Test-time Vocabulary Adaptation for Language-driven Object Detection

Supplementary Material

In this supplementary material, we first presents further experimental studies, including the impact of synonyms on VocAda-LLM and VocAda-CLIP, as well as a computational cost study of the proposed method. Next, we detail the evaluation metrics in Supp. B. Lastly, Supp. C provides the complete prompts for the Image Captioner (**IC**) and the LLM-based Class Selector (CS), along with their design details. We will publicly release our code and the intermediate results upon publication.

A. FURTHER STUDIES

Table 5: **Influence of synonyms on VocAda-LLM**, using the Detic detector and the OVE-COCO benchmark.

Method	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}
Baseline	27.8	51.1	45.0	-	-	-
w/o Synonyms	30.5	49.5	44.5	+2.7	-1.6	-0.5
w/ Synonyms	30.6	52.9	47.1	+2.8	+1.8	+2.1

Influence of synonyms on VocAda-LLM. Including synonyms for class names in the user-defined vocabulary when prompting the LLM-based **CS** module is a simple but crucial design choice, as shown in Tab. 5. During our initial exploration, we found, for example, that without synonyms some large and obvious objects like “Couch" or “TV” were often missed by the LLM-based **CS** of VocAda, even though they were included in the image descriptions. This occurred because these categories were phrased differently in the captions (*e.g.*, “Sofa” or “Television”) and, hence, the LLM-based **CS** processed them as not relevant and discarded them. Including synonyms as cues in the system prompt of the LLM prevents this erroneous filtering, resulting in superior performance.

Influence of synonyms on VocAda-CLIP. Using synonyms in the CLIP-based **CS** is less straightforward. Implementing synonyms for nouns would require querying an LLM at test time for each noun phrase, adding significant extra cost. We experimented with using synonyms for the class names queried offline, but this had no effect on the VocAda-CLIP results.

Study of computational cost. Tab. 6 compares inference speed and computational requirements. Metrics were measured on a Tesla V100 (32GB) using Detic Swin-B [5] as the detector, LLaVA-Next-7B [6] as the VLM, Llama3-8B [18] as the LLM, and CLIP ViT-L/14 [4], with the COCO-val [9] dataset. Although inference time increases, our method requires manageable resources (17GB for VocAda-CLIP and 31GB for VocAda-LLM), making it suitable for real-world applications. While VLMs and LLMs do slow down the vanilla detection pipeline, this can be mitigated with advanced deployment strategies like TensorRT or SGLang, which can speed up LLaVA and Llama3 by 6X and 2X, respectively.

Table 6: **Study of Computational Cost.** We evaluate the inference time and computational cost of the baseline detector (Detic Swin-B) and its integrations with VocAda-CLIP and VocAda-LLM on a Tesla V100 (32GB) using the COCO-val dataset with a batch size of 1. VocAda-LLM employs LLaVA-Next-7B [6] as the VLM and Llama3-8B [18] as the LLM, while VocAda-CLIP uses CLIP ViT-L/14.

Methods	Speed (sec/img)	GPU Requirement
Detic	0.115	8 GB
Detic w. VocAda-CLIP	5.572	17 GB
Detic w. VocAda-LLM	10.699	31 GB

Importantly, we believe that current computational limits should not hinder exploring new paradigms. Large VLMs and LLMs are increasingly integrated into detection pipelines and co-run with detectors in applications like autonomous driving. In such systems, VocAda adds minimal overhead by utilizing existing VLM outputs (captions).

Additional qualitative results. We present additional qualitative comparison of Oracle, Baseline, VocAda-CLIP and VocAda-LLM in Fig. 7. As observed, the baseline detector using the full vocabulary is easily confused by distracting classes, incorrectly classifying a “Curling” on a sports court as a “Teapot”. VocAda alleviates this confusion by adapting the vocabulary to the input image based on its interpretation of the semantic context. Even when VocAda does not lead to a correct detection, at least it avoids a mis-detection (see the curling stone in the right panel).

B. DETAILS ON THE EVALUATION METRICS

As illustrated in Fig. 8, given a predicted bounding box and the closest ground truth box, the Intersection over Union (IoU) is the ratio of their intersection area to their union area. For each object class, predictions are sorted by their confidence scores in descending order, and Average Precision (AP) is calculated as the area under the precision-recall curve. This combines precision and recall to provide a single performance measure for detection tasks.

