# Supplementary Materials For CL4D

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

## 1 Point Cloud Sequence generation

*SMPL Parameter Conversion:* Using the pose parameters from HumanML3D [4], we derive SMPL parameters using PriorMDM [5]. Using these SMPL parameters we generate dynamic human point cloud sequences to construct the 4D representations as explained in the following.

The original SMPL parameters are provided as:

- Joint rotations: $\boldsymbol{\theta} \in \mathbb{R}^{24\times6\times T}$ (rotation-6D representation),
- Root translation: $\mathbf{t} \in \mathbb{R}^{3\times T}$,
- Shape parameters: $\boldsymbol{\beta} \in \mathbb{R}^{10}$.

Since Unity requires quaternion-based rotations, we convert the $(24, 6, T)$ rotation-6D representation into quaternion format $(24, 4, T)$. The original motion data follows a right-handed, Z-up coordinate system, whereas Unity uses a left-handed, Y-up coordinate system. Therefore, we apply the following coordinate and rotation transformations to ensure consistency between the two systems:

- Translation: $(x, y, z) \rightarrow (-x, z, -y)$
- Quaternion: $(x, y, z, w) \rightarrow (-x, y, z, -w)$

The 24 SMPL joint names are mapped to indexed bone identifiers and loaded frame-by-frame into the SMPL mesh renderer within Unity.

*Scene Placement and Augmentation:* Each training animation is randomly placed within a $10 \times 10$ m area and assigned a random global rotation, serving as a form of spatial augmentation. In contrast, test sequences are positioned at the origin $(0, 0, 0)$ without additional rotation.

For rendering diversity, each animation is randomly assigned either a male or female SMPL mesh, a random shape from a set of predetermined shape parameters and a texture which is sampled from the SURREAL dataset [6].

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

*Point Cloud Rendering and Surface Sampling:* Animations are rendered at 20 FPS. For each frame, surface points are uniformly sampled to obtain approximately 2048 points per human instance. Although the rendering process produces RGB point clouds $(r, g, b, x, y, z)$, our model uses only the geometric coordinates $(x, y, z)$ during training and inference. This removes reliance on appearance cues and supports privacy-preserving deployment. We adopt area-weighted uniform surface sampling as follows:

*Step 1: Surface Area Computation.*

For each triangle in the mesh, we compute its surface area and store vertex coordinates. The total surface area $A_{\text{total}}$ is obtained by summation. The number of sampled points is computed as:

$$N = \text{round}(A_{\text{total}} \times \text{density}),$$

where the density is chosen to yield approximately 2048 points. A cumulative distribution function (CDF) is constructed from per-triangle areas.

*Step 2: Triangle Selection.* For each of the $N$ points, we sample $r \sim \mathcal{U}(0, A_{\text{total}})$ and use binary search over the CDF to select a triangle proportional to its area.

*Step 3: Barycentric Sampling.* Given triangle vertices $A, B, C$, two random variables $u, v \sim \mathcal{U}(0, 1)$ are sampled. If $u+v > 1$, we set $u = 1-u$ and $v = 1-v$. The surface point is then computed as:

$$P = A + u(B - A) + v(C - A),$$

producing uniformly distributed points across the mesh surface.

## 2 DynAction4D-Cluttered Environment Creation

### 2.1 Environment Generation and Clutter Synthesis

To synthesize diverse and realistic cluttered scenes, we curated a collection of 61 common household objects sourced from the Humoto dataset. To ensure scale variation across scenes, these objects were systematically categorized into three groups based on their bounding volumes: *large* ($n = 16$), *medium* ($n = 21$), and *small* ($n = 24$). The specific objects comprising each category are detailed as follows:

- **Large Objects:** bed, dining chair, low chair, working chair, table, side table, kitchen counter, small kitchen counter, utility cart, clothes rack, drawer, shelf, floor lamp, vacuum flask body, trash can, wash tub.
- **Medium Objects:** basket, 90-degree basket, woven basket, medium organizer, small organizer, mango, orange, laptop, laptop top, laptop bottom, mixing bowl, serving bowl, plastic bowl, stacked plastic bowls, wok turner, frying pan, ukulele, guitar, vase, tray, step stool.
- **Small Objects:** spoon, fork, knife, peeler, pen, screwdriver, whisk, turner, soap dispenser, soap dispenser body, soap dispenser top, USB, phone, notebook, mug, deep plate, side plate, cutting board, lint roller, shower squeegee, tap, can, flower, hammer.

To rigorously evaluate model generalization to unseen geometries, the total object pool was randomly partitioned into mutually exclusive training (70%) and testing (30%) sets.

We procedurally generated 20 distinct training environments and 8 testing environments. To guarantee morphological diversity within each scene, the environment instantiation algorithm mandates the inclusion of at least one randomly sampled object from each of the three size categories (restricted to the respective train/test split). To further increase scene complexity and semantic richness, 1 to 3 additional objects were uniformly sampled from the split's total available object pool and injected into the environment. Duplicate object instances within a single environment were filtered out, yielding unique, procedurally generated clutter compositions for every environment.

### 2.2 Motion Sequence Distribution

To construct the Cluttered dataset segment, motion sequences from the HumanML3D dataset were mapped to the synthesized environments. To prevent temporal or stylistic bias during training, the motion sequences corresponding to each split were first randomly shuffled.

The sequences were then uniformly partitioned across the generated environments. Formally, for a given split containing $S$ total sequences and $E$ generated environments, each environment was assigned approximately $\lfloor S/E \rfloor$ unique sequences. Any residual sequences resulting from indivisibility were allocated to the final environment to ensure exhaustive assignment.

