AoP-SAM: Automation of Prompts for Efficient Segmentation

Yi Chen
School of Electrical Engineering
KAIST
Daejeon, South Korea
chenyi@kaist.ac.kr

Mu-young Son
School of Electrical Engineering
KAIST
Daejeon, South Korea
kkt1690@kaist.ac.kr

Chuanbo Hua
School of Industrial & Systems Engineering
KAIST
Daejeon, South Korea
cbhua@kaist.ac.kr

Joo-young Kim
School of Electrical Engineering
KAIST
Daejeon, South Korea
jooyoung1203@kaist.ac.kr

Abstract

The Segment Anything Model (SAM) is a powerful foundation model for image segmentation, showing robust zero-shot generalization through prompt engineering. However, relying on manual prompts is impractical for real-world applications, particularly in scenarios where rapid prompt provision and resource efficiency are crucial. In this paper, we propose the Automation of Prompts for SAM (AoP-SAM), a novel approach that learns to generate essential prompts in optimal locations automatically. AoP-SAM enhances SAM's efficiency and usability by eliminating manual input, making it better suited for real-world tasks. Our approach employs a lightweight yet efficient Prompt Predictor model that detects key entities across images and identifies the optimal regions for placing prompt candidates. This method leverages SAM's image embeddings, preserving its zero-shot generalization capabilities without requiring fine-tuning. Additionally, we introduce a test-time instance-level Adaptive Sampling and Filtering mechanism that generates prompts in a coarse-to-fine manner. This notably enhances both prompt and mask generation efficiency by reducing computational overhead and minimizing redundant mask refinements. Evaluations of three datasets demonstrate that AoP-SAM substantially improves both prompt generation efficiency and mask generation accuracy, making SAM more effective for automated segmentation tasks.

1 Introduction

Image segmentation is essential in computer vision, supporting applications like autonomous vehicle navigation [3] to medical diagnostics [7] and robotics perception [11]. The Segment Anything Model (SAM) is a *foundation model* trained on an extensive dataset containing billions of mask annotations, designed for tackling general image segmentation tasks [13]. SAM excels at segmenting a wide range of visual elements across diverse environments, enabling its *zero-shot* generalization ability through promptable features. Utilizing prompts, such as points and bounding boxes, SAM demonstrates remarkable adaptability across various applications [13].

The manual provision of prompts required for segmenting entire images in SAM is highly labor-intensive and time-consuming, making it impractical for applications requiring rapid prompt provision in hardware-constrained scenarios, such as industrial automation. Consequently, automatic prompt provision is essential, but current methods face two major challenges: 1) Unintelligent automation: In Automatic Mask Generation (AMG) mode, SAM's grid-based prompt generation can either miss small objects when using sparse grids, leading to reduced accuracy, or generate redundant masks with dense grids, resulting in inefficient mask generation that requires extensive refinement and slows

