INT: Instance-Specific Negative Mining for Task-Generic Promptable Segmentation

Jian Hu, Zixu Cheng, Shaogang Gong
Queen Mary University of London
{jian.hu,zixu.cheng, s.gong}@qmul.ac.uk

Abstract

Task-generic promptable segmentation aims to achieve segmentation of diverse samples under a single task description by utilizing only one task-generic prompt. Current methods primarily leverage the generalization capabilities of Vision-Language Models (VLMs) to infer instance-specific prompts from these task-generic prompts, which guide the segmentation process. However, when VLMs struggle with generalization for certain samples, accurately deriving instance-specific prompts becomes a challenge. To address this issue, we introduce **I**nstance-specific **N**egative Mining for **T**ask-Generic Promptable Segmentation (**INT**). The key idea of INT is to adaptively reduce the influence of irrelevant negative prior knowledge and use the most plausible prior knowledge, selected through negative mining with higher contrast, to refine the instance-specific prompts generation. Specifically, our approach consists of two components: instance-specific prompt generation, which progressive fliter out incorrect information for accurate instance-specific prompt generation, and semantic mask generation, which ensure the segmentation results accurately reflect the semantics of the instance-specific prompts. The effectiveness of our INT is validated on six datasets, including camouflaged objects and medical images, showing its robustness and applicability.

1 Introduction

Task-generic promptable segmentation leverages a task-generic prompt to derive instance-specific prompts for segmenting diverse images under a unified task. Unlike traditional segmentation approaches that require extensive labelled datasets, this method only uses a task-generic prompt applicable across all test samples, e.g. "the polyp" is a task-generic prompt for all images in a polyp segmentation task. This approach presents greater challenges due to the lack of labels, but it is more appealing for real-world applications.

A task-generic prompt, being both coarse and potentially ambiguous, can lead to poor segmentation when directly applied. Initial solutions [Hu *et al.*, 2024a; Liu *et al.*, 2023c] first proposed to use VLMs to mine information from images, deriving instance-specific prompts from a generic task prompt to guide SAM in segmenting task-related objects. However, they often struggle with complex samples where task-relevant objects are hard to distinguish. This can lead to inaccurate segmentation and degrade model performance. ProMaC [Hu *et al.*, 2024b] mitigated this issue by treating hallucinations as a form of prior knowledge. It masks the objects of interest to induce hallucinations, thereby extracting information that appears relevant to the task, aiding in the reasoning of instance-specific prompts. Although these methods have had some success, the lack of labels on samples makes it hard to check if the deduced prompts are accurate. As a result, once an incorrect prompt is generated, it becomes difficult to fix and the errors may increase over time. Yet the question of eliminating this need remains unexplored.

Despite the lack of manual annotation, the output changes of the VLMs show how they link candidate categories to the task. As shown in Fig. 1(a), a frog is hidden among branches

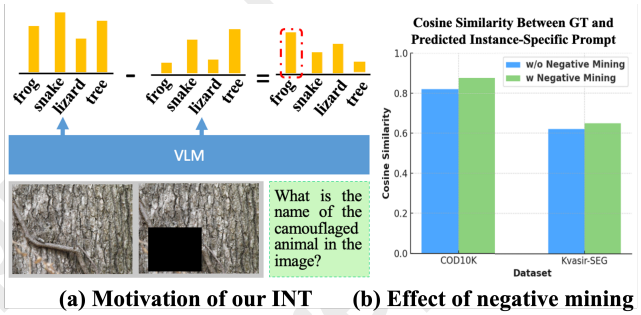

Figure 1: Motivation of our proposed InT. When task-related objects in the input to the VLM are occluded, the unique features of these objects are also obscured, leading to significant changes in the corresponding VLM output. In contrast, the features of other objects, which are not fully occluded, show only minor changes in the VLM output. We leverage this observation to assess the correctness of the generated instance-specific prompts without the need for ground truth, and through progressive negative mining, we gradually correct difficult-to-identify erroneous prompts. (b) Evaluation of Instance-Specific Prompts. We evaluate the CLIP semantic similarity between the instance-specific prompts we generate and the ground truth. Our contrastive negative mining approach effectively corrects erroneous samples, ensuring that the generated instance-specific prompts are more accurate.