## 3 Additional Ablation Studies

**Table 1: 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.

| | Text-to-Motion Retrieval | | | | | | | | | | Motion-to-Text Retrieval | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Global | | | | | Batchwise | | | | | Global | | | | | Batchwise | | | | |
| | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ |
| w. ViT-Tiny | **5.15** | **9.25** | **12.52** | 18.78 | 29.72 | 68.79 | 82.79 | 88.37 | 93.43 | **96.78** | 4.24 | 8.88 | 12.45 | 18.59 | 28.98 | 68.09 | 83.23 | **88.84** | 93.36 | **97.01** |
| w. ViT-Small | 4.80 | 8.81 | 12.42 | **19.17** | **29.95** | **69.61** | **83.72** | **88.87** | **93.44** | 96.71 | 4.59 | **9.32** | **12.68** | **19.75** | **31.02** | **70.41** | **83.78** | 88.76 | **93.47** | 96.85 |
| w. ViT-Base | 4.45 | 9.20 | 12.40 | 17.18 | 28.67 | 68.43 | 81.42 | 86.77 | 91.83 | 96.04 | **4.68** | 9.11 | 12.47 | 18.20 | 28.58 | 67.96 | 82.10 | 87.62 | 92.03 | 96.16 |
| w. ViT-Large | 4.66 | 8.99 | 12.15 | 17.83 | 27.86 | 66.95 | 80.81 | 86.53 | 91.42 | 95.90 | **4.68** | 8.74 | 11.80 | 17.48 | 27.91 | 66.31 | 80.09 | 85.89 | 91.78 | 95.72 |
| w. Frozen Text Enc. | 2.76 | 5.19 | 7.05 | 11.17 | 18.73 | 56.63 | 73.17 | 80.19 | 87.72 | 94.81 | 2.78 | 5.15 | 7.53 | 11.50 | 19.84 | 57.39 | 73.71 | 81.19 | 87.79 | 94.93 |
| w. untrained ViT | 3.52 | 6.93 | 9.25 | 13.93 | 22.74 | 63.32 | 78.48 | 84.93 | 90.90 | 96.46 | 3.36 | 6.00 | 9.13 | 13.79 | 23.37 | 62.79 | 78.73 | 84.67 | 90.86 | 96.32 |
| w. Shuffled frames | 3.08 | 5.68 | 8.39 | 12.96 | 21.42 | 60.97 | 76.39 | 82.90 | 89.70 | 96.04 | 2.97 | 5.89 | 8.69 | 12.56 | 20.84 | 58.97 | 74.50 | 82.04 | 89.79 | 96.16 |
| CL4D-mini | 4.24 | 8.58 | 11.85 | 17.80 | 28.12 | 67.18 | 81.14 | 86.82 | 92.04 | 96.64 | 4.78 | 9.23 | 12.12 | 17.52 | 27.65 | 66.02 | 81.68 | 87.58 | 92.78 | 96.94 |

**Table 2: Motion-text Retrieval results on the DynAction4D -HumanOnly dataset comparing different training batch sizes and different loss functions using Recall@1.**

| Loss fn. | Batch size | Text-to-Motion Retrieval | | | | | | | | | | Motion-to-Text Retrieval | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Global | | | | | Batchwise | | | | | Global | | | | | Batchwise | | | | |
| | | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ |
| SigLIP-like | 16 | 3.04 | 5.40 | 7.37 | 11.15 | 18.31 | 56.07 | 72.52 | 79.70 | 87.40 | 94.17 | 2.74 | 5.42 | 7.46 | 11.06 | 18.20 | 55.90 | 71.86 | 78.51 | 86.50 | 93.80 |
| | 32 | 4.20 | 7.79 | 11.06 | 16.90 | 27.93 | 65.60 | 78.73 | 84.47 | 89.73 | 94.07 | 3.92 | 7.83 | 11.27 | 17.41 | 27.82 | 65.41 | 79.01 | 84.43 | 89.43 | 94.51 |
| | 48 | 4.15 | 8.16 | 10.94 | 15.97 | 25.38 | 66.31 | 80.78 | 86.51 | 91.98 | 96.74 | 3.66 | 7.58 | 10.41 | 16.13 | 26.89 | 66.29 | 80.34 | 86.34 | 91.92 | 96.71 |
| | 64 | 4.03 | 7.70 | 10.92 | 17.01 | 27.68 | 66.97 | 81.10 | 87.10 | 92.25 | 96.57 | 4.20 | 8.60 | 11.94 | 17.50 | 28.03 | 67.14 | 81.69 | 86.85 | 91.76 | 96.39 |
| | 72 | 4.15 | 8.14 | 11.85 | 17.85 | 28.93 | 67.15 | 81.19 | 86.44 | 91.71 | 95.90 | 4.29 | 9.04 | 12.22 | 17.78 | 28.91 | 66.66 | 80.22 | 86.41 | 91.73 | 96.32 |
| InfoNCE (Ours) | 16 | 2.85 | 5.75 | 7.83 | 12.01 | 19.91 | 58.71 | 73.86 | 81.47 | 87.72 | 94.83 | 2.64 | 5.42 | 7.74 | 11.94 | 19.94 | 58.02 | 73.95 | 80.71 | 87.14 | 94.10 |
| | 32 | 4.82 | 8.90 | 12.15 | 17.55 | 28.81 | 65.62 | 79.00 | 84.35 | 89.78 | 94.42 | 5.08 | 9.06 | 12.24 | 18.27 | 29.79 | 65.86 | 78.39 | 84.12 | 89.85 | 94.65 |
| | 48 | 4.17 | 8.09 | 11.38 | 16.67 | 27.58 | 68.06 | 81.39 | 87.19 | 92.71 | **96.92** | 3.62 | 8.32 | 11.85 | 17.66 | 27.54 | 67.50 | 81.85 | 88.03 | 92.80 | **97.08** |
| | 64 | 4.45 | 9.20 | 12.40 | 17.18 | 28.67 | 68.43 | 81.42 | 86.77 | 91.83 | 96.04 | 4.68 | 9.11 | 12.47 | 18.20 | 28.58 | 67.96 | 82.10 | 87.62 | 92.03 | 96.16 |
| | 72 | **8.07** | **15.76** | **21.09** | **28.98** | **43.25** | **70.32** | **84.10** | **88.79** | **92.88** | 96.51 | **8.02** | **15.81** | **20.45** | **29.30** | **44.27** | **68.62** | **83.27** | **88.33** | **92.92** | 96.78 |

