, hence, not discussed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: Proper citations are provided for all the existing assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs are not used.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.