Iris: Empowering Video MLLMs with High-Frequency Pose Priors via Spatiotemporal Binding

Jiahang Zhang[1], **Yushuo Guan**[2], **Yuanxing Zhang**[2], **Pengfei Wan**[2], **Jiaying Liu**[1*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]KlingAI Research

Abstract

Multimodal Large Language Models (MLLMs) have demonstrated remarkable effectiveness in general video understanding. Human motion understanding, a dominant topic in video analytics, however, still remains as a bottleneck due to the lack of explicit structural cues in 2D visual patches and the significant motion information loss caused by sparse visual temporal sampling. To address this, we propose ***Iris*** (**I**ntegrating **r**epresentations of **i**n-the-wild **s**keletons), a novel pose-augmented video MLLM tailored for human-centric motion understanding. Iris adopts an asymmetric dual-stream architecture, pairing the traditional sparse vision stream with a lightweight, high-frequency human pose stream to provide spatially structured and temporally dense kinematic signals. To enable pose-vision modality correspondence awareness, we design a cross-modal spatiotemporal binding mechanism, featuring a pose-structured 3D RoPE for precise temporal alignment and a person-grounded cross-attention module for explicit spatial visual grounding. Furthermore, to enable robust large-scale training, we construct an automated pose curation pipeline that organizes pose data into a delicate heterogeneous representation. Extensive experiments demonstrate that Iris achieves superior performance on multiple fine-grained motion benchmarks, while also maintaining highly efficient token computation overhead. Our project page is available at `https://jhang2020.github.io/Projects/Iris/`.

1 Introduction

Multimodal Large Language Models (MLLMs) [20, 23, 53, 2] have demonstrated remarkable capacity by integrating visual and textual modalities, enabling them to handle diverse video-related applications, from macro-level semantic reasoning to video captioning and question answering.

Despite their success in capturing broad scene contexts, current MLLMs often achieve suboptimal performance on the inherently complex, fine-grained, and continuous spatiotemporal dynamics of human motions as revealed in recent studies [38, 4, 57, 32]. On the one hand, traditional 2D visual patches lack explicit structural cues, making the pure vision MLLMs struggle to accurately capture and model motion dynamics. On the other hand, existing paradigms predominantly rely on sparse visual sampling (*e.g.*, sampling 16 frames or 1 fps per clip) for token efficiency. While effective for general video context or static object analysis, it inevitably leads to temporal information loss of high-frequency kinematic details for motion understanding. To mitigate these issues, some recent works attempt to explicitly inject motion cues via visual prompting (*e.g.*, rendering markers to indicate trajectories or objects directly on RGB frames) [51, 3]. However, these approaches can inevitably cause visual corruption, *e.g.*, the occlusion of native appearance features, or introduce significantly increased token computational burdens [9], making them difficult to scale.

*Corresponding author. This work was done at KlingAI.

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

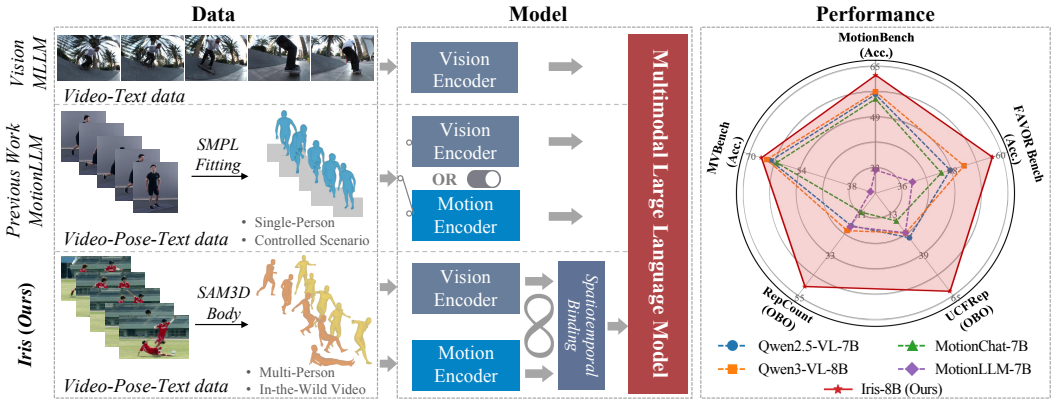


Figure 1: **Comparison of Iris with existing paradigms**. Unlike previous MLLMs with pure vision or unaligned, limited-diversity pose streams (single-branch inference with vision *or* motion), Iris leverages large-scale, in-the-wild video-pose-text triplet data. Through a novel *spatiotemporal binding* mechanism, Iris achieves deep correspondence-aware fusion, yielding state-of-the-art performance on multiple motion understanding benchmarks.

To address this, introducing ***human pose*** into MLLMs presents a highly promising alternative. Unlike raw RGB pixels, human pose serves as a highly lightweight and spatially structured representation for motion. Pioneering works like MotionLLM [6] have made valuable efforts exploring injecting pose tokens into MLLMs. However, existing methods remain in the nascent stage with several critical limitations as shown in Figure 1. ***At the data level***, existing methods predominantly rely on marker-based MoCap datasets [13, 30, 31] restricted to controlled, limited-diversity, single-subject environments, lacking in-the-wild generalization capacity. Furthermore, their training data are often purely derived from motion-isolated descriptions, lacking the motion-related visual grounding and context reasoning corpus. ***At the model level***, previous approaches naively sample the pose sequence for tokenization and then directly concatenate these pose tokens with the visual stream. This strategy ignores the efficiency of pose data for utilization and lacks spatiotemporal pose-vision correspondence priors, limiting the model for better cross-modal integration. Consequently, these restrict the model capacity for generalized and effective motion understanding in complex, real-world scenarios.

To this end, we present **Iris**, a novel pose-augmented video MLLM **I**ntegrating **r**epresentations of **i**n-the-wild **s**keletons for human-centric motion understanding. To bridge the aforementioned data gap, we first construct an automated, large-scale video-pose curation pipeline tailored for in-the-wild scenarios. Within this pipeline, we introduce a novel heterogeneous pose representation that involves 2D coordinates and 3D positions with camera-related decoupling cues, yielding highly robust kinematic priors. To bridge the model architectural gap, Iris adopts an asymmetric dual-stream design, *i.e.*, a traditional sparse vision stream paired with a lightweight, high-frequency pose stream, providing more informative motion dynamics. In contrast to naive multi-modal token concatenation, we introduce a *spatiotemporal binding mechanism* for better cross-modal correspondence learning, including a pose-vision parallel 3D RoPE for precise temporal alignment and a person-grounded cross-attention module for explicit spatial visual grounding. Extensive experiments demonstrate that Iris achieves superior performance on multiple fine-grained motion benchmarks, especially the repetitive action counting task, while preserving the native general video understanding capability.

Our contributions can be summarized as follows:

- We propose Iris, a novel pose-augmented video MLLM tailored for human-centric motion understanding. By introducing an asymmetric sampled high-frequency pose stream alongside the traditional sparse vision stream, Iris provides spatially structured and temporally dense motion priors, enabling fine-grained motion perception and more precise reasoning.
- To boost the pose-vision correspondence learning, we design a spatiotemporal binding mechanism including a temporally aligned pose-vision parallel 3D RoPE and a spatially grounded pose-vision interaction module. This effectively facilitates precise multi-modal alignment and integration.
- To enable robust large-scale training, we construct an automated video-pose curation pipeline featuring a heterogeneous pose representation. Extensive experiments demonstrate that Iris achieves

superior performance on multiple fine-grained motion understanding benchmarks, while with efficient computation overhead.

2 Related Work

MLLMs for Video Understanding. Driven by the success of Large Language Models (LLMs), recent years have witnessed a surge in MLLMs designed for video understanding. Pioneering works [20, 54, 58, 43, 2, 1] have demonstrated remarkable capabilities in tackling semantic reasoning, video captioning, and global visual QA. Typically, these models align pre-trained vision encoders with LLMs via lightweight projectors. Mavors [36] explores multi-granularity video representations. However, directly applying these pure-vision MLLMs to fine-grained, spatiotemporally continuous human motion understanding often proves sub-optimal [38, 4, 57, 32]. They lack explicit kinematic representations, hindered by the temporal information loss from sparse visual sampling and the absence of structural priors in standard 2D visual patches. SlowFast [11] and SlowFast-LLaVA variants [47, 48] use visual pathways at different temporal rates to balance spatial detail and motion coverage. Iris instead pairs sparse RGB with dense structured pose and explicitly binds these complementary modalities in space and time.

MLLMs with Human Pose / Motion Data. To inject explicit kinematic priors into LLMs, several works have explored integrating human pose/motion data. Early attempts primarily focused on unifying motion generation and understanding within a pure motion-text paradigm. For instance, MotionGPT [18] employs a VAE to map 3D motion into discrete tokens with T5 [33] LLM for autoregressive generation. Building upon this, MGMotionLLM [45] introduces part- and event-level fine-grained textual descriptions to enhance motion comprehension. However, because these models operate entirely without visual inputs, they are fundamentally restricted to context-agnostic motion analysis, struggling to generalize to real-world video understanding with rich visual contexts.

To bridge motion with visual context, subsequent works have attempted cross-modal integration. Li et al. [19] study subject-centered spatial understanding for human-centric image captioning. In the 2D domain, ChatPose [12] introduces a pose projection network at the output stage for text-to-pose generation and visual pose estimation, failing to support pose input or motion understanding. For daily activity understanding, SKI [37] transfers skeleton knowledge into vision-language models for RGB-only inference, while LLAVIDAL [34] progressively aligns video, skeleton, and object cues. For video-level tasks, MotionLLM [6] incorporates 3D pose data, yet it adopts an isolated inference process where either the motion branch *or* the vision branch is activated, inherently constraining the potential for simultaneous cross-modal interaction. HMVLM [16] utilizes Mixture-of-Experts with LoRA to integrate the pose tokens by direct concatenation. LLaMo [22] introduces language-guided frame selection to enhance representation quality, while still being bottlenecked by the limited-scale training data and backbone capacity. In contrast to these methods, our proposed Iris introduces a correspondence-aware, dual-branch inference paradigm. Empowered by a rigorously curated, large-scale video-pose-text dataset, Iris leverages a fine-grained spatiotemporal binding mechanism to achieve deeply integrated, generalized, and highly efficient human-centric motion understanding.

