LocLLM: Exploiting Generalizable Human Keypoint Localization via Large Language Model

Dongkai Wang Shiyu Xuan Shiliang Zhang
National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
{dongkai.wang, slzhang.jdl}@pku.edu.cn, shiyu_xuan@stu.pku.edu.cn

Abstract

The capacity of existing human keypoint localization models is limited by keypoint priors provided by the training data. To alleviate this restriction and pursue more general model, this work studies keypoint localization from a different perspective by reasoning locations based on keypiont clues in text descriptions. We propose LocLLM, the first Large-Language Model (LLM) based keypoint localization model that takes images and text instructions as inputs and outputs the desired keypoint coordinates. LocLLM leverages the strong reasoning capability of LLM and clues of keypoint type, location, and relationship in textual descriptions for keypoint localization. To effectively tune LocLLM, we construct localization-based instruction conversations to connect keypoint description with corresponding coordinates in input image, and fine-tune the whole model in a parameter-efficient training pipeline. LocLLM shows remarkable performance on standard 2D/3D keypoint localization benchmarks. Moreover, incorporating language clues into the localization makes LocLLM show superior flexibility and generalizable capability in cross dataset keypoint localization, and even detecting novel type of keypoints unseen during training[†].

1. Introduction

Human keypoint localization aims to locate target keypoints from input person image and is a fundamental task in computer vision and graphics. It has a wide range of applications in human pose estimation [26, 31–33] and facial landmark detection [23], *etc*. Existing keypoint localization methods typically utilize powerful neural networks, *e.g.*, Convolutional Neural Network (CNN) [26, 37] or Vision Transformer (ViT) [38] to either directly regress keypoint coordinates [11] or estimate the keypoint heatmaps [26, 37] to perform localization. Those methods learn keypoint priors encoded in the training set into the backbone, *e.g.*, along channel dimension in the last layer. This design reinforces the model responses on keypoints encoded in the backbone, but limits the generalization capability to detect keypoints in unseen human pose from different dataset, or to handle novel type of keypoint not included in the training set.

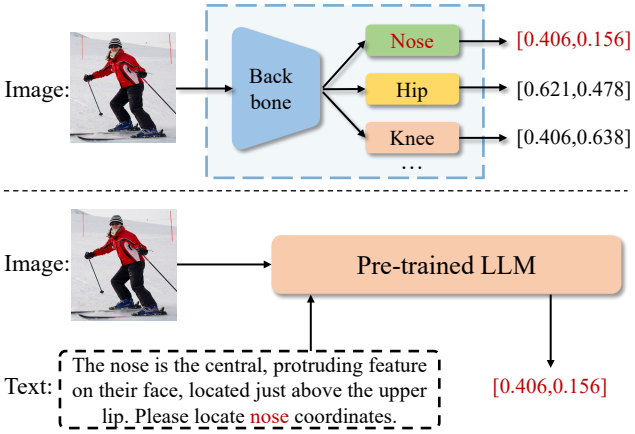

Figure 1. Upper: The conventional keypoint localization methods [11, 37, 38] encodes keypoint prior provided by the training set into model architecture and refers to encoded prior for keypoint localization. Bottom: The proposed LLM-based keypoint localization method refers to keypoint type, location, and relationship descriptions, and utilizes pre-trained powerful LLM [17, 43] to predict keypoint coordinates. Our method is more general to locate novel keypoints cross datasets, as textual descriptions can be provided flexibly.

To alleviate the restriction by the training set and pursue a more general model, this work aims to perform keypoint localization from a different perspective, *i.e.*, by referring to clues in textual descriptions that can be flexibly acquired. Inspired by the Large Language Model (LLM) [17, 43], we describe keypoint location through natural language and utilize the powerful reasoning capability of LLM for keypoint localization. Previous localization methods need to refer to encoded keypoint priors in network architecture, which is

[†]Project page: https://github.com/kennethwdk/LocLLM

hard to update. Differently, we explicitly send the keypoint name and location descriptions, along with input image to a LLM. Besides visual clues, this new pipeline allows to flexibly input descriptions of novel keypoints by indicating their type, location, and relationship with other keypoints. It also effectively adopts the reasoning capability of pre-trained LLM, therefore improving the generalization ability of keypoint localization.

The above intuition leads to our LocLLM, the first LLM-based localization model for generalizable keypoint localization. As illustrated in Fig. 1, LocLLM formulates the keypoint localization as a question-answer task, taking both image and text description as the input and outputting keypoint coordinates. LocLLM comprises a visual encoder, a projection layer to bridge image and text modalities and a pre-trained LLM. The visual encoder is responsible for learning image representations. The subsequent projector converts image representations into image tokens, which are combined with text tokens as the input of LLM. To effectively train LocLLM, we construct localization-based instruction conversations on existing keypoint localization benchmarks to connect the keypoint description with corresponding coordinates in input image. A parameter-efficient tuning method is proposed to effectively tune the whole model through instruction conversations.