**Table 3: Effect of num. frames for text-to-motion and motion-to-text retrieval evaluated using Batchwise Recall@1.** Eval. Batch size = 32.

| Number of Frames | Text-to-Motion Retrieval | | | | | | | | | | Motion-to-Text Retrieval | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Global | | | | | Batchwise | | | | | Global | | | | | Batchwise | | | | |
| | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ |
| 4 | 2.06 | 4.03 | 6.07 | 9.46 | 15.83 | 49.85 | 65.71 | 73.56 | 82.44 | 91.41 | 2.11 | 4.27 | 6.26 | 9.74 | 16.48 | 49.85 | 65.84 | 72.88 | 81.52 | 91.82 |
| 8 | 2.94 | 6.19 | 8.83 | 13.17 | 22.35 | 59.63 | 74.84 | 81.09 | 87.76 | 93.95 | 3.59 | 7.32 | 10.20 | 15.35 | 23.60 | 60.35 | 74.68 | 81.45 | 87.25 | 93.83 |
| 16 | 3.08 | 6.19 | 8.88 | 13.95 | 23.32 | 61.77 | 75.92 | 82.20 | 88.88 | **94.46** | 3.43 | 7.26 | 10.27 | 14.58 | 24.08 | 61.26 | 76.29 | 82.22 | 88.30 | 93.95 |
| 32 | **4.82** | **8.90** | **12.15** | **17.55** | **28.81** | **65.62** | **79.00** | **84.35** | **89.78** | 94.42 | **5.08** | **9.06** | **12.24** | **18.27** | **29.79** | **65.86** | **78.39** | **84.12** | **89.85** | **94.65** |

**Table 4: Motion-text retrieval results on DynAction4D segments.** 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 eval. batch size = 32. CL4D outperforms the other 4D encoders in all the DynAction4D segments.

| DynAction4D Segment | Method | Text-to-Motion Retrieval | | | | | | | | | | Motion-to-Text Retrieval | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Global | | | | | Batchwise | | | | | Global | | | | | Batchwise | | | | |
| | | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ |
| HumanOnly | P4 Transformer | 3.66 | 7.23 | 10.25 | 15.39 | 26.15 | 53.57 | 69.79 | 78.17 | 86.86 | 93.93 | 5.33 | 10.25 | 14.60 | 20.54 | 30.09 | 58.05 | 73.14 | 80.82 | 89.34 | 95.31 |
| | Pst Transformer | 2.69 | 5.61 | 7.70 | 11.22 | 19.43 | 49.09 | 65.65 | 75.12 | 84.12 | 94.07 | 3.34 | 7.83 | 10.34 | 14.51 | 24.01 | 51.73 | 68.32 | 77.21 | 85.75 | 93.70 |
| | MotionPointNet | 2.34 | 4.31 | 6.31 | 9.16 | 15.51 | 51.91 | 69.11 | 77.25 | 85.90 | 94.12 | 3.13 | 6.37 | 8.51 | 12.87 | 20.96 | 55.78 | 72.61 | 79.39 | 87.11 | 94.48 |
| | CL4D (Ours) | **8.07** | **15.76** | **21.09** | **28.98** | **43.25** | **70.32** | **84.10** | **88.79** | **92.88** | **96.51** | **8.02** | **15.81** | **20.45** | **29.30** | **44.27** | **68.62** | **83.27** | **88.33** | **92.92** | **96.78** |
| ObjInteractions | P4 Transformer | 5.94 | 10.05 | 16.44 | 23.74 | 41.10 | 26.55 | 38.49 | 50.35 | 64.88 | 86.00 | 7.31 | 10.50 | 15.98 | 23.29 | 42.47 | 24.93 | 41.78 | 56.32 | 73.69 | 88.59 |
| | Pst Transformer | 8.22 | 14.61 | 20.09 | 28.31 | 44.75 | 30.37 | 47.39 | 56.04 | 69.68 | 88.59 | 7.31 | 12.79 | 18.72 | 30.59 | 47.49 | 25.50 | 47.06 | 63.62 | 75.84 | 91.72 |
| | MotionPointNet | 12.33 | 24.20 | 31.96 | 44.29 | 61.19 | 41.37 | 55.54 | 65.08 | 78.97 | **92.69** | 11.87 | 21.46 | 31.05 | 43.38 | 61.19 | 36.18 | 55.62 | 67.03 | 78.80 | **93.58** |
| | CL4D (Ours) | **23.29** | **39.27** | **47.03** | **56.62** | **68.04** | **49.40** | **63.29** | **73.00** | **83.52** | 91.35 | **22.37** | **33.33** | **42.47** | **53.42** | **69.86** | **46.97** | **63.46** | **71.58** | **81.56** | 91.27 |
| Cluttered | Pst Transformer | 0.93 | 1.78 | 2.41 | 3.78 | 6.35 | 31.47 | 46.36 | 56.82 | 69.92 | 86.56 | 1.02 | 1.85 | 2.69 | 4.94 | 9.06 | 34.45 | 49.12 | 57.60 | 69.19 | 84.32 |
| | MotionPointNet | 1.23 | 2.41 | 3.52 | 5.77 | 9.81 | 41.71 | 57.93 | 67.27 | 78.75 | 90.93 | 1.53 | 2.97 | 4.06 | 6.28 | 11.68 | 43.24 | 59.65 | 69.07 | 78.87 | 91.18 |
| | CL4D (Ours) | **3.11** | **6.00** | **8.48** | **12.49** | **20.63** | **55.07** | **69.94** | **77.53** | **84.58** | **92.87** | **2.71** | **5.54** | **7.58** | **11.82** | **18.94** | **51.94** | **67.43** | **75.77** | **83.34** | **91.31** |

