CL4D: Contrastive Language–4D Pretraining for Vision-Language Reasoning in Dynamic Scenes

Kumal Hewagamage⋆[1], Isuranga Senavirathne⋆[1], Sasika Amarasinghe⋆[1], Hasitha Gallella⋆[1], Dulanga Weerakoon†[2], Vigneshwaran Subbaraju†[3,4], and Ranga Rodrigo†[1]

[1] University of Moratuwa, Sri Lanka {hewagamagekln.21, senavirathneiub.21, amarasingheywsp.21, gallellammhhb.21, ranga}@uom.lk
[2] Singapore-MIT Alliance for Research & Technology (SMART) Centre, Singapore dulanga.weerakoon@smart.mit.edu
[3] Agency for Science, Technology and Research (A*STAR), Singapore
[4] CNRS@CREATE, Singapore
Vigneshwaran_Subbaraju@a-star.edu.sg

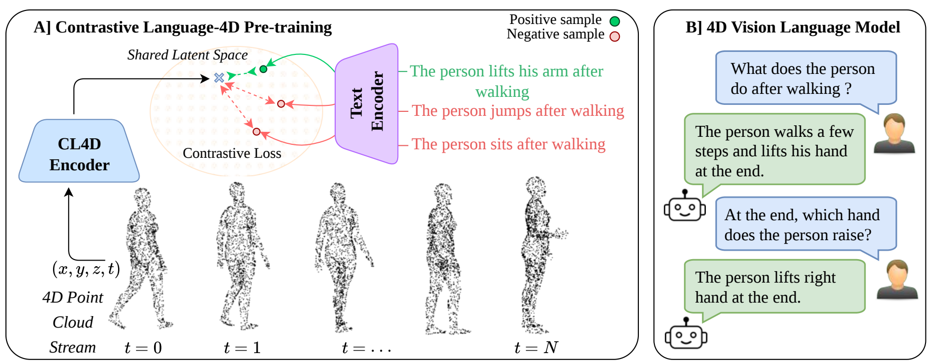


Fig. 1: Overview of the Language–4D framework: We introduce CL4D, a contrastive pre-training paradigm that bridges the gap between 4D sequences and natural language. (A) Contrastive Language-4D Pre-training aligns 4D sequences with text in a shared latent space via a multi-modal contrastive loss. (B) 4D Vision-Language Model which uses the learned representations for VLM downstream reasoning tasks.

Abstract. 4D understanding and reasoning is a fundamental capability for embodied AI agents operating in dynamic physical environments. However, existing vision encoders are largely limited to static 2D images or 3D point clouds without temporal modeling, or to 2D videos that lack accurate geometric depth reasoning. Consequently, current approaches fail to jointly capture spatial structure and motion evolution in dynamic scenes. We present CL4D, the first foundational 4D vision encoder that directly operates on dynamic point clouds, trained with a contrastive learning objective to align spatio-temporal geometric representations with natural language descriptions. By learning a shared

[1] *These authors contributed equally. †These authors jointly supervised this work.

embedding space between text and 4D scene dynamics, CL4D enables zero-shot motion-to-text and text-to-motion retrieval in dynamic environments and serves as a foundational 4D vision encoder for downstream 4D vision–language tasks. Building on this encoder, we introduce 4DVLM, a 4D vision–language model that conditions language generation on dynamic geometric representations. 4DVLM is the first VLM designed to operate directly on 4D point clouds without relying on 2D images, 2D videos, or static 3D point clouds. We train CL4D and subsequently 4DVLM on a newly constructed dataset termed DynAction4D capturing diverse human motions across varying object interactions and scene environments. Extensive experiments across multiple 4D human action benchmarks demonstrate that CL4D achieves state-of-the-art performance, with improvements of approximately ∼16.75% over prior methods. Furthermore, 4DVLM outperforms frontier video VLMs such as Gemini and GPT-5 even when these models are provided with RGB video sequences corresponding to the same scenes represented as 4D point clouds for 4DVLM.

Keywords: 4D Scene Understanding · Vision, language, and reasoning · Scene analysis and understanding

1 Introduction

The physical world is inherently four-dimensional, consisting of three spatial dimensions and one temporal dimension. Consequently, understanding and reasoning over dynamic 4D environments is a fundamental capability for embodied artificial intelligence (EAI). Models designed for such settings must develop a unified representation that captures both geometric structure and temporal dynamics within the spatio-temporal space of the physical world.

Existing approaches for physical world perception and reasoning rely predominantly on Vision Language Models (VLMs). However, most VLMs are designed for static 2D imagery, extracting visual features from single images, as in Llama [33], or extending to video-based representations, as in VideoLLM [5]. While video VLMs incorporate temporal signals, they remain fundamentally grounded in 2D pixel observations and lack explicit 3D geometric reasoning. More recent works, such as Ges3ViG [25] and PointLLM [37], extend multi-modal reasoning to 3D point cloud representations captured via LiDAR. Nevertheless, these methods are restricted to static 3D scenes and do not model temporal dynamics, limiting their applicability in dynamic physical environments.

Developing effective 4D reasoning models requires, as a foundational step, a robust 4D vision encoder. In 2D, 3D, and 2D with temporal modeling settings, contrastive pretraining [30] has emerged as the dominant paradigm for learning vision encoders aligned with language representations. Such encoders are trained to project visual and textual features into a shared embedding space, enabling strong transfer to downstream vision language tasks including Visual Question Answering (VQA) [3], Visual Grounding (VG) [8], and image or video captioning [1]. Despite the success of contrastive alignment in these domains, no prior

[2] Dataset and codes available at `https://4d-vision-uom.github.io/`.

work has extended this paradigm to 4D representations that capture dynamic 3D scenes. In particular, existing approaches do not learn language aligned embeddings from temporally evolving point clouds that jointly encode both geometry and motion. To address this gap, we introduce ***C***ontrastive ***L***anguage-***4D*** (***CL4D***), the first foundational 4D vision encoder. CL4D is trained using a contrastive learning objective to align dynamic 4D point cloud sequences representing human actions with their corresponding natural language descriptions. By learning a shared embedding space between spatio-temporal geometric features and action semantics, CL4D enables zero-shot action-to-text retrieval and establishes a foundation for language-grounded reasoning over dynamic 4D environments.