We conduct extensive experiments on different keypoint localization benchmarks including 2D benchmarks like COCO Keypoint [16], MPII [30] and Human-Art [8], and 3D benchmark Human3.6M [7]. On standard COCO Keypoint benchmark, LocLLM achieves 77.4 AP, which is comparable with existing SoTA CNN and ViT-based localization methods. LocLLM can also detect 3D human keypoint and achieves promising performance on Human3.6M benchmark. Moreover, LocLLM shows superior generalization ability under various settings. On the cross dataset generalization setup, LocLLM achieves 33.4 PCKh@0.1 on MPII, which is better than ViTPose [38] by 7.8. On HumanArt, our method obtains 64.8 AP, outperforming previous methods by a large margin. Moreover, LocLLM can detect novel types of keypoint such as *pelvis* and *neck* through text descriptions, which are not included in the training set. All above experiments demonstrate that LocLLM is a superior generalizable keypoint localization method.

To the best of our knowledge, LocLLM is the first LLM-based keypoint localization model that explicitly exploits natural language description of keypoint into localization. This design allows us to utilize the rich clues from flexible text description and leverage pre-trained LLM for location reasoning. LocLLM achieves promising performance on standard 2D/3D human keypoint localization benchmarks, cross dataset generalization, and novel keypoint detection. LocLLM also incorporates keypoint localization into Multi-modal LLM, which enhances its capability in more fine-grained visual content analysis.

2. Related Work

2.1. Human Keypoint Localization

Human keypoint localization aims to locate the person keypoints from input RGB images and plays an important role in computer vision and graphics. Existing keypoint localization methods can be divided into two categories: heatmap-based and regression-based methods.

Heatmap-based keypoint localization encodes keypoint location with a probability map [29]. This type of methods estimates heatmaps and retrieves keypoint coordinates with a post-processing operation. Currently, heatmap-based methods dominate the field of keypoint localization because heatmap is easy to learn with CNN or Vision Transformer. Pioneer works [20, 26, 37] design powerful CNN models to estimate high resolution heatmaps for human pose estimation and facial landmark detection. From estimated heatmaps, the target keypoint can be simply obtained by a post-processing shifting [20, 41].

Regression-based keypoint localization directly outputs keypoint coordinates from input image via a neural network, which is adopted by several classical methods [2, 30]. Many works have been proposed to improve the performance of direct regression. The first kind of methods changes the way of regression. Soft-armgax [28] and Sampling-argmax [12] regress keypoint locations by integrating a latent heatmap, which is proved to be superior to direct regression. The second kind of methods improves regression by proposing new loss functions. RLE [11] changes the predefined Gaussian or Laplace distribution in commonly used regression loss with a learned distribution via normalizing flow. Finally, researchers also propose more powerful backbones to improve the performance of direct regression, such as TokenPose [14] and PETR [25].

All above methods directly encode keypoint type clues into architecture and implicitly learn the keypoint location through training data. Therefore, their generalization capability is restricted by the model architecture and training data. In contrast, our method explicitly exploits keypoint type and location from language description and LLM, making it more generalizable to detect novel keypoints.

2.2. Multi-modal Large Language Model

Large Language Model (LLM) shows remarkable reasoning capabilities in natural language processing tasks, therefore researchers try to enhance it with additional modalities, *e.g.*, image, audio, motion, *etc.*, to develop Multi-modal LLM (MLLM). Flamingo [1] proposes Perceiver to extract representative visual tokens and add them into LLM through cross-attention. BLIP-2 [13] proposes Q-Former to align visual features with text tokens in LLM.

Instruction tuning [36] is a commonly adopted way to align vision and language modalities to improve the ability of MLLM. Mini-GPT4 [43] and LLaVA [17] construct a high-quality instruction tuning dataset and fine-tune only a single fully connection layer to construct MLLMs. InstructBLIP [5] introduces an instruction-aware visual feature extraction method and fine-tunes the entire Q-Former, showing promising zero-shot performance on various multi-modal tasks. mPlug-Owl [40] incorporates a visual abstractor to align the two-modalities, and fine-tune both the visual encoder and visual abstractor during the pre-training stage. AnyMAL [19] aligns not only image but also more modalities, such as video, audio and IMU motion sensor to LLM.

2.3. Large Language Model for Vision Tasks

Despite remarkable progress in MLLM, most methods still focus on vision-language tasks, such as VQA and image caption. The effectiveness of LLM in classical vision tasks, *e.g.*, detection, segmentation and localization has not been fully exploited. LISA [10] defines a new vision task named as reasoning segmentation and propose a framework to extract referring text embedding from MLLM, which is sent to a segmentation model like SAM [9] to perform segmentation. VisionLLM [35] proposes a framework to address vision tasks such as detection and segmentation through LLM with a complex image tokenizer. However, none of above work exploits the effectiveness of LLM in keypoint localization task, which requires locating target in sub-pixel level accuracy, rather than the coarse object level in detection and segmentation. To the best of our knowledge, this is the first work that exploits LLM for keypoint localization and demonstrates that LLM can achieve superior performance in generalizable keypoint localization.

