

UniPose: A Unified Multimodal Framework for Human Pose Comprehension, Generation and Editing

Yiheng Li^{1,2}, Ruibing Hou^{1*}, Hong Chang^{1,2}, Shiguang Shan^{1,2}, Xilin Chen^{1,2}

¹Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),

Institute of Computing Technology, CAS, China

²University of Chinese Academy of Sciences, China

yiheng.li@vipl.ict.ac.cn, {houruibing, changhong, sgshan, xlchen}@ict.ac.cn

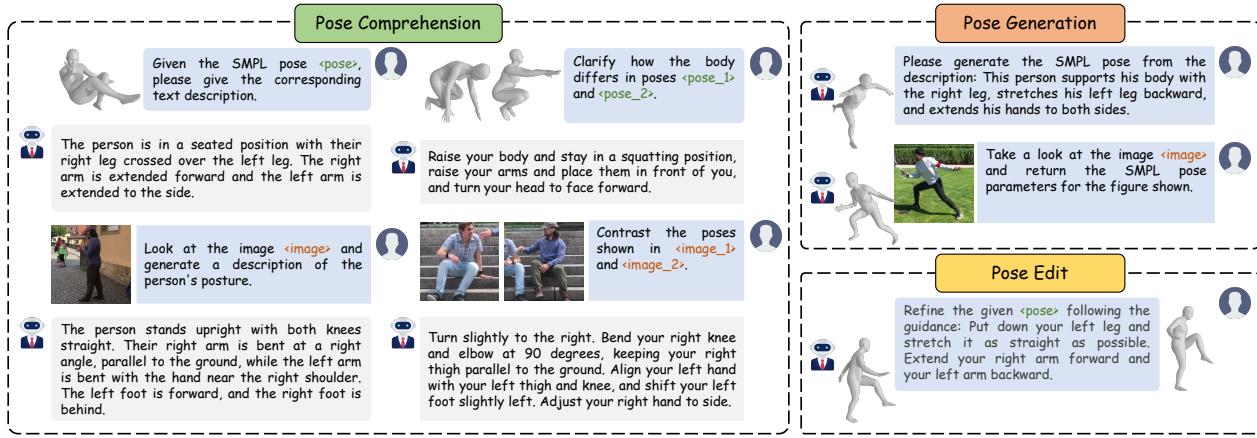


Figure 1. UniPose can handle pose comprehension, generation and editing tasks under different instructions within a unified framework.

Abstract

Human pose plays a crucial role in the digital age. While recent works have achieved impressive progress in understanding and generating human poses, they often support only a single modality of control signals and operate in isolation, limiting their application in real-world scenarios. This paper presents UniPose, a framework employing Large Language Models (LLMs) to comprehend, generate, and edit human poses across various modalities, including images, text, and 3D SMPL poses. Specifically, we apply a pose tokenizer to convert 3D poses into discrete pose tokens, enabling seamless integration into the LLM within a unified vocabulary. To further enhance the fine-grained pose perception capabilities, we facilitate UniPose with a mixture of visual encoders, among them a pose-specific visual encoder. Benefiting from a unified learning strategy, UniPose effectively transfers knowledge across different pose-relevant tasks, adapts to unseen tasks, and exhibits extended capabilities. This work serves as the first attempt at building a general-purpose framework for pose comprehension, generation, and editing. Extensive experiments highlight UniPose's competitive and even superior performance across various pose-relevant tasks. Code is

available at <https://github.com/liyiheng23/UniPose>.

1. Introduction

Human pose plays a pivotal role in various human-centric applications such as VR and healthcare. Numerous studies focus on *single-pose comprehension*, i.e., producing posture-relevant description from a 3D body pose [14] or human image [19], as well as *pose generation*, i.e., creating complex 3D poses from textual descriptions [14, 28, 41] or human images [9, 16, 20, 58, 71]. Recently, a few studies have explored the relationship between pairs of poses [13, 22, 35]. These studies investigate *pose-pair comprehension*, where textual instruction is produced based on the differences between two 3D poses, and *pose editing*, where corrected 3D body pose is generated based on an initial pose and modification instruction. However, a key limitation of existing work is that pose comprehension, generation, and editing are predominantly studied in isolation. In reality, human pose cognition and communication inherently involve seamless transitions between multiple pose-relevant modalities, including 3D SMPL poses [46], textual descriptions, and human images. This highlights the need for a unified multimodal framework capable of simultaneously handling pose comprehension, generation, and editing.

Tasks	Pose Comprehension				Pose Generation		Pose Editing
	Pose-to-Text	Image-to-Text	Pose-Diff	Image-Diff	Text-to-Pose	Pose Estimation	
Input→Output	Pose→Text	Image→Text	Pose Pair→Text	Image Pair→Text	Text→Pose	Image→Pose	Pose&Text→Pose
HMR 2.0 [24]	✗	✗	✗	✗	✗	✓	✗
PoseScript [14]	✓	✗	✗	✗	✓	✗	✗
PoseFix [13]	✗	✗	✓	✗	✗	✗	✓
ChatPose [19]	✗	✓	✗	✗	✓	✓	✗
ChatHuman [42]	✗	✓	✗	✗	✓	✓	✗
PoseEmbroider [15]	✓	✓	✓	✓	✓	✓	✗
UniPose (Ours)	✓	✓	✓	✓	✓	✓	✓

Table 1. Comparison of recent methods across various pose comprehension, generation and editing tasks.

Recent years have witnessed a significant breakthrough in large language models (LLMs) [26, 31, 68] and multi-modal LLMs (MLLMs), enabling general-purpose analysis of images [4, 44], videos [39, 74], motions [10, 32, 77], and audios [73, 75]. In the area of human poses, ChatPose [19], a recent innovation, leverages LLMs to generate 3D human poses from images and textual descriptions. Nevertheless, it focuses solely on single-pose generation, lacking the capacity for pose comprehension and editing. Moreover, existing MLLMs still fall short in providing comprehensive analysis of human poses, particularly concerning fine-grained part semantics and complex relationships between pose pairs. Consequently, a unified multimodal LLM that enables fine-grained pose comprehension, generation, and complex pose editing is still in highly demand.

Two main challenges need to be solved for building such a unified multimodal LLM framework. The first challenge is creating a unified representation space across 3D poses and texts, enabling the unification of diverse pose-relevant tasks. Existing work [19] processes 3D poses and texts differently, encoding 3D poses as continuous high-level features while tokenizing linguistic texts into discrete token sequences. This non-unified processing incurs an extra burden on LLMs to model interactions between 3D poses and texts, hindering the unifying of pose comprehension, generation and editing. The second challenge lies in achieving fine-grained pose perception within the visual branch of the multimodal framework. Most MLLMs [4, 19, 44, 66] employ CLIP [53] as their visual branch. While CLIP’s visual encoder aligns well with the text embedding space through image-text contrastive learning, it struggles to capture detailed pixel-level information, such as keypoints and parsing maps, due to the global supervision provided by image captions. This limitation constrains MLLM’s capabilities in fine-grained pose comprehension and generation.

To address these challenges, we propose UniPose, a uniform multimodal framework for human pose comprehension, generation and editing, which harnesses the powerful language generation abilities of LLMs to unify various pose-relevant tasks (Tab. 1). UniPose comprises three tires. **Firstly**, UniPose is equipped with a *pose tokenizer* for processing 3D poses and texts uniformly. Inspired by the obser-

vation that human poses exhibit a semantic coupling similar to language [32, 47, 69], we treat 3D pose as a specific language. Akin to language, the pose tokenizer compresses raw 3D pose into a sequence of discrete semantic tokens. By encoding both 3D pose and language within a shared vocabulary, we build a unified representation space across 3D poses and texts, which enables LLMs to be easily adapted to handle pose comprehension, generation, and editing. **Secondly**, unlike most MLLMs [4, 19, 44, 66] that solely rely on CLIP’s visual encoder [53], we adopt a mixture-of-visual-encoders that combines CLIP’s original visual encoder with a pose-specific visual encoder pre-trained on pose estimation task. This dual-encoder setup not only aligns visual representations with text embedding space but also enhances fine-grained pose perception, enabling more effective integration into the multimodal framework for improved pose comprehension and generation. **Thirdly**, we implement a mixed-attention mechanism within LLMs to handle the distinct internal logical relationships between pose and text tokens. Unlike text tokens, pose tokens encode spatial joint positions without causal dependencies, making unified autoregressive modeling suboptimal. To address this, we apply causal attention to text tokens and bidirectional attention to pose tokens. This mixed-attention strategy preserves LLM’s original reasoning capabilities while enhancing contextual pose perception, enabling more effective pose generation and editing.

To our knowledge, UniPose is the first approach to integrate seven core tasks of pose comprehension, generation, and editing into a uniform framework. Extensive experiments demonstrate that UniPose achieves competitive performance across multiple pose-relevant tasks. Additionally, through qualitative results, we demonstrate that UniPose possesses zero-shot generalization capabilities, *e.g.*, text-enhanced pose estimation.