that blend with the tree trunk, making it tough to spot. Using the VLM directly, even though the frog triggers a high activation, the tree trunk incorrectly scores higher for the 'snake' class, causing a misclassification of the branch as 'snake'. If we cover the likely area of the camouflaged animal, the identical feature identifying the 'frog' is hidden, but the 'snake' features remain visible on the tree trunk and bark. This leads to a significant decrease in the 'frog' score, while the 'snake' score is barely affected. This drastic alteration in the VLM response when key features are occluded provides valuable insight: we can assess the accuracy of prompts by observing changes in VLM responses when parts of an image are hidden. The element with greater variation is more likely to be the correct instance-specific prompt. Moreover, the difference between the correct category and easily mistaken ones in VLM outputs contrast is often small, leading to incorrect object identification when relying on a single output. In Figure 1(b), under the premise of including the ground truth mask, we randomly drew 2 masks and calculated the VLM changes before and after covering these masks, multiplying them by category to implement hard false class negative mining. Compared to using just one mask for covering, the instance-specific prompt inferred in this way has a higher CLIP semantic similarity to the ground truth class.

In this paper, building on this observation, we propose INT, which aims to gradually eliminate the impact of incorrect categories through progressive negative mining, utilizing changes in the VLM's output. It consists of instance-specific prompt generation and semantic mask generation. For the first component, the model splits the input image into different patches and processes them in parallel. In these patches, objects of interest might be fully or partially visible. This variation induces the model to use its pre-trained knowledge to guess the objects' names and locations within a patch. The names and locations of objects predicted for each patch are then sent to an image inpainting module, which erases the predicted objects and fills the space with the surrounding background. The outputs of the VLMs are compared before and after the image inpainting, and the predictions with the largest output difference are selected as the instance-specific prompts for this iteration. The normalized weights of these differences are multiplied with the VLM's output in the next iteration. This approach helps progressively improve classification of hard-to-distinguish categories and allows for correction of initially misclassified samples as the iterations continue. The semantic mask generation module uses an GroundingDINO [Liu *et al.*, 2023c] to task-related objects in the image. The detected bounding boxes and prompts are then processed by SAM and refined using semantic similarity with CLIP. Masks above the similarity threshold are combined to produce the segmentation for this iteration, and the corresponding soft mask is applied to the original image to enhance segmentation in the next iteration. **Out contributions are as follows:**

1). We introduce INT, a training-free test-time adaptation approach that uses progressive negative mining to identify more accurate instance-specific prompts in the absence of annotations, enabling accurate task-generic promptable segmentation.

2). Progressive negative mining identifies hard-to-distinguish error categories by cumulatively multiplying the changes in VLM outputs before and after masking across multiple iterations by category, thereby ensuring the accuracy of the generated instance-specific prompts.

3). Experiments on six datasets demonstrate the effectiveness of our method.

2 Related Works

Vision Language Models (VLMs) are adept at handling tasks in both vision and vision-language modalities. They encompass visual understanding models [Krizhevsky *et al.*, 2012; Radford *et al.*, 2021; Liu *et al.*, 2023a], visual generation models [Ramesh *et al.*, 2021; Rombach *et al.*, 2022], and general-purpose Interfaces [Zhang *et al.*, 2023a; Alayrac *et al.*, 2022]. Visual understanding models develop robust visual representations that serve as the backbone for various computer vision downstream tasks. Thanks to extensive image-text datasets, visual generation models are equipped to tackle a range of visual tasks, such as segmentation and object detection in images or videos [Kirillov *et al.*, 2023; OpenAI, 2024b], based on multimodal inputs. Although current models have succeeded in addressing many downstream visual tasks, they struggle with specific challenges such as medical image segmentation due to the scarcity of relevant data, making it difficult to achieve high performance in these demanding areas. Our INT introduces negative mining, employing VLMs to iteratively refine the correct prompts to enhance segmentation performance in these challenging tasks.

