Superman: Unifying Skeleton and Vision for Human Motion Perception and Generation

Supplementary Material

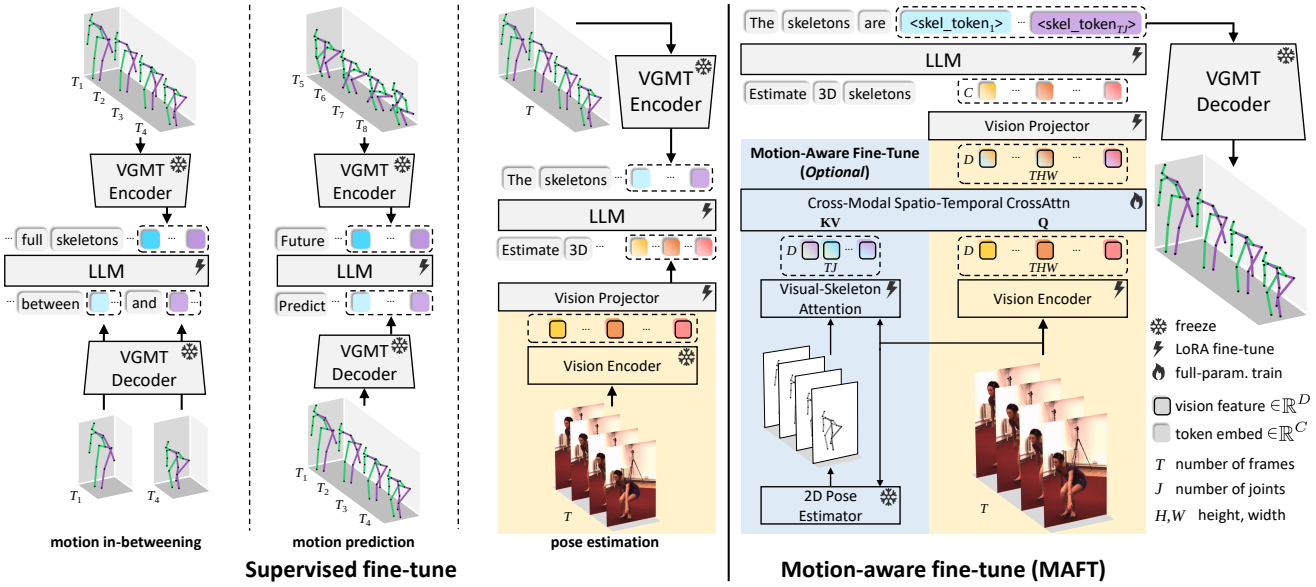

Figure S1. **Detailed Data Flow and Training Pipelines for Different Tasks. Left:** For generation tasks like *Motion In-betweening* and *Motion Prediction*, the model operates in a "Skeleton-to-Skeleton" mode. Historical or keyframe skeletons are tokenized by the frozen VGMT Encoder, and the LLM autoregressively predicts the missing/future tokens. **Right:** For *Pose Estimation*, the model operates in a "Video-to-Skeleton" mode. We compare the standard Supervised Fine-Tuning (SFT) pipeline against our proposed **Motion-Aware Fine-Tuning (MAFT)**. The MAFT pipeline explicitly injects skeletal geometry into the visual stream via the VSA module, enhancing feature alignment before LLM processing.

A. Implementation Details

A.1. Training Strategy

Tokenizer Training: The VGMT is trained for 500000 steps on Human3.6M. We use the AdamW optimizer with a learning rate of 0.0002 and batch size 256.

Unified MLLM Tuning: As shown in Fig. S1, unified MLLM tuning includes: *1) Multi-Task Instruction Tuning.* We adopt a joint training strategy where data from all three tasks—3D Pose Estimation, Motion Prediction, and Motion In-betweening—are mixed within each batch. To unify these disparate tasks, we utilize the specific instruction templates to convert raw multi-modal data into a standardized "User-Assistant" conversation format. The model is optimized using a standard autoregressive cross-entropy loss over output tokens. *2) Optional MAFT Integration.* For tasks involving video input (i.e., 3D Pose Estimation), we incorporate the Motion-Aware Fine-Tuning (MAFT) module. This light-weight component (adding $< 0.2\%$ trainable parameters) injects skeletal geometric priors into the visual stream via the Visual-Skeleton Attention (VSA) mechanism, enabling the MLLM to better attend to motion-relevant visual features. *3) Hyper-parameters.* The MLLM is fine-tuned using a learning rate of $1e-4$ with a cosine decay scheduler and a warmup ratio of 0.1. The global batch size is set to 16. All experiments are conducted on 2 NVIDIA H20 GPUs using PyTorch.