Building upon the proposed CL4D 4D vision encoder, we further introduce ***4D V***ision ***L***anguage ***M***odel (***4DVLM***), the first VLM designed to directly reason over dynamic 4D point clouds. Unlike prior VLMs that operate on 2D images, videos, or static 3D scenes, 4DVLM is grounded in temporally evolving 3D scenes, enabling joint spatial and temporal reasoning in dynamic physical environments.

In summary, we make the following **key contributions**:

1. **CL4D: A Contrastively Aligned 4D Vision Encoder.** We introduce CL4D, the first foundational 4D vision encoder trained to align dynamic 4D point cloud sequences with natural language using a contrastive learning objective. Unlike prior 4D encoders that rely on classification over a fixed set of action labels, CL4D learns a shared embedding space between spatio-temporal scene dynamics and textual descriptions. This enables zero-shot cross-modal retrieval between motion and text in dynamic environments and establishes a foundational representation for 4D vision–language tasks. Empirically, CL4D improves Recall@1 for text–motion retrieval by up to **16.75%** compared to prior 4D encoders.

2. **4DVLM: The First 4D Vision–Language Model.** Building upon CL4D, we propose 4DVLM, the first Vision–Language Model that directly reasons over temporally evolving 4D point clouds without relying on RGB images, videos, or static 3D scenes. By conditioning language generation on geometry-aware spatio-temporal embeddings, 4DVLM enables language-grounded reasoning in dynamic physical environments. Remarkably, 4DVLM achieves **state-of-the-art performance even compared to frontier video-based VLMs such as Gemini and GPT-5**, despite those models operating on RGB video inputs.

3. **DynAction4D: A Benchmark for Language–4D Alignment.** To enable systematic evaluation and large-scale contrastive training of 4D vision models, we introduce DynAction4D, a benchmark dataset consisting of dynamic human actions represented as temporally evolving 4D point cloud sequences paired with natural language descriptions. The dataset spans diverse action categories and scene environments with varying object layouts, providing a structured benchmark for studying language–4D alignment and evaluating 4D VLMs.

2 Related Work

Contrastive Pretraining for Vision–Language Alignment: Early vision encoders were largely developed using supervised learning on large-scale datasets such as ImageNet [7], leading to architectures including AlexNet [18], VGGNet [32], and ResNet [15]. These models were trained using cross-entropy classification objectives and later reused as general visual feature extractors for downstream tasks. However, such representations are tied to predefined label spaces and are not inherently aligned with language.

Contrastive pretraining addressed this limitation by aligning visual and textual representations in a shared embedding space. CLIP [30] introduced large-scale language–image contrastive learning, enabling strong zero-shot transfer across tasks. Subsequent works extended this paradigm to video and 3D representations. For videos, methods such as VideoCLIP [36], CLIP4Clip [24], and ActionCLIP [35] learn joint video–text embeddings by modeling temporal dynamics in RGB frame sequences. In the 3D domain, approaches including PointCLIP [41], ULIP [38], and OpenShape [22] align static point clouds with language through multi-view projections or joint multimodal pretraining.

Despite these advances, existing methods largely operate on static images, RGB video frames, or static 3D geometry, and do not explicitly model temporally evolving 3D environments.

4D Vision Encoders: Modeling dynamic point cloud sequences introduces additional challenges due to the irregular and unordered nature of point clouds across time. Early approaches focused on extracting spatio-temporal features for action recognition rather than language alignment. For example, Motion PointNet [16] models temporal variations using a PointNet-style architecture with temporal aggregation, optimized using cross-entropy objectives for closed-set action classification.

More recent works attempt to bridge motion understanding and language by learning cross-modal representations. Motion Patches [39] converts human motion sequences into grid-like patches compatible with Vision Transformers pretrained on images, enabling contrastive alignment between motion and language representations. While these approaches demonstrate promising zero-shot capabilities, they rely on structured skeleton representations rather than raw dynamic point clouds.

Consequently, existing methods either rely on classification-based objectives or structured motion representations, limiting their applicability for open-vocabulary reasoning over raw dynamic 4D environments.

Vision Language Models: Large vision–language models have recently achieved strong performance on multimodal reasoning tasks such as visual question answering, visual grounding, and captioning. Systems such as GPT-5 [26], Gemini [2], and LLaVA [21] integrate visual encoders with large language models to enable reasoning over images and videos. Dedicated video VLMs such as VideoLLM [5] and Video-LLaVA [19] further explore spatio-temporal reasoning using RGB video inputs.

Recent works have also explored grounding language models in 3D scenes. Approaches such as M3DRefCLIP [43] and Ges3ViG [25] study 3D visual grounding, while PointLLM [37], LL3DA [6], and OneLLM [14] extend instruction tuning to static point clouds.

However, existing methods operate either on 2D RGB data or static 3D geometry. None directly support language-based reasoning over dynamic 4D point cloud sequences. In contrast, 4DVLM enables vision–language reasoning directly on temporally evolving 4D point clouds, allowing joint understanding of geometry and motion dynamics.

3 Methodology

Our framework is designed to enable language-grounded reasoning over dynamic 4D point cloud sequences. We adopt a two-stage training strategy Fig. 4. In the first stage, we learn semantically aligned spatio-temporal representations through contrastive Language–4D pretraining (CL4D). In the second stage, we rely on the pretrained 4D encoder to construct a 4D Vision–Language Model (4DVLM) that supports instruction following and visual question answering over dynamic point clouds. To facilitate this two-stage learning process, we introduce the DynAction4D dataset.

