VoxAfford: Multi-Scale Voxel-Token Fusion for Open-Vocabulary 3D Affordance Detection

Haowen Sun, Shaolong Zhang, Mingyang Li, Chengzhong Ma, Xinzhe Chen
Qiongjie Cui, Xingyu Chen, Zeyang Liu*, Xuguang Lan*
National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, Xi'an, China

Abstract: Open-vocabulary 3D affordance detection requires localizing interaction regions on point clouds given novel affordance descriptions. Recent methods extend multimodal large language models (MLLMs) with special output tokens that are decoded into segmentation masks. However, these tokens are produced through autoregressive generation, which models sequential dependencies rather than spatial neighborhood relations, leaving them semantically rich but spatially impoverished for 3D localization. We propose **Vox**el-enhanced **Afford**ance detection (**VoxAfford**), which bypasses this bottleneck by injecting multi-scale geometric features from a frozen pre-trained 3D VQVAE encoder into the output tokens after generation. Each output token uses its affordance semantics as a query to retrieve relevant geometric patterns from its paired voxel scale via cross-attention, with a learned compatibility gate controlling the injection strength. The enhanced tokens are then aggregated into a spatially-aware affordance prompt through semantic-conditioned attention and propagated alongside per-point features to generate the final mask. Experiments on open-vocabulary affordance detection tasks show that VoxAfford achieves state-of-the-art performance with an approximately 10% absolute improvement in instance-level mIoU, and real robot experiments confirm zero-shot transfer to novel objects.

Keywords: 3D Affordance Detection, Large Language Models, Open-Vocabulary

1 Introduction

Affordance detection identifies regions on objects that support specific interactions [1, 2, 3, 4]. Unlike semantic segmentation, where labels are primarily determined by object identity, affordance labels arise from the joint consideration of task intent, object properties, and local geometric configurations. This task-conditioned geometric reasoning makes affordance fundamentally harder than category-level labeling. 3D point clouds preserve the geometric structure needed for this reasoning and are therefore widely adopted for affordance detection [5, 6, 7]. Building on this representation, most existing methods train per-point classifiers on a fixed set of affordance categories [8, 9, 10]. These methods cannot generalize to affordance types

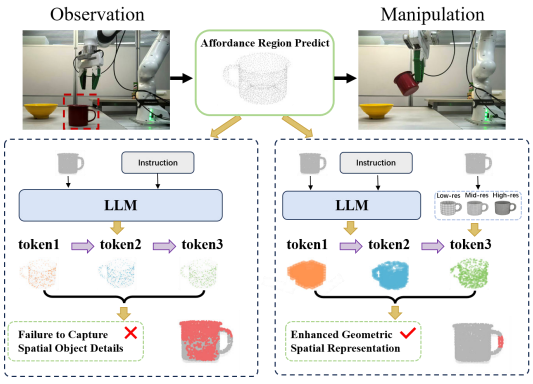


Figure 1: **MLLM-based affordance region prediction for robotic manipulation**.

*Corresponding authors: Zeyang Liu (zeyang.liu@xjtu.edu.cn) and Xuguang Lan (xglan@mail.xjtu.edu.cn).

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

absent from the training set. Open-vocabulary affordance detection removes this restriction by accepting free-form language queries, but it also introduces a new challenge: the model must determine what geometric conditions a novel affordance requires and locate the corresponding regions on the object.

Recent methods adopt multimodal large language models (MLLMs) to enable open-vocabulary affordance detection (Figure 1) [11, 12, 13, 14], but the output tokens that guide mask prediction carry limited geometric information about the target region. In these methods, the LLM output vocabulary is extended with special tokens whose hidden representations are decoded into segmentation masks by a downstream decoder [15, 16, 17]. Some methods enrich the 3D input to the LLM with stronger encoders so that the model receives more geometric detail [18, 19, 20, 21]. Others apply auxiliary spatial losses on the output tokens to encourage them to encode positional information [22, 23]. However, both approaches require geometric information to pass through the autoregressive generation process before reaching the output tokens. The output tokens therefore remain spatially impoverished regardless of how much geometric information is supplied at the input or how the output is supervised.

We attribute this limitation to a structural mismatch between autoregressive generation and 3D geometry. Autoregressive models build representations along a sequential token order, whereas 3D geometric properties are determined by spatial neighborhood relations. The language embedding space therefore lacks an inductive bias for encoding spatial topology. With sufficient data, autoregressive models can in principle acquire spatial structure, as shown in large-scale image generation [24, 25, 26, 27]. However, 3D affordance datasets are orders of magnitude smaller, making it impractical to learn spatial representations through fine-tuning alone. A more practical alternative is to supply pre-learned geometric representations from an external source. This shifts the learning problem from acquiring geometric representations to aligning them with affordance semantics. Such alignment is especially critical in open-vocabulary settings, where the decoder must generalize to novel affordance types and cannot rely on memorized associations from training.

In this paper, we propose **Vox**el-enhanced **Afford**ance detection (**VoxAfford**), which injects hierarchical voxel features into the LLM output tokens after generation. A frozen pre-trained 3D VQVAE encoder [28, 29, 30] extracts geometric features at three resolutions (16^3, 32^3, 64^3), capturing complementary cues from global topology to local surface detail. Each output token serves as a semantic query that retrieves relevant patterns from its paired scale via cross-attention, while a learned gate measures the compatibility between affordance semantics and retrieved geometry to control the injection strength. The enhanced tokens are aggregated and propagated to a per-point decoder for mask prediction. Experiments show that VoxAfford achieves state-of-the-art performance with an approximately 10% absolute improvement in instance-level mIoU on open-vocabulary benchmarks, and real robot experiments confirm transfer to novel objects under both full and partial observations.

2 Related Work

2.1 3D Affordance Detection

Affordance [31] characterizes how objects support agent interactions. Early methods predict pixel-level regions from RGB images [32, 33, 34]; 3D AffordanceNet [8] extends this to per-point heatmaps on point clouds, followed by work on actionable parts [35, 36], inter-category graph reasoning [37], language–point alignment [38], and partial-view generalization [5, 2]. All operate under closed-set taxonomies. For open-vocabulary queries, early attempts map 3D features to 2D VLM spaces [39, 40, 13], while recent methods reformulate detection as instruction-conditioned reasoning with 3D MLLMs [41, 42]: 3D-AffordanceLLM [6] decodes masks from a dedicated output token, and SeqAfford [15] introduces sequential tokens for multi-region affordances but relies on additional reasoning prompts to drive token generation, whereas our method operates directly on the affordance query without any prompt engineering. While these designs inherit the LLM's zero-shot reasoning [43, 44], the mask-guiding tokens live entirely in the language space and carry limited

