AoP-SAM: Automation of Prompts for Efficient Segmentation

Yi Chen, Muyoung Son, Chuanbo Hua, Joo-Young Kim

KAIST, Korea Advanced Institute of Science and Technology, Daejeon, 34141, South Korea
{chenyi, kkt1690, cbhua, 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.

Introduction

Image segmentation, a critical task in CV, underpins applications ranging from autonomous vehicle navigation (Feng et al. 2020) to medical diagnostics (Hesamian et al. 2019) and robotics perception (Hurtado and Valada 2022). Segment Anything Model (SAM) is a *foundation model* designed to tackle general image segmentation and has been trained on a vast dataset with billions of mask annotations (Kirillov et al. 2023). SAM excels at segmenting a wide range of visual elements across diverse environments, enabling it to solve various downstream segmentation problems through prompt engineering. These prompts, which include points or bounding boxes, allow SAM to achieve *zero-shot* generalization (Kirillov et al. 2023), making it adaptable to numerous applications.

The manual provision of prompts required for segmenting entire images in SAM is highly labor-intensive and time-consuming, making it impractical for applications that demand rapid prompt generation in hardware-constrained scenarios, such as industrial automation. Consequently, automatic prompt generation is essential for these use cases. However, as shown in Figure 1, current approaches to automating prompts for generalized tasks face significant limitations due to two key issues: 1) Unintelligent automation, the vanilla SAM employs a grid-based search of point prompts, named Automatic Mask Generation (AMG) (Kirillov et al. 2023), for producing prompts. If the grid search is too sparse, it risks missing numerous small objects or important details. Conversely, if the search is too dense, it produces an excessive number of redundant masks, necessitating significant refinement and ultimately slowing down the overall processing time. 2) Time and Resource Inefficiency, automating prompts with bounding boxes in SAM enables the use of existing deep-learning models to generate bounding boxes from images, offering an alternative for prompt generation. For instance, the Object-Aware Sampling (OAS) method from (Zhang et al. 2023c) utilizes YOLOv8 (Wang et al. 2023a), a state-of-the-art architecture known for efficient detection with bounding boxes, to automate the prompt production process. However, this approach is not directly aligned with SAM and introduces substantial computational overhead, posing challenges in resource-limited scenarios. These constraints significantly diminish the applicability and effectiveness of foundational segmentation models like SAM, particularly in automated annotation tasks and situations where rapid prompt generation is crucial.

In this work, we propose a novel approach, AoP-SAM, for the Automation of Prompts within the SAM family of models, e.g. (Kirillov et al. 2023; Zhang et al. 2023a,c). This method enables the efficient generation of essential point prompts for accurate segmentation without the need for human intervention. We first designed a learnable prompt predictor specifically for SAM. Unlike independent deep learning modules that rely solely on image input to generate bounding boxes as prompt inputs, our predictor is tightly integrated with SAM. It takes both the image input and the computed image embedding—i.e., the input and output of SAM's image encoder—leveraging this information to learn and generate a prompt confidence map. This map

