This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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

Junho Kim[1], Hyungjin Chung[1,*], Byung-Hoon Kim[1,2,*]

[1]EverEx [2]Yonsei University

{jh.kim, hj.chung, bh.kim}@everex.co.kr

## Abstract

*Category-agnostic pose estimation (CAPE) has traditionally relied on support images with annotated keypoints, a process that is often cumbersome and may fail to fully capture the necessary correspondences across diverse object categories. Recent efforts have explored the use of text queries, leveraging their enhanced stability and generalization capabilities. However, existing approaches often remain constrained by their reliance on support queries, their failure to fully utilize the rich priors embedded in pre-trained large language models, and the limitations imposed by their parametric distribution assumptions. To address these challenges, we introduce CapeLLM, the first multimodal large language model (MLLM) designed for CAPE. Our method only employs query image and detailed text descriptions as an input to estimate category-agnostic keypoints. Our method encompasses effective training strategies and carefully designed instructions for applying the MLLM to CAPE. Moreover, we propose an inference mechanism that further enhances the reasoning process for unseen keypoints. while flexibly modeling their underlying spatial distribution and uncertainty, allowing for adaptive refinement based on contextual cues. We conducted extensive experiments to apply the MLLM to CAPE effectively, focusing not only on the model architecture and prompt design but also on ensuring robustness across input variations. Our approach sets a new state-of-the-art on the MP-100 benchmark in the 1-shot and even 5-shot setting, marking a significant advancement in the field of category-agnostic pose estimation. Code is available here.*

## 1. Introduction

Traditional pose estimation typically focuses on a single category, such as human [30, 37, 40], vehicle [23, 24], or animal [13, 41]. However, category-specific pose estimation lacks generalizability and cannot be applied to objects beyond the training set.

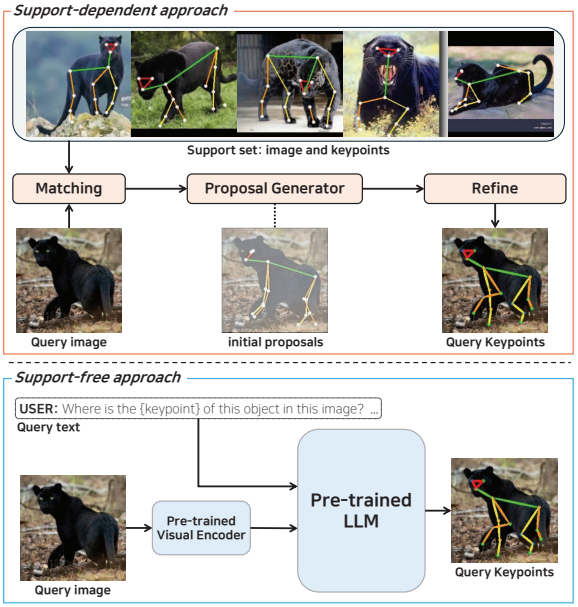



Figure 1. Architectural Comparison. Prior methods (top) are support-dependent approaches requiring support images and keypoint annotations, but ours (bottom) is support-free approach taking only text description of the keypoints in stead of the supports.

Category-Agnostic Pose Estimation (CAPE) methods address this issue [3, 9, 20, 25, 26, 28, 39]. These methods predict keypoint positions of novel objects by employing the existing input image (called "query image") with a set of supports. Typically, the support data comprises a pair: an image that belongs to the same category as the query but has a different pose from it, coupled with its corresponding keypoint annotations.

Most of works in CAPE adopt a two-stage architecture [28] that incorporates maximizing the similarity between query and support features and a process to refine the

*Corresponding authors

similarity to enhance performance consistency. Recently, the types of support data have been diversified. Skeletal structures, representing the connections between keypoints, are used as additional clues to figure out the keypoint location, achieving not only higher accuracy but also robustness against occlusions [9]. Another type of support is to replace traditional support information with a textual one, such as a sequence of keypoint names [26], obtaining some freedom from the inherent reliance on the supports and therefore outperforming the conventional designs [9, 28, 39].

However, employing the support information comes with inherent drawbacks. Since this method aligns support and query images that differ in many aspects aside from belonging to the same category, inadequate generalization during training can cause the performance of the model to vary depending on the quality of the support data, even with the same query image. In addition to the inconsistency, it is often impractical to update the annotations whenever a new set of categories emerges. To mitigate this issue, CapeX [26] incorporated a text-based approach, where keypoint names are used instead of the support images. While this design choice enhanced the flexibility, CapeX is still heavily reliant on the skeletal representations. This poses a conundrum: Can text be the sole source of conditioning for CAPE, completely eliminating the need for additional auxiliary queries?

In this work, we answer this question with a positive by overcoming these challenges by introducing a Multimodal Large Language Model (MLLM) to CAPE. Our contributions can be summarized as follows:

- We introduce CapeLLM, the first support-free framework in CAPE with advanced query-text comprehension capabilities, leveraging an MLLM.
- We elucidate the design choices of using MLLMs for CAPE, from the design of the user query to specific training strategies. Interestingly, we reveal that tailored instructions are the key to unleashing the capabilities of MLLMs in CAPE.
- We propose dynamic round training, enabling spatial reasoning across multiple target poses.
- We propose a flexible decoding strategy that can implicitly model a general probability distribution over keypoints, rather than a fixed parameteric model such as a Gaussian
- We achieve state-of-the-art results on the MP-100 benchmark for CAPE, even outperforming the 5-shot accuracy of the previous art [9].