3.1 DynAction4D Dataset

We first present our proposed benchmark dataset for Language–4D pretraining aimed at enabling vision–language reasoning in dynamic physical environments. The proposed dataset primarily consists of dynamic 4D point cloud sequences capturing various human actions across diverse scene environments, along with their corresponding textual descriptions. DynAction4D comprises four primary segments, each representing distinct dynamic scenarios. We now describe each dataset segment of the DynAction4D dataset.

1. HumanOnly Segment As shown in Figure 3a, HumanOnly segment, considers a variety of actions performed by a single human, derived from the HumanML3D dataset [13]. HumanML3D is a human motion–language dataset that covers a broad range of activities, including daily actions (e.g., walking, jumping), sports (e.g., swimming, playing golf), acrobatics (e.g., cartwheel), and artistic movements (e.g., dancing). This dataset provides SMPL parameters for each motion sequence along with corresponding natural language descriptions that explain the actions. The training split for HumanOnly contains 23k sequences, while the test split consists of 4k sequences.

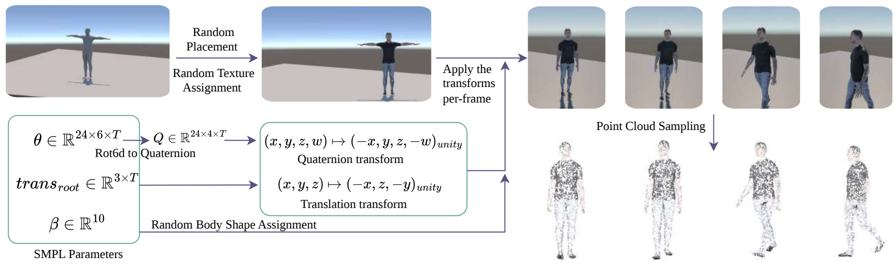
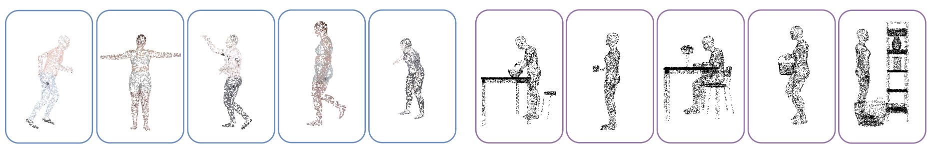


Fig. 2: Overview of the DynAction4D generation pipeline. SMPL parameters are first converted to Unity-compatible animations, then augmented using various methods such as random body shape assignment and random texture variations from [34]. The resulting meshes are then uniformly sampled in Unity to generate human action point cloud sequences. This pipeline yields highly diverse, temporally evolving point cloud sequences suitable for training complex dynamic scene understanding models.



(a) DynAction4D-HumanOnly **(b) DynAction4D-ObjInteractions**

(c) DynAction4D-Cluttered

Fig. 3: Sample frames from point cloud sequences across the three dataset segments in DynAction4D. (a) HumanOnly: Five sequences, each depicting a distinct human body shape and action. **(b) ObjInteractions:** Five sequences, each featuring a human interacting with a different object. **(c) Cluttered:** Three sequences showing humans performing actions in cluttered environments.

2. ObjInteractions Segment: This segment considers human actions that involve interactions with objects. To support this, we construct 4D scenes using human–object interaction sequences from the Humoto dataset [23], which models diverse daily activities involving 72 objects comprising 735 distinct object interactions.The training split for ObjInteractions contains 510 sequences, while the test split consists of 219 sequences.

3. Cluttered Segment: This Cluttered segment considers a scenario where a human performs various actions in cluttered scene environments. To construct such a scenario, we place human action sequences taken from the HumanML3D dataset within scenes populated with Humoto objects at varying placements and clutter conditions. The training split for the Cluttered segment contains 23k sequences, while the test split consists of 4k sequences.

Textual Data Generation Text descriptions are inherited from the original action datasets, where each motion sequence is paired with a natural language annotation. These text–motion pairs are used for language-aligned contrastive pre-training.

4D-VQA Segment: For training the proposed 4DVLM, we create a novel 4D-VQA dataset. For this we employ videos rendered in Unity from existing human action sequences generated from the HumanML3D dataset. To synthesize this dataset segment, we prompt Gemini 3.0 Flash [2] with rendered Unity videos and HumanML3D text descriptions, generating structured queries designed to test spatial and temporal reasoning. The QA pairs span three distinct motion-analysis categories: (a) **Action:** Focuses on high-level semantic understanding and sequence classification. (b) **Body-Spatial Qualities:** Focuses on 3D spatial reasoning, geometric relationships, and limb positioning. (c) **Temporal Qualities:** Focuses on the dynamics and frame-to-frame state changes of the motion.

3.2 CL4D: Contrastive Language-4D Pre-training

In the first stage of training, we introduce a novel Spatio-temporal 4D Vision Encoder as illustrated in Fig. 4(A). This encoder learn semantically aligned spatio-temporal representations via contrastive language-4D pre-training (CL4D). Our objective is to learn language-aligned representations for dynamic point cloud sequences (x, y, z, t) through large-scale contrastive pre-training.

Point Encoder (PE): Consider a 4D sequence with N point cloud frames. Each frame of the input sequence is first processed by a standard PointNet [29] encoder PE that operates on local point groups of each frame. We divide each frame into k number of groups, each containing g number of points. The encoder produces k embeddings of dimension d_1 for each frame $p_t \in X_v$ point cloud stream. We prepend a learnable CLS token, resulting in $(k+1)$ embeddings per frame that form the feature tensor $f_t \in \mathbb{R}^{(k+1)\times d_1}$ for the input to the spatial encoder V_s, as defined in Eq. (1).