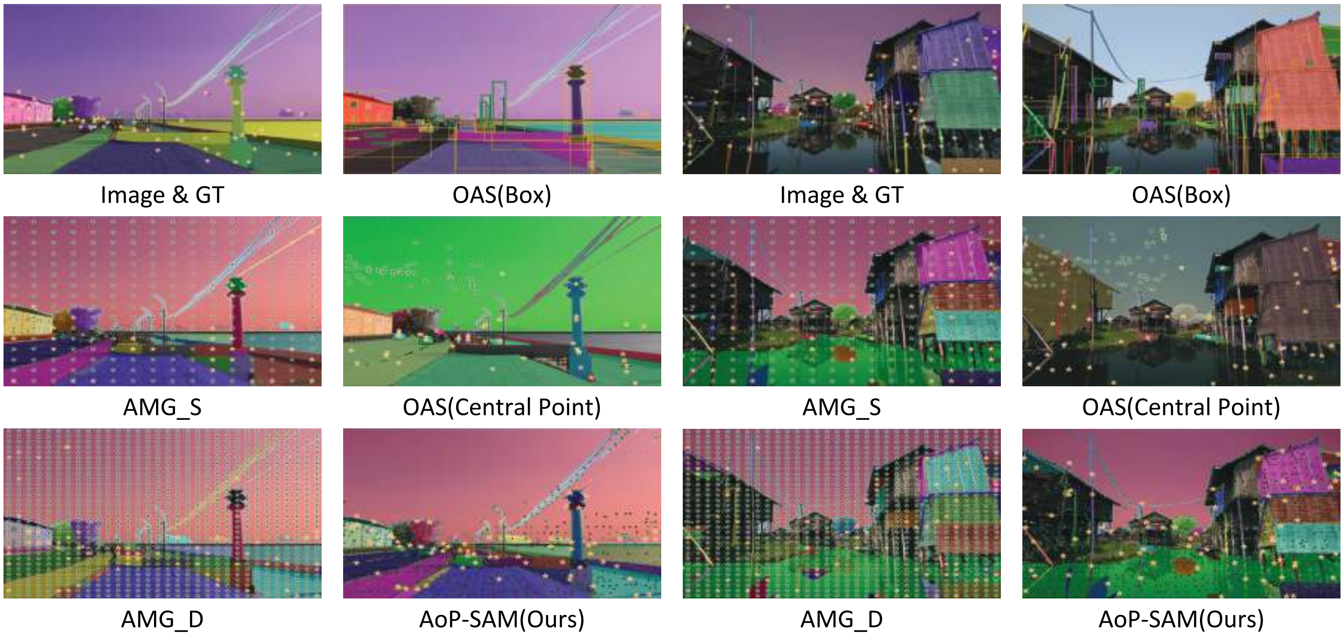

Image & GT
OAS(Box)
Image & GT
OAS(Box)
AMG_S
OAS(Central Point)
AMG_S
OAS(Central Point)
AMG_D
AoP-SAM(Ours)
AMG_D
AoP-SAM(Ours)

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 prompts in vanilla SAM (AMG_S for sparse, AMG_D for dense) or extra detection models (OAS: Box or Central Point), 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 Adaptive Sampling and Filtering (ASF) mechanism.

predicts the locations of essential prompts that can be used for accurate segmentation within SAM. Second, to reduce the number of redundant mask generations, we propose an Adaptive Sampling and Filtering (ASF) technique that operates in a coarse-to-fine manner. Initially, we sample point prompts coarsely from the Prompt Confidence Map to begin mask generation. Then, we leverage the generated mask to filter out any remaining prompts that would produce the same mask, thereby enhancing overall efficiency. Therefore, AoP-SAM can effectively and efficiently produce prompts for segmentation tasks without compromising the accuracy and flexibility of the original SAM.

Our contributions are as follows:

- To eliminate the need for manual provision of prompts tailored to each general image, our AoP-SAM approach automatically predicts and generates prompts effectively and efficiently.
- We introduce a simple yet efficient prompt prediction method that utilizes SAM's computed data to pinpoint potential prompt locations. Additionally, ASF ensures that only the most essential prompts are utilized in a coarse-to-fine selection process, thereby enhancing overall segmentation efficiency.
- Extensive experiments on three benchmarks have shown the effectiveness of our proposed AoP-SAM.

Related Work

Prompting technique in zero-shot foundation models

Foundation models initially emerged in NLP, with large language models like the GPT series demonstrating strong zero-shot generalization to unseen tasks and data. Prompt-based learning methods were then introduced, enabling these models to generalize to downstream tasks by interpreting prompts as task instructions rather than requiring parameter fine-tuning. 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 (Sanh et al. 2021). This approach, inspired by human-like adaptability, quickly gained popularity in NLP (Brown et al. 2020). These advancements influenced CV, where prompt engineering with frozen pre-trained models led to SAM, excelling in *zero-shot* learning and precise object segmentation based on spatial prompts.

Methods for Automating Prompts

While SAM allows manual prompts for mask generation (e.g., clicking or dragging on an image), this approach is impractical for real-world applications. The manual provisions of prompts required by SAM are highly labor-intensive and time-consuming. Moreover, the segmentation performance is heavily dependent on the prompt quality. Crafting precise prompting needs expert domain-specific knowledge, which is not available for all circumstances. To address this, SAM introduces an Automatic Mask Generation (AMG) mode, which autonomously positions numerous prompts in a grid-search manner and generates masks without continuous human input (Kirillov et al. 2023). However, sparse grids may miss small objects, while dense grids (e.g., 32×32 points) result in redundant prompts for large objects, requiring post-filtering. A special version of SAM trained for fully automatic mask generation created the extensive SA-1B dataset (Kirillov et al. 2023) (11 million images, over 1 billion masks) but sacrifices inference speed, further increasing latency.

