mmLIP: mmWave Radar-Language Interactive Pretraining via Point Confidence

Jeongwan Shin[1,2], Jaehyeon Kim[1], Jae-Ho Choi[1,2]*

[1] DGIST, [2] KAIST InnoCORE LLM

Abstract

Millimeter-wave (mmWave) radar provides core sensing capabilities to traditional vision, particularly under occlusion, low-light, and privacy-constrained conditions. Recent efforts have explored integrating radar signals with large language models (LLMs) for high-level semantic reasoning over the received radar data. However, existing approaches typically project radar inputs into borrowed vision or LiDAR embedding spaces, which are not tailored to radar-specific physical cues such as Doppler and reflection intensity. This design introduces a representational bottleneck, hindering the extraction of informative radar semantics. To overcome this limitation, we propose **mmLIP**, a novel radar-language interactive pretraining framework that learns radar-specific alignment without relying on external embedding spaces. Our approach directly aligns radar representations with the text embedding space via a point-level contrastive objective, enabling fine-grained correspondence between radar points and textual tokens. In addition, we introduce a confidence-aware contrastive learning mechanism that adaptively reweights radar tokens based on their semantic relevance, promoting informative signals while suppressing clutter. Supported by our newly curated radar-text pairs, mmLIP captures structured radar semantics. Extensive experiments demonstrate that mmLIP successfully integrates with diverse LLMs or vision-language models, achieving state-of-the-art performance on zero-shot bidirectional retrieval as well as text generation tasks, including captioning and question answering.[1]

1 Introduction

Millimeter-wave (mmWave) radar has emerged as a robust alternative to RGB vision systems, effectively mitigating challenges related to occlusion, poor illumination, and privacy constraints [1–3]. Motivated by these physical advantages, a growing body of research aims to elevate the utility of radar by integrating radar signals with large language models (LLMs) [4–8]. By transforming mmWave radar into text-aligned semantic representations, these approaches enable high-level understanding and reasoning, as well as human-centric interaction through natural language [9–12].

Prior work in this line of research mainly follows two strategies. One approach aligns radar representations with text via a vision-centric joint embedding space [13] defined by a pre-trained vision-text encoder (*e.g.*, CLIP [14]), where radar features are first projected into the vision space and alignment is learned through joint training with an LLM for text generation [12] (Figure 1(a)). Another approach first processes radar inputs using lightweight layers or encoders pretrained on LiDAR data, and then linearly projects them via a shallow projector (*e.g*, an MLP) into the LLM input space for text generation [11], without explicitly learning radar-aware representations (Figure 1(b)).

*Corresponding author.

[1] Project site: `https://mmlip-project.github.io/`

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

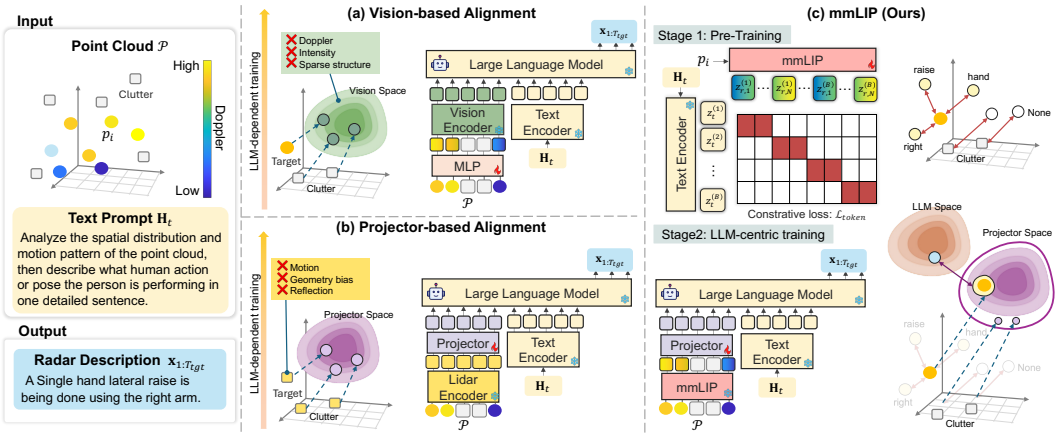

Figure 1: Comparison of radar–language alignment paradigms: (a) Vision-based alignment projects radar signals into a vision embedding space via a shallow MLP layer. (b) Projector-based alignment implicitly aligns radar inputs with text representations using an additional LiDAR-pretrained projector. (c) Our method directly learns a radar-aware encoder with token-level contrastive alignment.

While both paradigms successfully bridge the gap between radar inputs and text generation, they suffer from limitations in capturing radar-specific representations. As shown in Figure 1, they rely on borrowed intermediate embedding spaces that are completely agnostic to the unique physical characteristics of mmWave radar. mmWave radar exhibits modality-specific characteristics, such as Doppler and electromagnetic reflection intensity, and is inherently affected by background clutter. Since vision and LiDAR models are not designed to capture these electromagnetic properties, they often fail to extract meaningful radar-specific features, making a radar-specific point encoder essential. However, because a radar-specific encoder explicitly designed for radar-language alignment does not yet exist, prior works have been constrained to relying on suboptimal integration with LLM [11, 12].

To overcome this bottleneck, we propose **mmLIP**, a radar-language interactive pretraining method to explicitly learn a radar-specific alignment at the fine-grained token-level. Rather than relying on encoders trained on other modalities, our pre-training strategy utilizes a novel point-level contrastive learning objective to directly align radar representations with the text embedding space. This enables the model to directly learn radar-specific knowledge and explicit radar-text correspondence from scratch. To further handle the sparse and noisy nature of radar points, we introduce a *Point Confidence Loss* (PCL) that adaptively reweights radar tokens based on their semantic relevance to textual semantics. This objective enables the model to emphasize informative reflections while suppressing clutter (target-irrelevant signals), leading to more robust and precise cross-modal alignment. Pretrained with our own curated radar-text pairs, the resulting representations capture rich and structured information, supporting a range of zero-shot bidirectional tasks, including text–radar and radar–text retrieval, question answering (QA), and fine-grained motion understanding (*e.g.*, limb-level QA). Finally, integrating the learned representations with language models consistently improves downstream performance on diverse text-based tasks under both closed- and open-ended settings.

In summary, our contributions are threefold:

- We propose mmLIP, a mmWave radar–language interactive pretraining framework that enables direct token-level alignment between individual radar points and textual tokens, without relying on intermediate modalities or projector-based alignment.
- We introduce a point confidence-aware learning objective that incorporates radar-specific physical cues (*e.g.*, Doppler and intensity) to adaptively reweight tokens, emphasizing motion-related reflections while effectively suppressing background clutter.
- We demonstrate that our approach, when integrated with diverse publicly available LLMs, consistently achieves superior text generation performance across multiple tasks (*e.g.*, QA, description, and limb-level QA) under both closed- and open-ended settings.

2 Related Works

Multi-modal Encoder. Multimodal representation learning [14–20] has been actively studied, largely driven by large-scale pretraining on paired image and text data [21–23], which has enabled the emergence of powerful joint embedding spaces. Building upon this paradigm, recent works have explored hub-modal alignment [13], where a dominant modality, typically vision or language, acts as an anchor to align additional modalities under limited paired supervision. This line of research has naturally extended to heterogeneous sensory inputs, including depth [24], thermal [25], IMU [26], and audio [27–29], broadening the scope of cross-modal generalization. Methodologically, prior work has explored several training paradigms for cross-modal representation learning. A predominant approach leverages contrastive objectives, such as InfoNCE [30] or CLIP-style losses [14], to enforce cross-modal alignment in a shared latent space [31, 32]. Following these advances, recent works have explored extending multimodal encoders to mmWave radar inputs [17], particularly using radar heatmap representations as input to learn radar-aware embeddings. However, despite these advances, encoder architectures tailored for radar point cloud data remain largely underexplored. This is partly because radar point clouds are inherently sparse, noisy, and modality-specific, which makes them challenging for cross-modal alignment and limits the applicability of vision-centric frameworks.

Radar-LLM Integration. Recent works have explored integrating mmWave radar with language models [9, 11, 12, 17] to enable high-level semantic reasoning and human-centric natural language interaction over non-visual sensing data. These methods primarily perform radar–text alignment through LLM-centric learning. A common strategy is to leverage pretrained multimodal encoders [12], often built upon vision-centric frameworks [14] to project radar signals into an existing embedding space, often by treating radar as an image-like modality or aligning it with visual representations. Another line of work [11] integrates radar inputs with LLMs by introducing modality-specific tokenizers or embedding layers, which are jointly optimized during language modeling without explicitly learning structured radar representations. While these approaches demonstrate promising results, they rely on implicit alignment through language modeling objectives or intermediate modalities, rather than explicitly learning a shared embedding space between radar and text.

In contrast, recent advances in multimodal learning demonstrate that directly pretraining modality-specific encoders (*e.g.*, for vision or audio) and integrating them with language models leads to improved representation quality and cross-modal reasoning [33–35]. These approaches emphasize learning structured, modality-aware embeddings prior to alignment with language, leading to more robust and semantically meaningful interactions. However, encoder-centric approaches that learn token-level radar representations remain underexplored for radar, where the lack of representation learning and systematic analysis limits the ability to capture fine-grained radar semantics.

3 Preliminaries

We formalize radar-language alignment within the LLM learning framework and revisit two representative paradigms in prior work: *vision-based alignment* and *projector-based alignment*. These formulations serve as the basis for analyzing the limitations and motivate our proposed method.

Radar-language Alignment Formulation. Given a radar point cloud $\mathcal{P} = \{p_i\}_{i=1}^{N}$ with N radar points, where each point $p_i = (x_i, y_i, z_i, d_i, s_i)$ includes 3D coordinates, Doppler, and intensity, and a sequence of L text tokens $\{w_k\}_{k=1}^{L}$, radar points and text tokens are encoded into an embedding space as $\mathbf{z}_{r,i} = f_r(p_i) \in \mathbb{R}^D$ and $\mathbf{z}_{t,k} = f_t(w_k) \in \mathbb{R}^D$, where $f_r(\cdot)$ and $f_t(\cdot)$ denote the radar and text encoders, respectively. A similarity score between a radar point and a text token is defined as

$$s_{i,k} = \frac{\mathbf{z}_{r,i}^{\top} \mathbf{z}_{t,k}}{\|\mathbf{z}_{r,i}\| \, \|\mathbf{z}_{t,k}\|}.$$

The alignment objective aims to increase similarity by leveraging the observation that action-related tokens are more likely to correspond to radar points exhibiting high Doppler values and distinctive spatial locations, while ensuring that clutter points do not exhibit high similarity with any tokens.