Spatio-Temporal Transformer Encoder (V_{st}): To jointly model the spatial structure and temporal dynamics of these f_t features from PE, we introduce a transformer based encoder defined as V_{st}, which is a combination of the spatial encoder V_s and the temporal encoder V_t. The processing order of this module is defined in Eqs. (2)–(4).

$$f_t = \mathrm{PE}(p_t) \in \mathbb{R}^{(k+1)\times d_1}, \quad p_t = X_v[t], \quad t = 1, \dots, N \tag{1}$$

$$\tilde{h}_t = \{\mathrm{MAX}(h_t), \mathrm{CLS}(h_t), \mathrm{MEAN}(h_t)\} \in \mathbb{R}^{3\times d_2}, \quad h_t = V_s(f_t) \in \mathbb{R}^{d_2} \tag{2}$$

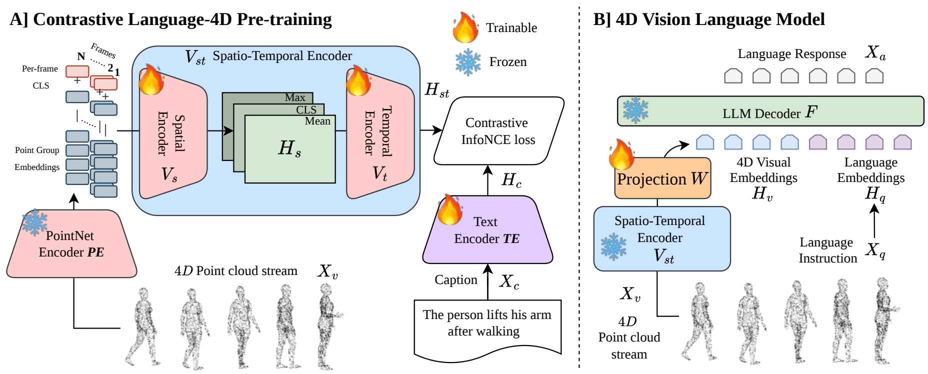


Fig. 4: CL4D framework overview: (A) **Contrastive Language-4D Pre-training** (Sec. 3.1): A dynamic point cloud sequence (x, y, z, t) is encoded by the PointNet-based PE and a spatio-temporal encoder $V_{st}(V_s, V_t)$ to obtain 4D visual embeddings, which are aligned with text features from TE via a contrastive objective. (B) **4D Visual Question Answering** (Sec. 3.2): The learned V_{st} representations are fused with language embeddings in a LLaVA setup to enable reasoning over dynamic point clouds and generate action-aware responses.

$$H_s = \{\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_N\} \in \mathbb{R}^{3\times d_2 \times N} \quad for \quad t = 1, \ldots, N \tag{3}$$
$$H_{st} = V_t(H_s) \in \mathbb{R}^{d_3} \tag{4}$$

The V_s captures global spatial relationships extracted to f_t for each frame by PE and V_t models the inter-frame temporal dependencies across all N frames. The embeddings inside V_{st} transformer blocks are d_3 dimensional.

The input f_t tensors will go through V_s individually and produce the h_t sequence of embeddings. From each h_t we compute means, max, and cls embeddings and stack up for a $\tilde{h}_t$ per each frame as in Eq. (2). After processing all the N frames, we collect all the $\tilde{h}_t$ in to a single $3 \times d_2 \times N$ dimensional 3D tensor H_s as defined in Eq. (3). Then the H_s 3D tensor will be patched up to $m \times m$ patches to follow similar to a standard vision transformer. It will create a new $(d_2 \times N)/(m \times m)$ length patch sequence. These patches will be encoded again into the d_3 dimensional embeddings through a linear projection and fed to the V_t transformer blocks to model temporal relationships. At the end of V_t we get the mean embedding as H_{st} of the output embedding sequence of the transformer (Defined in Eq. (4)).

In our best setup, we used embedding dimensions $d_1 = 512$, $d_2 = 512$ and $d_3 = 768$. Point Encoder PE produces with $k = 64$ embeddings per frame with $g = 32$ group size for the DynAction4D dataset. Both V_s and V_t transformers have 12 Attention Heads and MLP Ratio is 4 while transformer depth is $V_s = 4$ and $V_t = 12$ for each. V_s is initialized from scratch and temporal encoder V_t is initialized from $ViT - B/16$ weights prior to training, enabling stable optimization and improved motion modeling. V_t used with patch size $m = 16$ to process the $H_s \in \mathbb{R}^{3\times 512\times 32}$ which created 32×2 patches.

Text Encoder (TE): Language descriptions (X_c) are encoded using $distilbert-base-uncased$ [31]. We use the CLS token representation (H_c) as the global text embedding which has d_3 dimensions.

Contrastive Alignment Objective: Let $\{v_i\}_{i=1}^{B}$ denote visual embeddings (H_{st}) obtained by mean pooling the output token sequences of V$_{st}$ for the batch size B, and $\{t_i\}_{i=1}^{B}$ denote the corresponding CLS embeddings (H_c) from the text encoder. We optimize a symmetric cross-entropy objective over the $B \times B$ similarity matrix using cosine similarity $s(\cdot, \cdot)$ with temperature τ:

$$\mathcal{L}_{m2t} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s(v_i, t_i)/\tau)}{\sum_{j=1}^{B}\exp(s(v_i, t_j)/\tau)}, \tag{5}$$

$$\mathcal{L}_{t2m} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s(v_i, t_i)/\tau)}{\sum_{j=1}^{B}\exp(s(v_j, t_i)/\tau)}, \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{m2t} + \mathcal{L}_{t2m}. \tag{7}$$

This objective encourages aligned visual-text pairs to have high similarity while pushing apart mismatched pairs within the batch B. This alignment enables semantically grounded 4D representations that jointly encode geometric structure and motion dynamics. Furthermore, CL4D can serve as a general 4D vision backbone for a range of 4D vision-language reasoning tasks, where language and dynamic point clouds are projected into a unified embedding space.