3 Iris: A Pose-Augmented MLLM

3.1 Motivation and Overview

As discussed before, most existing video MLLMs rely on visual features solely, which are inherently limited in capturing temporally dense action patterns and spatially grounded body semantics. Human pose, as a structured and lightweight body representation, provides motion-descriptive features due to its explicit semantics and efficiency. To this end, we propose Iris, a pose-augmented video MLLM that introduces human pose as a new modality alongside RGB frames. Crucially, Iris features an *asymmetric sampling* strategy and a correspondence learning paradigm, involving a sparsely sampled visual stream and a densely sampled pose stream with a novel *spatiotemporal cross-modal binding* mechanism. The overall framework is shown in Figure 2 with details unfolded below.

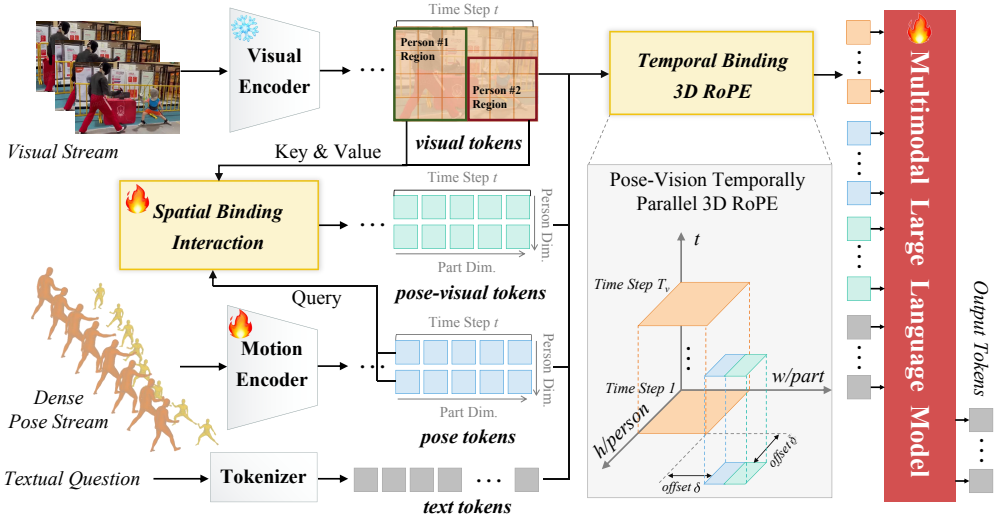


Figure 2: **Overview of *Iris* model**. Given a video, the visual encoder extracts *visual tokens* at a low frame rate, while the pose encoder tokenizes an asymmetrically densely sampled pose sequence into part-aware *pose tokens*. A spatial binding interaction module grounds each person's pose tokens to their corresponding visual region with cross-attention, producing *pose-visual tokens*. Then, temporal binding 3D RoPE assigns shared temporal position indices to visual (orange) and pose / pose-visual (blue and cyan) tokens. All tokens are finally fed into the MLLM for multi-stage training.

3.2 Heterogeneous Pose Representation

Rather than adopting raw human body keypoint coordinates only, we construct a heterogeneous per-keypoint descriptor to exploit the MLLM modeling capacity. Formally, for each keypoint k at frame t, we define a 9-dimensional feature vector:

$$\mathbf{p}_k^t = [u_k,\ v_k,\ x_k,\ y_k,\ z_k,\ x'_0,\ y'_0,\ z'_0,\ c_k]\,, \tag{1}$$

where (u_k, v_k) are normalized 2D image coordinates, serving as spatial priors for potential visual grounding; (x_k, y_k, z_k) are 3D coordinates relative to the pelvis center, representing local body articulation separately from pelvis translation; (x'_0, y'_0, z'_0) is the absolute pelvis position in the camera coordinate system to encode global displacement cues; and $c_k \in \{0, 1\}$ is a binary visibility proxy indicating whether the projected keypoint lies within the frame and its person instance mask. Among them, we normalize the (x_k, y_k, z_k) and (x'_0, y'_0, z'_0) dimensions to control the scale and stabilize the training of the motion (pose) encoder.

3.3 Part-Aware Pose Tokenization with Asymmetric Sampling

Given a temporally downsampled video, the visual encoder tokenizes it into the visual tokens $x_v \in \mathbb{R}^{T_v \times HW \times D}$. For the pose modality, we instead perform more dense sampling with a pose sequence of $T_p = T_v \times T_{\text{clip}}$ frames, where T_{clip} is the temporal densification factor. This asymmetric sampling exploits the lightweight nature of pose sequences to preserve fine-grained temporal dynamics that would otherwise be lost under sparse visual sampling.

The pose input is organized as a tensor $\in \mathbb{R}^{T_p \times P \times V \times 9}$, where P, V, and 9 are the person number, keypoint number, and per-keypoint feature dimension. Following MotionPatch [52], we organize keypoints into $N_p = 5$ semantic body parts: torso, left/right arm, left/right leg. For each part, the corresponding keypoints are concatenated along the feature dimension and encoded via a 1D temporal convolution with stride T_{clip}, producing part-wise tokens with T_v time steps (the same as x_v). Then, these tokens are summed with learnable spatial, temporal, and person positional embeddings and passed through a standard 8-layer Transformer encoder. The final tokens are projected to the MLLM hidden dimension via a 2-layer MLP projector, yielding a pose token sequence of shape $[T_v, P, N_p, D]$, *i.e.*, $N_p = 5$ part tokens per person at each time step, which serves as input to the spatiotemporal binding module.

3.4 Pose-Vision Spatiotemporal Binding

Directly stacking the pose and visual tokens is suboptimal because it ignores the spatiotemporal correspondence of pose and video. As illustrated in Figure 2, the multimodal token sequence finally fed to the MLLM consist of visual tokens $x_v \in \mathbb{R}^{T_v \times HW \times D}$, pose tokens $x_p \in \mathbb{R}^{T_v \times P \times N_p \times D}$, pose-visual interacted tokens $x_{pv} \in \mathbb{R}^{T_v \times P \times N_p \times D}$, and textual tokens as queries in sequence. Next, we will introduce the *spatiotemporal binding mechanism* for pose-vision correspondence learning.

Temporal Binding via Pose-Vision Parallel 3D RoPE. It aims to associate pose tokens with their corresponding visual content at the same time step, despite their separate positions in token sequence.

Specifically, we choose the popular Qwen-VL model families [2, 1] to build Iris, which employ Multimodal Rotary Position Embedding (M-RoPE) [43]. It assigns each token a 3D position index (t, h, w) encoding its temporal and spatial location for video tokens. For visual video tokens, the position can be roughly denoted as (ignoring the global offset accumulated from previous tokens):

$$\text{Position-ID}(x_v^{t,h,w}) = (t \cdot \Delta,\ \ h,\ \ w), \tag{2}$$

where Δ is the temporal interval inherited from the video's sampling rate, and h, w are the spatial positions. We extend this scheme to pose tokens by replacing the spatial axes (h, w) with pose dimension $(person, part)$, assigning each pose token:

$$\text{Position-ID}(x_p^{t,p,n_p}) = (t \cdot \Delta,\ \ p + \delta,\ \ n_p + \delta), \tag{3}$$

where n_p denotes the part index and δ is a hyperparameter to define offsets, placing pose tokens in a dedicated region of the position space for modality isolation. Critically, the temporal coordinate $t \cdot \Delta$ is ***shared*** between visual and pose tokens at the same time step, constructing a temporally parallel pose-vision position system within the MLLM. This enables the model to learn the temporal correspondence between visual and pose tokens in the subsequent self-attention layers.

Spatial Binding via Person-Grounded Pose-Visual Attention. While temporal binding establishes *when* pose and visual tokens correspond, it does not resolve *where* in the image each pose instance is located, which is also an ambiguity in multi-person scenarios where multiple persons coexist in a single frame. Although the input 2D joint image coordinates provide grounding clues, the model still struggles to learn this nontrivial correspondence. Therefore, we propose a spatial binding mechanism that explicitly grounds each person's pose tokens to their corresponding visual region.

The key observation is that the 2D keypoint coordinates (u_k, v_k) in our heterogeneous pose representation naturally provide a spatial prior, from which we can derive the bounding box in the image plane, implying the visual patches in the target person region. We leverage this to construct a *person-specific* cross-attention between pose and visual tokens. Formally, for each time step t and person p, we derive a bounding box $\mathcal{B}^{t,p}$ from the 2D keypoint coordinates. The corresponding visual tokens $x_v^t \in \mathbb{R}^{HW \times D}$ are then masked such that only patches falling within $\mathcal{B}^{t,p}$ are attended to. The pose-visual interacted tokens x_{pv} are obtained via:

$$x_{pv}^{t,p} = \text{CrossAttn}\left(Q = x_p^{t,p},\ \ K = V = x_v^t,\ \ Mask = \mathcal{M}^{t,p}\right), \tag{4}$$

where $\mathcal{M}^{t,p} \in \{0, 1\}^{HW}$ is a binary mask derived from $\mathcal{B}^{t,p}$, implemented as a simple key padding mask in the attention operation. We use two cross-attention layers as CrossAttn. This forces each person's pose tokens to attend exclusively to their corresponding spatial region in the visual stream, avoiding cross-person binding interference and extracting useful visual clues in an early fusion manner. The resulting $x_{pv} \in \mathbb{R}^{T_v \times P \times N_p \times D}$ are *concatenated with the original pose tokens x_p along the part dimension for temporal binding 3D RoPE*. Then, they are fed into the MLLM alongside the visual tokens, providing both raw structural pose features and spatially grounded visual context.

The cross-attention is implemented as a lightweight module with a small bottleneck dimension, followed by a projection head. Crucially, the projection head is zero-initialized, ensuring x_{pv} initially contributes negligibly to allow the spatial binding to activate gradually during training.