Promptable Segmentation involves segmenting objects with user-provided inputs, such as points, boxes, or scribbles. Models like SAM [Kirillov *et al.*, 2023], AV-SAM [Mo and Tian, 2023], GroundingSAM [Liu *et al.*, 2023c], and SEEM [Zou *et al.*, 2023b] extend this to multimodal inputs, including video and audio. However, these approaches often depend on manual prompts, which are prone to ambiguity and subjectivity, and are typically effective only for specific tasks. GenSAM [Hu *et al.*, 2024a] introduces a manual-free setting, using a task-generic prompt for instance-specific segmentation across images without additional user inputs. It leverages VLMs to infer object names as prompts for SAM, but its lack of spatial information can result in inaccurate predictions in complex scenes. ProMaC [Hu *et al.*, 2024b] introduced hallucination as a type of prior knowledge to leverage task-relevant information as much as possible to aid in the generation of more accurate instance-specific prompts. However, since there are no ground truths, how to effectively evaluate the quality of instance-specific prompts and exclude the effects of erroneous prompts becomes an unresolved problem.

Prompt Engineering is a developing field that focuses on creating and refining prompts to improve the effectiveness of large language models (LLMs) for a variety of tasks, encompassing both language and vision modifications. In the language domain, zero-shot prompting [Wei *et al.*, 2021] is employed to leverage LLMs' generalization capabilities for new tasks, although it can lead to inaccurate outcomes. Recent advancements include chain-of-thought prompting [Wei *et al.*, 2022] and graph prompting [Liu *et al.*, 2023d], which

enhance complex reasoning skills. Other strategies such as generated knowledge prompting [Liu *et al.*, 2021] and self-consistency [Wang *et al.*, 2022] have also been implemented to boost prediction accuracy. For vision-related tasks, prompt tuning is the primary method, using vision-driven [Jia *et al.*, 2022; Zhou *et al.*, 2023], language-driven [Zhou *et al.*, 2022; Ma *et al.*, 2023], and vision-language driven approaches [Zang *et al.*, 2022; Xing *et al.*, 2022] to create vision prompts that enhance model performance. Despite some successes in multimodal prompt engineering works [Zhang *et al.*, 2023b; Hu *et al.*, 2024a; Hu *et al.*, 2024b] in generating instance-specific prompts, these methods struggle to accurately identify correct prompts without annotations. Our INT approach effectively addresses this challenge by leveraging progressive negative mining.

3 Methodology

We present INT, a training-free cycle-generation method that segments multiple unknown classes of objects with only a single task-generic prompt. This innovative approach leverages negative mining to iteratively reduce the impact of potential erroneous predictions derived from task-generic prompts in an unlabelled setting. Specifically, for an image $X \in \mathbb{R}^{H\times W\times 3}$ from a test set, INT utilizes a task-generic prompt P_g to generate a final segmentation mask $M \in \mathbb{R}^{H\times W}$. This eliminates the need for separate supervision for each image, streamlining the process across datasets in the same task category. The prompt generator identifies multiple candidates for instance-specific prompts, which are evaluated by comparing the VLM outputs before and after image removal. The differences in the outputs are normalized and iteratively weighted, influencing the selection of instance-specific prompts in subsequent iterations. In each iteration, the candidate with the largest difference is chosen as the instance-specific prompt. This selected prompt guides the segmentation process, which further refines the generation of improved instance-specific prompts in the next iteration.