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

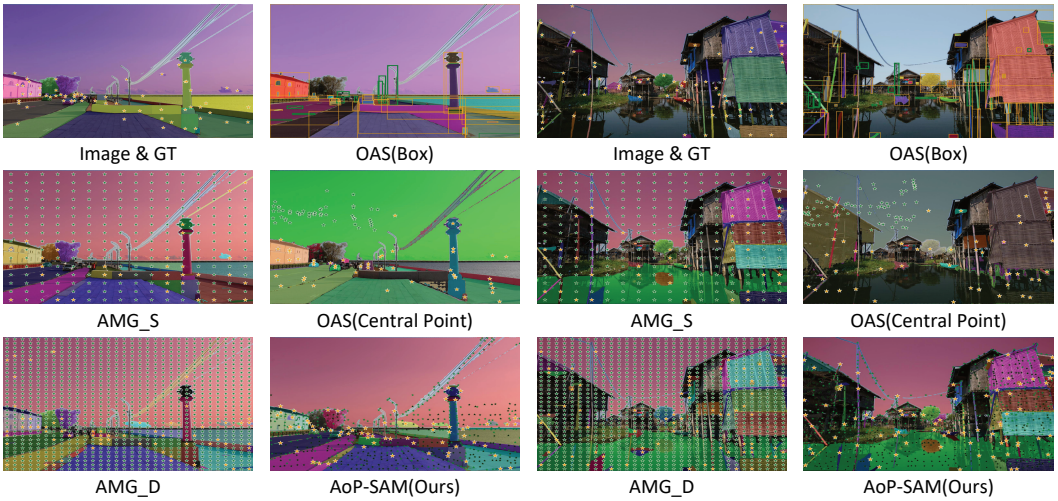

Figure 1: In SAM, automating prompt provision eliminates the need for manual input, significantly improving the efficiency of mask segmentation. However, current approaches, such as grid-based prompt generation in vanilla SAM or methods that rely on extra object detection models, often introduce excessive mask refinements or computational overhead, leading to increased latency and reduced efficiency. In contrast, our proposed AoP-SAM efficiently generates essential prompts for accurate mask generation within SAM, entirely without human intervention. In the illustrations above, different colors represent various segmentation results, with orange labels (stars or boxes) indicating valid prompts, green labels marking invalid prompts, and black stars in our results representing the filtered prompts, processed in a coarse-to-fine manner by the test-time instance-wise ASF mechanism.

down overall performance. 2) Time and Resource Inefficiency: Object detection models used for generating object-aware prompts [23] introduce considerable computational overhead and latency due to the large sizes of deep-learning models employed. These two main limitations hinder the broader applicability of SAM as a foundational segmentation model.

In this work, we introduce the Automation of Prompts for SAM (AoP-SAM), a novel approach for automating prompt generation that is widely applicable across the SAM family of models, e.g. [13, 22, 23]. This method enables the efficient generation of essential prompts for precise segmentation, eliminating the need for human intervention. Our approach incorporates a learnable Prompt Predictor that seamlessly integrates with SAM, utilizing the image embeddings produced by SAM's image encoder to generate a Prompt Confidence Map (PCM), which identifies regions with potential candidates. Additionally, we propose a test-time instance-wise Adaptive Sampling and Filtering (ASF) mechanism that initially samples prompts coarsely and then finely filters out redundant ones based on the generated mask, significantly enhancing overall efficiency. Therefore, AoP-SAM improves the efficiency of segmentation without limiting SAM's accuracy or flexibility.

2 Related work

2.1 Methods for automating prompts

Manually setting prompts in SAM for segmentation is labor-intensive, time-consuming, and often requires domain-specific expertise [14], making it impractical for real-world applications. To alleviate this, SAM's AMG mode automates prompt provision by placing prompts in a grid to create masks [13]. However, sparse grids may miss small objects and result in low accuracy performance, while denser grids, like 32×32 points, lead to redundant masks and slow down processing due to the need for extensive mask refinement. A specialized version of SAM has been developed to create the large SA-1B dataset, but it sacrifices inference speed for accuracy, further hindering efficiency [13].

An alternative approach involves using bounding boxes as prompts for SAM, facilitated by existing object detection models. However, methods like Object-Aware Sampling (OAS) [23] employ YOLOv8 [20] introduce considerable computational overhead as these models are not designed or optimized for this purpose. This leads to inefficiencies, particularly in resource-constrained environments, diminishing SAM's overall effectiveness. In contrast, AoP-SAM is designed for allocating optimal locations of essential prompts, improving overall efficiency and segmentation accuracy.