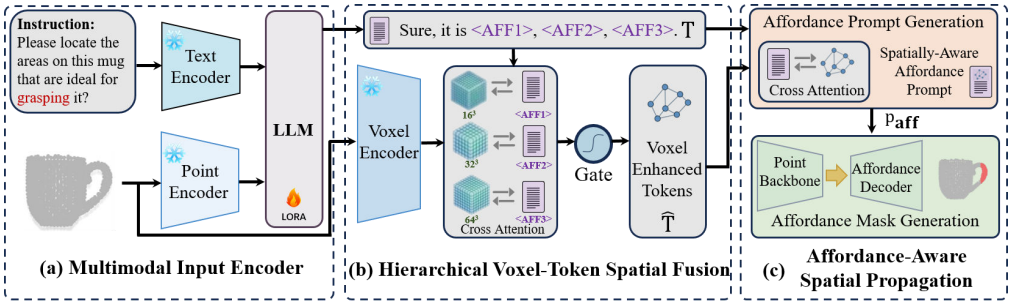


Figure 2: **Overview of the VoxAfford framework**. (a) Multimodal Input Encoder. The Text Encoder and Point Encoder feed the query and point cloud into a LoRA-finetuned LLM that generates affordance tokens. (b) Hierarchical Voxel-Token Spatial Fusion. The Voxel Encoder extracts multi-scale features from the voxelized point cloud, and cross-attention fuses them with the affordance tokens to produce Voxel-Enhanced Tokens. (c) Affordance-Aware Spatial Propagation. A semantic query derived from the original LLM tokens attends over the Voxel-Enhanced Tokens to produce a Spatially-Aware Affordance Prompt. The prompt and point features conditioned on the original tokens jointly drive mask prediction.

3D spatial information. We address this by injecting geometric representations into the tokens after generation, bypassing the autoregressive bottleneck rather than relying on richer inputs.

2.2 Multimodal LLMs for 3D Understanding

Following strong 2D MLLMs [45, 46, 47, 48], PointLLM [42] and 3D-LLM [41] project point cloud features into the LLM space for 3D QA and captioning. For dense prediction, prior work strengthens either the input or the output. On the input side, PointNet++ [49] and PointBERT [50] encode local geometry from raw coordinates, Uni3D [19] and Recon++ [18] learn transferable representations via large-scale pre-training, and ShapeLLM [29] feeds hierarchical VQVAE [28] voxel features as LLM input tokens. On the output side, methods decode special tokens into masks [23, 22, 29]. However, geometric information still passes through the autoregressive bottleneck. We incorporate multi-scale voxel features into output tokens via cross-attention, alleviating this spatial bottleneck.

3 Method

Given an object point cloud $P_c \in \mathbb{R}^{N\times 3}$ and a natural language affordance query Q_a, VoxAfford predicts a per-point binary mask $M_a \in \{0,1\}^N$ over the queried region. Figure 2 illustrates the architecture. The semantic path passes Q_a and P_c through a text encoder and a point encoder f_{pe}, respectively, into a LoRA-finetuned LLM that autoregressively generates K affordance tokens. The geometric path voxelizes P_c and feeds it to a frozen pre-trained VQVAE encoder, whose hierarchical downsampling yields features at three resolutions, from global shape topology to local surface detail. The hierarchical voxel-token spatial fusion module (Section 3.1) injects these features into the affordance tokens via cross-attention with a learned compatibility gate, producing voxel-enhanced tokens that carry 3D structure without traversing autoregressive generation. The affordance-aware spatial propagation module (Section 3.2) then generates a spatial prompt from the voxel-enhanced tokens and combines it with dense point features from a point backbone f_{PB} to drive an affordance decoder f_{AFD} for mask prediction. Section 3.3 describes the two-stage training.

3.1 Hierarchical Voxel-Token Spatial Fusion

The goal of voxel-token fusion is to let each affordance token selectively retrieve geometric patterns from the voxel features based on its affordance semantics, so that different affordance queries attend to different spatial regions on the same object. The voxel features encode 3D geometry in a structured grid that preserves spatial neighborhood relations absent in unordered point clouds, but they

are agnostic to the queried affordance. The fusion therefore combines the token's semantic specificity with the voxel features' spatial structure through multi-scale cross-attention retrieval, followed by selective injection via a learned compatibility gate.

Multi-scale voxel feature extraction and retrieval. Affordance prediction requires reasoning over geometric relations at multiple granularities. For example, localizing a graspable region relies more on local surface detail, while identifying a sittable surface depends more on global shape understanding. A single voxel resolution cannot provide both. We extract features at three resolutions (16^3, 32^3, 64^3) from the hierarchical downsampling path of a frozen pre-trained 3D VQVAE encoder. The large-scale shape reconstruction pre-training provides geometric priors that transfer to novel objects. These features capture geometric information at different spatial granularities.

Each affordance token is paired with one voxel resolution and retrieves geometric patterns from its paired scale through cross-attention:

$$\mathbf{g}_j = \text{CrossAttention}(\mathbf{t}_j,\, \phi_j(\mathbf{V}_j),\, \phi_j(\mathbf{V}_j)), \quad j \in \{1, 2, 3\} \tag{1}$$

where $\mathbf{t}_j$ is the j-th LLM affordance token, $\mathbf{V}_j$ denotes the voxel features at the j-th resolution, and ϕ_j is a per-scale linear projection. Different affordance queries produce different attention distributions over the same voxel grid, retrieving geometric patterns relevant to the queried interaction.

Selective geometric injection. The retrieved geometric patterns are not all relevant to the queried affordance, and injecting irrelevant geometry would add noise. We therefore gate the injection by a scalar $\alpha_j = \sigma(\mathbf{w}_j^\top [f_s(\mathbf{t}_j);\, f_g(\mathbf{g}_j)] + b_j)$ that measures the compatibility between the affordance semantics and the retrieved geometry, and update the token as

$$\hat{\mathbf{t}}_j = \mathbf{t}_j + \alpha_j \mathbf{W}_j \mathbf{g}_j, \tag{2}$$

where σ is the sigmoid function, f_s and f_g are linear projections, and $\mathbf{w}_j$, b_j, $\mathbf{W}_j$ are per-scale learnable parameters. The fixed pairing between each token and its voxel resolution provides an inductive bias that drives multi-scale specialization without explicit supervision. We denote the resulting set $\hat{\mathbf{T}} = \{\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \hat{\mathbf{t}}_3\}$ as the voxel-enhanced tokens.

3.2 Affordance-Aware Spatial Propagation

The voxel-token fusion module produces voxel-enhanced tokens $\hat{\mathbf{T}} = \{\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \hat{\mathbf{t}}_3\}$ that carry both affordance semantics and multi-scale geometric information. The point backbone f_{PB} independently extracts dense point embeddings $\mathbf{F}$ from the input point cloud. The spatial propagation module bridges these two representations to produce the per-point affordance mask.

