Appendix

A. Illustration of SAM-based Quasi-superpixel Classification and Seed Generation

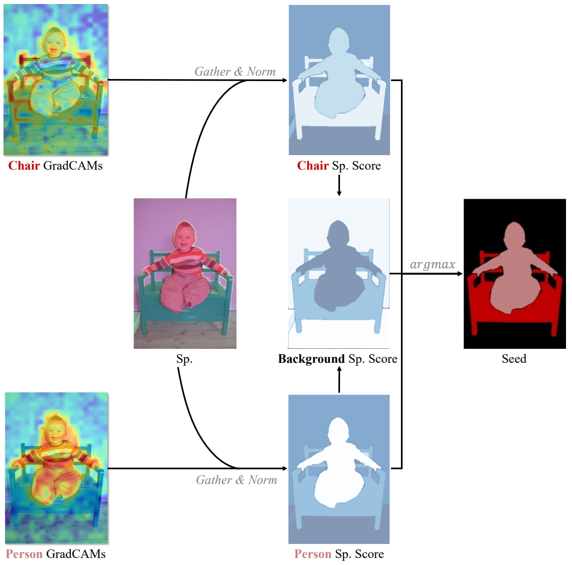

Figure S1. Procedures for SAM-based quasi-superpixel classification and seed map generation, where “Sp.” is the abbreviation of “quasi-superpixel”. On quasi-superpixel score maps, brighter colors indicate higher scores.

We illustrate the computation process of SAM-based quasi-superpixel classification and seed map generation in figure S1, which depicts gathering quasi-superpixel foreground scores from GradCAMs, computing background score and determining the semantic class of each quasi-superpixel and pixel. Please refer to Section 4.2 of the paper for more algorithm details.

B. Framework Efficiency

In Table S1, we compare our time costs and parameter size with some related works. Our method has the fewest learnable parameters except for CLIP-ES [9], which is our training-free baseline. Similarly, our time spent on CAM generation is minimal except for CLIP-ES. Furthermore, our post-processing refinement cost is negligible compared to all other methods, which is a useful feature for online refinement during training. Note that we did not include the time required for SAM-based quasi-superpixel generation, as superpixels only need to be generated once for each training set and can be reused for all subsequent experiments. Compared to CLIP-ES, we only consume a little additional training time and parameter size. However, in return, we achieve a considerable performance boost and eliminate the manual selection of prompt context.

C. More Final Segmentation Results

In this section, we trained additional combinations of segmentation networks and backbones using the fine seeds generated by our framework. We present the results on PASCAL VOC [6] and MS COCO [8] in Tables S2 and S3, which also include the results from Tables 7 and 8 of the paper.

For PASCAL VOC, there is a slight performance difference between DeepLab V2 [1] and DeepLab V3+ [2]. Switching from V3+ to V2 does not affect the conclusions drawn in Section 5.4 of the paper. Mask2Former [3] using a larger pre-train dataset and a heavier backbone leads to a few improvements on PASCAL VOC. Additionally, the experiments conducted on MS COCO demonstrate that Swin-B [11] and Swin-L perform similarly, while pretraining on ImageNet-21K [5] significantly improves the performance.

D. Text-to-Semantic-Mask Usage

SAM [7] attempts to generate masks using CLIP text features as prompts (denoted as text-to-mask). However, its performance is not satisfactory. One solution to make SAM accept text prompts is combining Grounding-DINO [10], which obtains object bounding boxes based on text inputs and then uses the object boxes to prompt SAM and generate instance masks. We noticed that the seed generation networks in WSSS, such as CLIP-ES [9] and ViT-PCM [13], can be seen as text-to-semantic-mask methods for the specific data domain. Such methods require training with data that have image-level labels. During inference, the seed generation network is prompted by text (class label) and obtains semantic segmentation masks. We report the performance of our seed generation framework for text-to-semantic-mask in Table S4. We also compare adopting CLIP-ES [9] for text-to-semantic-mask. As can be seen,

Table S1. Time and learnable parameters size of different methods for seed generation on PASCAL VOC 2012 *trainaug* set with 10,582 images. We report the size of learnable parameters in the "Param. Size" column. Post-processing refinement methods include applying dense CRF (CRF), training affinity networks (RW), using class-aware attention-based affinity (CAA), and utilizing our SAM-based seeding module (SAMS). The time unit is hour and the parameter size unit is MB. For MCTformer, the inference and CRF processes are combined.