3.5 Training Strategy

We adopt a three-stage progressive training strategy to integrate pose data into the pretrained MLLM.
Stage 1: Contrastive Pretraining of Pose Encoder. Training the pose encoder end-to-end from scratch within the full MLLM is challenging, as gradient signals must propagate through the entire

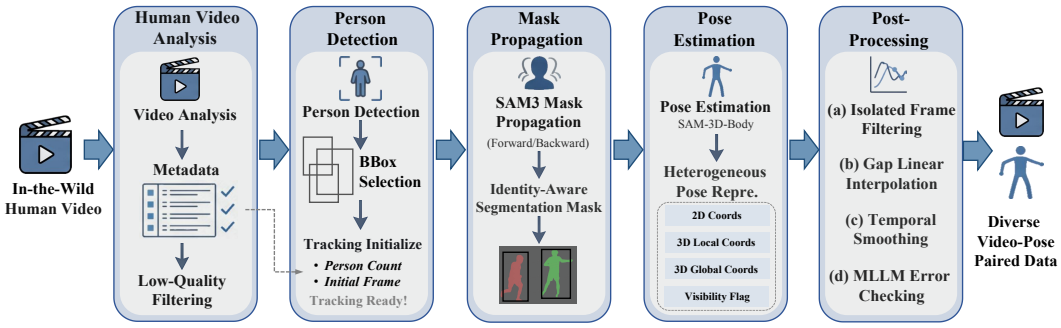


Figure 3: Illustration of main data flow in our pose data collection pipeline.

LLM before reaching the pose encoder. To this end, we first perform contrastive pretraining to cold-start the pose encoder independently. Specifically, we pair pose sequences with motion-descriptive captions collected from our data pipeline, filtering out visual appearance descriptions with the Qwen3-VL model to retain only motion-relevant text. The pose encoder and a BERT [8] text encoder are jointly trained with a contrastive objective following previous practice [28, 56], to capture the motion semantics.

Stage 2: Cross-Modal Alignment with Proxy Tasks. With the pretrained pose encoder fixed, we train the MLP projector, the spatial binding attention module, and the embeddings of the newly introduced pose special tokens. Further, we introduce two proxy tasks to provide explicit supervision for pose-visual correspondence learning. 1) *Person Grounding*, we query the attributes of the target person corresponding to the input pose in a multi-person video; 2) *Body Part Discrimination*, which provides keypoint-derived bounding boxes and asks the model to identify the corresponding body region. Task details are provided in Appendix B.1.

Stage 3: Full Model Supervised Fine-Tuning (SFT). Finally, all Iris modules except the visual encoder are trained for SFT with the next-token-prediction loss on our collected video-pose-text instruction data. Note that we also include pose-free (video-text only) samples to maintain the MLLM's native pure visual ability. *When no pose data is available, Iris degrades to the pure vision MLLM with the same forward process of baseline Qwen-VL backbone.*

3.6 Automatic Data Collection Pipeline

Taming MLLMs involving a new pose modality requires massive, diverse training data covering multi-person, occlusion, and varied camera motion scenarios. Prior pose-aware models mostly rely on marker-based MoCap [6, 45, 18], which is costly, limited to controlled environments, and difficult to scale. We instead build a fully automatic pipeline on top of SAM-3D-Body [50, 49], a markerless estimator readily applicable to in-the-wild video. As illustrated in Figure 3, the pipeline proceeds through five stages: *video quality analysis and filtering*, *person detection and tracking initialization*, *mask propagation*, *heterogeneous pose estimation*, and *post-processing* for smoothing and quality checking. Please refer to Appendix A for full pipeline details.

Empowered by our automated data collection pipeline, we construct a large-scale instruction dataset comprising ∼550K samples, including video-pose-text triplets and vision-only QAs. To ensure context diversity, the data is curated from a combination of public datasets, including Molmo V2 [7], NExT-QA [46], Countix [10], CameraBench-train [24], and MotionBench-train [14], as well as an in-the-wild video corpus collected via our pipeline. Specifically, the training mixture consists of ∼150K vision-only QA samples to maintain native capabilities and ∼400K pose-paired QA samples covering diverse motion patterns and visual-grounded contexts. These pose-paired data include ∼80K proxy task samples for explicit spatial grounding to accelerate pose-vision alignment. Furthermore, we select ∼30K caption samples and filter the visual attributes of the original captions with Qwen to obtain the motion-only pose-text pair samples for contrastive training in Stage 1. More details can be found in Appendix A.

4 Experiments

4.1 Implementation Details

We build Iris upon Qwen2.5-VL-7B [2] and Qwen3-VL-8B [1], denoted Iris-7B and Iris-8B, respectively. The motion encoder consists of an 8-layer Transformer and a projection head. The bottleneck dimension of the pose-visual interaction is 1024. We uniformly sample 16 RGB frames and T_p =160 pose frames per video. After Qwen's internal temporal merging (merge size=2), the effective visual time steps are T_v =8, yielding a densification factor of T_{clip} =20. For training efficiency and memory constraints, the maximum number of input pose sequences (person identities) is capped at 3 per video. We select a set of 24 body surface keypoints covering the face, torso, and limbs as inputs. Test poses are extracted automatically without manual refinement (Appendix A.2).

Training proceeds in three stages, using one GPU for Stage 1 and 24 for Stages 2 and 3. We employ the AdamW optimizer with a cosine learning rate scheduler. **Stage 1** jointly trains the motion encoder and BERT via contrastive learning on ∼30K pose-caption pairs for 150 epochs with a batch size 512. **Stage 2** trains the MLP projector, spatial binding module, and pose special-token embeddings on ∼400K pose-video-text pairs for 2 epochs with learning rate 2×10^{-4}. **Stage 3** performs full model SFT on all ∼550K data for 1 epoch. Full details are provided in Appendix B.

4.2 Experiment Results and Analysis

We present the experimental results on fine-grained motion understanding, repetitive action counting, and general video understanding to verify the effectiveness of Iris.

1) Fine-Grained Motion Understanding. We evaluate Iris on MotionBench [14] (publicly available dev set with 4020 QAs) and FAVOR-Bench [42] (closed-ended evaluation part with 8184 QAs), two comprehensive benchmarks targeting fine-grained motion understanding. As shown in Table 1, Iris achieves state-of-the-art performance among MLLMs of similar volume. Specifically, Iris-Qwen2.5-VL-7B surpasses its pure-vision base model by **4.3** and **6.3** percentage points in overall accuracy on MotionBench and FAVOR Bench, respectively. Notably, scaling to the stronger Qwen3-VL-8B baseline yields even more substantial gains of **5.2** (57.1% → 62.3%) and **6.7** (51.1% → 57.8%) percentage points, demonstrating the remarkable motion understanding capacity of our Iris model.

For human-centric categories such as MR (Motion Recognition), RC (Repetitive Counting), and the aggregated OM, existing visual MLLMs often struggle due to the sparse and unstructured visual signals. By introducing the explicitly structured and high-frequency sampled pose modality, Iris effectively captures transient kinematics and temporally dense patterns. Furthermore, Iris exhibits remarkable superiority in the CM (Camera Motion) category. This is partly due to the camera-related training data for SFT. Moreover, our heterogeneous pose representation (Sec. 3.2) also contributes to the improvement by representing global pelvis translation separately from local human articulation. More discussions are in Appendix C.3.

2) Repetitive Action Counting. We evaluate repetitive action counting on the mainstream benchmarks, *i.e.*, RepCount [15] and UCFRep [55], using normalized Mean Absolute Error (MAE $= \frac{1}{N}\sum |\tilde{c}_i - c_i|/\tilde{c}_i$) and Off-by-One accuracy (OBO $= \frac{1}{N}\sum[|\tilde{c}_i - c_i| \leq 1]$) as metrics, where $\tilde{c}_i$ and c_i denote the ground truth and predicted count respectively. For fair comparison, all models are evaluated under the same setting with 16-frame RGB input, whereas our Iris models inherently leverage the additional dense pose sequence data without introducing much computational overhead.

As shown in Table 2, Iris demonstrates a significant improvement on temporal counting tasks, which most generalist MLLMs struggle to handle. Specifically, Iris-Qwen2.5-VL-7B achieves substantial gains over its base model, reducing MAE by 0.695 and improving OBO by 20.7 percentage points on RepCount. More remarkably, when built upon the stronger Qwen3-VL-8B baseline, Iris yields more superior improvements, significantly refreshing the temporal counting capacity of the baseline model. We attribute these unprecedented gains directly to our asymmetric sampling strategy. Repetitive counting requires precise perception of motion periodicity and boundaries, which are inevitably destroyed when a video is aggressively temporally downsampled, *e.g.*, 16 frames or 1fps. By injecting densely sampled pose sequences, Iris effectively reconstructs the fine-grained temporal dynamics.

Table 1: Comparison with the vision MLLMs and motion-specific MLLMs on MotionBench and FAVOR Bench. OM denotes the accuracy of all FAVOR Bench categories except CM. † denotes additional pose data input. * denotes additional augmented visual inputs based on question categories.