3. Method

3.1. Overview

The goal of human keypoint localization is to estimate the coordinates of target keypoints from input image, which can be conceptually denoted as,

$$\{\mathcal{K}_i\}_{i=1}^{n} = \text{locate}(\mathcal{I}), \tag{1}$$

where $\mathcal{K}_i$ denotes the coordinates of the i-th type keypoint, *e.g.*, person shoulder or knee, n is the total type of keypoint defined in each dataset. Following Eq. 1, previous methods [11, 37] encode keypoint clues into network architecture and learn location prior from training set, which limits their generalization ability.

Different from above formulation, in this paper we investigate keypoint location description and utilize the powerful large language model to perform localization, rewriting Eq. (1) into

$$\{\mathcal{K}_i\} = \text{locate}(\mathcal{I}, \mathcal{T}_i), \tag{2}$$

where $\mathcal{T}_i$ contains the text description of i-th target keypoint. In this way, the keypoint type is not solely encoded into the model but also through the text description input, allowing us to explicitly exploit keypoint type, location and relationship, and even detect novel keypoint.

Following Eq. (2) and most MLLM work [17, 39], we formulate the keypoint localization as a visual question answer (VQA) task and utilize the powerful reasoning capability of LLM to achieve our goal. Specifically, we propose LocLLM, a generative model that aims to complete multi-modal sentences to output keypoint coordinates. As shown in Fig. 2, LocLLM consists three main components: a visual encoder $\Phi_V(\cdot)$, a linear projector $\Phi_P(\cdot)$ and a large language model $\Phi_L(\cdot)$. The input of LocLLM contains two parts, the image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$ and text instruction $\mathcal{T}$. The output is target keypoint coordinates $\mathcal{K}$, this process can be denoted as,

$$\mathcal{K} = \Phi_L(\Phi_P(\Phi_V(\mathcal{I})), \mathcal{T}). \tag{3}$$

The visual encoder $\Phi_V(\cdot)$ takes an image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$ as input and outputs a sequence of image features $\mathcal{F} = (f_1, f_2, ..., f_m)$, where m is the number of image features. The image features are further projected into image tokens by a single linear layer projector $\Phi_P(\cdot)$ [17], *i.e.*,

$$\{v_1, v_2, ..., v_m\} = \Phi_P(\Phi_V(\mathcal{I})). \tag{4}$$

The text $\mathcal{T}$ will also be processed by the tokenizer of $\Phi_L(\cdot)$ to generate text tokens $\{t_1, ..., t_l\}$, which is combined with image tokens to be sent to $\Phi_L(\cdot)$. Utilizing the self-attention mechanism, the LLM is capable of understanding the contextual relationships between different types of tokens, enabling it to generate responses based on both text and image inputs. Formally, the output of LLM $\Phi_L(\cdot)$ is also a sequence, *i.e.*,

$$\{k_1, k_2, ..., k_s\} = \Phi_L(\{v_1, ..., v_m, t_1, ..., t_l\}), \tag{5}$$

where s denotes the length of output tokens, k_i is generated sequentially based on all previous tokens $\{v_1, ..., v_m, t_1, ..., t_l, k_1, ..., k_{i-1}\}$. Then k_i will be mapped to LLM vocabulary by a linear classifer $\mathcal{C}(\cdot)$. During training, we encode the keypoint coordinates $\mathcal{K}$ into ground truth vocabulary class sequence $\{k_1^*, k_2^*, ..., k_s^*\}$ and add standard Cross Entropy loss on the output classification score, which can be denoted as,

$$\mathcal{L} = \sum_i \textbf{CE}(\mathcal{C}(k_i), k_i^*). \tag{6}$$

During inference, we decode the output tokens to vocabulary words by selecting the words with the highest probability in $\mathcal{C}(k_i)$ to get estimated keypoint coordinates.

Previous works [17, 39, 43] reveal that the text instruction $\mathcal{T}$ plays a key role in unleashing the power of LLM to