Mean Average Precision (mAP) is the mean of the AP values, averaged across novel (unseen), base (seen), or all classes, denoted by AP^{novel}, AP^{base}, and AP^{all}, respectively. AP_{50} refers to mAP when IoU is considered with a threshold of 0.5. Otherwise, AP values are computed for thresholds from 0.5 to 0.95 in steps of 0.05 and then averaged.

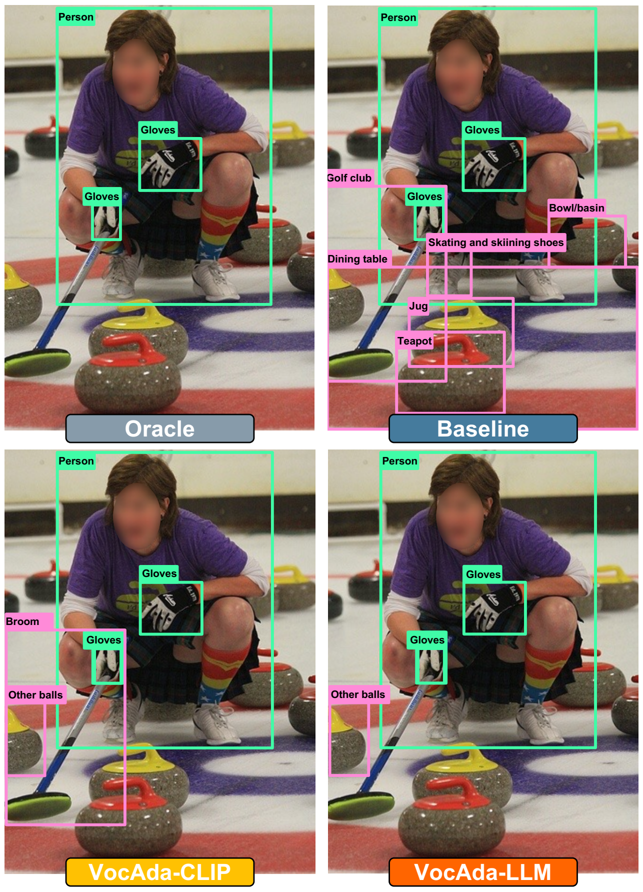


Fig. 7: **Additional Qualitative Results on Objects365.** We use Detic (Swin-B backbone) trained on LVIS and ImageNet-21k as the OvOD detector. Correct and incorrect detections appear in green and pink, respectively, with a 0.5 confidence threshold.

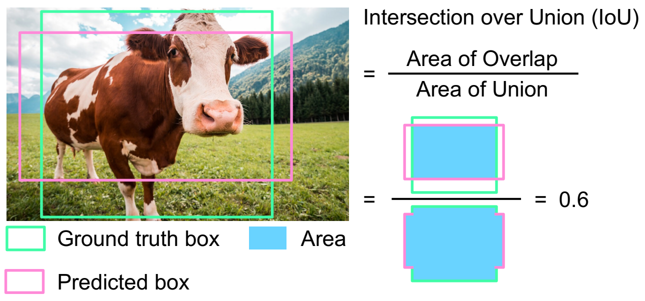

Intersection over Union (IoU)
= Area of Overlap / Area of Union
= 0.6
Ground truth box
Area
Predicted box

Fig. 8: Example of Intersection over Union (IoU) calculation.

Taking the calculation of AP_{50} as an example, we start by computing the IoU for each predicted bounding box and ground truth pair. A prediction is considered a True Positive (TP) if: *i)* its IoU is 0.50 or higher, and *ii)* its predicted class label matches the ground truth; otherwise, it's a False Positive (FP). Detections are sorted by confidence scores in descending order, and for each prediction, we evaluate its IoU and class label against the ground truth. Precision and recall are calculated at each detection: precision is the ratio of TPs to the total number of predictions (TPs + FPs), and recall is the ratio of TPs to the total number of ground truth objects (TPs + FNs). These values are used to plot the precision-recall curve, and the area under this curve represents the AP_{50} measurement. The final AP_{50} is averaged across all evaluated classes, summarizing the model's performance in terms of both localization and classification for the test dataset.

