Leveraging Hallucinations to Reduce Manual Prompt Dependency in Promptable Segmentation

Jian Hu[1], Jiayi Lin[1], Junchi Yan[2], Shaogang Gong[1]
[1]Queen Mary University of London, [2]Shanghai Jiao Tong University
{jian.hu, jiayi.lin, s.gong}@qmul.ac.uk, yanjunchi@sjtu.edu.cn
https://lwpyh.github.io/ProMaC/

Abstract

Promptable segmentation typically requires instance-specific manual prompts to guide the segmentation of each desired object. To minimize such a need, task-generic promptable segmentation has been introduced, which employs a single task-generic prompt to segment various images of different objects in the same task. Current methods use Multimodal Large Language Models (MLLMs) to reason detailed instance-specific prompts from a task-generic prompt for improving segmentation accuracy. The effectiveness of this segmentation heavily depends on the precision of these derived prompts. However, MLLMs often suffer hallucinations during reasoning, resulting in inaccurate prompting. While existing methods focus on eliminating hallucinations to improve a model, we argue that MLLM hallucinations can reveal valuable contextual insights when leveraged correctly, as they represent pre-trained large-scale knowledge beyond individual images. In this paper, we utilize hallucinations to mine task-related information from images and verify its accuracy for enhancing precision of the generated prompts. Specifically, we introduce an iterative **Pro**mpt-**Ma**sk **C**ycle generation framework (ProMaC) with a prompt generator and a mask generator. The prompt generator uses a multi-scale chain of thought prompting, initially exploring hallucinations for extracting extended contextual knowledge on a test image. These hallucinations are then reduced to formulate precise instance-specific prompts, directing the mask generator to produce masks that are consistent with task semantics by mask semantic alignment. The generated masks iteratively induce the prompt generator to focus more on task-relevant image areas and reduce irrelevant hallucinations, resulting jointly in better prompts and masks. Experiments on 5 benchmarks demonstrate the effectiveness of ProMaC. Code given in https://lwpyh.github.io/ProMaC/.

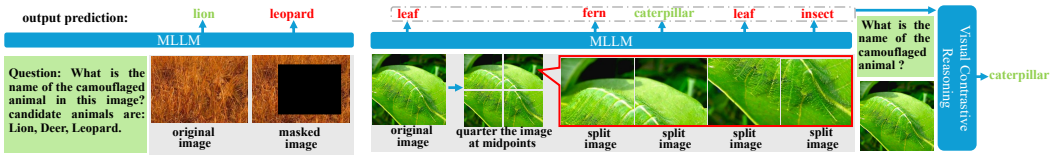
1 Introduction

Current promptable segmentation methods rely on instance-specific manual prompts to guide segmentation, greatly limiting its large-scale application. Recently, a manual-free task-generic promptable segmentation approach was introduced [21]: only a single task-generic prompt is needed for all samples under the same task, e.g., "camouflaged animal" is a task-generic prompt for all images in a camouflaged object detection task. The model segments task-relevant objects in various images based on this generic prompt, significantly reducing the annotation workload.

A task-generic prompt is both coarse and potentially ambiguous, can result in poor segmentation when directly applied. To address this problem, existing methods [21, 38] utilize the prior knowledge

This work was supported by Veritone, Adobe, the Apocrita HPC facility from the QMUL Research-IT, NSFC (92370201, 62222607) and Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

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



(a) Hallucination by co-occurrence prior. (b) Using hallucinations can benefit accuracte prompt generation.

Figure 1: (a) During MLLM pretraining, leopards often co-occur with grass. If the lion is masked, the model incorrectly identifies it as a leopard based on the grass. (b) Directly inputting the image into MLLM causes the hidden caterpillar being incorrectly predicted as a leaf. Splitting the image results in interested objects being incomplete or absent, prompting MLLM to induce hallucinations and utilize prior knowledge to predict potential task-related objects within the image. Our visual contrastive reasoning then eliminates the hallucinations and validates the gathered predictions, aiding in the accurate identification of the caterpillar.

embedded in Multimodal Large Language Models (MLLMs) to infer more detailed, instance-specific prompts, such as bounding boxes or keywords, to guide the segmentation. However, these MLLMs often generate hallucinations due to object co-occurrence priors [68, 65], mistakenly predicting non-existent elements based on the environment as instance-specific prompts (Fig.1(a)). This can mislead segmentation and degrade model performance. While it is common to consider MLLM's hallucinations as detrimental and should be eradicated [52], this phenomenon actually demonstrates a MLLM's significant capacity for contextual inference based on prior training. We want to explore MLLM hallucinations as a valuable untapped knowledge resource for scene understanding, critical in complex segmentation scenarios. In practice, when task-related objects are not prominently visible, hallucinations can fill in missing information with plausible predictions based on learned patterns of association. Moreover, they can also extend beyond these familiar patterns, exploring and identifying new relationships within the data that were not explicitly taught during training. This dual ability to replicate and innovate makes hallucinations a valuable asset for enhancing model performance in complex or new situations. This predictive reasoning capacity not only fills perceptual gaps but also enriches the model's understanding, as hallucinations utilize prior knowledge to replicate and discover new patterns, enhancing insight into the target domain (see Fig.1(b)). Despite the potential benefits, using hallucinations to extract useful information from images to aid task remains unexplored.

In this work, instead of direct eliminating hallucinations, we utilize them as prior knowledge to mine extended task-related information from a given test image, performing scene understanding on the image before segmentation, then systematically reduce irrelevant hallucinations iteratively by visual masking verification, optimizing jointly instance-specific prompts and masks. To this end, we introduce an iterative, training-free Prompt-Mask Cycle Generation method (ProMaC) that refines segmentation through cyclic interactions between a prompt and mask generator (see Fig. 2). The prompt generator uses a multi-scale chain-of-thought prompting mechanism, which utilizes hallucinations to hypothesize and visual masking to verify, thereby creating more accurate instance-specific prompts. We trigger the hallucinatory tendencies of MLLMs, the process starts by dividing the image into patches at different scales and positions. Such partial visibilities of objects facilitate MLLMs to hypothesize potential object semantic labels and visual locations based on its prior knowledge. For validating the correctness of these hypotheses, we formulate a visual contrastive reasoning mechanism to generate contrastive images that contain only the background without any potential task-related objects. This helps identify all possible co-occurrence hallucinations caused by the background. By comparing these contrastive images with the original images, the MLLM effectively distinguishes between accurate hypotheses and those influenced by misleading prior knowledge, leading to more reliable prompts. Given the current promptable segmentation models' strength at mask prediction but struggle with label prediction, the mask generator uses mask semantic alignment to ensure that the produced masks align with the task semantics. These aligned masks not only serve as outputs but also guide the prompt generator in subsequent cycles, enhancing both prompt and mask quality continuously. **Our contributions are three-folds:**

1). We introduce a training-free Prompt-Mask Cycle Generation (ProMaC) to perform two tasks: Explore MLLM hallucinations as prior knowledge to enhance contextual scene understanding on each test image; systematically reduce irrelevant hallucinations to verify iteratively and optimize jointly both generated prompts and visual masking in object segmentation.

2). We formulate an iterative optimization method including a prompt generator and a mask generator. To improve prompt relevance, the prompt generator utilizes a multi-scale chain of thought approach. It

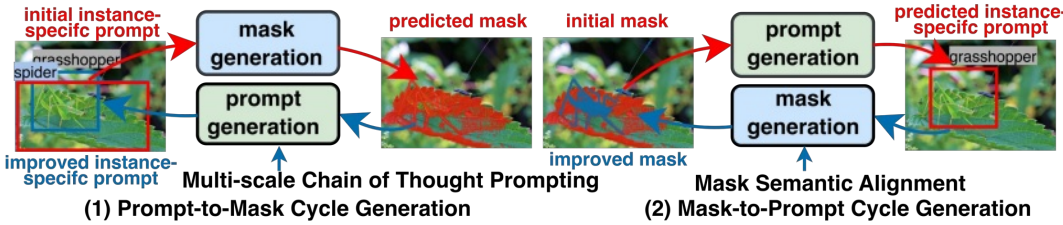


Figure 2: An overview of ProMac: Masks created iteratively by the mask generator guide the prompt generator to jointly improve instance-specific prompts and visual masking in segmentation.

first leverages hallucinations to expand task-related plausible prompts, then applies visual contrastive reasoning to validate and reduce irrelevant prompts. ProMaC's mask generator overcomes SAM's shortcomings in label prediction by creating masks that align semantically with generated prompts.

3). Comprehensive comparative evaluations on 5 different segmentation tasks with 12 diverse datasets against 22 existing models demonstrate the effectiveness of ProMaC.

2 Related Works

Promptable Segmentation refers to object segmentation with active interactions from user inputs. Interaction methods vary from points, boxes, to scribbles. SAM [29], AV-SAM[41], GroundingSAM [38] and SEEM [71] accept video, audio, and multimodal inputs. However, they often rely on manual prompts, which can be unclear and subjective. Even with these prompts, they typically excel only in specific tasks. To address this issue, GenSAM [21] introduces a manual-free promptable segmentation setting, where only one task-generic prompt is provided. This prompt can be applied to all images within the task for instance-specific segmentation without any additional manual prompting. GenSAM primarily utilizes MLLM to infer the names of task-related objects in the images and uses them as instance-specific prompts for SAM to guide segmentation. However, GenSAM lacks spatial information about objects and may lead to inaccurate prompt predictions in complex scenes.