Model	Frames	MotionBench (%)							FAVOR Bench (%)		
		Overall	MR	LM	CM	MO	AO	RC	Overall	OM	CM
Proprietary MLLMs											
Gemini 1.5 Pro [40]	1fps	51.0	51.0	52.0	54.0	67.0	40.0	22.0	49.9	51.2	41.6
GPT-4o [17]	1fps	-	-	-	-	-	-	-	42.1	43.0	36.0
Open-source MLLMs											
Oryx-34B [26]	64	49.0	48.0	52.0	44.0	65.0	42.0	32.0	-	-	-
Qwen2.5-VL-7B [2]	16	56.2	61.5	60.1	45.7	73.2	41.8	29.8	47.8	49.9	34.3
Qwen3-VL-8B [1]	16	57.1	62.0	60.3	57.4	73.2	42.0	25.5	51.1	52.3	43.1
MiMoVL-SFT-7B [39]	16	55.6	61.4	59.0	54.0	69.4	40.7	27.0	48.2	49.6	39.1
InternVL3.5-8B [44]	16	56.5	62.6	60.8	47.3	72.5	44.5	24.8	51.3	53.8	34.7
TE-Fusion-9B [14]	16	58.0	64.0	59.0	51.0	69.0	41.0	**39.0**	-	-	-
MotionSight-7B [9]	1fps*	55.6	59.7	58.1	48.3	73.6	40.1	33.5	45.1	46.2	38.1
MotionChat-7B [9]	16	54.8	58.3	58.8	49.9	71.3	42.4	29.0	45.7	47.3	35.0
MotionLLM-7B [6]	8	32.4	33.4	34.6	26.8	33.0	40.0	20.3	38.8	39.0	37.5
Iris-Qwen2.5-VL-7B	16†	**60.5**	**63.9**	**63.4**	**63.6**	**74.6**	43.2	38.5	**54.1**	**55.0**	**47.6**
Iris-Qwen3-VL-8B	16†	**62.3**	**66.4**	**64.5**	**67.0**	**74.9**	**47.0**	37.5	**57.8**	**58.5**	**53.2**

Table 2: Comparison of repetitive action counting task.

Model	RepCount		UCFRep	
	MAE ↓	OBO ↑	MAE ↓	OBO ↑
Qwen2.5-VL-7B	1.115	17.3%	1.099	28.2%
Qwen3-VL-8B	0.664	20.0%	0.696	25.2%
MiMoVL-SFT-7B	0.751	22.0%	0.877	25.0%
InternVL3.5-8B	0.928	13.3%	1.241	18.5%
MotionChat-7B	1.306	10.1%	3.544	17.5%
Iris-Qwen2.5-VL-7B	**0.420**	**38.0%**	**0.466**	**43.7%**
Iris-Qwen3-VL-8B	**0.346**	**50.0%**	**0.344**	**62.1%**

Table 3: Results on MVBench **without pose data input**. Accuracy on Kinematic & Motion (K&M) subset and Context & Appearance (C&A) subset is reported.

Model	MVBench (%)		
	Overall	K&M	C&A
Qwen2.5-VL-7B	63.3	62.4	64.3
Iris-7B	**66.1**	**64.8**	**67.4**
Qwen3-VL-8B	64.5	61.4	**67.6**
Iris-8B	**66.3**	**65.2**	67.4

3) General Video Understanding. We evaluate Iris on MVBench [21] **without pose data input during inference** to verify that our model Iris preserves the pure-vision-based general video understanding capacity. For fine-grained analysis, we conceptually group the benchmark into *Kinematic & Motion* and *Context & Appearance* super-categories. As shown in Table 3, Iris maintains consistent overall performance with the strong pure-vision baselines. Beyond the expected improvements in motion-centric tasks, Iris robustly retains knowledge in the *Context & Appearance* category. For instance, Iris-Qwen3-VL-8B yields a highly comparable score of 67.4% (vs. 67.6%) on context-centric tasks. This 0.2-percentage-point gap is consistent with comparable context and appearance performance after learning explicit human kinematics.

4.3 Ablation Study

Unless otherwise specified, ablations use Qwen2.5-VL-7B; the RGB frame scaling experiment uses Qwen3-VL-8B.

1) Motivation: Pose-Augmented MLLM Framework. We decouple the architecture gains step-by-step to present a comprehensive analysis in Table 4:

- **Data Efficacy:** Fine-tuning solely on collected RGB videos (*RGB-SFT*) improves the baseline, validating the effectiveness of our curated data. Yet, sparse visual signals fundamentally bottleneck fine-grained motion understanding. Therefore, we introduce our Iris model.
- **Pose-Integrated Training:** *Iris* utilizes the pose-visual data for cross-modal learning. First, we test the model version using *Test RGB Only* during inference, which yields a leap over RGB-SFT (*e.g.*, 57.9% → 59.6% on MotionBench). This demonstrates that pose data effectively enhances the visual representation learning during training. However, temporal counting is still inherently challenging with the sparse visual inputs during inference.

Table 4: Key ablation study of pose data. V denotes visual (RGB) modality, and P denotes pose modality. MR: Motion Recognition; CM: Camera Motion; HAC: Holistic Action Classification.

Method	Train / Infer. Modality	MotionBench		FAVOR Bench		UCFRep	
		Overall	MR / CM	Overall	HAC / CM	MAE ↓	OBO ↑
Qwen2.5-VL-7B	V / V	56.2	61.5 / 45.7	47.8	54.8 / 34.3	1.0989	28.16%
Qwen2.5-VL-RGB-SFT	V / V	57.9	60.6 / 61.8	51.7	55.5 / 46.0	1.0717	33.01%
Iris (Test RGB Only)	V+P / V	59.6	62.7 / 62.8	53.2	55.6 / 45.9	1.1026	34.95%
Iris (RGB+Pose)	V+P / V+P	**60.5**	**63.9 / 63.6**	**54.1**	**58.0 / 47.6**	**0.4656**	**43.69%**

Table 5: Ablation on different pose injection methods. TB: temporal binding; SB: spatial binding.

Pose Injection Method	MotionBench	FAVOR Bench
Pose Position Texts	55.8	47.5
Pose Heatmap	57.8	50.9
Motion Encoder	58.4	52.0
Motion Encoder+TB	59.7	53.0
Motion Encoder+TB,SB	**60.5**	**54.1**

- **Pose-Integrated Training and Inference:** Providing full *RGB+Pose* in inference achieves the best scores. Explicit dense priors enhance kinematics as well as camera motion understanding. Crucially, it significantly improves action counting by providing temporally dense motion information.

2) Ablation on Different Pose Injection Methods. In Table 5, we compare three pose injection strategies. *Pose Position Texts* directly serializes 2D joint coordinates as text tokens fed into the LLM, which yields the weakest performance as the LLM struggles to parse raw numerical coordinates into motion semantics. *Pose Heatmap* renders 2D keypoints as heatmaps and processes them through the vision encoder, improving over the text baseline but introducing significant additional tokens and remaining bottlenecked by sparse visual signals. *Motion Encoder* tokenizes the pose sequence with our dedicated encoder, achieving further gains by preserving explicit kinematic structure. Building upon this, we further develop spatiotemporal binding mechanisms, which are discussed below.

3) Effect of Spatiotemporal Binding. As shown in Table 5, both binding mechanisms contribute positively to the Motion Encoder baseline, which concatenates learned pose embeddings with the visual and textual inputs without explicit correspondence binding. Temporal binding alone brings consistent improvement on MotionBench and FAVOR Bench, demonstrating that aligning pose and visual tokens in temporal RoPE position space is critical for MLLM to establish cross-modal temporal correspondence. Spatial binding further improves performance by grounding each person's pose tokens to their corresponding visual region, yielding the best scores.

4) Dense Pose vs. More RGB Frames. Table 6 compares Iris (16 RGB and 160 pose frames) with Qwen3-VL-8B using 16–160 RGB frames under the same inference settings. Increasing RGB input from 16 to 160 frames raises FLOPs by $9.7\times$ and latency by $1.8\times$, while MotionBench accuracy increases from 57.1% to 59.7% and RepCount OBO remains below 0.26. Iris achieves 62.3% and 0.500, respectively, at nearly the MLLM inference cost of the 16-frame baseline (2.883 vs. 2.866 s; 24.67 vs. 24.19 TFLOPs). These results support dense structured pose as an efficient complement to sparse RGB input. The motion encoder has 74M parameters; the pose extraction procedure is described in Appendix A.

5) Qualitative Analysis. Figure 4 illustrates two representative cases comparing Iris with the baseline. In the top repetitive counting case, the baseline severely undercounts the fast-paced pommel horse swings due to the temporal information loss from sparse RGB sampling. In contrast, empowered by the densely sampled pose stream, Iris accurately captures the high-frequency periodic motion, yielding a precise count. In the bottom fine-grained action sequence task, the baseline struggles to correctly identify the temporal order of complex body-object interactions, while Iris successfully parses the detailed chronological breakdown of the actions, validating the effectiveness of our model.

More analysis and results can be found in Appendix C.

Table 6: RGB frame scaling and MLLM inference cost on a single GPU with 512px inputs, KV cache, FlashAttention, and 64 generated tokens. Pose extraction is excluded. MotionBench accuracy is in %; RepCount OBO is a fraction.

Method	Frames		Tokens	Time (s)	TFLOPs	Motion Bench ↑	RepCount	
	RGB	Pose					OBO ↑	MAE ↓
Qwen3-VL-8B	16	–	1309	2.866	24.19	57.1	0.200	0.664
Qwen3-VL-8B	32	–	2576	3.074	47.66	58.7	0.213	0.897
Qwen3-VL-8B	64	–	5109	3.552	94.66	59.3	0.253	0.676
Qwen3-VL-8B	160	–	12179	5.145	235.67	59.7	0.247	0.752
Iris-8B	16	160	1419	2.883	24.67	**62.3**	**0.500**	**0.346**

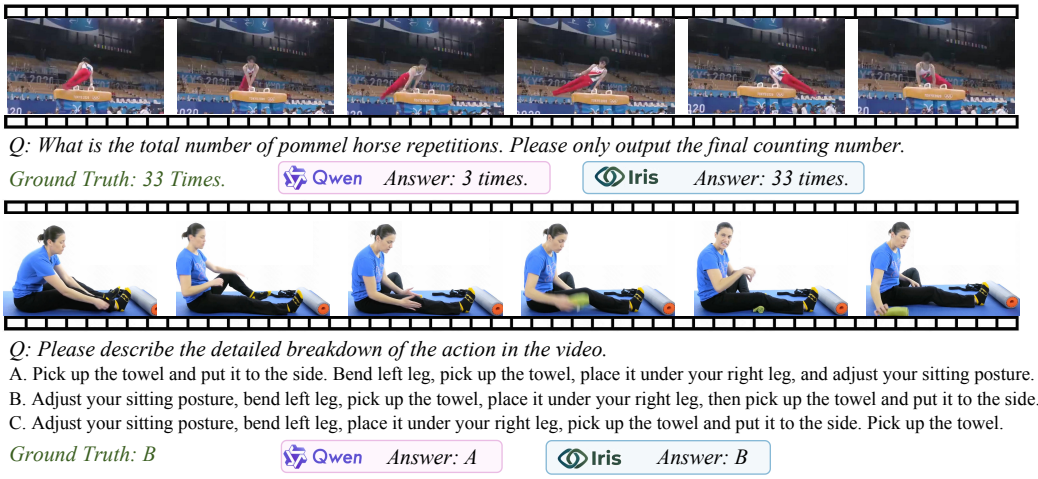


Figure 4: Qualitative examples. Qwen denotes the baseline while Iris denotes our model.