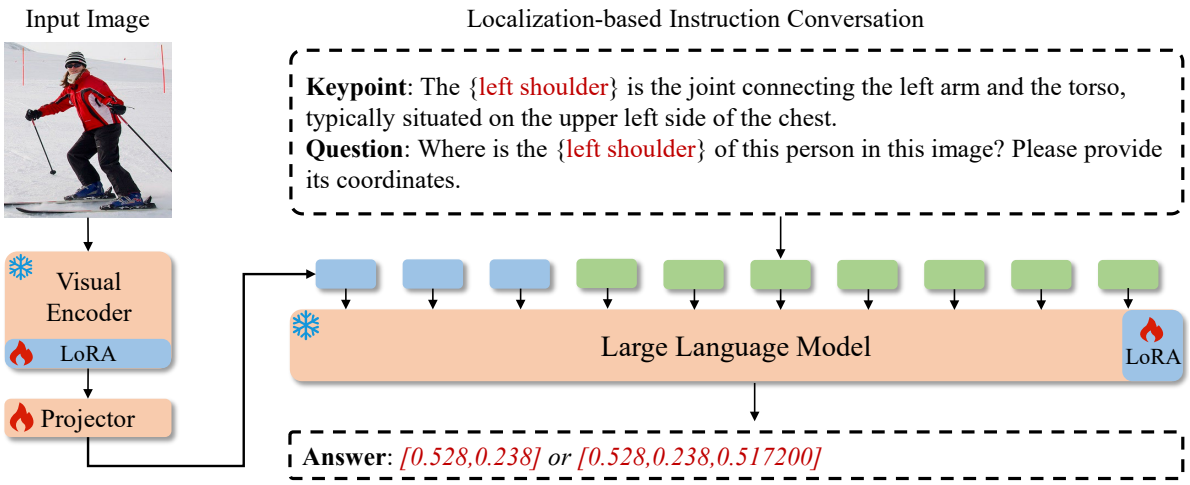

Figure 2. The proposed LocLLM for keypoint localization via large language model. LocLLM takes image and text instruction as input and contains three parts: a visual encoder, a projector and a decoder-only LLM. The image input is processed by visual encoder and projector to extract image tokens. The LLM takes the image tokens and text tokens as input and output corresponding keypoint coordinates. During training, we freeze the visual encoder and LLM and only update a small set of learnable parameters with projector, therefore relieving the training cost.

Instruction Template	
Image:	{image tokens}
Keypoint:	{keypoint location description}
Question:	{question to perform localization}
Answer:	{keypoint coordinates}

Table 1. Illustration of the proposed localization-based instruction conversation template.

complete corresponding tasks. Moreover, how to effectively tune LocLLM to generate accurate keypoint coordinates is also a challenge. Therefore, following two parts proceed to introduce the detailed localization-based instruction conversation construction to instruct LLM to perform keypoint localization, and a parameter-efficient tuning pipeline to train the LocLLM, respectively.

3.2. Localization-based Instruction Conversation

Constructing proper instructions is a key step to tune LLM towards a specific task, which is verified in many visual instruction tuning methods [17, 43]. To enable LLM perform keypoint localization accurately, we create the following localization-based instruction conversation as LocLLM input. The instruction template is shown in Table 1 and an example can be found in Fig. 2.

Keypoint Description. Different from previous work that only provide question, we additionally provide a sentence that describes the keypoint location on human body to help LLM locate target keypoint. For each type of keypoint, we ask ChatGPT to generate corresponding description and manually check it with Wikipedia. Manual intervention aims to ensure the descriptions are reliable. The generation and manual intervention are offline and no longer needed once the descriptions are checked. The detailed description of each keypoint can be found in supplemental materials and its effectiveness is verified in Sec. 4.

Keypoint Coordinates Format. One challenge in localization-based instruction conversation is how to format the keypoint coordinates so that LLM can predict them accurately. According to previous work OFA [34], Shikra [3] and Pink [39], we investigate two types of keypoint coordinates format in instruction conversation.

The first is adopting location token to represent keypoint coordinates, which is used in many methods such as OFA [34]. For example, a keypoint with $(95, 123)$ coordinates can be converted into two $\langle 095 \rangle$ $\langle 123 \rangle$ tokens. Considering that the image size is fixed, *e.g.*, 224×224, we can add a set of fixed location token into tokenizer to represent keypoint coordinate. However, the drawback of location token is that their embedding should be additionally learned, and it is hard to represent decimal coordinate such as depth in 3D keypoint representation.

The second is to directly adopt decimal string to represent keypoint coordinates. Specifically, we normalize the keypoint coordinates into the range $[0, 1]$ with respect to the image size in spatial dimension or camera 3D bounding box size in depth dimension and preserve 3/6 decimal places for each number, *i.e.*,

$$[0.abc, 0.def, 0.ghijkl], \tag{7}$$

where lowercase letters denote any number between 0 and 9. For decimal string the tokenizer will split it into a set of words, *e.g.*, "$0.abc$" into {"0", ".", "a", "b", "c"}, which is

already included in LLM vocabulary. Therefore, the advantage of the second format is that we do not need to add new tokens into LLM vocabulary and train corresponding embedding layer. Moreover, the decimal string format allow us to easily extend the framework to locate 3D keypoint with minimal modification, *i.e.*, just extend the string to include depth dimension.

Multi-Round Conversation. One image may contain multiple target keypoints, *e.g.*, knee and shoulder. Therefore it is not efficient to ask only one keypoint location in one conversation. To boost the training efficiency, we follow the VQA method to construct multi-round conversation to ask multiple keypoint locations for one input image in a single forward pass.

3.3. Parameter-Efficient Tuning

Due to the huge parameters of LLM, it is not feasible to fine-tune the entire model with limited GPU resource. Moreover, fully fine-tuning LLM and visual encoder requires millions of image-text pairs to avoid model collapse, which is unrealistic in keypoint localization task.

To perform an efficient training and enable the entire model to benefit from multi-modal localization-based instruction conversation, we freeze the visual encoder and LLM and introduce a small set of learnable parameters into them. This approach prevents the visual encoder and LLM from suffering semantic loss due to the limited instruction text data and provide a parameter-efficient training way to perform keypoint localization. Specifically, we adopt LoRA [6] to training LocLLM. Given a weight matrix $W \in \mathbb{R}^{d\times k}$ in pre-trained model , the LoRA is defined as follows,

$$\hat{W} = W + \Delta W = W + W_B W_A, \tag{8}$$

where $W_A \in \mathbb{R}^{r\times k}$ and $W_B \in \mathbb{R}^{d\times r}$ denote the weight matrices of LoRA module, r is the hidden dimension which is much smaller than d and k. W_B is initialized to zero to ensure that at the beginning of the training LoRA does not change the original output.