Training Details: Using the proposed contrastive alignment objective, CL4D is trained for 150 epochs with a batch size of $B = 72$. Model parameters are optimized using AdamW optimizer with decoupled learning rates: 1.0×10^{-4} for the V_{st} components and projection heads, and 1.0×10^{-5} for the text encoder.

3.3 4DVLM: 4D Vision Language Model

Building upon the contrastive pre-trained 4D vision encoder, we construct a 4D Vision-Language Model (Fig. 4(B)) capable of performing vision–language reasoning directly on 4D point clouds.

Visual Token Projection: We first take the spatio-temporal embeddings produced by V_{st}, and linearly project them to a 4096-dimensional space to match the hidden dimension of LLaMA.

Language Backbone: We initialize the language model from Llava-v1.5-7b, which is based on the Vicuna 7B backbone. The projected visual tokens are injected into the language model following the standard LLaVA-style multi-modal conditioning mechanism [21].

Training Objective: The model is optimized using the standard auto-regressive language modeling loss employed in LLaVA. Given a visual sequence and a question X_q, the model generates an answer conditioned on projected 4D tokens:

$$\mathcal{L}_{\text{VLM}} = -\sum_t \log p(y_t \mid y_{<t}, X_q, \text{Proj}(\text{V}_{st})). \tag{8}$$

Training Setup: With the proposed training objective, we freeze the language encoder and the CL4D spatio-temporal 4D encoder, and train only the projection layer for 10 epochs using the VQA split of DynAction4D . Training is performed with a learning rate of 1.0×10^{-3} and the AdamW optimizer.

4 Results

4.1 Evaluation of CL4D

Table 1: Motion-text retrieval results on DynAction4D segments and RH20T dataset. The results show the Recall@1 for both Batch and Global settings under text-to-motion and motion-to-text retrieval using training batch size = 72 and test batch size = 32. CL4D and CL4D-mini outperforms the other 4D encoders in all the DynAction4D segments and RH20T dataset.

Dataset	Method	Text-motion retrieval		Motion-text retrieval	
		R@1↑ (Batch)	R@1↑ (Global)	R@1↑ (Batch)	R@1↑ (Global)
DynAction4D HumanOnly	P4Transformer [9]	53.57	3.66	58.05	5.33
	PST-Transformer [10]	49.09	2.69	51.73	3.34
	Motion PointNet [16]	51.91	2.34	55.78	3.13
	CL4D (Ours)	**70.32**	**8.07**	**68.62**	**8.02**
DynAction4D ObjInteractions	P4Transformer [9]	26.55	5.94	24.93	7.31
	PST-Transformer [10]	30.37	8.22	25.50	7.31
	Motion PointNet [16]	41.37	12.33	36.18	11.87
	CL4D (Ours)	**49.40**	**23.29**	**46.97**	**22.37**
DynAction4D Cluttered	PST-Transformer [10]	30.86	0.90	33.29	0.93
	Motion PointNet [16]	41.71	1.23	43.24	1.53
	CL4D (Ours)	**55.07**	**3.11**	**51.94**	**2.71**
RH20T [11]	P4Transformer [9]	9.51	0.84	5.77	0.84
	PST-Transformer [10]	54.76	20.17	39.61	15.13
	Motion PointNet [16]	68.21	31.09	52.28	24.37
	CL4D Mini (Ours)	**77.24**	**37.82**	**57.10**	**27.73**

Evaluation Setup: We evaluate CL4D on two primary retrieval tasks. (a) *Text-to-motion retrieval*, where a given text embedding is matched against a batch of 4D scene embeddings produced by CL4D , with only one embedding corresponding to the correct match. (b) *Motion-to-text retrieval*, where a given

4D scene embedding obtained from CL4D is matched against a batch of textual embeddings, with only one text embedding corresponding to the correct scene.

We adopt Recall@k (R@k) [28] as the primary evaluation metric. In the *R@k (Batch)* setting, retrieval is performed within a randomly sampled batch of embeddings. In contrast, *R@k (Global)* evaluates retrieval across the entire test dataset, where the query must retrieve the correct match from all available embeddings. DynAction4D is the primary benchmark for evaluation of CL4D.

Comparison with Prior 4D Encoders: CL4D represents the first 4D encoder trained with a contrastive learning objective that aligns scene and text embeddings within a unified representation space. In contrast, prior 4D encoders are typically trained using a standard cross-entropy objective to classify motion embeddings into a predefined and finite set of action labels. As a result, such encoders are not naturally suited for cross-modal retrieval tasks such as text-to-motion or motion-to-text retrieval when given a 4D point cloud as input.

To evaluate the effectiveness of CL4D against prior 4D encoders, we re-adapt these methods using the same contrastive alignment objective and train them on the proposed DynAction4D dataset. The comparison results are summarized in Table 1. Notably, CL4D achieves improvements of 16.75%, 8.1%, and 13.36% in R@1 under the batch setting for text-to-motion retrieval on the HumanOnly, ObjInteractions, and Cluttered segments of DynAction4D, respectively.

These results demonstrate that CL4D enables stronger zero-shot alignment between textual descriptions and dynamic 4D scene embeddings without relying on a predetermined and finite set of action labels, as required by prior classification-based approaches.

Generalizability of CL4D : To further demonstrate the generalizability of CL4D to real-world point cloud data involving actions performed by non-human agents, we evaluate CL4D on the robotic manipulator action dataset RH20T [11] in Table 1. RH20T comprises of manipulation sequences across diverse skills, contexts, robots, and camera viewpoints, all collected in the real world. CL4D-mini achieves state-of-the-art performance, surpassing all baseline methods by at least 9.03% in the batch setting for text-to-motion retrieval and 4.82% for motion-to-text retrieval for tested splits.