Vision-based Alignment. To align radar and text modalities, $f_r(\cdot)$ is first employed as a radar point tokenizer that maps each radar point to $\mathbf{z}_{r,i}$. These tokens are then processed by a vision encoder

$f_v(\cdot)$ to obtain a vision-centric representation in a pretrained vision embedding space:

$$\mathbf{H}_r = f_v(\mathbf{Z}_r) \in \mathbb{R}^{N\times D}, \quad \mathbf{Z}_r = \{\mathbf{z}_{r,i}\}_{i=1}^{N}.$$

The resulting features are concatenated with text embeddings $\mathbf{H}_t$, obtained from the input prompt (e.g., instruction or query), and provided as input to a LLM:

$$\mathbf{H} = [\mathbf{H}_r; \mathbf{H}_t].$$

The model is trained end-to-end using the standard autoregressive language modeling objective

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log p(x_t \mid x_{<t}, \mathbf{H}; \theta),$$

where θ denotes the trainable parameters. Through this process, the LLM implicitly supervises the radar tokenizer, enabling it to produce representations aligned with textual semantics. However, this process may lead to the loss of radar-specific characteristics, as the vision-centric embedding space lacks the capacity to represent radar-specific cues such as Doppler and signal intensity.

Projector-based Alignment. Another line of work directly integrates radar representations with an LLM. In this approach, $\mathbf{Z}_r$ are obtained from pretrained encoders (*e.g.*, LiDAR-based encoders) or shallow MLP-based tokenizers, without explicitly learning radar-specific representations. These tokens are then mapped into the LLM input space via a learnable projector $g(\cdot)$ for text generation:

$$\mathbf{H}_r = g(\mathbf{Z}_r) \in \mathbb{R}^{N\times D_{\text{llm}}},$$

where D_{llm} denotes the hidden dimension of the LLM. $\mathbf{H}_r$ are concatenated with $\mathbf{H}_t$ and fed into the LLM, following the same formulation and autoregressive training objective as in vision-based alignment. However, this approach does not explicitly model the importance of individual radar points, making it difficult to distinguish between informative points and clutter. As a result, irrelevant or noisy points may be propagated to the LLM without proper filtering, potentially influencing the text generation process. To address this issue, we propose to learn point-level confidence through pre-training, enabling the model to selectively emphasize semantically relevant radar tokens.

4 Method

We propose a radar-language interactive pretraining approach that enables fine-grained, token-level alignment between radar points and text tokens, which we call **mmLIP**. This section presents three key components: a radar encoder, radar-language interactive pretraining, and integration with LLMs.

4.1 Radar Encoder for Modeling mmWave Physical Properties

We design a Transformer-based radar encoder $f_r(\cdot)$ that directly processes mmWave radar points, in contrast to prior approaches relying on MLP projectors or point encoders pretrained on LiDAR data. Radar encoder explicitly takes Doppler (d_i) and intensity (s_i) as input features. Each radar point is first embedded into a D-dimensional feature space and processed by a stack of Transformer layers:

$$\mathbf{h}_i^{(0)} = \phi(p_i) = Wp_i + b, \quad \mathbf{h}_i^{(\ell+1)} = \sum_{j=1}^{N} \alpha_{ij}^{(\ell)}\, \mathbf{W}_v^{(\ell)} \mathbf{h}_j^{(\ell)},$$

where $\phi(\cdot)$ denotes a learnable linear embedding layer. Through self-attention, the encoder captures relational dependencies among radar points, enabling it to model target motion patterns (via d_i) and reflection characteristics (via s_i) that are unique to mmWave radar. The final representation is given by $\mathbf{z}_{r,i} = f_r(p_i) = \mathbf{h}_i^{(L)}$, which preserves fine-grained spatial and physical characteristics.

4.2 Radar-language Interactive Pretraining

Token-level alignment. To achieve fine-grained cross-modal alignment, we employ a token-level contrastive objective in which each radar point is treated as an independent query and matched against textual token representations, in contrast to prior approaches relying on global feature aggregation

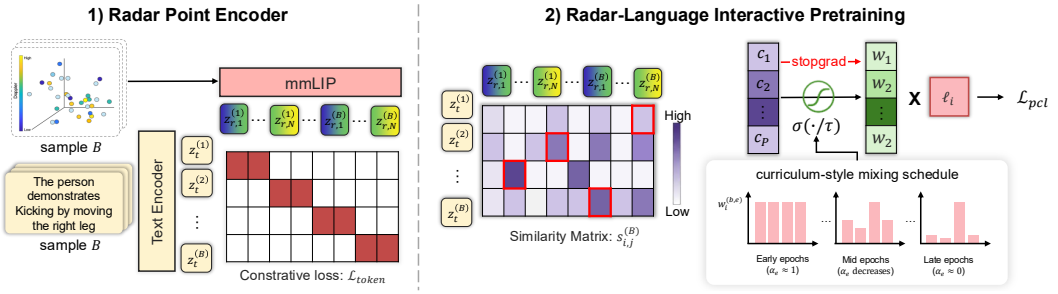

Figure 2: Overview of PCL: (1) Compute token-level contrastive losses between radar point tokens and text embeddings. (2) Derive confidence-aware weights from the similarity matrix and apply a curriculum-style mixing schedule to aggregate the weighted losses into the final objective.

(*e.g.*, CLS tokens). Building upon the similarity defined in Section 3, we extend it to a token-level formulation across the batch, where $s_{i,j,k}^{(b)}$ denotes the similarity between the i-th radar token of sample b and the k-th text token of sample j, with $j \in \{1, \ldots, B\}$ indexing text samples in the batch. We then apply a token-level InfoNCE objective, where each radar token is encouraged to align with text tokens from its paired sample ($j = b$) while pushing tokens from other samples away ($j \neq b$):

$$\mathcal{L}_{\text{token}} = -\sum_{b,i,k} m_i^{(b)} \log \frac{\exp\left(s_{i,j=b,k}^{(b)}/\tau\right)}{\sum_{j=1}^{B}\sum_{k'}\exp\left(s_{i,j,k'}^{(b)}/\tau\right)}.$$

In addition, we jointly learn a global alignment signal by leveraging the text CLS embedding, which provides a complementary semantic anchor to guide the overall radar representation.

Point Confidence Loss. To suppress clutter tokens and emphasize target tokens during training, we further introduce a confidence-aware reweighting strategy. For each radar point, we estimate a relevance score based on its similarity to the global text representations in the batch:

$$s_{i,j}^{(b)} = \frac{\mathbf{z}_{r,i}^{(b)\top}\mathbf{z}_t^{(j)}}{\|\mathbf{z}_{r,i}^{(b)}\|\,\|\mathbf{z}_t^{(j)}\|}, \qquad c_i^{(b)} = \max_j s_{i,j}^{(b)}, \qquad \text{conf}_i^{(b)} = \texttt{stopgrad}\left(\sigma\left(c_i^{(b)}/\tau_w\right)\right),$$

where $\mathbf{z}_t^{(j)}$ denotes the global text representation (CLS token) of sample j, $c_i^{(b)}$ denotes the raw semantic relevance score of the i-th radar token in sample b, $\sigma(\cdot)$ is the sigmoid function, and τ_w controls the sharpness of confidence weighting.

Tokens that fail to match any text receive a smaller confidence score, while informative tokens are assigned larger weights. The stop-gradient operation prevents the model from trivially reducing $\text{conf}_i^{(b)}$ to escape the loss. Because the confidence estimation can be unreliable in the early training stage, we apply a curriculum-style mixing schedule:

$$w_i^{(b)} = \alpha_e + (1-\alpha_e)\cdot \text{conf}_i^{(b)}, \qquad \alpha_e = \max\left(0,\ 1 - \frac{e}{E_{\text{warm}}}\right),$$

where e denotes the current training epoch, E_{warm} is the warm-up duration, α_e controls the transition from uniform weighting to confidence-based weighting, and $w_i^{(b)}$ is the final confidence weight assigned to each radar token. At early epochs, $\alpha_e \approx 1$, so all valid tokens are weighted nearly uniformly. As training progresses, $\alpha_e \to 0$, and the model increasingly focuses on confident tokens, where the target (*i.e.*, human) reflections are stronger. The PCL is defined as a confidence-weighted variant of the token-level contrastive loss:

$$\ell_i^{(b)} = -\log \frac{\exp(s_{i,b}^{(b)}/\tau)}{\sum_{j=1}^{B}\exp(s_{i,j}^{(b)}/\tau)}, \qquad \mathcal{L}_{\text{pcl}} = \frac{\sum_{b,i} w_i^{(b,e)}\,\ell_i^{(b)}}{\sum_{b,i} w_i^{(b,e)}},$$

where $\ell_i^{(b)}$ denotes the per-token InfoNCE (contrastive) loss that measures how well the i-th radar token in sample b aligns with its corresponding text representation against other samples in the batch. In practice, the model is trained with both $\mathcal{L}_{\text{token}}$ and $\mathcal{L}_{\text{pcl}}$, which respectively capture fine-grained alignment and clutter-aware robustness. Detailed parameter settings are provided in the Appendix D.

4.3 Integration with Large Language Models

Once the radar encoder is aligned with the text space, we integrate the radar tokens into a frozen LLM for downstream generation tasks. Given pre-aligned radar tokens $\mathbf{Z}_r \in \mathbb{R}^{N \times D_e}$ from our encoder, a lightweight learnable projector $g(\cdot)$ (a two-layer MLP with GELU) maps them into the LLM space:

$$\mathbf{H}_r = g(\mathbf{Z}_r) \in \mathbb{R}^{M \times D_{\text{llm}}},$$

where $M = N$ for the per-point variant (each radar point becomes an LLM token) and $M = 1$ for the mean-pool variant (a single summary token). The radar tokens are then prepended to the tokenized task prompt $\mathbf{H}_{\text{prompt}}$ and the target response $\mathbf{H}_{\text{tgt}}$ to form the input sequence

$$\mathbf{H} = [\mathbf{H}_r;\ \mathbf{H}_{\text{prompt}};\ \mathbf{H}_{\text{tgt}}].$$

The combined sequence is fed to the frozen LLM, and only $g(\cdot)$ is trained with a standard autoregressive objective applied *exclusively* to the target response:

$$\mathcal{L}_{\text{LM}} \ = \ -\sum_{t=1}^{T_{\text{tgt}}} \log p\big(x_t^{\text{tgt}} \,\big|\, x_{<t}^{\text{tgt}},\ \mathbf{H}_r,\ \mathbf{H}_{\text{prompt}}; \theta_g\big)\,,$$

where θ_g denotes only the projector parameters; the radar encoder and the LLM remain frozen. Loss is masked out on radar and prompt positions, so $g(\cdot)$ is optimized to convert radar embeddings into a soft prompt that the LLM can interpret without further fine-tuning. Unlike prior approaches that rely solely on implicit alignment through language modeling objectives, our method provides *semantically aligned* radar tokens as input to the LLM. The pre-aligned, clutter-aware, robust, and semantically consistent token-level representations significantly reduce the modality gap, allowing the frozen LLM to effectively attend to fine-grained radar features and support tasks such as zero-shot question answering, radar–text retrieval, and motion-aware reasoning.