We add LoRA modules into both visual encoder and LLM, and fine-tune them with the linear projector. In our method, the whole trainable parameters are 8.7M, which is much smaller than the whole model and common CNN and ViT models. Following previous methods [17], LocLLM is trained in two stages. In this first stage, we align the image and text by only fine-tuning the projection layer on image-text pairs CC3M [24]. In the second stage, we freeze the visual encoder and LLM and fine-tune the newly added LoRA module and projector on the localization-based instruction conversations. Therefore, LocLLM can benefit from the multi-modal conversation and perform keypoint localization accurately.

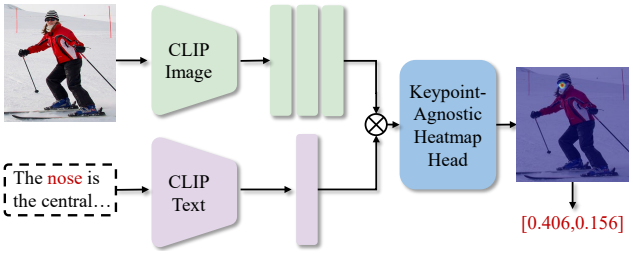

Figure 3. Illustration of the CLIP-based keypoint localization.

3.4. Baseline: CLIP-based Keypoint Localization

Another potential way to utilize keypoint description to guide localization is to adopt vision-language model such as CLIP [22]. Different from LLM, CLIP aligns the image and text feature space through millions of image-text pairs, so that we can extract text feature and use it to guide image feature extraction in conventional localization method. Therefore, we also propose a simple CLIP-based keypoint localization baseline to compare with LocLLM, with the aim to indicate that LLM is important in utilizing keypoint description for localization.

As shown in Fig. 3, we adopt CLIP image and text encoders to extract corresponding features from input image and text, which can be denoted as $\mathcal{F}_v$ and f_t. A text-conditioned feature map can be obtained by element-wise multiplication above two features, *i.e.*, $\mathcal{F}_v^t = \mathcal{F}_v \odot f_t$, which is sent to a keypoint-agnostic head to estimate corresponding heatmap. Note that the proposed baseline is different from CLAMP [42], which also adopts CLIP to locate keypoints. CLAMP only uses text to enhance the feature, and still uses n-channel heatmap head to estimate n heatmaps defined by training data. Therefore, it cannot be used to locate novel type keypoints that are out of training set. In contrast, the proposed CLIP baseline introduces the text-conditioned feature and keypoint-agnostic head to locate keypoint, thus is not limited to the fixed keypoint set in training set. Performance of this baseline is tested in Sec. 4.

4. Experiments

4.1. Datasets and Evaluation Metrics

We conduct experiments on different datasets, including image-text pair dataset CC3M [24], 2D human keypoint localization datasets COCO Keypoint [16], MPII [30] and HumanArt [8], and 3D human keypoint localization dataset Human3.6M [7].

Filtered CC3M [24] is constructed by LLaVA [24] and is the widely adopted visual instruction tuning dataset. It contains 595K image-text pairs. We adopt this dataset for the first stage training of LocLLM.

COCO Keypoint [16] contains 64K images of 270K persons labeled with 17 keypoints. Its `train` set contains 57K

images, 150K persons. The `val` set contains 5K images, 6.3K persons is used for evaluation. We adopt this dataset to construct 2D localization-based instruction conversation to train LocLLM at the second stage.

Human3.6M [7] is a large scale indoor benchmark for 3D human keypoint localization, which consists of 3.6 million images from 4 camera views. Following the standard protocols, We adopt subjects 1, 5, 6, 7, 8 to construct 3D localization-based instruction conversation to training LocLLM at the second stage and test model on subjects 9, 11. Besides above datasets, we further evaluate the generalization ability of LocLLM on other human keypoint localization datasets, *e.g.*, MPII [30] and HumanArt [8].

We follow the standard evaluation metric to report performance on each dataset. We report PCKh@0.5/0.1 on MPII and mAP on other 2D datasets. For 3D human keypoint localization, we report the Mean Per Joint Position Error (MPJPE) to evaluate the error of each method.

4.2. Implementation Details

Model Architecture. We adopt ViT-L/14 as visual encoder, which is pre-trained with DINOv2 [21] weights. For pre-trained LLM, we adopt an instruction-tuned model vicuna-7B [4].The projection layer is a single fully connection layer. The LoRA modules are inserted into the q and v of each self-attention layer of both visual encoder and LLM, with a hidden dimension $r = 8$.

Training Details. AdamW is adopted as the optimizer. In the first stage, the model is trained for 1 epoch with a batch size of 128 and weight decay of 0.0. After a warm-up period of 200 steps, the learning rate starts at 0.03 and decays to 0 with the cosine schedule. In the second stage, the model is trained on COCO Keypoint for 3 epochs for ablation study and 12 epochs for final comparison on 2D human keypoint localization. For 3D keypoint localization, we follow RLE [11] to train model on mixed Human3.6M and MPII. The model is trained with a batch size of 64 by gradient accumulation and weight decay of 0.05. The warm-up phase consists of 10k steps and the learning rate starts at 5e-4. The input image is resized to 224×224. Note that our model has only 8.7M trainable parameters, making it feasible to train with consumer GPUs, e.g., four 24G NVIDIA 3090s.