## 2. Related Work

### 2.1. Category-Agnostic Pose Estimation

Approaches tackling CAPE can be broadly divided into two strategies: structural modifications [3, 20, 25, 28, 39] and changes in the information provided [9, 26]. The simplest structure was first proposed, which predicts keypoint positions by matching support data with the query image [39]. This method connects different pieces of information using cross-attention and outputs similarity scores. While intuitive, it suffers from significant overfitting to the training data. To address this issue, an additional process was introduced to refine the matching results. This enhancement resolves the problems of the previous one-stage architecture and achieves higher performance by reducing dependence on the initial matching outcomes [28].

Keeping the structure similar, performance can also be improved by increasing the amount of provided information. To consider interconnectivity among keypoints rather than treating them as independent entities, skeleton data is incorporated during training [9]. By simply adding connection information, this approach not only achieved better quantitative performance than before but also demonstrated stable results in scenarios like occlusions. In addition to these methods, various approaches have been proposed, such as extracting and utilizing meaningful information from the provided support data [25], or defining meta-points [3, 20] to link keypoints with similar semantics across multiple categories. However, these methods have not solved the fundamental problem that the models heavily rely on the auxiliary information, inducing discrepancy in prediction when changing the information, even with the same query image.

Recently, CapeX [26] proposed to use text information for replacing the support query. Instead of using support images and their corresponding annotations, keypoint names for each category are defined as input and applied to the model. Trained by minimizing image-text similarity using a text encoder [15], CapeX achieves competitive performance. Although leveraging texts in CAPE was shown to be promising, CapeX still relies on the skeletal information proposed in a prior work [9]. Without it, the performance of CapeX degrades heavily. Furthermore, due to the limited capability of the text encoder, only simple words were used as cues, without leveraging the rich semantic information embedded in language. On the other hand, as shown in Table 1, CapeLLM effectively resolves both these issues by leveraging MLLMs, enabling the used of detailed text descriptions, while being completely free from other support queries.

### 2.2. MLLMs for Visual Localization

MLLMs have been widely incorporated into vision tasks, even in dense prediction tasks such as object detection and segmentation [5, 14, 32, 33, 36], which were considered to be challenging for large language models (LLMs) to solve. Pix2Seq [5] introduces a novel perspective by transforming conventional object recognition problems into language

| Method | Category-Agnostic | Support-free | LLM-based |
|---|---|---|---|
| GraphCAPE [9] | ✓ | × | × |
| CapeX [26] | ✓ | ▲ | × |
| CapeLLM (**Ours**) | ✓ | ✓ | ✓ |

Table 1. Difference between prior methods and ours.

modeling tasks. By eliminating overly task-specific modules, it streamlines the model architecture and enables language modeling by quantizing bounding boxes into input sequences. VisionLLM [32] leverages LLMs to expand predefined vision-centric assignments into open-ended ones. [14, 33] use existing models as decoders instead of decoding text outputs. They make the outputs of LLMs be suitable for specific applications by employing foundation models in each vision domain [12, 18]. A work that is closely related to ours is LocLLM [31], which employs as input an image and a text instruction composed of the keypoint name and description. However, LocLLM is limited in that it is category-specific. As will be shown in later chapters, directly incorporating this scheme into CAPE offers several drawbacks[1]. To mitigate this issue, in the following, we investigate the optimal design choices for unleashing the capabilities of MLLM in CAPE.

## 3. CapeLLM

This section is structured as follows. In Sec. 3.1, we describe the model architecture for estimating novel keypoints from given textual descriptions. Then, we present how to design the keypoint names and their corresponding descriptions in Sec. 3.2. Finally, in Sec. 3.3, we introduce two distinct training strategies to endow the spatial reasoning capabilities to CapeLLM.

### 3.1. CapeLLM Model Architecture

Similar to prior works [17, 31], we utilize a pre-trained visual encoder in conjunction with a language model. To extract features across various categories, we adopt DINOv2 [21], which has been pre-trained on large-scale images with a variety of classes, as our visual encoder. For the language model, we select LLaMA3.1 [8], which has a powerful capability in a wide range of language tasks. The overall architecture is schematically illustrated in Figure 2.

Our goal is to estimate keypoint coordinates of unseen categories only grounded on a query image and a text sequence that contains detailed information about the keypoint. To achieve this, the input image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ is first divided into small patches, where $H$ and $W$ denote the height and width of the image, respectively. The patch images are processed through the visual encoder $f_\phi$ to obtain the patch-processed image features $\tilde{\mathbf{V}} \in \mathbb{R}^{N_v\times C}$, where

[1] See, e.g.Tab. 4.

