



OpenKD: Opening Prompt Diversity for Zero- and Few-shot Keypoint Detection (Supplementary Material)

Changsheng Lu¹, Zheyuan Liu¹, and Piotr Koniusz^{2,1}

¹The Australian National University ²Data61/CSIRO
changshenglu@gmail.com, zheyuan.david.liu@outlook.com,
piotr.koniusz@data61.csiro.au

Summary of the supplementary material:

- §A presents more details about **how to interpolate auxiliary texts** with LLM. The chain-of-thought prompting and the examples of text interpolation are given to showcase the efficacy of the proposed method.
- §B provides the examples of **diverse text prompts** and **text parsing**. More details about the construction of diverse text prompt sets are given.
- §C depicts more details about implementation.

A Details of Text Interpolation

We use the LLM to perform text interpolation $\mathcal{I}^t(z; t(\mathbf{p}_1), t(\mathbf{p}_2), c)$ by asking a good question, *i.e.*, designing a good prompt to LLM. The detailed **vanilla prompt** is given as follows:

“Please give me three most common body parts/keypoints at z between $t(\mathbf{p}_1)$ and $t(\mathbf{p}_2)$ of c . Pay attention to the left and right. Please answer in concise words like “1. 2. 3.”. Please do not include $t(\mathbf{p}_1)$ and $t(\mathbf{p}_2)$ in answers. Provide no excessive explanations.”

In the above formulation, z is the interpolation node, $t(\mathbf{p}_i)$ is the name of starting or end point, and c is the object name. In order to produce the structured output, several constraints can be appended to the prompt. For example, “*Pay attention to the left and right*” could help LLM take care of the symmetry of animal anatomy.

Although the vanilla prompt can deal with most cases, we require a more advanced prompt to improve the reasoning of LLM, as the keypoint types may be complex. The Chain of Thought (CoT) [48] has been demonstrated to be an effective method to improve the reasoning of LLM. Inspired by CoT, we propose the **improved prompt** with CoT as follows:

Q: Please give me one most common body part/keypoint at 1/2 between left-front knee and left-front paw of an animal. Please answer in concise words. Provide no excessive explanations.

A: The starting point is left-front knee. The end point is left-front paw. The answer should be between the starting point and end point. Left-front ankle is between the starting point and end point. The answer is left-front ankle.

Table 7: Examples of interpolating texts by prompting GPT3.5 with chain of thought on the Animal pose dataset. We use the API interface provided by OpenAI to perform reasoning. All the examples are generated automatically without human intervention.

System Instruct You are a helpful assistant that produces keypoints of an animal.

User Q: Please give me one most common body part/keypoint at 1/2 between left-front knee and left-front paw of an animal. Please answer in concise words. Provide no excessive explanations.
 A: The starting point is left-front knee. The end point is left-front paw. The answer should be between the starting point and end point. Left-front ankle is between the starting point and end point. The answer is left-front ankle.
 Q: Please give me three most common body parts/keypoints at 1/2 between nose and left ear of an animal. Pay attention to the left and right. Please answer in concise words like “1. 2. 3.”. Please do not include nose and left ear in answers. Provide no excessive explanations.

GPT3.5 1. Left eye ▷ Correct answer, mostly desirable
 2. Left cheek ▷ Correct answer, less desirable
 3. Left temple ▷ Correct answer, less desirable

User Q: Please give me one most common body part/keypoint...
 A: The starting point is ... The end point is ... The answer should be...
 Q: Please give me three most common body parts/keypoints at 1/2 between nose and right ear of an animal. Pay attention to the left and right. Please answer in concise words like “1. 2. 3.”. Please do not include nose and right ear in answers. Provide no excessive explanations.