Hallucinations in MLLMs refers to models generate content that does not exist in the input data [65]. This issue often arises from the models leveraging extensive prior training rather than just the immediate input, leading to false predictions on fine-grained details. There are some efforts to mitigate this problem, including refining training processes [55, 44] and improving model architectures [4]. Other efforts focus on aligning model outputs more closely with actual data, employing feedback mechanisms for real-time adjustments [54]. While current works focus on eliminating hallucinations to enhance performance [31, 64], our work explores how to utilize hallucinations to expand and reason plausible context and validate them iteratively to remove irrelevant generalizations.

Visual Marking for MLLMs has been explored in recent research to prompt MLLMs through manipulation of visual inputs: (i) adding learnable soft tokens to visual inputs for efficient parameter tuning [1, 28], (ii) using image sequences as demonstrations of a new task [2, 9], and (iii) overlaying visual markers like masks, boxes, and circles onto visual inputs to ground regions [61, 54]. Our work falls into the third category, employing visual guidance for reasoning. Yang et al. [59] propose set-of-mark (SoM) prompts, where images are segmented and numbered regions to improve GPT-4V [43] visual grounding. However, as detailed in Tab.5, we confirm previous findings [5] that this visual marker approach struggles with open-source MLLMs like LLaVA. Instead of proprietary models [54] or fine-tuning [5, 8, 23], our training-free ProMaC uses inpainting task-related regions and contrasting model output distributions to prompt MLLMs.

3 Methodology

We introduce ProMaC, a cycle-generation method for segmenting unknown multiple classes of objects training-free with only a single task-generic prompt. Specifically, given an image $X \in \mathbb{R}^{H \times W \times 3}$ from a test set, ProMaC employs a task-generic prompt P_g across datasets in the same task to produce a final segmentation mask $M \in \mathbb{R}^{H \times W}$, thereby removing the need for individual supervision for each image. The prompt generator leverages prior knowledge gained to reason and deduce instance-

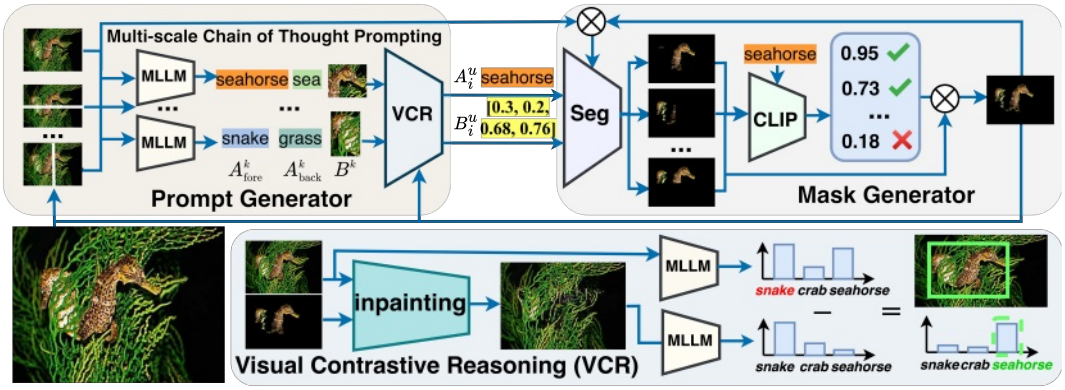


Figure 3: ProMaC consists of a prompt generator and a mask generator for cyclical optimization. The prompt generator employs multi-scale chain-of-thought prompting. It initially use hallucinations for exploring task-related information within image patches. It identifies task-relevant objects and their backgrounds (A^k_{fore}, A^k_{back}) along with their locations (B^k). Subsequently, it uses visual contrastive reasoning to refine and finalize instance-specific prompts (A^u_i, B^u_i) by eliminating hallucinations. The mask generator then processes these prompts into the segmentation model ("Seg"), producing a mask aligned with task semantics. This mask further guides the visual contrastive reasoning process, which leverages an inpainting model to eliminate masked regions, creating contrastive images. These images enable the prompt generator to further refine its prompts, enhancing segmentation accuracy.

specific prompts, which then guide a mask generator to create masks aligned with task semantics. These masks act both as the current segmentation outcome and as visual markers for generating subsequent prompts. Training-free ProMaC relies solely on test-time adaptation.

3.1 Prompt Generator

Prompt generator employs MLLMs to generate instance-specific prompts based on image content and prior knowledge. It transforms the general prompt P_g, into an instance-specific prompt for each individual instance, providing more detailed descriptions of task-relevant objects. MLLM with parameters θ receives an image X and query P as inputs. X provides contextual visual information to assist the model in generating a relevant response y to the query P. The response y is sampled auto-regressively from the probability distribution conditioned on P and X as follows:

$$y_t \sim p_\theta(y_t \mid X, P, y_{<t}) \propto \exp(\text{logit}_\theta(y_t \mid X, P, y_{<t})) \tag{1}$$

where y_t denotes the token at time step t, and $y < t$ represents the sequence of generated tokens up to the time step $(t-1)$. In practice, predicted task-relevant objects can often blend into the background due to texture, color, size, or position, leading to inaccuracies in instance-specific prompts. To address this problem, we explore MLLM hallucinations as contextual prior knowledge from pretraining, rather than eliminate them. These hallucinations are particularly useful when direct visual cues are absent or ambiguous, helping the model fill in information gaps and hypothesize potential task-related elements within the image that are not prominent. By revealing these often-overlooked subtle associations, hallucinations provide a more comprehensive scene understanding of the image content. This deeper contextual understanding provide a reasoning context for generating more accurate and relevant instance-specific prompts candidates.Thus, using hallucinations to uncover task-related knowledge helps overcome challenges from visual ambiguities and object camouflage in complex scenes. To this end, we propose a multi-scale chain-of-thought prompting strategy that stimulates hallucinations to leverage prior knowledge, fully extracts task-relevant information, and then uses this information to enhance the precision of the generated instance-specific prompts.

3.1.1 Multi-scale Chain of Thought Prompting

Multi-scale Chain of Thought Prompting consists of two processes: Gathering candidate knowledge and generating accurate instance-specific prompts. To efficiently collect task-relevant information from an image, as shown in Fig. 3, we divide the input image into patches at various scales by cutting horizontally, vertically, or by leaving it whole. These patches are then processed by the MLLM to

gather preliminary instance-specific prompts. The differing levels of task-relevant object visibility in each patch prompt the MLLM to induce hallucinations. These hallucinations utilize prior knowledge to explore connections between the image data and the associated task, aiding in the detection of potential bounding boxes and object names. The process is computed by:

$$B^k = \text{MLLM}\left(X^k, C^k, P_B\right), \qquad A^k_{\text{fore}}, A^k_{\text{back}} = \text{MLLM}\left(X^k, C^k, P_A\right), \tag{2}$$

where C^k is the caption generated by MLLM for the $k-$th image patch X^k. P_g is task-generic prompt. For bounding box prediction, the prompt P_B, which instructs "*This image is from the P_g detection task, output the bounding box of the P_g.*". This guides the MLLM to predict the bounding box B^k of the task-related objects within the patch. For predicting name, the prompt P_A, stating "*Output the name of the P_g and its environment in one word.*" is used, guiding the MLLM to predict the names of the task-related objects A^k_{fore} and their backgrounds A^k_{back} from each patch. The preliminary data, including object names A^k_{fore} and bounding boxes B^k, gathered from various patches, are compiled into candidate lists $\mathcal{A}_i$ and $\mathcal{B}_i$. Here, i denotes the iteration in the iterative learning cycle. In this process, the hallucinations employed are essentially based on object co-occurrence priors, where objects commonly associated with background elements during pre-training are predicted to be task-relevant, even if they are not present in the current image. This prior knowledge is useful during the knowledge collection stage as it uncovers implicit relationships and details in the image. However, it can also reduce accuracy of the later fine-grained instance-specific prompts generation. Therefore, it is crucial to control these hallucinations in the latter stage to prevent incorrect predictions.

Visual Contrastive Reasoning. To mitigate hallucinations caused by object co-occurrence priors, recent research highlights particularly relevant regions of an image to direct MLLMs focus toward task-related elements, thereby minimizing background interference and enhancing model accuracy [59, 54]. To achieve this, visual markers are employed to steer MLLM attention on task-relevant visual regions, thereby reducing hallucinations. While closed-source MLLMs like GPT-4V [43] can interpret these markers effectively, they are costly and large. In contrast, models like LLaVA [37] are open-source, but cannot process visual markers such as points or bounding boxes, and employing these markers might disrupt the original pixel data, degrading performance on LLaVA (see Tab. 5). Moreover, accurate pixel-level visual markers are unavailable in our setting. To solve this problem, we aim to enable LLaVA to focus on task-related regions without altering the original pixel data, thereby effectively minimizing hallucinations and enhancing the precision of instance-specific prompts.

