KptLLM: Unveiling the Power of Large Language Model for Keypoint Comprehension

Jie Yang[1,2,5] **Wang Zeng**[4,5] **Sheng Jin**[3,5] **Lumin Xu**[4] **Wentao Liu**[5*] **Chen Qian**[5] **Ruimao Zhang**[1*]
[1] Sun Yat-sen University [2]The Chinese University of Hong Kong, Shenzhen
[3]The University of Hong Kong [4]The Chinese University of Hong Kong
[5] SenseTime Research and Tetras.AI

https://kptllm.github.io

Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have greatly improved their abilities in image understanding. However, these models often struggle with grasping pixel-level semantic details, *e.g.*, the keypoints of an object. To bridge this gap, we introduce the novel challenge of *Semantic Keypoint Comprehension*, which aims to comprehend keypoints across different task scenarios, including keypoint semantic understanding, visual prompt-based keypoint detection, and textual prompt-based keypoint detection. Moreover, we introduce KptLLM, a unified multimodal model that utilizes an identify-then-detect strategy to effectively address these challenges. KptLLM underscores the initial discernment of semantics in keypoints, followed by the precise determination of their positions through a chain-of-thought process. With several carefully designed modules, KptLLM adeptly handles various modality inputs, facilitating the interpretation of both semantic contents and keypoint locations. Our extensive experiments demonstrate KptLLM's superiority in various keypoint detection benchmarks and its unique semantic capabilities in interpreting keypoints.

1 Introduction

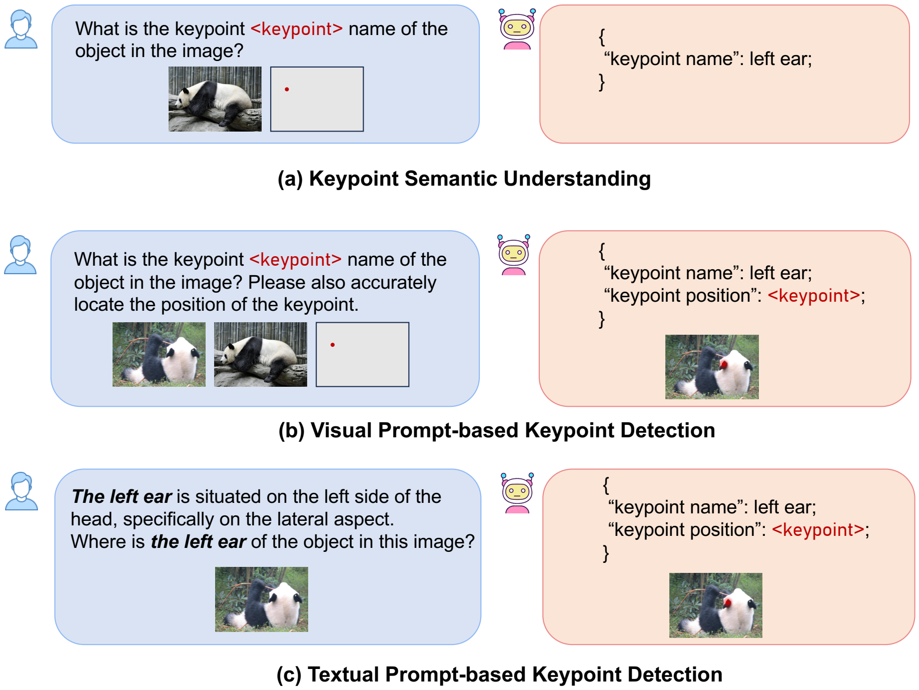
Recent advancements in deep learning and natural language processing have facilitated the rise of Large Language Models (LLMs) that display human-like fluency in text comprehension and generation [1, 2, 3, 4, 5, 6, 7]. By incorporating visual information, researchers have developed Multimodal Large Language Models (MLLMs) [8, 9, 10, 11, 12], specifically designed for visual-language tasks, showcasing remarkable abilities in image understanding. However, these models encounter difficulties in capturing fine-grained semantic details, particularly at the point level, which are crucial for various real-world applications. The exploration of MLLMs for keypoint comprehension remains under-explored in the literature.

Keypoint detection is a fundamental aspect of computer vision that supports various applications such as controllable image/video generation [13, 14, 15], human-centric perception [16, 17], and AR/VR systems [18, 19, 20]. Initially, research in this field focused on closed-set problems, aiming to predict the locations of predefined semantic keypoints of a certain object category (*e.g.* human body). As the demand for generalization grew, researchers started investigating the detection of keypoints for novel objects by providing visual prompts (*i.e.*, a support image of a novel object with its keypoint definitions) [21, 22] or utilizing textual prompts (*i.e.*, keypoint names) [23]. Despite significant progress in these areas, existing models still fall short of achieving genuine semantic comprehension

[*] Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-4467



What is the keypoint <keypoint> name of the object in the image?

{
"keypoint name": left ear;
}

(a) Keypoint Semantic Understanding

What is the keypoint <keypoint> name of the object in the image? Please also accurately locate the position of the keypoint.

{
"keypoint name": left ear;
"keypoint position": <keypoint>;
}

(b) Visual Prompt-based Keypoint Detection

The left ear is situated on the left side of the head, specifically on the lateral aspect.
Where is ***the left ear*** of the object in this image?