There is another direction currently by using modern object detection models to generate object-aware prompts (Zhang et al. 2023c) adopts YOLOv8, which is a SOTA architecture for efficient detection with bounding boxes. With the generated box, people can either use its center as an object-aware point prompt or directly adopt the box itself as the prompt. However, this method brings heavy computational overhead due to the size of the object detection models, as they are not specialized for generating prompts. In contrast, our proposed approach is dedicated to predicting prompts and integrated ASF to further improve efficiency by relieving potential redundant generation through a sample-level test-time adaptation.

Test-time Adaptation

Test-time domain adaptation aims to improve model performance on test data that differs from the training data due to a domain gap (Wang et al. 2020; Hu et al. 2020). This adaptation is categorized into two main approaches: backward-based and backward-free. Backward-based adaptation utilizes self-supervised learning, often through entropy minimization, to learn the characteristics of the target domain (Wang et al. 2020; Hu et al. 2019). In contrast, backward-free adaptation relies on batch normalization statistic adjustments, as demonstrated by DUA's running average technique (Mirza et al. 2022) and DIGA's distribution adaptation for semantic segmentation (Wang et al. 2023b). Previous research has also explored test-time domain adaptation for camouflage object segmentation in SAM using a general task description (Hu et al. 2024). Similarly, in our work, we implement instance-level test-time domain adaptation, focusing on adaptively removing redundant prompt candidates. This method enhances mask generation efficiency across diverse datasets without requiring sample-level supervision.

Method

We propose AoP-SAM to efficiently produce essential prompts for accurate mask generations in SAM. In this section, we first briefly review the architecture of SAM to show how our proposed Prompt Predictor collaborates with SAM. Then we introduce our Prompt Predictor, which identifies essential prompt locations that contribute to segmentation performance and further derives a prompt confidence map to guide prompt generation. We also describe the training and inference process of the Prompt Predictor. which is both data and computationally efficient. Lastly, we present the ASF technique to sample and filter prompts during the test-time adaptation.

Preliminaries: SAM

SAM is an advanced image segmentation framework composed of three key components: an Image Encoder, a Prompt Encoder, and a Mask Decoder. These modules collaborate to process images and generate segmentation masks. (1) Image encoder: SAM begins with a robust yet computationally intensive module that extracts essential features from the input image, producing a 64×64 spatial resolution embedding as a compact representation of critical image characteristics. (2) Prompt encoder: The Prompt Encoder processes interactive inputs like points, boxes, or masks, converting them into embeddings that guide the Mask Decoder. This enhances accuracy and supports SAM's remarkable zero-shot generalization. (3) Mask decoder: In the final stage, a two-layer transformer-based module that combines image and prompt embeddings to generate precise segmentation masks, effectively delineating objects or regions of interest. SAM optimizes efficiency by embedding image and prompt inputs only once.

SAM's zero-shot generalization is underpinned by the SA-1B dataset, containing over 1 billion masks and 11 million images—400 times larger than prior segmentation datasets. This extensive dataset allows SAM to segment new images without additional training. However, training SAM is resource-intensive: for instance, training the ViT-H-based SAM model on SA-1B for two epochs requires 256 GPUs and a batch size of 256 images (Kirillov et al. 2023), emphasizing the significant resources dedicated to the image encoder. This high computational cost motivates the reuse of the image encoder's outputs in later computations to maximize efficiency. For more details, see (Kirillov et al. 2023).

Prompt Predictor for Prompt Confidence Map

To enhance the automation of prompt production, we address two main challenges: the detection of essential entities within full images and the efficient identification of potential prompt locations. We propose a lightweight Prompt Predictor that integrates processed data from the Segment Anything Model (SAM), reducing computational complexity while maintaining tight coupling with SAM, which improves system efficiency. As illustrated in Figure 2, the image segmentation process begins with the computation of image embeddings. Whenever new prompts are provided, corresponding masks can be generated, meaning that prompt embeddings are computed and then injected into the mask decoder to produce these masks. Following recent methodologies, we initiate prompt generation after the image embedding is completed. This approach allows our Prompt Predictor to reuse pre-processed image inputs and their corresponding embeddings to generate a Prompt Confidence Map. This map identifies regions of high confidence, which