Ablation & Sensitivity Studies: In Table 2, we evaluate several architectural variations of CL4D . In particular, we vary the ViT backbone used in the temporal encoder of CL4D among the Tiny, Base, Small, and Large variants. Among these configurations, CL4D with the ViT-Small backbone achieves the highest recall accuracy for cross-modal retrieval under the batch evaluation setting.

When the text encoder is frozen (*w. Frozen Text Enc.*), retrieval accuracy decreases across all metrics, highlighting the importance of jointly optimizing both the 4D scene and textual embeddings for stronger contrastive alignment in the shared embedding space. When SigLIP loss is used (*w. SigLIP loss*) instead of the standard InfoNCE loss, performance across all metrics also decreases.

Across all ViT variants, we initialize the temporal encoder with pretrained ViT weights. In the *w. Untrained ViT* setting, we instead use an untrained ViT-Base model for the temporal encoder. This configuration leads to lower accuracy, indicating the importance of prior 2D-learned ViT features for effective temporal modeling. The CL4D-mini, a parameter-efficient version of CL4D yields only a marginal drop in Recall@1 while achieving an approximate 60% reduction in GFLOPs compared to CL4D with ViT-Tiny. CL4D-mini employs a reduced spatial encoder along with ViT-Tiny. Finally, in the *w. Shuffled Frames* setting, we randomly shuffle the order of the input point cloud frames to evaluate whether CL4D relies on the correct temporal ordering of motion sequences. In this case, we observe a significant drop of 11.44% in motion-to-text R@1 under the batch evaluation setting.

Furthermore, to evaluate the robustness of CL4D under real-world point cloud degradations prevalent in LiDAR sensors, we tested against partially captured point clouds obtained via single-viewpoint back-face culling [17] and under random reconstruction errors (normalised std.=0.01) in Table 2. CL4D-mini remains robust under both conditions, losing only a modest 0.37% and 1.33% in text-to-motion R@1 under the batch evaluation setting for partial point clouds and random reconstruction errors, respectively.

In Figure 5, we vary the training batch size used for contrastive pretraining from $B = 16$ to $B = 72$. Across all configurations of CL4D , we observe a consistent improvement in R@1 as the batch size increases. This trend can be attributed to the larger number of negative samples available in each batch, which provides stronger contrastive supervision and leads to improved alignment between textual and 4D scene embeddings in the shared representation space.

In Figure 6, we vary the number of point cloud frames used as input to CL4D . For both motion-to-text and text-to-motion retrieval tasks, we observe a consistent increase in R@1 as the number of frames increases. This improvement can be attributed to the richer temporal information captured with a higher number of frames, indicating that CL4D is able to effectively model temporal dynamics for improved cross-modal retrieval performance.

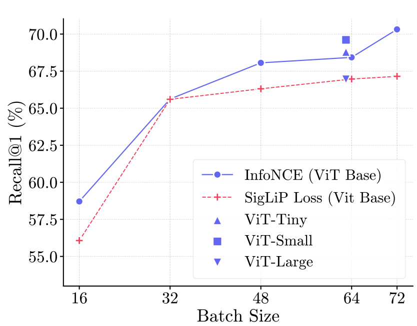


Fig. 5: Training batch size vs. Batchwise Recall@1 on text-to-motion retrieval, with results referenced in Table 2. Eval. batch size = 32.

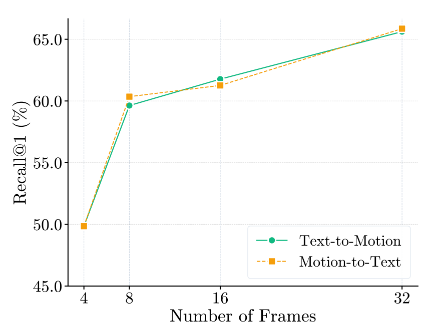


Fig. 6: Effect of num. frames for text-to-motion and motion-to-text retrieval evaluated using Batchwise Recall@1. Eval. Batch size = 32.

Table 2: Motion-text Retrieval results on the DynAction4D-HumanOnly dataset comparing our model (CL4D) and other experimental setups. The results show Recall@1 (R@1 %) for both Batch (eval. batch size 32) and Global settings under text-to-motion and motion-to-text retrieval with training batch size = 64. For other variations, we study the effect of freezing the text encoder, using a randomly initialized ViT encoder, shuffling the frames in the point cloud sequence, and optimizing the spatial and temporal encoder, with ViT-Base used as the baseline for all experiments. Under P.Cloud variations, we evaluate CL4D-mini under real-world point cloud degradations prevalent in LiDAR sensors.

	Model Variation	Text to Motion		Motion to Text		FLOPs↓ (10^9)
		R@1↑ (Batch)	R@1↑ (Global)	R@1↑ (Batch)	R@1↑ (Global)	
ViT Variants	w. ViT-Tiny	68.79	**5.15**	68.09	4.24	176.20
	w. ViT-Small	**69.61**	4.80	**70.41**	4.59	177.64
	w. ViT-Base	68.43	4.45	67.96	4.68	183.28
	w. ViT-Large	66.95	4.66	66.31	4.68	202.42
Other Model Variations	w. Frozen Text Enc.	56.63	2.76	57.39	2.78	183.28
	w. SigLIP Loss	66.97	4.03	67.14	4.20	183.28
	w. untrained ViT	63.32	3.52	62.79	3.36	183.28
	w. Shuffled frames	60.97	3.08	58.97	2.97	183.28
	CL4D-mini	67.18	4.24	66.02	**4.78**	**70.22**
P.Cloud Variations (CL4D-mini)	w. Partial P.Cloud	66.81	4.33	65.66	4.52	70.22
	w. random recon. error	65.85	3.69	64.64	3.82	70.22

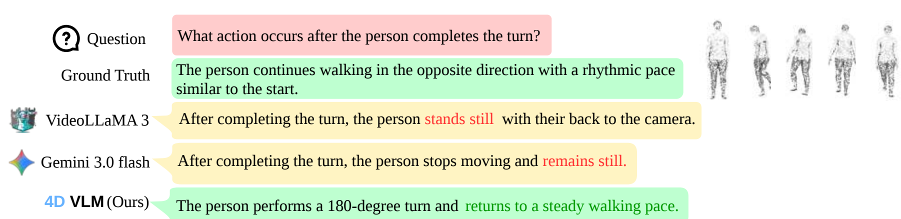


Fig. 7: Generated outputs of our 4DVLM on sample VQA instances from DynAction4D-VQA segment, depicting our model's ability to answer questions about human action sequences with significantly higher accuracy and contextual awareness than existing video-based models.