C. PROMPT DETAILS

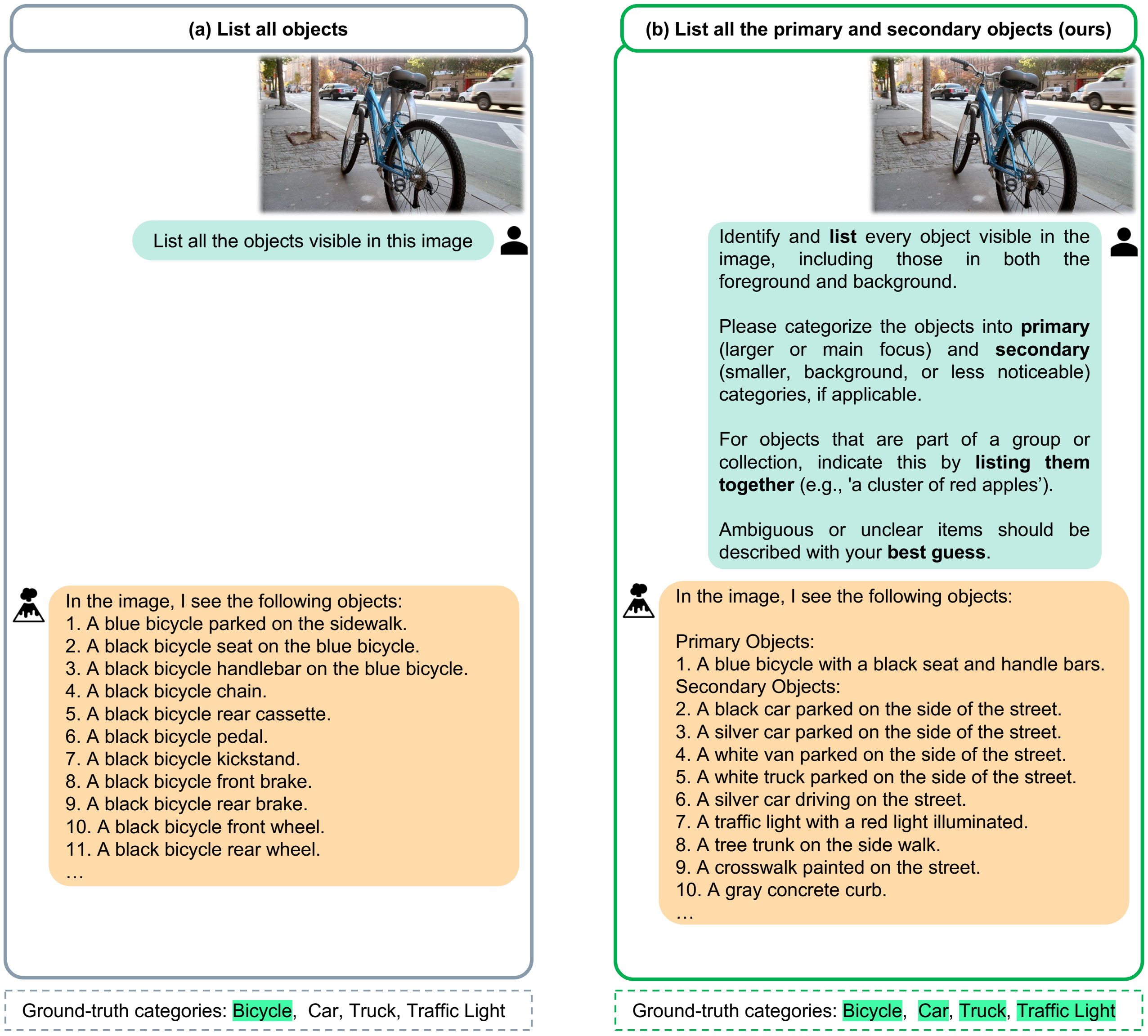
In this section, we provide the complete prompts used in VocAda and elaborate on the design choices for prompting.

C.1. Prompting Image Captioner (IC)

The comprehensiveness of the description generated by the **IC** is crucial for the subsequent steps of VocAda. The image description should capture as many categories present in the current image as possible. Even state-of-the-art VLMs often neglect background objects in images, focusing on more prominent foreground objects when prompted with a simple prompts such as "List all the objects visible in this image". For instance, as shown in Fig. 9(a), although the cars and trucks in the background are clearly visible, the **IC** only describes the foreground object "bicycle". To address this, as shown in Fig. 9(b), we propose a prompt strategy that instructs the **IC** to not only list all visible objects but also categorize them into *primary* and *secondary* groups. Even though we do not need the grouping results *per se*, this technique effectively guides the **IC** to comprehensively describe both large and focused foreground objects (primary) and small and background objects (secondary), such as "Traffic Light" in Fig. 9(b).