5 Radar–Text Pair Construction

Action Annotation. We first integrate three mmWave datasets (mmBody [36], MM-Fi [37], and mRI [38]) that provide RGB recordings together with radar point clouds. The RGB recordings are used as visual references for annotation. When the visual observations are degraded, occluded, or otherwise ambiguous, we additionally generate skeleton-based visualizations to help annotators identify the corresponding human motions. Three annotators inspect the visual references, segment the recordings into temporal intervals containing actions, and assign an action label. After the annotation, the annotators cross-review their annotations and consolidate them into a final set.

Description and QA Construction. For each action segment, annotators write a natural-language description based on the action label and the corresponding visual reference. They also manually construct multiple-choice questions, including the correct answer and distractor options. Beyond action-related QA, we include body-part questions that ask whether individual limbs, such as the left and right arms and legs, are involved in the motion. We further construct an action recognition task, where the model identifies the performed action without multiple-choice options.

LLM Paraphrasing. Building on the human-constructed radar-text pairs, we apply LLM-based paraphrasing to increase linguistic diversity while preserving their semantic intent. This augmentation reduces the reliance of the radar encoder and downstream language model on a limited set of sentence structures or fixed linguistic templates and improves generalization to diverse textual expressions.

Quality Verification. To ensure the reliability of the resulting QA data and the balance of the dataset, we conduct a quality verification. Three annotators independently review every generated description and QA pair. If even a single reviewer judges a sample to be ambiguous or semantically unclear, it is discarded and replaced with a version rewritten by a human annotator.

6 Experimental Results

6.1 Experimental Setup

Datasets. We evaluate the proposed mmLIP encoder on the held-out test splits of the datasets described in Sec. 5. For mmBody [36], we follow the official train–test split provided by the dataset,

Table 1: Bidirectional radar–text retrieval results. Performance is evaluated Recall@K (R@1, R@5, R@10). Pre-Enc. denotes the pretrained encoder, Rep. denotes the representation level (CLS or token-level), and Obj. denotes the training objective. R, V, and T indicate radar, vision, and text encoders, respectively. Best results are highlighted in **bold** and second-best results are underlined.

Method	Training Configuration			Radar to Text									Text to Radar		
	Pre-Enc.	Rep.	Obj.	Description			QA			Action Recognition			Description		
				R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Vision-based	V	CLS	$\mathcal{L}_{\text{Const}}$	0.91	1.82	7.27	0.00	4.55	10.91	0.91	4.55	10.91	0.00	3.64	4.55
	V	Token	$\mathcal{L}_{\text{Token}}$	4.55	6.36	9.09	1.82	6.36	12.73	2.73	10.00	13.64	0.91	2.73	6.36
	V		$+\mathcal{L}_{\text{pcl}}$	4.55	8.18	19.09	1.82	9.09	16.36	2.73	10.91	17.27	4.55	5.45	8.18
Vision-anchored	V+T	CLS	$\mathcal{L}_{\text{Const}}$	5.45	8.18	19.09	0.91	6.36	14.55	2.73	10.91	14.55	3.64	8.18	9.09
	V+T	Token	$\mathcal{L}_{\text{Token}}$	11.82	22.73	27.27	14.55	25.45	34.55	11.82	22.73	31.82	12.73	27.27	35.45
	V+T	Token	$+\mathcal{L}_{\text{pcl}}$	15.45	22.73	30.00	**15.45**	28.18	38.18	12.73	24.55	32.73	19.09	28.18	35.45
Projector-based	-	CLS	$\mathcal{L}_{\text{Const}}$	1.82	5.45	10.91	1.82	8.18	12.73	1.82	10.00	15.45	4.55	9.09	10.91
mmLIP (Ours)	T	CLS	$\mathcal{L}_{\text{Const}}$	18.18	27.27	34.55	11.82	28.18	38.18	12.73	30.91	38.18	21.82	32.73	41.82
	T	Token	$\mathcal{L}_{\text{Token}}$	**20.00**	27.27	34.55	14.55	30.00	38.18	16.36	31.82	41.82	20.91	32.73	44.55
	T	Token	$+\mathcal{L}_{\text{pcl}}$	**20.00**	**29.09**	**35.45**	**15.45**	**30.91**	**42.73**	**18.18**	**32.73**	**43.64**	**22.73**	**35.45**	**49.09**

where the test set includes unseen actions and subjects. For MM-Fi [37] and mRI [38], we adopt cross-subject splits to evaluate generalization to unseen subjects (see Appendix C for more details).

Compared Encoder Designs. We compare three representative encoder paradigms for radar–text alignment. We reproduce the design philosophy of prior mmWave-to-LLM approaches [11, 12] and evaluate them under the evaluation protocol described earlier. First, following [12], we consider vision-based alignment, which projects radar features into a vision-centric embedding space using a pretrained vision encoder (*i.e.*, CLIP). Second, vision-anchored alignment performs multi-stage alignment by first mapping radar to the vision space and subsequently aligning it with text representations. Third, projector-based alignment employs a radar encoder (*e.g.*, an MLP) that is jointly trained with the LLM during language fine-tuning, implicitly learning radar–text alignment without explicit pretraining, as in prior work [11]. Finally, mmLIP (ours) directly aligns radar tokens with text embeddings in a shared space without relying on intermediate modalities. For each paradigm, we consider two types of representation during training: a CLS-based representation and a token-level representation. The token-level representation is trained using two losses, L_{token} and L_{pcl}.

Tasks. We consider two tasks to assess the effectiveness of our approach. (1) Bidirectional radar-text retrieval: we perform retrieval in both radar-to-text and text-to-radar directions. The text candidates include human action descriptions, question-answer pairs, and action recognition (AR) labels, with all retrieval tasks conducted in a *zero-shot manner*. (2) LLM-based text generation: mmLIP representations are integrated with an LLM and vision language model (VLM) for conditional text generation. Closed-ended tasks (*e.g.*, AR and limb-level QA) and open-ended tasks (*e.g.*, description, QA, and AR) are both considered to evaluate the LLM's text generation capabilities.

Evaluation Metrics. Retrieval performance is measured using Recall@1, Recall@5, and Recall@10. Given a radar query, we rank candidate text embeddings based on cosine similarity and report the top-K retrieval accuracy. We perform instance-level retrieval: a retrieval is counted as correct only when one of the paired ground-truth texts for the radar query appears within the top-K results. Texts paired with other radar samples are treated as negatives, even when they correspond to the same action category. The text-to-radar direction is evaluated symmetrically. For text generation, we adopt different evaluation metrics depending on the task type. Closed-ended tasks are evaluated using accuracy (%) to measure exact match performance, while open-ended tasks are evaluated using METEOR [39] and LLM-as-Judge [40, 41] to assess semantic correctness and response quality.

6.2 Comparison of Alignment Paradigms

Bidirectional Retrieval Performance. Table 1 presents bidirectional radar–text retrieval results across different encoder paradigms and training configurations. mmLIP consistently outperforms all baseline encoder paradigms across both radar-to-text and text-to-radar retrieval tasks. In particular, compared to vision-anchored alignment, mmLIP improves R@1 by approximately 4-9% across different tasks, and shows even larger gains at higher recall levels (*e.g.*, R@10). Compared to projector-based alignment, the improvement is more significant, with gains of over 15% in R@1

Table 2: Comparison of our method with baseline approaches for LLM and VLM integration. We report Accuracy (%) for closed-ended tasks and METEOR (%) / projector-based evaluation for open-ended tasks. Best results are highlighted in **bold**, second-best results are underlined, and $\Delta_{\text{2nd-best}}$ denotes the absolute improvement of mmLIP over the second-best result.

