CapeLLM: Support-Free Category-Agnostic Pose Estimation with Multimodal Large Language Models

Supplementary Material

A. Keypoint Descriptions

We create the names and descriptions of keypoints for all 100 categories. The names can be divided into two types: one that has its own unique name, e.g., `left shoulder`, `right eye`, and the other that does not have its own name. the latter is difficult to define due to the densely distributed position. We concentrate on designating the latter and determine the names using their relative positions in each category; for example, "upper", "central", "lower". The descriptions are represented with the keypoint position in the category and its relation with other keypoints; e.g., in the *animal body*, the description of `left front paw` is defined as "The left front paw is the lower end of the left forelimb, used for movement and manipulation of objects. It is positioned below the left elbow and connected with the left elbow". A detailed example can be found in Table 19.

B. Exploring other Design Choices

B.1. Instruction

w/ description	w/ keypoint list	PCK@0.05	PCK@0.2	mPCK
×	×	72.60	96.22	89.86
✓	×	**78.43**	**96.98**	**91.98**
✓	✓	77.36	95.80	90.97

Table 10. Effect of additional info for keypoints in training. Default config .

Diverse questions	Add conversation outline	PCK@0.05	PCK@0.2	mPCK
×	×	**78.43**	**96.98**	**91.98**
✓	×	74.24	96.56	90.56
×	✓	75.08	96.27	90.63
✓	✓	68.24	95.93	88.52

Table 11. Effect of adding a conversation outline and diversifying question expressions. Default config .

Instruction variations As mentioned in Sec 3.2, we include not only the names but also descriptions of the keypoints in the instructions to help the model better to reason the location of keypoints. We examine how the description affects model performance by training the model without descriptions. The result in Table 10 shows that without descriptions, the accuracy decreases over 2%p in mPCK, suggesting that the keypoint description plays a significant role in enhancing to find the exact position. We experiment another scenario to include all keypoint names for each category in the instruction as "Keypoint List". As shown in Table 10, unlike keypoint descriptions, the list of keypoint names is not helpful for improving the model, rather reducing its performance. Next, we explore whether two optional conditions affect the performance or not: one is encompassing a conversation outline [17] and the other is to diversify the question expression in instruction. The outline slightly modified from the prior work [17] seems not to influence to solve the problem that predicts coordinates, and the random question does not have any positive effect on the performance, actually leading to a decrease in the model's performance(Table 11).

Multi-round	PCK0.05	PCK0.10	PCK0.15	PCK0.20	PCK0.25	mPCK
$k = 1$	78.29	91.55	95.19	96.89	97.88	91.96
$k = 2$	72.82	88.06	92.79	95.30	96.56	89.11
$k = 4$	**78.43**	**91.34**	**95.26**	**96.98**	**97.90**	**91.98**
$k = 6$	74.33	89.82	94.17	96.36	97.46	90.43
$k = 8$	75.28	89.89	93.99	96.16	97.41	90.55

Table 12. Ablation in multi-round k. Default config .

Choice of round k We investigate the optimal number of rounds k in the conversation. Table 12 shows that under the same training conditions, the highest performance was observed when k is set to 4. No explicit tendency was found as k changed.

LLM	Step-by-step instruction	PCK@0.05	PCK@0.2	mPCK
Llama3.1-8B [8]	×	**78.43**	**96.98**	**91.98**
	✓	76.06	96.48	91.11
Llama3.2-1B [1]	×	76.46	96.41	91.20
	✓	76.65	96.75	91.49

Table 13. Performance comparison with *step-by-step instruction* across different LLMs. Default config .

Different style of instruction We take another structure of instruction question-answering in a step-by-step manner, so-called *step-by-step instruction*(Figure 6). Specifically, Rather than providing instruction as Figure 2, we question what the object is and then inquire the coordinates of keypoints. We expect this approach would help the model better understand the input. Interestingly, the effect of this mechanism varies depending on the LLM, as in Table 13. It appears that different LLMs require different approaches to better understand the instruction.

B.2. Architecture

Choice of visual encoder We conduct an ablation study for the visual encoder in CapeLLM. We choose three pop-

Visual Encoder	PCK0.05	PCK0.10	PCK0.15	PCK0.20	PCK0.25	mPCK
DINO-v2-reg [7]	62.52	86.00	92.83	95.83	97.34	86.90
Hiera [27]	56.13	83.31	91.99	95.67	97.35	84.89
DINO-v2 [21]	**78.43**	**91.34**	**95.26**	**96.98**	**97.90**	**91.98**

Table 14. Ablation in visual encoders. Default config .