**Table 5:** Model Parameters and Computational Complexity for different ViT Variants; Here we can observe the CL4D-mini has the lowest parameter count and computational cost.

| Model | Training Params ↓ | FLOPS (Gflops) ↓ |
|---|---|---|
| w. ViT-Tiny | 34.65 M | 176.2 |
| w. ViT-Small | 50.82 M | 177.64 |
| w. ViT-Base | 115.0 M | 183.28 |
| w. ViT-Large | 332.53 M | 202.42 |
| CL4D-mini | **8.93 M** | **70.22** |

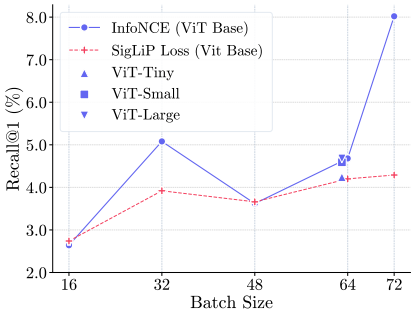



**Fig. 1:** Training batch size vs. Global Recall@1 on motion-to-text retrieval. Eval. batch size = 32.

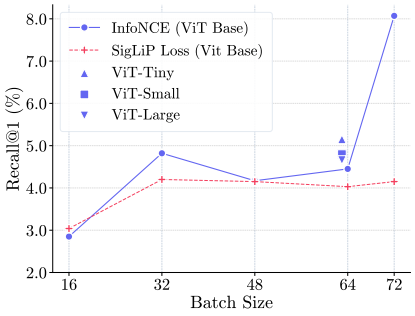



**Fig. 2:** Training batch size vs. Global Recall@1 on text-to-motion retrieval. Eval. batch size = 32.

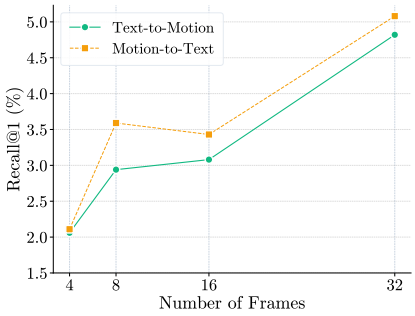



**Fig. 3:** Effect of Num. of frames used in the point cloud sequences for Global Recall@1 text-motion retrieval tasks.

## 4 Data Generation Prompt

We first use the following prompt to generate the detailed caption using both the video of the mesh of the point cloud sequence and the HumanML3D dataset caption. We do not consider texture and color information to generate question pairs. Listing 1.1 to Listing 1.4 shows the different prompt templates used for data generation and evaluations.

**Listing 1.1:** Prompt template for generating **detailed captions**

```
### CONTEXT
I am providing {n} consecutive frames from a video derived from the HumanML
    dataset.
Reference Ground Truth: "{humanML_groundtruth_caption}"

### TASK
Generate a single, detailed English caption that objectively narrates
the movement shown in the frames.

### GUIDELINES
1. **Visual Priority:** Use the Reference Ground Truth ONLY as a high-level
   guide for the context of the action. You must verify that the action
```

```
   actually occurs in the specific frames provided. If the visual contradicts
   the text, trust the text.
2. **Subject Description:** Refer to the subject simply as "the person". Do
   not describe their body type, age, face, or clothing unless it is
   mechanically relevant to the interaction (e.g., "holding a skirt").
3. **Motion Focus:** Focus 90 percent of the caption on the mechanics of the
   movement (limbs, posture, speed, trajectory).
4. **Chronology:** Describe the sequence of movements in strict chronological
   order.
5. **Constraints:**
   - NO background description.
   - NO symbolic interpretation (e.g., do not say "he looks sad," say
     "he walks with a slumped posture").
   - NO distinct physical features (hair, eyes, skin).

### OUTPUT
Provide only the caption text.
```

**Listing 1.2:** Prompt template for generating **QA pairs** from detailed captions

```
You are given a detailed caption describing a motion sequence from a dataset.
Your task is to generate generalised Question-Answer (QA) pairs derived
*strictly* from the information in the caption.

### CORE PRINCIPLES
1. **One action per question.** Each question must ask about exactly ONE
   action or movement, not a chain of actions.
2. **Keep questions general.** Ask broad questions about what the person
   does, not hyper-specific questions about minute details.
3. **No answer leakage.** The question must NOT contain or hint at the
   answer. The reader should not be able to guess the answer from the
   question alone.

### CATEGORIES
1. **Action:** Ask what the person does during a particular phase.
   - GOOD: "What does the person do at the start of the sequence?"
   - GOOD: "How does the person move after standing up?"
   - BAD: "Does the person raise their left arm?" (answer is leaked)
   - BAD: "How does the person bend their knees while simultaneously
     rotating their torso?" (too specific, multiple actions)
2. **Sequence:** Ask about the order of events without revealing what happens
    .
   - GOOD: "What does the person do after the first action?"
   - GOOD: "What is the final movement in the sequence?"
   - BAD: "Does the person jump before walking?" (answer leaked)
3. **Body Position:** Ask about body posture or positioning during a movement
    .
   - GOOD: "What is the position of the person's arms during the movement?"
   - GOOD: "How is the person's body oriented at the end?"
   - BAD: "Are the person's arms raised above the head?" (answer leaked)

### RULES
- Generate 3 to 6 QA pairs per caption.
- If the caption does not contain information for a category, omit that
  category. Do NOT hallucinate.
- The answer must stand alone without needing the question for context.
  Refer to "the person" instead of pronouns.
- Do not copy-paste the caption. Rephrase naturally.
- Avoid yes/no questions. Prefer open-ended "What" / "How" questions.

### INPUT CAPTION
'{generated_caption}'

### OUTPUT FORMAT
Provide the output in valid JSON format only. Do not add markdown backticks.
{
  'qa_pairs': [
```

```
    {
      'type': 'Action',
      'question': '...',
      'answer': '...'
    },
    {
      'type': 'Sequence',
      'question': '...',
      'answer': '...'
    },
    {
      'type': 'Body Position',
      'question': '...',
      'answer': '...'
    }
  ]
}
```

