Supplemental Material for 'Beyond Weak Supervision: MLLMs-Guided Graded Knowledge Distillation for Unsupervised Camouflaged Object Detection'

Huafeng Chen[1] Chenguang Zhu[2] Yueming Lyu[1]✉, Caifeng Shan[1]✉
[1]Nanjing University [2]Northwestern Polytechnical University
{huafengchen@smail, ymlv@, cfshan@}nju.edu.cn zcg23@mail.nwpu.edu.cn

Contents

A. Limitations

Our method proposes Graded Knowledge Distillation (GKD), in which pixel-level enhancement utilizes a cross-entropy map to generate a weight map for selective pixel-wise distillation. This operation essentially distills pixels with high certainty. As shown in Figure 3 of the main text, the deweight regions are mostly located near the boundaries. However, there is no free lunch. While this approach reduces the risk of introducing noise, it also reduces the model's ability to learn boundary details. As a result, the predicted maps may show insufficient sharpness along the edges in some cases, as illustrated in Figure 1.

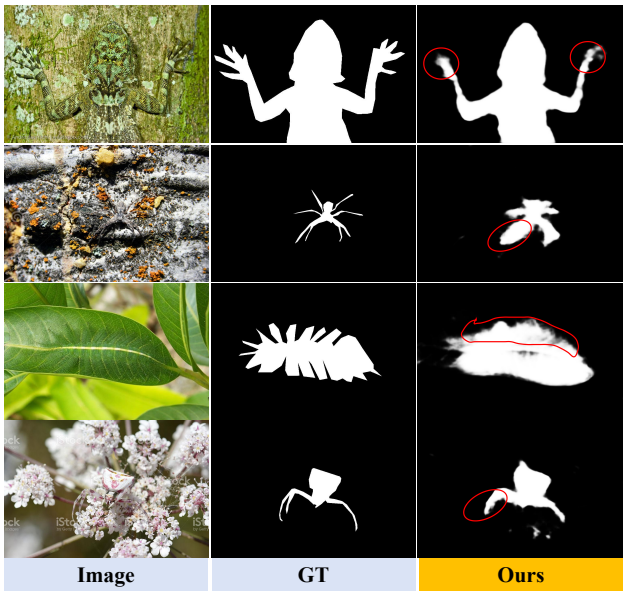

Figure 1. Failure cases.

B. Method Details

B.1. Details of ETC and FGC in GME

B.1.1. Details of ETC.

The bounding boxes generated by MLLMs are not entirely accurate and tend to be larger than ground truth boxes. Due to the inherently ambiguous boundaries between foreground and background in COD, when using these oversized boxes as prompts, SAM often erroneously responds to the entire background region rather than the actual foreground target, as shown in Figure 2 (a). Therefore, GME introduces ETC to count the number of response masks truncated by boundaries and filters out masks with excessive truncation numbers. Specifically, for a given candidate mask V_i, the ETC first determines whether foreground activation exists at each of its four boundaries. If foreground activation is detected at any boundary, the truncation num-

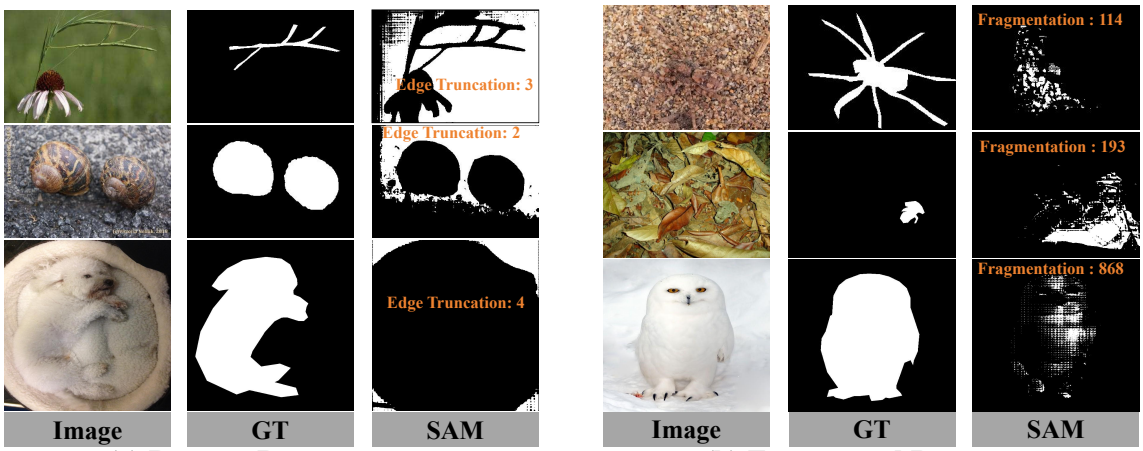

Figure 2. Visualization of poor masks in the normal-quality grade. a) reverse response: SAM erroneously segments the background region as foreground. b) fragmented response: SAM generates numerous discrete small response regions.