Fine-tuning method	PCK0.05	PCK0.10	PCK0.15	PCK0.20	PCK0.25	mPCK
None (Frozen)	69.69	88.16	92.62	95.07	96.41	88.39
LoRA [10]	**78.43**	**91.34**	**95.26**	**96.98**	**97.90**	**91.98**
Full parameters	6.93	23.72	42.41	55.31	64.56	38.59

Table 15. Ablation in fine-tuning methods. Default config .

ular visual encoders: DINO-v2 [21], Hiera [27], DINO-v2-reg [7], which are pre-trained on same dataset. Table 14 shows that using DINO-v2 [21] yields the highest performance. The known issue in DINO-v2, artifacts in the feature maps [7]), seems to have little impact on performance in the CAPE task. A noteworthy point is the number of image tokens. Although Hiera [27] has 20% less image tokens than the other two encoders, the performance gap is just about 1%p, implying that retaining a larger number of image tokens does not necessarily have something to do with performance. Then, we examine three types of fine-tuning methods: full fine-tuning, fine-tuning with LoRA [10], and freezing. In constrat with the traditional MLLMs [17, 32, 43], visual encoder with LoRA was more advantageous than the other two options as [31](Table 15). Notably, the full fine-tuning approach, where all parameters are learnable, drastically deteriorate the performance. This fact seems to imply that when using relatively small datasets, leaving all parameters trainable may lead to overfitting, thus resulting in severe degradation in performance.

LLM	PCK0.05	PCK0.10	PCK0.15	PCK0.20	PCK0.25	mPCK
Llama3.2-1B [1]	76.46	91.05	94.69	96.41	97.40	91.20
Vicuna-7B-v1.5 [42]	62.15	84.40	91.51	94.79	96.33	85.84
Mistral-7B-v0.3 [11]	77.63	91.32	94.90	96.46	97.54	91.57
Llama3.1-8B [8]	**78.43**	**91.34**	**95.26**	**96.98**	**97.90**	**91.98**

Table 16. Ablation in LLM. Default config .

Choice of LLM To analyze the performance variations coming from different LLMs, we select four most recent and popular language models: Vicuna-7B [42], Mistral-7B [11], Llama3.1-8B [8], and Llama3.2-1B [1]. We find that the overall accuracy gets improved as the size of the LLM increases(Table 16). Exceptionally, Llama3.2-1B [1] exhibits an overwhelming result surpassing that of a 7B-sized LLM, Vicuna-7B-v1.5, which appears to be the effect of effectively transferring the knowledge of a larger model through distillation training methods [1]. A larger vocabulary size seems to play a essential role to positively influence the integration of visual information and language.

Instruction	Output format	PCK@0.05	PCK@0.2	mPCK
Base instruction	text	**78.43**	**96.98**	**91.98**
	special token	76.06	96.48	91.11
Step-by-step instruction	text	76.46	96.41	91.20
	special token	76.65	96.75	91.49

Table 17. Comparison with token output format. Default config .

Pre-training method	PCK@0.05	PCK@0.2	mPCK
w/o pre-training	**78.43**	**96.98**	**91.98**
Direct QA	78.98	96.60	91.96
Step-by-step QA	78.05	96.23	91.40

Table 18. Comparison in pre-training methods. Default config .

Token output format We explore a method that utilizes token embeddings <KEYPOINT> instead of text-based outputs. To introduce this method to our pipeline, some modifications in instruction should be made: the coordinates are replaced with special token <KEYPOINT> as answers, accordingly the vocabulary size increases, and input embeddings are turned into the trainables. The tokens are turned into the output embeddings from the LLM and are fed into a task-specific decoder. Typically, while a grounding-based pre-trained decoder is used in some tasks [14, 33, 36], no suitable decoders exist for CAPE. So, we create a simple decoder that transforms the embeddings into the coordinates and train it from scratch. We validate this method on both default instruction(as Figure 2) and step-by-step one(Figure 6). Despite the lack of pre-training, the method using <KEYPOINT> outputs comparable result to models with default architecture(Table 17).