Model Type	Method	Model	Param	Closed-ended		Open-ended		
				AR	Limb	Description	QA	AR
LLM	Projector-based	Phi-3.5	3.8B	17.6	39.1	33.9/51.2	8.5/55.3	34.6/13.2
	Vision-based			19.4	43.5	33.9/43.0	5.9/57.8	44.4/25.4
	Vision-anchored			18.5	51.2	34.9/53.4	10.5/58.6	44.2/23.0
	mmLIP (Ours)			**27.8**	**61.6**	**49.1/64.4**	**19.5/67.6**	**47.3/51.2**
	$\Delta_{\text{2nd-best}}$			+8.4	+10.4	+14.2/+11.0	+9.0/+9.0	+2.9/+25.8
	Projector-based	Mistral	7B	26.9	50.7	40.8/56.9	15.3/65.2	51.2/30.3
	Vision-based			27.8	52.3	33.9/52.3	15.9/67.3	62.3/31.8
	Vision-anchored			30.6	57.2	40.8/57.9	16.6/70.3	54.3/27.5
	mmLIP (Ours)			**35.2**	**65.7**	**55.0/69.8**	**24.9/74.6**	**71.6/51.2**
	$\Delta_{\text{2nd-best}}$			+4.6	+8.5	+14.2/+11.9	+8.3/+4.3	+9.3/+19.4
	Projector-based	Qwen2.5	7B	24.1	51.2	33.9/55.6	15.6/39.2	47.3/43.4
	Vision-based			27.8	53.8	34.9/57.5	16.32/49.2	58.3/52.8
	Vision-anchored			26.9	53.6	37.9/64.9	19.3/60.3	55.6/49.3
	mmLIP (Ours)			**32.4**	**61.6**	**54.9/74.9**	**24.6/77.2**	**70.6/57.9**
	$\Delta_{\text{2nd-best}}$			+4.6	+7.8	+17.0/+10.0	+5.3/+16.9	+12.3/+5.1
	Projector-based	LLaMA-3.1	8B	22.2	44.5	39.4/60.4	18.3/63.3	52.3/46.2
	Vision-based			27.8	52.3	40.8/64.4	21.3/73.4	55.1/49.5
	Vision-anchored			30.6	54.2	39.4/62.8	19.5/61.9	59.0/50.3
	mmLIP (Ours)			**34.2**	**57.2**	**57.3/77.1**	**23.2/76.1**	**70.8/67.6**
	$\Delta_{\text{2nd-best}}$			+3.8	+3.0	+16.5/+12.7	+1.9/+2.7	+11.8/+17.3
VLM	Projector-based	Phi-3.5-Vision	4.2B	17.6	39.9	34.9/42.5	16.1/38.3	40.8/38.6
	Vision-based			11.1	43.5	37.9/43.3	10.4/25.4	41.7/42.1
	Vision-anchored			26.9	51.2	37.9/47.4	18.2/51.8	44.5/34.7
	mmLIP (Ours)			**31.5**	**59.0**	**54.5/67.9**	**24.1/72.6**	**70.6/71.5**
	$\Delta_{\text{2nd-best}}$			+4.6	+7.8	+16.6/+20.5	+5.9/+20.8	+26.1/+29.4
	Projector-based	LLaVA-OV	7B	27.8	50.7	40.9/50.9	17.6/59.3	48.5/40.9
	Vision-based			27.8	52.3	50.9/53.6	18.2/62.3	51.8/55.3
	Vision-anchored			30.6	57.9	33.9/40.5	14.3/53.2	52.9/57.6
	mmLIP (Ours)			**35.2**	**63.6**	**52.6/68.2**	**24.5/74.5**	**69.7/70.9**
	$\Delta_{\text{2nd-best}}$			+4.6	+5.7	+1.7/+14.6	+6.3/+12.2	+16.8/+13.3
	Projector-based	Qwen2-VL	7B	22.2	39.1	33.9/50.9	8.5/67.5	48.9/39.2
	Vision-based			27.8	50.7	49.1/61.8	10.8/68.3	51.2/51.2
	Vision-anchored			27.8	61.6	47.1/55.2	8.6/66.9	47.3/57.3
	mmLIP (Ours)			**38.9**	**66.4**	**55.0/69.9**	**25.3/75.3**	**72.2/72.6**
	$\Delta_{\text{2nd-best}}$			+11.1	+4.8	+5.9/+8.1	+14.5/+7.0	+21.0/+15.3

and more than 20% in R@10 in several cases. These results indicate that relying on intermediate vision representations or lightweight tokenizers is insufficient for accurately capturing radar-specific complex structures in radar data. Vision-anchored methods suffer from information loss due to modality mismatch and multi-stage projection, while projector-based alignment is limited by the capacity of the MLP-based tokenizer. In contrast, mmLIP directly aligns radar tokens with text embeddings in a shared space, preserving radar-specific attributes such as spatial structure and motion cues, which leads to more accurate and robust cross-modal retrieval.

Comparison of Training Configuration. We compare four radar point encoder training paradigms under different training settings, including CLS-based, token-level, and PCL. Vision-based and vision-anchored methods underperform due to the reliance on intermediate vision spaces, which introduce information loss and modality mismatch. Projector-based alignment improves over vision-based approaches but remains limited by the use of a lightweight radar tokenizer. In contrast, mmLIP directly aligns radar tokens with text embeddings, consistently achieving the best performance across all tasks. We further analyze the impact of representation choices and training objectives on radar-text alignment. Token-level representations consistently outperform CLS-based global representations across all methods, highlighting the importance of preserving fine-grained structure in radar data, which is inherently sparse and locally distributed. Furthermore, applying only the token-level loss improves performance over CLS-based training, but incorporating PCL yields additional gains. Compared to both CLS-based and token-only configurations, the combination of token-level representation with PCL achieves the best performance, indicating that reweighting radar tokens based on their relevance helps suppress noise and enables more reliable cross-modal alignment.

6.3 Comparison with Existing Radar-Language Methods

We further compare mmLIP with representative prior radar–language methods, including Radar-LLM [11], HoloLLM [12], and mmCLIP [17]. Since these methods were originally trained and evaluated under different datasets and protocols, we retrain all baselines on our training data and evaluate them under the same downstream tasks and evaluation settings for a fair comparison. For mm-

Table 3: Comparison with existing radar–language methods under the same training and evaluation protocol. We report Accuracy (%) for closed-ended tasks and METEOR (%) / LLM-as-Judge scores for open-ended tasks. Best results among point-cloud-based methods are highlighted in **bold**.

Model	Closed-ended		Open-ended		
	AR	Limb	Description	QA	AR
mmCLIP (heatmap) [17]	39.84	67.51	57.9 / 71.6	28.3 / 77.1	74.4 / 76.54
mmCLIP (point cloud) [17]	30.56	38.43	36.5 / 43.4	16.3 / 36.2	41.3 / 36.6
HoloLLM [12]	25.93	39.58	33.3 / 51.6	8.2 / 55.1	32.1 / 12.7
RadarLLM [11]	22.22	61.11	48.5 / 54.1	8.3 / 67.1	45.8 / 58.3
mmLIP (Ours)	**38.90**	**66.40**	**55.0 / 69.9**	**25.3 / 75.3**	**72.2 / 72.6**

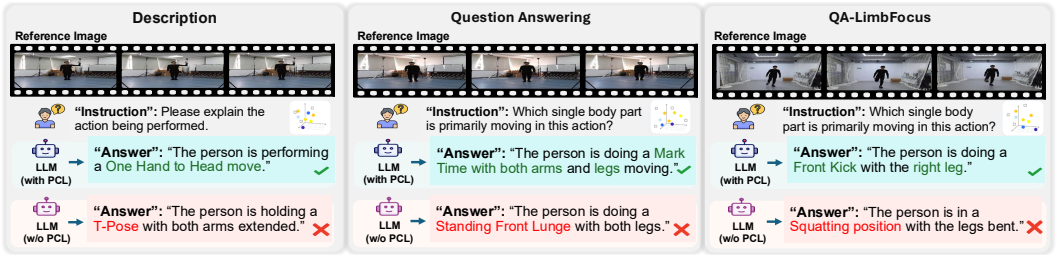

Figure 3: Qualitative comparison of LLM outputs with and without PCL across description, QA, and limb-focused QA tasks. The reference images and corresponding instructions are shown for each example, where green and red text indicate correct and incorrect responses, respectively.

CLIP, we report both its original heatmap-based formulation and a point-cloud variant to disentangle the effect of the input representation from that of the learned radar–language representation.

Among point-cloud-based approaches, mmLIP consistently outperforms all baselines across every task. In particular, under the same point-cloud input setting, mmLIP substantially improves over mmCLIP in both closed-ended action recognition (38.90 vs. 30.56) and limb-level recognition (66.40 vs. 38.43). These results indicate that the performance gains are primarily attributable to the proposed radar-specific representation learning and alignment objectives rather than differences in input modality. Notably, mmLIP also achieves performance comparable to the heatmap-based mmCLIP despite directly operating on sparse radar point clouds. While radar heatmaps preserve denser reflection information, they generally require substantially greater storage, memory, and computation than sparse point-cloud representations. The strong performance of mmLIP therefore demonstrates a favorable trade-off between representation efficiency and radar–language understanding, supporting point-cloud-based alignment as a scalable direction for multimodal radar reasoning.

6.4 Integration with Large Language Models

We evaluate the effect of radar–text alignment on downstream performance when integrated with LLMs [42–45] and VLMs [42, 46, 47]. Aligned radar tokens are injected via a learnable projector, and evaluated on both closed-ended and open-ended tasks. Table 2 shows that mmLIP consistently outperforms all baselines across both LLMs and VLMs. For example, on LLaMA-3.1, action recognition accuracy improves from 30.6 (vision-anchored) to 34.2, and limb-level accuracy increases from 54.2 to 57.2. Similar gains are observed across other models, demonstrating the effectiveness of direct token-level radar–text alignment. Overall, VLMs tend to achieve higher performance than LLMs, likely due to their prior exposure to vision-centric multimodal pretraining, which provides stronger cross-modal representation capabilities. Nevertheless, approaches leveraging LLMs tend to achieve higher evaluation scores, as LLM-generated outputs are generally more fluent and coherent. However, other methods tend to underutilize these benefits, likely due to the absence of explicit token-level radar–text alignment. In contrast, mmLIP explicitly models token-level alignment, allowing it to better capture radar-specific semantics and achieve more effective cross-modal reasoning.

6.5 Qualitative Analysis

To intuitively demonstrate the effectiveness of mmLIP, we visualize radar–text similarity maps in Fig. 4 and present qualitative comparisons of LLM outputs with and without PCL in Fig. 3.

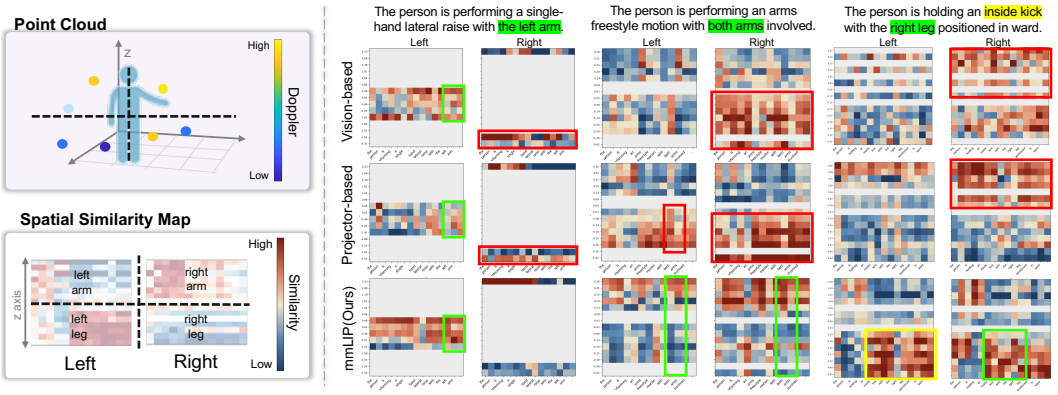

Figure 4: **Left**: An example of the similarity score visualization, where the spatial similarity map highlights the correspondence between radar points and textual tokens. **Right**: Comparison of similarity maps across three methods (vision-based, projector-based, and our mmLIP) for three human pose descriptions. Highlighted boxes denote key regions; notably, red boxes indicate incorrectly high similarity, whereas other boxes emphasize meaningful correspondences with the text.