A.2. Instruction Templates

To unify diverse motion analysis tasks within a single MLLM framework, we employ instruction tuning. We convert raw data (videos and skeleton sequences) into a standardized conversation format, where the user provides an instruction containing multimodal inputs (e.g., `<video>`, `<skeleton>`), and the model generates the corresponding target skeleton tokens. Tab. S1 details the specific instruction templates used for each task during the training.

- **Vid2Skel (3D Pose Estimation):** The model inputs visual tokens (and optional 2D estimated poses) and is instructed to describe the motion using skeleton tokens.
- **SkelPred (Motion Prediction):** The model inputs a history sequence of skeleton tokens and predicts future frames. Both sequences contain 16 frames (320ms).

Table S1. **Instruction Templates for Multi-Task Training.** We utilize specific text prompts to guide the MLLM for different tasks. The bottom section illustrates a concrete example of the actual input prompt (with special vision tokens) and the corresponding output response.

Task	Input Modality	Instruction Template (User Prompt)	Output
3D Pose Estimation (Vid2Skel)	Video	Please describe the motion of the person in the video <video> using skeleton tokens. Your response should be formatted as: "Frame 1: torso: ... left_arm: ... right_arm: ... left_leg: ... right_leg: ... Frame 2: ... "	Skeleton Tokens ($K_{1:T}$)
Motion Prediction (SkelPred)	History Skeleton	Here's a motion sequence represented using skeleton tokens: <skeleton> Predict the future motion using skeleton tokens that have the same number of frames as the history motion. Your response should be formatted as: "Future Frame 1: torso: ... left_arm: ... right_arm: ... left_leg: ... right_leg: ... Future Frame 2: ... "	Future Skeleton Tokens
Motion In-betweening (MIB)	Start/End Keyframes	Here's a motion sequence with missing in-between frames. It contains only the start and end keyframes, represented using skeleton tokens: <skeleton> Generate the in-between frames to create a smooth transition between the provided keyframes. Your response should be formatted as a complete sequence: "Frame 1: ... left_arm: ... right_arm: ... left_leg: ... right_leg: ... Frame 2: ... "	Intermediate Skeleton Tokens

Real Input Prompt Example (with Special Tokens):

User: Please describe the motion of the person in the video <|vision_start|><|video_pad|>...<|video_pad|><|vision_end|> using skeleton tokens. Your response should be formatted as: "Frame 1: torso: ... left_arm: ... right_arm: ... left_leg: ... right_leg: ... Frame 2: ... "

Real Output Example (Body-Part Level Grouping):

Assistant: There are 4 frames in total. Here are the skeleton tokens for 5 body parts in each frame:
Frame 1: torso: <skel_1848><skel_7294><skel_188><skel_5434><skel_975>. left_arm: <skel_5563><skel_6334><skel_7382>. right_arm: <skel_1656><skel_7242><skel_7368>. left_leg: <skel_4964><skel_445><skel_426>. right_leg: <skel_2974><skel_3963><skel_1134>.
Frame 2: torso: <skel_7234><skel_7612><skel_7832><skel_2617><skel_3466>. left_arm: <skel_2904><skel_165><skel_5270>. ...

- **MotionInBetween (In-betweening):** The model inputs sparse keyframes (start and end) and generates the intermediate transition sequence.

B. Additional Visualization Results

We further demonstrate the superiority of Superman through qualitative comparisons across tasks.

B.1. Robustness in 3D Pose Estimation

Fig. S2 presents a visual comparison for pose estimation.

- **Correction of Upstream Errors:** The bottom block of Fig. S2 highlights a challenging scenario where the off-the-shelf 2D pose estimator (CPN [1]) fails to detect the right foot due to motion blur (see red box). Baseline methods like HiC [2], which rely heavily on 2D keypoints, propagate this error into the final 3D output.
- **Effect of MAFT:** Our model with Motion-Aware Fine-Tuning (MAFT) effectively mitigates this issue. By leveraging the Cross-Attention mechanism to query visual features directly from the video, *Superman* "looks back" at the raw frames to recover the missing foot information.

B.2. Temporal Coherence in Motion Prediction

Fig. S3 compares the prediction capabilities on the Human3.6M dataset. The task involves predicting 320ms of future motion based on historical context. As observed in the "Sitting" sequence, baseline methods such as MotionGPT3 [4] and HiC [2] tend to generate stiff or slightly drifting motions as time progresses. In contrast, *Superman* maintains high temporal fidelity, accurately predicting the descent of the body and the knee bending trajectory, closely matching the ground truth.