C. Pre-Training Strategy

We attempt two types of pre-training process: *direct QA* and *step-by-step QA*. The *direct QA* has an instruction that it is in the form of asking and answering the name of the keypoint corresponding to the coordinates, as in Figure 7. On the other hand, step-by-step QA in Figure 8 has an instruction that is in the form of asking about the category, inquiring the existence of the keypoint in the image, and then inducing the selection of the keypoint corresponding to the coordinates. Referring to the related works [22, 32, 36], all layers except for projection layer are frozen in this stage. As a consequence, there is no positive effect on the performance gain, as shown in Table 18. In light of the use of large-scale pre-training data in the previous methods [22, 31, 32, 36], we conjecture that the limited number of images in each category might result in this outcome.

Keypoint	**Description**
Left eye	The left eye is one of the two visual organs located on the face. It is positioned slightly to the left of the nose and just below the brow ridge, visible from the front.
Right eye	The right eye is the visual organ located on the right side of the face. It is situated to the right of the nose and directly opposite the left eye.
Nose	The nose is the central, protruding feature on the face, located just above the upper lip. It is positioned between and slightly below the eyes
Neck	The neck is the part of the body connecting the head to the torso that refers to the area from the shoulders to the hip joints. It is located below the head, near the junction where the shoulders meet the body.
Root of tail	The root of the tail is at the base of the spine, where the tail begins. It is located near the lower back, above the hips.
Left shoulder	The left shoulder is the joint connecting the left arm to the torso. It is situated to the left of the neck and above the left elbow.
Left elbow	The left elbow is the joint in the middle of the left arm, connecting the upper arm to the forearm. It is located between the left shoulder and the left front paw and connectd with them.
Left front paw	The left front paw is the lower end of the left forelimb, used for movement and manipulation of objects. It is positioned below the left elbow and connected with the left elbow.
Right shoulder	The right shoulder is the joint connecting the right arm to the torso. It is located to the right of the neck and above the right elbow.
Right elbow	The right elbow is the joint in the middle of the right arm, connecting the upper arm to the forearm. It is situated between the right shoulder and the right front paw and connectd with them.
Right front paw	The right front paw is the lower end of the right forelimb, used for movement and manipulation of objects. It is located below the right elbow and connectd with the right elbow.
Left hip	The left hip is the joint connecting the left leg to the torso. It is positioned below the root of the tail and above the left knee.
Left knee	The left knee is the joint in the middle of the left leg, connecting the upper leg to the lower leg. It is located between the left hip and the left back paw and connectd with them..
Left back paw	The left back paw is the lower end of the left hind limb, used for movement and support. It is situated below the left knee.
Right hip	The right hip is the joint connecting the right leg to the torso. It is positioned below the root of the tail and above the right knee.
Right knee	The right knee is the joint in the middle of the right leg, connecting the upper leg to the lower leg. It is located between the right hip and the right back paw and connectd with them.
Right back paw	The right back paw is the lower end of the right hind limb, used for movement and support. It is situated below the right knee.

Table 19. An example of descriptions: *animal body*

USER: What do you think is the central object in this image?

ASSISTANT: The object that this image is trying to express seems to be a {category name}.

USER: Can you identify location of {nose} on the object? The {nose} is the central, protruding feature on the face, located just above the upper lip. It is positioned between and slightly below the eyes. With this description, please provide its coordinates.

ASSISTANT: {coordinates}

Figure 6. Step-by-step instruction. The nose is in the example above, which can be replaced with whatever you want to find out. The underlined is the description of nose, which can also be replaced according to the keypoint.

USER: What is the name of a keypoint at [0.123,0.456] in the image? Tell me the name and why.

ASSISTANT: The keypoint name is nose, because the nose is the central, protruding feature on the face, located just above the upper lip. It is positioned between and slightly below the eyes.

Figure 7. Instruction of direct QA for pre-training.

Joint list: nose, left eye, right eye, ...

USER: What do you think is the central object in this image?

ASSISTANT: The object that this image is trying to express seems to be a {human body}.

USER: What joints can you identify on the {human body} in the image? Answer according to the order in the joint list.

ASSISTANT: The {human body} in the image seems to have several joints in the list, named {nose, left eye, right eye}. Those joints are visible in the image.

USER: Can you identify which joint is located at the coordinate {location}? This coordinate represent relative position within the image, where the top-left corner is defined as [0, 0] and the bottom-right corner as [1, 1]. For example, [0.5, 0.5] indicates the center of the image

ASSISTANT: the joint located at coordinates is {left eye}.

Figure 8. Instruction of step-by-step QA for pre-training.