In Fig. 9(b), we show the full prompt used for the **IC** (LLaVA-Next-7B [6]) in VocAda to describe the image, creating textual measurements of the objects visible in the image.

In addition, there are two design choices worth mentioning. First, in our prompt, we instruct the **IC** to list a group of objects together (*e.g.*, "a cluster of red apples") instead of one by one. This technique prevents the **IC** from generating *repetitive* patterns, which are lengthy and not useful for the following steps. The goal of the **IC** is to comprehensively capture object categories likely to appear in the current images. Therefore, we further ask the **IC** to provide "best guesses" for unclear items in the prompt. This design force the IC to reason possible objects that might be present in the image based on its interpretation. While this might introduce extra noise, the Class Selector module can alleviate most of them, especially if they are unrelated to the global image context. In VocAda, we use the exact prompt shown in Fig. 9(b) for all experiments.



(a) List all objects

List all the objects visible in this image

In the image, I see the following objects:
1. A blue bicycle parked on the sidewalk.
2. A black bicycle seat on the blue bicycle.
3. A black bicycle handlebar on the blue bicycle.
4. A black bicycle chain.
5. A black bicycle rear cassette.
6. A black bicycle pedal.
7. A black bicycle kickstand.
8. A black bicycle front brake.
9. A black bicycle rear brake.
10. A black bicycle front wheel.
11. A black bicycle rear wheel.
...

Ground-truth categories: Bicycle, Car, Truck, Traffic Light

(b) List all the primary and secondary objects (ours)

Identify and **list** every object visible in the image, including those in both the foreground and background.

Please categorize the objects into **primary** (larger or main focus) and **secondary** (smaller, background, or less noticeable) categories, if applicable.

For objects that are part of a group or collection, indicate this by **listing them together** (e.g., 'a cluster of red apples').

Ambiguous or unclear items should be described with your **best guess**.

In the image, I see the following objects:

Primary Objects:
1. A blue bicycle with a black seat and handle bars.
Secondary Objects:
2. A black car parked on the side of the street.
3. A silver car parked on the side of the street.
4. A white van parked on the side of the street.
5. A white truck parked on the side of the street.
6. A silver car driving on the street.
7. A traffic light with a red light illuminated.
8. A tree trunk on the side walk.
9. A crosswalk painted on the street.
10. A gray concrete curb.
...

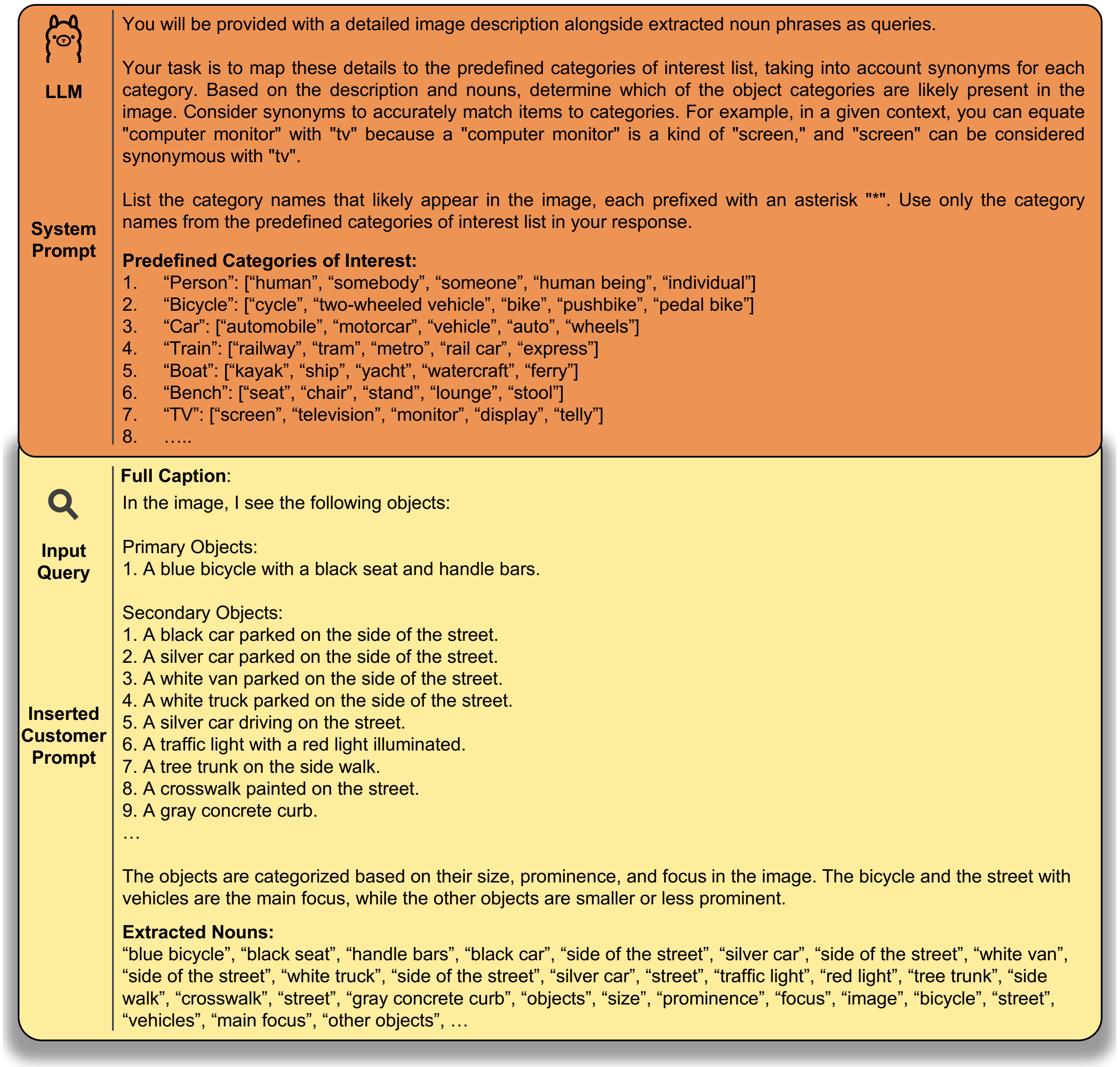
Ground-truth categories: Bicycle, Car, Truck, Traffic Light