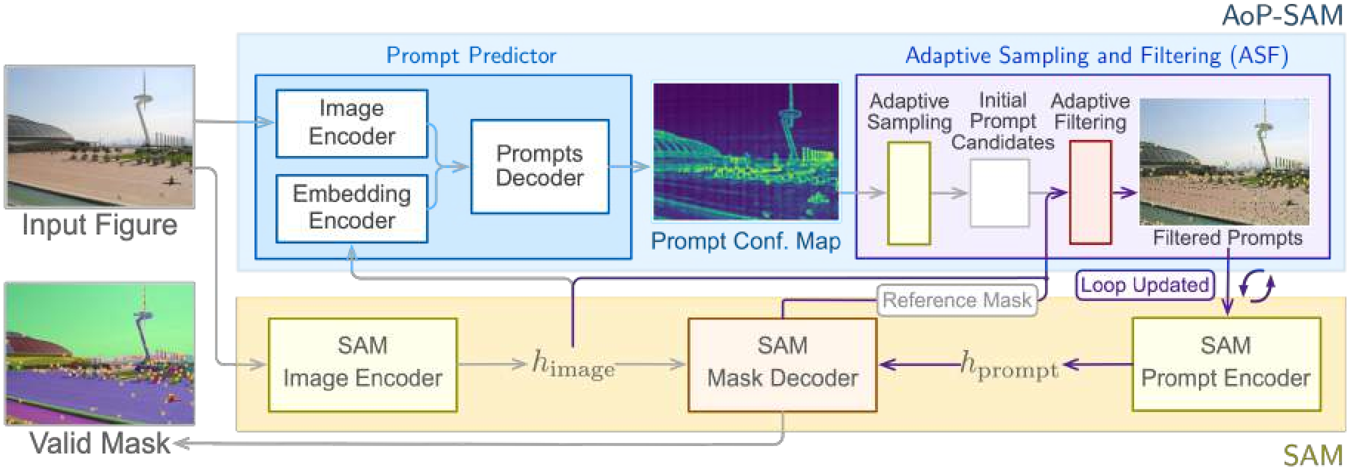

Figure 2: The architecture of our proposed AoP-SAM consists of two key components: the prompt predictor and the Adaptive Sampler and Filter (ASF) Module. The prompt predictor operates by taking the image input and the computed image embedding from SAM's image encoder as inputs. Prompt predictor then generates a Prompt Confidence Map (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 results.

can then be selected as prompts and used for segmentation within the SAM framework.

Prompt Predictor Architecture. The model employs two distinct CNN-based encoders: one for processing the original image and the other for handling the ViT embedding. The image encoder captures spatial features through three convolutional layers, each followed by ReLU activations to enhance feature extraction and introduce nonlinearity. For the ViT embedding, the process begins by reshaping it to match the spatial dimensions of the image input, ensuring seamless integration with the image data during decoding. After reshaping, the ViT embedding is passed through its own encoder, transforming it into a 32-channel feature map. This alignment of the ViT embedding with the image's feature space allows the model to effectively fuse and decode information from both inputs, leveraging their strengths for more accurate and robust outcomes.

Following the encoding process, the Prompt Predictor concatenates the outputs from both the image encoder and the ViT embedding encoder along the channel dimension. This operation merges the feature maps, creating a fused representation that integrates both spatial and contextual information from the original image and the ViT embedding. The fused feature map is then processed through a series of convolutional layers in the Prompt decoder. These layers progressively reduce the dimensionality of the combined features, refining the representation and distilling it into a more compact form that retains the most relevant information. The decoding process culminates in a Sigmoid-activated layer, producing the final output as a Prompt Confidence Map. This map effectively highlights regions of interest, from which prompts can be sampled. The Sigmoid function normalizes the output to a range between 0 and 1, making it well-suited for generating probability maps that allow for the flexible selection of high-confidence regions, ensuring the prompts generated are both relevant and useful for mask segmentation within the SAM framework.

The Prompt Predictor is designed with efficiency in mind, maintaining a small memory footprint and low computational burden. The use of lightweight CNN-based encoders for both the original image and the ViT embedding ensures efficient feature extraction without the need for overly complex architectures. Each encoder consists of a limited number of convolutional layers, reducing computational load while capturing essential features. By reshaping the ViT embedding to match the image dimensions and aligning it within the same feature space, the model streamlines the process, avoiding unnecessary intermediate computations. The final step of concatenating outputs along the channel dimension, followed by a compact decoder, further minimizes resource usage. The decoder efficiently refines the fused feature map and produces the final Prompt Confidence Map through a single Sigmoid layer, which is computationally inexpensive. This efficient design allows the Prompt Predictor to deliver high performance while keeping memory usage and computational demands to a minimum, making it an ideal solution for automating prompt generation in a manner that is both resource-efficient and fast.