5 Conclusion

In this paper, we introduce Iris, a novel pose-augmented video MLLM for human-centric video understanding. By integrating high-frequency, lightweight, and structured human pose sequences through a spatiotemporal cross-modal binding mechanism, which consists of a pose-structured 3D RoPE for temporal alignment and a person-grounded cross-attention for spatial grounding, Iris effectively captures dense kinematics and decouples complex motion dynamics. Extensive experiments demonstrate that Iris achieves state-of-the-art performance on fine-grained motion understanding benchmarks, especially in repetitive action counting, while robustly preserving its general video understanding capabilities. We believe Iris paves a promising and efficient path for injecting structured kinematic priors into generalist MLLMs.

Acknowledgments and Disclosure of Funding

This work was supported by the Beijing Major Science and Technology Project under Contract No. Z251100008425023.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025.

[3] Peijun Bao, Zihao Shao, Wenhan Yang, Boon Poh Ng, and Alex C Kot. E3M: Zero-shot spatio-temporal video grounding with expectation-maximization multimodal modulation. In *ECCV*, pages 227–243, 2024.

[4] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. TemporalBench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

[5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[6] Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. MotionLLM: Understanding human behaviors from human motions and videos. *arXiv preprint arXiv:2405.20340*, 2024.

[7] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

[9] Yipeng Du, Tiehan Fan, Kepan Nan, Rui Xie, Penghao Zhou, Xiang Li, Jian Yang, Zhenheng Yang, and Ying Tai. MotionSight: Boosting fine-grained motion understanding in multimodal llms. In *ICLR*, 2026.

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *IEEE CVPR*, pages 10387–10396, 2020.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[12] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J Black. ChatPose: Chatting about 3d human pose. In *IEEE CVPR*, pages 2093–2103, 2024.

[13] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *IEEE CVPR*, 2022.

[14] Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. MotionBench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *IEEE CVPR*, pages 8450–8460, 2025.

[15] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. TransRAC: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *IEEE CVPR*, pages 19013–19022, 2022.

[16] Lei Hu, Yongjing Ye, and Shihong Xia. HMVLM:human motion-vision-language model via MoE LoRA. In *NeurIPS*, 2025.

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[18] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as a foreign language. In *NeurIPS*, 2023.

[19] Bozhou Li, Jiahang Zhang, Yue Ding, Yushuo Guan, Bohan Zeng, Yiyan Ji, Xinlong Chen, Yang Shi, Yifan Dai, Yuran Wang, Chengzhuo Tong, Pengfei Wan, Yuanxing Zhang, and Wentao Zhang. Human-centric image captioning with subject-centered spatial understanding. In *ACM MM*, 2026.

[20] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[21] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *IEEE CVPR*, pages 22195–22206, 2024.

[22] Lei Li, Sen Jia, Jianhao Wang, Zhongyu Jiang, Feng Zhou, Ju Dai, Tianfang Zhang, Zongkai Wu, and Jenq-Neng Hwang. Human motion instruction tuning. *arXiv preprint arXiv:2411.16805*, 2025.

[23] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. *arXiv preprint*, arXiv:2311.10122, 2023.

[24] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, et al. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025.

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pages 38–55, 2024.

[26] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-demand spatial-temporal understanding at arbitrary resolution. In *ICLR*, 2025.

[27] Peng Lu, Tao Jiang, Yining Li, Xiangtai Li, Kai Chen, and Wenming Yang. RTMO: Towards high-performance one-stage real-time multi-person pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1491–1500, 2024.

[28] Guangtao Lyu, Chenghao Xu, Jiexi Yan, Muli Yang, and Cheng Deng. Towards unified human motion-language understanding via sparse interpretable characterization. In *ICLR*, 2025.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. 2018.

[30] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016.

[31] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. BABEL: Bodies, action and behavior with english labels. In *IEEE CVPR*, 2021.

[32] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *ICML*, 2024.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2023.

[34] Dominick Reilly, Rajatsubhra Chakraborty, Arkaprava Sinha, Manish Kumar Govind, Pu Wang, Francois Bremond, Le Xue, and Srijan Das. LLAVIDAL: A large language vision model for daily activities of living. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24297–24308, 2025.

[35] Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, et al. YOLO advances to its genesis: A decadal and comprehensive review of the you only look once (YOLO) series. *Artificial Intelligence Review*, 58(9):274, 2025.

[36] Yang Shi, Jiaheng Liu, Yushuo Guan, Zhenhua Wu, Yuanxing Zhang, Zihao Wang, Weihong Lin, Jingyun Hua, Zekun Wang, Xinlong Chen, Bohan Zeng, Wentao Zhang, Fuzheng Zhang, Wenjing Yang, and Di Zhang. Mavors: Multi-granularity video representation for multimodal large language model. arXiv preprint arXiv:2504.10068, 2025.

[37] Arkaprava Sinha, Dominick Reilly, Francois Bremond, Pu Wang, and Srijan Das. SKI models: Skeleton induced vision-language embeddings for understanding activities of daily living. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(7):6931–6939, 2025.

[38] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *IEEE TCSVT*, 2025.

[39] Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi,

Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. MiMo-vl technical report, 2025.

[40] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[41] Michal Tölgyessy, Martin Dekan, L'uboš Chovanec, and Peter Hubinskỳ. Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, 21(2):413, 2021.

[42] Chongjun Tu, Lin Zhang, Pengtao Chen, Peng Ye, Xianfang Zeng, Wei Cheng, Gang Yu, and Tao Chen. FAVOR-Bench: A comprehensive benchmark for fine-grained video motion understanding. In *NeurIPS*, 2025.

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[44] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025.

[45] Bizhu Wu, Jinheng Xie, Keming Shen, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. Mg-motionllm: A unified framework for motion comprehension and generation across multiple granularities. In *IEEE CVPR*, 2025.

[46] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *IEEE CVPR*, pages 9777–9786, 2021.

[47] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. SlowFast-LLaVA: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

[48] Mingze Xu, Mingfei Gao, Shiyu Li, Jiasen Lu, Zhe Gan, Zhengfeng Lai, Meng Cao, Kai Kang, Yinfei Yang, and Afshin Dehghan. SlowFast-LLaVA-1.5: A family of token-efficient video large language models for long-form video understanding. In *Conference on Language Modeling*, 2025.

[49] Timing Yang, Sicheng He, Hongyi Jing, Jiawei Yang, Zhijian Liu, Chuhang Zou, and Yue Wang. Fast SAM 3D Body: Accelerating sam 3d body for real-time full-body human mesh recovery. *arXiv preprint arXiv:2603.15603*, 2026.

[50] Xitong Yang, Devansh Kukreja, Don Pinkus, Anushka Sagar, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jiawei Liu, Nicolas Ugrinovic, et al. SAM 3D Body: Robust full-body human mesh recovery. *arXiv preprint arXiv:2602.15989*, 2026.

[51] En Yu, Liang Zhao, Yana Wei, Jinrong Yang, Dongming Wu, Lingyu Kong, Haoran Wei, Tiancai Wang, Zheng Ge, Xiangyu Zhang, et al. Merlin: Empowering multimodal LLMs with foresight minds. In *ECCV*, pages 425–443, 2024.

[52] Qing Yu, Mikihiro Tanaka, and Kent Fujiwara. Exploring vision transformers for 3D human motion-language models with motion patches. In *IEEE CVPR*, 2024.

[53] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Empirical Methods in Natural Language Processing: System demonstrations*, 2023.

[54] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMa: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, pages 543–553, 2023.

[55] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *IEEE CVPR*, pages 670–678, 2020.

[56] Jiahang Zhang, Lilang Lin, Shuai Yang, and Jiaying Liu. SGAR: Structural generative augmentation for 3d human motion retrieval. In *NeurIPS*, 2025.

[57] Jianrui Zhang, Mu Cai, and Yong Jae Lee. VinoGround: Scrutinizing LMMs over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*, 2024.

[58] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect our contribution.

   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: Please see the Limitations and Future Work section (Appendix D).

   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: The paper does not include theoretical results.

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: Please see the Experiments Section 4 in main paper and Implementation Details in Appendix 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: [No]

Justification: Code and model checkpoints will be released upon publication. Extensive implementation details are provided in the main paper and appendix to facilitate reproducibility.

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: Please see the Experiments Section 4.1 in main paper and Implementation Details in Appendix 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: Due to limited resources and following the practice of previous works, we do not report the error bars.

   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: Please see this in Implementation Details section in Appendix B.

   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 conducted in the paper conforms with the NeurIPS Code of Ethics.

   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: Please see the Broader Impact section (Appendix E) in the Appendix.

    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 poses no such risks.

    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: This work properly credits and respects the assets used.

    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: [Yes]

    Justification: Please see the Method and Experiments (Sec. 3 4) in Main Paper.

    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: [N/A]

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

    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: LLMs are used as model backbones and for automated video analysis, caption filtering, and data checking, as described in Sec. 3 and Appendix A.

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.

A Data Collection Pipeline **22**

A.1 Training-Data Curation Pipeline 22

A.2 Automatic Test-Time Pose Processing 24

A.3 Pose Estimation Details 24

A.4 Pose Data Visualization 25

B Iris Implementation Details **26**

B.1 Proxy Tasks for Efficient Pose-Vision Alignment 26

B.2 Spatial Binding Module Details 27

B.3 Training Configurations 27

C More Experimental Results **28**

C.1 Robustness to Synthetic Pose Perturbations 28

C.2 Preliminary Observation on Left-Right Discrimination 28

C.3 Heterogeneous Pose Representation 28

C.4 Densification Factor (Pose Sequence Length) 29

C.5 Spatial Binding Implementation 29

C.6 Qualitative Results 29

D Limitations and Future Work **31**

E Broader Impact **31**

We present more details of our method, experiments, and implementation below.