Despite the absence of instance-level annotations, promptable segmentation models produce masks with detailed textures, which provide rich positional and textural information about interested regions. We use these masks as visual markers to guide a MLLM to focus on task-related areas during the generation of instance-specific prompts. Inspired by classifier-free guidance [20, 47], we introduce visual contrastive reasoning (VCR), a training-free visual marking method to help MLLM focus on specific regions, reducing hallucinations. The relevance of a region is assessed by observing MLLM output changes when key areas are excluded. It guides the MLLM to focus on areas with notable changes (bottom of Fig.3). Based on Eq. (1), we derive a probability distribution by comparing original image X with a modified image, $X' = \text{process}(X, \text{IM})$, where region IM is excluded.

$$\begin{aligned} y_t &\propto p_\theta(y_t \mid X, P, y_{<t}) \left(\frac{p_\theta(y_t \mid X, P, y_{<t})}{p_\theta(y_t \mid \text{process}(X, \text{IM}), P, y_{<t})}\right)^\alpha \\ &\sim \text{softmax}[(1+\alpha)\cdot \text{logit}_\theta(y_t \mid X, P, y_{<t}) - \alpha \cdot \text{logit}_\theta(y_t \mid \text{process}(X, \text{IM}), P, y_{<t})], \end{aligned} \tag{3}$$

where α is the level of focus on region IM. A higher α increases emphasis on that region. Following [54], we set $\alpha = 1$ in all tasks. It preserves the integrity of the original image pixels X, while constructing contrastive samples X' that encourage the model to focus on task-related regions. Ideally, X' should exclude task-related objects while maintaining a uniform appearance and overall context with the original image. But directly marking X' disrupts its pixels, making contrastive sample generation challenging.

Contrastive Sample Generation. To address it, we employ inpainting, where the mask M_{i-1} obtained from the previous iteration segmentation is treated as the inpainting mask IM_i to guide the creation of X'. We use a negative prompt P_n: "A_i^{fore}*, is a* P_g", to ensure that the inpainted X' does not contain potentially task-related objects A^{fore}_i. Additionally, we use a positive prompt P_p: "A_i^{back}*, high quality, detailed, blended to the original image.*", to ensure consistency between the generated portion and the surrounding background A^{back}_i. The corresponding inpainting is defined as:

$$X' = F_{in}(X, \text{IM}_i, P_p, P_n), \tag{4}$$

where F_{in} represents the inpainting module, and we choose Stable Diffusion to perform this operation. This method ensures the generated X' excludes task-related objects without disrupting the pixel continuity. In the first iteration, since IM_i does not yet exist, we use bounding box predictions from various patches $\mathcal{B}_i$ as an alternative. As X' contains only the background, comparing it with X eliminates co-occurrence hallucination caused by the background and highlights differences in task-related regions, subtly guiding the model to focus on these areas. Finally, we use visual contrastive reasoning to identify accurate instance-specific prompt to guide segmentation as follows,

$$B_i^u = \text{VCR}\left(X, X', C, P_B\right), \qquad A_i^u = \text{VCR}\left(X, X', C, P_A\right), \tag{5}$$

where VCR represents our visual contrastive reasoning, and C is the caption of the image. The collected knowledge, $\mathcal{A}i$ and $\mathcal{B}i$, is integrated into the prompt P_A and P_B. This process aids in identify the ultimate instance-specific names A_i^u and bounding boxes B_i^u of the objects.

3.2 Mask Generator

Until now, we described how to use SAM-generated masks as a visual marker to guide the model to focus on task-relevant areas for generating accurate instance-specific prompts. But this method relies on an assumption that the mask accurately delineates task-related regions. However, SAM is trained on large-scale prompt-mask pairs without category labels, it excels at identifying masks based on image textures but lacks label prediction capabilities. Consequently, the SAM-generated mask may not always align with task semantics, yet such alignment is crucial for our method.

3.2.1 Mask Semantic Alignment

We need to utilize texture generalization capabilities of SAM to describe possible task-related objects within the prompt-targeted areas, while also ensuring that the generated masks align with the task semantics. To achieve this, we divide the input image into patches of varying scales using horizontal, vertical, and uncut divisions as outlined in the last section. these processed patches are then reintegrated onto the original image with surrounding areas blacked out, and fed into SAM to ensure it focuses exclusively on the patch. Finally, masks generated from different patches are aggregated based on their relevance to task semantics, providing an accurate representation of task-related objects. The masks for each patch is generated as follows,

$$m_i^k = \text{SAM}(\text{Spatial CLIP}(A_i^u, X_i), B_i^u, X_i^k), \tag{6}$$

where mask m_i^k is obtained by inputting the corresponding image patch X_i^k and associated prompts into SAM during the $i-$th iteration. Following [21], Spatial CLIP maps the text prompt A_i^u to regions in the image X_i that correspond to the content of the prompt. The processed images, along with the generated instance-specific text prompts A_i^u, are then input into CLIP to assess semantic similarity.

$$s(m_i^k) = \text{CLIP}(m_i^k \odot X_i, A_i^u), \tag{7}$$

the operation $\odot$ results in retaining only those parts of X_i that are covered by the predicted mask. $s(m_i^k)$ represents the similarity between masked image and A_i^u, calculated using CLIP. The similarity scores obtained from different patches are denoted as $S_i = [s(m_i^1), s(m_i^2), \ldots, s(m_i^k)]$. After normalizing the elements within S_i, the closer the normalized $s(m_i^k)$ is to 1, the more semantically aligned m_i^k is with the instance-specific text prompt A_i^u. Finally, we compute the weighted sum of the normalized $s(m_i^k)$ and m_i^k as follows.

$$M_i = \sum_{k=1}^{K} (s(m_i^k) * m_i^k), \tag{8}$$

M_i is the output mask of the $i-$th iteration of X. The generated M_i leverages SAM's mask prediction capabilities to create highly detailed masks. Simultaneously, through the mask semantic alignment process, it ensures that the output mask aligns with the task's semantics, thereby overcoming the limitation of SAM's mask prediction lacking semantic understanding. The mask is applied to the original image as a weight, to generate the next iteration image X_i for segmentation. This excludes irrelevant regions to reduce interference during segmentation.

$$X_{i+1} = w \cdot (X_i \odot M_i) + (1 - w) \cdot X_i, \tag{9}$$

where w is a hyperparameter, which we have assigned a value of 0.3.

Table 1: Results on Camouflaged Object Detection (COD) under different settings. Best are in **bold**.

Methods	Venue	Camouflaged Object Detection											
		CHAMELEON [50]				CAMO [30]				COD10K [14]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
Scribble Supervision Setting													
WSSA[62]	CVPR20	0.067	0.692	0.860	0.782	0.118	0.615	0.786	0.696	0.071	0.536	0.770	0.684
SCWS[60]	AAAI21	0.053	0.758	0.881	0.792	0.102	0.658	0.795	0.713	0.055	0.602	0.805	0.710
TEL[62]	CVPR22	0.073	0.708	0.827	0.785	0.104	0.681	0.797	0.717	0.057	0.633	0.826	0.724
SCOD[17]	AAAI23	**0.046**	**0.791**	**0.897**	0.818	**0.092**	0.709	0.815	0.735	0.049	0.637	0.832	0.733
SAM-S[29]	ICCV23	0.076	0.729	0.820	0.650	0.105	0.682	0.774	0.731	0.046	0.695	0.828	0.772
WS-SAM[16]	NeurIPS23	**0.046**	0.777	**0.897**	**0.824**	**0.092**	**0.742**	**0.818**	**0.759**	**0.038**	**0.719**	**0.878**	**0.803**
Point Supervision Setting													
WSSA[62]	CVPR20	0.105	0.660	0.712	0.711	0.148	0.607	0.652	0.649	0.087	0.509	0.733	0.642
SCWS[60]	AAAI21	0.097	0.684	0.739	0.714	0.142	0.624	0.672	0.687	0.082	0.593	0.777	0.738
TEL[62]	CVPR22	0.094	0.712	0.751	0.746	0.133	0.662	0.674	0.645	0.063	0.623	0.803	0.727
SCOD[17]	AAAI23	0.092	0.688	0.746	0.725	0.137	0.629	0.688	0.663	0.060	0.607	0.802	0.711
SAM[29]	ICCV23	0.207	0.595	0.647	0.635	0.160	0.597	0.639	0.643	0.093	0.673	0.737	0.730
SAM-P[29]	ICCV23	0.101	0.696	0.745	0.697	0.123	0.649	0.693	0.677	0.069	0.694	0.796	0.765
WS-SAM[16]	NeurIPS23	**0.056**	**0.767**	**0.868**	**0.805**	**0.102**	**0.703**	**0.757**	**0.718**	**0.039**	**0.698**	**0.856**	**0.790**
Task-Generic Prompt Setting													
CLIP_Surgey+SAM	Arxiv23	0.147	0.606	0.741	0.689	0.189	0.520	0.692	0.612	0.173	0.488	0.698	0.629
GPT4V+SAM [43, 29]	Arxiv23	0.180	0.557	0.710	0.637	0.206	0.466	0.666	0.573	0.187	0.448	0.672	0.601
LLaVA1.5+SAM [37, 29]	NeurIPS23	0.168	0.561	0.718	0.666	0.314	0.401	0.585	0.501	0.170	0.530	0.728	0.662
X-Decoder [69]	CVPR23	0.124	0.654	0.748	0.716	0.104	0.628	0.745	0.709	0.171	0.556	0.705	0.652
SEEM [71]	NeurIPS23	0.094	0.011	0.307	0.454	0.192	0.023	0.315	0.404	0.143	0.001	0.280	0.425
GroundingSAM [29, 38]	ICCV23	0.122	0.662	0.776	0.744	0.157	0.656	0.753	0.707	0.085	0.670	0.813	0.764
GenSAM [21]	AAAI24	0.073	0.696	0.806	0.774	0.106	0.669	0.798	0.729	0.058	0.695	0.843	0.783
ProMaC	Ours	**0.044**	**0.790**	**0.899**	**0.833**	**0.090**	**0.725**	**0.846**	**0.767**	**0.042**	**0.716**	**0.876**	**0.805**

3.3 Mask Prompt Cycle Generation

The mask generated from the last iteration will guide the prompt generator in the next iteration to focus on potential task-related regions, eliminating the erroneous effects of irrelevant hallucinations and generating more accurate instance-specific prompts. These prompts, in turn, help the mask generator produce better masks. Through iterative prompt generation and mask generation jointly, we yield both better instance-specific prompts and visual masks. Finally, the masks from different iterations are averaged, and the mask closest to the mean is considered the final output.

$$\mathrm{i}^* = \arg\min_i \left(\left| M_i - \frac{\sum_i (M_1, \ldots, M_{\mathbf{I}})}{i_{\text{result}}} \right| \right). \tag{10}$$

Here, $\mathbf{I}$ is the number of adaptation epoches and M_{i^*} is the corresponding final mask for image X.