**Listing 1.3:** Prompt Template for video evaluation.

```
You are a helpful assistant that analyzes videos. Answer the user's questions
      based on the video content directly and concisely.
```

**Listing 1.4:** Prompt Template for point cloud sequence evaluation

```
Default template of Vicuna_V1
```

## 5 Use of Pre-Trained ViT weights for Temporal Encoder initialization

The use of a pre-trained Vision Transformer (ViT) [2] weight initialization for our temporal encoder is motivated by its capacity to leverage high-level "visual knowledge" and global receptive fields acquired from large-scale image datasets like ImageNet [1]. In domains like 3D human motion, where high quality annotated data is significantly scarcer than natural images, transfer learning can be used to overcome the data scale problem [7]. By stacking spatial features into a grid-like representation, we effectively transform temporal dynamics into a "motion spectrogram" or motion image, where complex physical actions transformed as visual textures. This approach follows established methodologies in audio tagging and motion retrieval [3], [7], which treat time-series sequences as 2D visual inputs to exploit the feature extraction capabilities of pre-trained image models. Unlike traditional recurrent architectures, the ViT's self-attention mechanism captures global spatial-temporal dependencies and long-range relationships in a single pass.

## 6 Data Statistics

### 6.1 DynAction Dataset Statistics

The composition of our DynAction dataset segments of DynAction4D-HumanOnly, DynAction4D-ObjInteractions and DynAction4D-Cluttered were shown in the

plots Fig. 4 to Fig. 6 respectively. Here we analyzed the compositions in terms of human actors' total distance traveled, Net displacement, average velocity, number of frames per each sequence and the composition of the objects.

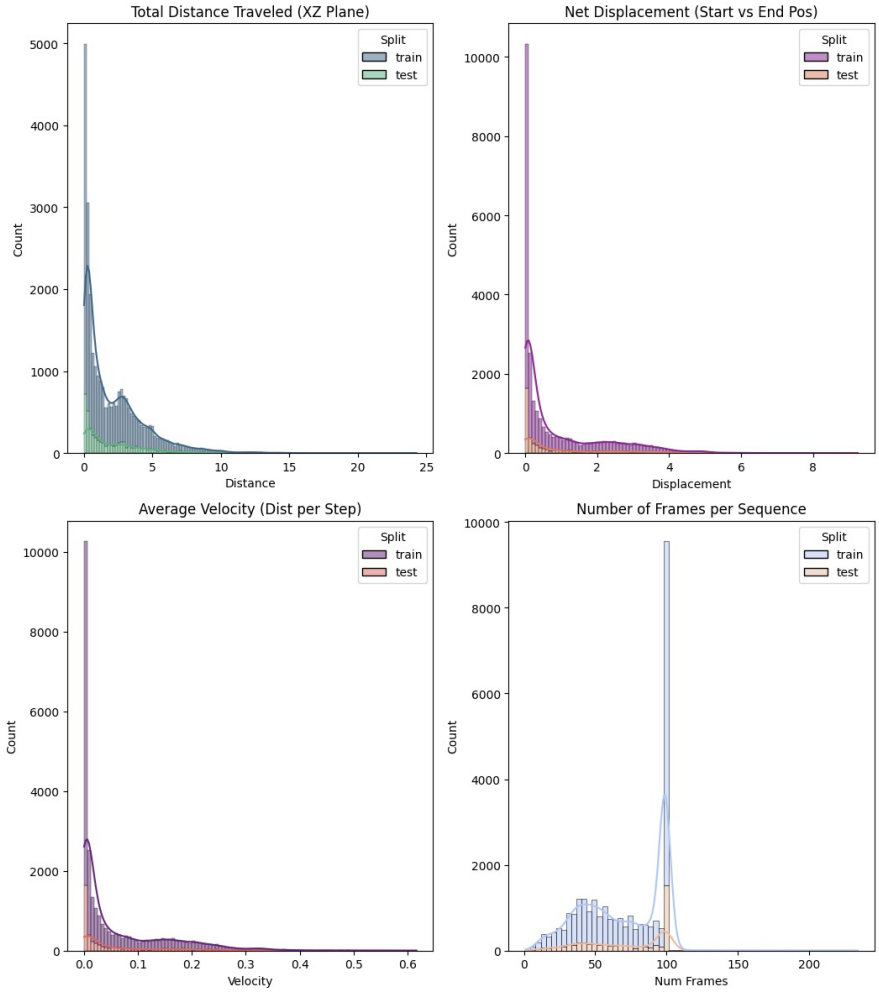



**Fig. 4:** Statistics of the DynAction4D-HumanOnly dataset segment.

### 6.2 DynAction-VQA Dataset training Split Statistics

The training split contains **22,965** QA files with a total of **134,070** question–answer pairs. On average, each sequence contains **5.84** QA pairs (min = 3, max = 9). The average number of QA pairs per category per sequence is: *Action*: 1.94, *Temporal*: 1.82, and *Body-Spatial*: 2.08.