4.3. Ablation Study

Localization-based Instruction Conversation. We first analyze each component in constructing localization-based instruction conversation. Results are shown in Table 2. The keypoint description is useful to help LocLLM to locate keypoints. As discussed in Sec.3.2, there are two ways to represent keypoint coordinates, *i.e.*, discrete location token and decimal string. As shown in Table 2, we observe that decimal string is superior to location token. Observing the loss we can conclude that introducing location token into LLM requires to retrain the embedding layer, which is hard to learn from a small scale dataset.

Component	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Keypoint Description					
No description	70.8	91.4	78.0	68.0	75.8
With description	**72.4**	**92.4**	**80.1**	**69.2**	**77.4**
Keypoint Format					
Location token	67.2	91.4	75.9	64.2	71.9
Decimal string	**72.4**	**92.4**	**80.1**	**69.2**	**77.4**
Conversation Round					
Single	68.9	92.4	76.6	66.2	73.1
Multiple	**72.4**	**92.4**	**80.1**	**69.2**	**77.4**

Table 2. Component analysis of localization-based instruction conversation on COCO `val` set.

$\Phi_V(\cdot)$	$\Phi_P(\cdot)$	$\Phi_L(\cdot)$	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
	✓		39.5	77.1	36.2	38.5	40.9
✓	✓		70.3	92.3	78.0	67.6	74.4
	✓	✓	55.1	87.0	60.2	52.9	58.3
✓	✓	✓	**72.4**	**92.4**	**80.1**	**69.2**	**77.4**
Only second stage			70.6	92.0	78.9	67.8	75.3

Table 3. Ablation study on parameter-efficient tuning each compoint of LocLLM on COCO `val` set.

We also investigate the effectiveness of conversation round. Training LocLLM with single-round conversation achieves 68.9 AP on COCO `val` set, this can be improved to 72.4 AP when adopting a multi-round conversation paradigm during training. This indicates that providing more examples can help the model to perform better on human keypoint localization task.

Paramter-Efficient Tuning. The way of tuning the LLM is also important to the final performance. Due to the limited GPU resource and annotation, we could not finetune the whole model. Therefore, we insert some leanable parameter modules into the model to conduct parameter-efficient tuning. This experiment investigates the effects of insert locations of learnable parameter module to the performance of LocLLM. As shown in Table 3, only training projector layer cannot learn much information from the data and performs badly on COCO `val` set. Inserting learnable module into either visual encoder or LLM can both substantially improve the localization performance. Among them, we find that tuning both visual encoder and LLM achieves the best performance.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Heatmap-based						
SimplePose [37]	74.4	92.6	82.5	71.5	79.2	77.6
HRNet [26]	76.8	93.6	83.6	74.0	81.5	79.6
SimCC [15]	76.5	93.2	83.1	73.6	81.5	79.7
ViTPose [38]	77.4	93.6	84.8	**74.7**	**81.9**	80.2
Regression-based						
DeepPose [30]	53.8	82.6	59.2	52.2	57.3	66.8
RLE [11]	74.0	91.5	81.6	70.9	78.5	76.8
Language-based						
CLIP baseline	73.1	92.5	81.3	70.3	77.4	76.5
Ours (LocLLM)	**77.4**	**94.4**	**85.2**	74.5	81.8	**80.6**

Table 4. Comparison with other methods on COCO Keypoint `val` set in 2D keypoint localization. All results are obtained by evaluating the official model weights provided by the authors using GT bbox without flip test.

Method	Eat	Pose	Sit	Wait	Walk	Avg
Sun *et al.* [27]	54.2	53.1	71.7	53.4	47.1	59.1
PoseNet [18]	50.1	46.8	61.9	49.9	41.8	53.3
Sun *et al.* [28]	49.5	43.8	58.9	47.8	38.9	49.6
RLE [11]	44.5	43.1	59.2	44.1	**37.5**	48.6
Ours (LocLLM)	**41.2**	**40.0**	**53.6**	**41.8**	37.8	**46.6**

Table 5. Comparison with other methods on Human3.6M benchmark in monocular 3D keypoint localization.

4.4. 2D/3D Human Keypoint Localization

This section demonstrates that LLM can perform well on conventional keypoint localization tasks, including 2D human pose estimation and 3D human pose estimation. We compare LocLLM with recent methods on COCO Keypoint for 2D keypoint localization and Human3.6M for 3D keypoint localization. Results are shown in Table 4 and Table 5.

We first compare LocLLM with other methods in 2D human pose estimation task and report performance on COCO Keypoint `val` set. As shown in Table 4, existing 2D human pose estimation methods can be divided into heatmap-based method: SimplePose [37], HRNet [26], SimCC [15] and ViTPose [38], and regression-based methods, including: DeepPose [30] and RLE [11]. Our method can be viewed as a language-based method that utilize text keypoint description to locate keypoint position, therefore we also report the performance of CLIP baseline in Fig. 3. As shown in Table 4, our method achieves superior performance on COCO `val` set, which is comparable to recent SoTA methods such as ViTPose and RLE. With a few learnable parameters (8.7 M), LocLLM can achieve comparable performance with recent SoTA methods.