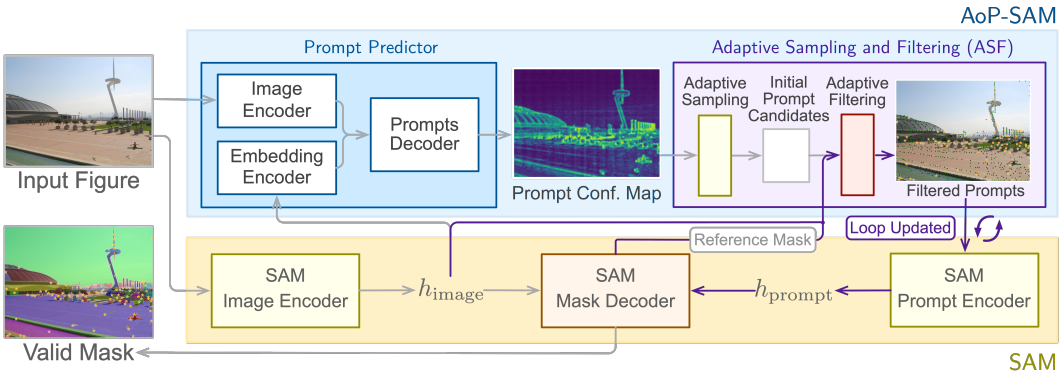

Figure 2: The architecture of our proposed AoP-SAM consists of two key components: the prompt predictor and ASF Module. The prompt predictor operates by taking the image input and the computed image embedding from SAM's image encoder as inputs. Prompt predictor then generates a PCM that highlights potential regions for prompt candidates. During test-time, these candidates are adaptively sampled and filtered by ASF, predicting prompts that might lead to redundant masks based on the generated mask references. This process eliminates unnecessary prompts, ensuring that only the essential ones are used to generate the final mask resultes.

2.2 Test-time adaptation

Test-time domain adaptation improves model performance on test data with a domain gap using either backward-based or backward-free approaches [19, 10]. Backward-based methods, like entropy minimization, capture target domain features [19, 9], while backward-free methods, such as DUA's running average and DIGA's distribution adaptation [16, 21], adjust batch normalization statistics. Recently, test-time adaptation has been applied to camouflage object segmentation in SAM [8]. AoP-SAM follows this strategy and introduces an instance-level test-time ASF mechanism that eliminates redundant prompt candidates and improves efficiency for generalized segmentation across datasets, without requiring sample-level supervision.

3 Method

3.1 Prompt predictor for prompt confidence map

To streamline the process of prompt provision and reduce computational complexity, we introduce a lightweight prompt predictor efficiently integrated with SAM. As shown in Figure 2, the image segmentation process begins with SAM computing image embeddings. When new prompts are provided, their prompt embeddings are derived and injected into the mask decoder, along with the image embedding, to generate the corresponding masks. Following SAM's methodologies, AoP-SAM starts to generate and feed prompts into the prompt encoder after image embeddings are computed, allowing the prompt predictor to reuse this data and also image inputs. The predictor then creates a PCM, which highlights high-confidence regions to identify optimal locations of essential prompts for precise and effective segmentation within SAM.

Prompt predictor architecture. The model incorporates two CNN-based encoders: one for processing the original image and another for handling the ViT embedding. The image encoder consists of three convolutional layers with ReLU activations to extract spatial features and introduce non-linearity. The ViT embedding is reshaped to match the image's spatial dimensions and passed through its own encoder to produce a 32-channel feature map. This alignment enables the model to fuse spatial and contextual information from both the image and ViT embedding, improving segmentation accuracy. As illustrated in Figure 2, this resulting fused feature map is then processed through convolutional layers in the prompt decoder, reducing its dimensionality while refining the representation. The final step uses a sigmoid-activated layer to generate PCMs, highlighting regions containing essential prompts. By normalizing the output to a $[0, 1]$ range, the sigmoid activation enables the selection of high-confidence prompt candidate regions for segmentation within the SAM framework.

Training of prompt predictor Unlike traditional object detection models trained on datasets like COCO [15], our approach for AoP-SAM is data-efficient and aligned with SAM by utilizing the SA-1B dataset, which contains over 1 billion masks and corresponding prompts [13]. This ensures

AoP-SAM inherits SAM's robustness and generalization ability, particularly in handling diverse and unseen data. We use point prompts from SA-1B as ground truth, which align with the PCM generated by our prompt predictor. A curated training set from SA-1B covers various semantic classes and scenarios. The image encoder, embedding encoder, and prompt decoder are trained with PCM ground truth refined by uniform and gaussian kernels for precision. Using MSELoss and adam optimization, the model was trained for 1000 epochs with gradient accumulation to support larger batch sizes.