A Data Collection Pipeline

A.1 Training-Data Curation Pipeline

For training-data construction, the automated pipeline converts raw in-the-wild videos into video-pose data through five serial stages, as shown in Figure 5. MLLM-based metadata analysis and error checking are offline, training-only quality-control steps. The test-time procedure is described separately in Appendix A.2.

Human Video Analysis and Input Parsing. We utilize Qwen3-VL-32B-Instruct [1] to analyze each video clip as preliminary work. The model outputs structured metadata, including foreground person count per sampled frame, occlusion degree, overall visual quality assessment, and torso visibility. This metadata serves two purposes: (1) filtering out low-quality clips, *e.g.*, heavy torso occlusion and small person, before the expensive pose estimation stage, and (2) providing a target person count and a recommended initialization frame for subsequent tracking. Concretely, for single-person clips, the pipeline selects the earliest sampled frame where exactly the target foreground person is detected, while for multi-person clips, it identifies the frame with the target foreground count to avoid initialization on crowded or partially visible frames.

Person Detection and Tracking Initialization. Given the recommended initialization frame, we run GroundingDINO [25] with the generic prompt "person" (box threshold 0.30, text threshold 0.25) to detect all person instances across the sampled clip (maximum 160 frames). Detections are ranked by confidence score, and the top-K boxes (where K equals the target person count and the maximum of K is set to 3) are used to initialize the tracking state of SAM3 [5], with each detection assigned a unique persistent identity. When no recommended frame is available, we fall back to scanning

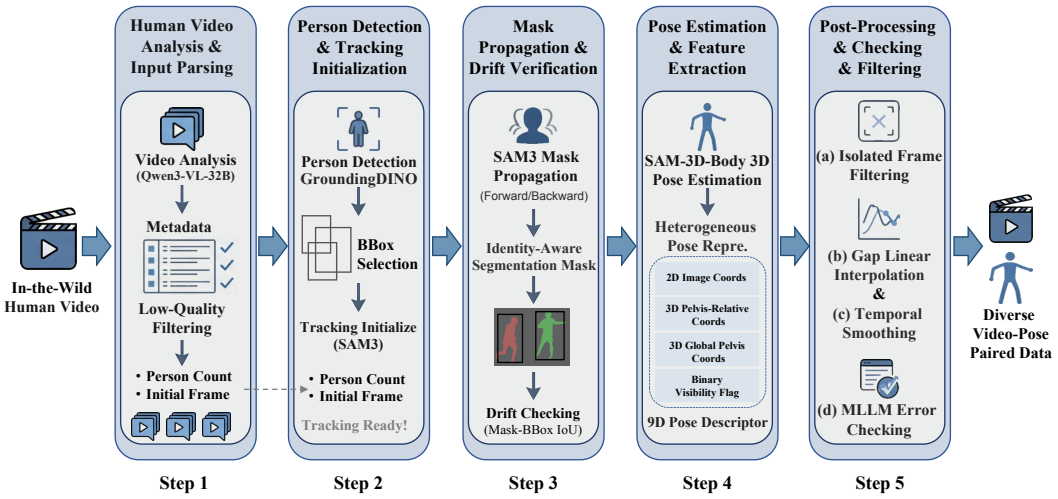


Figure 5: Overview of our pose data collection pipeline.

the video at stride$=8$ intervals to find the frame yielding the highest detection confidence sum that satisfies the target count.

Mask Propagation and Drift Verification. From the initialization frame, SAM3 propagates instance masks both forward and backward through the sampled clip, producing per-frame identity-aware segmentation maps stored as palette-indexed images where each pixel value encodes the corresponding person's identity. To ensure tracking quality, we apply a drift verification step. Concretely, at every 50 frames, the bounding box derived from the propagated mask is compared via IoU against fresh GroundingDINO detections on that frame. Any tracked identity whose average IoU falls below the threshold is discarded before proceeding to pose estimation.

Pose Estimation and Feature Extraction. For each retained identity, we run SAM-3D-Body [50] frame-by-frame conditioned on the propagated instance mask, enabling person-level disambiguation under occlusion and multi-person conditions. From the output of each frame, we construct the 9-dimensional per-keypoint descriptor defined in Eq. 1 of the main paper. Specifically, 2D normalized image coordinates (u_k, v_k) are obtained by dividing pixel coordinates by the frame resolution; 3D pelvis-relative joint coordinates (x_k, y_k, z_k) are computed by subtracting the hip-center offset (mean of left and right hip keypoints) from the raw 3D predictions; the global pelvis position (x'_0, y'_0, z'_0) is recovered as the camera-space translation of the hip center; and the binary visibility flag $c_k \in \{0, 1\}$ is determined by checking whether a keypoint's pixel location falls within the corresponding identity's segmentation mask.

Post-processing and Filtering. Raw pose sequences are post-processed via four sequential refinement steps. (a) Isolated frame filtering: any valid frame whose temporal neighborhood contains no other valid frames is marked invalid, suppressing spurious single-frame detection noise. (b) Gap interpolation: contiguous invalid segments of length ≤ 3 frames are filled via linear interpolation of pose coordinates and global translation from the surrounding valid boundary frames. (c) Temporal smoothing: an exponential moving average with α=0.7 is applied independently to joint coordinates and global translation to suppress trajectory jitter, operating only on joints that are visible in both the current and the previous frame. At the same time, samples with temporal jitter exceeding the threshold will be directly discarded. (d) MLLM checking: finally, we plot the keypoints on the frame, which are fed into the Qwen3-VL 32B to check if the estimated pose exhibits significant errors, *e.g.*, pose estimation on animals, persons in photos or posters, too small persons, or pose estimation failures due to heavy occlusion. In training-data construction, we set a relatively strict standard to ensure the quality of the filtered pose data. The overall training pose rejection rate is approximately 14%.



Figure 6: Example pose data obtained by our video pose estimation pipeline.

A.2 Automatic Test-Time Pose Processing

Test-time poses are extracted automatically from the input video, without manual refinement. Person presence and count are determined from detector confidence, and detection and tracking are initialized without MLLM-derived metadata. The estimated poses undergo only fast, deterministic processing, including jitter filtering with a fixed threshold, in a single automatic script. The offline MLLM metadata analysis and error checking used for training-data curation are disabled at test time. No manual pose correction or offline selection of test examples is performed; when no usable pose sequence remains, the video is evaluated with RGB-only input. Thus, benchmark results use the automatically produced test inputs throughout. The MLLM inference timings in Table 6 exclude pose extraction.

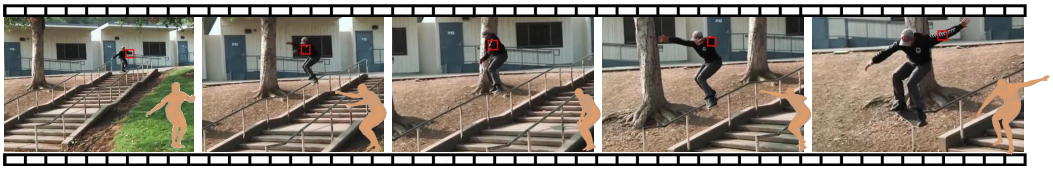
A.3 Pose Estimation Details

Rather than using the full body mesh or all available keypoints from SAM-3D-Body, we select a compact set of 24 semantically meaningful keypoints covering the face, torso, and limbs, as listed in Table 7. This selection is designed to be compatible with standard pose estimation conventions as much as possible for transferrability. Notably, we deliberately exclude the majority of hand finger joints despite SAM-3D-Body supporting detailed hand mesh recovery. Hand keypoints tend



Q: *What is the appearance of the person highlighted by the input pose?* **Person Grounding Proxy Task**

A: *The person is a young boy with short black hair, wearing a light blue button-up jacket over a black graphic t-shirt and black pants.*



Q: *Which body part is marked by the red box?* **Body Part Discrimination Proxy Task**

A: *Left shoulder.*

Figure 7: Example QA data of our constructed proxy tasks. The top row is the person grounding task, and the bottom row is the body part discrimination task.

to exhibit substantially higher estimation error due to frequent self-occlusion and fast motion, and their inclusion would also require activating the hand decoder, significantly increasing per-frame inference cost. Furthermore, since we only require 3D keypoint positions and 2D projections rather than surface meshes, we skip the hand decoder entirely and bypass the subsequent MHR-to-SMPL topology conversion, retaining only the body decoder output and the 3D keypoints derived from it via forward kinematics. This reduces both memory and per-frame inference cost without affecting the pose information used downstream.

Table 7: Selected 24 keypoints used in Iris pose representation.

Index	Keypoint	Index	Keypoint
0	Nose	12	Left Hand Tip
1	Left Eye	13	Right Hand Tip
2	Right Eye	14	Left Hip
3	Left Ear	15	Right Hip
4	Right Ear	16	Left Knee
5	Neck	17	Right Knee
6	Left Shoulder	18	Left Ankle
7	Right Shoulder	19	Right Ankle
8	Left Elbow	20	Left Heel
9	Right Elbow	21	Right Heel
10	Left Wrist	22	Left Big Toe
11	Right Wrist	23	Right Big Toe

Pose extraction is an additional preprocessing step for joint RGB-pose inference and is excluded from the MLLM inference measurements in Table 6. The framework is also compatible with lightweight pose estimators [35, 27] and directly available depth-camera keypoints [41].

A.4 Pose Data Visualization

We present some pose estimation results by our pipeline for visualization in Figure 6. Thanks to the robust performance of SAM-3D-Body [50] and our carefully designed video pose estimation pipeline, we can achieve satisfactory pose results in most scenarios. Failure cases often occur in situations with extreme occlusion. At this time, the pose data will undergo severe fluctuations and will be discarded in our post-processing filtering step.

