Supplementary Material: New Additions and Clarifications

1 New Additions and Clarifications

1.1 New Additions

- **Speed-Accuracy Tradeoff:** We now include a comprehensive analysis of the speed-accuracy tradeoff. Figure 1 shows that, with caching, our implementation of SearchDet achieves inference times comparable to state-of-the-art models (e.g., T-Rex 2) while maintaining competitive performance.
- **Scalability Enhancements:** Additional details have been added regarding scalability improvements. In particular, we now describe using caching for frequently queried classes and alternative scraping methods (such as using APIs like Bing ImageSearch) to reduce download latency compared to our current Selenium-based approach.
- **Technical Clarifications:** Further explanations address ambiguities in web-retrieved images, segmentation sensitivity, hyperparameter settings, and distinctions in our few-shot learning approach.

1.2 Detailed Explanation: Speed-Accuracy Tradeoff

Our method, SearchDet, is designed with real-world applicability in mind. To reduce download latency, we suggest caching frequently queried classes. Additionally, our modular framework allows the integration of faster alternatives to HQ-SAM (e.g., Efficient-SAM, which offers a 48.9% speedup, or PyTorch Compiled SAM, which is 8x faster than the base model). As illustrated in Figure 1, our unoptimized implementation achieves inference times on par with leading detection models while delivering state-of-the-art performance.

2 Qualitative Analysis of Binning

To further elucidate the improvements made in the final version, we introduce a new section on the qualitative analysis of our frequency-based adaptive thresholding and binning process for mask selection. In our method, after computing

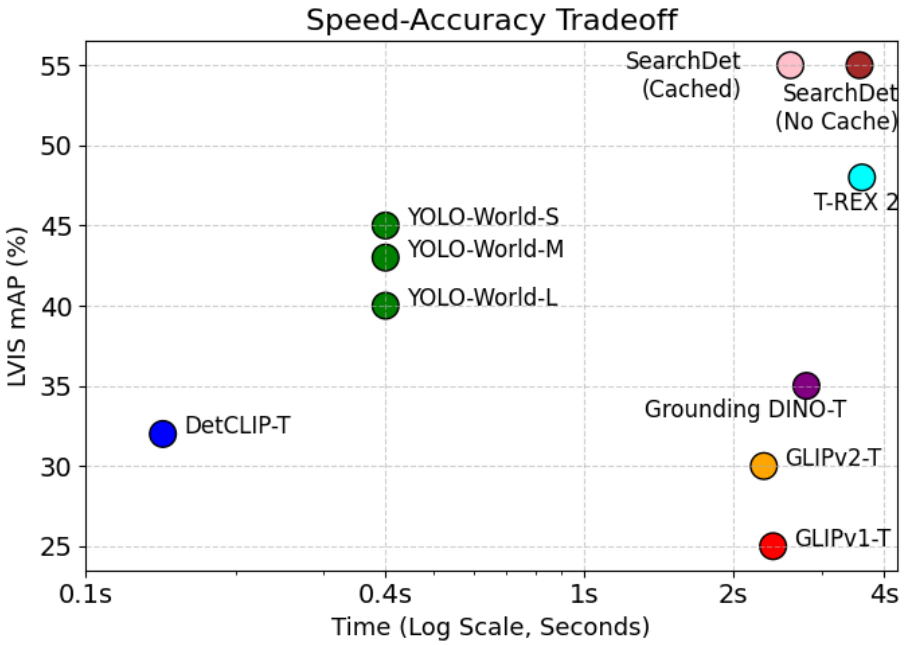

Figure 1: Speed-Accuracy Tradeoff: Performance vs. Inference Time on LVIS. With caching, SearchDet's inference time is comparable to GroundingDINO and faster than T-Rex.

the Euclidean distances between the adjusted query embedding and each segmented mask, the distances are sorted and partitioned into bins. Within each bin, the frequency of candidate masks is analyzed to select the most representative mask. This adaptive strategy robustly determines the threshold for mask selection even in the presence of noise or ambiguous segmentation proposals.