3.2 ASF for essential prompts

To efficiently extract essential prompts from the PCM, a gaussian filter is applied to smooth the PCM, enhancing key regions while minimizing noise. Local maxima are then coarsely sampled as potential prompt candidates and mapped back to the original image coordinates, ensuring they represent significant features for prompt generation. However, this approach may still produce redundant prompts representing the same entity. To address this, we apply a finer filtering process, using the generated masks as references to identify and remove redundant prompts. A Prompt Elimination Map (PEM) is then constructed by comparing the spatial and semantic information of the generated masks. Each pixel in the map is assigned an elimination score, indicating the likelihood of redundant prompt generation at that location.

We utilize the image feature from image encoder and generated reference masks as inputs for adaptive filtering. The normalized image feature, denoted as $F_{\text{norm}} \in \mathbb{R}^{h \times w \times c}$, contains the original image information, where h and w are the dimensions and c is the feature dimension. The normalized reference masks $M_{\text{norm}} \in \mathbb{R}^{h \times w}$, with n predicted masks, are downsampled and performed a cosine similarity with F_{norm} to obtain n PEMs. On top of this, we adopt another average pooling to aggregate all n local maps to obtain the overall PEM of the generated mask as

$$\text{PEM} = \frac{1}{n} \sum_{i=1}^{n} \left(F_{\text{norm}} \times M_{\text{norm}}^{i} \right), F_{\text{norm}} \in \mathbb{R}^{h \times w \times c}, M_{\text{norm}} \in \mathbb{R}^{n \times h \times w}$$

Moreover, an elimination threshold is calculated based on the IoU scores of the generated masks and their corresponding elimination scores. Prompts that exceed this threshold are considered redundant and removed from the prompt candidates. This adaptive filtering ensures only essential prompts are retained for further mask generation, improving overall efficiency while minimizing refinements.

4 Experiments

Table 1: Results on image segmentation with bounding box and point supervision. Best are in **bold**.

Automating	SA-1B				COCO				LVIS			
Prompts Methods	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P
MobileSAM Image Encoder												
AMG-S [13]	29.8	-	4.5	38.6	56.0	-	**1.9**	33.5	56.2	-	**1.9**	33.4
AMG-D [13]	46.9	-	9.1	71.0	60.9	-	**1.9**	55.9	61.1	-	**1.9**	55.5
OAS(Box) [23]	50.7	0.191	7.3	100	55.5	0.187	4.2	44	55.7	0.188	4.0	38
OAS(Central Point) [23]	48.7	0.188	7.7	141.0	53.9	0.167	4.3	69.0	54.5	0.164	4.3	68.1
AoP-SAM	**51.4**	**0.101**	**4.1**	71.7	**61.5**	**0.096**	2.1	58.1	**62.3**	**0.094**	2.1	57.5
ViT-L Image Encoder												
AMG-S [13]	40.0	-	5.7	55.5	61.4	-	4.4	48.8	63.2	-	**4.3**	49.5
AMG-D [13]	65.6	-	10.3	108.9	67.7	-	**4.3**	86.0	69.2	-	**4.3**	86.5
OAS(Box) [23]	65.8	0.150	9.1	100	63.3	0.152	5.4	44	62.9	0.151	5.3	38
OAS(Central Point) [23]	67.6	0.149	9.7	199.3	64.2	0.133	5.5	98.4	63.5	0.132	5.5	98.9
AoP-SAM	**71.1**	**0.120**	**5.4**	118.3	**68.4**	**0.116**	4.4	97.0	**69.8**	**0.117**	4.4	97.2
ViT-H Image Encoder												
AMG-S [13]	40.8	-	7.1	56.3	63.3	-	5.7	49.8	64.9	-	5.6	50.5
AMG-D [13]	66.8	-	11.8	109.6	69.5	-	5.7	87.4	71.0	-	5.6	88.0
OAS(Box) [23]	66.9	0.160	10.4	100	64.1	0.152	6.8	44	63.3	0.153	6.6	38
OAS(Central Point) [23]	68.3	0.154	11.1	207.6	65.1	0.134	6.9	102.1	63.0	0.134	6.8	102.4
AoP-SAM	**70.6**	**0.122**	**6.6**	107.8	**70.1**	**0.120**	**5.5**	90.0	**71.9**	**0.122**	**5.5**	89.7