We observe that the voxel-enhanced tokens and the original LLM tokens $\mathbf{T}$ are suited for different roles in this process. The enhanced tokens $\hat{\mathbf{T}}$ encode spatial awareness of where the affordance region is likely located through their injected geometric structure, making them effective for generating a global spatial prompt that guides the decoder toward the correct region. The original tokens $\mathbf{T}$ preserve the LLM's semantic understanding of what the queried affordance means, making them better suited for conditioning individual point features with affordance semantics. If the roles were reversed, the prompt would lack spatial information for localization, while the injected geometric signals would conflict with the point backbone's own spatial encoding. We therefore design a dual-pathway propagation that routes each token type to its appropriate role.

Affordance prompt generation. Different affordance types rely on different spatial scales, so the aggregation of voxel-enhanced tokens should be conditioned on the affordance semantics rather than using a fixed weighting scheme. We compute an affordance query from the mean of the original LLM tokens and attend over the voxel-enhanced tokens to produce the affordance prompt:

$$\mathbf{p}_{\text{aff}} = \text{LN}\Big(\mathbf{q}_{\text{aff}} + \text{CrossAttention}(\mathbf{q}_{\text{aff}},\, \hat{\mathbf{T}},\, \hat{\mathbf{T}})\Big), \quad \mathbf{q}_{\text{aff}} = f_q(\frac{1}{K}\sum_{j=1}^{K} \mathbf{t}_j), \tag{3}$$

where K is the number of affordance tokens, f_q is a learnable projection, and LN denotes layer normalization. Through this attention mechanism, a grasp query naturally assigns higher weight to

the 64^3 token for fine-grained surface detail, while a sit query attends more to the 16^3 token for global shape context.

Affordance mask generation. The affordance prompt provides global spatial guidance but does not modify individual point representations. To condition each point with affordance semantics, we inject the original LLM tokens $\mathbf{T}$ into the dense point features $\mathbf{F}$ through cross-attention with a zero-initialized residual connection:

$$\hat{\mathbf{F}} = \mathbf{F} + f_{\text{zero}}(\text{CrossAttention}(\mathbf{F}, \mathbf{T}, \mathbf{T})), \tag{4}$$

where f_{zero} is a linear projection initialized with zero weights, following the same principle as the compatibility gate in Section 3.1. A two-way transformer decoder f_{AFD} [51] then produces the per-point mask:

$$M_a = f_{\text{AFD}}(\mathbf{p}_{\text{aff}}, \hat{\mathbf{F}}). \tag{5}$$

3.3 Training Strategy

Our goal is to predict per-point affordance masks for natural language queries, including affordance types absent from training. This requires both basic segmentation ability and affordance-specific spatial reasoning. We therefore adopt a two-stage strategy. The first stage pre-trains the point backbone and decoder on a part segmentation task. The second stage trains the full model on affordance data, where the voxel-token fusion and spatial propagation modules learn to inject and propagate geometric information.

Segmentation pre-training. We pre-train the point backbone f_{PB} and the affordance decoder f_{AFD} on PartNet [52], which provides part-level annotations for 3D objects. Given a text description of an object part and the corresponding point cloud, the model predicts a binary segmentation mask of the described part. The point encoder, VQVAE encoder, and LLM backbone stay frozen. This stage is necessary because affordance training data is limited, making it hard for the point backbone and decoder to learn basic segmentation and affordance-specific spatial reasoning jointly from scratch. Pre-training first establishes segmentation capability, so the affordance stage can focus on aligning affordance semantics with geometric structure. The pre-trained weights of f_{PB} and f_{AFD} initialize the next stage.

Affordance training with voxel-token fusion. The full model is trained on affordance data with the point encoder, VQVAE encoder, and LLM backbone frozen. The LLM is adapted with LoRA [53], and the voxel-token fusion module, the spatial propagation module, f_{PB}, and f_{AFD} are trainable. The training objective is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{BCE}} + \lambda_3 \mathcal{L}_{\text{Dice}}, \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ is the autoregressive cross-entropy loss for text generation, $\mathcal{L}_{\text{BCE}}$ is the per-point binary cross-entropy loss, and $\mathcal{L}_{\text{Dice}}$ is the dice loss for mask prediction.

4 Experiments

We design experiments to answer three questions: (1) Can VoxAfford achieve open-vocabulary zero-shot affordance prediction and transfer to real robot manipulation on novel objects? (2) Do the three tokens capture geometric relations at distinct spatial granularities? (3) Does the spatial deficiency originate from the autoregressive generation process rather than insufficient input encoding, and are the component designs justified?

4.1 Experimental Setup

Datasets and metrics. PartNet [52] provides 573,585 part instances across 25,571 3D models and 24 categories for segmentation pre-training. Affordance training uses question-answer pairs constructed from 3D AffordanceNet [8]. We evaluate on OpenAD under the open-set and closed-set splits from [8], where the open-set split holds out test affordance categories from training. We report

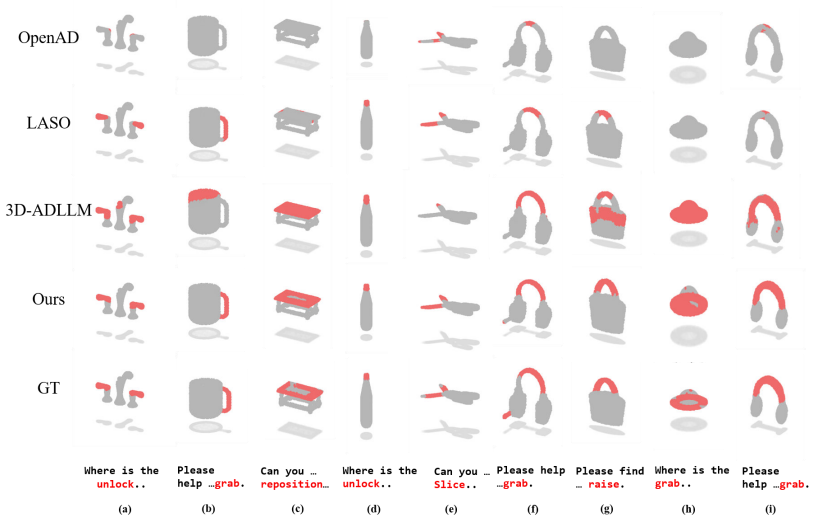


Figure 3: The visualization results of our VoxAfford compared with others.

Table 1: Zero-shot open-vocabulary affordance detection results on OpenAD. We report class-level and instance-level metrics under full-view and partial-view settings. Best results are in **bold**. † denotes 3D-ADLLM equipped with the same pretraining stack as VoxAfford (PartNet segmentation pretraining and ShapeLLM-Omni VQVAE features added at the LLM input).