Fig. 9: Comparison of image captioner prompts. (a) A simple prompt is used to list all the visible objects in the image. (b) We design a better prompt to improve the comprehensiveness of the description by instructing the captioner to list primary and secondary objects. The ground-truth categories that are mentioned by the output caption are highlighted in Green.

C.2. Prompting LLM as Class Selector (CS)

In Fig. 10, we present the complete system and customer prompts used for the LLM Proposal-based **CS** in VocAda. Specifically, we first instantiate a LLM agent, such as Llama3-8B [18], with a system prompt that includes a task instruction and the user-defined vocabulary with their synonyms. The task instruction specifies the input query, the generated image caption $\mathcal{S}_I$ provided by the **IC** and the corresponding extracted noun phrases $\mathcal{P}_I$ from the **NE**, that the LLM will receive during inference. It then guides the LLM with a detailed task description, which is to select relevant categories likely to appear in the image from the embedded user-defined vocabulary based on the input, taking also synonyms into consideration. Subsequently, the LLM is instructed with the output format of the selected categories (prefixing each category name with an asterisk "*") for easier post parsing. This LLM instantiation is conducted before large scale inference.

Therefore, during inference, the LLM-based **CS** takes the complete image description $\mathcal{S}_I$ and the corresponding noun phrases $\mathcal{P}_I$ as the user input without any additional instructions and automatically outputs a selected category set as $\widetilde{\mathcal{C}}_I$.


LLM
System Prompt
Input Query
Inserted Customer Prompt

You will be provided with a detailed image description alongside extracted noun phrases as queries.

Your task is to map these details to the predefined categories of interest list, taking into account synonyms for each category. Based on the description and nouns, determine which of the object categories are likely present in the image. Consider synonyms to accurately match items to categories. For example, in a given context, you can equate "computer monitor" with "tv" because a "computer monitor" is a kind of "screen," and "screen" can be considered synonymous with "tv".

List the category names that likely appear in the image, each prefixed with an asterisk "*". Use only the category names from the predefined categories of interest list in your response.

Predefined Categories of Interest:
1. “Person”: [“human”, “somebody”, “someone”, “human being”, “individual”]
2. “Bicycle”: [“cycle”, “two-wheeled vehicle”, “bike”, “pushbike”, “pedal bike”]
3. “Car”: [“automobile”, “motorcar”, “vehicle”, “auto”, “wheels”]
4. “Train”: [“railway”, “tram”, “metro”, “rail car”, “express”]
5. “Boat”: [“kayak”, “ship”, “yacht”, “watercraft”, “ferry”]
6. “Bench”: [“seat”, “chair”, “stand”, “lounge”, “stool”]
7. “TV”: [“screen”, “television”, “monitor”, “display”, “telly”]
8. …..