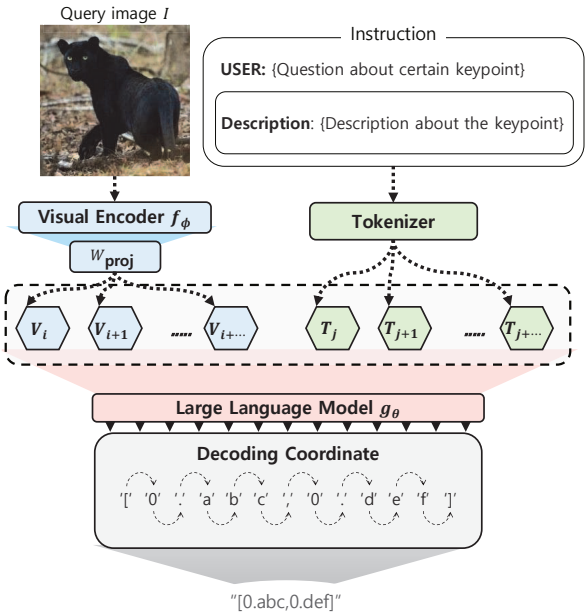



Figure 2. CapeLLM Architecture. CapeLLM consists of two pre-trained modules, visual encoder $f_\phi$ and LLM $g_\theta$. The visual tokens $V$ from visual encoder $f_\phi$ are fed into the LLM $g_\theta$ with the text tokens $T$. The decoding strategy predicts the keypoint coordinate directly as text generation.

$N_v$ is the number of patches and $C$ is the dimension of each patch. These patches are linearly transformed into image tokens $\mathbf{V} \in \mathbb{R}^{N_v\times D}$, via a simple learnable matrix $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{C\times D}$ in Eq (2):

$$\tilde{\mathbf{V}} = f_\phi(\mathbf{x}) \tag{1}$$

$$\mathbf{V} = \tilde{\mathbf{V}}\mathbf{W}_{\text{proj}}, \tag{2}$$

where $D$ represents the dimension of each image token.

These image tokens are prepended to the the query text token embeddings $\mathbf{T} \in \mathbb{R}^{N_t\times D}$, where $N_t$ denotes the number of text tokens, and then fed into the language model as the final input tokens $\mathbf{X} \in \mathbb{R}^{N\times D}$:

$$\mathbf{X} = [\mathbf{V}; \mathbf{T}], \tag{3}$$

where ; denotes concatenation of the two matrices along the token dimension, and $N = N_v + N_t$.

The decoder-only LLM $g_\theta$ processes the input $\mathbf{X}$ to produce the output token matrix $\mathbf{Z}$ which has the same shape as $\mathbf{X}$:

$$\mathbf{Z} = g_\theta(\mathbf{X}). \tag{4}$$

Finally, following previous practices as in [17, 31], a linear transformation with learnable parameters $\mathbf{W}_{\text{logit}} \in$

$\mathbb{R}^{D\times M}$ outputs the final logits $\mathbf{Y} \in \mathbb{R}^{N\times M}$ for each token, where $M$ is the size of the vocabulary:

$$\mathbf{Y} = \mathbf{Z}\mathbf{W}_{\text{logit}}. \tag{5}$$

While there are different strategies to decode the final predicted output of the keypoint, we choose a strategy that endows maximal flexibility in the output distribution. Specifically, we use a decodable floating-point representation by estimating with the following template: `[0.abc, 0.def]`, as shown in Fig. 2. Here, each decimal point is represented by a separate token.

Concretely, let $y \in \mathbb{R}$ be either of the two scalar values that should be predicted. Then, CapeLLM approximates this value by factorizing it over $K = 3$ digit tokens, i.e.

$$\begin{aligned} p(y|\mathbf{x}) &\approx p_{\phi,\theta}(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K|\mathbf{x}) \\ &= \prod_{k=1}^{K} p_{\phi,\theta}(\mathbf{Y}_k|\mathbf{x}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{k-1}). \end{aligned} \tag{6}$$

Then, one can show that (See Theorem 1 of [29]) CapeLLM models a truncated conditional density up to a resolution of $10^{-K}$. This leads to vastly enhanced flexibility, as opposed to the widely used Gaussian parametrization. See Sec. 4.4 for detailed empirical results.

### 3.2. Design of Instructions

Instructions exert a significant influence on MLLM's performance [6, 17, 35, 43]. In CAPE, the categories and keypoints in the training set differ from those in the test set. Accordingly, most of the previous approaches [3, 9, 20, 25, 28, 39] have utilized support sets and images with annotations, and recent work [26] relies on keypoint names. We assume that using only keypoint names to infer their positions within images proves insufficient. To predict unseen keypoints, we opt for detailed descriptions of the keypoints and integrate this information into the instructions. Since some categories have densely defined keypoints, especially human faces, we manually design these descriptions to clearly delineate the differences between them. In crafting the descriptions, we avoid vague or ambiguous phrasing intentionally, instead providing precise details about the spatial positions and relationships among keypoints. For instance, when describing the "front wheel" of a swivel chair, rather than saying "starting from this wheel, the remaining wheels are located clockwise," we expressed it as "next to the fifth wheel and in front of the second wheel.". We provide further details in Table 19.

In the instructions, the answer is the keypoint coordinates, represented as normalized floating-point values. Although there are various coordinate representation methods, e.g., integer-valued binning [5], or deviations from anchors [22], these approaches can be cumbersome due to the need for extra tokens or pre-defined anchors, while hampering flexibility. In contrast, we use the floating-point coordinate representation [4, 31] as our solution, which not only is simple and robust without the need to incorporate additional tokens but also leads to implicit modeling of arbitrary probability distributions of keypoints. See Sec. 4.4 for details.