Table 2: Results for Medical Image Segmentation (MIS) under task-generic prompt setting.

Methods	Venue	Polyp Image Segmentation								Skin Lesion Segmentation			
		CVC-ColonDB [51]				Kvasir [25]				ISIC [10]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
GPT4V+SAM [43, 29]	Arxiv23	0.578	0.051	0.246	0.242	0.614	0.128	0.236	0.253	0.514	0.387	0.366	0.334
LLaVA1.5+SAM [37, 29]	NeruIPS23	0.491	0.194	0.355	0.357	0.479	0.293	0.400	0.403	0.369	0.473	0.497	0.477
X-Decoder [69]	CVPR23	0.462	0.095	0.327	0.331	0.449	0.202	0.371	0.384	0.338	0.315	0.127	0.407
SEEM [71]	NeruIPS23	0.570	0.085	0.280	0.284	0.520	0.215	0.339	0.367	0.362	0.250	0.002	0.280
GroundingSAM [29, 38]	ICCV23	0.711	0.071	0.195	0.206	0.387	0.353	0.521	0.468	0.301	0.348	0.247	0.533
GenSAM [21]	AAAI24	0.244	0.059	0.494	0.379	0.172	0.210	0.619	0.487	0.171	0.699	0.744	0.678
ProMaC	Ours	**0.176**	**0.243**	**0.583**	**0.530**	**0.166**	**0.394**	**0.726**	**0.573**	**0.168**	**0.717**	**0.755**	**0.689**

Table 3: Result on Transparent Object Segmentation and Open-Vocabulary Segmentation Tasks.

(a) Transparent Object Segmentation.

Methods	GSD [34]				Trans10K-hard [56]			
	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
GPT4V+SAM [43, 29]	0.312	0.104	0.392	0.363	0.288	0.199	0.607	0.512
LLaVA1.5+SAM [37, 29]	0.197	0.202	0.545	0.433	0.272	0.167	0.621	0.555
X-Decoder [69]	0.191	0.240	0.643	0.480	0.568	**0.611**	0.218	0.280
SEEM [71]	0.184	0.224	0.573	0.479	0.557	0.501	0.013	0.256
GroundingSAM [29, 38]	0.168	0.230	0.572	0.483	0.436	0.415	0.047	0.424
GenSAM [21]	0.155	0.394	0.700	0.559	0.263	0.489	0.612	0.536
ProMaC	**0.147**	**0.409**	**0.723**	**0.569**	**0.251**	0.509	**0.654**	**0.557**

(b) Open-vocabulary Segmentation.

Methods	Venue	Seg. Anno.	Image-Text pairs	VOC	Context	Object
				mIoU↑	mIoU↑	mIoU↑
MaskCLIP[67]	ECCV22	-	-	38.8	23.6	20.6
TCL [6]	CVPR23	-	CC3M [48], CC12M [7]	51.2	24.3	**30.4**
GroupViT [57]	CVPR22	-	CC12M [7], YFCC14M [53]	52.3	22.4	-
ViewCo [46]	ICLR23	-	CC12M [7], YFCC14M [53]	52.4	23.0	23.5
SegCLIP [39]	ICML23	COCO [35]	CC [48]	52.6	24.7	26.5
OVSegmentor [58]	CVPR23	-	CC12M [7]	53.8	20.4	25.1
ProMaC	Ours	-	-	**59.3**	**30.7**	25.2

4 Experimental Setup

Baselines. To evaluate our approach across various scenarios, we first assess the performance of ProMaC on challenging segmentation tasks, including Camouflaged Object Detection (COD), Medical

Table 4: Ablation Study on COD and MIS Tasks

Method's Variants					CHAMELEON [50]				CVC-ColobNB [51]			
MCoT	IVP	ITP	VCR	MSA	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
	✓	✓	✓	✓	0.052	0.764	0.885	0.816	0.187	0.214	0.570	0.513
✓		✓	✓	✓	0.080	0.720	0.833	0.757	0.260	0.123	0.466	0.425
✓	✓		✓	✓	0.089	0.685	0.823	0.756	0.177	0.233	0.556	0.524
✓	✓	✓		✓	0.061	0.769	0.893	0.815	0.311	0.152	0.460	0.424
✓	✓	✓	✓		0.054	0.740	0.884	0.798	**0.156**	0.220	0.565	0.517
✓	✓	✓	✓	✓	**0.044**	**0.790**	**0.899**	**0.833**	0.176	**0.243**	**0.583**	**0.530**

Table 5: VCR Result on SR task

Model	Indiv.	Pairs	Set of 4
CLIP ViT-L-14 [45]	26.1	1.5	0.0
CLIP RN50x64 [45]	26.2	2.0	0.0
FLAVA [49]	30.4	10.9	0.0
ViP-LLAVA-13B [5]	70.9	57.5	21.8
LLAVA-1.5-13B [36]	73.1	60.6	28.9
+ VCR (Ours)	**75.4**	**63.6**	**36.7**

Table 6: Parameter ablation study on COD10K [14].

(a) Number of iteration **I**.

I	cos↑	IoU↑	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
1	0.864	0.563	0.080	0.626	0.818	0.765
2	0.876	0.589	0.050	0.683	0.859	0.796
3	0.879	0.593	0.045	0.702	0.869	0.802
4	**0.882**	0.601	0.042	0.714	0.875	**0.804**
5	0.881	**0.602**	**0.041**	0.718	0.875	**0.804**
6	**0.882**	0.599	**0.041**	**0.721**	**0.876**	0.803

(b) Image preprocess strategy.

Scale	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
Original	0.075	0.535	0.750	0.662
Havel	0.069	0.579	0.775	0.689
Quarters	0.087	0.423	0.673	0.586
Original+Havel	**0.042**	**0.714**	**0.875**	**0.804**
Original +Havel+Quarters	0.049	0.702	0.867	0.796

(c) Visual marker strategy.

strategy	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
None	0.058	0.690	0.855	0.789
Bbox	0.065	0.682	0.836	0.766
VCD	0.047	0.705	0.863	0.793
Ours	**0.042**	**0.714**	**0.875**	**0.804**

Image Segmentation (MIS), and Transparent Object Detection (TOD). These tasks are areas where SAM struggle [26]. In the COD task, we compare ProMaC with weakly supervised segmentation methods [29, 62, 60, 62, 17, 17, 22, 24, 18]. Two supervision levels are used for comparison: scribble supervision, where main structures for the foreground and background are sketched during training, and point supervision, where separate points are provided for both foreground and background. In the task-generic prompt setting, we introduce a challenging scenario by only providing a task description as a generic prompt for segmentation. ProMaC integrates LLaVA1.5 [37] with SAM [29]. We also experiment on the MIS and PIS tasks to demonstrate the effectiveness of our method using task-generic prompts compared to previous methods. We assess GPT4V+SAM and LLaVA1.5+SAM in this setting to demonstrate that current MLLM models cannot address it well. We also compare ProMaC against current SOTA promptable segmentation methods to showcase its effectiveness. Next, we evaluate ProMaC on Open-Vocabulary Segmentation (OVS), and compare with leading methods [33, 70, 71, 38, 63, 66, 32, 19]. Our Visual Contrastive Reasoning (VCR) strategy is applicable to other tasks, especially those requiring complex spatial understanding. We used the What's Up spatial reasoning dataset [27] to evaluate how well VCR guides models to focus on task-relevant regions in Spatial Reasoning (SR) task. Our results are the average of three trials.

Metric. For evaluating the first three tasks, metrics Mean Absolute Error (M), adaptive F-measure (F_β) [40], mean E-measure (E_ϕ) [15], and structure measure (S_α) [13] are used. Lower M or higher values for F_β, E_ϕ, and S_α reflect better performance. Mean Intersection over Union (mIoU) and accuracy measure OVS and SR performance respectively, with higher values indicating better results.

PyTorch Implementation Details. For the MLLM models, we utilize LLaVA-1.5-13B for evaluation purposes. For CLIP, we adopt the CS-ViT-B/16 pretrained model. The inpainting model is stable-diffusion-2-inpainting. The task-generic prompts for the COD task is "camouflaged animal". The MIS task consists of two sub-tasks: polyp image segmentation and skin lesion segmentation, each with its own task-generic prompts, "polyp" and "skin lesion" respectively. For TOD task, prompt is fixed as "glass". All tasks are optimized using training-free test-time adaptation, with each task iterating for four epochs, except for the polyp image segmentation task, which undergoes six epochs. Since the second epoch, VCR also operates on different patches, ensuring that while the non-inpainted parts still gather information through hallucinations, the inpainted parts eliminate hallucinations to generate accurate candidate prompts. The promptable segmentation methods is the ViT-H/16 version of SAM. Our experiment is conducted on a single NVIDIA A100 GPU. More details are in appendix.