Method	View	Class-level			Instance-level				
		mIoU^c	Acc^c	mAcc^c	mIoU^i	mAcc^i	mPrec^i	mRec^i	mAP^i_{50}
OpenAD-PointNet++	Full	13.53	3.97	16.40	3.46	74.59	11.84	5.84	0.02
	Partial	11.29	2.41	13.88	2.17	71.97	5.64	3.74	0.02
OpenAD-DGCNN	Full	11.15	3.84	13.86	3.79	74.42	11.13	6.67	0.04
	Partial	8.04	1.58	9.85	2.08	72.00	6.65	3.40	0.02
IAGNet	Full	16.16	19.07	23.92	11.13	74.36	24.58	19.23	2.80
	Partial	14.36	16.90	21.73	9.36	71.25	23.36	14.58	1.35
LASO	Full	22.41	15.90	30.22	20.47	71.47	37.95	34.93	2.42
	Partial	20.06	8.80	26.84	11.46	72.14	32.70	16.49	0.70
3D-ADLLM	Full	30.43	29.36	47.78	30.28	70.66	40.89	55.93	27.80
	Partial	27.25	27.87	39.04	28.72	68.28	41.71	47.73	25.63
3D-ADLLM†	Full	32.33	33.01	52.61	32.60	73.21	44.69	57.39	35.13
	Partial	29.13	30.14	42.34	33.05	66.64	40.19	**68.34**	34.98
VoxAfford (ours)	Full	**35.47**	**48.05**	**61.88**	**40.57**	**77.68**	**51.69**	**65.68**	**40.67**
	Partial	**31.87**	**49.41**	**62.18**	**38.00**	**72.19**	**45.84**	66.42	**37.57**

class-level metrics (mIoU^c, Acc^c, mAcc^c) averaged across affordance categories and instance-level metrics (mIoU^i, mAcc^i, mPrec^i, mRec^i, mAP^i_{50}) averaged across test instances. mAP^i_{50} uses 50% IoU as the matching threshold.

Baselines and implementation. We compare against recent zero-shot 3D affordance methods, including OpenAD with PointNet++ and DGCNN backbones [54], IAGNet [37], LASO [38], and 3D-ADLLM [16]. The base LLM is Phi-4-mini-instruct [55], the point encoder is PointBERT [50] pre-trained with ULIP-2 [56] on ModelNet [57], and the segmentation backbone is Point Transformer [58]. We fine-tune the LLM with LoRA at rank 32 and alpha 48. The VQVAE encoder loads from the ShapeLLM-Omni [30] checkpoint and stays frozen during training.

4.2 Comparison Results

Table 1 reports OpenAD open-set zero-shot results. VoxAfford outperforms all baselines on localization-sensitive metrics (mIoU and mAP_{50}) under both full-view and partial-view, while per-point accuracy (mAcc^i) remains comparable to the strongest baseline. This pattern matches our

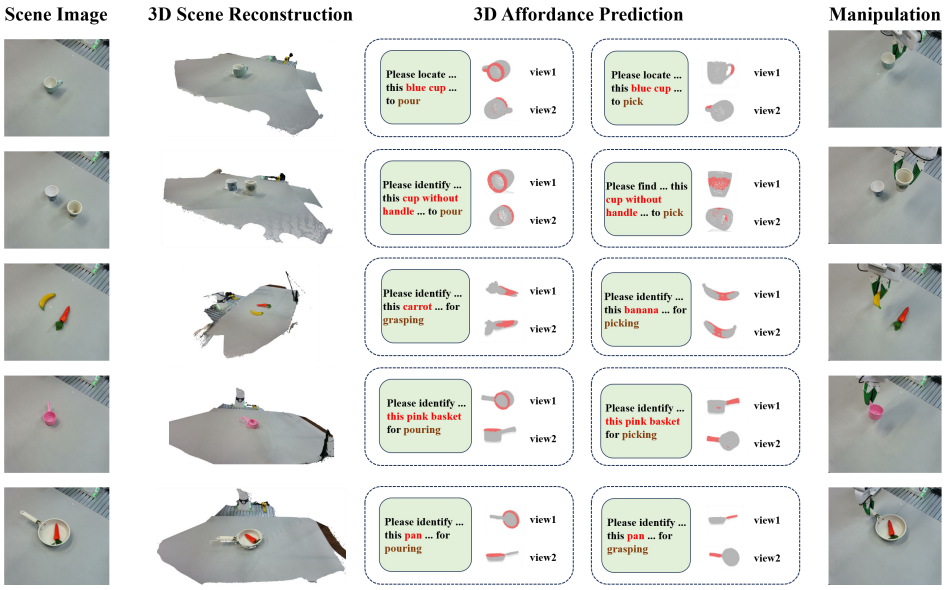


Figure 4: **Real world tasks**. The experimental task setup evaluated within five real environments.

voxel-token fusion design. The injected voxel features supply explicit 3D structure for recovering complete affordance regions, while per-point classification already benefits from LLM semantics and gains less from geometric injection. The hierarchical VQVAE features preserve global shape topology even under incomplete observations, so the fusion module still provides reliable cues in partial-view. To isolate the contribution of the fusion design from that of stronger pretraining components, we further evaluate 3D-ADLLM†, a controlled variant equipped with the same pretraining stack as VoxAfford (PartNet segmentation pretraining and ShapeLLM-Omni VQVAE features at the LLM input) but without our post-generation fusion. The substantial gap between 3D-ADLLM† and VoxAfford confirms that the gains are attributable to the post-generation voxel-token fusion rather than to backbone or pretraining differences. Table 2 further evaluates on AffordPose [59], where affordance-object combinations are absent from training. VoxAfford outperforms all baselines on every metric. The model never saw these pairings, so the gain indicates that the cross-attention fusion learns transferable affordance-geometry alignment rather than memorized associations.

4.3 Ablation Study

Ablation on input modality and LLM backbone. All ablations use the open-set zero-shot benchmark under full-view, instance-level evaluation. Variants share the same training procedure and hyperparameters. Table 3 shows that enriching the LLM input with voxel features cannot resolve the spatial deficiency. Both the voxel-only and concatenation variants force geometric information through autoregressive generation, which fails to preserve spatial neighborhood relations. Geometric injection therefore must happen after generation. The fusion also remains effective when Phi-4 [55] is replaced with Qwen3 [60], confirming that the design is not tied to a specific LLM.

Table 2: Zero-shot results on the AffordPose dataset.

Method	mIoU^{i}	mAcc^{i}	mPrec^{i}	mRec^{i}	mAP$^{i}_{50}$
OpenAD-PointNet++	7.61	21.36	64.13	13.01	0.37
OpenAD-DGCNN	8.02	15.83	66.76	13.52	0.39
LASO	34.49	55.44	76.12	37.88	8.40
3D-ADLLM	36.38	56.46	72.74	46.92	37.33
VoxAfford (ours)	**41.19**	**58.57**	**77.28**	**48.57**	**46.41**

Ablation on module design. Table 4 isolates each component. Removing the voxel-token fusion causes the largest drop and brings performance back to 3D-ADLLM, confirming that LLM tokens alone lack spatial information for localization. The feature injection, attention aggregation, and compatibility gate each contribute smaller but consistent gains by preserving per-point con-

Table 3: Ablation on input modality and LLM backbone.