### 3.3. Training Strategy

**Fixed round training** To adapt the MLLM to the CAPE task, we introduce Fixed and Dynamic-Round Training strategies. Although the CAPE benchmark dataset, MP-100, covers a wide array of categories, each category contains only around 200 samples on average, which is considerably lower compared to other benchmarks (MSCOCO [16], MPII [2], AP-10K [41]). Therefore, the approach of matching an individual image with just a subset of keypoints, as used in LocLLM [31], proves to be insufficient. Instead, we form (image, keypoints) pairs where every image is paired with all of its keypoints during training ("*Fixed-Round*" strategy). Initially, we partition the keypoints ($K_{\text{category}}$) into groups of $k$. Each set of keypoints is then combined with an image. It is important to note that we permit the repetition of images until every keypoint has been included in a pair. For example, in the bird category (where $K_{\text{bird}}$=15), if we group the keypoints in sets of 5 (i.e., $k$=5), a single bird image will produce 3 pairs. This approach guarantees that no keypoints are left unpaired during the training phase. Based on these individual pairs, we construct a multi-round conversation framework for model training.

**Dynamic round training** In addition to the Fixed-Round conversation method above, we also explore a "*Dynamic-Round*" strategy. Although both methods involve pairing an image with all the keypoints, the Dynamic-Round approach differs in that the number of keypoints linked to an image varies for each pair, unlike the fixed count of $k$ in the Fixed-Round method. This variability is intended to reinforce the reasoning process by utilizing information from other keypoints during prediction. The performance variations stemming from these two strategies are discussed in Section 4.3.

## 4. Experiment

### 4.1. Environment

**Benchmark** We utilize the MP-100 benchmark [39], aligning with prior methods [3, 9, 20, 25, 26, 28, 39]. The benchmark comprises 100 categories and approximately 20,000 images. These categories are split into train, validation, and test sets in a 70/10/20 ratio without any category overlap. To ensure and identify robustness on unseen categories, the dataset is organized into five different splits, arranging each category to appear once in the test

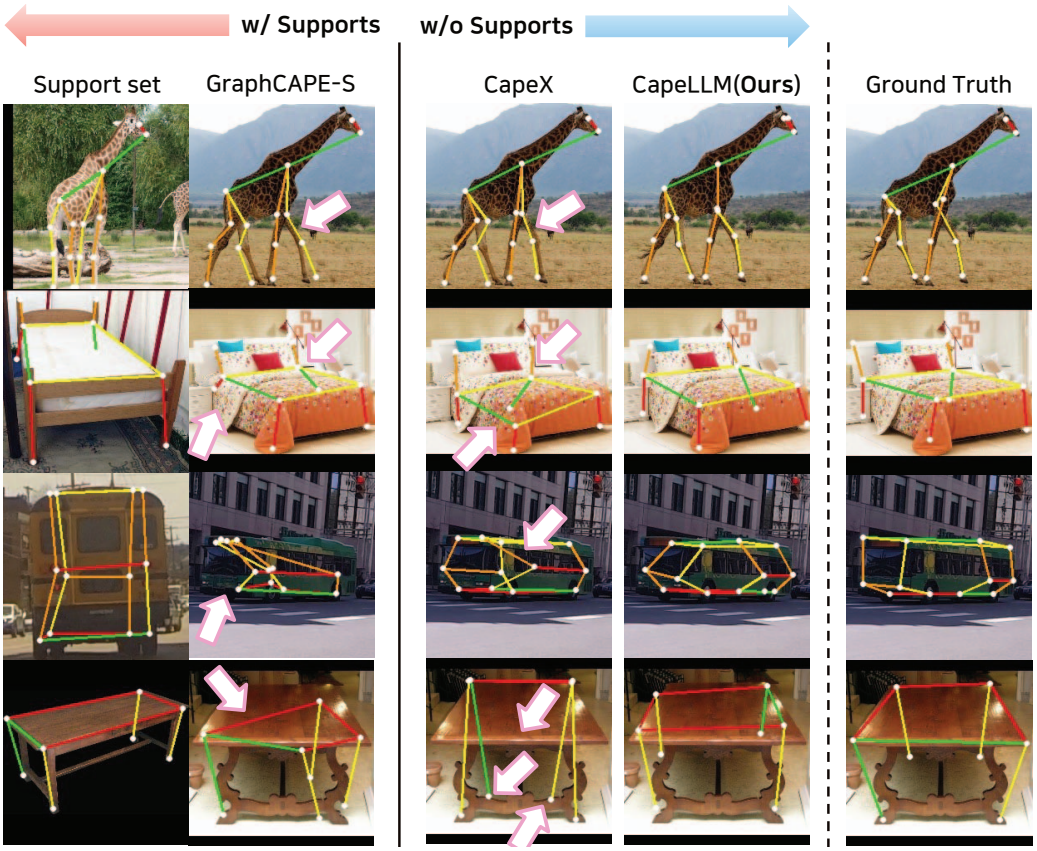



Figure 3. Qualitative results on MP-100. The support set is only used for GraphCAPE [9]

