DetRefiner: Model-Agnostic Detection Refinement with Feature Fusion Transformer

Supplementary Material

1. Reproducible Evaluation of Open-Vocabulary Object Detectors

Before applying DetRefiner, we reproduce ten representative open-vocabulary detectors across all benchmarks under a unified evaluation protocol. All reported improvements are measured with respect to these reproduced baselines to ensure consistent comparison.

GLIP [3] is implemented using its official repository[1] with `min_image_size` set to 800. All other detectors are instantiated via the HuggingFace *transformers* library[2][3][4].

For evaluation, we set the box score threshold to 0 for all models, ensuring that all predicted boxes are retained before computing AP. Unless otherwise specified, we use default inference-time settings for each model (e.g., NMS thresholds), except for the unified zero score threshold and the GLIP image resolution setting.

We note that differences in implementation (e.g., HuggingFace vs. official repositories), evaluation configurations such as the maximum number of predictions per image (300 for COCO and 100,000 for LVIS in our setup), and dataset-specific choices (e.g., ODinW13 test splits) may lead to discrepancies with originally reported results. To facilitate reproducibility, we release the full evaluation code and configurations at https://github.com/hitachi-rd-cv/detrefiner.

For **COCO** [4], we follow the common open-vocabulary setting and feed a single concatenated text prompt containing all 80 category names into models. The category names are concatenated in ascending order of the category IDs defined in the ground-truth annotations.

For **LVIS** [1], we adopt the same strategy as GLIP [3]: the 1,203 category names are first sorted by their category IDs and then partitioned into groups of 40, where each group is used as a separate text prompt. The last three categories, which do not form a full group of 40, are used as a single text prompt containing only these three category names.

For both COCO and LVIS, we apply a simple text preprocessing step to the category names: we convert all characters to lowercase, replace underscores and hyphens with spaces, and remove parentheses. We observed that these pre-processing steps have negligible impact on the final performance, but we apply them consistently for reproducibility.

Table 1. Detailed statistics of the ODinW13 datasets [2, 3].

Dataset	#Classes	Test Images
AerialMaritimeDrone-Large	5	15
Aquarium-Combined	7	127
CottontailRabbits	1	19
EgoHands-Generic	1	200
North-American-Mushrooms	2	5
Packages-Raw	1	4
PascalVOC	**20**	**3422**
Pistols-Export	1	297
Pothole	1	133
Raccoon	1	29
ShellfishOpenImages	3	116
ThermalDogsAndPeople	2	41
VehiclesOpenImages-416x416	5	200

For **ODinW13** [2, 3], we follow the official GLIP configuration[5]. We use as test images the datasets specified in the `DATASETS:TEST` field of each YAML file in the configuration directory, and we construct the text prompts from the corresponding `OVERRIDE_CATEGORY` field. The number of categories and test images for each ODinW13 dataset is summarized in Table 1.

2. Additional Visualization Results

Figure 1 illustrates how DetRefiner refines predictions from the base detector. It suppresses overconfident false positives and boosts missed true positives using both global and local cues. The bottom row indicates scene-level (class vector) and region-level (patch vector) calibration, which together improve open-vocabulary detection reliability.

pizza scene Fig. 1(a) shows a pizza image with several small toppings. The base detector (top-left) assigns low confidence to many pepper and mushroom instances, whereas applying the class and patch vectors (bottom row) upweights boxes supported by global and local cues, so the full DetRefiner (top-right) yields consistently high confidence for most true toppings.

Group photo Fig. 1(b) shows a crowded group photo with unusual color tones and many tiny neckties, socks, and awnings. Although categories such as *awning* and *sock* remain missed because the original detector cannot detect

[1] https://github.com/microsoft/GLIP/tree/main

[2] https://huggingface.co/docs/transformers/model_doc/grounding-dino

[3] https://huggingface.co/docs/transformers/model_doc/mm-grounding-dino

[4] https://huggingface.co/collections/iSEE-Laboratory/llmdet

[5] https://github.com/microsoft/GLIP/tree/main/configs

them even with a zero-threshold setting, DetRefiner (top-right) reliably detects tiny neckties in the crowd.

street scene Fig. 1(c) presents a street scene with signboard, lamppost, trousers, and manhole. Here DetRefiner uses the class vector to boost scene-consistent categories and the patch vector to further upweight boxes aligned with local structures, producing a more complete and reliable confidence distribution than the base detector.

indoor table scene Fig. 1(d) shows an indoor table with a flower arrangement, tablecloth, and knife. While the base detector mainly scores the central flower highly, DetRefiner raises the confidence of table-related boxes whose features match knives and tablecloth regions, leading to dense and well-calibrated scores for the relevant objects.

Figure 2 shows two additional scenes highlighting both the strengths and limitations of DetRefiner. For each example, the top row shows ground-truth boxes, the middle row compares the base detector (left) with DetRefiner (right), and the bottom row shows predictions from the class-vector branch (left) and the patch-vector branch (right).

skateboard scene In Figure 2(a), the base detector assigns low confidence to the wheels and tends to miss them. DetRefiner successfully recovers the wheels with higher confidence, indicating that fused global–local cues can rescue missed objects. At the same time, spurious boot predictions are actually reinforced and even newly introduced. Because the objects detected as boot are visually ambiguous and can be interpreted as boots, both the class and patch branches assign relatively high probability to *boot*, so the fused score becomes even larger than the base detector's score and the false positives remain.

street worker scene In Figure 2(b), the base detector hallucinates a recliner and misses several signboards and a broom. DetRefiner suppresses the recliner false positive and recovers some signboards, but still fails to detect the broom and some other signboards. In addition, it introduces visually plausible yet incorrect predictions such as *car_automobile* around the scooter, where the local appearance is highly confounding. Moreover, an overconfident *seahorse* prediction is only slightly reduced and remains above the threshold. These cases illustrate that DetRefiner is effective for moderate miscalibration and missed detections, but highly misaligned base scores can still dominate the final confidence and may even increase false positives for ambiguous classes.