In Table 5 we further demonstrate that LocLLM can perform 3D keypoint localization in monocular RGB image, which is rarely exploited by previous MLLM methods. Benefited by the decimal string representation of keypoint coordinates, LocLLM can be easily extended to 3D keypoint localization by simply adding a depth dimension in outputs. We follow RLE [11] to conduct experiments and compare with previous methods. As shown in Table 5, LocLLM achieves superior 3D keypoint localization performance, indicating its capacity in depth understanding.

4.5. Cross Dataset Generalization

This section aims to evaluate the generalization ability of LocLLM on locating keypoint for unseen human pose from other datasets. We conduct cross dataset validation to test LocLLM trained on COCO on various different human pose estimation benchmarks such as Human-Art [8] and MPII [30]. The results are shown in Table 6.

Human-Art is a benchmark that contains human pose in both natural scenes such as sports or outdoor, and artificial scenes including cartoon, digital art, ink painting and *etc*. It is suitable to evaluate the generalization ability of keypoint localization methods. As shown in Table 6, we compare LocLLM with previous conventional keypoint localization methods in Table 4. Conventional localization methods achieves promising performance on COCO, but suffer a large performance drop on Human-Art. For example, ViTPose achieves 77.4 AP on COCO, but only obtains 53.8 AP on Human-Art. In contrast, our method achieves 64.8 AP on Human-Art, significantly better than compared methods. This indicates the superior generalization ability of our method in cross dataset validation.

4.6. Novel Keypoint Localization

We finally show that LocLLM can even locate novel type of keypoint that are never seen during training, to demonstrate its superior generalization ability. Existing methods encode the keypoint prior into the network architecture, making them hard to generalize to unseen keypoints. Our LocLLM is not subject to this restriction and can locate novel type of keypoints by referring to text descriptions.

To verify this claim, we conduct two experiments on COCO Keypoint and MPII. For the first experiment, we remove 4 keypoints (*right elbow*, *left wrist*, *left knee*, *right ankle*) from total keypoints (17) during training and do not apply any data-augmentation. In other words, 13 types of keypoints are used for training, and the model is tested to detect 17 keypoints. The results are shown in Table 7, where "Full keypoint" uses all keypoints for training, hence is regarded as the upper bound. By removing 4 types of keypoint from training, the performance of CLIP basline deteriorates greatly, *e.g.*, from 73.1 to 43.1. In contrast, our LocLLM can still achieve a reasonably good performance.

Method	Human-Art						MPII					
	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	Shou.	Elbo.	Hip	Knee	Mean	Mean0.1
SimplePose [37]	48.4	73.0	50.7	27.2	50.7	52.8	93.7	85.6	85.4	81.6	84.7	22.2
SimCC [15]	51.7	75.2	54.8	26.2	54.3	57.0	92.2	84.1	82.8	80.5	83.2	28.5
HRNet [26]	53.4	76.3	56.5	30.4	55.9	57.5	93.4	86.1	85.0	81.9	85.8	26.9
ViTPose [38]	53.8	77.9	57.4	31.4	56.6	58.7	94.5	88.2	87.3	85.0	86.9	25.8
CLIP baseline	49.3	75.7	51.8	27.7	52.0	54.5	95.5	88.8	87.5	84.7	87.1	25.0
Ours	**64.8**	**87.4**	**70.4**	**40.9**	**67.4**	**69.3**	**96.1**	**90.3**	**89.8**	**88.0**	**89.3**	**33.4**

Table 6. Comparison with other methods on cross dataset generalization on Human-Art and MPII. All models are trained on COCO Keypoint `train` set. We only evaluate the accuracy of keypoints that appear in COCO Keypoint.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Full keypoint	72.4	92.4	80.1	69.2	77.4
CLIP baseline	43.0	89.2	28.7	42.0	44.9
Ours	67.1	91.2	76.3	64.9	71.1

Table 7. Results of removing 4 keypoints from training and testing on COCO Keypoint `val` set.

Method	Seen				Unseen	
	Shou.	Elbo.	Hip	Knee	Pelvis	Neck
CLIP baseline	95.5	88.8	87.5	84.7	1.7	5.6
Ours	96.1	90.3	89.8	88.0	**43.6**	**36.9**

Table 8. Comparison with other methods on novel keypoint localization on MPII.