2. Related Work

Human Pose Comprehension. Pose comprehension involves generating natural language descriptions of human pose or differences between pose pairs. For single-pose comprehension, traditional methods classify basic human actions from images [78], videos [63, 64, 67], or skele-

tions data [2, 11, 23, 50]. However, these methods typically lack detailed descriptions of specific body part positioning. To address this gap, [14] introduces PoseScript dataset which pairs human poses with detailed body parts descriptions, and propose a pose-to-text generation model that uses cross-attention to embed pose information within a text transformer for nuanced pose descriptions. For pose-pair comprehension, [13, 22, 35] describe differences between source and target poses based on images, videos, or 3D poses. For example, PoseFix [13] uses an MLP to fuse source and target pose, then uses cross-attention in a text transformer to generate descriptions of pose differences. While these approaches enhance understanding of human poses from multimodal data, they are typically task-specific, with limited control conditions and application scenarios.

Human Pose Generation. Pose generation synthesizes human poses conditioned on text or images. Text-conditioned pose generation generally falls into two categories: shape-oriented [25, 57] and pose-oriented [8, 28, 41], which generate 3D poses from descriptions of body attributes (*e.g.*, slim waist) and simple actions (*e.g.*, running), respectively. Image-conditioned pose generation (also referred to pose estimation) includes optimization-based and regression-based approaches. Optimization-based methods [7, 17, 18, 52, 55] iteratively estimate 3D pose parameters, ensuring the projection of predicted 3D joints aligns with 2D keypoints. Regression-based methods [9, 16, 24, 34, 71] use deep neural networks to directly predict 3D pose parameters from input images. Although these methods have achieved promising results in pose generation, they lack the capability of pose comprehension and editing.

Multimodal Large Language Models. Large Language Models (LLMs) [26, 31, 59, 72] have shown remarkable capabilities in textual comprehension and reasoning. These models have been adapted for multimodal tasks, leading to the development of multimodal large language models. For example, models like mPLUG-Owl3 [74], MiniGPT-4 [80] and LLaVA [38, 44, 45] uses a visual encoder to extract image features and a projection layer to align image embeddings with text embeddings, enhancing general visual perception. Moving towards task-specific applications, LISA [37] and Video-LISA [5] extend MLLMs for segmentation by integrating SAM [36] for generating fine-grained segmentation masks. Additionally, Show-o [70] and Transfusion [79] combine MLLMs with diffusion models [27] to unify image understanding and generation. A recent work, ChatPose [19], applies LLMs to pose-related tasks, aiming to build a versatile pose generator. However, it remains limited in its capacity for pose understanding and editing.

3. Method

To equip LLM with the capability to comprehend, generate, and edit human poses, we propose a unified framework

named UniPose. As illustrated in Fig. 2, UniPose comprises three main components: a pose tokenizer, which quantizes original 3D poses (represented as SMPL [46] pose parameters) into discrete tokens (Sec. 3.1), a visual processor, which extracts fine-grained, pose-relevant features from visual inputs, and a pose-aware LLM, which supports unified modeling across multiple modalities (Sec. 3.2). To address pose-relevant tasks, we employ a four-stage straining scheme encompassing pose tokenizer training, pose-text alignment pre-training, vision projector pre-training, and instruction tuning (Sec. 3.3). During inference, pose tokens are decoded back to their original SMPL format by associated de-tokenizer, enabling various pose-relevant tasks to be executed via instructions (Sec. 3.3)

3.1. Pose Tokenizer

To represent 3D pose in discrete tokens, we build the pose tokenizer based on Vector Quantized Variational Autoencoders (VQ-VAE) [61], as shown in Fig. 2. The pose tokenizer consists of an encoder \mathcal{E} , a decoder \mathcal{D} , along with a learnable codebook $\mathcal{B}_p = \{b_m\}_{m=1}^M$ containing M discrete vectors. Formally, we represent a 3D pose p using SMPL pose parameters, *i.e.*, $p = [\gamma, \theta]$ where $\gamma \in \mathbb{R}^6$ denotes the root orientation and $\theta \in \mathbb{R}^{6K}$ denotes the rotations with K joints. Then the pose encoder \mathcal{E} that consists of several 1-D convolutional layers projects p into a latent embedding $z = \mathcal{E}(\theta)$ with $z \in \mathbb{R}^{L_p \times d_p}$, where L_p is the number of pose tokens and d_p is the latent dimension. Next, we transform z into a collection of codebook entries through discrete quantization. Specifically, the process of quantization replaces each item of z with its nearest entry in the codebook \mathcal{B}_p , obtaining the quantized latent vector $\hat{z} \in \mathbb{R}^{L_p \times d_p}$ as follows:

$$\hat{z} = \arg \min_{b_m \in \mathcal{B}_p} \|z - b_m\|_2. \quad (1)$$

After quantization, the pose decoder \mathcal{D} , consisting of several 1-D deconvolutional layers, projects \hat{z} back to raw pose space as $\hat{p} = \mathcal{D}(\hat{z})$. Following [61], we train the pose tokenizer using the loss function $\mathcal{L}_{vq} = \mathcal{L}_r + \mathcal{L}_e + \mathcal{L}_c$ where \mathcal{L}_r , \mathcal{L}_e , and \mathcal{L}_c denote reconstruction loss, embedding loss and commitment loss respectively. Further training and objective details are provided in the Appendix.

After training the pose tokenizer, the pose p can be represented as a sequence of discrete codebook indices of the quantized latent vector, namely *pose tokens* $\mathbf{u} \in \mathbb{R}^{L_p}$ as:

$$\mathbf{u} = \arg \min_{m \in \{1, \dots, M\}} \|z - b_m\|_2. \quad (2)$$

3.2. Pose-aware Vision-Language Model

Visual Processor. Previous works [4, 44] commonly use CLIP visual encoder [53] as the visual branch. However, since CLIP is optimized by global and coarse-grained supervision signals from image captions, it struggles to cap-

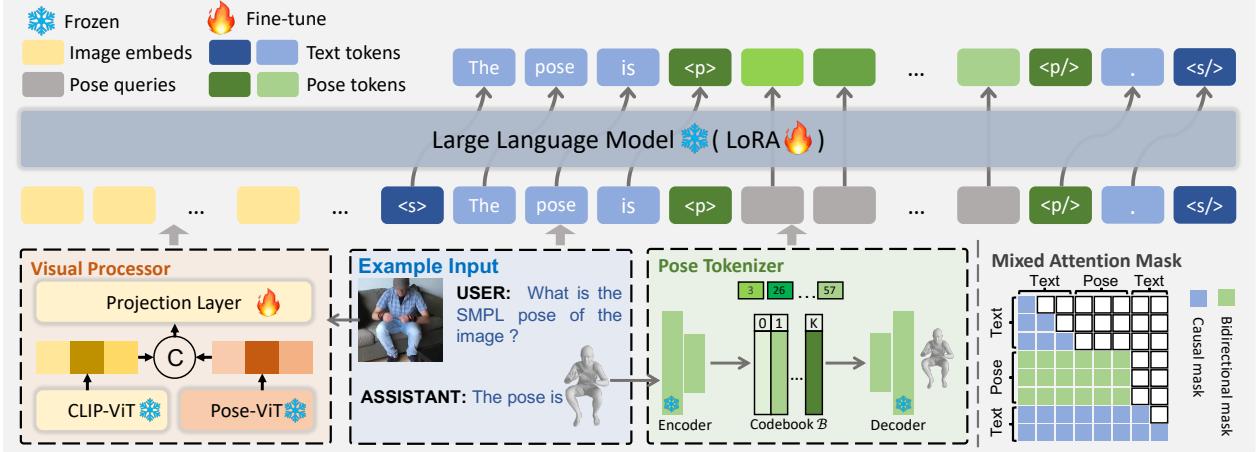


Figure 2. Method overview: UniPose comprises a Pose Tokenizer, Visual Processor and a pose-aware language LLM. Combining Pose Tokens learned by pose tokenizer, Visual Embeddings from visual processor and Text Tokens from text tokenizer, UniPose enables joint modeling of pose comprehension, generation and editing within a unified visual-language backbone.

ture pose-relevant details. Differently, the pose estimation task demands precise localization of human keypoints, which encourages the visual encoder to capture fine-grained pose features. Therefore, we integrate a pose-specific Vision Transformer [24], pretrained on the pose estimation task, into the visual branch, as shown in Fig. 2. Specifically, denote the CLIP visual encoder and pose-specific vision transformer as f_a and f_b , respectively. Given an input image x , we extract visual embeddings by CLIP as $\mathbf{v}_a = f_a(x)$ where $\mathbf{v}_a \in \mathbb{R}^{L_v \times d_a}$, L_v is the number of visual patch tokens and d_a is its visual embedding dimension. The embedding output by pose-specific vision transformer is $\mathbf{v}_b = f_b(x)$ where $\mathbf{v}_b \in \mathbb{R}^{L_v \times d_b}$. Then we concatenate the embedding output by these two encoders along the channel dimension, and apply a trainable projector layer (with projection matrix $W \in \mathbb{R}^{(d_a+d_b) \times d}$) to align the dimension of the concatenated visual features to that of text features as $\mathbf{v} = [\mathbf{v}_a | \mathbf{v}_b]^T W$. Here $\mathbf{v} \in \mathbb{R}^{L_v \times d}$ with d as text embedding dimensions of LLM. The fused visual features \mathbf{v} can be concatenated with pose or text tokens as input to LLM.