Table 2: Ablation study of variants with our AoP-SAM on image segmentation.

Method's variants	SA-1B				COCO				LVIS			
	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P	mIoU ↑	Inf_{Lat} ↓	Peak_{Mem} ↓	#P
AoP-SAM w/o AS, AF	57.2	0.059	7.2	106.4	67.9	0.078	5.7	70.4	60.9	0.075	5.7	60.6
AoP-SAM w/o AF	72.8	0.130	10.1	120.1	70.5	0.122	5.7	97.9	71.7	0.121	5.7	97.5
AoP-SAM	71.3	0.122	6.6	107.8	70.1	0.112	5.7	91.1	71.9	0.122	5.7	89.7

4.1 Experiments setup

In our experiments, we evaluate prompt provision for SAM using three datasets: SA-1B, COCO, and LVIS [13, 15, 4]. We compare two types of supervision—bounding box and point prompts—and

Table 3: Ablation study on hyper-parameters employed in AoP-SAM. Best are in **bold**

(a) Sampling Smoothing Factor

Factor	mIoU ↑	$\mathrm{Inf}_{\mathrm{Lat}}$ ↓	$\mathrm{Peak}_{\mathrm{Mem}}$ ↓	#P
1	**72.4**	0.124	51.5	114.4
2	70.4	0.122	42.2	107.8
3	67.3	**0.118**	32.7	100.3
4	63.4	0.122	**24.6**	91.2

(b) Confidence Intensity Threshold

Thr.	mIoU ↑	$\mathrm{Inf}_{\mathrm{Lat}}$ ↓	$\mathrm{Peak}_{\mathrm{Mem}}$ ↓	#P
0.1	**70.9**	0.121	10.1	109.2
0.2	70.4	0.122	9.75	107.8
0.3	68.7	**0.116**	**9.52**	103.9
0.4	66.4	0.117	9.60	99.5

(c) Prompt Spacing Factor

Factor	mIoU ↑	$\mathrm{Inf}_{\mathrm{Lat}}$ ↓	$\mathrm{Peak}_{\mathrm{Mem}}$ ↓	#P
4	**72.7**	0.123	10.0	114.8
5	71.6	0.123	9.88	111.4
6	70.4	0.122	**9.75**	107.8
7	68.9	**0.117**	9.82	104.2

(d) Prompt Elimination Threshold

Thr.	mIoU ↑	$\mathrm{Mask}_{\mathrm{Lat}}$ ↓	$\mathrm{Ratio}_{\mathrm{elim.}}$ ↑	#P
1.25	68.4	**0.671**	**51.5**	100.9
1.3	70.4	0.799	42.3	107.8
1.35	71.6	0.930	32.7	113.2
1.4	**72.2**	1.041	24.6	116.4

baseline methods, including AMG-S (Sparse grid), AMG-D (Dense grid), and OAS [23], which employs YOLOv8 for generating bounding boxes or using central points of boxes as prompts. For evaluation, we use the greedy IoU algorithm [13, 24] to calculate the mean IoU (mIoU) and measure efficiency through Prompt Inference Latency ($\mathrm{Inf}_{\mathrm{Lat}}$) and Peak Memory ($\mathrm{Peak}_{\mathrm{Mem}}$). Experiments are conducted on a Nvidia Titan RTX GPU, using PyTorch, with point prompts allowing multiple masks and box prompts restricted to single masks. For more details, see Appendix Section 6.3.