B Iris Implementation Details

B.1 Proxy Tasks for Efficient Pose-Vision Alignment

To provide explicit supervision for fast pose-vision correspondence learning in Stage 2, we design two lightweight proxy tasks that directly target the spatial grounding capability of the spatial binding module as presented in Figure 7.

Person Grounding. In ***multi-person*** videos, the model must associate a given pose sequence with the correct individual among multiple coexisting subjects. To supervise this grounding capacity, we construct QA pairs where the pose sequence of a specific person identity is provided as input, and the model is asked to answer questions that require identifying the corresponding individual in the visual stream. The questions cover two aspects: (1) *appearance grounding*, asking the model to describe the visual attributes (*e.g.*, clothing, hair, accessories) of the person whose motion matches the input pose; and (2) *action grounding*, asking the model to describe what the target person is doing at a specific moment in the video. Since these questions can only be correctly answered by localizing the pose to the right person in the frame, they provide direct supervision for the person-grounded cross-attention module to learn pose-to-visual region correspondence in multi-person scenarios.

Body Part Discrimination. While coarse body-level grounding can be established relatively easily, fine-grained left-right discrimination is inherently ambiguous in 2D visual space due to viewpoint variation, body symmetry, and self-occlusion. To address this, we construct a body part discrimination task that specifically targets laterality understanding. Concretely, for each video frame, we derive tight bounding boxes around individual body parts (*e.g.*, left shoulder, right knee) directly from the 2D keypoint coordinates in our heterogeneous pose representation. These keypoint-derived bounding boxes are rendered as visual overlays on the frame, and the model is asked to identify which specific body part or side (*e.g.*, "left" vs. "right") is highlighted by the box. This task forces the model to jointly reason over the structural pose layout and the visual appearance within the highlighted region, supervising part-level semantic alignment and particularly strengthening the model's ability to resolve left-right ambiguity that pure visual reasoning struggles with.

Table 8: Training configuration across three stages. The LLM row refers to pretrained backbone weights; pose special-token embeddings are listed separately. Temporal binding has no new learnable parameters.

Configuration	**Stage 1** *Contrastive Pretrain*	**Stage 2** *Cross-Modal Alignment*	**Stage 3** *Full SFT*
Trainable Modules			
Pose Encoder	✓	×	✓
BERT / Contrastive Heads	✓	–	–
MLP Projector	×	✓	✓
Spatial Binding	×	✓	✓
LLM	×	×	✓
Pose Special-Token Embeddings	–	✓	✓
Vision Encoder	×	×	×
Optimizer & Schedule			
Optimizer	AdamW	AdamW	AdamW
LR (pose encoder)	1×10^{-4}	–	2×10^{-5}
LR (BERT)	1×10^{-5}	–	–
LR (contrastive heads)	3×10^{-4}	–	–
LR (projector / interact)	–	2×10^{-4}	3×10^{-5}
LR (LLM)	–	–	1×10^{-5}
LR Schedule	Cosine + Warmup	Cosine + Warmup	Cosine + Warmup
Warmup Ratio	5%	5%	5%
Weight Decay	0.01	0.01	0.01
Data & Compute			
# GPUs	1	24	24
Batch Size (global)	512	288	288
Epochs	150	2	1

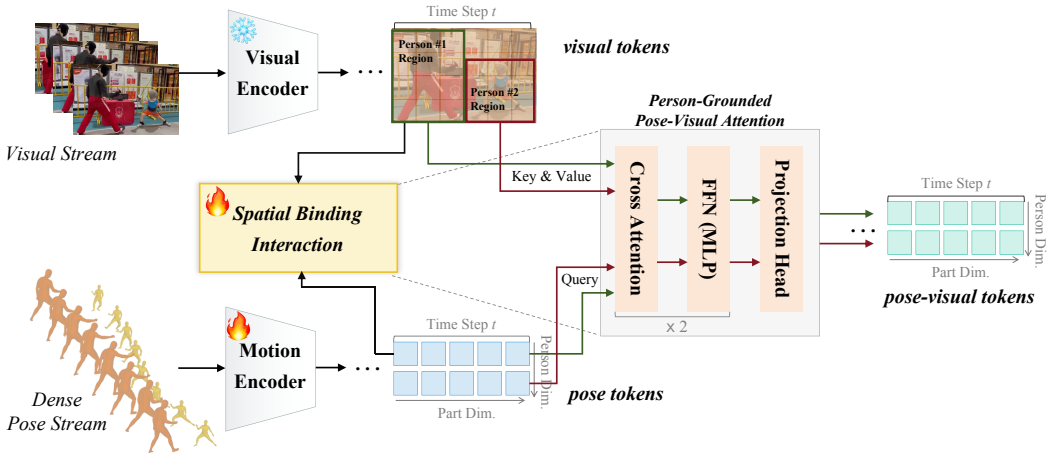


Figure 8: Illustration of the spatial binding module.

B.2 Spatial Binding Module Details

We present an illustration of the spatial binding module in Figure 8 for better understanding. As shown, for each time step t, the visual encoder produces a set of visual tokens covering the full frame, from which person-specific regions are derived using bounding boxes obtained from the 2D keypoint coordinates. Simultaneously, the motion encoder produces pose tokens of shape $[T_v, P, N_p, D]$, where P is the number of persons and N_p=5 is the number of semantic body part tokens per person. For each person, the corresponding pose tokens serve as *Query*, while the visual tokens within the person's bounding box region serve as *Key* and *Value* in a masked cross-attention operation. This person-grounded cross-attention is followed by an FFN, repeated for L=2 layers, and a projection head, producing the pose-visual tokens x_{pv} of the same shape $[T_v, P, N_p, D]$. Crucially, the final projection head is zero-initialized, ensuring that x_{pv} contributes negligibly at the start of training and activates gradually, preserving the pretrained MLLM's stability. The resulting pose-visual tokens are concatenated with the original pose tokens along the part dimension, yielding $2N_p = 10$ pose-related tokens per person and effective visual time step, and fed into the MLLM together with the visual and textual tokens.

B.3 Training Configurations

We describe the detailed training configuration for each of the three stages, and Table 8 gives a summary.

Stage 1: Contrastive Pretraining of Pose Encoder. The pose encoder and BERT [8] text encoder are jointly trained on a single GPU using a CLIP-style contrastive objective following the previous work [28, 56]. We use the AdamW optimizer with differentiated learning rates: 1×10^{-5} for the text encoder, 1×10^{-4} for the pose encoder, and 3×10^{-4} for the contrastive projection heads, with weight decay 0.01. Training runs for 150 epochs with a batch size of 512, a cosine learning rate schedule with 5% linear warmup, and gradient clipping at $\max_norm = 1.0$. The InfoNCE [29] loss is adopted as the objective for training.

Stage 2: Cross-Modal Alignment. With the pose encoder frozen, we train the MLP projector and the spatial binding cross-attention module, while keeping the pretrained LLM backbone weights and vision encoder frozen. Only the embeddings of the three newly added pose-related special tokens (<|pose_start|>, <|pose_end|>, <|pose_token|>) receive gradient updates via a gradient hook that zeroes out gradients for all pre-existing vocabulary entries. Training is conducted on 3 nodes $\times$ 8 GPUs (24 GPUs total) with DeepSpeed ZeRO optimization. We use a per-device batch size of 12 with no gradient accumulation (effective global batch size 288), a learning rate of 2×10^{-4}, cosine decay with 5% warmup, and train for 2 epochs.

Table 9: Inference-only synthetic pose perturbations with Qwen3-VL-8B as the RGB-only baseline. Gaussian σ is measured on the normalized coordinate scale (local 3D standard deviation ≈ 0.16). Block-drop zeros consecutive pose frames. MotionBench accuracy is in %; RepCount reports normalized MAE.

Setting / Perturbation	**MotionBench ↑**	**RepCount MAE ↓**
Qwen3-VL-8B (RGB only)	57.1	0.664
Iris: estimator noise only	62.3	0.346
Iris: $\sigma = 0.05$	62.3	0.349
Iris: $\sigma = 0.05$ + 8-frame block-drop	62.0	0.366
Iris: $\sigma = 0.15$ + 32-frame block-drop (20%)	61.8	0.379

Table 10: Ablation study on the heterogeneous pose representation design.

Method	**MotionBench**		**FAVOR Bench**	
	Overall	Camera Motion	Overall	Camera Motion
2D Coordinates	59.2	62.1	52.8	45.9
3D Positions	59.8	62.3	53.3	46.3
Ours 9D Pose Descriptor	**60.5**	**63.6**	**54.1**	**47.6**

Stage 3: Full Model Supervised Fine-Tuning. All Iris parameters except the visual encoder are unfrozen for full SFT with the next-token-prediction objective. We apply a differentiated learning rate scheme: the interaction module, projector, and person embeddings are trained at 3×10^{-5}; the pose encoder at 2×10^{-5}; and the LLM parameters at the base rate of 1×10^{-5}. The same 24-GPU setup is used with DeepSpeed ZeRO, per-device batch size 12, cosine decay with 5% warmup, and training for 1 epoch over the full mixed dataset.

The spatial binding module uses L=2 cross-attention layers with a bottleneck dimension of 1024. We uniformly sample 16 RGB frames and T_p=160 pose frames per video clip. Temporal merging of each pair of RGB frames yields T_v=8 effective visual time steps, so $T_{\text{clip}} = T_p/T_v = 20$. δ is set to 100 by default. The maximum number of tracked persons per video is capped at 3.

C More Experimental Results

C.1 Robustness to Synthetic Pose Perturbations

We assess Iris-8B under inference-only perturbations of its 160-frame pose input, keeping the model weights and 16 RGB input frames unchanged. Table 9 adds Gaussian noise to normalized 2D and 3D coordinates and zeros contiguous blocks of pose frames. The unperturbed setting uses the automatic test-time pipeline in Appendix A.2 and retains the pose estimator's natural errors.

Performance degrades gradually across the tested perturbations. Under the strongest setting, MotionBench accuracy decreases by 0.5 percentage points (62.3% to 61.8%), while RepCount MAE increases from 0.346 to 0.379. Iris outperforms the RGB-only backbone on both reported metrics in every tested setting, indicating tolerance to these synthetic coordinate errors and temporal dropouts.