5 Results and Analysis

Results on COD Task. The COD tasks focus on finding animals that blend into their complex surroundings. We evaluated ProMaC on three representative datasets: CHAMELEON [50], CAMO [30], and COD10K [14]. As shown in Tab. 1, we compared ProMaC with others that utilize varying levels of supervision. Overall, methods with scribble supervision generally perform better than those with point supervision. Importantly, ProMaC only uses a single generic task prompt for the entire task and it stil outperforms all point-supervised methods. It also surpasses methods with scribble supervision on the CHAMELEON and CAMO datasets, and matches the top-performing scribble-supervised methods on COD10K. It demonstrates the superiority of ProMaC.

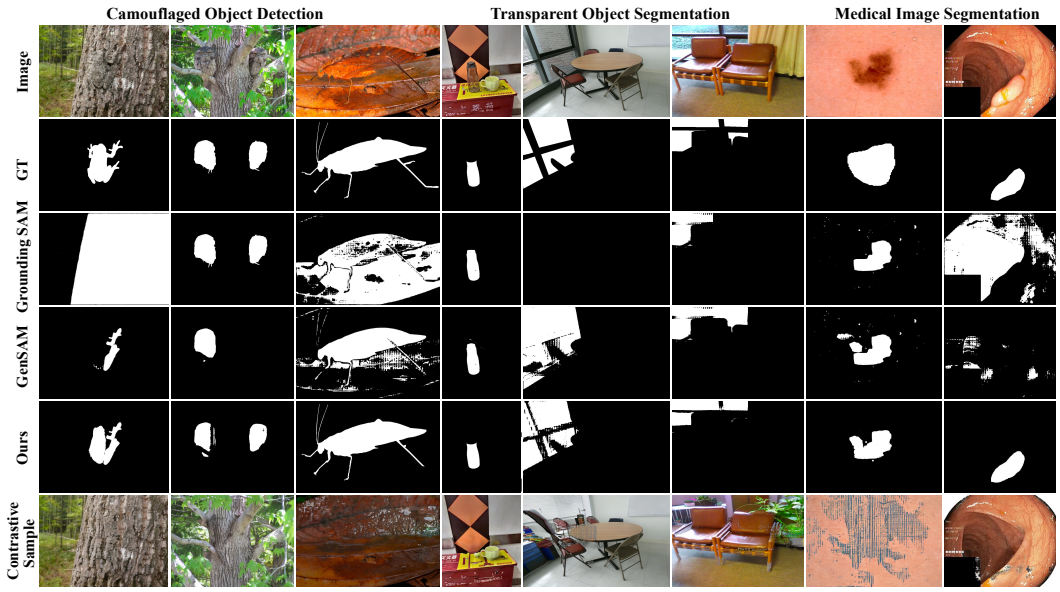


Figure 4: Visualization of various segmentation methods among various segmentation tasks.

Results on MIS and TOD Task. The MIS task identifies pathological tissues in medical images. We used three datasets: ColonDB [51] and Kvasir [25] for polyp image segmentation, and ISIC [10] for skin lesion segmentation. We compared our approach with others using task-generic prompt settings (see Tab. 2). While other models underperform in medical imaging due to limited generalization, ProMaC improves significantly over the baseline by iteratively mining task-related knowledge. For the TOD task, we evaluated ProMaC on the GSD [34] and Trans10K-hard [56] datasets (See Tab. 3(a)). Using the task-generic prompt setting, our method achieves the best results despite challenging scenarios. This demonstrates ProMaC's versatility and adaptability across complex visual tasks.

Results on OVS and SR Task. We evaluated ProMaC's effectiveness on the OVS task for multi-class segmentation based on a list of candidate classes. Specifically, we tested it on the validation splits of PASCAL VOC (21 classes) [12, 11], Pascal Context (59 classes) [42], and COCO-Object (80 classes) [3], using LLaVA to identify and confirm the presence of candidate classes. After obtaining masks, we resolved overlaps using the argmax operation based on SAM probabilities. Tab. 3(b) shows how ProMaC compares to other state-of-the-art OVS methods. Unlike some methods trained specifically on these datasets (risking knowledge leaking), ProMaC is not. Yet, ProMaC still outperforms all others on PASCAL VOC and Pascal Context and is competitive on COCO-Object. Additionally, as shown in Tab. 5, we integrated our VCR into LLaVA1.5 for enhanced spatial reasoning. This integration allows LLaVA to better focus on critical areas, thereby boosting performance.

Module Analysis. As shown in Tab. 4, we perform an ablation study on the COD and MIS tasks to assess the effects of different modules. "MCoT" is multi-scale chain of thought prompting. "ITP" and "IVP" refer to using only instance-specific text prompts or visual prompts. "VCR" is visual contrastive reasoning, and "MSA" is mask semantic alignment. The first row shows replacing MCoT with just one original image results in reduced performance, highlighting the importance of using hallucinations to extract task-relevant information. The second and third rows show that single modal prompts perform worse than multimodal prompts, highlighting the significance of multimodal prompting. Removing VCR causes a significant drop in performance, indicating that visual prompts are crucial for directing LLaVA's focus on relevant areas during inference. The comparison between the fifth and final rows emphasizes the importance of mask alignment with task semantics. The consistent positive results across tasks confirm the robustness and effectiveness of our approach.

Parameter Analysis. Tab. 6(a) examines how iterations influence performance. "cos" measures the cosine similarity between the predicted text prompt and the ground truth class through CLIP. "IoU" assesses the overlap between the predicted bounding box and the ground truth, comparing it against a rectangular outline of the mask. Mask predictions improve and stabilize after the fourth epoch. Tab. 6(b) investigates the effects of various image processing techniques. "Original" uses no modifications,

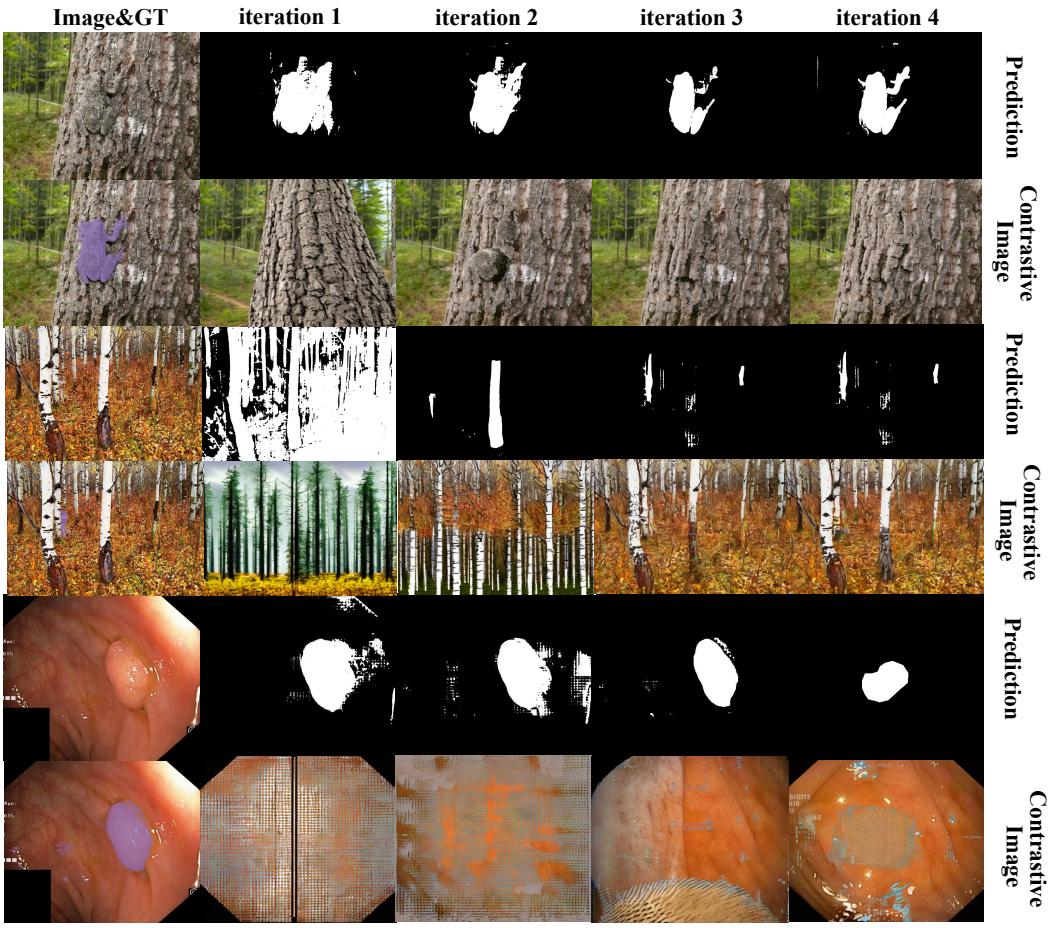


Figure 5: Visualization of the generated masks and contrastive samples over iterations.

"Halve" divides the image horizontally or vertically into halves, and "Quarters" divides it into four quarter-sized patches. Testing shows that combining "Original" and "Halve" yields the best results by balancing global and local information without excessive fragmentation.

