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

Supplementary Material

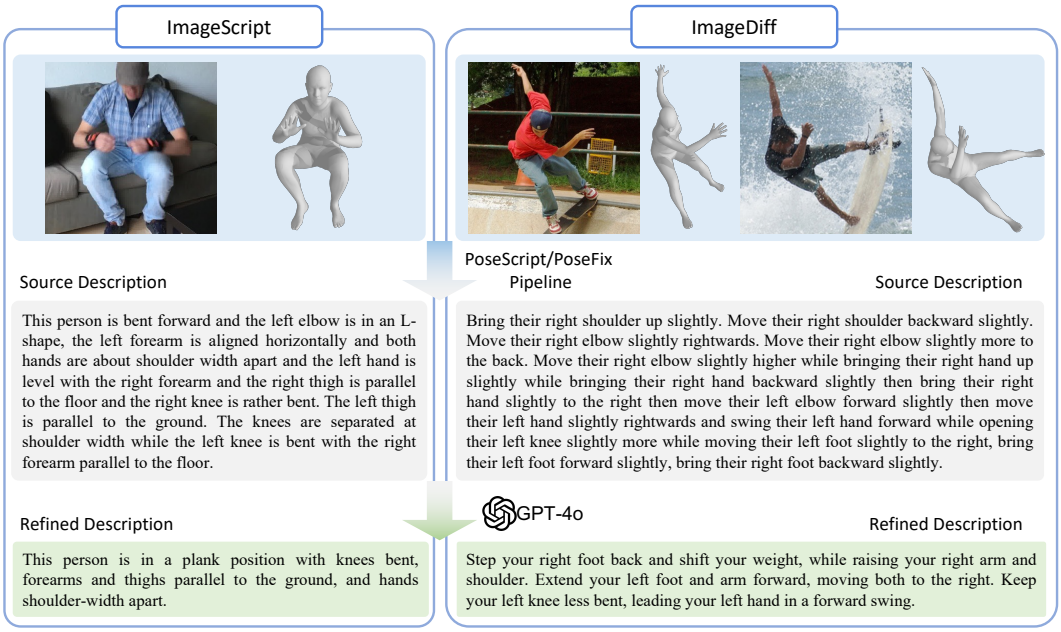

Figure 1. The annotation workflow for ImageScript (left) and ImageDiff (right) datasets.

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 referred to as PoseEst (*e.g.*, Human3.6M [8], MPI-INF-3DHP [13], COCO [10], MPII [2], and 3DPW [16]) offer precise human poses paired with images. PoseScript [5] 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 [5], we first normalize the joint positions of each pose annotation from PoseEst datasets using the neutral SMPL body model [11], 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 textural 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 language. To address this, we use GPT-4 [1] to refine the cap-

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.

tions, 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 [4], 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 to modify the human pose from image A (the source im-

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.	<caption>
	Pose-Diff	Provide a summary of how SMPL pose <pose> differs from <pose>. Detail any SMPL pose changes seen between <pose> and <pose>.	
	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.	<pose>
	Text-to-Pose	Could you generate the SMPL pose from the description: <caption>? Using the description <caption>, please create the corresponding SMPL 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.

age) 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 [4], 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 [5] 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 inspiration from the success of PoseScript [5] and PoseFix

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 × 336 / 256 × 256		
Patch Size	14 × 14 / 16 × 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.

[4] 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:

$$\begin{aligned} &\mathcal{L}_{vq} = \mathcal{L}_r + \mathcal{L}_e + \mathcal{L}_c, \text{ where,} \\ &\mathcal{L}_r = \lambda_1 \left\|\widehat{\boldsymbol{p}} - \boldsymbol{p}\right\|_2 + \lambda_2 \left\|\widehat{\boldsymbol{v}} - \boldsymbol{v}\right\|_2 + \lambda_3 \left\|\widehat{\boldsymbol{j}} - \boldsymbol{j}\right\|_2, \quad (1) \\ &\mathcal{L}_e = \left\|sg\left[\boldsymbol{z}\right] - \widehat{\boldsymbol{z}}\right\|_2^2, \quad \mathcal{L}_c = \left\|\boldsymbol{z} - sg\left[\widehat{\boldsymbol{z}}\right]\right\|_2^2, \end{aligned}$$

where $\boldsymbol{v}$ and $\boldsymbol{j}$ denotes the ground truth SMPL mesh vertices and joints positions derived from $\boldsymbol{p}$, $\widehat{\boldsymbol{v}}$ and $\widehat{\boldsymbol{j}}$ denotes the predicted vertices and positions derived from $\widehat{\boldsymbol{p}}$, $sg[\cdot]$ 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 [6], 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 [3] with one layer for text feature extraction, with word embeddings and the text tokenizer derived from a pretrained DistilBERT [14] model. Both pose and text are encoded into 512-dimensional feature vectors. Following PoseScript [5], we adopt the Batch-Based Classification (BBC) loss as the training objective:

$$\mathcal{L}_{BBC} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\gamma(x_i, y_i))}{\sum_j \exp(\gamma\delta(x_i, y_j))} \quad (2)$$

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, and the learnable temperature parameter γ is initialized to

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 [5] and PoseFix [4] 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 ↓		MOYO ↓	
	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.**

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 [5].