Mixed Attention Mechanism. Existing LLMs [31, 54, 59] typically employ autoregressive modeling with causal attention, excelling at generating sequential data such as text and audio [73, 75]. However, pose tokens, which encode spatial positions of human joints, are inherently non-sequential, making traditional autoregressive generation suboptimal. To address this issue, we propose modeling pose tokens as a whole. Inspired by [70, 79], we modify the standard causal attention in LLM, integrating bidirectional attention for pose tokens as depicted in Fig. 2. Specifically, we apply causal attention to text sequence, but apply bidirectional attention within the pose token sequence. To avoid information leakage, we initialize L_p learnable pose queries $\mathcal{Q} = \{q_1, \dots, q_{L_p}\}$ during the generation and editing of 3D poses, as shown in Fig. 2. These queries are used to predict

corresponding pose tokens in a single forward step. This design enables each pose token to attend to others within the same pose token sequence, while restricting access to only previously encountered text tokens.

Unified Multimodal Language Model. As shown in Fig. 2, equipped with a visual processor and pose tokenizer, we can compress original visual data x and pose data p into visual feature sequence $\mathbf{v} \in \mathbb{R}^{L_v \times d}$ and pose token sequence $\mathbf{u} \in \mathbb{R}^{L_p}$, respectively. To incorporate pose discrete tokens into LLMs, we expand the original text vocabulary \mathcal{V}_t of LLM with pose vocabulary \mathcal{V}_p ¹, forming a new unified text-pose vocabulary $\mathcal{V} = \{\mathcal{V}_t, \mathcal{V}_p\}$. Equipped with the unified vocabulary \mathcal{V} , various pose-related tasks can be formulated in a general format, where both input and output tokens are drawn from the same vocabulary, with the input optionally combined with the visual feature \mathbf{v} . These discrete tokens can represent natural language, 3D pose, or combination, depending on the specific task to be solved. This naturally enables UniPose to unify pose comprehension, generation, and editing in a unified manner.

During training, denote the visual embedding sequence as $\mathbf{v} = \{v^i \in \mathbb{R}^d\}_{i=1}^{L_v}$, the pose token sequence as $\mathbf{u} = \{u^i \in \mathcal{V}\}_{i=1}^{L_p}$, the text token sequence of single-pose description as $\mathbf{t} = \{t^i \in \mathcal{V}\}_{i=1}^{L_t}$, and the text token sequence of pose-difference description as $\mathbf{d} = \{d^i \in \mathcal{V}\}_{i=1}^{L_d}$, we apply distinct optimization objectives for each task, tailored to the specific type of input and the desired output, as follows:

- *Single-Pose Comprehension.* Single-pose comprehension aims to generate a pose description from a 3D pose or image. Formally, given the sequence \mathbf{v} , \mathbf{u} and \mathbf{t} as defined

¹The pose vocabulary \mathcal{V}_p preserves the order of the pose codebook \mathcal{B}_p . In implementation, we add two special tokens, $\langle p \rangle$ and $\langle p / \rangle$, which denotes the start and end of a pose sequence, into the vocabulary \mathcal{V}_p .

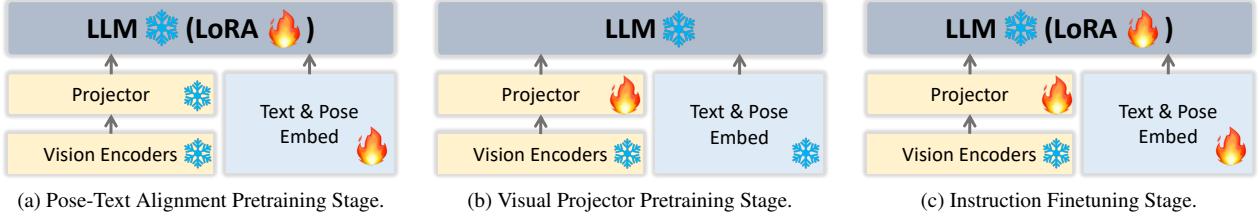


Figure 3. The training paradigm of UniPose.

above, the LLM predicts the probability distribution of potential next text token at each step, $p_\theta(t^i|\mathbf{v}/\mathbf{u}, t^{<i})$, conditioned on the visual or pose tokens in an autoregressive manner. The objective is to maximize the log-likelihood of this predicted pose description distribution:

$$\mathcal{L}_1 = \sum_{i=1}^{L_t} \log p_\theta(t^i|\mathbf{v}/\mathbf{u}, t^{<i}), \quad (3)$$

where θ represents the trainable parameters.

- **Pose-pair Comprehension.** Pose-pair comprehension aims to generate a textual description of the difference between a pair of 3D poses or images. Formally, given visual features \mathbf{v}_1 and \mathbf{v}_2 for an image pair, pose tokens \mathbf{u}_1 and \mathbf{u}_2 for a 3D pose pair, and pose-difference description tokens \mathbf{d} , the LLM predicts the probability distribution of the next pose-difference text token, $p_\theta(d^i|(\mathbf{v}_1, \mathbf{v}_2)/(\mathbf{u}_1, \mathbf{u}_2), d^{<i})$, conditioned on the pair of visual or pose tokens in an autoregressive manner. The objective is to maximize the log-likelihood of this predicted pose-difference description distribution:

$$\mathcal{L}_2 = \sum_{i=1}^{L_d} \log p_\theta(d^i|(\mathbf{v}_1, \mathbf{v}_2)/(\mathbf{u}_1, \mathbf{u}_2), d^{<i}). \quad (4)$$

- **Pose Generation.** Pose generation aims to generate 3D poses from pose textural descriptions or images. For this task, we use a *mixed attention mechanism* where the input pose tokens are replaced with predefined pose queries \mathcal{Q} . Formally, given \mathbf{v} , \mathbf{t} and \mathbf{u} as defined above, LLM predicts the probability distribution of potential *whole* pose tokens in a single step, $p_\theta(\mathbf{u}|\mathbf{v}/\mathbf{t}, \mathcal{Q})$, conditioned on the visual or pose-description text tokens and pose queries. The objective is to maximize the log-likelihood of this predicted pose distribution:

$$\mathcal{L}_3 = p_\theta(\mathbf{u}|\mathbf{v}/\mathbf{t}, \mathcal{Q}). \quad (5)$$

- **Pose editing.** Pose editing aims to generate a corrected 3D pose based on an initial pose and modification instruction. Similar to pose generation, a *mixed attention mechanism* is used for this task. Formally, given \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{d} as defined above, LLM predicts the probability distribution of potential *whole* pose tokens for the corrected pose, $p_\theta(\mathbf{u}_2|\mathbf{d}, \mathbf{u}_1, \mathcal{Q})$, conditioned on the initial pose tokens,

modification instruction tokens and pose queries. The objective is to maximize the log-likelihood of this predicted corrected-pose distribution:

$$\mathcal{L}_4 = p_\theta(\mathbf{u}_2|\mathbf{u}_1, \mathbf{d}, \mathcal{Q}). \quad (6)$$

At the last, given a batch size of inputs with different task types, the overall training loss is computed as the sum of the individual objections: $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4$.

3.3. Training and Inference Paradigm

The training procedure comprises four stages, and the training paradigm of the last three stages is shown in Fig. 3.

Pose Tokenizer Training. We first train a pose tokenizer using the objective \mathcal{L}_{vq} . The pose tokenizer encodes 3D pose as a sequence of discrete tokens, enabling seamless integration with texts within LLM. To maintain stability during LLM training, the pose tokenizer is kept frozen during the subsequent stages of training.

Pose-Text Alignment Pretraining. To enable LLM to handle discrete pose tokens, we train LLM on pose-text corpus. This process aims to align the pose and text modalities for unified reasoning within the LLM. In this stage, we consider four pose-text relevant tasks in Tab. 1, *i.e.*, 2 pose comprehension tasks (Pose-to-Text and Pose-Diff), 1 pose generation task (Text-to-Pose) and the Pose Editing task. Based on these tasks, we train LLM using LoRA [29] with the objective \mathcal{L} , as shown in Fig. 3a.

Visual Projector Pretraining. After establishing alignment between pose and text modalities, this training stage focuses on mapping images into the shared pose-text space. In this stage, we consider three image-text relevant tasks in Tab. 1, *i.e.*, 2 pose comprehension tasks (Image-to-Text and Image-Diff) and 1 pose generation task (Image-to-Pose). Based on these tasks, we train the vision-language projector to align visual data with language models with the objective \mathcal{L} , as shown in Fig. 3b.