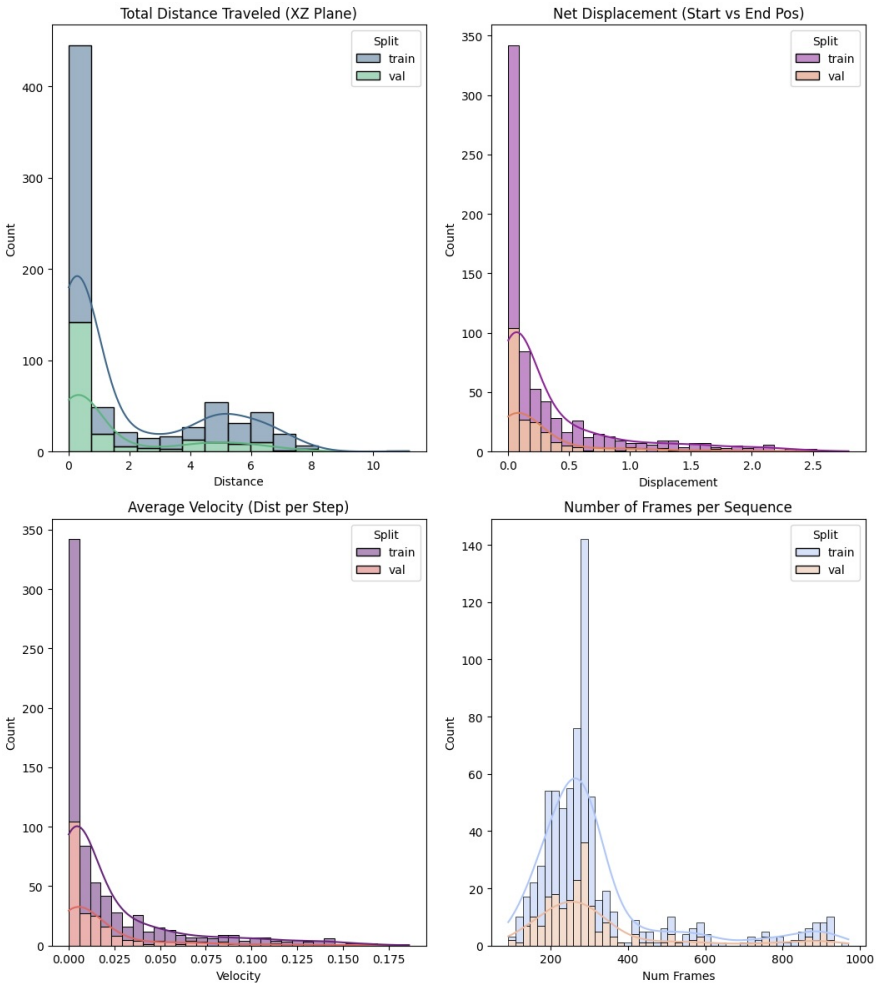



**Fig. 5:** Statistics of the DynAction4D-ObjInteractions dataset segment.

**Table 6:** Word count statistics for questions ($q\_len$) and answers ($a\_len$) in the DynAction-VQA training split.

| **Type** | $q\_len$ | | | $a\_len$ | | |
|---|---|---|---|---|---|---|
| | Min | Max | Mean | Min | Max | Mean |
| Action | 5 | 36 | 11.66 | 7 | 55 | 21.66 |
| Body-Spatial | 5 | 23 | 12.82 | 7 | 43 | 18.67 |
| Temporal | 5 | 23 | 12.43 | 7 | 45 | 19.76 |

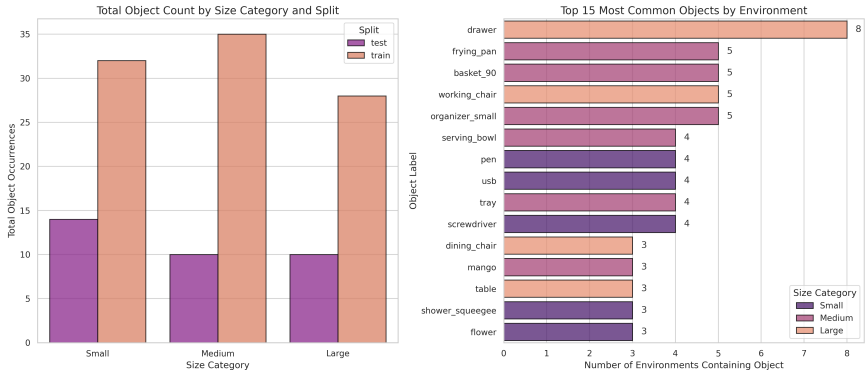



**Fig. 6:** Statistics of the DynAction4D-Cluttered dataset segment.

### 6.3 DynAction-VQA Dataset test Split Statistics

The test split contains **4,314** QA files with a total of **25,117** QA pairs. Each sequence contains on average **5.82** QA pairs (min = 3, max = 8). The average number of QA pairs per category per sequence is: *Action*: 1.94, *Temporal*: 1.81, and *Body-Spatial*: 2.06.

**Table 7:** Word count statistics for questions (*q_len*) and answers (*a_len*) in the DynAction-VQA test split.

| **Type** | *q_len* | | | *a_len* | | |
|---|---|---|---|---|---|---|
| | Min | Max | Mean | Min | Max | Mean |
| Action | 5 | 26 | 11.68 | 7 | 52 | 21.75 |
| Body-Spatial | 6 | 23 | 12.82 | 7 | 54 | 18.68 |
| Temporal | 5 | 23 | 12.45 | 7 | 41 | 19.76 |

### 6.4 Overall DynAction-VQA Dataset Statistics

Across the full dataset, DynAction-VQA contains **27,279** QA files and **159,187** QA pairs. Each sequence contains on average **5.84** QA pairs (min = 3, max = 9). The average QA distribution per sequence is: *Action*: 1.94, *Temporal*: 1.82, and *Body-Spatial*: 2.07.