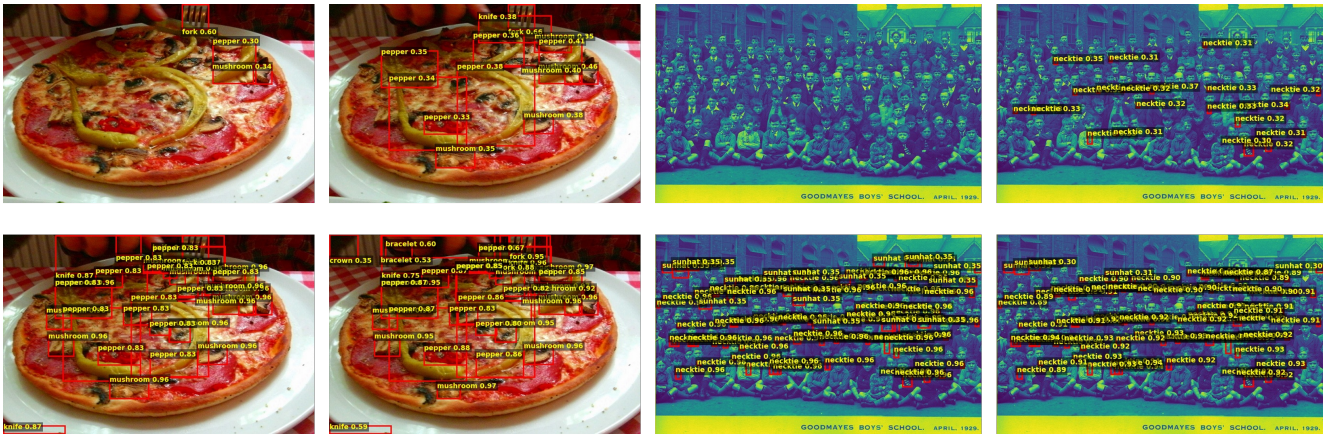
References

[1] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 1

[2] Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems*, 35:9287–9301, 2022. 1

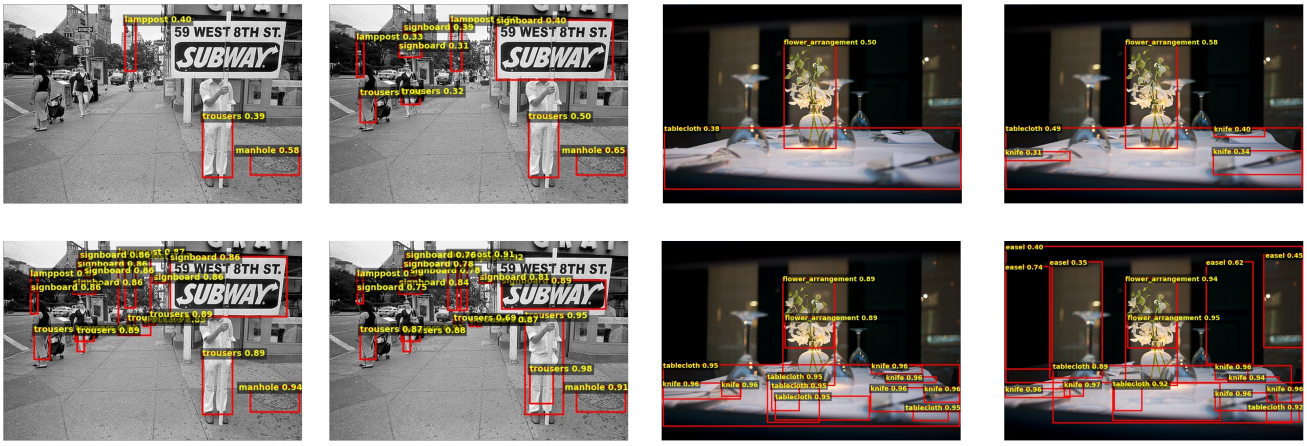
[3] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10975, 2022. 1

[4] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1



(a) Ground Truth: fork, knife, mushroom, pepper

(b) Ground Truth: awning, necktie, sock



(c) Ground Truth: lamppost, manhole, signpost, trousers

(d) Ground Truth: flower_arrangement, knife

Figure 1. Another qualitative comparison of detection results before and after applying DetRefiner. Top: base detector (left) vs. base detector + DetRefiner (right). Bottom: predictions based on the class vector (left) and patch vector (right). DetRefiner suppresses overconfident false positives and recovers missed objects by combining global and local cues. For visualization, a box score threshold of 0.3 and an IoU threshold of 0.3 are applied for class-wise NMS on all images.

(a) Ground Truth: skateboard, wheel

(b) Ground Truth: broom, motor_scooter, signboard

Figure 2. Additional qualitative success and failure cases. Top row: ground-truth bounding boxes. For each example, the middle row shows predictions from the base detector (left) and from the base detector with DetRefiner (right). The bottom row shows predictions from the class-vector branch (left) and patch-vector branch (right). For visualization, a box score threshold of 0.3 and an IoU threshold of 0.3 are applied for class-wise NMS on all images.