Visual Marker Strategy. Tab. 6(c) assesses the impact of different visual marker strategies. "None" uses no visual prompts, while "Bbox" places bounding boxes directly on the image. "VCD" employs previous methods that introduce Gaussian noise into comparison images for contrastive reasoning. Results indicate that bounding boxes decrease performance, suggesting LLaVA struggles with this type of markers. Although VCD methods improve performance, they distort pixel data, making them less effective than our approach. Our VCR generates contrastive samples that focus on task-relevant areas without altering the image, reducing hallucinations and enhancing performance.

Visualization. Fig. 4 and Fig.5 visually compares our ProMaC with other methods across 3 tasks and also shows the contrastive images we generated. GenSAM handles clear objects well but struggles with complex background. Although GenSAM performs well in complex backgrounds, but struggles with challenging tasks. ProMaC delivers solid segmentation results across different tasks, and our contrastive images remove task-related regions while maintaining semantic and pixel consistency.

6 Conclusion

In this work, we introduce an iterative ProMaC that uses MLLM hallucinations to guide automatic prompt generation, significantly improving segmentation without training. This iterative approach aligns masks with task semantics, enhancing model performance. Testing on multiple benchmarks has demonstrated ProMaC's effectiveness in a wide range of complex segmentation tasks.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

[2] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[4] Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. Neural machine translation with monolingual translation memory. *arXiv preprint arXiv:2105.11269*, 2021.

[5] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. *arXiv preprint arXiv:2312.00784*, 2023.

[6] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11165–11174, 2023.

[7] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

[8] Junjie Chen, Li Niu, Liu Liu, and Liqing Zhang. Weak-shot fine-grained classification via similarity transfer. *Advances in Neural Information Processing Systems*, 34:7306–7318, 2021.

[9] Junjie Chen, Li Niu, Jianfu Zhang, Jianlou Si, Chen Qian, and Liqing Zhang. Amodal instance segmentation via prior-guided expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 313–321, 2023.

[10] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[11] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The pascal visual object classes challenge 2012 (voc2012) results. 2012 http://www. pascal-network. org/challenges. In *VOC/voc2012/workshop/index. html*, 2012.

[12] 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*, 88:303–338, 2010.

[13] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017.

[14] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):6024–6042, 2021.

[15] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *Scientia Sinica Informationis*, 6(6), 2021.

[16] Chunming He, Kai Li, Yachao Zhang, Guoxia Xu, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. *arXiv preprint arXiv:2305.11003*, 2023.

[17] Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson WH Lau. Weakly-supervised camouflaged object detection with scribble annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 781–789, 2023.

[18] He, Chunming and Li, Kai and Zhang, Yachao and Tang, Longxiang and Zhang, Yulun and Guo, Zhenhua and Li, Xiu. Camouflaged object detection with feature decomposition and edge reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22046–22055, 2023

[19] He, Chunming and Li, Kai and Zhang, Yachao and Zhang, Yulun and Guo, Zhenhua and Li, Xiu and Danelljan, Martin and Yu, Fisher. Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects. In *arXiv preprint arXiv:2308.03166*, 2023.

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[21] 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.

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

[23] 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.

[24] Jian Hu, Haowen Zhong, Fei Yang, Shaogang Gong, Guile Wu, and Junchi Yan. Learning unbiased transferability for domain adaptation by uncertainty modeling. In *European Conference on Computer Vision*, pages 223–241. Springer, 2022.

[25] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26*, pages 451–462. Springer, 2020.

[26] Wei Ji, Jingjing Li, Qi Bi, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications. *arXiv preprint arXiv:2304.05750*, 2023.

[27] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023.

[28] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023.

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[30] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Computer vision and image understanding*, 184:45–56, 2019.

[31] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *arXiv preprint arXiv:2311.16922*, 2023.

[32] Ang Li, Jian Hu, Ke Ding, Xiaolu Zhang, Jun Zhou, Yong He, and Xu Min. Uncertainty-based heterogeneous privileged knowledge distillation for recommendation system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2471–2475, 2023.

[33] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023.

[34] Jiaying Lin, Zebang He, and Rynson WH Lau. Rich context aggregation with reflection prior for glass surface detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13415–13424, 2021.

[35] 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.

[36] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

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

[39] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023.

[40] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2014.

[41] Shentong Mo and Yapeng Tian. Av-sam: Segment anything model meets audio-visual localization and segmentation. *arXiv:2305.01836*, 2023.

[42] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014.

[43] OpenAI. Gpt-4v: Enhancing gpt-4 for visual processing. 2024. Accessed: 2024-05-20.

[44] Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*, 2020.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[46] Pengzhen Ren, Changlin Li, Hang Xu, Yi Zhu, Guangrun Wang, Jianzhuang Liu, Xiaojun Chang, and Xiaodan Liang. Viewco: Discovering text-supervised segmentation masks via multi-view semantic consistency. *arXiv preprint arXiv:2302.10307*, 2023.

[47] Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*, 2023.

[48] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2556–2565, 2018.

[49] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022.

[50] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018.

[51] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2015.

[52] Lv Tang, Peng-Tao Jiang, Zhihao Shen, Hao Zhang, Jinwei Chen, and Bo Li. Generalization and hallucination of large vision-language models through a camouflaged lens. *arXiv preprint arXiv:2311.11273*, 2023.

[53] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[54] David Wan, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. Contrastive region guidance: Improving grounding in vision-language models without training. *arXiv preprint arXiv:2403.02325*, 2024.

[55] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[56] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 696–711. Springer, 2020.

[57] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.

[58] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2935–2944, 2023.

[59] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[60] Siyue Yu, Bingfeng Zhang, Jimin Xiao, and Eng Gee Lim. Structure-consistent weakly supervised salient object detection with local saliency coherence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3234–3242, 2021.

[61] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021.

[62] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12546–12555, 2020.

[63] Shizhao Zhang, Hongya Tuo, Jian Hu, and Zhongliang Jing. Domain adaptive yolo for one-stage cross-domain detection. In *Asian conference on machine learning*, pages 785–797. PMLR, 2021.

[64] Yi-Fan Zhang, Weichen Yu, Qingsong Wen, Xue Wang, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Debiasing large visual language models. *arXiv preprint arXiv:2403.05262*, 2024.

[65] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[66] Haowen Zhong, Chao Wang, Hongya Tuo, Jian Hu, Lingfeng Qiao, and Zhongliang Jing. Transfer learning based on joint feature matching and adversarial networks. *Journal of Shanghai Jiaotong University (Science)*, 24:699–705, 2019.

[67] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

[68] Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*, 2023.

[69] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127, 2023.

[70] Xueyan Zou*, Zi-Yi Dou*, Jianwei Yang*, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee*, and Jianfeng Gao*. Generalized decoding for pixel, image and language. 2022.

[71] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv:2304.06718*, 2023.

A Appendix

A.1 More Explanation on Task-generic Promptable Segmentation

Task-generic promptable segmentation, first introduced by [21], aims to solve a key challenge in promptable segmentation: the requirement for manual prompts, such as bounding boxes, scribbles, or points, for each sample within the same task. This method allows a model, like the Segment Anything Model (SAM), to segment samples based on these prompts. Task-generic promptable segmentation seeks to enable models to automatically infer the task-related objects in different images based on a general task description. This approach eliminates the need to manually annotate every task-related object for different images under the same task.Compared to previous methods, this approach only requires a single task description as a task-generic prompt for a batch of five annotated samples within the same task, significantly reducing the annotation burden and better aligning with practical needs. Although this setting presents greater challenges than past methods, our ProMaC system achieves excellent performance across a variety of tasks. It even surpasses the performance of state-of-the-art (SOTA) methods trained on weakly supervised datasets for camouflaged object detection, demonstrating the robustness and superior performance of our approach.

A.2 More Experiments on the Motivation

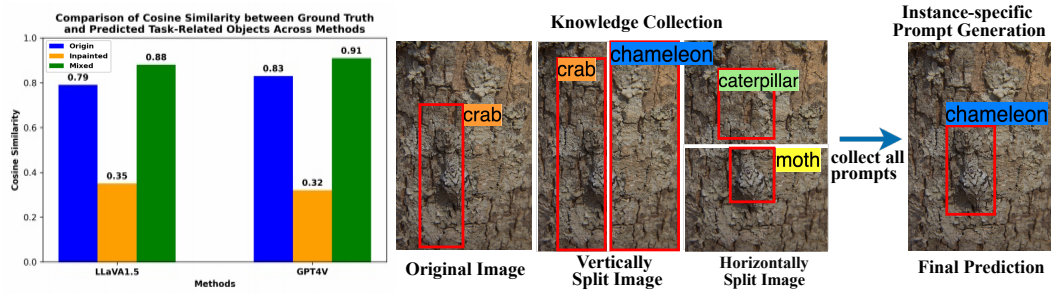


Figure 6: Left: In the bar chart, we analyze MLLM predictions with two versions of an image: the original (blue) and another with task-related objects removed via inpainting (orange). We then compare their predictions to the ground truth using CLIP similarity on the COD10K dataset. Despite missing key objects, the inpainted image's predictions still somewhat match the ground truth. When we select the higher similarity score from both images as the final score (green), it surpassed that of the original alone. It shows that prior knowledge from hallucinations can also provide useful information for generating prompts. Right: A example of using hallucinations to assist instance-specific prompt generation. Specifically, utilizing hallucination can leverage prior knowledge of image elements to better recognize and locate task-related objects. Directly inputting the image into LLaVA results in the hidden chameleon being incorrectly predicted. Splitting the image results in interested objects being incomplete or absent, prompting LLaVA to induce hallucinations and utilize prior knowledge to uncover potential task-related knowledge within the image. This knowledge assists in final accurately identifying and locating the chameleon.