Input	LLM	mIoU^i	mAcc^i	mAP$^i_{50}$
Voxel	Phi-4	29.19	66.44	23.34
PC + Voxel	Phi-4	32.00	72.49	29.74
PC	Qwen3	33.68	71.41	31.71
PC	Phi-4 (ours)	**40.57**	**77.68**	**40.67**

Table 4: Component ablation.

Variant	mIoU^i	mAcc^i	mAP$^i_{50}$
w/o voxel-token fusion	29.64	74.83	27.00
w/o feature injection	33.29	73.87	30.41
w/o token aggregation	35.97	74.54	33.82
w/o compatibility gate	35.32	71.71	32.05
VoxAfford (full)	**40.57**	**77.68**	**40.67**

Table 5: Ablation on token assignment in the dual-pathway propagation.

Prompt	Injection	mIoU^i	mAcc^i	mAP$^i_{50}$
$\mathbf{T}$	$\hat{\mathbf{T}}$	34.66	70.38	31.75
$\hat{\mathbf{T}}$	$\hat{\mathbf{T}}$	32.90	68.85	28.94
$\mathbf{T}$	$\mathbf{T}$	30.12	74.17	26.43
$\hat{\mathbf{T}}$	$\mathbf{T}$ (ours)	**40.57**	**77.68**	**40.67**

Table 6: Ablation on scale combinations and token count.

Scales	#	mIoU^i	mAcc^i	mAP$^i_{50}$
16^3 only	1	31.80	69.20	29.10
32^3 only	1	30.71	73.00	28.49
64^3 only	1	34.82	71.30	30.60
3×32^3 (same)	3	33.00	68.85	30.70
16^3+32^3+64^3	3	**40.57**	**77.68**	**40.67**

ditioning, scale-aware weighting, and selective injection of relevant geometry. Table 5 compares four routings for the dual-pathway propagation. Our default sends voxel-enhanced tokens $\hat{\mathbf{T}}$ to the prompt pathway and original tokens $\mathbf{T}$ to the injection pathway, and all alternatives degrade performance. Voxel-enhanced tokens carry multi-scale geometry suited for global prompts, while original tokens preserve clean semantics suited for per-point conditioning. The default routing matches each token type to its representational strength.

Multi-scale diversity in voxel-token fusion. Table 6 contrasts multi-scale diversity with token count. Three 32^3 tokens improve mIoU over one 32^3 token (33.00 versus 30.71), but remain below one 64^3 token in mIoU while achieving slightly higher mAP$^i_{50}$. The full multi-scale model achieves the best results on both metrics, indicating that the gains from additional same-scale tokens do not replace the benefits of complementary multi-scale geometry. Appendix B.4 visualizes token-to-point attention before and after voxel injection.

4.4 Qualitative Results

Figure 3 shows qualitative comparisons on open-set zero-shot affordance detection. VoxAfford produces more spatially precise masks than the baselines and aligns closely with the ground truth. OpenAD and LASO frequently under-predict and leave fragmented or incomplete regions. VoxAfford delineates exact interaction boundaries across diverse geometric structures.

Table 7: Real robot success rates on five novel objects.

Method	Cup w/ handle	Cup w/o handle	Fruits	Ladle	Pan	Average
3D-ADLLM	5/10	0/10	6/10	5/10	2/10	18/50
VoxAfford (ours)	**8/10**	**5/10**	**6/10**	**5/10**	**9/10**	**33/50**

4.5 Real Robot Experiments

Experimental setup. We deploy VoxAfford on a Franka Research 3 arm with a parallel-jaw gripper to test zero-shot transfer. Three Orbbec Femto Bolt RGB-D cameras reconstruct a scene point cloud, from which the target object is segmented by workspace cropping and table plane removal. Camera-to-robot extrinsics are obtained through hand-eye calibration. Camera viewpoints and extrinsics remain fixed, while object pose, robot initial state, and clutter are randomized per trial. Both methods use identical paired scenes and the same execution pipeline. Figure 4 shows five tabletop scenes with novel objects absent from training. Each task runs 10 times, and Table 7 reports per-object success rates.

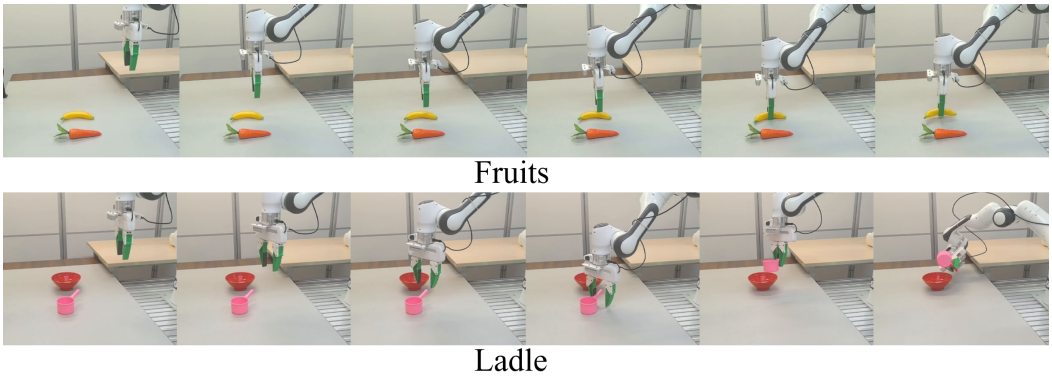


Figure 5: Visualization of real-world robot manipulation experiments. Top: The robot arm successfully grasps and picks up a banana from the tabletop while avoiding the nearby carrot. Bottom: The robot arm reaches for and grasps a ladle, transferring it into a red bowl. Each row shows a sequential progression of the manipulation task from left to right.

Execution pipeline. Given a language query, the model predicts the affordance region. We compute a grasp pose from the centroid and surface normal of the highest-confidence region, and use cuRobo to generate collision-free trajectories. After grasping, a predefined motion such as lifting or tilting completes the manipulation.

VoxAfford reaches 66% average success. The pan and the cup with handle yield the highest rates, since their affordance regions sit on geometrically salient structures that voxel features reliably capture. The cup without handle and the ladle are harder, lacking distinctive boundaries. These results confirm zero-shot transfer to novel objects without fine-tuning. Additional results are provided in Appendix C.

Figure 5 visualizes the approach, grasp, and subsequent manipulation in two representative trials. In the top row, the robot grasps and lifts a banana while avoiding a nearby carrot. In the bottom row, it reaches for a ladle, grasps it, and transfers it into a red bowl. These sequences connect the predicted interaction region to the executed robot motion.

5 Conclusion

We present VoxAfford, a method that injects hierarchical voxel features into LLM output tokens after autoregressive generation for open-vocabulary 3D affordance detection. Existing MLLM-based methods produce output tokens that encode affordance semantics but lack 3D spatial structure for guiding mask prediction. VoxAfford pairs each output token with a designated voxel resolution and fuses geometric information through cross-attention with a learned compatibility gate. The voxel-enhanced tokens form a spatially-aware affordance prompt and propagate alongside per-point features for mask generation. Experiments show state-of-the-art performance with an approximately 10% absolute improvement in instance-level mIoU, and the model transfers to real robot tasks.