{
"keypoint name": left ear;
"keypoint position": <keypoint>;
}

(c) Textual Prompt-based Keypoint Detection

Figure 1: This work aims to address the problem of semantic keypoint comprehension, which aims to understand keypoints across different task scenarios: ***(a) Keypoint Semantic Understanding*** takes the object image and a keypoint prompt (*i.e.*, the position of the target keypoint) as inputs, then generate responses that interpret keypoint semantics; ***(b) Visual Prompt-based Keypoint Detection*** takes a query image and a support image with a keypoint prompt as inputs and then outputs the corresponding keypoint positions and semantics of the query image; ***(c) Textual Prompt-based Keypoint Detection*** utilizes detailed descriptions of keypoints through extensive text, to perform more generalizable keypoint detection.

of keypoints akin to humans. These models primarily rely on direct learning of visual patterns for keypoint localization through extensive data fitting, while neglecting semantic understanding of the keypoints, thus leading to misinterpretation of the prompts and inaccurate predictions. Moreover, the input-output structures are designed in fixed and predefined formats, restricting their usage to predetermined methods and impeding the flexibility required for interfacing with users.

Motivated by the aforementioned challenges, this paper delves into a more comprehensive problem of *Semantic Keypoint Comprehension* to evaluate the model capability of comprehensively understanding keypoints both visually and semantically. As shown in Fig. 1, we investigate three distinct capabilities via different task instructions: ***(a) Keypoint Semantic Understanding*** aims to infer the desired keypoint semantics, given the target image and a keypoint prompt (*i.e.*, the position of the target keypoint) as inputs. It provides the potential for an AI model with high-level visual understanding and analytical capabilities, crucial for tasks such as structural comprehension, action recognition, and medical image analysis. ***(b) Visual Prompt-based Keypoint Detection***, also referred to as category-agnostic pose estimation, takes a query image and a labeled support image with the keypoint annotation as inputs and then outputs the corresponding keypoint positions in the query image. This capability requires the model to acquire keypoint definitions from visual prompts, enabling it to perform cross-class and cross-keypoint localization tasks using sample images provided by users. ***(c) Textual Prompt-based Keypoint Detection***, also known as open-vocabulary keypoint detection, aims to utilize detailed descriptions of keypoints through extensive text for keypoint localization. The keypoint detectors directly receive the human language guidance, facilitating keypoint localization on arbitrary object and keypoint categories in a zero-shot manner.

We introduce KptLLM, a novel framework that utilizes an identify-then-detect strategy to address the challenging problem of semantic keypoint comprehension. It formulates all three capabilities depicted in Fig. 1, by first identifying the semantic meaning of keypoints and then detecting their positions via a chain-of-thought approach, akin to human cognition. KptLLM is a unified framework that comprises four key components designed to accommodate various modality inputs and infer both the semantics and location of the keypoint. Specifically, we first extract visual features of both query and support images to obtain query visual tokens and support image features. Secondly, we encode the support keypoint prompt, which describes the position of keypoint on the support image, to generate keypoint prompt embedding. Thirdly, prompt-oriented features are derived by integrating support image features with keypoint prompt embedding, and are utilized to form keypoint prompt tokens. Lastly, LLMs take query visual tokens, keypoint prompt tokens, and task-related language tokens as input, and then generate the semantic description of the target keypoint and its corresponding position on the query image. By harnessing commonsense knowledge in LLMs, KptLLM can assist in keypoint localization of novel object categories, potentially leading to enhanced generalizability in performance. In addition, the chain-of-thought design elicits the powerful keypoint understanding capabilities of LLMs, which helps to distinguish visually ambiguous keypoints (*e.g*. left and right arms). Extensive experiments demonstrate KptLLM's superiority on semantic keypoint comprehension, showcasing its unique semantic understanding capabilities in interpreting keypoints and state-of-the-art performance in various keypoint detection benchmarks.

In summary, the contributions of this work are three-fold: (1) We pioneer the investigation of a novel problem in semantically interpretable keypoint analysis, termed *Semantic Keypoint Comprehension*, which aims to enhance MLLMs with improved image understanding at a finer-grained keypoint level; (2) We introduce KptLLM, a unified multimodal model that utilizes an identify-then-detect strategy to effectively address three tasks of semantic keypoint comprehension. KptLLM underscores the initial discernment of semantic significance in keypoints, followed by the precise determination of their positions through a chain-of-thought process. (3) We demonstrate KptLLM's superiority in various existing keypoint detection benchmarks and its unique semantic capabilities in interpreting keypoints. We hope our work could inspire future research on keypoint understanding and localization, while also fostering enhanced human-AI interface in fine-grained visual understanding.