GPT3.5 1. Right eye. ▷ Correct answer, mostly desirable
 2. Temple. ▷ Correct answer, less desirable
 3. Cheek. ▷ Correct answer, less desirable

User Q: Please give me one most common body part/keypoint...
 A: The starting point is ... The end point is ... The answer should be...
 Q: Please give me three most common body parts/keypoints at 1/2 between left-front thigh and left-front paw of an animal. Pay attention to the left and right. Please answer in concise words like “1. 2. 3.”. Please do not include left-front thigh and left-front paw in answers. Provide no excessive explanations.

GPT3.5 1. Left-front knee. ▷ Correct answer, mostly desirable
 2. Left-front ankle. ▷ Correct answer, less desirable
 3. Left-front paw. ▷ Wrong answer, not desirable

Q: Please give me three most common body parts/keypoints at z between $t(\mathbf{p}_1)$ and $t(\mathbf{p}_2)$ of c . Pay attention to the left and right. Please answer in concise words like “1. 2. 3.”. Please do not include $t(\mathbf{p}_1)$ and $t(\mathbf{p}_2)$ in answers. Provide no excessive explanations.

Examples: Table 7 shows three real examples of text interpolation using our improved prompt. Note that all the examples are generated at one-time run sequentially, without any human intervention, using our code. As demonstrated, our approach can successfully infer the auxiliary keypoint texts.

Listing 2.1: Examples of diverse text prompts on the Animal pose dataset, AWA, CUB, and NABird. For each dataset, we show five sampled diverse texts.

```
# Animal pose dataset
Please tell me the position of the left-front leg, left ear.
Can you find the left-back leg, left-front leg on cat?
Can you recognize the nose, left-front paw, left ear, left-back paw of
  dog?
Give me the position of the right-front knee, right-front leg.
Is there a way to pinpointing the left-back paw, right ear, right-back
  knee on cow?

# AWA dataset
Please pinpointing the front_right_knee, back_left_paw, left_eye,
  front_right_paw on ox.
Please determine the body_middle_left of otter.
The right_earbase, back_left_thai of wolf.
Give me the position of the back_left_thai, neck_base on zebra.
Please detect front_right_thai of grizzly bear.

# CUB dataset
Can you detect the crown, back, right eye, right wing of carolina wren?
Can you recognize the belly, right eye, right leg, forehead, nape, crown?
Spot the tail, right leg, back.
Where is the nape, right wing, tail, crown of a warbling vireo?
Please identify the throat, nape on brandt cormorant.

# NABird dataset
Can you locate the breast, belly, right eye, right wing, tail, nape on
  Bullock’s Oriole?
Please detect right wing, belly, bill, right eye.
Say’s Phoebe’s crown, back, breast, bill, belly, left eye.
Tell me where is the back, crown, nape, right wing of California Gull?
Ovenbird’s right eye, crown, belly, back.
```

B Details of Diverse Text Prompts

Examples of diverse texts: A glance of diverse text prompts is shown in Listing 2.1. For brevity, we show five randomly sampled diverse texts per dataset.

Examples of parsing: In our implementation, the detailed prompt to LLM for parsing is as follows:

Please extract the animal and keypoint keywords from the below text: “ $\langle text \rangle$ ”. Give the answer in simple words, like “Animal type:, Keypoint part:”. If no animal is mentioned, set animal type to N/A.

In the above example, the $\langle text \rangle$ is replaced by a specific diverse text prompt. For example, we select three diverse texts from Listing 2.1 and then use our proposed approach to parse them. Table 8 shows that our method can well extract the object category and keypoint texts from the diverse text.

Table 8: Examples of parsing texts via GPT3.5.