Method	CAM Generation		Post-Processing				Total Time	Param. Size
	Train	Inference	CAA	SAMS	CRF	RW		
CLIMS [14]	2.1	0.3	-	-	0.2	6.5	9.1	183M
MCTformer [15]	0.5	2.5	-	-	-	3.0	6.0	121M
CLIP-ES [9]	-	0.4	0.01	-	0.2	-	0.6	-
Ours	1.9	0.4	0.01	0.01	-	-	2.3	0.8M

our method achieves much higher performance at the cost of additional training and inference of SAM.

Compared to combining Grounding-DINO [10] with SAM, our method that combines prompt-learnable CLIP has some limitations, such as not supporting free-form text input, cannot perform instance segmentation, and requiring fine-tuning on specific domains. However, on the other hand, CLIP [12] supports more semantic classes compared to Grounding-DINO. Furthermore, when fine-tuning is required for new classes or specific data distributions, our framework only requires image-level annotations rather than object box annotations. We hope to explore this further in future work.

Table S2. Performance comparison of our method with different final segmentation networks and backbones on PASCAL VOC 2012 *val* sets. We denote segmentation network type at "Seg." column. The ‡ indicates backbone pretrained on ImageNet-21k [5].

Seg.	Backbone	mIoU
DeepLab V2	R101	76.7
DeepLab V3+	R101	77.3
Mask2Former	Swin-B	80.3
Mask2Former	Swin-B‡	81.4
Mask2Former	Swin-L‡	82.6

Table S3. Performance comparison of our method with different final segmentation networks and backbones on MS COCO 2014 *val* set. We denote segmentation network type at "Seg." column. The ‡ indicates backbone pretrained on ImageNet-21k [5].

Seg.	Backbone	mIoU
DeepLab V3+	R101	48.6
Mask2Former	Swin-B	51.8
Mask2Former	Swin-B‡	55.1
Mask2Former	Swin-L‡	55.4

E. More Ablation Studies on Training Loss

The values of most CLIP logits are situated in the saturated range of the sigmoid, leading to inefficient training by binary cross entropy loss. Moreover, we use positive class Softmax-GradCAMs with sigmoid activation to calculate probabilities for pixel-wise cross entropy loss. Note that we only obtain the Softmax-GradCAMs of positive classes, so probabilities can only be derived from sigmoid rather than softmax. However, due to CLIP parameters being frozen, the absolute values of Softmax-GradCAMs are trapped within their initial small range (from 0 to 10^{-3}), and sigmoid probabilities stay near 0.5. This prevents the convergence of cross entropy loss and weakens training robustness.

Table S4. Performance comparison of text-to-semantic-mask usage on PASCAL VOC 2012 *val* sets.

Method	mIoU
CLIP-ES [9]	73.8
Ours	80.6

In this section, we attempted to scale CLIP logits or Softmax-GradCAMs to a reasonable range with a set of linear scaling parameters, which are adjusted manually or learned automatically. The results are shown in Table S5, indicating that both manual and automatic scaling improved performance to some extent. However, our multi-label contrastive and CAM activation losses still achieved the best performance without introducing any additional parameters. In addition, our CAM activation loss still outperforms pixel-wise cross entropy with scaled input by a large margin. This is because the CAM activation loss aligns the supervision signal and seed generation process. During seed generation, Softmax-GradCAM is truncated with 0 and then subjected to min-max normalization. Then, values close to 1 are considered foreground candidates, while values close to 0 represent the background. Similarly, the CAM activation loss expects CAM values in the foreground to be close to the maximum response, i.e., close to 1 after normalization, while background values should be below 0.

Table S5. The performance evaluation of employing different multi-label classification and segmentation loss on PASCAL VOC 2012 *trainaug* set. $\mathcal{L}_{BCE}$ stands for the binary cross entropy loss widely employed by other WSSS methods. $\mathcal{L}_{CE}$ represents the pixel-wise cross entropy loss whose probabilities are obtained by inputting CAM activation values into a sigmoid function. "Manual Scale" applies manually adjusted linear scaling parameters to scale the CLIP logits or Softmax-GradCAMs, and "Auto Scale" scales values with learnable scaling parameters.

$\mathcal{L}_{BCE}$	$\mathcal{L}_{MCL}$	$\mathcal{L}_{CE}$	$\mathcal{L}_{CAL}$	Manual Scale	Auto Scale	mIoU(%)
✓						65.9
✓				✓		70.7
✓					✓	70.1
	✓					71.5
	✓	✓				52.1
	✓	✓		✓		68.8
	✓	✓			✓	64.6
	✓		✓			74.2

F. Refine Full-Supervised Results with SAM-based Seeding Module

Table S6. The effects of refining the **full-supervised** final segmentation results using SAMS or CRF on PASCAL VOC 2012 *val* set.