B.3. Generalization in Motion In-betweening

Fig. S4 evaluates the Motion In-betweening task, where the model must synthesize a coherent sequence connecting a

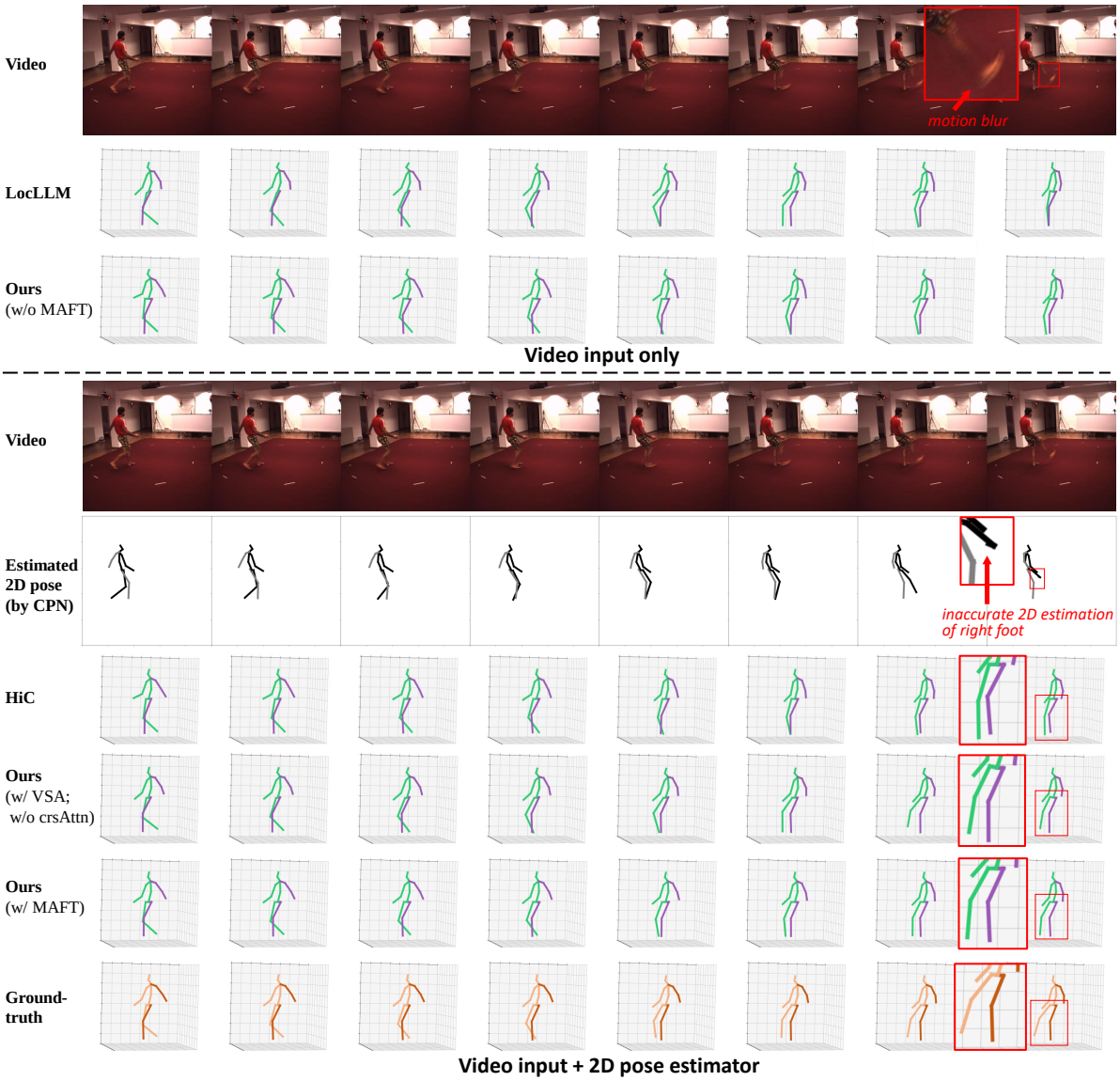

Figure S2. **Qualitative Results for 3D Pose Estimation on Human3.6M: The Impact of MAFT. Top Block (Video Input Only):** Compared to LocLLM [3], our method (even without MAFT) generates more physically plausible poses from raw video, handling motion blur more effectively. **Bottom Block (Video + 2D Pose Input):** This case highlights the robustness of our MAFT module. The off-the-shelf 2D pose estimator (CPN [1]) fails to detect the right foot due to occlusion/blur (see red box). Consequently, the baseline HiC [2], which relies heavily on 2D inputs, produces an erroneous pose. In contrast, **Ours (w/ MAFT)** successfully corrects this error by attending to the visual features via the cross-attention mechanism, recovering the correct leg position that matches the Ground Truth.

start and an end frame.