**Table 8:** Word count statistics for questions ($q\_len$) and answers ($a\_len$) in combined DynAction-VQA dataset.

| **Type** | $q\_len$ | | | $a\_len$ | | |
|---|---|---|---|---|---|---|
| | Min | Max | Mean | Min | Max | Mean |
| Action | 5 | 36 | 11.67 | 7 | 55 | 21.67 |
| Body-Spatial | 5 | 23 | 12.82 | 7 | 54 | 18.68 |
| Temporal | 5 | 23 | 12.43 | 7 | 45 | 19.76 |

## 7 Samples from the DynAction4D-VQA segment and the 4DVLM Predictions

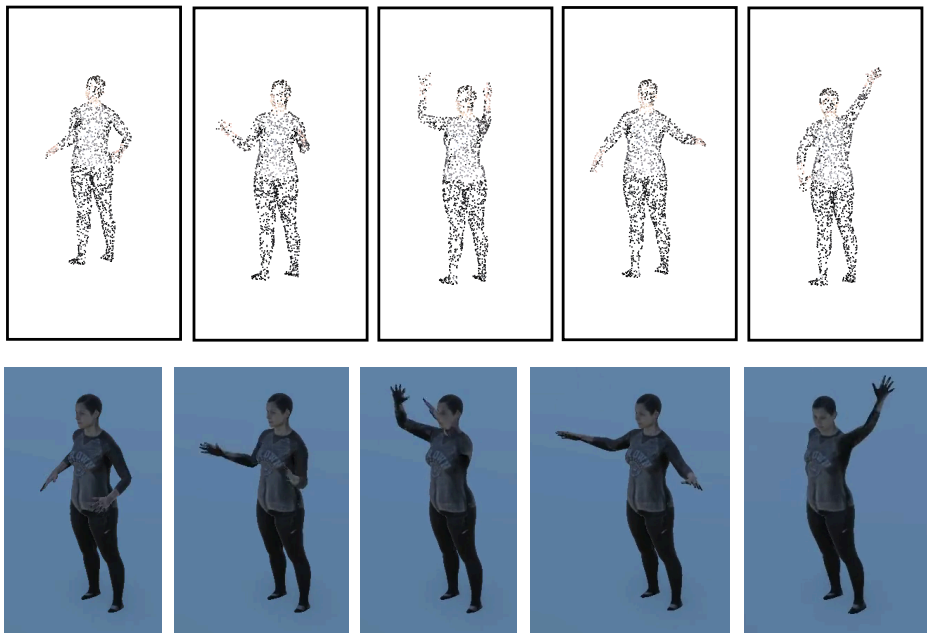
Here we provide some figures of our DynAction4D-VQA segment samples in Fig. 7 to Fig. 10. Each sample includes VQA samples, a point cloud sequence with rendered video sequence used for evaluations with video-LLM models.

## References


1. 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)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), `https://arxiv.org/abs/2010.11929`
3. Gong, Y., Chung, Y., Glass, J.R.: PSLA: improving audio event classification with pretraining, sampling, labeling, and aggregation. CoRR **abs/2102.01243** (2021), `https://arxiv.org/abs/2102.01243`
4. 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)
5. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior. In: The Twelfth International Conference on Learning Representations (2024)
6. 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)
7. 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)

– **Q:** What does the person do at the very beginning of the motion sequence?
**GT:** The person walks forward with a balanced posture and coordinated arm movements.
**4DVLM:** The person starts from a standing position and begins to walk forward.
– **Q:** What action occurs after the person completes the turn?
**GT:** The person continues walking in the opposite direction with a rhythmic pace similar to the start.
**4DVLM:** The person performs a 180-degree turn to the right and returns to a steady walking pace.
– **Q:** How does the entire sequence conclude?
**GT:** The person gradually decreases speed until coming to a complete stop with both feet aligned together.
**4DVLM:** The person returns to a neutral standing position and remains still for a few moments.
– **Q:** How does the person change direction during the sequence?
**GT:** The person slows down and performs a 180-degree clockwise turn by pivoting on the balls of the feet.
**4DVLM:** The person performs a 180-degree turn to the right while maintaining a steady pace.

**Fig. 7:** Qualitative results for `sequence_000022`. The top row visualizes the point cloud sequence, while the bottom row displays the corresponding rendered video. The corresponding question-answering pairs, ground truth (GT), and model predictions (4DVLM) are provided below.



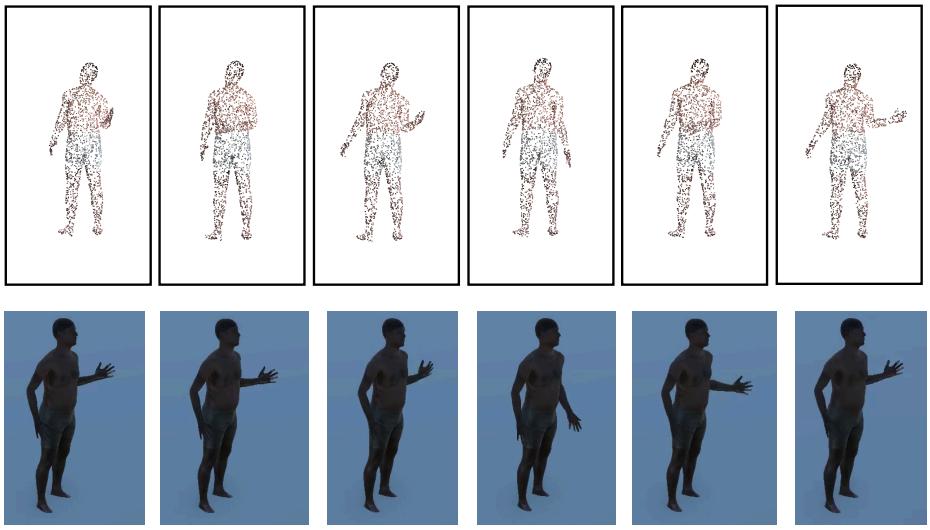
– **Q:** What occurs immediately after the person forms a V-shape with their arms?
**GT:** The person moves the arms downward, resting the left hand on the hip while stretching the right arm out to the side.
**4DVLM:** The person performs a rapid, alternating arm swing to the right side.