Instruction Finetuning. To enhance the instruction-following capability of UniPose, we construct a multitask, multimodal instruction dataset with 200 templates for each task from Tab. 1. For example, an instruction for Image-to-Pose task could be “Could you estimate the SMPL pose of the individual in this image <image>”, with <image> standing for the image embedding extracted by the visual processor. Using this

Task	Dataset	Method	R-Precision ↑			Linguistic metrics ↑		
			Top-1	Top-2	Top-3	BLEU-4	ROUGE-L	METEOR
Pose-to-Text	PoseScript [14]	PoseScript [14]	91.6	95.6	97.0	12.9	33.9	34.2
		UniPose †	18.1	30.0	39.1	10.8	30.1	29.5
		UniPose	85.6	95.2	97.6	12.1	33.3	30.8
Pose-Diff	PoseFix [13]	PoseFix [13]	64.6	77.1	83.0	12.0	33.5	36.7
		UniPose †	8.4	14.6	19.2	8.5	28.2	27.3
		UniPose	67.9	81.8	88.6	13.8	33.7	31.2
Image-to-Text	ImageScript	LLaVA [44]	5.7	12.0	18.9	3.2	21.8	32.9
		Qwen-VL [4]	8.9	15.6	19.8	1.4	15.9	21.6
		GPT4V [1]	17.7	24.0	32.3	7.1	29.1	34.2
		UniPose †	22.4	32.8	41.2	18.2	42.4	45.2
		UniPose	24.5	35.4	43.2	18.2	42.5	44.7
Image-Diff	ImageDiff	GPT4V [1]	7.3	13.5	18.8	1.3	16.1	21.8
		UniPose †	13.0	18.8	26.4	14.0	34.1	40.1
		UniPose	13.5	25.0	33.8	15.9	36.5	39.6

Table 2. **Comparisons on pose comprehension tasks.** We compare the pose-retrieval precision (R-Precision) and linguistic metrics on various datasets. UniPose † represents training UniPose on the single corresponding task.

instruction data, we jointly train the visual projector and LLM with LoRA, as shown in Fig. 3c.

Inference. During inference, we adopt tailored decoding strategies according to task type. For pose comprehension tasks, we use a standard auto-regressive approach, where text tokens are generated sequentially, step-by-step. For pose generation and editing tasks, as shown in Fig. 2, once the model predicts the start_of_pose token $\langle p \rangle$, we append L_p predefined pose queries to the conditional text tokens, which is fed into LLM. Then LLM predicts the corresponding pose token for each query in parallel, which significantly accelerates its inference speed.

4. Experiments

4.1. Experimental Setup

Datasets. For pose tokenizer training, we use the standard training split of AMASS [48] and MOYO [60], following TokenHMR [16]. For UniPose training, we integrate three types of data: (1) Text-Pose Data. We use PoseScript [14] and PoseFix [13] datasets to link language and pose modality. PoseScript [14] provides natural language descriptions paired with 3D human poses, allowing the model to understand fine-grained pose semantics. PoseFix [13] includes pairs of 3D poses and textual descriptions that specify how to modify the source pose to achieve the target pose. (2) Image-Pose Data. Following [16, 24], we use standard human pose estimation training datasets, including Human3.6M [30], MPI-INF-3DHP [49], COCO [43], and the MPII [3] dataset, and evaluate on 3DPW [62] and Human3.6M [30] test sets. (3) Image-Text Data. Since no existing dataset combines human images with pose descriptions, we create the ImageScript and ImageDiff datasets to bridge this gap in visual-textual pose comprehension. Further dataset details are provided in the Appendix.

Metrics. We adopt the evaluation metrics from PoseScript [14] and PoseFix [13]. (1) **Pose comprehension tasks.** We use two types of metrics. Pose-text retrieval metric: *R-Precision*, which evaluates the accuracy of matching poses with corresponding descriptions. We rank the Euclidean distances between the query pose and 32 text descriptions (1 ground truth and 31 randomly selected mismatched descriptions), and report Top 1/2/3 R-Precision; Linguistic metrics: *BLEU-4* [51], *Rouge-L* [40] and *METEOR* [6], which assess the quality of generated pose descriptions. (2) **Pose generation tasks.** We use two types of metrics. Reconstruction metrics: *MPJPE* and *PA-MPJPE*, which computes the average per-joint position error between generated and ground-truth pose; Pose-text retrieval metric: following [19], we report Top 5/10/20 R^{T2P} and R^{P2T} , which represents the text-to-pose and pose-to-text retrieval recall, respectively. (3) **Pose editing tasks.** In addition to the reconstruction metrics, we also report the *Frechet Inception Distance (FID)*, which measures the distance between the generated and ground-truth pose distribution. To calculate these metrics, following [13, 14], we train a retrieval model with pose and text feature extractors using contrastive loss, which encourages matched pose-text pairs to have geometrically close feature vectors.

Implementation Details. For pose tokenizer, we set the codebook size to 2048 and each 3D pose is represented with 80 discrete tokens. We utilize LLaVA-1.6V [44] as our visual-language model backbone. For training the pose tokenizer, we use AdamW as the optimizer with a batch size of 256 and an initial learning rate of 2e-4. The pose tokenizer is trained for 240 epochs on a single RTX 4090 GPU. UniPose is trained 6 epochs in the Pose-Text Alignment Pre-training stage, and 2 epochs in the remaining stages using 4 A100 GPUs. Further implementation details are provided in the Appendix.

Method	R ^{T2P} ↑			R ^{P2T} ↑			Pose Reconstruction Metric ↓		
	Top-5	Top-10	Top-20	Top-5	Top-10	Top-20	MPJPE	PA-MPJPE	FID
PoseScript [14]	73.3	82.5	89.4	70.0	82.5	87.4	318.0	161.3	0.075
ChatPose [19]	17.6	25.3	35.8	28.0	39.0	54.4	-	-	-
ChatHuman [42]	41.8	52.6	65.1	42.1	52.3	66.5	-	-	-
UniPose †	67.5	77.6	85.5	62.8	74.8	83.6	342.7	190.0	0.046
UniPose	73.7	82.4	89.6	70.9	80.5	89.6	308.6	171.1	0.038

Table 3. Comparisons on Text-to-Pose generation task. The retrieval and reconstruction metrics are reported on PoseScript [14] dataset.

Method	3DPW [62] ↓		H3.6M [30] ↓	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
HMR [34]	130.0	76.7	88.0	56.8
PyMAF [76]	92.8	58.9	57.7	40.5
SMPLer [71]	73.7	43.4	45.2	32.4
HMR2.0 [24]	70.0	44.5	44.8	33.6
Zolly [65]	76.2	47.9	49.4	32.3
MEGA [21]	67.5	41.0	-	-
TokenHMR [16]	71.0	44.3	-	-
ChatPose [19]	163.6	81.9	126.0	82.4
UniPose †	97.4	61.2	65.8	39.4
UniPose	94.7	59.1	69.2	41.8

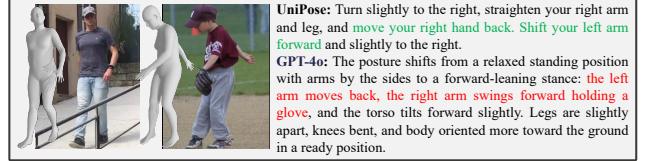
Table 4. Comparisons on pose estimation task. Reconstruction metrics are reported on the 3DPW and Human3.6M datasets.

4.2. Comparisons on Pose-relevant Tasks

Comparisons on Pose comprehension. We evaluate UniPose on 4 pose comprehension tasks, *i.e.*, Pose-to-Text, Pose-Diff, Image-to-Text and Image-Diff. The comparison results are shown in Tab. 2. As seen in the table, UniPose achieves competitive performance across all evaluated tasks, highlighting its capability to address diverse pose comprehension tasks within a single model. (1) For Pose-to-Text task, we compare UniPose with PoseScript [14] on the PoseScript dataset. As shown in Tab. 2, UniPose achieves slightly lower performance than PoseScript. However, PoseScript is tailored for single-pose description generation and lacks the capacity to model relationships between different poses. (2) For Pose-Diff task, we compare UniPose with PoseFix [13] on the PoseFix dataset. As shown in Tab. 2, UniPose outperforms PoseFix on most metrics, demonstrating its superiority in capturing relationships between pairs of poses. (3) For Image-to-Text task, we compare UniPose with existing visual-language MLLMs, including LLaVA [44], Qwen-VL [4] and GPT4V [1], on the ImageScript dataset. As shown in Tab. 2, UniPose significantly outperforms these MLLMs. The substantial gains can be attributed to the use of pose-specific visual encoder, which enables UniPose to extract fine-grained pose information from visual inputs. (4) For Image-Diff task, we compare UniPose with GPT4V on the ImageDiff dataset. UniPose still outperforms GPT4V, demonstrating that UniPose not only captures fine-grained pose features from a single image but also learns the relationships between human



(a) Comparison with Qwen-VL [4] and GPT-4o [1] on Image-to-Text task.



(b) Comparison with GPT-4o [1] on Image-Diff task.

Figure 4. Examples on Image-to-Text and Image-Diff tasks. We mark incorrect captions in red and correct in green. UniPose can accurately perceive a person’s orientation from images. poses across multiple images.

Fig. 4a and Fig. 4b visualize the generated textural descriptions of UniPose, Qwen-VL [4] and GPT4V [1]. The visualizations reveal that existing MLLMs struggle to comprehend fine-grained pose information. Specifically, Qwen-VL [4] and GPT4V [1] fail to distinguish human body orientation, whereas UniPose can accurately locate the human body orientation from visual inputs.