User	Please extract the animal and keypoint keywords from the below text: “Please tell me the position of the left-front leg, left ear.”. Give the answer in simple words, like “Animal type:, Keypoint part:”. If no animal is mentioned, set animal type to N/A.
GPT3.5	Animal type: N/A; Keypoint part: left-front leg, left ear ▷ Correct parsing
User	Please extract the animal and keypoint keywords from the below text: “Can you find the left-back leg, left-front leg on cat?”. Give the answer in simple words, like “Animal type:, Keypoint part:”. If no animal is mentioned, set animal type to N/A.
GPT3.5	Animal type: cat; Keypoint part: left-back leg, left-front leg ▷ Correct parsing
User	Please extract the animal and keypoint keywords from the below text: “Tell me where is the back, crown, nape, right wing of California Gull?”. Give the answer in simple words, like “Animal type:, Keypoint part:”. If no animal is mentioned, set animal type to N/A.
GPT3.5	Animal type: California Gull; Keypoint part: back, crown, nape, right wing ▷ Correct parsing

Moreover, in Table 5b (main paper), we randomly draw 1000 diverse texts from each dataset and then perform parsing. The parsed results will be compared to groundtruth (GT). A parsed text is correct if its IoU to GT ≥ 0.9 . The *Acc. kp* and *Acc. obj* in Table 5b (main paper) refer to the parsing accuracy of keypoint/object texts. As one can see, our parsing method with GPT3.5 can achieve over 96% accuracy in parsing keypoints from text, which shows that LLM is a good language parser.

Diverse texts construction: To convincingly evaluate the effectiveness of our approach, we construct *four* diverse text prompt sets for the popular Animal pose dataset, AwA, CUB, and NABird. Firstly, we manually and meticulously collect 100 **diverse text templates** which cover most real-world scenarios, as shown in Listing 2.2. Secondly, for each annotated instance in each dataset, we randomly sample one template and *one to N valid keypoints*. Lastly, we replace the $\langle \text{keypoint} \rangle$ and $\langle \text{object} \rangle$ (if existing) with the sampled keypoints and object category. For example, if the sampled template is “where is the $\langle \text{keypoint} \rangle$ for $\langle \text{object} \rangle$?”, the sampled keypoints are “left eye and right eye”, and the object category of annotated instance is “cat”, then we can synthesize a diverse text prompt as “where is the left eye and right eye for cat?”. We note that each annotated instance may have a varying number of valid keypoints due to self-occlusion, pose, and appearance. Thus, the synthesized text is quite diverse.

Listing 2.2: Partial diverse text templates. For brevity, we show 20 templates out of 100.

```

<obj>'s <keypoint>.
the <keypoint> of <obj>.
detect the <keypoint> of <obj>.

```

```

can you detect the <keypoint> of <obj>?
where is the <keypoint> for <obj>?
please detect <keypoint> of <obj>
please identify the <keypoint> on <obj>.
recognize the <keypoint> of <obj>.
please recognize the <keypoint> of <obj>.
Spot the <keypoint> of <obj>.
Locate the <keypoint> on <obj>.
please find the <keypoint> of <obj>.
distinguish the <keypoint> on <obj>.
please determine the <keypoint> of <obj>.
pinpointing the <keypoint> on <obj>.
pick out the <keypoint> on <obj>.
Could you find the <keypoint> on <obj>?
Please make out the <keypoint> on <obj>.
Give me the position of the <keypoint> on <obj>.
Please tell me the position of the <keypoint> on <obj>

```

C Additional Implementation Details

By default, the adaptation nets \mathcal{A}_v and \mathcal{A}_t of our OpenKD model use one bottleneck [17] and one transformer block [43], respectively. The decoder \mathcal{D} uses two convolutional blocks. The upsampler \mathcal{U} upscales the heatmap at a ratio of 2. During text sampling with false text control (FTC), we set $R = 3$ and $\eta = 1$ for Animal pose and AwA datasets, $R = \eta = 3$ for NABird, and $R = \eta = 10$ for CUB. For all datasets, we set the threshold α to 0.01. Considering the keypoint-text alignment for adapted features is weak in early training, we use 10k episodes for bootstrapping. Specifically, before 10k episodes, we adopt the original CLIP image/text features for FTC. After 10k episodes, we switch to the adapted CLIP image/text features. We note that these hyper-parameters are easy to tune and perform well in experiments.