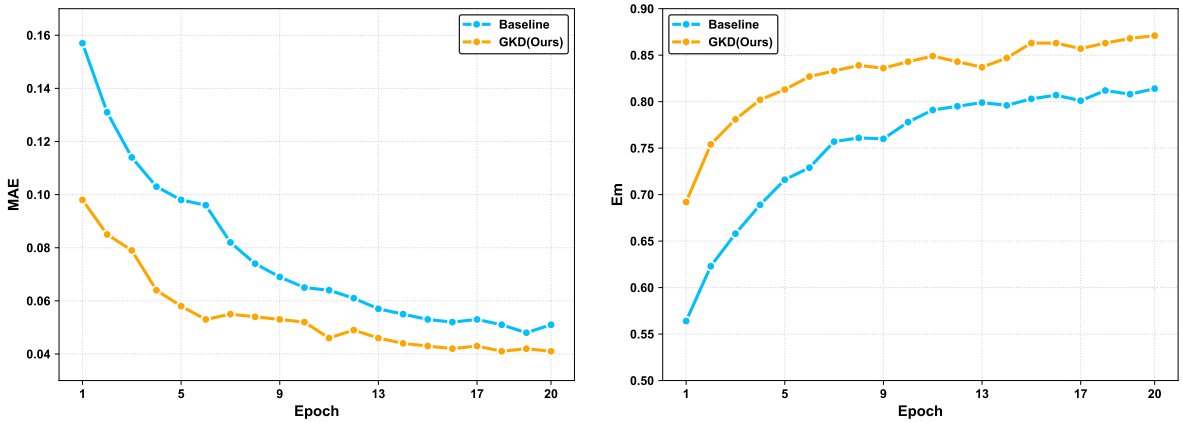

Figure 3. Training efficiency. Performance across different training epochs.

ber is incremented by one. Subsequently, the ETC eliminates masks with relatively large truncation numbers:

$$RR^i = \mathbb{I}(trum(V^i) < n), \tag{1}$$

where $trum(\cdot)$ is the truncation numbers of mask V^i and n is the truncation threshold.

B.1.2. Details of FGC.

Due to the inherent challenges of COD tasks and the imperfect accuracy of input bounding box prompts, SAM tends to generate fragmented responses when processing highly challenging samples, indicating low confidence in its segmentation predictions for these cases, as shown in Figure 2 (b). Therefore, the FGC counts the number of connected components and filters out masks with an excessive number of connected components. Specifically, the FGC employs an optimized two-pass connected-component labeling algorithm [7] to compute the number of connected components, subsequently filtering out masks with higher component counts:

$$FR^i = \mathbb{I}(conc(V^i) < \alpha), \tag{2}$$

where $conc(\cdot)$ is the connected components numbers of mask V^i and α is fragmentation threshold.

The GME conducts normal-quality mask reevaluation and filtration by simultaneously considering two critical factors (truncation number and fragmentation levels):

$$ReFilter(V^i) = \mathbb{I}(RR^i + FR^i = 2), \tag{3}$$

B.2. Implementation Details of Zero-shot Setting

Compared to the unsupervised training setting, the zero-shot setting cannot remove low-quality masks, so we designed a targeted strategy specifically for the zero-shot scenario. Specifically, if all candidate masks of a sample are

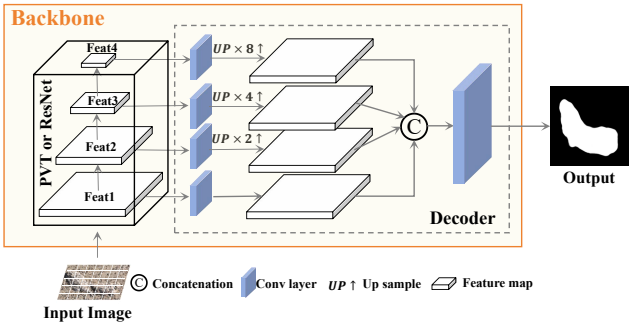

Figure 4. The detailed architecture of the backbone network.

identified as low-quality, we reuse the MLLMs to obtain point labels via CA-CoT, and use these points as prompts for SAM to generate new candidate masks. We then use GME to determine whether any non-low-quality mask exists. If such a mask is found, it is used as the final result. If none exist, the smallest mask by area among all candidate masks is selected as the final result. It is worth noting that although this process requires additional inference using MLLMs and SAM, the number of samples that actually need re-inference is relatively small, with only 863 out of 6397 samples affected. This results in only a 13 percent increase in inference cost.