- **In-Domain (Human3.6M):** In the top row, our model generates a smooth turning motion, naturally interpolating the limb rotations.
- **Out-of-Domain (3DPW):** The bottom row demonstrates generalization on the unseen 3DPW dataset. Despite never being trained on this data, *Superman* synthesizes a realistic stepping motion. Comparison methods often struggle with such large gaps or unseen poses, resulting in artifacts. Our success here attributes to the robust, vision-grounded motion vocabulary learned by the VGMT.

C. Visualization of Intermediate Features

We analyze *Superman* by visualizing the intermediate representations learned by its key components. This provides qualitative evidence for the mechanisms of VSA sampling, MAFT attention, and the semantic structure of the learned codebook.

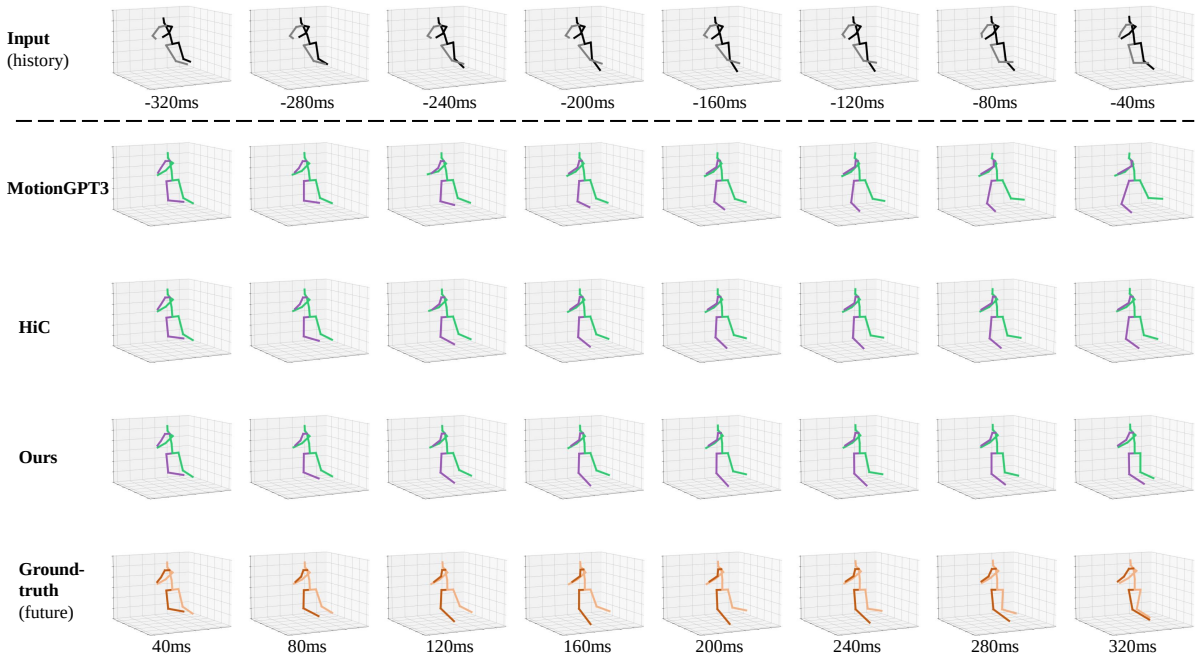

Figure S3. **Qualitative Comparison for Motion Prediction on Human3.6M.** The model predicts the future 320ms motion given a history sequence (grey). While baselines like MotionGPT3 [36] and HiC [17] exhibit slight temporal jitter or drift in the leg positioning during the "Sitting" action, **Superman (Ours)** generates a smooth and accurate trajectory that closely aligns with the Ground Truth (orange), demonstrating superior temporal coherence.

C.1. Adaptive Sampling in VSA

A critical challenge in vision-guided motion generation is the unreliability of upstream 2D pose estimators, particularly under rapid motion or occlusion. To validate how our Visual-Skeleton Attention (VSA) module addresses this, we visualize the internal sampling process in Fig. S5 & S6.