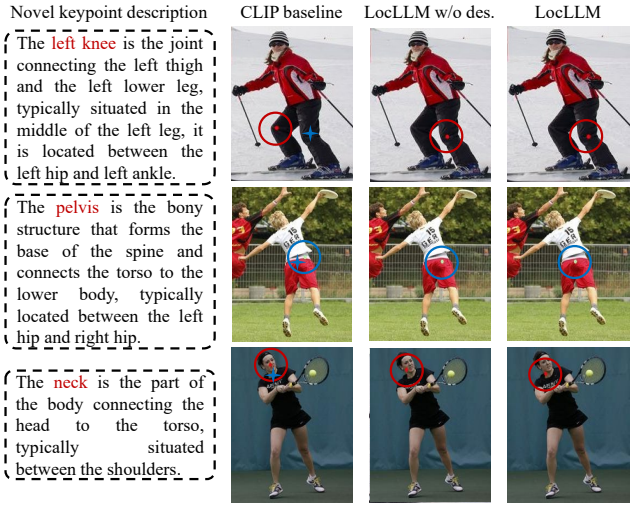

Figure 4. Localization results of three novel keypoints which are not seen during training (denoted by blue star in the first column image). It can be observed that CLIP baseline matches each novel keypoint to a similar keypoint in the training set, *e.g.*, it locates the left knee to right knee, pelvis to right hip, and neck to nose. In contrast, our LocLLM can locate novel keypoint accurately.

For the second experiment, we use all 17 keypoints of COCO Keypoint for training, then test the model on another dataset with different keypoint definition, *i.e.*, the MPII dataset. Note that, the *Pelvis* and *Neck* keypoints in MPII are not seen by model trained on COCO Keypoint. We hence also report the performance on them. As shown in Table 8, our LocLLM achieves 43.6 accuracy on *Pelvis*, which is much better than the CLIP baseline with only 1.7 accuracy. In Fig. 4 we show some examples on novel keypoint localization. It can be observed that our method can accurately locate novel keypoints with the help of text description. Previous method fails to locate them and is confused with similar keypoints which are appeared during training.

5. Conclusion and Discussion

In this paper we introduce the first LLM-based keypoint localization model named LocLLM. Different from previous work that encodes keypoint priors into the model architecture and implicitly learn keypoint relationship from training data, LocLLM explicitly encodes keypoint type and relationship through language description, and utilizes the powerful reasoning capability of LLM to locate keypoints. We conduct experiments on different localization tasks to show the superior generalization ability of LocLLM. As shown in experiments, LocLLM performs well in detecting keypoints from unseen human pose, and locating novel type of keypoints unseen during training. We hope this work inspire future research on generalizable keypoint localization.

Our method can be improved in several aspects. First, the effectiveness of LocLLM relies on accurate textual descriptions. The effectiveness of LocLLM on keypoints that are hard to be described in language remain to be explored, such as facial landmark detection. Second, the huge parameters of LLM require considerable GPU resource to process a large batch of images, hence degrades its efficiency.

Acknowledgement This work is supported in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011, in part by the Okawa Foundation Research Award.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 35: 23716–23736, 2022. 2

[2] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2016. 2

[3] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 4

[4] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 6

[5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 3

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 2013. 2, 5, 6

[8] Xuan Ju, Ailing Zeng, Jianan Wang, Qiang Xu, and Lei Zhang. Human-art: A versatile human-centric dataset bridging natural and artificial scenes. In *CVPR*, 2023. 2, 5, 6, 7

[9] 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. *arXiv preprint arXiv:2304.02643*, 2023. 3

[10] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 3

[11] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, 2021. 1, 2, 3, 6, 7

[12] Jiefeng Li, Tong Chen, Ruiqi Shi, Yujing Lou, Yong-Lu Li, and Cewu Lu. Localization with sampling-argmax. *NeurIPS*, 2021. 2

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[14] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *CVPR*, 2021. 2

[15] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. Simcc: A simple coordinate classification perspective for human pose estimation. In *ECCV*, 2022. 7, 8

[16] 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*, 2014. 2, 5

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 3, 4, 5

[18] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019. 7

[19] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. Anymal: An efficient and scalable any-modality augmented language model. *arXiv preprint arXiv:2309.16058*, 2023. 3

[20] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 2

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[22] 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*, 2021. 5

[23] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCVW*, 2013. 1

[24] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5

[25] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *CVPR*, 2022. 2

[26] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 1, 2, 7, 8

[27] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *ICCV*, 2017. 7

[28] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 2, 7

[29] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *NeurIPS*, 2014. 2

[30] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 2, 5, 6, 7

[31] Dongkai Wang and Shiliang Zhang. 3d human mesh recovery with sequentially global rotation estimation. In *CVPR*, 2023. 1

[32] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for instance-level human analysis. *IEEE TPAMI*, 2023.

[33] Dongkai Wang, Shiliang Zhang, Yaowei Wang, Yonghong Tian, Tiejun Huang, and Wen Gao. Humvis: Human-centric visual analysis system. In *ACM MM*, 2023. 1

[34] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, 2022. 4

[35] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 3

[36] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3

[37] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 1, 2, 3, 7, 8

[38] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 1, 2, 7, 8

[39] Shiyu Xuan, Qingpei Guo, Ming Yang, and Shiliang Zhang. Pink: Unveiling the power of referential comprehension for multi-modal llms. *arXiv preprint arXiv:2310.00582*, 2023. 3, 4

[40] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 3

[41] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, 2020. 2

[42] Xu Zhang, Wen Wang, Zhe Chen, Yufei Xu, Jing Zhang, and Dacheng Tao. Clamp: Prompt-based contrastive learning for connecting language and animal pose. In *CVPR*, 2023. 5

[43] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 3, 4