B.3. Detailed Structure of Backbone Network

The design of the backbone network is not the focus of our work. We use either PVT [6] or ResNet [2] as the encoder for feature extraction. Then, we design a simple segmentation head to produce the final segmentation results from the encoder. As shown in Figure 4, for an input image $I \in \mathbb{R}^{3\times H\times W}$, we put it into the encoder to get the output features $Feat_i$ for the i-th. Then, we get the multi-scale features $(F_{eat1}, F_{eat2}, F_{eat3}, F_{eat4})$ with $(\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32})$ resolution of input images. We downsize the channel dimension of F_{eati} into 64 by using 3×3 convolutional layers. Next, these feature maps are unified into the same size by an up-sampling operation, and combined through the concatenation. Finally the output map $\hat{S} \in \mathbb{R}^{1\times W\times H}$ is obtained by the 3×3 convolution layer.

C. Experiments

C.1. Impact of the Hyperparameter

There are three main hyperparameters in the model's GME module: the high-quality threshold τ_h and passable threshold τ_l in SIM, and the fragmentation threshold α in FGC. Although there is also a boundary truncation threshold n in ETC, it can only take four discrete meaningful values: 1, 2, 3, and 4. Therefore, we do not consider it a hyperparameter. We test diverse values for these parameters, as shown in Table 1. Based on these results, we set τ_h = 0.9, τ_l = 0.6, α = 50, and n = 2. It is worth noting that the value of α has little impact on the results under different settings. Therefore, there are only two key hyperparameters that require tuning.

τ_h	MAE↓	τ_l	MAE↓	α	MAE↓	n	MAE↓
0.95	0.083	0.65	0.082	30	0.084	1	0.083
0.90	**0.081**	**0.60**	**0.081**	40	0.082	**2**	**0.081**
0.85	0.086	0.55	0.084	**50**	**0.081**	3	0.084
0.80	0.095	0.50	0.089	60	0.081	4	0.089

Table 1. The impact of τ_h, τ_l, α and n in the Graded Mask Evaluator.

Method	$\delta = 0.3$	$\delta = 0.5$	$\delta = 0.7$	$\delta = 0.8$	$\delta = 0.9$	Mean
Qwen2.5-VL-3B						
Baseline	77.2	63.9	48.5	37.6	21.1	49.7
Ours	**88.6**	**79.4**	**63.9**	**52.2**	**29.8**	**62.8**
Qwen2.5-VL-7B						
Baseline	84.6	76.8	62.5	50.3	28.9	60.6
Ours	**90.7**	**82.4**	**67.3**	**54.6**	**32.0**	**65.4**
Qwen2.5-VL-72B						
Baseline	90.7	82.9	69.3	57.1	33.2	66.6
Ours	**93.1**	**85.4**	**71.9**	**58.7**	**35.1**	**68.9**

Table 2. Ablation study of CA-CoT across MLLM scale. δ represents IoU thresholds.

Method	$\delta = 0.3$	$\delta = 0.5$	$\delta = 0.7$	$\delta = 0.8$	$\delta = 0.9$	Mean
Baseline	75.3	60.6	45.2	34.8	19.6	47.1
Ours	**86.1**	**73.7**	**57.3**	**47.2**	**27.8**	**58.4**

Table 3. Ablation Study of CA-CoT on Shikra-7B.

Method	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.4$	$\delta = 0.5$	Mean
Baseline	74.1	55.5	39.1	22.9	15.3	41.4
Ours	**87.8**	**70.6**	**48.5**	**28.4**	**19.4**	**50.9**

Table 4. Ablation Study of CA-CoT on LLaVA-1.5-7B.

C.2. Training Efficiency

We visualize the curves of various metrics during training, as shown in Figure 3. The training settings are kept consistent, including the optimizer, learning rate, and training epochs. We observe that GKD not only improves the performance of knowledge distillation but also enhances its efficiency. Compared to training without GKD, using GKD allows the model to achieve the same performance with fewer training epochs and faster convergence. This is because GKD focuses on the most informative knowledge while effectively avoiding the negative impact of inaccurate knowledge during training, enabling more efficient knowledge distillation.