Visualization Protocol: We visualize the sampling behavior of different attention heads across temporal frames. In the visualization:

- The **Start Point** of a green line represents the initial reference point provided by the off-the-shelf 2D pose estimator (CPN).
- The **Green Line** represents the learned offset vector (Δp) predicted by the VSA module.
- The **End Point (Green Circle)** indicates the final adaptive sampling location on the feature map.
- The **Area of the Circle** corresponds to the learned attention weight, where larger circles indicate higher contribution to the aggregated feature.

Analysis of Correction Capability: As highlighted in the bottom row of Fig. S5 (Frame 8), the fast-moving right foot causes significant motion blur, leading the fixed 2D estimator to incorrectly localize the joint on the leg or background (see the white dot). However, our VSA module detects this semantic misalignment. The predicted offsets (green lines) diverge from the erroneous initial point and accurately point towards the actual "ghosting" region of the blurry foot. Furthermore, different attention heads (Head 1-4) learn to focus on complementary features—some attending to the heel and others to the toe trajectory—aggregating a robust visual representation despite the noisy input. This confirms that VSA functions not just as a sampler, but as a dynamic *visual corrector*.

Fig. S6 demonstrates the tracking of the left foot. Even when the 2D estimation is relatively stable, the VSA module actively refines the sampling points. Different attention heads (Head 1-4) can be seen focusing on distinct semantic parts (e.g., heel vs. toe) or expanding the receptive field to capture context, ensuring a rich feature representation.

C.2. Semantic Analysis of the Codebook

Finally, to validate that our discrete tokens carry explicit physical meanings, we perform a "Semantic Sphere" analysis. For every code index k in the learned vocabulary, we calculate the average displacement vector of all motion segments assigned to it. We then visualize these vectors as fiber-like lines radiating from the origin, normalized to unit length to emphasize directionality.

Fig. S7 presents the results for four critical end-effectors: **Left/Right Wrists** and **Left/Right Ankles**. These joints typically exhibit the highest degrees of freedom and complexity in human motion. Key observations:

- **Isotropic Coverage (No Mode Collapse):** As illustrated in the figure, the motion vectors for all four joints form