Comparisons on Pose Generation. We further evaluate UniPose on 2 pose generation tasks, *i.e.*, text-to-pose and pose estimation. The comparison results are shown in Tab. 3 and Tab. 4. (1) For Text-to-Pose task, we compare UniPose with existing text-conditional pose generation models [14, 19, 42] on PoseScript dataset. As shown in Tab. 3, UniPose achieves the best performance on most metrics. We attribute this to the use of the mixed-attention mechanism in LLM, which effectively captures the bidirectional dependencies among pose tokens, thus improving pose generation performance. (2) For pose estimation task, we compare UniPose with traditional pose estimation approaches [16, 24] and MLLM-based approaches [19], on 3DPW [62] and H3.6M [30] datasets. As shown in Tab. 4, UniPose largely outperforms other MLLMs, yet does not match the performance of methods specifically designed for estimating 3D human pose. This is not surprising, as traditional pose estimation methods have been developed over many

Method	MPJPE ↓	PA-MPJPE ↓	FID ↓
PoseFix [13]	300.2	144.1	0.019
UniPose †	310.8	157.0	0.019
UniPose	270.3	138.9	0.015

Table 5. Comparisons on pose editing task. Reconstruction metrics are reported on PoseFix [13] dataset.

CLIP-ViT	Pose-ViT	Pose Estimation ↓		Image-to-Text ↑		
		MPJPE	PA-MPJPE	BLEU-4	ROUGE-L	METEOR
✓	✗	193.4	86.1	11.1	30.2	33.9
✗	✓	96.3	59.1	12.5	31.0	34.8
✓	✓	96.1	58.9	13.3	31.7	35.2

Table 6. Ablation study on the components of the visual processor.

Attention Type	Text-to-Pose						Pose-to-Text			
	R ^{T2P} ↑			R ^{P2T} ↑			Latency (s) ↓	BLEU-4 ↑	ROUGE-L ↑	METEOR ↑
Causal Attention	9.0	14.2	20.8	9.3	14.7	22.3	2.5	26.9	39.5	38.0
Mixed Attention	13.8	20.3	28.8	15.9	23.0	32.0	0.2	25.0	39.1	36.7

Table 7. Ablation study on different attention mechanisms.

year and often incorporate custom network modules and loss functions to enhance estimation accuracy.

Comparisons on Pose Editing. For pose editing task, we compare UniPose with PoseFix [13] on PoseFix dataset. As shown in Tab. 5, UniPose significantly outperforms PoseFix across all metrics, validating its superiority in pose editing.

4.3. Ablation Studies & Analysis

Single-task training v.s. Multi-task training. Tab. 2, 3, 4, 5 also report the performance of UniPose training on single task (denoted as UniPose †). As shown, multi-task training consistently outperforms single-task training, underscoring the importance of unifying pose comprehending, generation and editing within a single model.

Visual Processor. We compare the impact of different vision encoders used in the Visual Processor of UniPose. In this part, the models are trained solely on Pose Estimation and Image-to-Text tasks for 2 epochs. As shown in Tab. 6, with only the CLIP-ViT encoder, the model performs poorly on pose estimation task. We argue that CLIP-ViT primarily focuses on aligning global semantic information between images and text, struggling to capture detailed human pose information. By incorporating an additional ViT model trained specifically for pose estimation, UniPose gains the ability to capture fine-grained pose details, significantly improving its performance on pose estimation task. Moreover, the pose information extracted from images enhances the performance on Image-to-Text task, enabling UniPose to generate more precise descriptions of human poses.

Attention mechanism. We evaluate the performance of UniPose using causal attention and mixed attention. In this part, the models are trained solely on Text-to-Pose and Pose-to-Text tasks for 6 epochs. More training details are provided in the Appendix. As shown in Tab. 7, on Text-to-Pose task, the model with mixed attention achieves higher retrieval accuracy compared to causal attention. The results indicate that capturing bidirectional dependencies among pose tokens enhances pose generation. Additionally, the

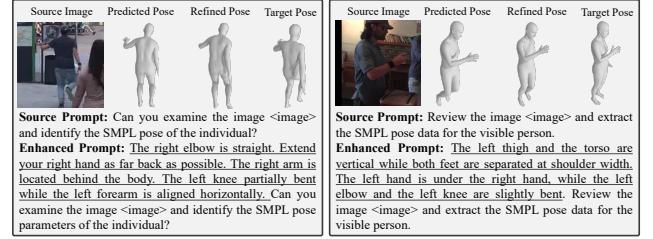


Figure 5. Enhance pose estimation with input pose description.

bidirectional attention mechanism enables single-step generation of all pose tokens, significantly accelerating inference. However, mixed attention performs worse than causal attention on Pose-to-Text task. This may be due to the interference of the bidirectional attention with the causal dependencies essential for text generation, potentially compromising the semantic precision of the generated content.

Zero-shot Task Analysis. Benefiting from a unified learning format, UniPose effectively transfers knowledge across different pose-relevant tasks and adapts to unseen tasks. Fig. 5 provides a zero-shot analysis: without additional training, UniPose can leverage pose descriptions to enhance its pose estimation results. This ability is especially advantageous in scenarios where ambiguity or occlusion affects accurate human pose estimation from images.

5. Conclusion

In this work, we present UniPose, the first attempt to integrate human pose comprehension, generation, and editing within a unified framework. By employing a pose tokenizer, we build a unified representation space that bridges 3D poses and texts, enabling seamless interactions across modalities. Additionally, the mixture-of-visual encoder captures intricate pose details, thereby enhancing fine-grained pose perceptions. The mixed-attention mechanism further enhances pose generation quality while significantly accelerating inference speed. Extensive evaluations across various pose-relevant tasks demonstrate the effectiveness of UniPose in pose comprehension, generation, and editing.

References

- [30] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, 2013. 6, 7, 1
- [31] AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b (2023). *arXiv*, 2023. 2, 3, 4
- [32] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *NeurIPS*, 36:20067–20079, 2023. 2
- [33] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472. IEEE, 2011. 5
- [34] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 3, 7
- [35] Hyounghun Kim, Abhay Zala, Graham Burri, and Mohit Bansal. Fixmynpose: Pose correctional captioning and retrieval. In *AAAI*, pages 13161–13170, 2021. 1, 3
- [36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3
- [37] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, pages 9579–9589, 2024. 3
- [38] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv*, 2024. 3
- [39] 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*, 2023. 2
- [40] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6
- [41] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Chang-wen Chen. Being comes from not-being: Open-vocabulary text-to-motion generation with wordless training. In *CVPR*, pages 23222–23231, 2023. 1, 3
- [42] Jing Lin, Yao Feng, Weiyang Liu, and Michael J Black. Chathuman: Language-driven 3d human understanding with retrieval-augmented tool reasoning. *arXiv*, 2024. 2, 7
- [43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6, 1
- [44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 3, 6, 7
- [45] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024. 3
- [46] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH*, 34(6):248:1–248:16, 2015. 1, 3
- [47] Mingshuang Luo, Ruibing Hou, Hong Chang, Zimo Liu, Yaowei Wang, and Shiguang Shan. M^3 gpt: An advanced multimodal, multitask framework for motion comprehension and generation. *NeurIPS*, 2024. 2
- [48] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451, 2019. 6
- [49] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, pages 506–516. IEEE, 2017. 6, 1
- [50] Arnab Mondal, Stefano Alletto, and Denis Tome. Hummuss: Human motion understanding using state space models. In *CVPR*, pages 2318–2330, 2024. 3
- [51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 6
- [52] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 3
- [53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 3
- [54] 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. *JMLR*, 21(140):1–67, 2020. 4
- [55] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, pages 11488–11499, 2021. 3
- [56] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv*, 2019. 4
- [57] Stephan Streuber, M Alejandra Quiros-Ramirez, Matthew Q Hill, Carina A Hahn, Silvia Zuffi, Alice O’Toole, and Michael J Black. Body talk: Crowdshaping realistic 3d avatars with words. *TOG*, 35(4):1–14, 2016. 3
- [58] Qingping Sun, Yanjun Wang, Ailing Zeng, Wanqi Yin, Chen Wei, Wenjia Wang, Haiyi Mei, Chi-Sing Leung, Ziwei Liu, Lei Yang, et al. Aios: All-in-one-stage expressive human pose and shape estimation. In *CVPR*, pages 1834–1843, 2024. 1
- [59] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv*, 2023. 3, 4
- [60] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *CVPR*, pages 4713–4725, 2023. 6