Training of Prompt Predictor Unlike traditional object detection models that typically train on datasets like COCO (Lin et al. 2014), our approach to training AoP-SAM is both data-efficient and closely aligned with SAM by leveraging the SA-1B dataset (Kirillov et al. 2023). The SA-1B dataset, containing over 1 billion masks and their corresponding prompts, was specifically chosen to ensure that AoP-SAM inherits the robustness and generalizability of SAM. By training on the same dataset, we align AoP-SAM with SAM's capabilities, particularly in handling diverse and unseen data. Given that point-type prompts align most effectively with the Prompt Confidence Map, which can be generated by our Prompt Predictor, we use these point prompts from the SA-1B dataset as ground truth to train our model. To ensure a diverse and challenging training set, we carefully curated a selection of samples from the SA-1B dataset, encompassing a broad range of semantic classes and complex scenarios. This careful curation is crucial for maintaining the model's ability to generalize effectively across various object scales and complexities.

During training, the learnable parameters include the Image Encoder, ViT Encoder, and Prompt Decoder. To generate the Prompt Location Map Ground Truth, we place point prompts within a blank map that matches the dimensions of the pre-processed image sample. Since the initial ground truth map is too sparse for effective training, we refine it using a combination of uniform and Gaussian kernels. This refinement enhances the precision of prompt locations, making the training feasible and significantly improving the prompt generation process. The training process involved iterative refinement of the model's parameters using a MSELoss function, with optimization carried out via the Adam optimizer. We employed a learning rate of and trained the model for 1000 epochs, using gradient accumulation to handle larger batch sizes effectively.

ASF for Essential Prompts

To effectively sample prompt candidates from the Prompt Confidence Map, a Gaussian filter is first applied to smooth the confidence map, enhancing key regions while reducing noise. Following this, local maxima within the smoothed confidence map are identified by isolating critical points of interest, taking into account both a minimum distance between peaks and an absolute threshold. These local maxima represent potential prompt candidates. The identified points, initially located within the resized output dimensions, are then mapped back to the original image coordinates by scaling them according to the ratio between the resized output and the original image size. This mapping results in a set of coordinates that accurately reflect the positions of significant features within the original input space, making them suitable as prompt candidates.

Even though sampling the candidates from the Prompt Confidence Map can make a good selection for Automating Prompts, however, it is still possible to make some redundant prompts in some cases, therefore, we adapt a further fine filtering to remove redundant candidates. In the processes of mask generation in SAM, due to memory constraints, the prompt can be divided into several batches and processed iteratively, therefore, we can take advantage of computated masks as references to predict which prompts remained in the prompt candidates will be redundant and result in the same mask with them, so we need to figure out the spatial location and semantic meaning of these generated masks first and then obtain the Prompt Elimination Map (Zhang et al. 2023e) of the processed prompts.

We utilize the image feature map result from the vit_embedding and also reference masks during mask generation, where the image feature map containing the original image input information and mask data containing the information of location are generated, we denote image feature map as $F_{\text{feat}} \in \mathbb{R}^{h \times w \times c}$ and each of these n masks as $M_{\text{ref}} \in \mathbb{R}^{h \times w}$ with h, w denoting the dimension of the image feature map, c as the feature dimension and n is the number of predicted masks.

The n down-sampled reference masks M_{ref} are used to extract the mask feature M_{feat} from the image feature map one by one and we can get a set of n mask features. Each mask feature then adopts an average pooling to aggregate its global visual embedding. After this, we can obtain a Prompt Elimination Map with confidence C for each reference mask by doing a cosine similarity between pixel-wisely L2-normalized mask and image feature M_{norm} and F_{norm} as

$$\{C^i\}_{i=1}^{n} = \{F_{\text{norm}} \times M_{\text{norm}}^{i}\}_{i=1}^{n}\ , N_{\text{norm}} \in \mathbb{R}^{n \times h \times w} \qquad (1)$$

On top of this, we adopt another average pooling to aggregate all n local maps to obtain the overall Prompt Elimination Map of the generated mask as

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i\ , C \in \mathbb{R}^{h \times w} \qquad (2)$$

By incorporating the Prompt Elimination Maps of every high-quality mask, the Elimination Map can take the visual appearance of different objects from existing masks into consideration, and acquire a relatively comprehensive location estimation. Each pixel on the upsampled Elimination Map has a Elimination Score that indicates the likelihood of that pixel is in the same spatial location and having the same semantic meaning as generated masks in the original image, The higher the Elimination Score of each pixel is in the map, the more likely prompting at that pixel will resulting in the duplicated masks with the existing masks.