Methods	Backbones	Sup.	CAMO				COD10K				NC4K			
			MAE↓	S_m↑	E_m↑	F_β^w↑	MAE↓	S_m↑	E_m↑	F_β^w↑	MAE↓	S_m↑	E_m↑	F_β^w↑
CVP * [5] MM'24	Qwen2.5-VL$_{3B}$, DINO V2, SAM-HQ	Z	0.096	0.757	0.804	0.689	0.067	0.727	0.795	0.583	0.079	0.774	0.841	0.686
ProMaC [3] NIPS'24	Qwen2.5-VL$_{3B}$, CLIP, SAM, SD V2	Z	**0.092**	**0.786**	**0.852**	**0.704**	0.042	0.808	0.874	0.649	0.069	0.782	0.851	0.723
UCOD-MKD (Ours)	Qwen2.5-VL$_{3B}$, SAM	Z	0.101	0.738	0.786	0.652	**0.036**	**0.830**	**0.882**	**0.755**	**0.059**	**0.823**	**0.866**	**0.766**

Table 5. Comparison of our method with recent state-of-the-art zero-shot approaches. $*$ indicates code that is not publicly released and is reproduced based on the paper.

C.3. Ablation Study of CA-CoT across MLLM Scales

To validate the generalizability of our CA-CoT module, we conduct experiments across MLLMs of different parameter scales, including 72B, 7B, and 3B. As shown in Table 2, CA-CoT consistently improves the performance of MLLMs across all models.

C.4. Ablation Study on Different Types of MLLMs

To demonstrate the generality of our CA-CoT, we additionally conduct experiments on several widely used MLLMs, including LLaVA-1.5-7B [4] and Shikra-7B [1]. As shown in Tables 2, 3, and 4, our method consistently improves the performance of these MLLMs on the COD task.

Moreover, we observe that Qwen2.5-VL and Shikra outperform LLaVA on the COD task. This is understandable, as LLaVA differs in model architecture from other MLLMs and lacks targeted optimization for visual localization. Therefore, for a fair comparison, under the zero-shot setting, we replace the MLLMs used in ProMac and CVP with Qwen2.5-VL-3B. As shown in the Table 5, our method remains superior even when using the same MLLM. It is worth noting that even with Qwen2.5-VL, the performance of CVP and ProMac does not improve significantly. This is because their zero-shot paradigms differ from ours: CVP tends to first obtain a bounding box and then a point prompt to generate a mask, which dilutes Qwen2.5-VL's advantage in box localization; meanwhile, ProMac relies heavily on semantic-based decision-making, whereas LLaVA holds an advantage in aligning visual and textual semantics.

D. Discussion

D.1. Discussion of Unsupervised COD

We construct a teacher model using an MLLM and train a student model with a tailored distillation strategy, achieving strong performance that significantly surpasses previous unsupervised methods. We believe that our approach represents a promising direction for unsupervised COD for two reasons: 1) MLLMs are developing rapidly, with both their performance improving quickly and their parameter sizes decreasing (e.g., the emergence of 2B models), which greatly lowers the barrier for real-world deployment. 2) Although using an MLLM may increase inference cost, employing it as a teacher to guide the training of a lightweight student model substantially accelerates the training process (as shown in Figure 3). This implies that the overall training cost is greatly reduced, making the inference overhead negligible compared with the cost of training. 3) Although current MLLMs still cannot reach the quality of human-annotated labels on challenging tasks such as COD, they can still be selectively leveraged through various strategies. Moreover, with the rapid progress of MLLMs (e.g., Qwen3-VL) and segmentation models such as SAM (e.g., SAM3), the potential upper bound of this pipeline will continue to rise.

D.2. Discussion of Zero-shot COD

Existing zero-shot pipelines vary widely and lack a unified framework, largely because many backbone models are available, including DINOv2, various MLLMs, diffusion models, SAM, and CLIP. Our zero-shot paradigm achieves state-of-the-art performance with the fewest base models—using only an MLLM and SAM. This demonstrates the feasibility of a zero-shot COD pipeline constructed solely from an MLLM and SAM. By focusing on optimizing the MLLM for COD and applying simple post-hoc filtering on SAM-generated masks, strong performance can be obtained without relying on complex pipeline designs.

References

[1] 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. 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[3] Jian Hu, Jiayi Lin, Junchi Yan, and Shaogang Gong. Leveraging hallucinations to reduce manual prompt dependency in promptable segmentation. *Advances in Neural Information Processing Systems*, 37:107171–107197, 2024. 4

[4] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 4

[5] Lv Tang, Peng-Tao Jiang, Zhi-Hao Shen, Hao Zhang, Jin-Wei Chen, and Bo Li. Chain of visual perception: Harnessing multimodal large language models for zero-shot camouflaged ob-

ject detection. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 8805–8814, 2024. 4

[6] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 3

[7] Kesheng Wu, Ekow Otoo, and Kenji Suzuki. Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis and Applications*, 12:117–135, 2009. 2