- [61] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 3
- [62] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, pages 601–617, 2018. 6, 7, 1
- [63] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*, pages 14549–14560, 2023. 2
- [64] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Lu Yuan, and Yu-Gang Jiang. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *CVPR*, pages 6312–6322, 2023. 2
- [65] Wenjia Wang, Yongtao Ge, Haiyi Mei, Zhongang Cai, Qingping Sun, Yanjun Wang, Chunhua Shen, Lei Yang, and Taku Komura. Zolly: Zoom focal length correctly for perspective-distorted human mesh reconstruction. In *ICCV*, pages 3925–3935, 2023. 7
- [66] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv*, 2023. 2
- [67] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv*, 2024. 2
- [68] Julong Wei, Shanshuai Yuan, Pengfei Li, Qingda Hu, Zhongxue Gan, and Wenchao Ding. Occllama: An occupancy-language-action generative world model for autonomous driving. *arXiv*, 2024. 2
- [69] Qi Wu, Yubo Zhao, Yifan Wang, Yu-Wing Tai, and Chi-Keung Tang. Motionllm: Multimodal motion-language learning with large language models. *arXiv*, 2024. 2
- [70] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv*, 2024. 3, 4
- [71] Xiangyu Xu, Lijuan Liu, and Shuicheng Yan. Smpler: Tam-ing transformers for monocular 3d human shape and pose estimation. *TPAMI*, 2023. 1, 3, 7
- [72] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv*, 2024. 3
- [73] Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. Uni-audio 1.5: Large language model-driven audio codec is a few-shot audio task learner. *arXiv*, 2024. 2, 4
- [74] Jiabo Ye, Haiyang Xu, Huawei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models, 2024. 2, 3
- [75] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv*, 2023. 2, 4
- [76] Hongwen Zhang, Yating Tian, Xinch Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021. 7
- [77] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In *AAAI*, pages 7368–7376, 2024. 2
- [78] Zhichen Zhao, Huimin Ma, and Shaodi You. Single image action recognition using semantic body part actions. In *ICCV*, pages 3391–3399, 2017. 2
- [79] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv*, 2024. 3, 4
- [80] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*, 2023. 3

UniPose: A Unified Multimodal Framework for Human Pose Comprehension, Generation and Editing

Supplementary Material

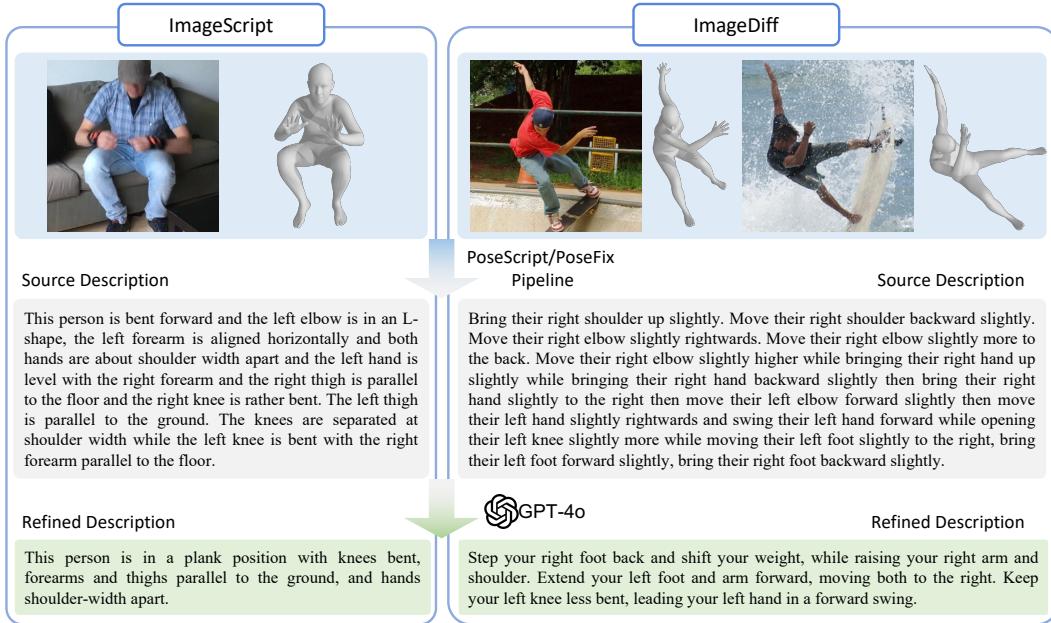


Figure 1. The annotation workflow for ImageScript (left) and ImageDiff (right) datasets.

In this Appendix, we present a comprehensive overview of UniPose, covering its datasets, implementation details, performance evaluation, and limitations. First, we introduce two new image-text datasets, ImageScript and ImageDiff, along with a detailed description of the training data used for UniPose (Sec. A). Next, we outline the implementation details of the pose tokenizer, retrieval models, and UniPose, including their architectural designs and training configurations (Sec. B). Additionally, we present experimental results to evaluate the performance of the tokenizer and retrieval models (Sec. C). Finally, we offer additional qualitative results (Sec. D) and conclude with an analysis of UniPose’s limitations (Sec. E).

A. Data Collection

To address the lack of datasets combining human images with pose descriptions, we present the ImageScript and ImageDiff datasets, specifically designed to bridge this gap in visual-textual pose comprehension.

A.1. ImageScript

ImageScript dataset aims to provide accurate and detailed textual descriptions of human poses depicted in images. Existing pose estimation datasets, collectively re-

ferred to as PoseEst (*e.g.*, Human3.6M [30], MPI-INF-3DHP [49], COCO [43], MPII [3], and 3DPW [62]) offer precise human poses paired with images. PoseScript [14] introduces a pipeline for automatically generating textual descriptions of human poses. Building on these efforts, our ImageScript dataset integrates human images, poses, and detailed textual descriptions to advance visual-textual pose comprehension.

The ImageScript dataset comprises 52k image-text pairs, with the images sourced from the PoseEst datasets. Following PoseScript [14], we first normalize the joint positions of each pose annotation from PoseEst datasets using the neutral SMPL body model [46], employing default shape coefficients and a global orientation of 0. To ensure diversity, we apply the farthest point sampling algorithm to select samples using the mean per joint error (MPJE) as the distance metric. Starting with a randomly selected pose, we iteratively add the pose with the highest MPJE to the selected set until the desired sample size is reached.

For textual annotations, we utilize the automatic pipeline from PoseScript to generate three diverse captions for each sampled pose. However, automatically generated captions often contain excessive detail and repetition, lacking the simplicity and fluency characteristic of human lan-

System Prompt:

As an AI text assistant specializing in human pose analysis, your task is to simplify AI-generated descriptions of human postures. These AI-generated descriptions often contain excessive detail and repetition. Your goal is to create concise summaries that highlight the key features of the posture in natural, fluent language. Focus on capturing the overall pose without unnecessary detail, ensuring the final description is brief and human-like, preferably under 45 words.

User Prompt:

Your goal is to summarize over-detailed and repetitive AI-generated descriptions in natural and fluent language, highlighting the key features of posture. Before formulating the pose description, think and answer the following questions:

1. Can multiple pose details be combined and simplified? (For example, Replace "The right knee and thigh are straight, and the left leg is also straight" with "Stand upright with both legs").
2. What action is the person performing? (e.g., running, jumping, kneeling on a single knee).
3. Can detailed descriptions be replaced with specific actions? (e.g., sit in meditation, side kick. avoid vague terms like "warming up" or "doing yoga").

You will be provided with AI-generated human pose descriptions. The description you generate should meet the following requirements:

1. Your description should be accurate, avoid using vague words such as "one leg", "one hand", etc. It is necessary to accurately specify left or right.
2. Your output should consist solely of the simplified description, and begin with "The person", "This person", "Someone" or other words that represent a person's personality.

Figure 2. Prompt to query GPT-4 for refining text in the ImageScript dataset.

System Prompt:

As an AI text assistant specializing in human pose analysis, your task is to simplify AI-generated descriptions of the changes between two human poses. Your goal is to create concise summaries that highlight the key features of the posture in natural, fluent language. Focus on capturing the overall pose without unnecessary detail, ensuring the final description is brief and human-like, preferably under 45 words.

User Prompt:

Note that your input will be a AI-generated description of the changes between two poses. Your goal is to summarize over-detailed and repetitive AI-generated description in natural and fluent language, highlighting the key features of posture. Before formulating the pose description, think and answer the following question:

1. Can multiple pose details be combined and simplified? (For example, Replace "move left foot higher, bring the left foot more to the front while moving their left knee forward slightly, move their left knee slightly to the right while bending their left knee." with "Kick your left leg up high in the air").
2. What is this person's action trend? (e.g., kicking high, straighten back, Push out chest).

You will be provided with AI-generated human pose descriptions. The description you generate should meet the following requirements:

1. Your output should consist of a simplified description, and include changes in human character orientation(e.g. turn left).
2. Your answer should not exceed 45 words.

Figure 3. Prompt to query GPT-4 for refining text in the ImageDiff dataset.

guage. To address this, we use GPT-4 [1] to refine the captions, transforming verbose and redundant descriptions into concise, natural expressions. Details of the query prompt and the annotation workflow are provided in Fig. 1 and Fig. 2, respectively.