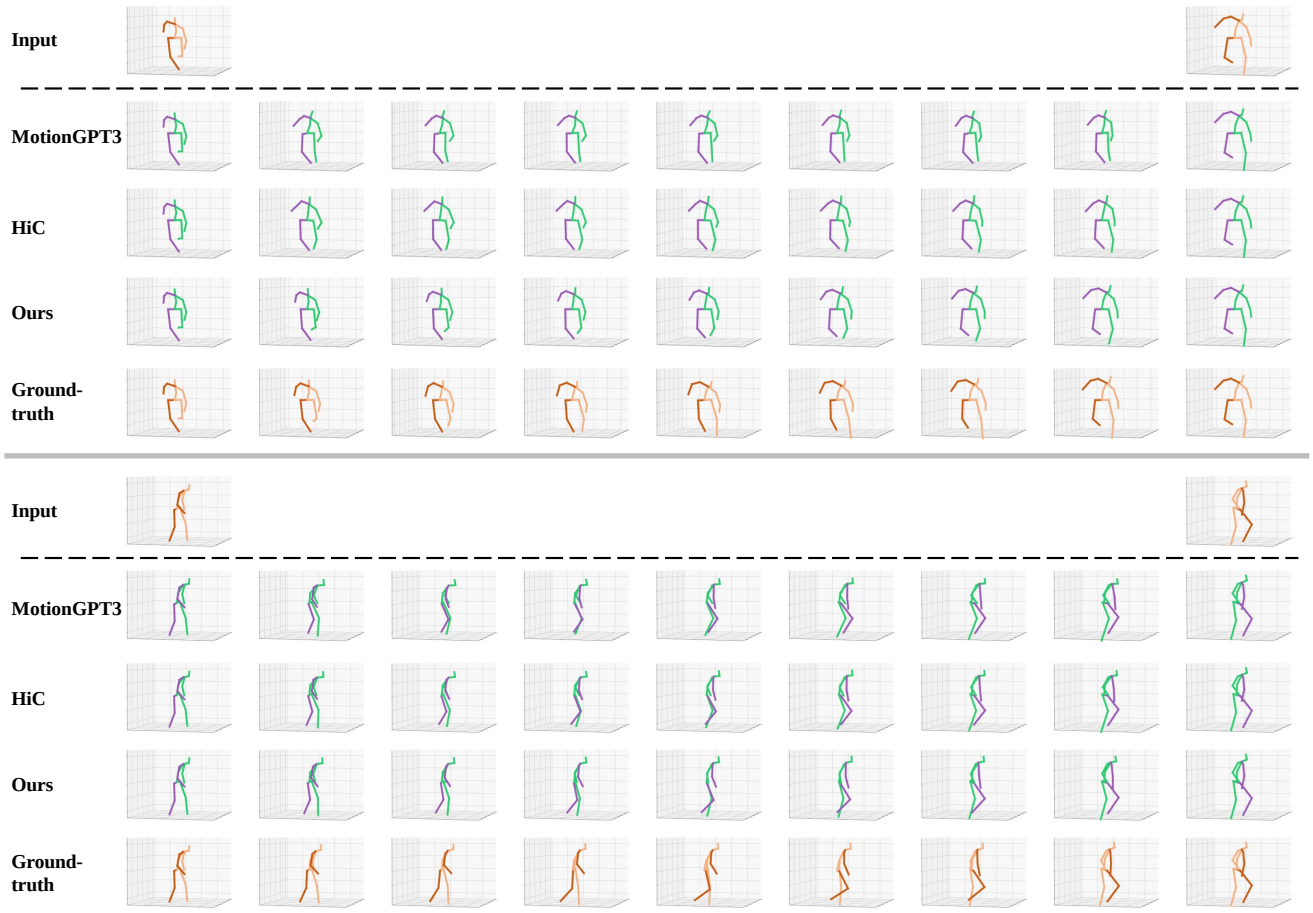

Figure S4. **Qualitative Results for Motion In-betweening on Human3.6M and 3DPW.** The task is to generate the intermediate motion sequence given only the first and last frames (Input). **Top Row (Human3.6M):** Our model generates a natural transition for the turning action, whereas MotionGPT3 struggles with the limb orientation. **Bottom Row (3DPW):** On the unseen 3DPW dataset, our model demonstrates strong generalization, synthesizing a realistic stepping motion that bridges the gap smoothly, outperforming comparison methods.

dense, almost perfect spheres. The uniform distribution of vectors in every direction (isotropic) provides compelling evidence that our codebook effectively covers the full manifold of possible motion directions. There are no significant "holes" or gaps, indicating that the model has avoided mode collapse and can generate diverse motions in any 3D direction.

- **High-Fidelity Granularity:** The sheer density of the fibers—representing over 2,000 active codes for these end-effectors—confirms that the tokenizer captures fine-grained kinematic details. The model allocates a vast vocabulary to describe subtle variations in hand and foot movements, which is crucial for high-quality motion generation.
- **Directional Semantics:** The smooth transition of colors (representing azimuthal direction) confirms that the codes are semantically organized. Specific codes map uniquely to specific physical directions, allowing the subsequent MLLM to control motion with precise directional intent.

References

[1] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018. 2, 3

[2] Mengyuan Liu, Xinshun Wang, Zhongbin Fang, Deheng Ye, Xia Li, Tao Tang, Songtao Wu, Xiangtai Li, and Ming-Hsuan Yang. Human-in-context: Unified cross-domain 3d human motion modeling via in-context learning. *arXiv preprint arXiv:2508.10897*, 2025. 2, 3

[3] Dongkai Wang, Shiyu Xuan, and Shiliang Zhang. Locllm: Exploiting generalizable human keypoint localization via large language model. In *CVPR*, 2024. 3

[4] Bingfan Zhu, Biao Jiang, Sunyi Wang, Shixiang Tang, Tao Chen, Linjie Luo, Youyi Zheng, and Xin Chen. Motiongpt3: Human motion as a second modality. *arXiv preprint arXiv:2506.24086*, 2025. 2