| Evaluation dataset | Model | split1 | split2 | split3 | split4 | split5 | Avg |
|---|---|---|---|---|---|---|---|
| Support-Query Pairs | GraphCAPE-S(1-shot) [9] | 94.63 | 89.79 | 90.30 | 87.81 | 90.07 | 90.52 |
| | GraphCAPE-S(5-shot) [9] | 95.81 | 90.78 | **90.94** | 90.42 | 92.27 | 92.04 |
| | CapeX [26] | 95.29 | 91.08 | 88.94 | 89.83 | 92.96 | 91.62 |
| | CapeLLM (**Ours**) | **97.01** | **92.40** | 90.58 | **90.90** | **92.11** | **92.60** |
| Only Query Images | CapeX [26] | 95.28 | 91.08 | 89.06 | 89.67 | **92.87** | 91.59 |
| | CapeLLM (**Ours**) | **96.98** | **92.34** | **90.57** | **90.87** | 92.24 | **92.60** |

Table 2. PCK@0.2 on the MP-100 dataset. An method using support-set [9] is usually evaluated on support-query pairs("*Support-Query Pairs*"), which means the the same query image can be assigned to different support images to form a pair. On the contrary, "*Only Query Images*" consists of unique queries, since it does not care about support-set.

set. The number of keypoints per category varies significantly, ranging from 8 to 68. We discover that images in some categories (e.g., "hand") do not correctly correspond to keypoints; we rectify these discrepancies before proceeding with experiments. We use PCK@0.2 as a measurement of accuracy and compare both quantitative and qualitative results with two models as representatives: GraphCAPE [9] and CapeX [26]. GraphCAPE [9] holds the highest 1-shot and 5-shot accuracy among models employing a query-support paired architecture(as seen in upper side of Figure 1). CapeX [26], based on the GraphCAPE framework, is a text-based query architecture utilizing keypoint names instead of support images and annotations. By connecting text features extracted via a text encoder[15] to im-

age features, CapeX [26] records higher 1-shot performance than GraphCAPE [9]. Because of their overwhelmingly superior performance compared to other models, both quantitatively and qualitatively, we set these two methods as our baselines.

**Implementation Details** CapeLLM consists of a visual encoder, a projection layer, and an LLM. We choose DINOv2 [21] as our vision encoder. Inspired by [17], the projection layer linking the vision and the text part is implemented as a linear layer. For the base language model, we use the pre-trained decoder-only model, LLaMA3.1 [8]. We employ LoRA [10] to fully leverage the capabilities of large-scale pre-trained models and adjust them to this task. Following the setting of [31], we also apply LoRA [10] to the query/value projection of the attention layers in both the visual encoder and the LLM, projecting them into the dimension of a rank 8. We selected AdamW as the optimizer and set the learning rate to $5 \times 10^{-4}$. The total number of training epochs was set to 12, with a warm-up phase over 3% of the total training steps. Input images were resized to 224×224 pixels, and the default number of rounds included in each instruction was set to 4. We train our model on a Linux server equipped with four RTX-A6000 GPUs; instead of setting the batch size to 1 per GPU, we set the accumulation steps to 32.

### 4.2. Main Results

**Quantitative Results** While GraphCAPE [9] configures the test dataset with support-query pairs, our method is completely support-free. For fair comparison, we make the same pairs as GraphCAPE [9] and measure the performance only using the queries in the pairs("Support-Query Pairs"). As shown in Table 2, our method outperforms the 1-shot accuracy of GraphCAPE [9] over 1%p and further achieves 0.56%p higher accuracy than its 5-shot performance. Note that the information provided for 5-shot CAPE is strictly larger than in our case. Remarkably, this result demonstrates that even without any support, it is possible to attain superior results in CAPE. We also compare our model with CapeX [26] using about 2,000 query images from each split("Only Query Images"). In Table 2. we achieve approximately 1%p higher accuracy than CapeX [26], achieving state-of-the-art.

**Qualitative Results** As can be seen in Figure 3, we demonstrate that our CapeLLM is superior to conventional methods [9, 26] across various categories. Specifically, in *animal body*, remarkable improvements have been found in end joints, such as knees and paws, which the preceding approaches [9, 26] have struggled to predict. In other categories, prior methods [9, 26] show imprecise results. For instance, GraphCAPE [9] fails due to differences of pose in the supports, and CapeX [26] confuses the front and rear joints. Our method, however, not only produces more accurate results than previous approaches [9, 26], but also works as intended based on the given instructions. In Figure 4, we further show several challenging cases where there exists sufficient occlusion in the input image. Remarkably, CapeLLM performs robustly even under these challenging scenarios, without the use of skeletal structure [9], which are often used to explicitly enhance the robustness under occlusion.