4.2 Experiment results and analysis

Performance and efficiency analysis. Table 1 compares various automated prompt methods across models from SAM family with different image encoders on three datasets. AoP-SAM consistently achieves the highest mIoU scores across all datasets and encoders, outperforming methods that use bounding box prompts, which typically provide more spatial information. This highlights AoP-SAM's superior ability to leverage prompts for accurate segmentation, surpassing both traditional methods and those relying on advanced object detection models. In addition to its accuracy, AoP-SAM also demonstrates competitive latency and memory efficiency. For instance, on the SA-1B dataset with the ViT-H encoder, AoP-SAM records a latency of 0.122s and peak memory usage of 6.6MB, both within acceptable limits while delivering top-tier segmentation performance. Overall, the OAS methods generally outperform the baseline AMG-S and AMG-D methods but still fall short of AoP-SAM. This suggests that while OAS enhances performance, the adaptive sampling and filtering techniques in AoP-SAM further improve both segmentation accuracy and prompt generation efficiency.

Component analysis We further analyze the impact of key components, including the prompt predictor, Adaptive Sampling (AS), and Adaptive Filtering (AF), in Table 2 across the same datasets. Enabling AS without AF results in a significant improvement in mIoU compared to using only the prompt predictor. However, the most balanced performance is achieved when both AS and AF are combined, emphasizing the importance of removing redundant prompts to enhance overall segmentation efficiency. This trade-off is important to consider in scenarios where processing speed or hardware resource efficiency is a key priority.

4.3 Hyperparameters analysis

Sampling Smoothing Factor In Table 3(a), we apply gaussian filtering to the heatmap using the sampling smoothing factor. A larger smoothing factor allows the model to cover a broader area, providing stronger smoothing, which helps reduce memory usage during preparation and processing.

From heatmap to point prompt In Table 3(b-c), we explore various parameter settings for converting the confidence map into optimized point prompts. By adjusting the confidence intensity threshold and prompt spacing factor, we aim to identify the most representative points for critical areas in the PCM. These adjustments refine the sensitivity of the point selection process, ensuring that the resulting point prompts are both accurate and reliable.

Prompt Elimination Threshold In Table 3(d), we assess the impact of the prompt elimination threshold on the prompt removal ratio. As the threshold decreases, the Ratio of Elimination ($\mathrm{Ratio}_{\mathrm{elim}}$) increases, leading to faster mask generation, though it may slightly affect accuracy.

5 Conclusion

We propose AoP-SAM, a novel approach designed to efficiently generate essential prompts for accurate mask generation in SAM. Our method features a lightweight Prompt Predictor, trained to predict optimal prompt locations, and a test-time ASF mechanism for automatic prompt generation. Evaluated on three segmentation datasets with three SAM family models, AoP-SAM improves both accuracy and efficiency, making it ideal for automated prompt-based segmentation tasks with SAM.

Acknowledgments

The authors would like to express their sincere gratitude to Adiwena Putra for comments on an earlier draft. This work was supported by the Graduate School of AI Semiconductor, KAIST, South Korea.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[4] 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.

[5] Dongsheng Han, Chaoning Zhang, Yu Qiao, Maryam Qamar, Yuna Jung, SeungKyu Lee, Sung-Ho Bae, and Choong Seon Hong. Segment anything model (sam) meets glass: Mirror and transparent objects cannot be easily detected. *arXiv preprint arXiv:2305.00278*, 2023.

[6] Paul Henderson and Vittorio Ferrari. End-to-end training of object class detectors for mean average precision. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 198–213. Springer, 2017.

[7] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32:582–596, 2019.

[8] Jian Hu, Jiayi Lin, Shaogang Gong, and Weitong Cai. Relax image-specific prompt requirement in sam: A single generic prompt for segmenting camouflaged objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12511–12518, 2024.