2 Related Work

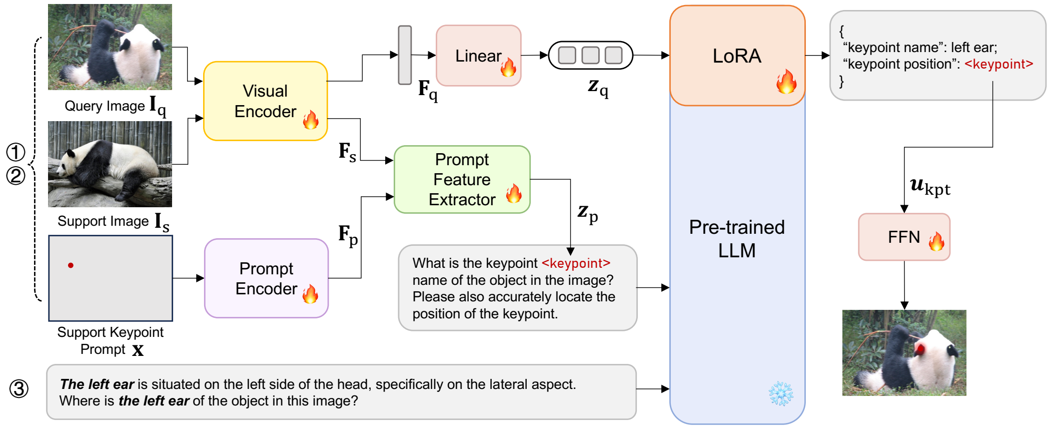
2.1 Keypoint Detection

Keypoint detection, also referred to as pose estimation, focuses on localizing the 2D keypoints of objects in the image. Traditional models for keypoint detection are typically designed for a single category, *e.g.*, human [24, 25, 26, 27, 28, 29, 30], animal [31, 32, 33] and clothes [34]. Based on the localization strategy, existing methods are generally divided into regression-based methods [35, 36, 37, 38, 39, 40] and heatmap-based methods [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51]. More recently, methods that can recognize and localize keypoints for unseen object categories in the training datasets, are gaining increasing attention from the community. Category-agnostic pose estimation [21, 22], also referred to few-shot keypoint detection, aims to estimate the pose of any category in query images with visual prompts (*i.e.*, a few support images of a novel class and its corresponding keypoint annotations). Another line of research explores open-vocabulary keypoint detection [23, 52], which aims to localize keypoints based on text prompts in zero-shot settings. In this work, we investigate semantic keypoint comprehension and propose a novel unified framework to comprehend keypoints across three different task scenarios, including (a) keypoint semantic understanding; (b) visual prompt-based keypoint detection; (c) textual prompt-based keypoint detection.

2.2 Multimodal Large Language Model

Inspired by the success of Large Language Models (LLMs) [1, 2, 53, 3, 4, 54, 5, 6, 7], researchers are exploring ways to transfer the formidable capabilities of LLMs into the realm of vision, developing Multimodal Large Language Models (MLLMs) [55, 8, 9, 10, 56, 57, 11, 12, 58, 59, 60, 61, 62, 19]. These models exemplify an autoregressive mechanism predicated on a transformer decoder architecture [63]. The integration of visual representation from vision encoders [64, 65] into the domain of LLMs ushers a new era of visual comprehension and reasoning. Such integration is predominantly facilitated through a Multilayer Perceptron (MLP) that seamlessly transforms visual features into the input embedding space of LLMs [10, 56, 57], or via a cross-attention mechanism that attends to visual contents through a series of attention layers [55, 8, 11]. However, most of these VLMs can only provide text outputs, inhibiting the complex applications requiring detailed visual perception. VisionLLM [66] tackles a range of conventional vision-centric tasks by instruction

https://doi.org/10.52202/079017-4467


①
②
Query Image $\mathbf{I}_\mathrm{q}$
Support Image $\mathbf{I}_\mathrm{s}$
Support Keypoint Prompt $\mathbf{x}$
Visual Encoder
Prompt Encoder
$\mathbf{F}_\mathrm{q}$
Linear
$\boldsymbol{z}_\mathrm{q}$
$\mathbf{F}_\mathrm{s}$
$\mathbf{F}_\mathrm{p}$
Prompt Feature Extractor
$\boldsymbol{z}_\mathrm{p}$
LoRA
Pre-trained LLM
$\boldsymbol{u}_\mathrm{kpt}$
FFN
③

{
"keypoint name": left ear;
"keypoint position": <keypoint>
}

What is the keypoint <keypoint> name of the object in the image? Please also accurately locate the position of the keypoint.

The left ear is situated on the left side of the head, specifically on the lateral aspect. Where is ***the left ear*** of the object in this image?

Figure 2: We introduce KptLLM, a unified framework designed to address three tasks of semantic keypoint comprehension: ① ***Keypoint Semantic Undertanding***, which processes a support image $\mathbf{I}_s$ and a support keypoint prompt $\mathbf{x}$ to generate responses that interpret the semantic information of the specified keypoint; ② ***Visual Prompt-based Keypoint Detection*** aims to detect the corresponding keypoint in the query image $\mathbf{I}_q$ based on the understanding of the support keypoint prompt; ③ ***Textual Prompt-based Keypoint Detection*** leverages textual keypoint descriptions to directly infer the corresponding keypoint positions in the query image.

tuning LLMs. However, it may fall short of fully leveraging the comprehensive reasoning faculties of LLMs. Kosmos-2 [67], Qwen-VL [68] and DetGPT [69] further exploit the power of LLMs to enable user-guided detection. Moreover, GPT4RoI [70], Ferret [71], Shikra [72], and PerceptionGPT [73] innovates by incorporating spatial boxes or masks as inputs and training with region-text pairs, offering region-level visual comprehension. Notably, a concurrent work, LocLLM [74], utilizes LLMs for human keypoint localization via textual description. In contrast, we take a step further by enabling LLMs to comprehend keypoints of various objects via multi-modal (*e.g.*, textual or visual) prompts under different task formulations. This advancement not only broadens the utility of MLLMs for keypoint detection but also enhances interpretive depth, allowing for a more comprehensive understanding and grounding across a wider range of visual information.

3 Methodology

This section introduces our proposed unified framework, referred to as KptLLM, which effectively addresses three semantic keypoint comprehension scenarios. As illustrated in Fig. 2, KptLLM accepts multiple images (query and support images) along with a support keypoint prompt (*i.e.*, the position of the target keypoint in the support image) and textual user instructions as the input. The output comprises both the response text and the desired keypoint position. Specifically, KptLLM comprises four key architectural components: (1) A visual encoder that extracts features from both query and support images (see Sec. 3.1); (2) A prompt encoder that converts support keypoint prompts into prompt embeddings (see Sec. 3.2); (3) A prompt feature extractor that derives prompt-oriented features from the corresponding image features (see Sec. 3.3); (4) A pre-trained LLM that processes multimodal tokens for keypoint comprehension (see Sec. 3.4).

3.1 Visual Encoder

The Visual Encoder is designed to process two types of images in parallel: query and support images. Generally, it receives an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and generates a feature map $\mathbf{F} = \mathcal{V}(\mathbf{I}) \in \mathbb{R}^{h\times w\times d}$. Here, d represents the feature dimension, and h and w are the spatial dimensions obtained by downsampling the original image dimensions H and W.