Qualitative Results. Fig. 3 presents qualitative comparisons of LLM outputs with and without PCL across open-ended tasks, including description, question answering, and limb-focused QA. Without PCL, the model often produces incorrect or imprecise responses, failing to capture fine-grained motion details or correctly identify relevant body parts (e.g., both arms or the right leg). For example, the model without PCL incorrectly predicts a "T-Pose" or "Squatting" whereas the model with PCL correctly recognizes motions. These results demonstrate that confidence-aware token reweighting effectively suppresses clutter and enables more reliable grounding in radar-based reasoning.

Similarity Visualization. Fig. 4 presents qualitative visual comparisons of radar–text similarity maps across different alignment strategies. Vision-based and projector-based methods often exhibit diffuse or misplaced high-similarity regions, incorrectly activating irrelevant tokens or failing to localize fine-grained correspondences. In contrast, mmLIP produces more precise and localized similarity patterns, correctly aligning salient radar points with semantically relevant textual tokens (e.g., specific limbs or actions). These results indicate that explicit token-level alignment enables more accurate cross-modal grounding and better captures fine-grained radar semantics.

7 Conclusion

We address the problem of aligning radar point clouds with language representations at the token-level and propose **mmLIP**, a radar–language interactive pretraining method. mmLIP directly aligns radar tokens with textual tokens in a shared embedding space, enabling fine-grained meaningful radar–text correspondence that facilitates effective integration with LLMs, while PCL suppresses clutter and emphasizes informative signals. The learned representations capture structured radar semantics, enabling zero-shot bidirectional retrieval and generalizing effectively to diverse text-driven tasks, including generation and question answering, when integrated with LLMs. Experimental results demonstrate that mmLIP achieves robust performance across diverse LLMs and VLMs, highlighting the importance of explicit, token-level radar-specific alignment for multimodal reasoning.

Limitations and Future Works. Despite these promising results, our current study focuses primarily on human-centric scenarios. A natural next step is to scale mmLIP toward a foundation model by leveraging larger and more diverse datasets. In particular, extending beyond human pose understanding to broader applications such as autonomous driving and general scene understanding is a key future direction. We believe this direction will further improve generalization and enable more robust multimodal reasoning across diverse real-world settings.

Acknowledgments and Disclosure of Funding

This work was supported in part by the InnoCORE program of the Ministry of Science and ICT (No. N10250156), the Artificial Intelligence industrial convergence cluster development Project funded by the Ministry of Science and ICT (MSIT, Korea) and Gwangju Metropolitan City, the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25525987, Digital Columbus Project), IITP grant funded by the Korea government (MSIT) (No. RS-2025-02219277, AI Star Fellowship Support Project (DGIST)), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25490788). (Corresponding author: Jae-Ho Choi.)

References

[1] Fangqiang Ding, Zhen Luo, Peijun Zhao, and Chris Xiaoxuan Lu. milliflow: Scene flow estimation on mmwave radar point cloud for human motion sensing. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXIV*, page 202–221, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72690-3. doi: 10.1007/978-3-031-72691-0_12. URL https://doi.org/10.1007/978-3-031-72691-0_12.

[2] Zijian Gu, Jianwei Ma, Yan Huang, Honghao Wei, Zhanye Chen, Hui Zhang, and Wei Hong. Hgsfusion: radar-camera fusion with hybrid generation and synchronization for 3d object detection. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i3.32328. URL https://doi.org/10.1609/aaai.v39i3.32328.

[3] Jaeho Choi, Soheil Hor, Shubo Yang, and Amin Arbabian. Mvdoppler-pose: Multi-modal multi-view mmwave sensing for long-distance self-occluded human walking pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27750–27759, June 2025.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, 2023. URL https://arxiv.org/abs/2302.13971.

[5] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[7] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and .... GPT-4 Technical Report, 2023. URL https://arxiv.org/abs/2303.08774.

[8] OpenAI. GPT-4o System Card, 2024. URL https://arxiv.org/abs/2410.21276.

[9] Yifan Yan, Shuai Yang, Xiuzhen Guo, Xiangguang Wang, Wei Chow, Yuanchao Shu, and Shibo He. mmexpert: Integrating large language models for comprehensive mmwave data synthesis and understanding. In *Proceedings of the Twenty-Sixth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '25, page 1–10, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713538. doi: 10.1145/3704413.3764420. URL https://doi.org/10.1145/3704413.3764420.

[10] Jie Yang, Wanchen Hu, Shitong Bai, Shuangyang Li, Siyao Li, Kai Wan, and Xin Wang. Large language models for beam prediction under the mmwave massive mimo hybrid-field channels. *IEEE Communications Letters*, 30:627–631, 2026. doi: 10.1109/LCOMM.2025.3647492.

[11] Zengyuan Lai, Jiarui Yang, Songpengcheng Xia, Lizhou Lin, Lan Sun, Renwen Wang, Jianran Liu, Qi Wu, and Ling Pei. Radarllm: Empowering large language models to understand human motion from millimeter-wave point cloud sequence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40:5791–5799, 2026.

[12] Chuhao Zhou and Jianfei Yang. HoloLLM: Multisensory foundation model for language-grounded human sensing and reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=cHMP2IAhML.

[13] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[15] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, page 776–794, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58620-1. doi: 10.1007/978-3-030-58621-8_45. URL https://doi.org/10.1007/978-3-030-58621-8_45.

[16] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EqoXe2zmhrh.

[17] Qiming Cao, Hongfei Xue, Tianci Liu, Xingchen Wang, Haoyu Wang, Xincheng Zhang, and Lu Su. mmclip: Boosting mmwave-based zero-shot har via signal-text alignment. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, SenSys '24, page 184–197, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706974. doi: 10.1145/3666025.3699331. URL https://doi.org/10.1145/3666025.3699331.

[18] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, October 2023.

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[20] Mohamed O Abdelfattah, Kaouther Messaoud, and Alexandre Alahi. OSKAR: Omnimodal self-supervised knowledge abstraction and representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=LWuhOoHpo5.

[21] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

[22] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander J Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 27092–27112. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/56332d41d55ad7ad8024aac625881be7-Paper-Datasets_and_Benchmarks.pdf.

[23] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

[24] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. 2022.

[25] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. Llvip: A visible-infrared paired dataset for low-light vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3496–3504, October 2021.

[26] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, and Liu. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, June 2022.

[27] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12475–12486, June 2021.

[29] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL `https://arxiv.org/abs/1807.03748`.

[31] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/jia21b.html`.

[32] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=cpDhcsEDC2`.

[33] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[34] Jitesh Jain, Jianwei Yang, and Humphrey Shi. VCoder: Versatile Vision Encoders for Multimodal Large Language Models. In *CVPR*, 2024.

[35] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, Kavya Srinet, Babak Damavandi, and Anuj Kumar. AnyMAL: An efficient and scalable any-modality augmented language model. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1314–1332, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.98. URL `https://aclanthology.org/2024.emnlp-industry.98/`.

[36] Anjun Chen, Xiangyu Wang, Shaohao Zhu, Yanxu Li, Jiming Chen, and Qi Ye. mmBody Benchmark: 3D Body Reconstruction Dataset and Analysis for Millimeter Wave Radar. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3501–3510, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548262. URL `https://doi.org/10.1145/3503161.3548262`.

[37] Jianfei Yang, He Huang, Yunjiao Zhou, Xinyan Chen, Yuecong Xu, Shenghai Yuan, Han Zou, Chris Xiaoxuan Lu, and Lihua Xie. MM-Fi: Multi-Modal Non-Intrusive 4D Human Dataset for Versatile Wireless Sensing. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=1uAsASS1th`.

[38] Sizhe An, Yin Li, and Umit Ogras. mRI: Multi-modal 3D Human Pose Estimation Dataset using mmWave, RGB-D, and Inertial Sensors. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=0a2-cdfBxun.

[39] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.

[40] Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.306. URL https://aclanthology.org/2025.findings-acl.306/.

[41] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

[42] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, and Vishrav Chaudhary. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

[43] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[44] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[45] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, and Dhruv Mahajan. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[46] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. URL https://arxiv.org/abs/2408.03326.

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

A Point-wise Confidence Learning

Given a batch of radar samples and text descriptions, our goal is to estimate the importance of each radar point based on its semantic alignment with text. For a radar sample b, we first compute the cosine similarity between each radar point embedding $z_{r,i}^{(b)}$ and each text embedding $z_t^{(j)}$:

$$s_{i,j}^{(b)} = \frac{\left(z_{r,i}^{(b)}\right)^\top z_t^{(j)}}{\left\|z_{r,i}^{(b)}\right\| \left\|z_t^{(j)}\right\|}. \tag{1}$$

This results in a similarity matrix $S^{(b)} \in \mathbb{R}^{P \times B}$, where each entry measures how well a radar point matches a text description. To determine whether a radar point carries meaningful semantic information, we compute a relevance score by selecting the maximum similarity across all texts:

$$c_i^{(b)} = \max_j s_{i,j}^{(b)}. \tag{2}$$

This operation evaluates whether a given radar point is semantically aligned with any text in the batch. Importantly, we take the maximum over the text dimension rather than over points, as our objective is to assess the relevance of each individual point independently.

Algorithm 1: PCL

Input: Radar embeddings $\{z_{r,i}^{(b)}\}_{i=1}^{P}$, text embeddings $\{z_t^{(j)}\}_{j=1}^{B}$
Output: Weighted contrastive loss $\mathcal{L}_{\text{pcl}}$
for *each radar sample* b **do**
  for *each radar point* i **do**
    Compute similarity:
    $s_{i,j}^{(b)} \leftarrow \frac{(z_{r,i}^{(b)})^\top z_t^{(j)}}{\|z_{r,i}^{(b)}\| \|z_t^{(j)}\|} \quad \forall j$
    Relevance score:
    $c_i^{(b)} \leftarrow \max_j s_{i,j}^{(b)}$
    Confidence weight:
    $\tilde{w}_i^{(b)} \leftarrow \sigma(c_i^{(b)}/\tau_w)$
    $w_i^{(b)} \leftarrow \texttt{stopgrad}(\tilde{w}_i^{(b)})$
  Compute weighted loss:
  $\mathcal{L}_{\text{pcl}}^{(b)} \leftarrow \sum_i w_i^{(b)} \ell_i^{(b)}$
return $\mathcal{L}_{\text{pcl}} = \sum_b \mathcal{L}_{\text{pcl}}^{(b)}$

We then convert the relevance score into a confidence weight using a sigmoid function:

$$\tilde{w}_i^{(b)} = \sigma\left(\frac{c_i^{(b)}}{\tau_w}\right), \tag{3}$$

where τ_w is a temperature parameter that controls the sharpness of the weighting distribution. A smaller τ_w leads to a more peaked distribution, emphasizing highly relevant points, while a larger τ_w produces smoother weights. However, directly using $\tilde{w}_i^{(b)}$ as a learnable weight introduces a degenerate optimization issue. Since the weights depend on similarity scores, gradients can propagate through both the loss term and the weight itself. Specifically, the gradient of the loss can be written as:

$$\frac{\partial \mathcal{L}}{\partial s} = w_i \frac{\partial \ell_i}{\partial s} + \ell_i \frac{\partial w_i}{\partial s}. \tag{4}$$

The second term allows the model to reduce the overall loss by decreasing the weight w_i, rather than improving the alignment between radar points and text. This leads to a trivial solution where the model suppresses difficult points instead of learning meaningful representations.