Backbone	Post-Processing	mIoU(%)
DeepLab V3+	-	79.5
	CRF	$78.4_{-1.1}$
	SAMS	$80.9_{+1.4}$
Mask2Former	-	86.0
	CRF	$84.7_{-1.3}$
	SAMS	$85.5_{-0.5}$

In Table 3 of the paper, attempts are made to refine the segmentation networks trained with seed by dense CRF or our SAM-based seeding module (SAMS). In this section, we conduct the same experiments on fully supervised conditions to demonstrate that the utility of SAMS is not limited to WSSS but can be applied to various semantic segmentation sub-tasks. The experimental results are presented in Table S6. On DeepLab V3+, SAMS achieves the same performance improvement as WSSS, while CRF remains ineffective. However, SAMS does not bring further improvements when employed to the high baseline results obtained by Mask2Former.

G. More Implementation Details

We implement our proposed method in PyTorch. All of our experiments are conducted on a single RTX 3090 GPU with 24GB memory. When training the proposed method for seed generation, we enable the second-order derivative to ensure that the gradients of softmax-GradCAM are propagated correctly. Additionally, we detach the activation weights (Eq. 2 of paper) during softmax-GradCAM computation. For the final segmentation network, we train DeepLab V3+ [2] with a batch size of 16 and Mask2Former [3] of 4. Moreover, 120 epochs are trained on PASCAL VOC and 32 on MS COCO. Other training and inference settings, such as the optimizer, scheduler, learning rate, etc., are set following implementations of MM-Segmentation [4].

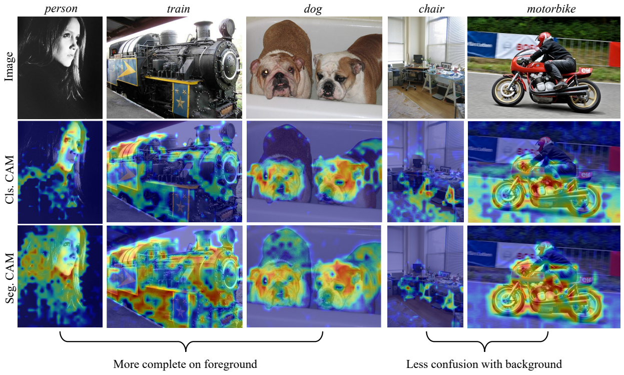

Figure S2. The GradCAMs generated by classification (Cls.) and segmentation (Seg.) tasks. The target class is labeled on the top. Note that CAA are not employed for a clarity comparison.

H. More Qualitative Results

In Figure S2, we provide more results of GradCAMs generated by classification and segmentation tasks. We can observe that GradCAMs of the segmentation task are more complete and less likely to spread to background region, which proves the effectiveness of our coarse-to-fine design.

In Figure S3, we visualize the original SAM masks and our SAM-based quasi-superpixel, together with seeds generated base on them. It can be observed that our SAM-based quasi-superpixel tends to use as few masks as possible to identify the entire instance at once, with minimal overlap between masks. Therefore, seed based on quasi-superpixel is more effective at segmenting complete objects and ensuring consistent and accurate semantics within each instance.

In Figure S4, we visualize the seeds generated from dense CRF and our SAM-based seeding module. We observe that SAMS can generate clearer and more accurate boundaries. When multiple foreground classes overlap and occlude each other, CRF is prone to confusion at the boundary, whereas SAMS avoids such confusion. Finally, for elongated or color-variant objects, CRF often fails to propagate CAM throughout the entire object, while SAMS consistently identifies the complete object.