[9] Jian Hu, Hongya Tuo, Chao Wang, Lingfeng Qiao, Haowen Zhong, and Zhongliang Jing. Multi-weight partial domain adaptation. In *BMVC*, page 5, 2019.

[10] Jian Hu, Hongya Tuo, Chao Wang, Lingfeng Qiao, Haowen Zhong, Junchi Yan, Zhongliang Jing, and Henry Leung. Discriminative partial domain adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 632–648. Springer, 2020.

[11] Juana Valeria Hurtado and Abhinav Valada. Semantic scene segmentation for robotics. In *Deep learning for robot perception and cognition*, pages 279–311. Elsevier, 2022.

[12] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

[13] 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:2304.02643*, 2023.

[14] Chengyin Li, Prashant Khanduri, Yao Qiang, Rafi Ibn Sultan, Indrin Chetty, and Dongxiao Zhu. Autoprosam: Automated prompting sam for 3d multi-organ segmentation, 2024.

[15] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[16] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14765–14775, 2022.

[17] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[18] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for mulit-class object recognition and segmentation. In *European Conference on Computer Vision (ECCV)*, January 2006.

[19] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[20] Gang Wang, Yanfei Chen, Pei An, Hanyu Hong, Jinghu Hu, and Tiange Huang. Uav-yolov8: A small-object-detection model based on improved yolov8 for uav aerial photography scenarios. *Sensors*, 23(16):7190, 2023.

[21] Wei Wang, Zhun Zhong, Weijie Wang, Xi Chen, Charles Ling, Boyu Wang, and Nicu Sebe. Dynamically instance-guided adaptation: A backward-free approach for test-time domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24090–24099, 2023.

[22] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[23] Chaoning Zhang, Dongshen Han, Sheng Zheng, Jinwoo Choi, Tae-Ho Kim, and Choong Seon Hong. Mobilesamv2: Faster segment anything to everything. *arXiv preprint arXiv:2312.09579*, 2023.

[24] Chaoning Zhang, Fachrina Dewi Puspitasari, Sheng Zheng, Chenghao Li, Yu Qiao, Taegoo Kang, Xinru Shan, Chenshuang Zhang, Caiyan Qin, Francois Rameau, et al. A survey on segment anything model (sam): Vision foundation model meets prompt engineering. *arXiv preprint arXiv:2306.06211*, 2023.

6 Appendix

6.1 Prompting technique in zero-shot foundation models

Foundation models originated in NLP with large language models like GPT, which excel at zero-shot generalization to unseen tasks [1]. Instead of fine-tuning, prompt-based learning helps these models adapt to downstream tasks by interpreting prompts as task instructions, which improves transfer learning. The leading hypothesis regarding the effectiveness of prompts suggests that models interpret these prompts as specific task instructions, enabling them to generalize to tasks not encountered during training [17]. These demonstrations are examples provided to the model to guide its understanding of a new task, and this learning ability is crucial as it underscores the models' flexibility in adapting to new information, mirroring human learning processes and soon prompting has been wildly used in NLP [2]. The success of prompt engineering in NLP has influenced computer vision, leading to the development of SAM, a model that leverages spatial prompts for precise *zero-shot* segmentation.

6.2 Introduction of SAM and its architecture

SAM is a versatile image segmentation framework consisting of three key components: (1) Image Encoder, which extracts essential features from the input image, producing a 64×64 spatial resolution embedding that represents the image's critical characteristics; (2) Prompt Encoder, which processes interactive inputs (points, boxes, masks), converting them into embeddings that guide the segmentation process; and (3) Mask Decoder, a two-layer transformer-based module that combines image and prompt embeddings to generate precise segmentation masks. SAM's design optimizes efficiency by embedding image and prompt inputs once and allowing prompt embeddings to be processed in batches, generating multiple masks without overwhelming system resources. SAM is trained on the massive SA-1B dataset, which includes over 1 billion masks and 11 million images, enabling exceptional zero-shot generalization. However, training SAM is computationally intensive, requiring significant GPU resources. This high computational cost motivates the reuse of the image encoder's outputs in later computations to maximize efficiency. For more details, refer to [13].