Query Image. The query image represents the image that is to be analyzed. We extract its spatial features through the vision encoder $\mathcal{V}$, resulting in $\mathbf{F}_q$. Following LLaVA [10], we apply a linear layer to project $\mathbf{F}_q$ into language space: $\mathbf{z}_q = \texttt{Linear}(\mathbf{F}_q)$. As a result, query visual tokens aligned with the LLM dimension are obtained and fed to the LLM.

Support Image. The support image serves as a reference example. We extract its spatial features, which are represented as $\mathbf{F}_s$. Unlike the query image features, $\mathbf{F}_s$ is not directly input into LLM. Instead, it is processed by the prompt feature extractor to derive prompt-oriented features.

3.2 Prompt Encoder

In addition to processing images, we need to incorporate an additional prompt consisting of 2D coordinates $\mathbf{x} \in \mathbb{R}^2$, which describes the keypoint location within the image. Inspired by SAM [75], we introduce a prompt encoder to adapt this prompt input to be aligned with the image feature space $\mathbf{F}$. The prompt encoder encodes the keypoint coordinates using a sine-cosine position embedding (`PE`), followed by a Multi-Layer Perceptron (`MLP`):

$$\mathbf{F}_p = \texttt{MLP}(\texttt{PE}(\mathbf{x})). \tag{1}$$

3.3 Prompt Feature Extractor

The Prompt Feature Extractor is designed to extract the prompt-specific features from image features. As illustrated in Fig. 2, the semantics of the keypoint prompt directly correspond to the support image. We initialize the prompt feature extractor with a two-layer transformer that incorporates the cross-attention mechanism (`CrossAttnLayers`). This mechanism employs $\mathbf{F}_p$ as the query and $\mathbf{F}_s$ as the key and value to extract keypoint-specific visual features indicated by the prompt:

$$\mathbf{z}_p = \texttt{CrossAttnLayers}(\mathbf{F}_p, \mathbf{F}_s), \tag{2}$$

where $\mathbf{z}_p$ denotes the keypoint-specific visual features. In essence, compared with average pooling-based feature extraction method [21], the prompt feature extractor is trainable and capable of incorporating global image features to enhance keypoint identification. This is particularly beneficial for distinguishing mirror-symmetric keypoints, such as the left and right eyes, which can be highly ambiguous when relying solely on local image features. Our ablation study demonstrates the performance improvements achieved through the utilization of this component.

3.4 Multimodal LLM for Keypoint Comprehension

Given a query image and an optional prompt specifying the keypoint of interest, our goal is to generate textual descriptions and keypoint locations that convey fine-grained keypoint information within the image. Recognizing the exceptional ability of LLMs in handling multimodal tokens for different perception tasks [76, 10, 77, 73, 78], we further leverage LLM for keypoint comprehension, which could effectively process various inputs: (1) the visual tokens $\mathbf{z}_q$ of the query image, (2) the prompt tokens $\mathbf{z}_p$, and (3) a sequence of language tokens $\mathbf{t}$, which depend on the three semantic keypoint comprehension scenarios.

Keypoint Semantic Decoding. We design the model to directly generate textual descriptions that interpret keypoint semantics, following the standard approach used by LLMs for text generation. Generally, the architecture of an LLM typically comprises Transformer layers (`TransformerLayers`) followed by a final Feed Forward Network (`FFN`). The latent embedding $\mathbf{u}$, which captures the fused multimodal information, can be computed as:

$$\mathbf{u} = \texttt{TransformerLayers}([\mathbf{z}_q, \mathbf{z}_p, \mathbf{t}]), \tag{3}$$

where $[\mathbf{z}_q, \mathbf{z}_p, \mathbf{t}]$ denotes the concatenation of the visual, prompt, and language tokens. This embedding $\mathbf{u}$ is then passed through the `FFN` and a `Softmax` function to generate the probability distribution $\mathbf{p}$ over the vocabulary for the next token:

$$\mathbf{p} = \texttt{Softmax}(\texttt{FFN}(\mathbf{u})). \tag{4}$$

Keypoint Position Decoding. Inspired by [78, 73], we introduce a special token, `<keypoint>`, into the vocabulary. Consequently, the 2D keypoint position $\mathbf{y}$ can be computed from the output latent embedding $\mathbf{u}_{\text{kpt}}$ of the special token using another `FFN` prediction head:

$$\mathbf{y} = \texttt{FFN}(\mathbf{u}_{\text{kpt}}) \in \mathbb{R}^2. \tag{5}$$

Indentify-then-Detect (ItD) Strategy. Instead of relying on extensive data fitting to learn fixed keypoint localization patterns, we adopt an approach where the LLM initially identifies the semantics of keypoints. Subsequently, a chain-of-thought process is employed to accurately detect the locations of these keypoints. Experiments demonstrate improved performance compared to alternative baselines.

https://doi.org/10.52202/079017-4467

3.5 Training and Inference Details

To retain the learned general knowledge of the pre-trained LLM, we employ LoRA [79] for efficient fine-tuning of LLM, while fully fine-tuning other modules of the framework. The training and inference processes for different tasks are outlined below.

Keypoint Semantic Understanding. As shown in Fig. 1-(a), this task focuses on extracting semantic textual information associated with specific keypoints within an image. The training objective is to minimize the language modeling loss, computed as the cross-entropy loss over the vocabulary of the LLM's tokenizer. Specifically, the loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{lm}}(\mathbf{a}, \hat{\mathbf{a}}), \tag{6}$$

where $\mathbf{a}$ is the text response predicted by the model, and $\hat{\mathbf{a}}$ is the ground-truth text response. During inference, given an image provided by the user and the corresponding keypoint position as a prompt, our model comprehends and generates the semantic meaning of the specified keypoint.