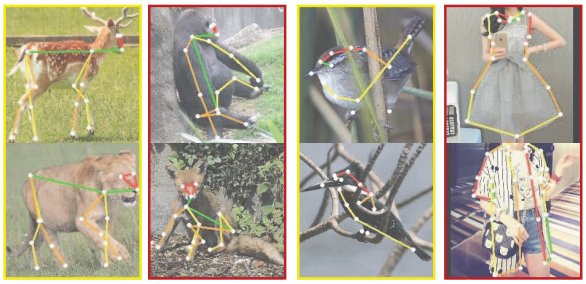



Figure 4. Qualitative results in diverse occlusions. There are diverse cases of occlusion: (a) Self-occlusion, (b) self-occlusion with articulated pose, (c) occlusion-by-object, and (d) self-and-object occlusion. CapeLLM performs robustly even in such challenging cases with occlusion.

| Method | Training | Inference | | Metric | | |
|---|---|---|---|---|---|---|
| | | Single | Cumulative | PCK0.15 | PCK0.20 | PCK0.25 |
| LocLLM-style [31] | - | ✓ | × | 91.00 | 94.85 | 96.98 |
| | | × | ✓ | 90.56 | 93.39 | 95.03 |
| Ours | Fixed | ✓ | × | 95.26 | 96.98 | 97.90 |
| | | × | ✓ | 93.02 | 94.59 | 95.42 |
| | Dynamic | ✓ | × | 94.31 | 96.05 | 97.26 |
| | | × | ✓ | **95.62** | **97.28** | **98.27** |

Table 3. Result of cumulative reasoning. Default config.

### 4.3. Cumulative Reasoning for Pose Estimation

Inspired by the previous works related to Chain-of-Thought (CoT) [17, 34, 38], we investigate whether the information from certain keypoints influences the estimation of others. Specifically, we devise an inference mechanism, called "*Cumulative Reasoning*", for helping to predict keypoint coordinates more precisely using the implicit capability of MLLMs, and analyze its effectiveness comparing with default process, single-round inference ("Single" in Table 3). For this reasoning task, the prompt employed for predicting each keypoint is prepended to the prompt for the subsequent keypoints, thereby establishing a cumulative context. As detailed in Table 3, while the LocLLM [31]-style and the fixed-training scheme result in decreased performance (see the 2nd and 4th rows), dynamic-round training strategy not only encourages the model to achieve better results than

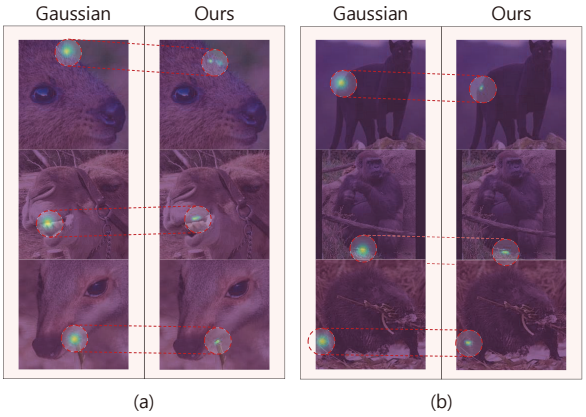



Figure 5. Distribution modeling for keypoints. "Gaussian" displays a fixed-variance Gaussian around the ground-truth target point. "Ours" displays the distribution modeled by CapeLLM, achieved by sampling multiple points, and using kernel density estimation. In (a) **animal face**, (top) *top left side of the left eye*; (mid) *right side of the lip*; (bottom) *left side of the lip*. In (b) **animal body**, (top) *root of tail*; (mid) *right back paw*; (bottom) *left knee*.

single-round inference by 1.2%p (see the last row), but also outperforms the fixed-round strategy in Table 4. This experiment suggests that CapeLLM can extract richer spatial and relational cues based on the accumulated information from other keypoints, which in turn enables it to reason about the target keypoint's position.

### 4.4. Implicit Distribution Modeling

Many previous CAPE methods are based on heatmap-based learning, and most of them rely on parametric distribution modeling, among them predominantly a Gaussian distribution. It is also often the case that a fixed variance is used [19], further constraining the situation. While such a strategy is known to be robust than point estimates, it also greatly limits the capacity of the neural network by constraining the output distribution. This strategy is also often physically implausible, as the points to be estimated are often located near the edge of the support. While convolving the point with a Gaussian would induce some probabilities in all the nearby regions, it should ideally allocate the probability of 0 for out-of-bounds. (See Fig. 5 for several examples of this). On the other hand, CapeLLM generates the output coordinates in an unconstrained fashion, which can approximate the true posterior with arbitrary precision [29]. Using this property, we examine a density estimation for keypoints based on our decoding process, ("decoding-based method") and compare this method with the prevalent Gaussian modeling. As illustrated in Figure 5, leveraging the inherent randomness in the MLLM's decoding strategy seems to make the points located closer to the ground truth point. Also, while the distribution of Gaussian modeling stretches across the background, that of the decoding-based method are kept within the foreground. This observation implies that the decoding-based modeling can be another option to build a density function of keypoint and resolve the inherent constraints in the conventional method, such as fixed-variance Gaussian [19].

| Method | Training | PCK@0.05 | PCK@0.2 | mPCK |
|---|---|---|---|---|
| LocLLM-style [31] | - | 55.15 | 94.85 | 84.00 |
| Ours | Fixed | **78.43** | **96.98** | **91.98** |
| | Dynamic | 76.55 | 96.05 | 90.92 |

Table 4. Result in different training strategies. Default config.