C.2 Preliminary Observation on Left-Right Discrimination

Preliminary qualitative inspection suggests that structured pose cues may help Iris distinguish the subject's left and right body parts under viewpoint changes. Here, left and right are defined relative to the human subject. This observation is exploratory; a systematic evaluation and detailed analysis are left to future work.

C.3 Heterogeneous Pose Representation

Table 10 ablates the design of our 9-dimensional per-keypoint descriptor. We compare three configurations, including only 2D image coordinates (u_k, v_k), only 3D positions (x_k, y_k, z_k), and our full

Table 11: Ablation study on the pose sequence length. Densification factor T_{clip} =pose frame length / effective visual time steps (8).

Densification Factor T_{clip}	**MotionBench** (Acc.)	**FAVOR Bench** (Acc.)	**UCFRep** (MAE)
1	58.5	52.8	0.8772
10	60.1	53.6	0.5489
20	**60.5**	**54.1**	0.4656
30	60.4	54.1	**0.4303**

Table 12: Ablation study on the spatial binding implementation, *i.e.*, how to obtain the pose-visual tokens.

Spatial Binding Implementation	**MotionBench**	**FAVOR Bench**
Residual Fusion	60.3	53.7
Directly Concat	**60.5**	**54.1**

heterogeneous descriptor that combines both, along with the global pelvis translation and visibility flag.

Both the 2D-only and 3D-only variants already surpass the naive Qwen2.5-VL-7B baseline, confirming the value of explicit structural cues. However, the two modalities are complementary. 2D coordinates provide direct spatial grounding in the image plane, enabling the spatial binding module to localize body parts in the visual stream, while 3D positions capture depth-aware kinematic structure. The full 9D descriptor achieves the best performance across all metrics. Notably, the Camera Motion (CM) category benefits most from the decoupled global-local joint positions, which help model decouple ego-motion from local body articulation implicitly.

C.4 Densification Factor (Pose Sequence Length)

Table 11 ablates the temporal densification factor T_{clip}, which controls the ratio of pose frames T_p to effective visual time steps $T_v = 8$. $T_{\text{clip}} = 1$ supplies one pose frame per effective visual time step, without temporal densification.

First, $T_{\text{clip}} = 1$ yields noticeably inferior results across all benchmarks, supporting the value of temporal densification beyond introducing pose as an additional modality. Besides, motion understanding benchmarks (MotionBench and FAVOR Bench) saturate at $T_{\text{clip}} = 20$, because most motion understanding tasks can tolerate slight temporal information degradation. In contrast, the repetitive counting task (UCFRep MAE) continues to benefit from higher densification ($T_{\text{clip}} = 30$), consistent with its strict requirement for capturing high-frequency motion periodicity. We adopt $T_{\text{clip}} = 20$ as our default configuration, which yields strong performance across all tasks while maintaining a manageable pose sequence length of $T_p = 160$ frames for efficiency.

C.5 Spatial Binding Implementation

Table 12 compares two designs for producing the pose-visual tokens x_{pv} output from the cross-attention module. In the *Residual Fusion* variant, the cross-attended output is added back to the original pose tokens as residuals via a connection before being fed to the MLLM. In this case, pose tokens are processed into pose-visual tokens, and only the new pose-visual tokens, the visual tokens, and the textual tokens are fed into the MLLM. In the *Directly Concat* design (ours), the original pose tokens x_p and the cross-attended tokens x_{pv} are kept separate and concatenated along the part dimension, so the MLLM receives both the raw structural pose features and the spatially grounded visual context independently. Direct concatenation slightly outperforms residual fusion on both benchmarks. In our final implementation, we choose the directly concat version.

C.6 Qualitative Results

Figure 9 presents four representative comparison cases between Qwen3-VL-8B and Iris-8B, covering fine-grained action recognition, subject-centered left-right reasoning, and repetitive counting.

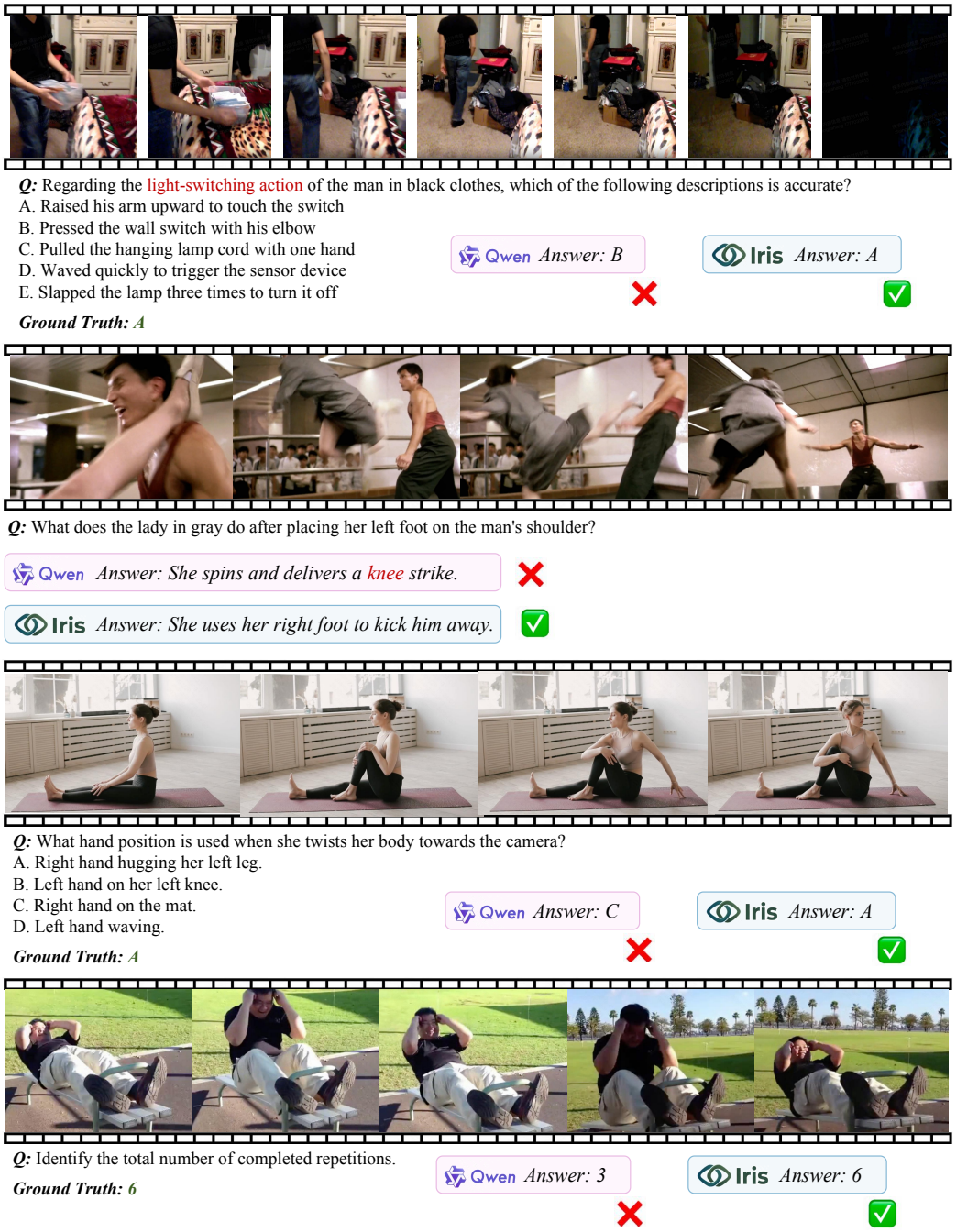


Figure 9: Qualitative results comparison between Qwen3-VL-8B and Iris-8B on fine-grained motion understanding tasks.

Fine-grained action understanding (rows 1 & 2). In the first case, given a multiple-choice question about a light-switching action, the baseline incorrectly selects a plausible but wrong option ("pressed the wall switch with his elbow"), failing to distinguish the subtle arm trajectory involved. Iris correctly identifies the action as raising the arm upward to touch the switch. In the second case, the baseline misidentifies the subsequent action after the lady places her left foot on the man's shoulder, describing an incorrect strike motion, while Iris correctly recognizes that she follows through with a right-foot kick with the aid of the explicit human body pose information. Both cases demonstrate Iris's capacity

to resolve fine-grained limb kinematics and sequential action dynamics that are ambiguous under sparse visual sampling.

Repetitive action counting (row 4). For a sit-up counting task, the baseline severely undercounts (3 vs. ground truth 6), consistent with its general failure mode on temporal counting under sparse visual sampling. Iris produces the exact ground truth count, benefiting from the densely sampled pose stream that faithfully preserves the high-frequency periodicity of the motion.

D Limitations and Future Work

Despite its strong performance, we discuss several limitations that also point to promising directions for future work. First, the pose stream relies on automated pose estimation, which can produce unreliable results under severe body occlusion or extreme truncation. Severely unstable pose sequences may be discarded by our post-processing pipeline, as described in Appendix A.2. The model then falls back to RGB-only inference and cannot exploit dense kinematic input for those samples. Second, the current pipeline caps the number of tracked person identities at three per clip for training and data efficiency. While this is sufficient for the vast majority of real-world scenarios, it limits direct applicability to densely crowded scenes, where *determining visual focus* and *handling more severe person occlusion* require additional design considerations.

E Broader Impact

Iris advances fine-grained human motion understanding in video, with potential positive applications across a range of domains, including sports analysis, rehabilitation monitoring, human-computer interaction, and embodied AI. Notably, the introduction of structured pose representations as a complementary modality offers an additional privacy-preserving dimension. Since pose sequences abstract away personal appearance, systems built on Iris can perform motion analysis without relying on or exposing much facial identity or biometric appearance features, making them more suitable for privacy-sensitive deployments. On the risk side, as with any technology that enables detailed analysis of human body movements, there is a potential for misuse in surveillance or behavioral profiling.