– **Q:** How are the person's hands and arms positioned after the initial shoulder-height lift?
**GT:** The person raises the arms into a broad V-shape with the palms of the hands directed forward.
**4DVLM:** The person's hands are held together in front of the chest with the palms facing upward.

– **Q:** What is the position of the person's lower body throughout the activity?
**GT:** The person remains standing in a single spot with the feet firmly planted and the lower body completely still.
**4DVLM:** The person maintains a stable and stationary stance with their feet planted firmly on the ground.

– **Q:** What does the person do with their left arm after extending it out to the side?
**GT:** The person pulls the left arm in toward the chest area and then moves it upward toward the face.
**4DVLM:** The person brings the left arm back toward the body and down toward the waist.

**Fig. 8:** Qualitative results for `sequence_000048`. The top row visualizes the point cloud sequence, while the bottom row displays the corresponding rendered video. The corresponding question-answering pairs, ground truth (GT), and our model's predictions (4DVLM) are provided below.



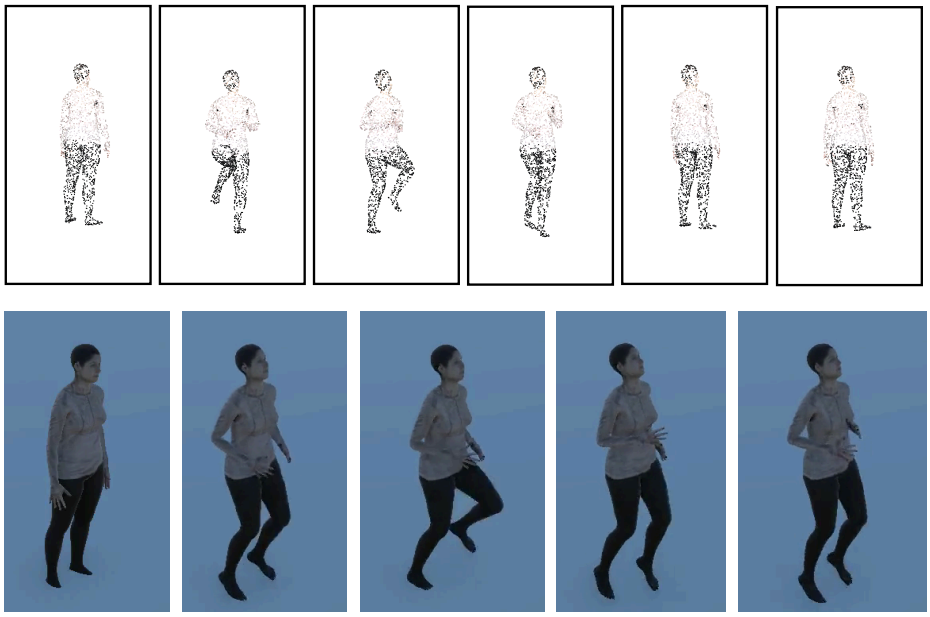
– **Q:** What is the configuration of the right hand during the upward motion?
**GT:** The person keeps the right hand open with the fingers extended during the ascent.
**4DVLM:** The person keeps the right hand in a neutral position with the palm facing forward.

– **Q:** How is the person's lower body positioned during the sequence?
**GT:** The person stands with straight legs placed about shoulder-width apart to maintain a stable and still base.
**4DVLM:** The person maintains a stable and upright posture with their feet planted firmly on the ground.

– **Q:** What happens after the person raises their hand to mid-chest height?
**GT:** The person lowers the arm back down toward the hip before beginning the next repetition.
**4DVLM:** The person lowers their hand back down to their side.

– **Q:** What movement does the person perform with their right arm?
**GT:** The person repeatedly lifts and lowers the right forearm by pivoting at the elbow in a rhythmic flexion and extension motion.
**4DVLM:** The person raises the right arm to a vertical position and then brings it down toward the left side of the body.

**Fig. 9:** Qualitative results for `sequence_000055`. The top row visualizes the point cloud sequence, while the bottom row displays the corresponding rendered video. The corresponding question-answering pairs, ground truth (GT), and our model's predictions (4DVLM) are provided below.



– **Q:** What is the posture of the person's upper body during the jogging motion?
**GT:** The person maintains an upright torso throughout the duration of the stationary jog.
**4DVLM:** The person maintains a slight forward lean with the torso tilted slightly forward.

– **Q:** How are the person's arms positioned throughout the main activity?
**GT:** The person keeps the elbows bent and swings the arms forward and backward in coordination with the opposite leg movements.
**4DVLM:** The person keeps their arms hanging loosely at their sides with the elbows bent.

– **Q:** What is the final action performed in the sequence?
**GT:** The person slows their pace until both feet are lowered back to the ground and they return to a relaxed standing posture.
**4DVLM:** The person returns to a neutral standing position with both feet placed together.

– **Q:** What does the person do once the initial movement is established?
**GT:** The person performs a rhythmic, alternating jogging motion in place, driving their knees upward while the arms swing in opposition to the legs.
**4DVLM:** The person performs a series of repetitive, rhythmic, and synchronized arm movements.

**Fig. 10:** Qualitative results for `sequence_000099`. The top row visualizes the point cloud sequence, while the bottom row displays the corresponding rendered video. The corresponding question-answering pairs, ground truth (GT), and our model's predictions (4DVLM) are provided below.