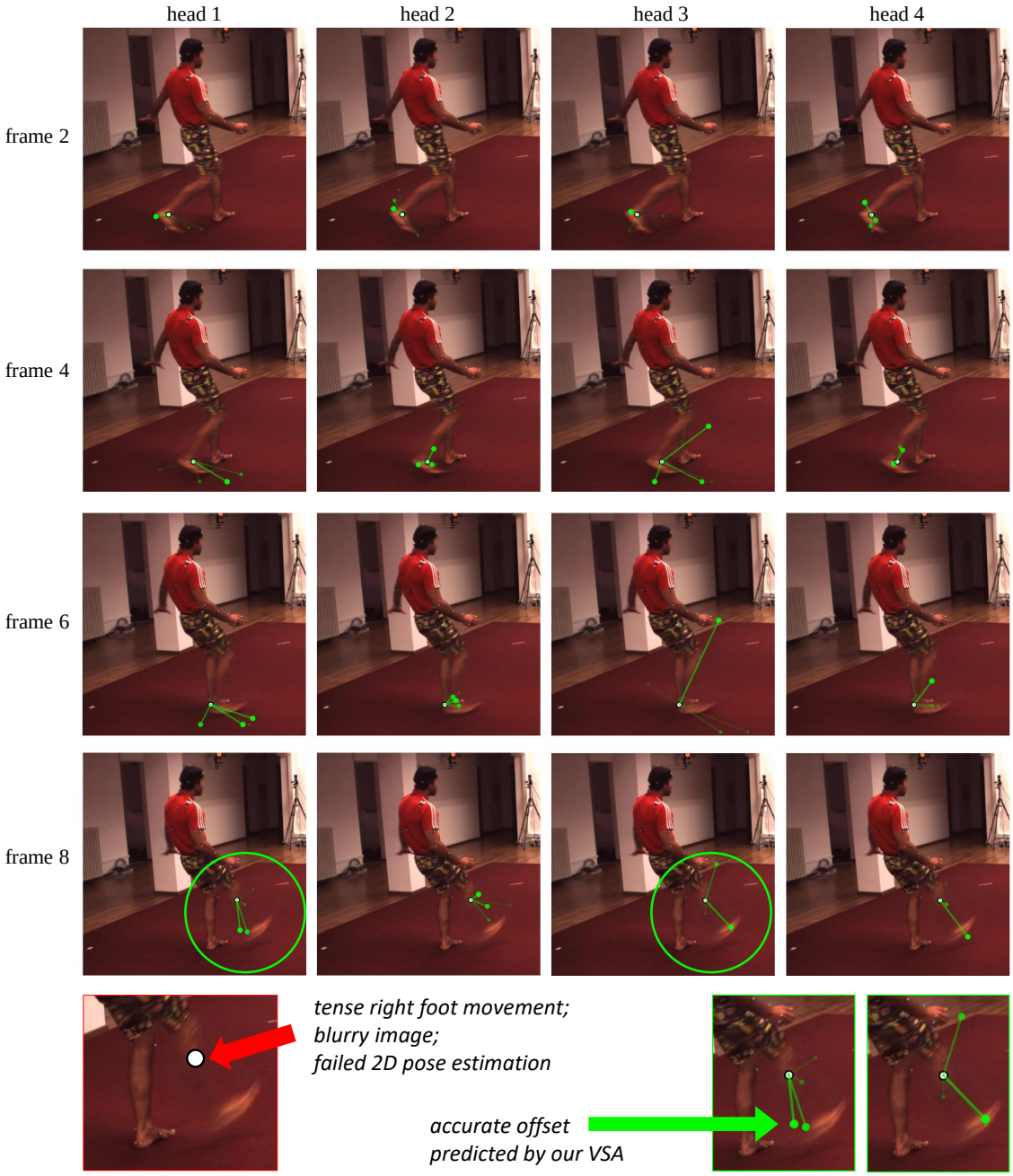

Figure S5. **Visualization of Adaptive Sampling in Visual-Skeleton Attention (VSA)** – Right foot as an example. We visualize the learned sampling offsets and weights for different attention heads (columns) across video frames (rows). The **start point** of each green line indicates the initial, potentially noisy 2D keypoint estimated by CPN. The **green line** represents the predicted offset, pointing to the **green circle** which denotes the final sampling location. The **area** of the green circle reflects the attention weight assigned to that sampling point. **Highlight (Bottom Row):** In Frame 8, rapid motion causes the right foot to blur, leading to a failed 2D estimation (white dot). Our VSA module successfully identifies the error, predicting large offsets that redirect attention from the erroneous leg position back to the true, blurry foot region (red and green boxes), thereby ensuring robust feature extraction.