6.3 Experiments setup

Datasets. Generalized image segmentation focuses on segmenting every meaningful entity in an image. In this study, we use three key datasets: SA-1B, COCO, and LVIS. The SA-1B dataset, used for training SAM, contains over 1 million images and 1 billion masks [13]. The COCO dataset includes 41,000 images and 200,000 masks, covering a wide range of common objects [15]. LVIS, designed for long-tail distributions, provides 5,000 images and 25,000 masks, emphasizing fine-grained categories [4]. These datasets allow us to thoroughly evaluate the effectiveness of our Automating Prompts method across diverse and challenging scenarios.

Baseline. In our comparison of current methods for automating prompts in SAM, we introduce and evaluate two types of prompts: bounding box prompts and point prompts. The methods AMG-S and AMG-D represent the vanilla grid search with 16×16 and 32×32 prompts, respectively, as utilized in SAM [13]. We also examine the Object-Aware Sampling (OAS) method, which employs YOLOv8 to generate bounding box prompts [23]. Furthermore, we implement an additional method that uses the central point of the bounding box generated by OAS as point prompts. Note that AoP-SAM is trained on a subset dataset of SA_1B and tested on a separate test set, similarly all the comparative methods we employ are also trained and tested on different sets.

Based on previous studies [13, 24], evaluating the accuracy of SAM, which produces masks without predefined labels, presents challenges because traditional metrics for semantic segmentation, (mean Intersection over Union with labels, mIoU) [18, 5], instance segmentation (mean Average Precision, mAP) [15, 6], and panoptic segmentation (Panoptic Quality, PQ) [12] are not directly applicable. To address this, we adopt the greedy Intersection over Union (IoU) algorithm [24], which matches each generated SAM mask with the closest ground truth mask based on the highest IoU scores, then calculates the mean IoU (mIoU) across all matches. In addition to evaluating accuracy performance, we also assess the efficiency of Methods of Automating Prompts in time- or resource-constrained environments using Inference Latency ($\mathrm{Inf}_{\mathrm{Lat}}$) for producing prompts and peak memory ($\mathrm{Peak}_{\mathrm{Mem}}$) consumption during mask generation as key metrics. Additionally, we count the number of essential prompts ($\#\mathrm{P}$) as a reference point for comparing methods. It is important to note that a higher value

of mIoU, or lower values of $\mathrm{Inf}_{\mathrm{Lat}}$. and $\mathrm{Peak}_{\mathrm{Mem}}$., indicate higher efficiency. Although there is no clear preference for the number of essential prompts, intuitively, a smaller number of prompts yielding high accuracy performance is considered advantageous.

Implementation Details Following the previous prompting settings [13], we enable the option for generating multiple mask outputs from a single prompt for point prompts, while disabling it for box prompts [23]. No background prompts are provided in either case. We also implemented quality checks for all methods, removing low-quality masks (e.g., those with low confidence or stability scores) during performance evaluation.

For coarsely sampling point prompts from the PCM, we first apply a Smoothing Factor=2, a confidence intensity threshold=0.2, and a prompt spacing factor=2. In each iteration, the output mask from the previous iteration serves as a reference to generate a Prompt Elimination Map via the ASF, adaptively filtering out selected prompt candidates during test-time to prevent redundant mask generation in future iterations. The experiments are conducted using the PyTorch framework on a single Nvidia Titan RTX GPU.

6.4 Limitation

Due to the use of point-type prompts, it is not possible to process all prompts in a single batch. As a result, AoP-SAM requires iterative processing, which results in need of refinement and impacts the overall end-to-end segmentation time. In future work, it is crucial to explore methods for removing or minimizing these iterations, enabling a more efficient approach that can significantly enhance the end-to-end speed and overall system performance without compromising segmentation quality.