Dataset statistics. The dataset generated using PoseScript's automatic pipeline is referred to ImageScript-A, while the GPT-4-refined version is named ImageScript-R. Image-pose pairs are initially sampled from Human3.6M (15k), MPI-INF-3DHP (25k), COCO (5k), and MPII (5k) datasets. Textual pose descriptions for each pose are then generated using the automatic pipeline, forming the ImageScript-A dataset. To construct the ImageScript-R training set, 6,250 examples are uniformly sampled from ImageScript-A. Additionally, 2000 samples from the 3DPW dataset are selected to create the ImageScript-R test set. The captions

in ImageScript-R are refined using GPT-4, transforming the automatically generated descriptions into more concise and natural expressions.

A.2. ImageDiff

ImageDiff dataset is designed to provide textual descriptions of human pose differences between image pairs, enabling the model to effectively perceive and interpret pose variations across different visual inputs. Building on PoseFix [13], which introduced a pipeline for automatically generating comparative descriptions for 3D SMPL pose pairs, we propose ImageDiff, a dataset comprising image pairs, corresponding 3D pose pairs, and textual descriptions of pose differences.

The ImageDiff dataset consists of 52k triplets in the form of $\{image\ A, image\ B, text\}$, where the text describes how

Training paradigm	Task	Dataset	Samples
Pose-Text Align Pretraining	Pose-to-Text, Pose-Diff, Text-to-Pose, Pose-Edit	PoseScript-A PoseFix-A	70k 93k
Visual Projector Pretraining	Image-to-Text, Image-Diff, Pose Estimation	ImageScript-A ImageDiff-A PoseEst	50k 50k 100k
Instruction Finetuning	All tasks	PoseScript-H PoseFix-H ImageScript-R ImageDiff-R PoseEst	5k 5k 6k 6k 6k

Table 1. **Detailed datasets for training UniPose.** The PoseScript dataset provides human annotations (PoseScript-H) and expands its dataset with automated captions (PoseScript-A), as does the PoseFix dataset.

Task	Sub-Task	Input	OutPut
Pose Comp	Pose-to-Text	Generate a description of the SMPL pose: <pose>. Interpret the SMPL pose in <pose> and generate a written description.	
	Pose-Diff	Provide a summary of how SMPL pose <pose> differs from <pose>. Detail any SMPL pose changes seen between <pose> and <pose>.	<caption>
	Image-to-Text	Describe the pose of the individual in the <image>. Analyze <image> and describe the posture displayed.	
	Image-Diff	Compare <image> and <image>, outline how the person’s posture differs. Identify how the individual’s pose varies from <image> to <image>.	
Pose Gen	Pose Estimation	Could you estimate the SMPL pose of the individual in <image>? Look at the <image> and return the SMPL pose parameters for the figure shown.	
	Text-to-Pose	Could you generate the SMPL pose from the description: <caption>? Using the description <caption>, please create the corresponding SMPL pose.	<pose>
Pose Editing		Modify <pose> based on this instruction: <caption>. Refine <pose> by applying the description provided: <caption>.	

Table 2. Examples of instruction templates utilized during the instruction finetuning stage of UniPose training.

to modify the human pose from image A (the source image) to match image B (the target image). The corresponding pose annotations for images A and B are denoted as poses A and B. The process for selecting image B is consistent with the approach used in the ImageScript dataset. For selecting image A, following PoseFix [13], we first calculate the cosine similarity between the pose retrieval features (Sec. B.2) of each pose B and all other poses in the PoseEst datasets. The top 100 poses with the highest similarity are shortlisted as candidates for pose A. To ensure diversity, we leverage posecode information [14] to verify that each pose pair exhibits at least 10 distinct low-level pose properties.

The pose difference descriptions are generated using the automatic annotation pipeline from PoseFix, producing three captions for each sampled pose pair. Similar to ImageScript, we use GPT-4 to refine these captions, transforming the automatically generated annotations into concise, easy-to-read descriptions. The query prompt and annotation workflow are detailed in Fig. 1 and Fig. 3 respectively.

Dataset statistics. The dataset generated using PoseFix’s automatic pipeline is referred to as ImageDiff-A, while the

GPT-4-refined version is named ImageDiff-R. Images B are initially sampled from Human3.6M (15k), MPI-INF-3DHP (25k), COCO (5k), and MPII (5k) datasets, following the same setup as ImageScript-A. Images A are subsequently selected from the corresponding dataset following the method mentioned above. The human pose difference descriptions for each image pair are then generated via the automatic pipeline to construct ImageDiff-A. For ImageDiff-R, 6,250 examples are uniformly sampled from ImageDiff-A to form the training set, and 2000 image pairs are sampled from the 3DPW dataset for the test set. Finally, GPT-4 is employed to refine the text descriptions in ImageDiff-R.

A.3. Training Data Details

We employ specific tasks and datasets for each training stage of UniPose, as summarized in Tab. 1. In details:

- **Pose-Text Alignment Pretraining Stage.** We incorporate four pose-text-related tasks: two pose comprehension tasks (Pose-to-Text and Pose-Diff), one pose generation task (Text-to-Pose), and the Pose-Edit task. Drawing in-

Configuration	Pose-Text Align Pretraining	Visual Projector Pretraining	Instruction Finetuning
Batch Size	24	8	8
Learning Rate	1.5e-4	5e-5	5e-5
Epochs	6	2	2
Image Res	$336 \times 336 / 256 \times 256$		
Patch Size	$14 \times 14 / 16 \times 16$		
Warmup Epochs	0.03		
LR Schedule	Cosine		
Optimizer	AdamW		

Table 3. **Training hyperparameters of UniPose.** Image Res denotes the input image resolution of CLIP-ViT and Pose-ViT, and the same as Patch Size.

spiration from the success of PoseScript [14] and PoseFix [13] in leveraging automatic captioning pipelines to scale datasets, we use PoseScript-A and PoseFix-A, both rich in automatically generated captions, as the training set. This extensive data effectively facilitates the alignment of pose and text modalities.

- **Visual Projector Pretraining Stage.** We include three image-related tasks: two pose comprehension tasks (Image-to-Text and Image-Diff), and one pose generation task (Image-to-Pose), using ImageScript-A, ImageDiff-A, and the PoseEst datasets for training.
- **Instruction Fine-tuning Stage.** In this stage, the model is trained across all tasks to ensure it understands and generates text aligned with human expression. The training process uses the PoseEst dataset, human-annotated datasets such as PoseScript-H and PoseFix-H, and GPT-refined datasets like ImageScript-R and ImageDiff-R. Additionally, we design task-specific instruction templates to enhance UniPose’s instruction-following capabilities, detailed in Tab. 2.

B. Implementation details

B.1. Pose Tokenizer

We provide a detailed explanation of the training objectives for the pose tokenizer. The pose tokenizer is trained using reconstruction loss \mathcal{L}_r , embedding loss \mathcal{L}_e , and commitment loss \mathcal{L}_c . To further improve the generated pose quality, we utilize vertices and position regularization in the reconstruction loss, as follows:

$$\mathcal{L}_{vq} = \mathcal{L}_r + \mathcal{L}_e + \mathcal{L}_c, \text{ where,}$$

$$\mathcal{L}_r = \lambda_1 \|\hat{\mathbf{p}} - \mathbf{p}\|_2 + \lambda_2 \|\hat{\mathbf{v}} - \mathbf{v}\|_2 + \lambda_3 \|\hat{\mathbf{j}} - \mathbf{j}\|_2, \quad (7)$$

$$\mathcal{L}_e = \|\mathbf{sg}[\mathbf{z}] - \hat{\mathbf{z}}\|_2^2, \quad \mathcal{L}_c = \|\mathbf{z} - \mathbf{sg}[\hat{\mathbf{z}}]\|_2^2,$$

where \mathbf{v} and \mathbf{j} denotes the ground truth SMPL mesh vertices and joints positions derived from \mathbf{p} , $\hat{\mathbf{v}}$ and $\hat{\mathbf{j}}$ denotes the predicted vertices and positions derived from $\hat{\mathbf{p}}$, $\mathbf{sg}[\cdot]$ is

the stop gradient operator, and λ_1 , λ_2 and λ_3 are the weighting factors.

Training Configurations. For the training of Pose Tokenizer, we use AdamW as the optimizer with a batch size of 256 and an initial learning rate of 2e-4. The model is trained for 240 epochs and the weighting factors λ_1 , λ_2 and λ_3 are set to 20, 100, 100 respectively. We set the codebook size to 2048, representing each 3D pose with 80 discrete tokens. Following TokenHMR [16], we augment random joints with noise starting at 0.01, progressively increasing after every 5K iterations. To further enhance robustness to global orientation variations, we introduce random perturbations of -45 to 45 degrees in the z-direction and -20 to 20 degrees in the x and y directions. The effect of global orientation noise is analyzed in Sec. C.

B.2. Retrieval Model

To compute the Pose-Text retrieval metric, a retrieval model is required to rank a large collection of poses based on their relevance to a given textual query, and vice versa.

Pose-Text Retrieval Model consists of a pose encoder and a text encoder. For pose feature extraction, we directly employ the pose encoder from the pose tokenizer and add 1D Conv for dimensionality reduction. For the text encoder, we use a bidirectional GRU [12] with one layer for text feature extraction, with word embeddings and the text tokenizer derived from a pretrained DistilBERT [56] model. Both pose and text are encoded into 512-dimensional feature vectors. Following PoseScript [14], we adopt the Batch-Based Classification (BBC) loss as the training objective:

$$\mathcal{L}_{BBC} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\gamma(x_i, y_i))}{\sum_j \exp(\gamma\delta(x_i, y_j))} \quad (8)$$

where γ is a learnable temperature parameter, δ is the cosine similarity function, and (x_i, y_i) denotes pose-text pairs.