After getting the Elimination Map generated from all existing masks, we can then obtain the elimination threshold T_{elim} by calculating the Elimination Score of current processing prompts along with the confidence Intersection over Union (IoU) scores of their resulting masks, as follows:

$$T_{\text{elim}} = \frac{1}{n}\sum_{i=1}^{n} \text{IoU}^i \times C^i \qquad (3)$$

By multiplying the IoU score of the masks generated by prompts with the Elimination Scores of current processing prompts, the threshold biases more with prompts which can generate the higher-quality masks.

The remaining prompts in the pool will obtain their corresponding Elimination Scores, which are compared to the threshold. If a prompt's score exceeds the threshold, it is considered redundant and would generate duplicate masks if further processed; therefore, it should be eliminated from the prompt pool. By doing this way, redundant prompts will be eliminated and only essential prompts will be kept in the prompt pool and used to generate masks in the following iterations.

Image Encoders	Auto Prompts Methods	SA-1B				COCO				LVIS			
		mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*
MobileSAM	AMG_S	29.8	-	4.5	38.6	56.0	-	**1.9**	33.5	56.2	-	**1.9**	33.4
	AMG_D	46.9	-	9.1	71.0	60.9	-	**1.9**	55.9	61.1	-	**1.9**	55.5
	OAS(Box)	50.7	0.191	7.3	100	55.5	0.187	4.2	44	55.7	0.188	4.0	38
	OAS(Central Point)	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	AMG_S	40.0	-	5.7	55.5	61.4	-	4.4	48.8	63.2	-	**4.3**	49.5
	AMG_D	65.6	-	10.3	108.9	67.7	-	**4.3**	86.0	69.2	-	**4.3**	86.5
	OAS(Box)	65.8	0.150	9.1	100	63.3	0.152	5.4	44	62.9	0.151	5.3	38
	OAS(Central Point)	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	AMG_S	40.8	-	7.1	56.3	63.3	-	5.7	49.8	64.9	-	5.6	50.5
	AMG_D	66.8	-	11.8	109.6	69.5	-	5.7	87.4	71.0	-	5.6	88.0
	OAS(Box)	66.9	0.160	10.4	100	64.1	0.152	6.8	44	63.3	0.153	6.6	38
	OAS(Central Point)	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 1: Results on Image Segmentation with bounding box supervision and point supervision. Best are in **bold**.

Method's variant			SA-1B				COCO				LVIS			
Prompt Predictor	Adaptive Sampling	Adaptive Filtering	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	#*P*
✓			57.2	0.059	7.2	106.4	67.9	0.078	5.7	70.4	60.9	0.075	5.7	60.6
✓	✓		72.8	0.130	10.1	120.1	70.5	0.122	5.7	97.9	71.7	0.121	5.7	97.5
✓	✓	✓	71.3	0.122	6.6	107.8	70.1	0.112	5.7	91.1	71.9	0.122	5.7	89.7

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

Experiments

To evaluate AoP-SAM across various scenarios, we selected three different image encoders and implemented five automating prompts methods. This approach allows us to comprehensively assess both the accuracy and efficiency of our method under different conditions.

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 (Kirillov et al. 2023). The COCO dataset includes 41,000 images and 200,000 masks, covering a wide range of common objects (Lin et al. 2014). LVIS, designed for long-tail distributions, provides 5,000 images and 25,000 masks, emphasizing fine-grained categories (Gupta, Dollar, and Girshick 2019). 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 (Kirillov et al. 2023). We also examine the Object-Aware Sampling (OAS) method, which employs YOLOv8 to generate bounding box prompts (Zhang et al. 2023c). 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.

Evaluating SAM's accuracy is challenging as it generates masks without predefined labels, making traditional metrics like mIoU (Shotton et al. 2006; Han et al. 2023), *mAP* (Lin et al. 2014; Henderson and Ferrari 2017), and *PQ*(Kirillov et al. 2019) unsuitable (Zhang et al. 2023d). To address this, we use the greedy IoU algorithm (Zhang et al. 2023d), which matches each SAM mask with the closest ground truth mask based on IoU and calculates the mean IoU (mIoU) for 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 ($\text{Inf}_{\text{Lat.}}$)(s) for producing prompts and peak memory ($\text{Peak}_{\text{Mem.}}$)(GB) consumption during mask generation as key metrics. Additionally, we count the number of essential prompts (#*P*) as a reference point for comparing methods. It is important to note that a higher value of mIoU, or lower values of $\text{Inf}_{\text{Lat.}}$ and $\text{Peak}_{\text{Mem.}}$, 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 (Kirillov et al. 2023), we enable the option for generating multiple mask outputs from a single prompt for point prompts, while disabling it for box prompts (Zhang et al. 2023c). 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 Prompt