To prevent this behavior, we apply a stop-gradient operation:

$$w_i^{(b)} = \text{stopgrad}\left(\tilde{w}_i^{(b)}\right). \tag{5}$$

This removes the gradient flow through the weight, ensuring that:

$$\frac{\partial w_i}{\partial s} = 0, \tag{6}$$

and the gradient simplifies to:

$$\frac{\partial \mathcal{L}}{\partial s} = w_i \frac{\partial \ell_i}{\partial s}. \tag{7}$$

Finally, the overall objective is defined as a weighted sum of point-wise contrastive losses:

$$\mathcal{L}_{\text{pcl}} = \sum_i w_i^{(b)} \ell_i^{(b)}, \tag{8}$$

where $\ell_i^{(b)}$ denotes the contrastive loss for the i-th radar point. In this formulation, the confidence score serves as a fixed importance weight during optimization, allowing the model to focus on semantically meaningful points while avoiding degenerate solutions.

B Evaluation

We evaluate each radar-conditioned LLM on two zero-shot tasks defined over the test set: (i) a four-way *multiple-choice action recognition* (**MCQ**) task, and (ii) a per-limb *yes/no engagement* (**Limb-YN**) task. Both are scored *without* free-form generation — we read scores directly from the language model's next-token logits. This removes confounders from format compliance, sampling, and stop-token behaviour, and yields a measure of *what the model knows* rather than *whether it can phrase the answer*.

B.1 Task Formulation

Let $\mathbf{x}$ denote a radar point cloud, $\mathbf{e} = E(\mathbf{x})$ its encoder embedding (frozen), and $\mathbf{r} = P(\mathbf{e})$ the projected "radar token" sequence consumed by the LLM (frozen). For a textual prompt q, the LLM produces contextualised logits $\boldsymbol{\ell}(\mathbf{r}, q) \in \mathbb{R}^V$ over its vocabulary V at each position; we denote the logit at the final prompt position by $\boldsymbol{\ell}_{\text{end}}(\mathbf{r}, q)$.

MCQ. For sample i with ground-truth action $a_i^\star$, we draw three distractor actions from the action vocabulary $\mathcal{A}$ uniformly at random, conditioning the random seed on the sample key so that all models see identical choices for the same sample. The four candidates are shuffled and labelled $\{A, B, C, D\}$, yielding a mapping $c : \{A, B, C, D\} \to \mathcal{A}$ with $c(\ell^\star) = a_i^\star$ for some letter $\ell^\star$. The MCQ prompt q^{mcq} ends with the literal token `Answer:`.

Limb-YN. For each limb $L \in \mathcal{L} = \{\text{left arm}, \text{right arm}, \text{left leg}, \text{right leg}\}$ we ask whether L is actively involved. The ground-truth label is `yes` iff L appears in the entry's `QA-LimbFocus` list (otherwise `no`, including the case where the list contains the sentinel `none`). The Limb-YN prompt q_L^{ln} ends with `Answer:` and is preceded by a balanced two-shot illustration (one `yes`, one `no`, on dummy questions without radar input) to fix the answer surface form.

B.2 LLM-as-Judge Evaluation

Token-overlap metrics such as BLEU, ROUGE, and METEOR often fail to capture semantic correctness in radar-conditioned text generation. For example, a prediction may describe the correct action using different wording, or may copy reference words while describing an incorrect action. To better evaluate semantic quality, we use GPT-4.1 as an LLM-as-Judge.

For each test sample, the judge evaluates three outputs: `sentences`, `ActionRec`, and `QA`. Each output is compared against the ground-truth action label and scored using four criteria: action correctness, hallucination, action consistency, and linguistic quality.

B.2.1 Evaluation Rubric

Each criterion is assigned an integer score from 0 to 5. The final score is obtained by rescaling the sum of the four scores by 0.05, so that each task receives a score in $[0, 1]$.

Criterion	Question	Scoring Guide
Same Action	Does the output describe the same action as the ground truth?	5: exact same action or clear synonym. 3: related action in the same family. 0: unrelated or completely wrong action.
No Hallucination	Does the output avoid unsupported details?	5: all details are supported. 3: some plausible but unsupported details. 0: clearly hallucinated content.
No Other Action	Does the output mention only the target action?	5: only the target action is mentioned. 3: mostly correct but briefly mentions another action. 0: describes a different or multiple unrelated actions.
Linguistic Quality	Is the output fluent and on-topic?	5: fluent and grammatical. 3: understandable but awkward or repetitive. 0: broken, off-topic, or heavily repeated.

Table 4: Rubric used for LLM-as-Judge evaluation. Each criterion is scored from 0 to 5.

Field	Content
Ground Truth Action	The target action label for the sample.
Reference Sentences	Five ground-truth reference descriptions.
Model Outputs	The generated outputs for `sentences`, `ActionRec`, and `QA`.
Rubric	The four evaluation criteria: Same Action, No Hallucination, No Other Action, and Linguistic Quality.
Output Format	A JSON object containing scores from 0 to 5 and a one-sentence rationale for each task.

Table 5: Input and output format of the LLM-as-Judge prompt.

B.2.2 Scoring

For each task $\tau \in \{\texttt{sentences}, \texttt{ActionRec}, \texttt{QA}\}$, the per-sample judge score is defined on a 100-point scale as:

$$\mathrm{Judge}_\theta^\tau(i) = 5\left(s_A^{\tau,\theta,i} + s_B^{\tau,\theta,i} + s_C^{\tau,\theta,i} + s_D^{\tau,\theta,i}\right), \tag{9}$$

where s_A, s_B, s_C, and s_D denote the four rubric scores. Each score is an integer in $\{0, 1, 2, 3, 4, 5\}$, so the maximum per-task score is $5 \times 20 = 100$.

The model-level score for each task is the average over all test samples:

$$\mathrm{Judge}_\theta^\tau = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Judge}_\theta^\tau(i). \tag{10}$$

We also report an overall judge score by averaging the three task scores:

$$\mathrm{Judge}_\theta = \frac{1}{3}\sum_{\tau} \mathrm{Judge}_\theta^\tau. \tag{11}$$

B.2.3 Judge Prompt

Each judge call provides the ground-truth action, reference descriptions, and the model outputs for the three tasks. The judge is asked to return a JSON object containing the four scores and a short rationale for each task. We use GPT-4.1 with temperature set to 0 and enforce JSON output format. All three task outputs for a sample are evaluated in a single judge call to keep the scoring consistent within each sample.

Table 6: Comparison between LLM-as-Judge and human evaluation on 400 open-ended generations.

Model	LLM-as-Judge	Human Eval.	Absolute Gap ↓
mmLIP (Ours)	76.5	75.2	1.3

B.3 LLM-as-Judge Evaluation

To validate the reliability of the LLM-as-Judge evaluation, we conduct a human evaluation on 400 open-ended generations using three raters. The raters assess each response using the same criteria as the LLM-as-Judge: (1) semantic consistency with the ground-truth action, (2) accurate and sufficient description of the involved body parts, and (3) sufficient specificity to verify the described motion. The average human evaluation score is 75.2, differing from the LLM-as-Judge score of 76.5 by only 1.3 points, indicating close agreement at the aggregate level. We further analyze generations that human raters identify as inconsistent with the reference or difficult to judge and identify three major error types: (1) **incomplete body-part coverage**, where multi-limb actions such as *Volleyball Spike* or *Side Kick* are described using only a subset of the involved body parts; (2) **static descriptions that omit motion direction or dynamics**, as in "The person is holding a One Hand Sweep position"; and (3) **ambiguous or overly generic expressions**, such as "active arms and legs" or "coordinated movement" for *Mark Time*, which do not sufficiently specify individual limb motions. Overall, the dominant errors arise from incomplete descriptions of simultaneous multi-limb movements and the omission of dynamic, fine-grained motion information rather than grammatical quality.

C Construction of Training and Test Data

We construct training and test data by integrating three public mmWave human-sensing datasets: mmBody [36], MM-Fi [37], and mRI [38]. These datasets cover diverse radar devices, environments, subjects, and action categories, providing a broad basis for mmWave–language alignment. After merging the original annotations, we refine 139 action labels into 86 distinct categories by removing redundant, ambiguous, and semantically overlapping labels. For mmBody, we follow the original training and test split provided by the dataset. For MM-Fi, we adopt a cross-subject split, using S10, S20, S30, and S40 for testing and the remaining 36 subjects for training. Similarly, for mRI, subjects 10 and 20 are reserved for testing, while the remaining 18 subjects are used for training. The training portions of the three datasets are combined for model training, while their respective test portions are used exclusively for evaluation.

Radar Standardization and Annotation References. To handle heterogeneous radar formats across datasets, we standardize raw mmWave point clouds into a unified representation. Specifically, we represent radar point attributes (x, y, z, Doppler, and intensity) by rounding them to two decimal places, yielding compact textual inputs suitable for LLM processing while preserving spatial structure. To support reliable human annotation, we provide visual references derived from the original recordings. Clear RGB videos are used directly when available, whereas degraded, occluded, or visually ambiguous scenes are converted into skeleton-based videos to help annotators understand the underlying human motion.

Question–Answer Construction. Based on the refined action taxonomy, annotators generate natural language question–answer pairs for each sample. Each question is designed to evaluate a specific aspect of mmWave point-cloud understanding, such as action recognition, body-motion reasoning, or fine-grained semantic interpretation. For multiple-choice QA, answer candidates are selected from the predefined action categories to ensure consistency across samples. To improve linguistic diversity and reduce template bias, we further paraphrase the annotated questions using LLMs while preserving their semantic intent.