Pose Pair-Text Retrieval Model is designed for retrieving pose pairs and text in the Pose/Image-Diff task. Its architecture is similar to the pose-text retrieval model, with the key difference being that the pose encoder processes each pose in the pair separately. The extracted features are concatenated along the channel dimension and passed through multiple 1D Conv layers for dimensionality reduction. Both the pose encoder and text encoder generate 512-dimensional feature vectors, utilizing the same training objective as the Pose-Text retrieval model.

Training Configurations. Following PoseScript and PoseFix, the retrieval models are first pretrained on automatically generated captions (PoseScript-A and PoseFix-A) and then fine-tuned on human-written captions (PoseScript-H and PoseFix-H). The retrieval models are trained for 120 epochs across the pretraining and fine-tuning stages. We use the Adam optimizer, with a batch size of 512 for pretraining and 32 for fine-tuning. The learning rate is set to 2e-4,

Method	$R^{P2T} \uparrow$			$R^{T2P} \uparrow$			mRecall
	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	
Pose-Text Retrieval							
PoseScript	22.3	50.1	62.9	22.1	51.4	63.1	45.3
UniPose	31.3	60.1	73.0	31.4	62.5	73.8	55.5
Pose Pair-Text Retrieval							
PoseFix	13.9	33.2	45.2	14.1	30.1	42.5	30.0
UniPose	15.7	34.0	44.7	15.2	34.0	44.6	31.3

Table 4. **The retrieval results on the PoseScript [14] and PoseFix [13] datasets.** We report Top 1 / 5 / 10 R^{P2T} and R^{T2P} , along with the mean recall (mRecall), which is the average of all retrieval recall values.

	AMASS \downarrow		MOYO \downarrow	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
w/o. Noise	6.7	3.8	32.6	11.7
w/. Noise	6.2	3.7	23.1	11.3

Table 5. **Ablation on global orientation noise for the Pose Tokenizer.**

and the learnable temperature parameter γ is initialized to 10. In the main text, all experiments use our proposed retrieval model, except for Text-to-Pose task, which utilizes the retrieval model from PoseScript [14].

B.3. UniPose

The detailed training hyperparameter settings for UniPose are provided in Tab. 3. In the Pose-Text Alignment Pretraining stage, UniPose is trained for 6 epochs with a batch size of 24 and a learning rate of 1.5e-4. For the Visual Projector Pretraining and Instruction Fine-tuning stages, the model is trained for 2 epochs with a batch size of 8 and a learning rate of 5e-5, respectively. Each stage includes a warm-up period of 0.03 epochs. We adopt the cosine learning rate schedule and use the AdamW optimizer. UniPose incorporates two vision encoders: CLIP-ViT and Pose-ViT, with the input image resolutions and patch sizes of 336 / 14 and 256 / 16 respectively. The output feature map of the Pose-ViT is resized using bilinear interpolation to ensure the visual token count aligns with that of the CLIP-ViT.

C. Additional Experiments

C.1. Retrieval Model

Tab. 4 shows the retrieval results on the PoseScript and PoseFix test sets. All methods are pretrained on automatic captions (PoseScript-A and PoseFix-A) and fine-tuned on human-written captions (PoseScript-H and PoseFix-H). Our Pose-Text retrieval model significantly outperforms PoseScript across all metrics, improving retrieval performance by over 10%. For Pose Pair-Text retrieval, our model also

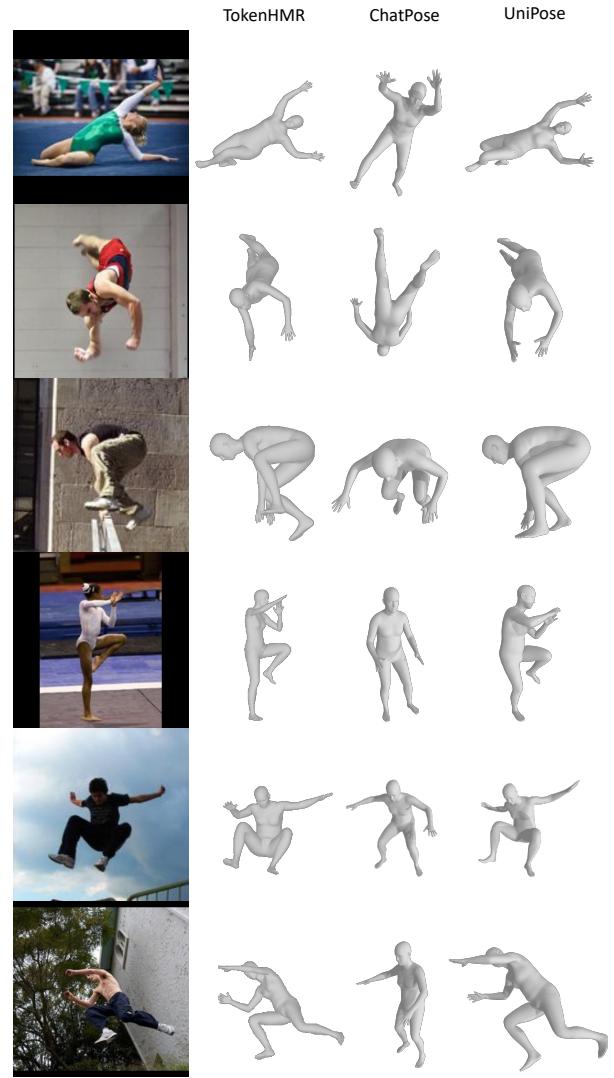


Figure 4. **Qualitative comparison on pose estimation task.** We compare multi-modal LLMs (ChatPose [19]) and traditional HMR methods (TokenHMR [16]) with our UniPose on LSP [33] dataset.

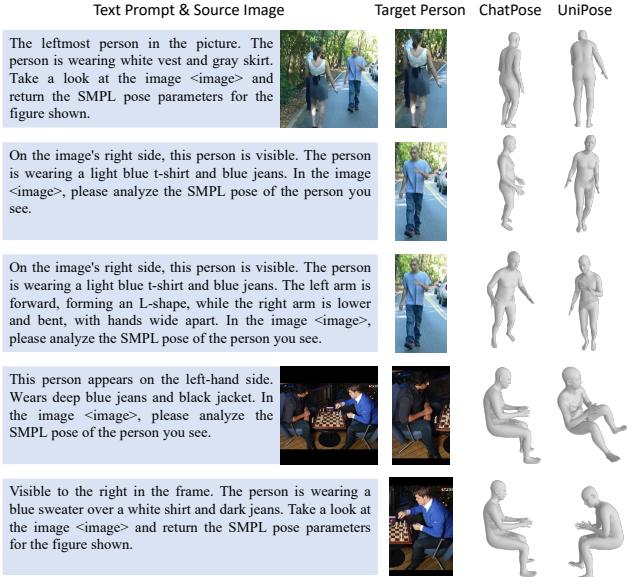


Figure 5. Qualitative comparison on reasoning-based pose estimation task. We evaluate the model’s reasoning capabilities in multi-person images.

achieves superior performance. The results demonstrate the effectiveness of our approach in aligning the pose representations with textual descriptions.

C.2. Pose Tokenizer

Tab. 5 illustrates the impact of global orientation noise on the Pose Tokenizer. All methods are trained on the standard training sets of AMASS [48] and MOYO [60], and evaluated on the AMASS test set and MOYO validation set. The results demonstrate that introducing random noise to global orientation enhances tokenizer robustness, particularly on the MOYO dataset, where MPJPE improves by 9.5. A stronger tokenizer benefits UniPose in handling various pose-related tasks. Therefore, we select the noise-augmented version as the final tokenizer.

D. Qualitative Evaluation

We present the qualitative results of UniPose on pose estimation tasks. In Fig. 4, we provide visualizations of UniPose’s performance on traditional pose estimation tasks, comparing it with both the traditional method TokenHMR [16] and MLLM-based method ChatPose [19]. The results show that our approach more accurately estimates human poses, even in scenarios with complex limb articulations.

In Fig. 5, we demonstrate UniPose’s performance on reasoning-based pose estimation tasks. For this, we select 8000 multi-person images from the PoseEst dataset and follow the annotation approach of ChatPose, leveraging GPT-4 [1] to label each individual’s behavior, clothes, and pose.

Fine-tuning UniPose on this dataset resulted in impressive reasoning capabilities, highlighting the model’s adaptability and generalization to new data.

E. Limitation

In pose estimation task, the performance of MLLMs-based models still lags behind specialized methods. We argue that these limitations may stem from the constraints imposed by the frozen visual encoder. Future research will focus on developing techniques that enable large language models to more effectively integrate pose-relevant visual features from diverse visual encoders, thereby enhancing their ability to handle complex pose estimation tasks.