### 4.5. Ablation Studies

As CapeLLM is, to the best of our knowledge, the first work that uses MLLM for CAPE, in this section, we conduct extensive ablation studies to reveal the crucial components that lead to the superior performance of our method. We further show that CapeLLM is robust to variations in the input image resolution and the decoding strategy. Further ablations are deferred to the appendix.

**Training strategy** To investigate the effectiveness of our training schemes, we compare them with the training strategy from LocLLM [31]. The key difference between these strategies lies in the process to construct image–keypoint pairs. Our findings reveal that our approaches outperform the approach in LocLLM across all metrics. Notably, the fixed-round training demonstrates an approximately 8%p improvement in mPCK, indicating that explicitly connecting all keypoints with a single image is more suitable for leveraging MLLMs in CAPE. Here, we determine $k$ value in fixed-round training empirically; the corresponding results are provided in the Appendix.

| Description Type | PCK@0.05 | PCK@0.2 | mPCK |
|---|---|---|---|
| Vague | 69.82 | 91.97 | 85.98 |
| Spatial & Relational | **78.43** | **96.98** | **91.98** |

Table 5. Comparison with vague descriptions. Our method.

| Name | Description | PCK0.05 | PCK0.20 | PCK0.25 |
|---|---|---|---|---|
| × | × | **78.43** | **96.98** | **97.90** |
| ✓ | × | 78.24 | 96.94 | 97.83 |
| × | ✓ | 70.11 | 96.67 | 97.69 |
| ✓ | ✓ | 66.16 | 96.09 | 97.26 |

Table 6. Robustness to variation in input. Default config.

| In Training | In Inference | | | Metric | |
|---|---|---|---|---|---|
| Random Replaced | Detail Desc | Replaced | Removed | PCK0.20 | mPCK |
| × | ✓ | × | × | **96.98** | **91.98** |
| × | × | ✓ | × | 95.28 | 86.66 |
| × | × | × | ✓ | 95.15 | 85.69 |
| ✓ | ✓ | × | × | 96.65 | 89.87 |
| ✓ | × | ✓ | × | 96.39 | 89.65 |
| ✓ | × | × | ✓ | 96.07 | 88.84 |

Table 7. Performance with varying descriptions. Default config.

**Choice of instructions** To investigate the impact of instructions on the performance, we define two types of variations in the instruction: (1) change in the keypoint names and descriptions, (2) the omission of keypoint description. Initially, we consider a scenario in which vague keypoint descriptions are provided, as mentioned in Sec. 3.2, and convert descriptions with the ambiguous ones in training. We find that incorporating detailed spatial and relational information among keypoints yields an improvement of up to 6%p in mPCK compared to the baseline (Table 5). This finding highlights the critical role of learning spatial positioning and contextual relationships for accurately predicting keypoint coordinates. We also assess the model's stability when confronted with input styles that differ from those seen during training. By utilizing GPT-4o, we prompt it to convert the keypoint names and descriptions in the test set into a simplified format that preserves their original meaning. According to Table 6, CapeLLM exhibits notably consistent performance across these variations. Further, we scrutinize two cases where the descriptions are excluded: (1) employing a simple QA method to query keypoint coordinates without any description, (2) where the description is replaced with the statement "There is no description to refer to.". In all cases, CapeLLM maintains relatively stable performance, and its robustness can be further enhanced when taking advantage of the random substitution of descriptions during training (Table 7).

| Resolution | PCK0.05 | PCK0.20 | PCK0.25 |
|---|---|---|---|
| 224×224 | 78.43 | 96.98 | 97.90 |
| 238×238 | **78.68** | 97.07 | 97.93 |
| 252×252 | 78.45 | **97.11** | **97.99** |
| 336×336 | 78.25 | 96.85 | 97.81 |

Table 8. Robustness to larger image resolution. Default config.

**Larger image resolution** We investigate how the performance varies when varying the input image size to be larger than the default 224×224 resolution. As presented in Table 8, across many different input image sizes up to 336×336, CapeLLM retains its robust performance.

**Decoding strategy** As our decoding strategy is directly based on LLM token decoding, we can leverage many different standard LLM sampling strategies to sample from the posterior $p(y|\mathbf{x})$ in (6). While we resort to greedy decoding for all of our main experiments for simplicity, we explore different strategies in Tab. 9. Here, we observe that CapeLLM is again robust to different sampling strategies, achieving state-of-the-art performance in *all* cases. We further find that with Top-$k$ sampling, we are able to achieve even better performance. This suggests that exploring more advanced decoding strategies could further enhance CapeLLM, which we leave as an avenue for future work.

| Decoding Strategy | PCK@0.2 | PCK@0.25 | mPCK |
|---|---|---|---|
| Greedy Search | 0.9698 | 0.9790 | 0.9198 |
| Sampling | 0.9606 | 0.9726 | 0.8958 |
| Sampling (`t=0.6`) | 0.9669 | 0.9776 | 0.9095 |
| Top-3 Sampling | **0.9707** | **0.9792** | **0.9200** |
| Top-5 Sampling | 0.9705 | 0.9791 | 0.9197 |
| Nucleus Sampling (`p=0.92`) | 0.9669 | 0.9777 | 0.9071 |
| Contrastive Search (`a=0.6`) | 0.9698 | 0.9790 | 0.9198 |

Table 9. Performance comparison of different decoding strategies. Default config.