(a) Sampling Smoothing Factor				(b) Confidence Intensity Threshold				(c) Prompt Spacing Factor				(d) Prompt Elimination Threshold			
Factor	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	Thr.	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	Factor	mIoU ↑	$\text{Inf}_{\text{Lat.}}$ ↓	$\text{Peak}_{\text{Mem.}}$ ↓	Thr.	mIoU ↑	$\text{Mask}_{\text{Lat.}}$ ↓	$\text{Ratio}_{\text{elim.}}$ ↑
1	**72.4**	0.124	51.5	0.1	**70.9**	0.121	10.1	4	**72.7**	0.123	10.0	1.25	68.4	**0.671**	**51.5**
2	70.4	0.122	42.2	0.2	70.4	0.122	9.75	5	71.6	0.123	9.88	1.3	70.4	0.799	42.3
3	67.3	**0.118**	32.7	0.3	68.7	**0.116**	**9.52**	6	70.4	0.122	**9.75**	1.35	71.6	0.93	32.7
4	63.4	0.122	**24.6**	0.4	66.4	0.117	9.60	7	68.9	**0.117**	9.82	1.4	**72.2**	1.04	24.6

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

Confidence Map, we first apply a Smoothing Factor=2, a Confidence Intensity Threshold=0.2, and a Prompt Spacing Factor=2 as initialized parameters. 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.

Experiment Results and Analysis

Experiment Results. Table 1 compares the performance of various Automating Prompt methods across different image encoders and evaluated on three datasets. Across all datasets and image encoders, AoP-SAM consistently achieves the highest mIoU scores, even though bounding box methods inherently benefit from more spatial information. This underscores the effectiveness of AoP-SAM in leveraging prompts for accurate segmentation, surpassing both traditional methods and those that rely on advanced object detection models. The AoP-SAM method not only improves accuracy but also demonstrates competitive latency and memory usage. For example, on the SA-1B dataset with the ViT_H encoder, AoP-SAM achieves a latency of 0.122s and peak memory usage of 6.6GB, which are within acceptable ranges while delivering superior segmentation performance. Overall, the OAS methods (using either box or central point prompts) generally perform better than the baseline AMG_S and AMG_D methods but fall short of AoP-SAM. This indicates that while object-aware sampling improves prompt effectiveness, the adaptive sampling and filtering techniques employed in AoP-SAM further enhance the accuracy of segmentation and efficiency of Automating Prompts.

Component Analysis. We further analyze the impact of components including the Prompt Predictor, Adaptive Sampling, and Adaptive Filtering in Table 2 on the same datasets. When Adaptive Sampling is enabled, there is a notable improvement in mIoU compared to using only the Prompt Predictor. However, the best performance is observed when both Adaptive Sampling and Adaptive Filtering are used together, highlighting the importance of filtering redundant prompts to enhance segmentation accuracy. The study shows that while the full AoP-SAM configuration achieves the highest mIoU, it slightly increases latency and memory usage, a key trade-off for speed-sensitive applications.

Sampling Smoothing Factor. In Table 3a, we apply Gaussian filtering to the heatmap using the Sampling Smoothing Factor. A larger Sampling Smoothing Factor allows the model to cover a broader area, providing more substantial smoothing, which is useful for reducing memory access during preparation and processing.

Prompt Methods	Latency (Sec) ↓ (SA-1B)	Latency (Sec) ↓ (COCO)	Latency (Sec) ↓ (LVIS)	Peak Mem ↓ (GB)
OAS(Box)	1.16	1.01	0.99	0.78
OAS(Central)	1.32	1.21	1.23	0.78
AoP-SAM	**0.65**	**0.77**	**0.84**	**0.042**

Table 4: Experimental results with MobileSAM for prompt automation efficiency on an Nvidia Jetson Orin Nano Edge GPU.

From heatmap to point prompt. In Table 3b-3c, we explore various parameter settings to transform the confidence map into optimized point prompts. By adjusting the Confidence Intensity Threshold and Prompt Spacing Factor, we aim to identify the optimal points that most accurately represent the critical areas in the confidence map. These adjustments help refine the sensitivity of the point selection process, ensuring that the resulting point prompts are both precise and reliable.

Prompt Elimination Threshold. We evaluate the impact of the Prompt Elimination Threshold on the prompt removal ratio in Table 3d. As the Prompt Elimination Threshold decreases, the prompt removal ratio increases, resulting in a speed-up effect of the mask generation while may slightly affect accuracy.