Visual Prompt-based Keypoint Detection. This task involves simultaneously comprehending the semantics of keypoints, generating textual descriptions of this understanding, and precisely localizing the keypoint coordinates, as shown in Fig. 1-(b). The overall training function further incorporates the L1 loss for keypoint regression:

$$\mathcal{L} = \lambda \|\mathbf{y} - \hat{\mathbf{y}}\| + \mathcal{L}_{\mathrm{lm}}(\mathbf{a}, \hat{\mathbf{a}}), \tag{7}$$

where $\hat{\mathbf{y}}$ is the ground-truth keypoint position, and λ is the loss weight that balances the learning of keypoint regression and text generation ($\lambda = 2$ in our implementation). During inference, the user provides two images: one as the query image for testing and the other as the support image for reference. Additionally, the keypoint definition for the support image should be provided as the support keypoint prompt. Our model then comprehends the semantics of the desired keypoint and detects its corresponding position in the query image.

Textual Prompt-based Keypoint Detection. As illustrated in Fig. 1-(c), this task aims to accurately localize keypoints based on the detailed keypoint descriptions. The loss function aligns with that of the visual prompt-based keypoint detection. During inference, users have the option to provide detailed descriptions of the desired keypoints or simply the keypoint names, based on which our model can detect the corresponding keypoints.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

In our experiments, we employ two datasets to evaluate the semantic keypoint comprehension in three scenarios: (1) The MP-100 dataset [21] for both ***Keypoint Semantic Undertanding*** and ***Visual Prompt-based Keypoint Detection***: This dataset is a pioneering dataset for category-agnostic pose estimation, which encompasses 100 different object categories with over 20,000 instances. The number of keypoints varies across categories, ranging from 8 to 68. Following the protocols established by POMNet [21], the dataset is divided into five distinct splits to ensure comprehensive coverage across different model training and validation scenarios. Each split contains all 100 categories, with 70 for training, 10 for validation, and 20 for testing. The splits are carefully designed to avoid category overlap, maintaining the independence and integrity of training and testing scenarios.

(2) The AP-10K dataset [32] for ***Textual Prompt-based Keypoint Detection***: The dataset comprises 23 animal families and 54 species, totaling 10,015 images. Each image is annotated with 17 keypoints, including two eyes, one nose, one neck, two shoulders, two elbows, two knees, two hips, four paws, and one tail. We follow CLAMP [23] to assess the models' ability to generalize to previously unseen animal species within a zero-shot learning paradigm. We establish two experimental scenarios based on the taxonomic relationship between the species in the training and test sets—specifically, whether they belong to the same animal order. Species within the same order typically share similar visual characteristics, whereas those from different orders exhibit greater diversity in appearance. These scenarios enable us to assess how different methods perform when generalizing to unseen species under varying conditions. Following CLAMP, we assign Bovidae and Canidae as the training and

Table 2: **Visual Prompt-based Keypoint Detection** on MP-100 [21] dataset. Performance (PCK) under 1-shot and 5-shot settings.

Methods	**1-shot**						**5-shot**					
	Split1	Split2	Split3	Split4	Split5	Mean	Split1	Split2	Split3	Split4	Split5	Mean
ProtoNet [80]	46.05	40.84	49.13	43.34	44.54	44.78	60.31	53.51	61.92	58.44	58.61	58.56
MAML [81]	68.14	54.72	64.19	63.24	57.20	61.50	70.03	55.98	63.21	64.79	58.47	62.50
Finetune [82]	70.60	57.04	66.06	65.00	59.20	63.58	71.67	57.84	66.76	66.53	60.24	64.61
POMNet [21]	84.23	78.25	78.17	78.68	79.17	79.70	84.72	79.61	78.00	80.38	80.85	80.71
CapeFormer [22]	89.45	84.88	83.59	83.53	85.09	85.31	91.94	88.92	89.40	88.01	88.25	89.30
KptLLM	**91.66**	**86.58**	**86.19**	**84.76**	**86.32**	**87.10**	**93.17**	**89.45**	**90.08**	**88.74**	**89.52**	**90.19**

test sets for the different order setting, while Canidae and Felidae are chosen as the training and test sets for the same order setting.

4.1.2 Evaluation & Metrics

Different evaluation methods and metrics are used for different tasks of semantic keypoint comprehension. (1) ***Keypoint Semantic Undertanding***: We use the MP-100 dataset [21] (Split-1), with the keypoint semantic labels adopted from X-Pose [52]. Some keypoints are excluded from the evaluation due to ambiguity or inadequacy in their descriptions, such as those used to describe the collar in the clothing category. By aggregating the results of tested keypoints, we derive corresponding accuracy rates (%). (2) ***Visual Prompt-based Keypoint Detection***: We employ the Probability of Correct Keypoint (PCK) metric, which is the standard evaluation measure for this task. Consistent with POMNet [21], we uniformly set the PCK threshold to 0.2 across all categories. Additionally, we compute and report the average PCK over all five data splits to provide a comprehensive indication of the model's overall effectiveness. (3) ***Textual Prompt-based Keypoint Detection***: Following CLAMP, we employ average precision (AP) as the primary metric for AP-10K. This metric is computed based on the object keypoint similarity (OKS). For detailed protocol definitions, please refer to [24].

4.1.3 Implementation Details

Architecture. We utilize LLaVA-V1.5-7B [10] as our base model, which incorporates the ViT-based visual encoder of CLIP for image encoding and Vicuna-7B (fine-tuned from Llama-2) as the LLM backbone. We employ LoRA for efficient fine-tuning LLM. Instead, all other modules, including the visual encoder, prompt encoder, prompt feature extractor, and a series of linear layers and feed forward networks, undergo full fine-tuning. The input image only contains a single object of interest, cropped according to the ground-truth bounding box and resized to 336×336, consistent with CLIP-ViT-L.