Limitations. VoxAfford relies on geometric priors from a frozen pre-trained VQVAE, and fixed token–scale pairing limits flexibility in assigning retrieval scales. Noisy or partial geometry and errors in object segmentation, calibration, or point cloud reconstruction can affect localization and execution. Our real-robot evaluation uses a single Franka Research 3 platform, so transfer across robot platforms remains unevaluated. The method focuses on single isolated objects; extending it to multi-object scenes with inter-object spatial reasoning and affordance disambiguation remains an open direction.

Acknowledgments

This work was supported in part by the NSFC under grant No. 62125305, No. U23A20339, No. 62573339, No. 62503380, Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under grant No. JYB2025XDXM210, and Natural Science Foundation of Shaanxi Province under Grant 2025SYSSYSZD-083, State Grid Corporation of China Science and Technology Projects 52060025005B-439-ZN, and Shaanxi Provincial Key R&D Program under Grant No. 2026JC-ZDGG-17.

References

[1] K. Rana, J. Abou-Chakra, S. Garg, R. Lee, I. Reid, and N. Suenderhauf. Learning from 10 demos: Generalisable and sample-efficient policy learning with oriented affordance frames. *arXiv preprint arXiv:2410.12124*, 2024.

[2] S. Wu, Y. Zhu, Y. Huang, K. Zhu, J. Gu, J. Yu, Y. Shi, and J. Wang. Afforddp: Generalizable diffusion policy with transferable affordance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6971–6980, 2025.

[3] R. Xu, J. Zhang, M. Guo, Y. Wen, H. Yang, M. Lin, J. Huang, Z. Li, K. Zhang, L. Wang, et al. A0: An affordance-aware hierarchical model for general robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13491–13501, 2025.

[4] Z. Fan and K. Liang. Video-to-image affordance grounding via visual conceptual learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3972–3980, 2025.

[5] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3822–3831. IEEE, 2025.

[6] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li. Affordancellm: Grounding affordance from vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7587–7597, 2024.

[7] Z. Wei, J. Lin, Y. Liu, W. Chen, J. Luo, G. Li, and L. Lin. 3daffordsplat: Efficient affordance reasoning with 3d gaussians. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 2821–2830, 2025.

[8] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia. 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.

[9] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas. O2o-afford: Annotation-free large-scale object-object affordance learning. In *Conference on robot learning*, pages 1666–1677. PMLR, 2022.

[10] H. Zhu, Q. Kong, K. Xu, X. Xia, B. Deng, J. Ye, R. Xiong, and Y. Wang. Grounding 3d object affordance with language instructions, visual observations and interactions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17337–17346, 2025.

[11] H. Wang, S. Wang, Y. Zhong, Z. Yang, J. Wang, Z. Cui, J. Yuan, Y. Han, M. Liu, and Y. Ma. Affordance-r1: Reinforcement learning for generalizable affordance reasoning in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 9738–9746, 2026.

[12] L. He, M. Liu, Q. Ye, Y. Zhou, X. Deng, and G. Ding. Task-aware 3d affordance segmentation via 2d guidance and geometric refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4654–4662, 2026.

[13] T. Van Vo, M. N. Vu, B. Huang, T. Nguyen, N. Le, T. Vo, and A. Nguyen. Open-vocabulary affordance detection using knowledge distillation and text-point correlation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13968–13975. IEEE, 2024.

[14] L. Wu, W. Wei, P. Yu, and J. Lan. Open-vocabulary 3d affordance understanding via functional text enhancement and multilevel representation alignment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 7988–7997, 2025.

[15] C. Yu, H. Wang, Y. Shi, H. Luo, S. Yang, J. Yu, and J. Wang. Seqafford: Sequential 3d affordance reasoning via multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1691–1701, 2025.

[16] H. Chu, X. Deng, Q. Lv, X. Chen, Y. Li, J. Hao, and L. Nie. 3d-affordancellm: Harnessing large language models for open-vocabulary affordance detection in 3d worlds. *arXiv preprint arXiv:2502.20041*, 2025.

[17] Z. Zhang, Y. Shi, L. Yang, S. Ni, Q. Ye, and J. Wang. Openhoi: Open-world hand-object interaction synthesis with multimodal large language model. *arXiv preprint arXiv:2505.18947*, 2025.

[18] R. Chen, M. Yin, J. Shen, and W. Ma. Recon3d: High quality 3d reconstruction from a single image using generated back-view explicit priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2802–2811, 2024.

[19] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[20] T. Mar, V. Tikhanoff, G. Metta, and L. Natale. Multi-model approach based on 3d functional features for tool affordance learning in robotics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 482–489. IEEE, 2015.

[21] S. Zhang, Z. Qi, R. Dong, X. Bai, and X. Wei. Positional prompt tuning for efficient 3d representation learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1171–1180, 2025.

[22] K.-C. Huang, X. Li, L. Qi, S. Yan, and M.-H. Yang. Reason3d: Searching and reasoning 3d segmentation via large language model. In *2025 International Conference on 3D Vision (3DV)*, pages 1177–1186. IEEE, 2025.

[23] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9579–9589, 2024.

[24] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

[25] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835, 2021.

[26] O. Gafni, A. Polyak, O. Ashual, S. Sheynin, D. Parikh, and Y. Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European conference on computer vision*, pages 89–106. Springer, 2022.

[27] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[28] Y. Chen, Y. Lan, S. Zhou, T. Wang, and X. Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28371–28382, 2025.

[29] Z. Qi, R. Dong, S. Zhang, H. Geng, C. Han, Z. Ge, L. Yi, and K. Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer, 2024.

[30] J. Ye, Z. Wang, R. Zhao, S. Xie, and J. Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025.

[31] J. J. Gibson. The senses considered as perceptual systems. 1966.

[32] Y. Tang, S. Zhang, X. Hao, P. Wang, J. Wu, Z. Wang, and S. Zhang. Affordgrasp: In-context affordance reasoning for open-vocabulary task-oriented grasping in clutter. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9433–9439. IEEE, 2025.

[33] X. Zhu, Y. Li, L. Cui, P. Li, H.-a. Gao, Y. Zhu, and H. Zhao. Afford-x: Generalizable and slim affordance reasoning for task-oriented manipulation. *arXiv preprint arXiv:2503.03556*, 2025.

[34] T.-T. Do, A. Nguyen, and 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.

[35] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[36] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021.

[37] Y. Yang, W. Zhai, H. Luo, Y. Cao, J. Luo, and Z.-J. Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10905–10915, 2023.

[38] Y. Li, N. Zhao, J. Xiao, C. Feng, X. Wang, and T.-s. Chua. 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.

[39] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. 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.

[40] Y. Shao, W. Zhai, Y. Yang, H. Luo, Y. Cao, and Z.-J. Zha. Great: Geometry-intention collaborative inference for open-vocabulary 3d object affordance grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17326–17336, 2025.