A.3 Further Demonstration on Implement Details

We conducted experiments across multiple datasets and compared our method with various approaches across different tasks. Initially, we benchmarked against methods such as GPT4V+SAM and LLaVA1.5+SAM to demonstrate that even the current state-of-the-art (SOTA) MLLM methods combined with SAM cannot directly address this issue. Here, GPT4 utilized the gpt-4-vision-preview model, and LLaVA1.5 used the LLaVA-1.5-13B model, with SAM employing the ViT-H/16 version. Both models were tested in a single iteration setup, as our experiments showed that multiple iterations resulted in performance degradation as iterations increased. Specifically, these MLLMs were tasked with inferring instance-specific prompts to guide SAM in segmentation. Similarly, in our setup, we fine-tuned the source code of representative promptable segmentation methods such as SEEM, GroundingSAM, and X-Decoder to adapt them to our setting. These models were directly fed task-generic prompts to segment each image, with performance evaluated based on a single iteration of inference. This methodological adjustment allowed us to assess the efficacy of leveraging task-generic prompts in practical segmentation tasks.

Our ProMaC system employs LLaVA-1.5-13B and the ViT-H/16 version of SAM for inference, utilizing instance-specific prompts generated by the prompt generator. These prompts include instance-specific text prompts and instance-specific bounding boxes. The instance-specific text prompts are processed by Spatial CLIP, mapped to corresponding image regions, and along with the instance-specific bounding boxes, are input into SAM to guide segmentation. All tasks, except for the PIS task, underwent four epochs of iteration, while the PIS task was iterated over six epochs. In the Camouflaged Object Detection task, the task-generic prompt used was "camouflaged animal". We also compared our results with past weakly supervised learning methods, which require at least one point per image in the training set as supervision. Despite requiring less manual supervision with our task-generic promptable segmentation setting, our method achieved comparable or even better performance. Experiments were also conducted in Medical Image Segmentation and Transparent Object Segmentation tasks, with task-generic prompts "polyp" and "skin lesion" respectively for each task. Notably, the datasets used for the transparent object detection task were exclusively glass, allowing the instance-specific text prompt "glass" to be obtained directly without inference, requiring only the inference of bounding boxes through multi-scale chains of thought prompting. This streamlined approach emphasizes the efficiency and adaptability of our system across varied segmentation tasks.

Table 7: Results for Our Method under various task-generic prompt settings.

Methods	CHAMELEON [50]				CAMO [30]			
	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$
ProMaC	0.044±0.003	0.790±0.003	0.899±0.003	0.833±0.01	0.090±0.002	0.725±0.002	0.846±0.002	0.767±0.002
Methods	COD10K [14]				CVC-ColonDB [51]			
	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$
ProMaC	0.042± 0.002	0.716±0.002	0.876±0.003	0.805±0.002	0.176±0.002	0.243±0.002	0.583±0.003	0.530±0.002
Methods	Kvasir [25]				ISIC [10]			
	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$
ProMaC	0.166 ± 0.003	0.394± 0.002	0.726 ± 0.004	0.573 ± 0.003	0.160 ± 0.003	0.729 ± 0.004	0.766 ± 0.003	0.703 ± 0.004
Methods	GSD [34]				Trans10K-hard [56]			
	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$	$M \downarrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$S_\alpha \uparrow$
ProMaC	0.147 ± 0.002	0.409 ± 0.003	0.723± 0.004	0.569 ± 0.003	0.251 ± 0.003	0.509 ± 0.004	0.654 ± 0.002	0.557 ± 0.002

A.4 Further Experiments Results Analysis and Visualization

In Tab. 7, we present the variance-inclusive experimental results of our ProMaC framework across three major datasets, obtained under three different seeds, with detailed environment setup available in the code instructions provided in the supplementary materials. For the COD dataset, we used LLaVA1.5+SAM as our baseline model to ensure fairness, with the comparative GenSAM model utilizing the same configuration. The three datasets utilized involve challenging scenarios with camouflaged animals or people, where some task-related targets are obscured or very small. Under such complex conditions, ProMaC achieves performance similar to or even better than weakly-supervised training methods with only a brief task description like "camouflaged animal" through test-time adaptation, demonstrating our method's superiority. Similarly, on the MIS and TOD tasks, our method significantly outperforms the comparative promptable segmentation approaches. On the OVS task, we first use LLaVA1.5 to identify which categories from a given multi-class list are actually present in the image. The categories identified are then further reasoned through our ProMaC to derive the final results. What'sUp is a benchmark designed to evaluate the spatial understanding abilities of MLLMs. It comprises 820 images that depict clear spatial relationships between two household items, such as a chair and a bowl, each image exclusively features the two objects in one of four distinct spatial configurations. We regard ProMaC's VCR as a visual marker strategy integrated into LLaVA1.5 to guide spatial reasoning and compared with traditional methods. Compared to previous approaches and baselines, our method significantly enhances performance, also highlighting the efficacy of the VCR module.

Moreover, in Fig.5, we analyze how masks evolve across multiple tasks with iterations. It is evident that as iterations increase, the segmentation results improve, the boundaries of task-related objects become clearer, and some targets initially undetected are progressively recognized. Additionally, in Fig.7-12, we display masks for different tasks and samples of generated inpainting of task-related objects in contrastive samples. These results further demonstrate the versatility and superiority of our method.

A.5 Limitation

We compared ProMaC with the recently introduced GPT-4o+SAM [**?** 29] approach on the CHAMELEON and CAMO datasets for the COD task, and on the CVC-ColonDB dataset for the MIS task, with results displayed in the table below. It's evident that MLLMs based on GOT-4V [43] and LLaVA1.5 [37] excel in the COD task but experience a significant performance drop in the MIS task. This suggests that LLaVA and GPT4V lack specialized data like polyp detection, which leads to their underperformance in tasks requiring high specificity. This issue, stemming from the generalization limitations of MLLM datasets, is also evident in our ProMaC, based on LLaVA. While ProMaC achieves impressive results on the COD task due to LLaVA's general capabilities, its performance on the more specialized MIS dataset, though better than the baseline, falls short when compared to the COD task. The newly proposed GPT-4o, trained on a broader dataset, outperforms GPT-4V across various fields and shows significant improvement on the MIS task, further highlighting the impact of the underlying MLLM's generalization capabilities on ProMaC. This points to a need for further exploration and research into the generalization potential of foundational MLLM models in future work.

Table 8: Comparison with present SOTA MLLM approaches.

Methods	Venue	Camouflaged Object Detection								Polyp Image Segmentation			
		CHAMELEON [50]				CAMO [30]				CVC-ColonDB [51]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
GPT4V+SAM [43, 29]	Arxiv23	0.180	0.557	0.710	0.637	0.206	0.466	0.666	0.573	0.578	0.051	0.246	0.242
LLaVA1.5+SAM [37, 29]	NeruIPS23	0.168	0.561	0.718	0.666	0.314	0.401	0.585	0.501	0.491	0.194	0.355	0.357
GPT-4o+SAM [**?** 29]	ArXiv24	0.073	0.638	0.779	0.706	0.116	0.582	0.727	0.659	**0.067**	**0.340**	**0.655**	**0.575**
ProMaC	Ours	**0.044**	**0.790**	**0.899**	**0.833**	**0.090**	**0.725**	**0.846**	**0.767**	0.176	0.243	0.583	0.530

Table 9: Comparison with present SOTA MLLM approaches.

Methods	Venue	Polyp Image Segmentation			
		CVC-ColonDB [51]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
GPT-4o+SAM [**?** 29]	ArXiv24	0.067	0.340	0.655	0.575
Qwen-14B+SAM	ArXiv24	0.189	0.271	0.533	0.536
ProMaC(LLaVA)	Ours	0.767	0.176	0.243	0.583
ProMaC(Qwen)	Ours	0.104	0.289	0.601	0.583

Figure 7: Visualization of the generated mask and contrastive samples on CHAMELEON dataset.

A.6 Robustness of ProMaC

ProMaC uses MLLM to infer instance-specific prompts that guide the promptable segmentation model. A key question arises: how does the model ensure convergence to the accurate task-related region when initial instance-specific prompts from early iterations are imprecise? To address this, we utilize hallucinations to mine a potential list of instance-specific prompts from multiple scales and repeated image segmentations, ensuring comprehensive exploration of possible task-related objects.

Figure 8: Visualization of the generated masks and contrastive samples on CAMO dataset.