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 achieves superior performance. The results demonstrate the

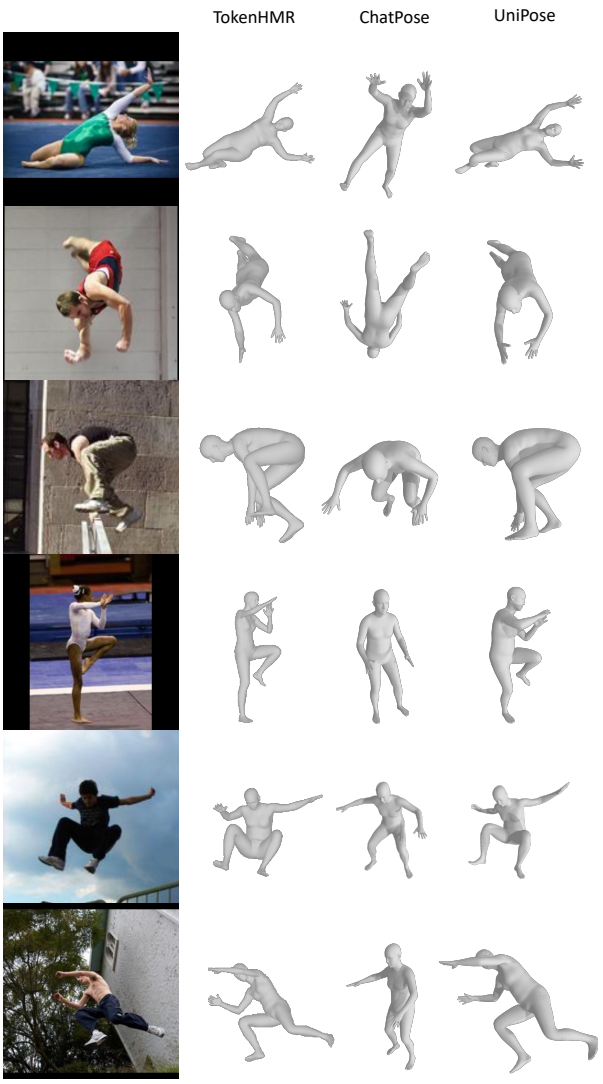

Figure 4. **Qualitative comparison on pose estimation task**. We compare multi-modal LLMs (ChatPose [7]) and traditional HMR methods (TokenHMR [6]) with our UniPose on LSP [9] dataset.

Task	R@3 ↓	BLEU-4 ↑	ROUGE-L ↑	METEOR ↑
Pose-to-Text	97.3 / 0.26	12.1 / 0.03	33.1 / 0.12	30.8 / 0.09
Pose-Diff	88.0 / 0.97	13.7 / 0.07	33.6 / 0.11	31.2 / 0.03
Image-to-Text	42.3 / 0.62	19.2 / 0.68	42.7 / 0.17	44.9 / 0.16
Image-Diff	35.0 / 0.89	15.7 / 0.12	36.2 / 0.21	39.5 / 0.05

Table 6. Means / Variances on pose comprehension tasks.

Task	MPJPE	PA-MPJPE	FID
Text-to-Pose	306.4 / 1.64	170.1 / 0.79	0.037 / 0.00
Pose-Edit	271.1 / 0.61	138.5 / 0.26	0.014 / 0.00
Pose-Estimation	94.6 / 0.05	59.1 / 0.03	-

Table 7. Means / Variances on pose generation and edit tasks.

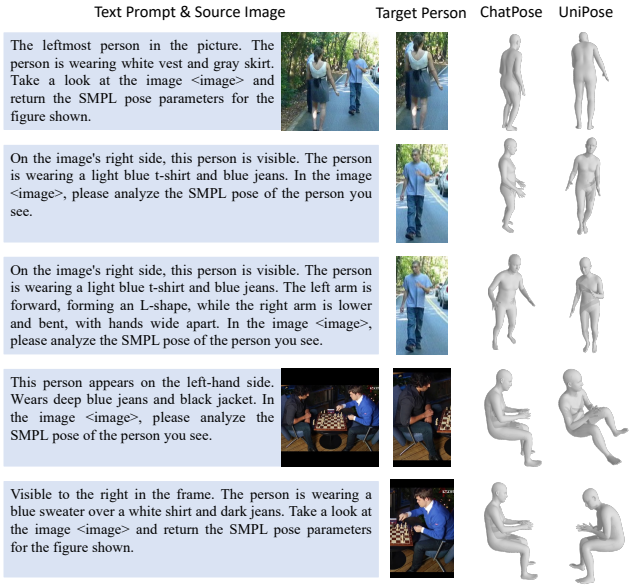

Figure 5. **Qualitative comparison on reasoning-based pose estimation task**. We evaluate the model's reasoning capabilities in multi-person images.

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 [12] and MOYO [15], 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.

Additionally, as shown in Tab. 6 and Tab. 7, we report the means and variances of all tasks across 3 experimental runs.

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 [6] and MLLM-based method ChatPose [7]. 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.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv*, 2023. 1, 6

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 1

[3] Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv*, 2014. 4

[4] Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. Posefix: correcting 3d human

poses with natural language. In *ICCV*, pages 15018–15028, 2023. 2, 3, 4, 5

[5] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. Posescript: Linking 3d human poses and natural language. *TPAMI*, 2024. 1, 3, 4, 5

[6] Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, Yao Feng, and Michael J Black. Tokenhmr: Advancing human mesh recovery with a tokenized pose representation. In *CVPR*, pages 1323–1333, 2024. 4, 5, 6

[7] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J Black. Chatpose: Chatting about 3d human pose. In *CVPR*, pages 2093–2103, 2024. 5, 6

[8] 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. 1

[9] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472. IEEE, 2011. 5

[10] 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. 1

[11] 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

[12] 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

[13] 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. 1

[14] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv*, 2019. 4

[15] 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

[16] 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. 1