Quality Review and Dataset Splitting. We conduct a multi-stage quality review to ensure annotation reliability and dataset balance. Three annotators independently inspect all QA pairs, and any pair marked as ambiguous, semantically unclear, or inconsistent by at least one annotator is removed. We also replace overly similar or redundant answer candidates with more discriminative options to reduce ambiguity during multiple-choice evaluation. Finally, we rebalance the QA distribution across

action domains and question types, split the resulting dataset into training and test sets, and use it for both mmLIP pretraining and downstream evaluation under diverse prompting strategies.

D Implementation Details

Architecture and computing resources. The radar tokenizer is a Transformer encoder with 4 layers, 4 heads, embedding dimension d=256, MLP ratio 2.0, and dropout 0.1. It ingests up to P_{max}=196 points, each represented as a 5-D feature (x, y, z, v_1, v_2) where v_1, v_2 are the two radar-specific return features. mmLIP couples this encoder to two pathways: (i) a *cross-modal attention* block (4 heads) that fuses the $P \times d$ point tokens with CLIP `ViT-B/32` text tokens projected to d, and (ii) a *standalone pool* that mean-pools point tokens and applies a 2-layer MLP into the contrastive space (d_{proj}=512). Our PCL variant additionally trains a per-point projection head used by the point-weighted contrastive loss. For LLM/VLM integration the projected radar tokens are mapped into the host model's hidden size by a 2-layer MLP ($d \to d_{\text{LLM}} \to d_{\text{LLM}}$) and prepended to the prompt embeddings. We evaluate seven LLM/VLM backbones: Mistral-7B-v0.3, LLaVA-OneVision-Qwen2-7B, Phi-3.5-mini-instruct, Phi-3.5-vision-instruct, Qwen2.5-7B-Instruct, Qwen2-VL-7B-Instruct, and Llama-3.1-8B. All experiments run on a workstation with 4× NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs (96 GB each); we use PyTorch 2.x with `transformers`, single-precision throughout (CLIP and the host LLM are kept in `float32`). One GPU is sufficient for any single job at our batch sizes; multiple GPUs are used only to parallelise across encoder/LLM configurations.

Stage 1: encoder pretraining. We pretrain the radar tokenizer with CLIP-style contrastive alignment on the joint mmBody/mmFI/mRI training split. For each batch of B=64 radar–text pairs, we compute two contrastive losses against the CLIP text embedding z_{cls}: one through the cross-attention pathway ($\mathcal{L}_{\text{cross}}$) and one through the standalone pool ($\mathcal{L}_{\text{stand}}$); the optimisation target is $\mathcal{L} = (1-\alpha)\,\mathcal{L}_{\text{cross}} + \alpha\,\mathcal{L}_{\text{stand}}$ with α=0.5, mirrored as a symmetric InfoNCE with a learnable temperature initialised to $\log(1/0.07)$. For ***mmLIP+PCL*** we add a point-weighted contrastive loss, where a sigmoid-confidence score $c_p \in [0, 1]$ over individual radar tokens reweights the contribution of each point; this term is annealed in by a curriculum factor α_e that decays linearly from 1 to 0 over the first 30 epochs, and its relative weight in the final loss is 0.3. CLIP text encoder weights are frozen throughout; only the Transformer, cross-attention, and projection heads are trained. We optimise with AdamW (learning rate 2×10^{-4}, weight decay 10^{-2}, β=$(0.9, 0.95)$) under a cosine schedule with linear warm-up over the first 5% of steps.

Stage 2: projector fine-tuning with a frozen LLM. We freeze the radar encoder and the host LLM, and train only the projector that bridges the two modalities. For each training step we randomly sample one of the four task heads – `sentences`, `ActionRec`, `QA` and `QA-LimbFocus` (5-way) – and use the corresponding GT text under a teacher-forced next-token cross-entropy loss applied only to the answer tokens (the prompt tokens are masked out with `ignore_index`). Two radar-token integration modes are supported and trained as separate configurations: (i) *mean-pool*, where the encoder's P tokens are pooled into a single d_{LLM}-dimensional "radar token" that is prepended to the prompt; and (ii) *per-point*, where every (unmasked) radar token is projected and prepended individually, yielding a sequence of P radar tokens. The projector is a 2-layer MLP with GELU activation (∼4 M trainable parameters at d_{LLM}=4096). We optimise with AdamW (learning rate 5×10^{-5}, weight decay 10^{-2}), linear warm-up of 5% steps and cosine decay. Batch size is 32 at sequence length up to 256 tokens including radar tokens. Each (LLM, encoder, mode) tuple is trained for 20 epochs and the best checkpoint by held-out token-F1 is retained; a single configuration finishes in 4–6 h on one GPU.

E Radar vs. Vision under Adverse Conditions

To evaluate the robustness of radar representations, we compare our method with vision-based models under diverse environmental conditions, including *Normal*, *Furnished*, *Rain*, *Smoke*, *Dark*, and *Occlusion*. For a fair comparison, we use the same language model for both modalities, feeding RGB inputs into a vision-language model (VLM) and radar inputs through our aligned radar representation. The comparison considers both retrieval and generation performance across description and question answering tasks.

As shown in Table 7, mmLIP consistently outperforms the vision-based model on the description generation task across all environmental conditions, indicating that radar representations provide reliable semantic cues for action-level understanding. Importantly, radar signals inherently encode motion-related information even within a single frame through Doppler measurements. This property allows radar-based representations to capture dynamic human motion cues without relying solely on visual appearance, leading to strong overall performance across diverse conditions. The gains are particularly large under visually degraded conditions such as *Dark* and *Occlusion*, where RGB-based perception becomes unreliable or even fails to capture visible human body structure. In these cases, mmLIP maintains robust performance by exploiting radar-specific cues such as motion, distance, and reflectivity.

Table 7: Comparison of vision- and radar-based models under varying environmental conditions on three radar–language tasks (LLM–as–Judge for Desc/QA, classification accuracy for QA-Limb).

	Qwen2-VL			**mmLIP (ours)**		
Cond.	Desc	QA	Limb	Desc	QA	Limb
Normal	0.359	0.746	**0.683**	**0.744**	**0.752**	0.540
Furn.	0.493	0.721	0.571	**0.779**	**0.814**	**0.714**
Rain	0.700	0.750	0.000	**0.800**	0.750	**1.000**
Smoke	0.360	**0.725**	**0.600**	**0.735**	0.720	**0.600**
Dark	0.375	0.000	0.000	**0.675**	**0.612**	**0.875**
Occl.	0.350	0.000	0.000	**0.788**	**0.812**	**0.750**
Total	0.374	0.490	0.309	**0.742**	**0.744**	**0.747**

For QA, mmLIP outperforms Qwen2-VL overall, achieving particularly clear improvements in challenging conditions such as *Furnished* and *Occlusion*. For limb-level classification, mmLIP also substantially outperforms Qwen2-VL overall, with particularly strong advantages under challenging conditions such as *Rain*, *Dark*, and *Occlusion*. These results indicate that radar representations are not only robust under adverse sensing conditions but also provide reliable cues for fine-grained body-part understanding. Overall, these results demonstrate that mmLIP provides robust radar-language representations for both semantic understanding and fine-grained recognition, with particularly strong advantages under adverse sensing conditions.

F Robustness Analysis

We further evaluate the robustness of mmLIP from two complementary perspectives: (i) robustness to irrelevant radar clutter at inference time, and (ii) sensitivity to noisy radar–text supervision during training.

Robustness to radar clutter. We first examine whether the learned radar representations remain reliable when the input point cloud contains additional irrelevant returns. Given a radar point cloud with N valid points, we inject irrelevant points at ratios $\rho \in \{0, 25, 50, 100, 200\}\%$ of N and evaluate description retrieval performance using R@1. Table 8 summarizes the results.

As the amount of injected clutter increases, all methods exhibit a performance drop. However, mmLIP with PCL consistently degrades more gracefully than the alternatives. At $\rho = 200\%$, it retains approximately 67% of its original R@1 performance, compared with approximately 48% for token-level training without PCL and 42% for the CLS-based baseline. These results indicate that token-level representations combined with $\mathcal{L}_{\mathrm{pcl}}$ improve robustness to irrelevant radar returns by suppressing clutter-sensitive features. Performance nevertheless decreases under extremely heavy clutter, suggesting that sufficiently dense irrelevant returns can still obscure meaningful target responses.

Sensitivity to annotation noise. We next investigate how sensitive mmLIP is to the quality of radar–text supervision. To simulate noisy annotations, we replace an increasing fraction of the training radar–text pairs with mismatched pairs and evaluate closed-ended QA accuracy. The results are reported in Table 9.

Performance consistently decreases as the proportion of mismatched supervision increases, showing that mmLIP depends directly on the quality of radar–text annotation. This result also supports the importance of the manual inspection and filtering procedures used during dataset construction, as noisy supervision substantially weakens the learned radar–language alignment.

Table 8: Robustness to injected radar clutter. For a point cloud containing N valid points, irrelevant points are added at different ratios of N. We report description retrieval R@1.

Rep.	Obj.	0%	25%	50%	100%	200%
CLS	$\mathcal{L}_{\text{const}}$	18.18	16.8	13.8	10.9	7.7
Token	$\mathcal{L}_{\text{token}}$	20.00	19.3	16.2	14.1	9.5
Token	$\mathcal{L}_{\text{token}} + \mathcal{L}_{\text{pcl}}$	**20.00**	**19.5**	**18.8**	**16.8**	**13.4**

Table 9: Sensitivity to noisy radar–text annotations. We inject mismatched radar–text pairs into training and report closed-ended QA accuracy (%).

Model	Task	0%	25%	50%
mmLIP (Ours)	AR	38.9	31.9	24.3
	Limb	66.4	57.9	48.3

G Qualitative

In this section, we provide a qualitative analysis of model behaviors across both closed-ended tasks (action recognition and limb-focused QA) and open-ended tasks (description, QA, and AR). We examine how different encoder paradigms—projector-based, vision-based, vision-anchored, and our mmLIP affect the resulting outputs when integrated with various LLM and VLM backbones. By comparing model predictions across tasks, we analyze differences in fine-grained reasoning, semantic consistency, and the ability to capture radar-specific cues, highlighting the strengths and limitations of each alignment strategy.