Full Caption:
In the image, I see the following objects:

Primary Objects:
1. A blue bicycle with a black seat and handle bars.

Secondary Objects:
1. A black car parked on the side of the street.
2. A silver car parked on the side of the street.
3. A white van parked on the side of the street.
4. A white truck parked on the side of the street.
5. A silver car driving on the street.
6. A traffic light with a red light illuminated.
7. A tree trunk on the side walk.
8. A crosswalk painted on the street.
9. A gray concrete curb.
…

The objects are categorized based on their size, prominence, and focus in the image. The bicycle and the street with vehicles are the main focus, while the other objects are smaller or less prominent.

Extracted Nouns:
“blue bicycle”, “black seat”, “handle bars”, “black car”, “side of the street”, “silver car”, “side of the street”, “white van”, “side of the street”, “white truck”, “side of the street”, “silver car”, “street”, “traffic light”, “red light”, “tree trunk”, “side walk”, “crosswalk”, “street”, “gray concrete curb”, “objects”, “size”, “prominence”, “focus”, “image”, “bicycle”, “street”, “vehicles”, “main focus”, “other objects”, …

Fig. 10: **Complete prompts used the LLM Proposal based Class Selector (CS). Top**: The system prompt includes the user defined categories enriched with a set of synonyms and the task instruction. The latter guides the LLM to select from the category list the ones that are relevant given as input an image description and the set of extracted noun phrases. This system prompt is used to instantiate the LLM agent as the **CS**. **Bottom**: During inference, the full image description, ($\mathcal{S}_I$ provided by the **IC**) alongside the extracted noun phrases ($\mathcal{P}_I$ from the **NE**) are fed to the system as customer prompt input. Subsequently, the LLM automatically propose the selected category names based on this input.

6. REFERENCES

[1] Ram Ramrakhya, Aniruddha Kembhavi, Dhruv Batra, Zsolt Kira, Kuo-Hao Zeng, and Luca Weihs, "Seeing the unseen: Visual common sense for semantic placement," in *CVPR*, 2024.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *ICCV*, 2017.

[3] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021.

[5] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra, "Detecting Twenty-thousand Classes using Image-level Supervision," in *ECCV*, 2022.

[6] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee, "LLaVA-NeXT: Improved reasoning, OCR, and world knowledge," 2024.

[7] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai, "Learning object-language alignments for open-vocabulary object detection," in *ICLR*, 2023.

[8] Chuofan Ma, Yi Jiang, Xin Wen, Zehuan Yuan, and Xiaojuan Qi, "CoDet: Co-Occurrence Guided Region-Word Alignment for Open-Vocabulary Object Detection," in *NeurIPS*, 2023.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014.

[10] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *ICCV*, 2019.

[11] Chaoyang Zhu and Long Chen, "A Survey on Open-Vocabulary Detection and Segmentation: Past, Present, and Future," arXiv:2307.09220, 2023.

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick, "LVIS: A Dataset for Large Vocabulary Instance Segmentation," in *CVPR*, 2019.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: a Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[14] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao, "RegionCLIP: Region-based Language-Image Pretraining," in *CVPR*, 2022.

[15] Mingxuan Liu, Tyler L. Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi, "SHiNe: Semantic hierarchy nexus for open-vocabulary object detection," in *CVPR*, 2024.

[16] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang, "mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration," in *CVPR*, 2024.

[17] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[18] AI Meta, "Introducing meta llama 3: The most capable openly available llm to date," *Meta AI*, 2024.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *ICCV*, 2021.

[20] Christiane Fellbaum, *WordNet: an Electronic Lexical Database*, MIT Press, 1998.

[21] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang, "Open-set image tagging with multi-grained text supervision," *arXiv:2310.15200*, 2023.

[22] Nils Reimers and Iryna Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *EMNLP*, 2019.

[23] 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:2310.06825*, 2023.

[24] OpenAI, "ChatGPT: A Large-Scale GPT-3.5-Based Model," `https://openai.com/blog/chatgpt`, 2022.

[25] Sachit Menon and Carl Vondrick, "Visual Classification via Description from Large Language Models," in *ICLR*, 2023.