The following figures illustrate various stages of this process:

These visualizations demonstrate how our binning process enhances mask selection by isolating the most representative candidate in each bin. The sorted distance distribution is first divided into bins, then the frequency of distances corresponding to each mask within a bin is computed. If a mask dominates a bin (exceeding a preset threshold, e.g., 80%), it is selected; otherwise, it is discarded. This approach enables robust and adaptive mask selection, even under challenging conditions.

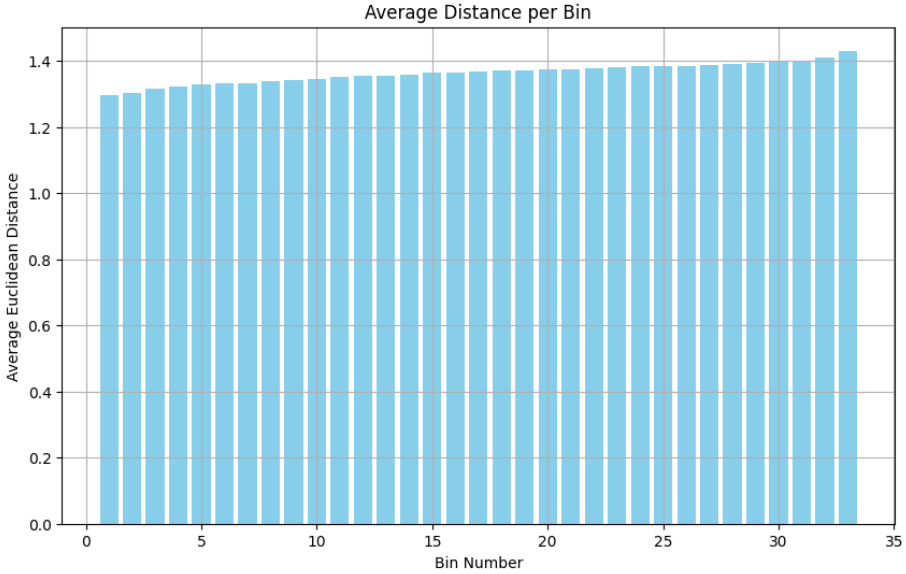

Figure 2: Sorted distance distribution between the query embedding and segmentation masks.

Figure 3: Mask selected.

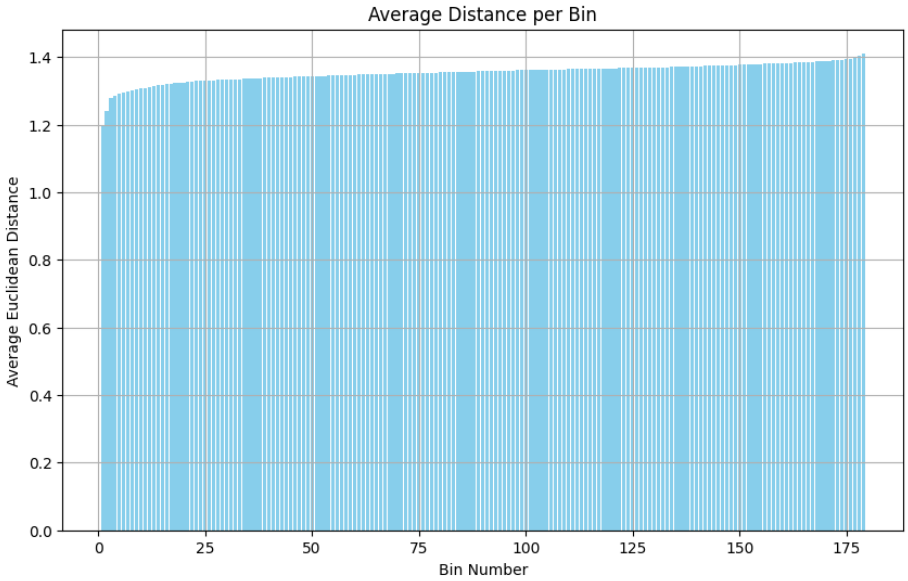

Figure 4: Sorted distance distribution between the query embedding and segmentation masks..

Figure 5: Final selected mask after applying frequency-based adaptive thresholding.