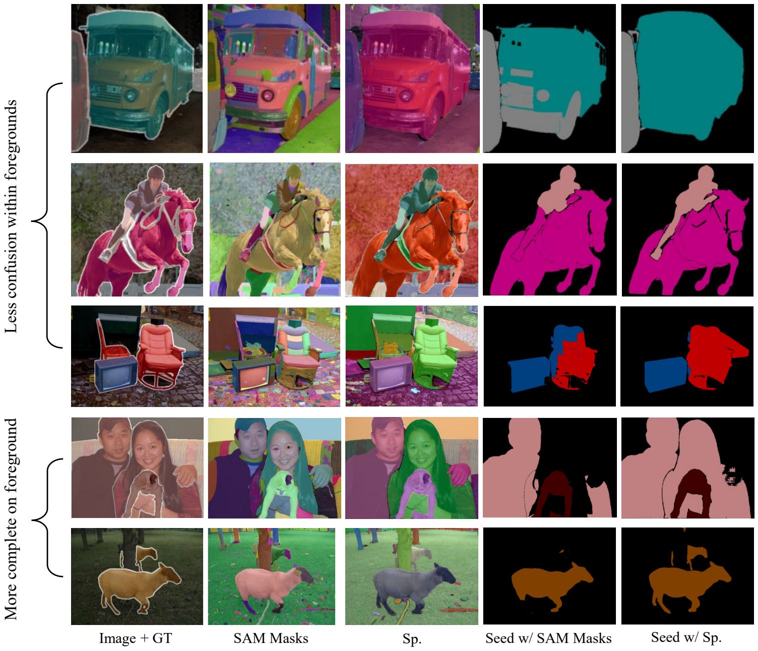
Less confusion within foregrounds
More complete on foreground
Image + GT
SAM Masks
Sp.
Seed w/ SAM Masks
Seed w/ Sp.

Figure S3. The seeds generated from original SAM masks and our SAM-based quasi-superpixel (Sp.).

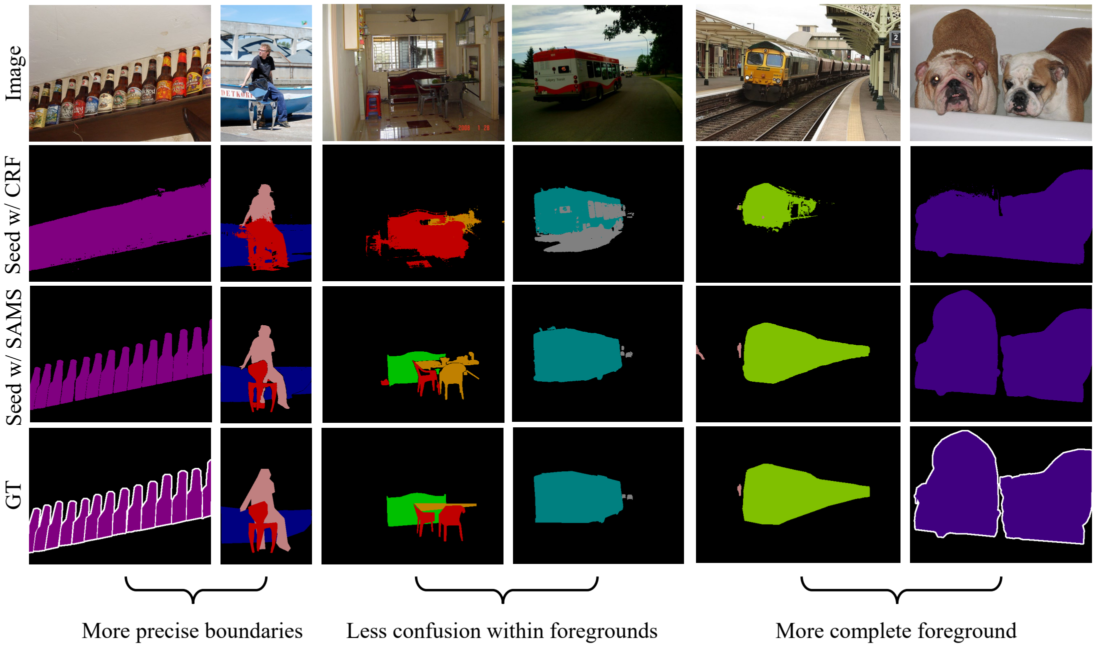

Figure S4. The seeds generated with different post-processing refinement methods, including dense CRF (CRF) and our SAM-based seeding module (SAMS).

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 1–18, 2018. 1, 3

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 1, 3

[4] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1, 2

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[8] 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*, pages 740–755, 2014. 1

[9] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *CVPR*, pages 15305–15314, 2023. 1, 2

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1, 2

[11] 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*, pages 10012–10022, 2021. 1

[12] 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*, pages 8748–8763, 2021. 2

[13] Simone Rossetti, Damiano Zappia, Marta Sanzari, Marco Schaerf, and Fiora Pirri. Max pooling with vision transformers reconciles class and shape in weakly supervised semantic segmentation. In *ECCV*, pages 446–463, 2022. 1

[14] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *CVPR*, pages 4483–4492, 2022. 2

[15] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, pages 4310–4319, 2022. 2