Training Details. LoRA parameters are configured with a rank of 128 and an alpha of 256. Optimization is conducted using AdamW, with a learning rate of 2e−4 and weight decay of 0. We utilize 8 NVIDIA A100-80G GPUs for training, and use the DeepSpeed engine to enhance training efficiency. Each GPU operates with a batch size of 16, and we employ a gradient accumulation step of 1.

4.2 Keypoint Semantic Understanding

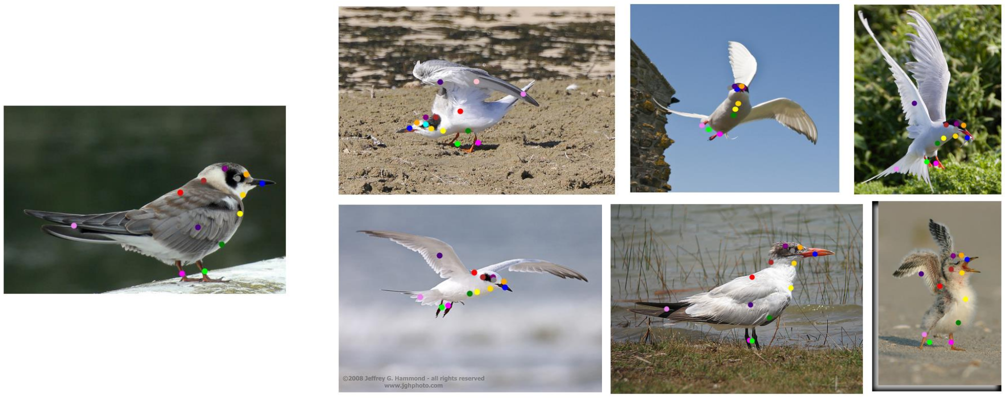
As depicted in Tab. 1, we present the accuracy for keypoint semantic understanding on MP-100 [21] Split-1 set. To facilitate a comprehensive comparison, we highlight the keypoint area in the image and feed the processed image, along with the task instruction, into LLaVA [10]. We report the performance of both the original LLaVA model and a version fine-tuned on the MP-100 dataset. The original LLaVA performs notably poorly in grasping keypoint semantics, indicating the inadequacy of traditional multimodal large language models in capturing fine-grained semantic details. Conversely, the fine-tuned LLaVA demonstrates significantly enhanced performance, thereby validating the efficacy of our training pipeline. Furthermore, our KptLLM surpasses the fine-tuned LLaVA by a substantial margin, particularly in terms of keypoint accuracy (83% vs 72%). It demonstrates the effectiveness of our keypoint prompt token in guiding attention to the fine-grained keypoint area.

Table 1: **Keypoint Semantic Understanding** on MP-100 (Split-1) [21].* means LLaVA is finetuned using LoRA.

Methods	Accuracy
LLaVA [10]	3%
LLaVA∗ [10]	72%
KptLLM	**83%**

Table 3: **Visual Prompt-based Keypoint Detection** for cross super-category evaluation on MP-100 [21]. Experiments are conducted under the 1-shot setting.

Method	Human Body	Human Face	Vehicle	Furniture
ProtoNet [80]	37.61	57.80	28.35	42.64
MAML [81]	51.93	25.72	17.68	20.09
Fine-tune [82]	52.11	25.53	17.46	20.76
POMNet [21]	73.82	79.63	34.92	47.27
CapeFormer [22]	83.44	80.96	45.40	52.49
KptLLM	**83.91**	**83.37**	**46.23**	**54.05**



(a) Support Image with Keypoints

(b) Query Image with Model Predictions

Figure 3: Using the same support image with support keypoints, our model could effectively detect different query images with various poses, object appearances, and environments.

4.3 Visual Prompt-based Keypoint Detection

1-shot & 5-shot Evaluation. We compare our method with the previous visual prompt-based methods ProtoNet [80], MAML [81], Fine-tune [82], POMNet [21], and CapeFormer [22]. Tab. 2 presents the PCK results of different approaches on the MP-100 dataset under both 1-shot and 5-shot settings. Compared with previous methods, KptLLM showcases the potential of MLLM in detecting keypoints through the use of visual prompts, consistently outperforming across all settings and data splits. More importantly, we integrate keypoint semantic understanding into the output response, introducing novel functionalities for comprehending the semantic aspects of support image keypoints.

Cross Super Category Evaluation. To thoroughly assess generalization across markedly different categories, we conduct a cross-supercategory evaluation following the protocol of POMNet [21]. While the MP-100 dataset ensures that training, validation, and test categories are non-overlapping, some categories may still exhibit similar features, *e.g.*, body characteristics commonly shared among different quadruped animals. To address this, we designate four supercategories—human face, human body, vehicle, and furniture—from the MP-100 dataset as test categories. The remaining categories are utilized for training, allowing us to better evaluate the model's ability to generalize across significantly diverse categories. As shown in Tab. 3, KptLLM consistently outperforms previous methods, highlighting the robustness and excellent generalization ability of our proposed method.

Qualitative Results. For the visual prompt-based keypoint detection task, the input necessitates a support image of the object to be tested, as well as keypoint positions that represent the definitions of those keypoints. In this study, we examine how the visual disparity between support images and query images affects the model's performance. As depicted in Fig. 3, our model is capable of effectively detecting keypoints in various query images when provided with the same support image and its corresponding keypoints. This effectiveness is maintained even in the presence of differences in object poses, appearances, and environmental conditions.

Table 4: **Textual Prompt-based Keypoint Detection** on AP-10K [32].