## 5. Conclusion

We introduce CapeLLM, the first fully support-free MLLM-based method for CAPE. By leveraging the reasoning capabilities of a pre-trained LLM, CapeLLM achieves state-of-the-art performance without requiring any support images or annotations, fundamentally challenging the paradigm of prior CAPE approaches. Our method integrates MLLMs with a lightweight projection layer, allowing seamless alignment between visual and textual modalities. To fully harness MLLMs for CAPE, we design structured keypoint instructions, providing explicit spatial and relational descriptions that enable robust keypoint estimation across unseen categories. Furthermore, we introduce two novel training strategies—fixed-round and dynamic-round training—which not only improve category-agnostic keypoint prediction but also enhance cumulative reasoning, allowing the model to refine its predictions iteratively based on contextual information. We believe CapeLLM serves as a foundational step toward the broader application of MLLMs in spatial reasoning and structured perception tasks. Future work can explore more advanced decoding strategies, multi-modal extensions, and scalability to real-world scenarios, paving the way for next-generation keypoint estimation models driven by MLLMs.

## References

[1] Meta AI. Llama 3.2: Vision Edge for Mobile Devices, 2024. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. 11, 12

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[3] Junjie Chen, Jiebin Yan, Yuming Fang, and Li Niu. Meta-Point Learning and Refining for Category-Agnostic Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23534–23543, 2024. 1, 2, 4

[4] 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

[5] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A Language Modeling Framework for Object Detection. In *International Conference on Learning Representations*, 2022. 2, 4

[6] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 4

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *The Twelfth International Conference on Learning Representations*, 2024. 12

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3, 6, 11, 12

[9] Or Hirschorn and Shai Avidan. A graph-based approach for category-agnostic pose estimation, 2024. 1, 2, 3, 4, 5, 6

[10] 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. In *International Conference on Learning Representations*, 2022. 6, 12

[11] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. 12

[12] 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

[13] Rollyn Labuguen, Jumpei Matsumoto, Salvador Blanco Negrete, Hiroshi Nishimaru, Hisao Nishijo, Masahiko Takada, Yasuhiro Go, Ken-ichi Inoue, and Tomohiro Shibata. MacaquePose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. *Frontiers in behavioral neuroscience*, 14:581154, 2021. 1

[14] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 2, 3, 12

[15] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. 2, 5

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2015. 4

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3, 4, 6, 11, 12

[18] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3

[19] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13264–13273, 2021. 7

[20] Khoi Duc Nguyen, Chen Li, and Gim Hee Lee. ESCAPE: Encoding Super-keypoints for Category-Agnostic Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23491–23500, 2024. 1, 2, 4

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. 3, 6, 12

[22] Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12977–12987, 2024. 4, 12

[23] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1906–1915, 2018. 1

[24] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7326–7335, 2019. 1

[25] Pengfei Ren, Yuanyuan Gao, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Dynamic Support Information Mining for Category-Agnostic Pose Estimation. In *Proceedings*

*of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2024. 1, 2, 4

[26] Matan Rusanovsky, Or Hirschorn, and Shai Avidan. CapeX: Category-Agnostic Pose Estimation from Textual Point Explanation. *arXiv preprint arXiv:2406.00384*, 2024. 1, 2, 3, 4, 5, 6

[27] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 12

[28] Min Shi, Zihao Huang, Xianzheng Ma, Xiaowei Hu, and Zhiguo Cao. Matching is not enough: A two-stage framework for category-agnostic pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7317, 2023. 1, 2, 4

[29] Xingyou Song and Dara Bahri. Decoding-based regression. *arXiv preprint arXiv:2501.19383*, 2025. 4, 7

[30] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1

[31] Dongkai Wang, Shiyu Xuan, and Shiliang Zhang. LocLLM: Exploiting Generalizable Human Keypoint Localization via Large Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 614–623, 2024. 3, 4, 6, 7, 12

[32] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 12

[33] Fei Wei, Xinyu Zhang, Ailing Zhang, Bo Zhang, and Xiangxiang Chu. Lenna: Language enhanced reasoning detection assistant. *arXiv preprint arXiv:2312.02433*, 2023. 2, 3, 12

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 6

[35] Junda Wu, Xintong Li, Tong Yu, Yu Wang, Xiang Chen, Jiuxiang Gu, Lina Yao, Jingbo Shang, and Julian McAuley. Commit: Coordinated instruction tuning for multimodal large language models. *arXiv preprint arXiv:2407.20454*, 2024. 4

[36] Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Ping Luo, Yu Qiao, and Jifeng Dai. VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 12

[37] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 1

[38] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. LLaVA-CoT: Let Vision Language Models Reason Step-by-Step, 2025. 6

[39] Lumin Xu, Sheng Jin, Wang Zeng, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, and Xiaogang Wang. Pose for everything: Towards category-agnostic pose estimation. In *European conference on computer vision*, pages 398–416. Springer, 2022. 1, 2, 4

[40] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022. 1

[41] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. AP-10K: A Benchmark for Animal Pose Estimation in the Wild. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 1, 4

[42] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023. 12

[43] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2024. 4, 12