4.2 Evaluation of 4DVLM

Evaluation Setup: We evaluate the performance of 4DVLM with the proposed benchmark dataset DynAction4D's VQA segment. We primarily use standard VLM benchmark metrics; BLEU [27], ROUGE [20], METEOR [4], BERTScore [42] and SimCSE [12] on VQA with a 4D point cloud as an input.

Table 3: Visual Question Answering (VQA) results on DynAction4D-VQA, comparing our 4DVLM with existing video-based VLMs. Since video models cannot directly process point cloud sequences, we render the point clouds as mesh videos and feed them into the respective video models for a fair comparison. Our model outperforms all video-based baselines across the evaluated metrics.

Method	BLEU↑	ROUGE			METEOR↑	BERTScore			SimCSE↑
		ROUGE-1(F1)↑	ROUGE-2(F1)↑	ROUGE-L(F1)↑		Precision↑	Recall↑	F1↑	
VideoLLaMA 3 [40]	0.0437	0.3382	0.1174	0.2951	0.2481	**0.4769**	0.3870	0.4317	0.8158
Gemini-3.0 Flash [2]	0.0447	0.3389	0.1156	0.2812	0.2673	0.4240	0.3875	0.4057	0.8009
Gemini 3.1 Pro [2]	0.0300	0.2763	0.0923	0.2427	0.1933	0.4562	0.2938	0.3736	0.7690
Gemini 3.1 Pro [2](M.V)	0.0320	0.2885	0.0955	0.2500	0.2064	0.4482	0.3115	0.3788	0.7760
GPT 5 [26]	0.0184	0.1935	0.0403	0.1612	0.1347	0.4111	0.2309	0.3191	0.7432
4DVLM (Ours)	**0.0729**	**0.3857**	**0.1563**	**0.3324**	**0.3152**	0.4515	**0.4396**	**0.4459**	**0.8189**

Comparison with the State-of-the-art Video VLMs In Table 3, we compare 4DVLM against several frontier VLM models. Notably, none of the prior methods are capable of directly consuming 4D point clouds for vision–language reasoning. Therefore, for all baseline VLMs we render the point clouds as mesh videos and feed them into the respective video models for a fair comparison. To provide video baselines with some 3D geometric understanding, we additionally compare against Gemini 3.1 Pro provided with three-viewpoint renders of the same scene, denoted as Gemini 3.1 Pro (M.V) in Table 3.

Overall, our approach consistently outperforms these frontier VLM baselines across lexical, semantic, and embedding-based evaluation metrics. These results suggest that geometry-aware 4D representations provide stronger grounding for action-centric language generation compared to RGB-based video representations. In Figure 7, we showcase an example output from 4DVLM, when compared against frontier video-based VLMs.

Table 4: Visual Question Answering (VQA) results on DynAction4D-VQA, comparing the 4D encoder in our 4DVLM against baseline 4D encoders. We replace the CL4D encoder in 4DVLM with alternative 4D encoders while keeping the projection layer the same across all experimental setups. Our method shows consistent gains across all metrics, demonstrating the effectiveness of our pretraining approach.

Method	BLEU↑	ROUGE			METEOR↑	BERTScore			SimCSE↑
		ROUGE-1(F1)↑	ROUGE-2(F1) ↑	ROUGE-L(F1)↑		Precision↑	Recall↑	F1↑	
w. PST-Transformer	0.0394	0.2910	0.0938	0.2554	0.2272	0.3808	0.3300	0.3558	0.7551
w. Motion PointNet	0.0320	0.2932	0.0897	0.2417	0.2513	0.3378	0.3594	0.3488	0.7738
w. CL4D (Ours)	**0.0729**	**0.3857**	**0.1563**	**0.3324**	**0.3152**	**0.4515**	**0.4396**	**0.4459**	**0.8189**

Comparison with Prior 4D Encoders: Now, we compare the performance of 4DVLM when using prior 4D encoders to generate 4D scene tokens instead of the proposed CL4D , as shown in Table 4. Although these prior encoders were originally trained using a standard cross-entropy objective, we retrain them with the same contrastive alignment objective for a fair comparison. Even under this setting, 4DVLM with CL4D consistently achieves higher performance across all evaluation metrics. Among the evaluated 4D encoders, PST-Transformer attains the closest performance to CL4D . Nevertheless, 4DVLM equipped with CL4D achieves an 85.03% higher BLEU score compared to PST-Transformer.

5 Discussion

Rationale for Choosing 4D Pointclouds for Motion Representation: Prior work on human motion understanding has primarily relied on skeleton-based representations, which, while effective for human-only motion, cannot generalise to arbitrary objects or non-human agents. In our ObjInteractions segment, there is no well-defined skeletal representation for the 73 interacting object categories, making skeleton-based methods fundamentally inapplicable. Raw 4D point clouds instead provide a unified representation that naturally encodes the geometry and motion of any dynamic entity, as further evidenced by our results on robotic manipulation sequences. With CL4D, we aim for a general 4D encoder for motion understanding across both human and non-human agents, as demonstrated by the DynAction4D-ObjInteractions segment and RH20T dataset, with broader generalisation remaining an important direction for future work.