[41] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36: 20482–20494, 2023.

[42] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.

[43] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen. 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.

[44] Y. Chen, S. Yang, H. Huang, T. Wang, R. Xu, R. Lyu, D. Lin, and J. Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024.

[45] J. Li, D. Li, S. Savarese, and S. Hoi. 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.

[46] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[47] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[48] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2 (1):1, 2023.

[49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[50] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. 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.

[51] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[52] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. 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.

[53] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[54] T. Nguyen, M. N. Vu, A. Vuong, D. Nguyen, T. Vo, N. Le, and A. Nguyen. 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.

[55] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[56] L. Xue, N. Yu, S. Zhang, A. Panagopoulou, J. Li, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, 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.

[57] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[58] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[59] J. Jian, X. Liu, M. Li, R. Hu, and J. Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023.

[60] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[61] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

A Implementation Details

A.1 Datasets and Evaluation Protocols

We follow the evaluation protocols defined in the 3D AffordanceNet (OpenAD) benchmark [54, 8] for both full-view and partial-view settings.

Full-view. The model receives a complete 3D point cloud of the object and a natural language affordance query. The task is to predict a per-point binary mask indicating the queried affordance region. Each point cloud contains 2,048 points uniformly sampled from the object surface.

Partial-view. In real-world scenarios, only partial observations of objects are available. Following [8], partial point clouds are generated by rendering depth images from sampled viewpoints and back-projecting to 3D. The model receives the partial point cloud and predicts the affordance mask on the visible points. This setting evaluates the model's robustness to incomplete geometric observations.

Training data construction. For segmentation pre-training (Stage 1), we sample a balanced subset from the PartNet dataset [52] across its 24 object categories. Each training sample consists of an object point cloud, a text description of a target part, and the corresponding binary segmentation mask. This stage establishes general segmentation capability before affordance-specific training. For affordance training (Stage 2), we construct question-answer pairs from the 3D AffordanceNet dataset [8]. Each pair contains an object point cloud, an affordance query in natural language, and the ground-truth affordance mask.

A.2 Baseline Implementation

OpenAD variants. OpenAD [54] provides two backbone variants, OpenAD-PointNet++ and OpenAD-DGCNN. Both variants use CLIP text embeddings to align point features with affordance descriptions. We use the official implementations with default hyperparameters.

IAGNet. IAGNet [37] takes paired image-point cloud data as input and models inter-category affordance relations through graph reasoning. To adapt IAGNet for our text-conditioned setting, we replace its image backbone with a language encoder while keeping the remaining architecture unchanged.

LASO. LASO [38] aligns language and point cloud representations for affordance detection. We use the official implementation and evaluate under the same open-set zero-shot protocol.

3D-ADLLM. 3D-ADLLM [16] extends MLLM-based affordance detection with sequential output tokens for mask prediction. We report results from the original paper under the same evaluation splits.

A.3 Model Architecture

We use Phi-4-mini-instruct [55] with 3.8B parameters as the base LLM. The point encoder is Point-BERT [50] pre-trained with ULIP-2 [56] on ModelNet [57]. The point backbone f_{PB} is Point Transformer [58]. The VQVAE encoder is loaded from the ShapeLLM-Omni [30] pre-trained checkpoint with a codebook size of 8,192 and remains frozen throughout all training stages. The affordance decoder f_{AFD} follows the two-way transformer architecture. Each input point cloud is uniformly sampled to 2,048 points.

The three affordance tokens are appended to the LLM vocabulary and paired with voxel resolutions 16^3, 32^3, and 64^3 respectively. The VQVAE encoder features at these resolutions have channel dimensions of 512, 128, and 32. Before cross-attention, all features are spatially pooled to a uniform resolution of 2^3 via adaptive average pooling, yielding 8 spatial tokens per scale. The cross-attention in the voxel-token fusion module uses 8 heads and is shared across the three branches to reduce parameter count. A dropout rate of 0.15 is applied on the cross-attention output.

The token aggregation module produces the affordance prompt through cross-attention with 8 heads. The feature injection pathway uses the original LLM tokens rather than the voxel-enhanced tokens to condition per-point features, following the dual-pathway design described in Section 3.2.

A.4 Training Hyperparameters

Stage 1. We pre-train f_{PB} and f_{AFD} on the PartNet subset for segmentation. The point encoder, VQVAE encoder, and LLM backbone are frozen during this stage. The pre-trained weights of f_{PB} and f_{AFD} initialize the corresponding modules in the next stage.

Stage 1 inherits the full ROPS protocol of 3D-ADLLM [16], from data construction to validation, with a 7:3 train and validation split and balanced sampling across all 24 PartNet categories. We use AdamW with a learning rate of 1×10^{-4} for 20 epochs. Only f_{PB} and f_{AFD} are trained, using BCE and Dice losses.

Stage 2. The full model is trained on affordance data with the point encoder, VQVAE encoder, and LLM backbone frozen. The LLM is adapted with LoRA applied to the `qkv_proj` layers, with rank 32, alpha 48, and dropout 0.1. The trainable components include LoRA parameters, the voxel-token fusion module, the spatial propagation module, f_{PB}, and f_{AFD}. We use the AdamW optimizer with an initial learning rate of 3×10^{-4}, cosine decay to 1×10^{-6}, and warmup from 1×10^{-8} over the first 120 steps. No weight decay is applied. The batch size is 8 per GPU with gradient accumulation over 4 steps, yielding an effective batch size of 64. Training runs for 20 epochs. The loss weights are $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = 1.0$. All training uses BF16 mixed precision on NVIDIA A800 GPUs with 80GB memory. Stage 2 training on 2×A800 GPUs takes approximately 12 hours.

A.5 Compatibility Gate Initialization

The compatibility gate maps the concatenated projections of the original affordance token and the retrieved geometric feature to a scalar through sigmoid activation, as defined in Section 3.1. For each scale j, the gate weight vector $\mathbf{w}_j$ is zero-initialized and the bias b_j is initialized to -2.0, yielding an initial scalar gate value of $\sigma(-2.0) \approx 0.12$. This initialization follows the zero-initialization principle from ControlNet [61]. Training therefore begins with weak geometric injection, and the gate learns to control the injection strength according to semantic–geometric compatibility.

B Additional Experimental Results

B.1 Closed-Set Comparison Results

The main text focuses on open-set zero-shot evaluation, where the test affordance categories are disjoint from training. Here we report closed-set results, where the test categories overlap with training, to verify that VoxAfford does not sacrifice closed-set performance for open-vocabulary generalization. As shown in Table 8 and Table 9, VoxAfford achieves the best mIoU and mAP_{50} under both full-view and partial-view settings while maintaining competitive accuracy. The closed-set improvements are smaller than the open-set gains reported in the main text. This is expected because closed-set methods can memorize affordance-geometry associations from training, which reduces the advantage of explicit geometric injection. The consistent improvement on mIoU and mAP_{50} confirms that the voxel-token fusion module provides complementary spatial information even when the affordance categories are seen during training.