Method	Train	Test	AP	AP_{50}	AP_{75}	AP_M	AP_L	AR
SimpleBaseline [48]	Bovidae	Canidae	41.3	79.4	36.4	26.8	41.3	49.1
CLAMP [23]	Bovidae	Canidae	46.9	84.4	45.6	**30.3**	46.9	53.8
KptLLM	Bovidae	Canidae	**62.2**	**94.5**	**68.7**	26.0	**62.3**	**65.8**
SimpleBaseline [48]	Canidae	Felidae	39.6	74.1	34.5	9.5	40.2	46.6
CLAMP [23]	Canidae	Felidae	48.4	85.7	44.0	13.6	48.9	55.1
KptLLM	Canidae	Felidae	**70.2**	**97.8**	**82.2**	**49.2**	**70.6**	**74.7**

Table 5: Ablation study of semantic understanding.

Strategies	PCK
w/o ItD	87.68
w/ ItD	**91.66**

Table 6: Ablation study of prompt feature extraction.

Strategies	PCK
Average Pooling	89.78
Ours	**91.66**

Table 7: The effect on combining visual and textual prompts.

Visual	Textual	PCK
✓		91.66
✓	✓	**92.18**

4.4 Textual Prompt-based Keypoint Detection

The results are presented in Tab. 4. Compared to previous textual prompt-based model-CLAMP [23], KptLLM achieves superior cross-species generalization. Specifically, our model demonstrates a 15.3 average precision (AP) improvement in the different order setting and a 21.8 AP increase in the same order setting. Notably, our model performs better in the same order setting, where species often share similar visual characteristics. Overall, we show that leveraging detailed keypoint descriptions through comprehensive text, combined with commonsense knowledge from LLMs, effectively enhances generalizable performance in keypoint localization.

4.5 Ablation Study

In this subsection, we perform ablation study on the design choices of our model. The experiments are conducted on the visual prompt-based keypoint detection task using the MP-100 Split-1 setting with the PCK metric reported.

Indentify-then-Detect (ItD) Strategy. KptLLM follows the identify-then-detect paradigm, where the model learns to first interpret the semantic information of the keypoint to be detected, and then predict the precise location of the keypoint. In Tab. 5, we validate the effectiveness of our ItD strategy. We observe notable enhancement, which can be attributed to the inter-task synergy that arises from the ItD mechanism.

Prompt Feature Extractor. In Tab. 6, we compare our prompt feature extractor (Sec. 3.3) with the average pooling based feature extraction method [21]. The results show that our prompt feature extractor significantly outperforms the baseline method (91.66 vs 89.78), which validates the efficacy of our prompt feature extractor in enhancing focus on fine-grained keypoint areas.

Combining Visual and Textual Prompts. In Tab. 7, rather than relying solely on the visual prompt for localization, we further incorporate the textual prompt to demonstrate the effect of this combination. The improved results indicate that the textual prompt could provide valuable high-level and semantically rich guidance, enhancing keypoint localization.

5 Conclusion

This paper introduces the novel challenge of *Semantic Keypoint Comprehension*, which aims to comprehend keypoints across different task scenarios. To address this challenge, we present KptLLM, a novel and unified multimodal large language model designed to adeptly process various modality inputs, facilitating the interpretation of both semantic contents and keypoint locations. Extensive experiments show the superiority of our model in three different tasks for comprehending keypoints, including keypoint semantic understanding, visual prompt-based keypoint detection, and textual prompt-based keypoint detection. We hope this work can open up new possibilities for more fine-grained multimodal vision-language understanding and provide valuable insights for future research.

6 Discussion

Limitations. (1) A major limitation of our work is the model's size and computational efficiency, which is a common challenge for MLLMs compared to traditional vision models. However, this is acceptable because, as a pioneering effort in utilizing LLMs for keypoint comprehension, our main contribution is demonstrating the potential of LLMs to understand and locate pixel-level details at keypoints. (2) Additionally, the datasets used in our experiments have constraints in the diversity of object and keypoint categories for both training and testing. This highlights the need to expand these datasets to validate the model's applicability in more diverse, real-world scenarios.

Future Work. (1) Improving the Capacity of the Vision Encoder: Our work follows LLaVA [10] by employing a CLIP-based ViT as the vision encoder. However, some studies [83, 84] have demonstrated that stronger vision encoders can lead to more significant improvements, e.g., DINOv2 [85]. **(2) Refining Keypoint Decoding Strategy:** Inspired by previous MLLMs for perception tasks [78, 73], we introduce a special token `<keypoint>` into the model's vocabulary. When the model generates this `<keypoint>` token, its hidden embedding is decoded to the corresponding keypoint position. Although this strategy has shown promising results, it remains sub-optimal for user interaction. A more direct approach is to output the keypoint coordinates as textual descriptions. However, training a model to express numerical values in text using cross-entropy loss is challenging because slight deviations in numerical values can lead to significant differences in the generated text. Therefore, it intuitively requires more data for effective training. **(3) Expanding Data Scale and Category Diversity:** The datasets used in our experiments follow standard benchmarks. However, both the MP-100 dataset [21] for visual prompt-based keypoint detection and the AP-10K dataset [32] for textual prompt-based keypoint detection contain only a small amount of data, which limits the model's generalization performance. Furthermore, the limited diversity of object and keypoint categories greatly reduces the model's applicability, making it insufficient for handling open-world scenarios. A promising direction is to leverage large-scale keypoint datasets for training, such as UniKPT [52], which could further explore the upper bounds of MLLMs for keypoint comprehension.

Broader Impact. The study aims to enhance MLLMs for understanding images at a more granular keypoint level. We also propose a new challenge of keypoint semantic understanding, which holds promise for benefiting tasks such as structural understanding, action recognition, and medical image analysis. Nevertheless, recognizing the potential negative impacts that are common to many MLLMs, our model also carries risks, including the amplification of societal biases and concerns regarding privacy and ethics. To address these issues, we are committed to implementing safeguards, including strict access controls and the establishment of clear usage policies and agreements.