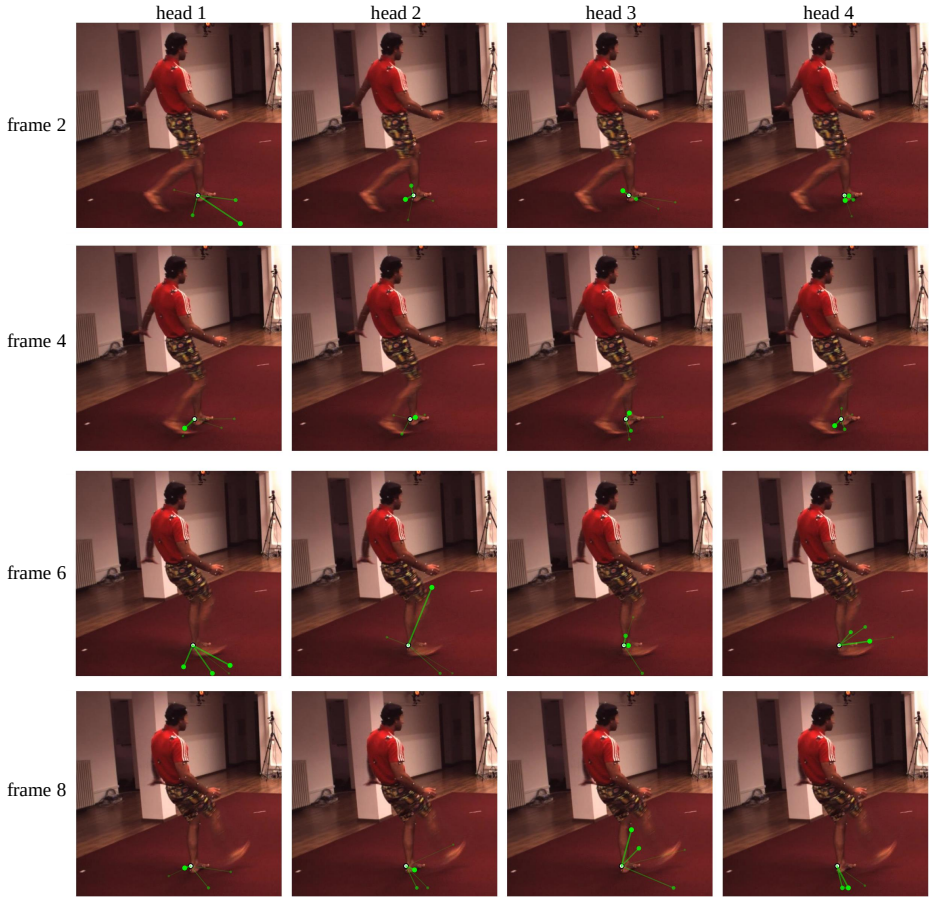

Figure S6. **Visualization of Adaptive Sampling in Visual-Skeleton Attention (VSA)** – Left foot as an example. The VSA sampling process for the planting and lifting phase of the left foot is illustrated. **Multi-Head Diversity:** Different attention heads exhibit diverse sampling strategies. For instance, while Head 2 tends to focus tightly on the joint center (small offsets), Head 4 explores a wider neighborhood (larger offsets) to capture contextual motion cues. This diversity allows the model to build a robust representation of the limb's state even during complex articulation.

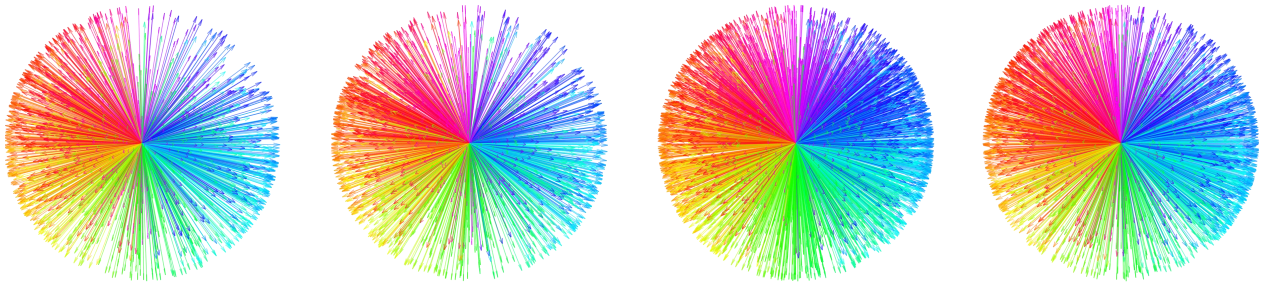

Figure S7. **Visualization of the "Semantic Spheres" of Motion Primitives across Different Joints.** Each sphere visualizes the discrete vocabulary learned by our Vision-Guided Motion Tokenizer (VGMT) for a specific body joint. Each fiber-like line represents a unique code in the codebook, plotted as its normalized average displacement vector (starting from the origin). The color indicates the motion direction (azimuthal hue). **(1) Omnidirectional Coverage:** The dense, isotropic spherical distribution demonstrates that the codebook effectively covers the full manifold of possible motion directions without mode collapse. **(2) High-Fidelity Granularity:** The high density of vectors (e.g., over 2,000 active codes for hands and feet) confirms that the tokenizer captures fine-grained kinematic details.