6 Conclusion

We present CL4D, a foundational 4D vision encoder that directly operates on temporally evolving 3D point clouds to generate spatio-temporal vision tokens capturing both geometric structure and motion dynamics. Unlike prior 4D encoders that rely on cross-entropy classification over a fixed set of action labels, CL4D employs a contrastive learning objective that aligns 4D scene representations with their corresponding language descriptions in a shared embedding space. This design enables semantically grounded 4D representations and significantly improves cross-modal retrieval performance. In particular, CL4D achieves improvements of 8.1–16.75% in batch Recall@1 for the text-to-motion retrieval task compared to prior 4D encoders. Building on this encoder, we further introduce 4DVLM, the first vision–language model designed to directly reason over dynamic 4D point clouds. By conditioning language generation on geometry-aware spatio-temporal features, 4DVLM enables language-grounded understanding of dynamic scenes. Experiments demonstrate that 4DVLM achieves superior performance compared to frontier video-based VLMs, including GPT-5 and Gemini, even when these models are provided with corresponding RGB video.

7 Acknowledgments

A part of the computational resources were from the funding provided by Manoj Bandara. Also the research was supported by the Singapore National Research Foundation (NRF), Prime Minister's Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore-MIT Alliance for Research and Technology (SMART) centre.

References

1. Abdar, M., Kollati, M., Kuraparthi, S., Pourpanah, F., McDuff, D., Ghavamzadeh, M., Yan, S., Mohamed, A., Khosravi, A., Cambria, E., et al.: A review of deep learning for video captioning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
2. Anil, R., Borgeaud, S., Alayrac, J.B., et al.: Gemini: A family of highly capable multimodal models (2025), https://arxiv.org/abs/2312.11805
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
4. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Goldstein, J., Lavie, A., Lin, C.Y., Voss, C. (eds.) Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan (Jun 2005), https://aclanthology.org/W05-0909/
5. Chen, G., Zheng, Y.D., Wang, J., Xu, J., Huang, Y., Pan, J., Wang, Y., Wang, Y., Qiao, Y., Lu, T., et al.: Videollm: Modeling video sequence with large language models. arXiv preprint arXiv:2305.13292 (2023)
6. Chen, S., Chen, X., Zhang, C., Li, M., Yu, G., Fei, H., Zhu, H., Fan, J., Chen, T.: Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26428–26438 (2024)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1769–1779 (2021)
9. Fan, H., Yang, Y., Kankanhalli, M.: Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14204–14213 (2021)
10. Fan, H., Yang, Y., Kankanhalli, M.: Point spatio-temporal transformer networks for point cloud video modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 2181–2192 (2022)
11. Fang, H.S., et al.: Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In: ICRA (2024)
12. Gao, T., Yao, X., Chen, D.: SimCSE: Simple contrastive learning of sentence embeddings. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6894–6910. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.552, https://aclanthology.org/2021.emnlp-main.552/
13. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5152–5161 (June 2022)
14. Han, J., Gong, K., Zhang, Y., Wang, J., Zhang, K., Lin, D., Qiao, Y., Gao, P., Yue, X.: Onellm: One framework to align all modalities with language. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26584–26595 (2024)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Huang, Z., Fan, Z., Xu, T., Han, J.: On exploring pde modeling for point cloud video representation learning. arXiv preprint arXiv:2404.04720 (2024)
17. Jang, H., Kim, M., Bae, J., Kim, Y.M.: Dynamic mesh recovery from partial point cloud sequence. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 15028–15038 (2023), https://api.semanticscholar.org/CorpusID:265216067
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
19. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: Proceedings of the 2024 conference on empirical methods in natural language processing. pp. 5971–5984 (2024)
20. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013/
21. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
22. Liu, M., Shi, R., Kuang, K., Zhu, Y., Li, X., Han, S., Cai, H., Porikli, F., Su, H.: Openshape: Scaling up 3d shape representation towards open-world understanding. Advances in neural information processing systems **36**, 44860–44879 (2023)
23. Lu, J., Huang, C.H.P., Bhattacharya, U., Huang, Q., Zhou, Y.: Humoto: A 4d dataset of mocap human object interactions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10886–10897 (2025)
24. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing **508**, 293–304 (2022)
25. Mane, A.M., Weerakoon, D., Subbaraju, V., Sen, S., Sarma, S.E., Misra, A.: Ges3vig: Incorporating pointing gestures into language-based 3d visual grounding for embodied reference understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9017–9026 (2025)
26. OpenAI: Introducing GPT-5. https://openai.com/index/introducing-gpt-5/ (2025), accessed: 2026-03-05
27. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Isabelle, P., Charniak, E., Lin, D. (eds.) Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002). https://doi.org/10.3115/1073083.1073135, https://aclanthology.org/P02-1040/
28. Petrovich, M., Black, M.J., Varol, G.: Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9488–9497 (2023)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (July 2017)

30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
31. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
34. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 109–117 (2017)
35. Wang, M., Xing, J., Mei, J., Liu, Y., Jiang, Y.: Actionclip: Adapting language-image pretrained models for video action recognition. IEEE transactions on neural networks and learning systems **36**(1), 625–637 (2023)
36. Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., Feichtenhofer, C.: Videoclip: Contrastive pre-training for zero-shot video-text understanding. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 6787–6800 (2021)
37. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds. In: European Conference on Computer Vision. pp. 131–147. Springer (2024)
38. Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1179–1189 (2023)
39. Yu, Q., Tanaka, M., Fujiwara, K.: Exploring vision transformers for 3d human motion-language models with motion patches. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 937–946 (2024)
40. Zhang, B., Li, K., Cheng, Z., Hu, Z., Yuan, Y., Chen, G., Leng, S., Jiang, Y., Zhang, H., Li, X., et al.: Videollama 3: Frontier multimodal foundation models for image and video understanding. arXiv preprint arXiv:2501.13106 (2025)
41. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8552–8562 (2022)
42. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
43. Zhang, Y., Gong, Z., Chang, A.X.: Multi3drefer: Grounding text description to multiple 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15225–15236 (2023)