Edge Device. We conducted the prompt automation experiment on an Nvidia Jetson Orin Nano Edge GPU, obtaining the results in Table 4. Due to hardware limitations, only MobileSAM (Zhang et al. 2023b) could run, as other pre-trained models exhausted the edge GPU memory. We focused on evaluating the efficiency of prompt automation, with accuracy expected to align with standard GPU results. These results further demonstrate AoP-SAM's reliability on edge devices, achieving lower inference latency and reduced peak memory usage, making AoP-SAM well-suited for deployment in resource-constrained environments.

Conclusion

We propose AoP-SAM, a novel approach designed to efficiently generate essential prompts for accurate mask generation in SAM. Our method introduces a lightweight prompt predictor, which is trained to predict optimal prompt locations, complemented by a test-time adaptive sampling and filtering technique that automatically produces these prompts for SAM. We evaluate the accuracy and efficiency of AoP-SAM on three segmentation datasets with three SAM family models. The results demonstrate that AoP-SAM enhances both the accuracy and efficiency of SAM in generalized image segmentation tasks, making it ideal for automated prompt-based segmentation tasks with SAM.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea governments(MSIT)(No. 2022-0-01036, Development of Ultra-Performance PIM Processor Soc with PFLOPS-Performance and GByte-Memory & No.2022-0-01037, Development of High Performance Processing-In-Memory Technology based on DRAM) and the authors would like to express their sincere gratitude to Adiwena Putra for comments on an earlier draft.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; and Dietmayer, K. 2020. 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.

Gupta, A.; Dollar, P.; and Girshick, R. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5356–5364.

Han, D.; Zhang, C.; Qiao, Y.; Qamar, M.; Jung, Y.; Lee, S.; Bae, S.-H.; and Hong, C. S. 2023. Segment anything model (sam) meets glass: Mirror and transparent objects cannot be easily detected. *arXiv preprint arXiv:2305.00278*.

Henderson, P.; and Ferrari, V. 2017. 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*, 198–213. Springer.

Hesamian, M. H.; Jia, W.; He, X.; and Kennedy, P. 2019. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32: 582–596.

Hu, J.; Lin, J.; Gong, S.; and Cai, W. 2024. 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, 12511–12518.

Hu, J.; Tuo, H.; Wang, C.; Qiao, L.; Zhong, H.; and Jing, Z. 2019. Multi-Weight Partial Domain Adaptation. In *BMVC*, 5.

Hu, J.; Tuo, H.; Wang, C.; Qiao, L.; Zhong, H.; Yan, J.; Jing, Z.; and Leung, H. 2020. Discriminative partial domain adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, 632–648. Springer.

Hurtado, J. V.; and Valada, A. 2022. Semantic scene segmentation for robotics. In *Deep learning for robot perception and cognition*, 279–311. Elsevier.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9404–9413.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Mirza, M. J.; Micorek, J.; Possegger, H.; and Bischof, H. 2022. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14765–14775.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Shotton, J.; Winn, J.; Rother, C.; and Criminisi, A. 2006. TextonBoost: Joint Appearance, Shape and Context Modeling for Mulit-Class Object Recognition and Segmentation. In *European Conference on Computer Vision (ECCV)*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; and Huang, T. 2023a. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors*, 23(16): 7190.

Wang, W.; Zhong, Z.; Wang, W.; Chen, X.; Ling, C.; Wang, B.; and Sebe, N. 2023b. 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*, 24090–24099.

Zhang, C.; Han, D.; Qiao, Y.; Kim, J. U.; Bae, S.-H.; Lee, S.; and Hong, C. S. 2023a. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. *arXiv preprint arXiv:2306.14289*.

Zhang, C.; Han, D.; Qiao, Y.; Kim, J. U.; Bae, S.-H.; Lee, S.; and Hong, C. S. 2023b. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*.

Zhang, C.; Han, D.; Zheng, S.; Choi, J.; Kim, T.-H.; and Hong, C. S. 2023c. Mobilesamv2: Faster segment anything to everything. *arXiv preprint arXiv:2312.09579*.

Zhang, C.; Puspitasari, F. D.; Zheng, S.; Li, C.; Qiao, Y.; Kang, T.; Shan, X.; Zhang, C.; Qin, C.; Rameau, F.; et al.

2023d. A survey on segment anything model (sam): Vision foundation model meets prompt engineering. *arXiv preprint arXiv:2306.06211*.

Zhang, R.; Jiang, Z.; Guo, Z.; Yan, S.; Pan, J.; Dong, H.; Gao, P.; and Li, H. 2023e. Personalize Segment Anything Model with One Shot. *arXiv preprint arXiv:2305.03048*.