Closed-ended Task. Fig. 5 presents qualitative comparisons on closed-ended tasks, including action recognition and limb-focused QA. Baseline approaches frequently produce incorrect predictions under both normal and challenging conditions, indicating limited understanding of fine-grained radar semantics. In the normal scenario, projector-based and vision-based methods incorrectly predict "Twist," while the vision-anchored method predicts "Front Lunge," whereas mmLIP correctly recognizes the action as "Bowing." Similarly, under the occlusion setting, baseline methods fail to identify the primary moving body part, predicting either "none" or an incorrect limb. In contrast, mmLIP correctly identifies the "left leg," demonstrating stronger robustness to occlusion and improved fine-grained reasoning. These results suggest that explicit token-level radar–text alignment enables more accurate grounding of radar-specific motion cues and improves robustness across diverse environments.

Open-ended Task. As shown in Table 7, mmLIP consistently outperforms the vision-based model on description generation across all environmental conditions, demonstrating the robustness of radar-language representations for action-level semantic understanding. This advantage is particularly clear in visually challenging scenarios such as *Dark* and *Occlusion*, where RGB-based perception becomes unreliable due to poor illumination or limited visibility of the human body. In contrast, radar signals inherently encode motion-related information even within a single frame through Doppler measurements, allowing mmLIP to capture dynamic cues such as movement direction, velocity, distance, and reflectivity. As a result, radar–based representations maintain strong performance even when visual appearance cues are severely degraded or unavailable.

Beyond the physical sensing advantage of mmWave radar, the strong performance of mmLIP is further supported by its token-level radar–text alignment. Unlike radar encoders that compress point-cloud features into a single global representation or rely on implicit language-model supervision, mmLIP aligns individual radar tokens with textual semantics at a fine-grained level. This design helps preserve local motion and structural cues, including limb movement, body orientation, and action-specific Doppler patterns. Consequently, mmLIP achieves comparable or better QA performance in most conditions and remains competitive on limb-level QA, especially under adverse environments such as *Rain*, *Dark*, and *Occlusion*. Overall, these results suggest that mmLIP provides robust radar-language understanding by combining the sensing robustness of mmWave radar with fine-grained token-level semantic alignment.

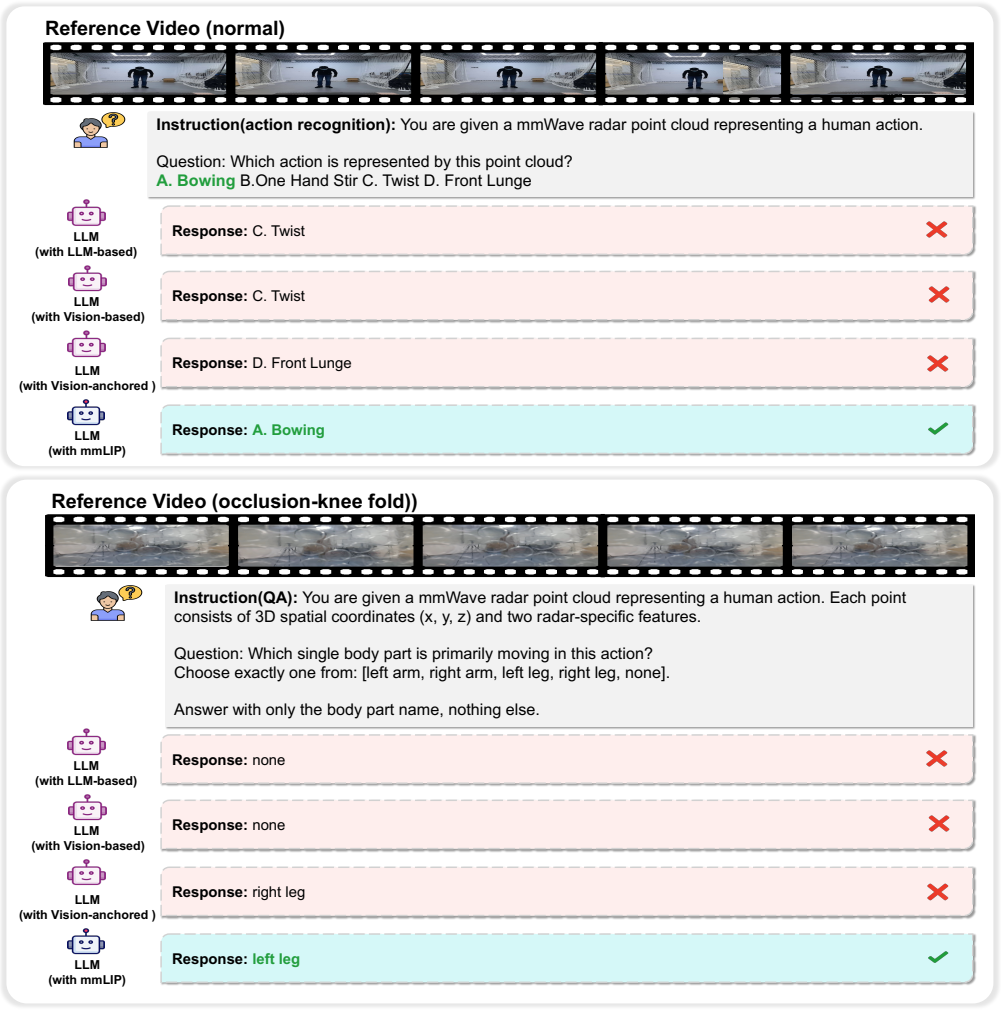

Figure 5: Qualitative comparison of model predictions across different conditions and encoder paradigms (projector-based, vision-based, vision-anchored, and mmLIP).

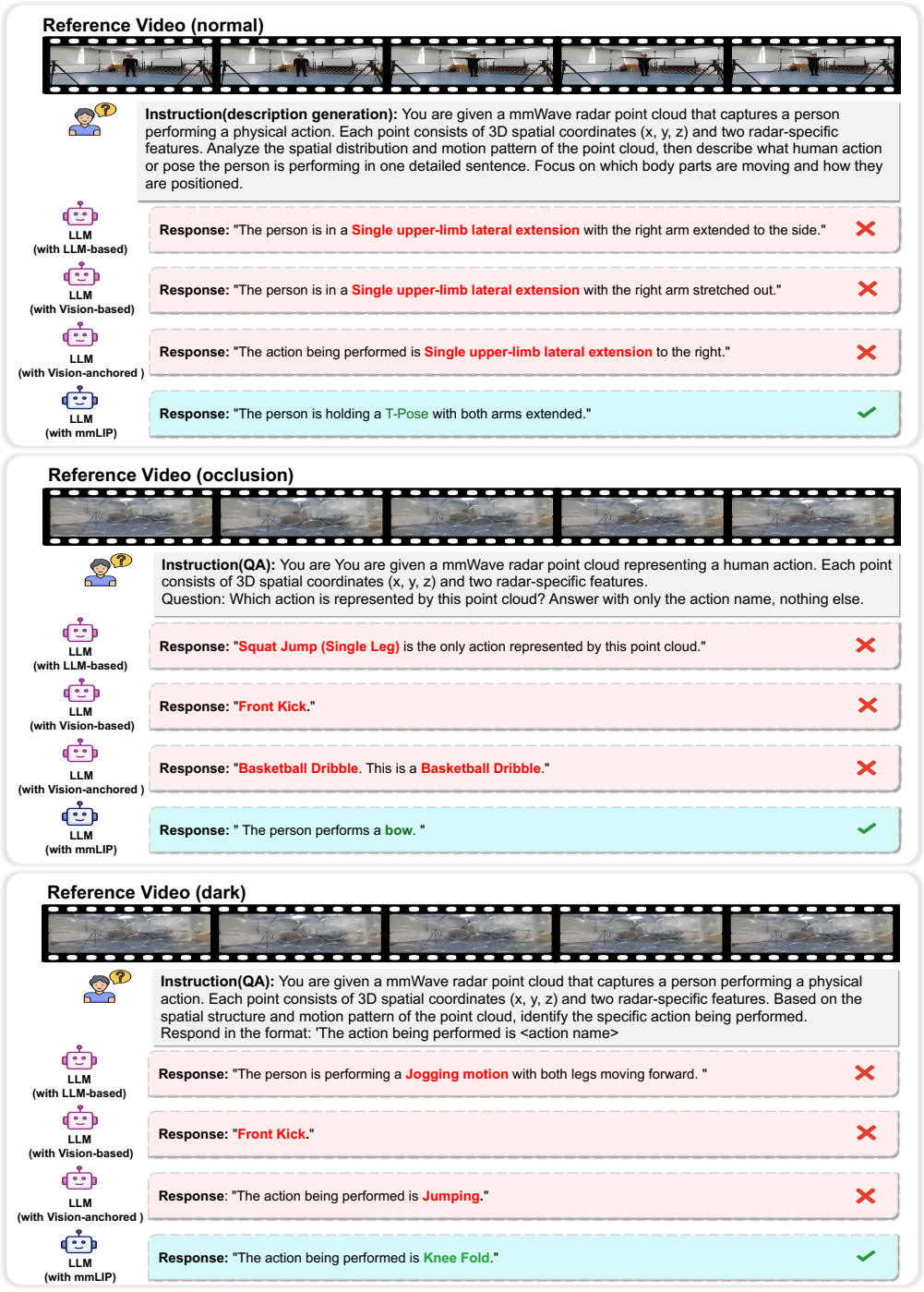

Figure 6: Qualitative comparison of model predictions across different conditions and encoder paradigms (projector-based, vision-based, vision-anchored, and mmLIP).

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: Section 4, Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: There is no theoretical result in our paper.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See section 4 and section B.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We will share a link to a github with the code used to run the experiments in the supplementary material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See section 5 and section B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The paper does not include error bars or statistical significance tests, as the experiments primarily follow standard single-run evaluation protocols commonly adopted in prior multimodal learning, retrieval, and text generation works.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See section D.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics. The work focuses on privacy-preserving mmWave radar sensing and multimodal representation learning using publicly available datasets, without involving sensitive personal information or harmful applications.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses the broader societal impacts of mmWave radar and multimodal reasoning systems. Positive impacts include privacy-preserving human understanding and robust perception under challenging conditions such as darkness or occlusion. Potential negative impacts include misuse for surveillance or unintended privacy concerns in human monitoring applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models or scraped datasets that require specific safeguards. The work primarily focuses on radar–language representation learning and evaluation using publicly available datasets.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper properly cites and credits all publicly available datasets, models, and prior works used in the experiments. We follow the original usage terms and licenses associated with these assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new datasets, models, or code assets.

    Guidelines:

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

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

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

    Answer:
    answerNA

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects requiring IRB approval.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The paper explicitly describes the use of LLMs as part of the core methodology, including radar–language alignment, text generation, question answering, and evaluation protocols.

    Guidelines:

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