Acknowledgements

The work is partially supported by the Young Scientists Fund of the National Natural Science Foundation of China under grant No.62106154, by the Natural Science Foundation of Guangdong Province, China (General Program) under grant No.2022A1515011524, and by Shenzhen Science and Technology Program JCYJ20220818103001002, and by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong (Shenzhen).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

https://doi.org/10.52202/079017-4467

[4] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[6] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[7] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[8] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[11] 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.

[12] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023.

[13] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024.

[14] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[15] Xuan Ju, Ailing Zeng, Chenchen Zhao, Jianan Wang, Lei Zhang, and Qiang Xu. Humansd: A native skeleton-guided diffusion model for human image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15988–15998, 2023.

[16] Jie Yang, Chaoqun Wang, Zhen Li, Junle Wang, and Ruimao Zhang. Semantic human parsing via scalable semantic transfer over multiple label domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19424–19433, 2023.

[17] Jie Yang, Bingliang Li, Fengyu Yang, Ailing Zeng, Lei Zhang, and Ruimao Zhang. Boosting human-object interaction detection with text-to-image diffusion model. *arXiv preprint arXiv:2305.12252*, 2023.

[18] Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. Motionllm: Understanding human behaviors from human motions and videos. *arXiv preprint arXiv:2405.20340*, 2024.

[19] Jie Yang, Xuesong Niu, Nan Jiang, Ruimao Zhang, and Siyuan Huang. F-hoi: Toward fine-grained semantic-aligned 3d human-object interactions. *arXiv preprint arXiv:2407.12435*, 2024.

[20] Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: Text-aligned whole-body motion generation. *arXiv preprint arXiv:2310.12978*, 2023.

[21] 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.

[22] 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.

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23272–23281, 2023.

[24] 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 *Eur. Conf. Comput. Vis.*, 2014.

[25] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[26] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[27] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Eur. Conf. Comput. Vis.*, 2020.

[28] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 2016.

[29] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[30] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Eur. Conf. Comput. Vis.*, 2020.

[31] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Int. Conf. Comput. Vis.*, 2019.

[32] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

[33] Muhammad Haris Khan, John McDonagh, Salman Khan, Muhammad Shahabuddin, Aditya Arora, Fahad Shahbaz Khan, Ling Shao, and Georgios Tzimiropoulos. Animalweb: A large-scale hierarchical dataset of annotated animal faces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[34] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[35] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Int. Conf. Comput. Vis.*, 2021.

[36] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Int. Conf. Comput. Vis.*, 2019.

[37] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Int. Conf. Comput. Vis.*, 2017.

https://doi.org/10.52202/079017-4467

[38] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[39] Jie Yang, Ailing Zeng, Shilong Liu, Feng Li, Ruimao Zhang, and Lei Zhang. Explicit box detection unifies end-to-end multi-person pose estimation. *arXiv preprint arXiv:2302.01593*, 2023.

[40] Jie Yang, Ailing Zeng, Feng Li, Shilong Liu, Ruimao Zhang, and Lei Zhang. Neural interactive keypoint detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15122–15132, 2023.

[41] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[42] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[43] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[44] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *Eur. Conf. Comput. Vis.*, 2020.

[45] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Eur. Conf. Comput. Vis.*, 2016.

[46] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[47] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[48] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, 2018.

[49] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021.

[50] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. *arXiv preprint arXiv:2104.03516*, 2021.

[51] 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.

[52] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. Unipose: Detecting any keypoints. *arXiv preprint arXiv:2310.08530*, 2023.

[53] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[54] 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.

[55] 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. *Advances in neural information processing systems*, 35:23716–23736, 2022.

https://doi.org/10.52202/079017-4467

[56] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[57] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[58] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023.

[59] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[60] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024.

[61] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2023.

[62] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 2017.

[64] 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 *Int. Conf. Mach. Learn.*, 2021.

[65] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[66] 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.

[67] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[68] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[69] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*, 2023.

[70] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023.

[71] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[72] 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.

[73] Renjie Pi, Lewei Yao, Jiahui Gao, Jipeng Zhang, and Tong Zhang. Perceptiongpt: Effectively fusing visual perception into llm. *arXiv preprint arXiv:2311.06612*, 2023.

[74] 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.

[75] 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.

[76] Yuhang Zang, Wei Li, Jun Han, Kaiyang Zhou, and Chen Change Loy. Contextual object detection with multimodal large language models. *arXiv preprint arXiv:2305.18279*, 2023.

[77] Jiarui Xu, Xingyi Zhou, Shen Yan, Xiuye Gu, Anurag Arnab, Chen Sun, Xiaolong Wang, and Cordelia Schmid. Pixel-aligned language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13030–13039, 2024.

[78] 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.

[79] 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.

[80] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Adv. Neural Inform. Process. Syst.*, 2017.

[81] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Int. Conf. Mach. Learn.*, 2017.

[82] Akihiro Nakamura and Tatsuya Harada. Revisiting fine-tuning for few-shot learning. *arXiv preprint arXiv:1910.00216*, 2019.

[83] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[84] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Xiaopeng Zhang, Jin Li, Hongkai Xiong, and Qi Tian. From clip to dino: Visual encoders shout in multi-modal large language models. 2023.

[85] 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.

https://doi.org/10.52202/079017-4467

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims are validated by experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Sec. 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-4467

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details can be found in Sec. 4.1.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-4467

Answer: [Yes]

Justification: Our code will be released at `https://kptllm.github.io`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details can be found in Sec. 3.5, and experimental setups can be found in Sec. 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-4467

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Sec. 4.1.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics. And the research conform with it in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Sec. 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-4467

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Please refer to Sec. 6.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-4467