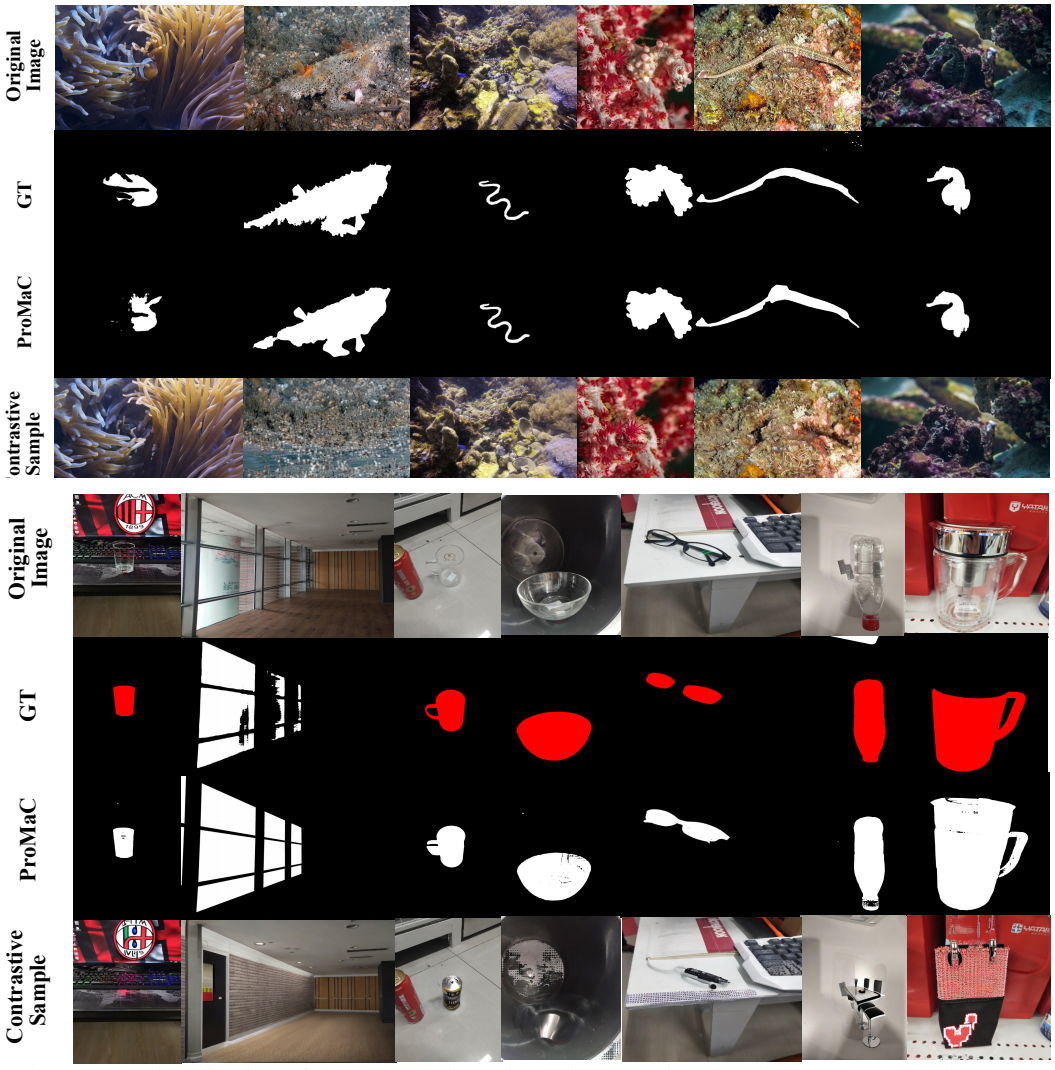


Figure 10: Visualization of the generated masks and contrastive samples on Trans10K dataset.

Figure 11: Visualization of the generated masks and contrastive samples on GSD dataset.

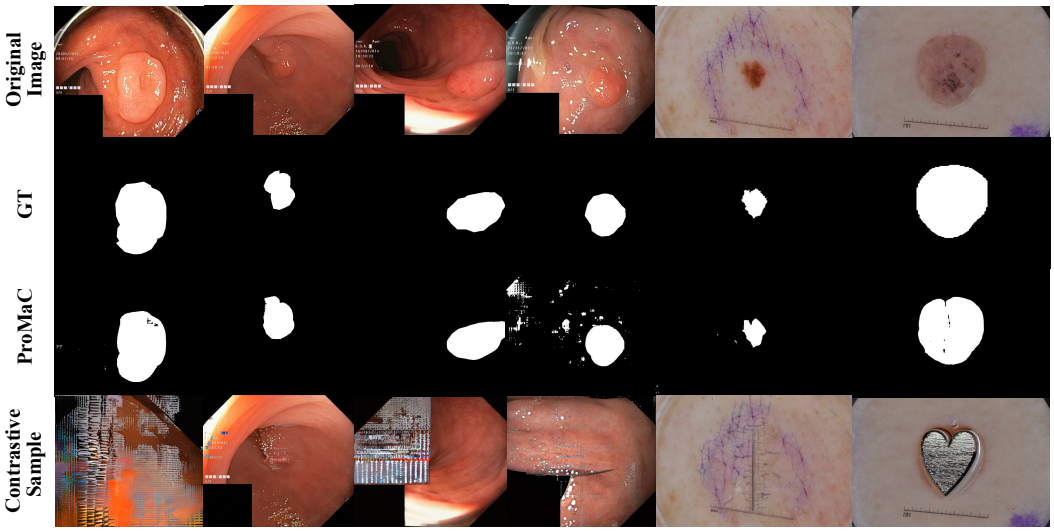


Figure 12: Visualization of the generated masks and contrastive samples on medical dataset.

Subsequently, VCR eliminates less accurate hallucinations to pinpoint the most precise instance-specific prompt from the candidate set. Additionally, our instance-specific prompts include both bounding boxes and keywords, enhancing fault tolerance through a multimodal setup. Experiments in Tab.4 demonstrate the robustness of our multi-modal prompts. Even if the generated prompts do not perfectly match the ground truth, semantic similarity often accurately locates objects of interest within images, facilitating gradual fine-tuning through successive iterations. Even if the initial iteration produces inaccurate prompts, the input for the next iteration remains the original image, preserving correct object information for potential correction in finer-grained iterations. In summary, by providing a wide range of candidates and employing a rigorous strategy for generating instance-specific prompts, along with multimodal prompt inputs and leveraging SAM's generalizability, ProMaC consistently produces accurate instance-specific prompts across various tasks.

A.7 Social impact

Our work is equivalent to general works in computer vision field, which aims at reducing the manual prompt dependency in promptable segmentation. Therefore, our work has similar potential societal impacts as previous works [21].

Algorithm 1 Algorithm of our ProMaC

procedure PROMAC(X, LLaVA1.5, SAM, P_g, P_A, P_B)
  Input: Sample $X \in \mathbb{R}^{H\times W\times 3}$; Multimodal Large Language Model LLaVA1.5, promptable segmentation model SAM, task-generic prompt P_g, keyword prompt P_A, bbox prompt P_B.
  Output: Segmentation result M_{i^*}.
  for iter $i = 1$ to **I do**
    Split image into K multi-scale patches.
    if $i == 1$ **then**
      for $k = 1$ to K **do**
        $C^k \leftarrow$ GENERATECAPTION(LLaVA1.5, X^k),
        $B^k \leftarrow$ GENERATEBBOX(LLaVA1.5, X^k, C^k, P_B),
        $A^k_{\text{fore}}, A^k_{\text{back}} \leftarrow$ GENERATEKEYWORD(LLaVA1.5, X^k, C^k, P_A),
      end for
    else if $i! = 1$ **then**
      for $k = 1$ to K **do**
        $C^k \leftarrow$ GENERATECAPTION(LLaVA1.5, X^k),
        $B^k \leftarrow$ GENERATEBBOX(VCR, X^k, X'^k, C^k, P_B),
        $A^k_{\text{fore}}, A^k_{\text{back}} \leftarrow$ GENERATEKEYWORD(VCR, X^k, X'^k, C^k, P_A),
      end for
    end if
    $\mathcal{A}_i \leftarrow \{A^1_{\text{fore}}, ..., A^K_{\text{fore}}\}, P_A = \{\mathcal{A}_i, P_A\}$,
    $\mathcal{B}_i \leftarrow \{B^1, ..., B^K\}, P_B = \{\mathcal{B}_i, P_B\}$,
    $X' \leftarrow$ INPAINTIMAGE(X, IM$_i$, P_p, P_n),
    $A^u_i, B^u_i \leftarrow$ FINALPROMPT GENERATION(X, X', P_A, P_B),
    $M_i \leftarrow$ MASKGENERATION(X_i, SAM, A^u_i, B^u_i),
    $X_{i+1} \leftarrow$ WEIGHTEDIMAGE(X_i, M_i),
  end for
  $M_{\text{i}^*} \leftarrow$ SELECTBESTMASK(M_i)
end procedure

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have claimed our main contributions accurately in abstract and introduction clearly. We also provide further explanations in appendix A.2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:We have claimed our limitation in the limitation section in appendix A.5.

   Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA] .

   Justification: We do not have theoretical proof.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our codes in the supplemental materials, we also provide implement details in both main text and the appendix. All of our experiment results are averaged with three trials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have released our codes and the instruction on how to run the codes in supplemental materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] .

   Justification: We have provide sufficient instructions on the settings and the details in the paper and appendix A.3 and A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] .

   Justification: We have illustrated it in experiment section and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: We gave clarified such things in implement details and instruction of the codes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes] .

   Justification: We have conducted the paper conform to make sure it follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] .

    Justification: We have discussed about the potential social impact in appendix A.7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

    Justification: We conduct experiments on public datasets, and our model is training-free apporaches, the model we used are all open-sourced model from the community.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] .

    Justification: we have cited the data and models we used in our paper properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA] .

    Justification: We do not introduce new datasets in the papers, and our codes will be released after publication.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA] .

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA] .

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.