B.2 Scale-Specific Evidence and Token–Scale Routing

Scale-specific evidence. The branches retrieve complementary geometric information at different granularities rather than fixed per-token semantic roles. With token count held at three, the heterogeneous-scale configuration outperforms same-scale repetition in both mIoU and mAP_{50}^{i} in Table 6, supporting the benefit of combining granularities beyond increasing token count alone.

Table 8: Closed-set affordance detection results over all classes. The top three rows (Point Transformer, PointNet++, DGCNN) are fully-supervised closed-set segmentation baselines that train directly on the test categories without language conditioning; they serve as upper-bound references and are excluded from the best-result comparison. Acc^c is not reported for these baselines because they do not produce per-category classification confidence.

Method	Full-view			Partial-view		
	mIoU^c	Acc^c	mAcc^c	mIoU^c	Acc^c	mAcc^c
Point Transformer	41.26	–	67.03	40.51	–	65.34
PointNet++	41.26	–	68.14	41.10	–	66.74
DGCNN	42.09	–	61.47	41.93	–	63.12
OpenAD-PointNet++	40.17	38.61	66.83	40.44	38.92	65.84
OpenAD-DGCNN	41.17	35.71	59.17	39.87	35.15	59.27
IAGNet	40.04	35.12	53.05	41.24	34.68	52.58
LASO	41.31	35.02	53.96	40.11	35.21	52.68
3D-ADLLM	42.85	41.84	66.35	41.92	43.40	61.93
VoxAfford (ours)	**43.32**	**59.69**	66.38	**42.01**	**54.03**	64.07

Table 9: Closed-set affordance detection results over all instances.

Method	Full-view				
	mIoU^i	mAcc^i	mPrec^i	mRec^i	mAP^i_{50}
OpenAD-PointNet++	28.34	64.11	33.91	61.45	5.12
OpenAD-DGCNN	26.98	65.94	34.38	54.76	4.77
LASO	44.43	**83.80**	**62.73**	60.25	21.13
3D-ADLLM	46.29	81.24	57.90	64.27	46.38
VoxAfford (ours)	**46.31**	81.83	56.23	**70.49**	**48.57**
Method	Partial-view				
	mIoU^i	mAcc^i	mPrec^i	mRec^i	mAP^i_{50}
OpenAD-PointNet++	29.50	63.26	35.21	61.34	6.77
OpenAD-DGCNN	17.07	67.11	27.96	30.15	1.87
LASO	43.35	**82.31**	**60.27**	59.57	20.85
3D-ADLLM	44.06	79.64	56.23	64.21	46.60
VoxAfford (ours)	**45.04**	78.61	54.41	**69.35**	**47.58**

Number of voxel scales. The frozen VQVAE natively provides three stages spanning global topology to local surface detail, and using them directly captures geometric information at complementary granularities. All single scales and pairs underperform the full combination in Tables 6 and 10, and a fourth scale would require retraining the encoder.

Token–scale routing. Routing is fixed only at retrieval, where one-to-one pairing covers every VQVAE scale without assignment supervision. Query-dependent scale use already exists at aggregation, since Equation 3 attends over all enhanced tokens with query-conditioned weights. Affordances requiring global and local cues thus draw on all scales at aggregation.

B.3 Pooling Resolution and Stage Combinations

Table 10 varies the pooling resolution with source stages fixed. The 2^3 setting achieves the best results, and retaining more voxel tokens at 4^3 instead degrades performance. Every branch keeps eight tokens after pooling in the default 2^3 setting. The 64^3 branch preserves fine-grained feature content from its VQVAE stage rather than extra spatial bins. Each stage pair underperforms the full combination, and heterogeneous stages beat 3×32^3 on mIoU under identical pooling in Table 6, confirming non-redundant contributions.

Table 10: Joint ablation on OpenAD under full-view setting.

	Pooling/stage			Stage pair ($\cdot^3$)		
Setting	1^3	2^3	4^3	16+32	16+64	32+64
mIoU^i	33.82	**40.57**	35.64	34.50	37.86	35.61
mAP$^i_{50}$	29.95	**40.67**	30.87	31.27	36.26	29.90

B.4 Attention Visualization of Voxel-Token Fusion

To verify that voxel injection redirects the affordance tokens toward the queried 3D region, we visualize the cross-attention each token induces over the dense point features before and after fusion. Figure 6 compares the attention map produced by the original LLM token $\mathbf{t}$ and the voxel-enhanced token $\hat{\mathbf{t}}$ on seven test objects covering different affordance categories. Before injection, attention from $\mathbf{t}$ spreads broadly across the object surface with little preference for the queried interaction region. This pattern is consistent with the structural argument that autoregressive output tokens carry weak spatial neighborhood information. After injection, the attention mass shifts toward the ground-truth affordance region. The consistent direction of the shift across categories indicates that the injected voxel features supply the spatial structure that the original tokens lack.



Figure 6: **Cross-attention from affordance tokens to point features, before and after voxel injection.** Each column shows one (object, affordance) pair. The first row reports the input point cloud. The second row shows the attention induced by the LLM output token $\mathbf{t}$ before voxel injection, which spreads diffusely with little preference for the queried region. The third row shows the attention from the voxel-enhanced token $\hat{\mathbf{t}}$ after fusion, where the attention mass redistributes toward the GT-annotated affordance region such as the bone end, the scissors handle, and the fabric opening. A residual diffuse component remains due to the gated residual injection. The fourth row shows the ground-truth affordance annotation. Brighter red indicates higher attention weight in the second and third rows, and GT membership in the fourth row.

C Real Robot Deployment Details

C.1 Oracle Masks

Oracle masks reach 41/50, as better masks yield more feasible grasp poses, leaving 5 calibration/reconstruction failures and 4 planning failures, providing an empirical reference for gains from improved affordance prediction.

C.2 Failure Analysis

Table 7 in the main text reports an average success rate of 66% (33/50) across five novel objects. We analyze the 17 failure cases and categorize them into three types.

Motion planning failure (9/17, 53%). This is the most frequent failure mode. cuRobo occasionally fails to find a collision-free trajectory to the target grasp pose, especially when the target pose is near workspace boundaries or close to the table surface. In these cases the affordance prediction is correct, but the robot cannot physically execute the planned grasp.

Calibration and reconstruction error (5/17, 29%). Residual errors in hand-eye calibration and depth sensor noise introduce spatial offsets between the reconstructed point cloud and the true object position. These offsets cause the robot to approach a shifted location, leading to imprecise grasps that slip or miss the target. This error is more pronounced for small objects such as fruits, where millimeter-level offsets are significant relative to the object size.

Affordance prediction failure (3/17, 18%). In three cases VoxAfford predicts an incorrect or incomplete affordance region. For example, on the handleless cup the model occasionally predicts the rim rather than the body as the graspable region, which does not yield a stable grasp. These failures occur on objects that lack geometrically distinctive affordance boundaries.