3.1 Instance-Specific Prompt Generation

Prompt Generation Using VLMs. For more accurate segmentation, the prompt utilizes VLMs to transform a generic prompt P_g into instance-specific prompts for each image. Specifically, given an image X and a query P, the VLM with parameters θ generates a response that captures task-relevant information. The image X provides essential visual context, aiding the model in formulating a relevant response y, which is derived auto-regressively from a probability distribution conditioned on P, X, and the previous tokens:

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

where y_t denotes the token at time t and $y_{<t}$ represents the sequence of tokens generated up to time $t-1$. Despite the advanced capabilities of VLMs, task-relevant objects may blend into their background due to factors like texture, color, or position, making the instance-specific prompts they generate prone to inaccuracies. Inaccurate instance-specific prompts can directly lead to incorrect segmentation. However, without any labels, assessing whether these prompts meet the task requirements becomes challenging. However, VLM predictions are often driven by the unique features of the objects they detect. When such features are obscured, the model's predictions for the corresponding category significantly drop (see Fig. 2). Therefore, if task-related objects can be accurately located and effectively removed, comparing the VLM outputs before and after removal provides a reliable method for evaluating the quality of the instance-specific prompts.

Hallucination-driven Candidates Generation. To generate accurate instance-specific prompts, it is essential to identify as many candidates as possible with a reasonable level of confidence to ensure that the correct prompt is not overlooked. Inspired by ProMaC [Hu *et al.*, 2024b], we divide the input image into patches of varying scales by cutting it horizontally, vertically, both, or leaving it uncut. Each patch is then processed individually by the VLM to generate preliminary instance-specific prompts. The visibility of task-relevant objects differs across patches, encouraging the VLM to leverage its prior knowledge to infer potential objects and their locations. This process helps identify candidate bounding boxes and object names by linking the visual data in each patch with the task context. The VLM processes each patch as follows:

$$B^k = \text{VLM}(X^k, C^k, P_B), \quad A^k_{\text{fore}}, A^k_{\text{back}} = \text{VLM}(X^k, C^k, P_A), \quad (2)$$

where C^k is the caption for the k-th image patch X^k, and P_g is the task-generic prompt. For bounding box predictions, the prompt P_B guides the VLM: "*This image pertains to the P_g detection task, output the bounding box of the P_g.*" This instructs the VLM to predict the bounding boxes B^k for objects related to the task within the patch. For object naming, the prompt P_A states: "*Output the name of the P_g and its environment in one word.*" This directs the VLM to predict the names of the task-related objects A^k_{fore} and their backgrounds A^k_{back} from each patch. Object names A^k_{fore} and bounding boxes B^k, collected from different patches, are compiled into candidate lists $\mathcal{A}_i$ and $\mathcal{B}_i$, where i denotes the iteration.

Prompts Selection with Negative Mining. After generating the candidate lists, we evaluate which candidates are most likely correct. When a prediction is accurate, removing the corresponding object causes a significant change in the VLM's output for that category. Building on the previous section, where we ensured the ground truth was included, we now mask candidate-indicated areas and measure the change in the VLM's output scores. To achieve this, inspired by contrastive decoding, we compare the VLM softmax output of the original unprocessed image patch X_k with that of the masked patch $X_k^{'}$ for each patch k. The category with the largest difference in output is selected as the final prediction for the corresponding patch. This approach ensures that the most significant and task-relevant object is accurately identified and used as the instance-specific prompt as follows:

$$D(y_i^k) = \max(\text{softmax}(\text{logit}_\theta(y_t \mid X_k, P, y_{<t})) - \text{softmax}(\text{logit}_\theta(y_t \mid X_k^{'}, P, y_{<t}))), \quad (3)$$

here, $X_k^{'}$ is processed by the image inpainting module, which uses the predicted mask m_i^k from the last iteration in Sec. 3.2 as the inpainting mask IM_i^k to guide the modification of the patch $X_k^{'}$. To ensure that $X_k^{'}$ remains free of task-related objects A_i^{fore}, we employ a negative prompt P_n: "A_i^{fore} *is not a*

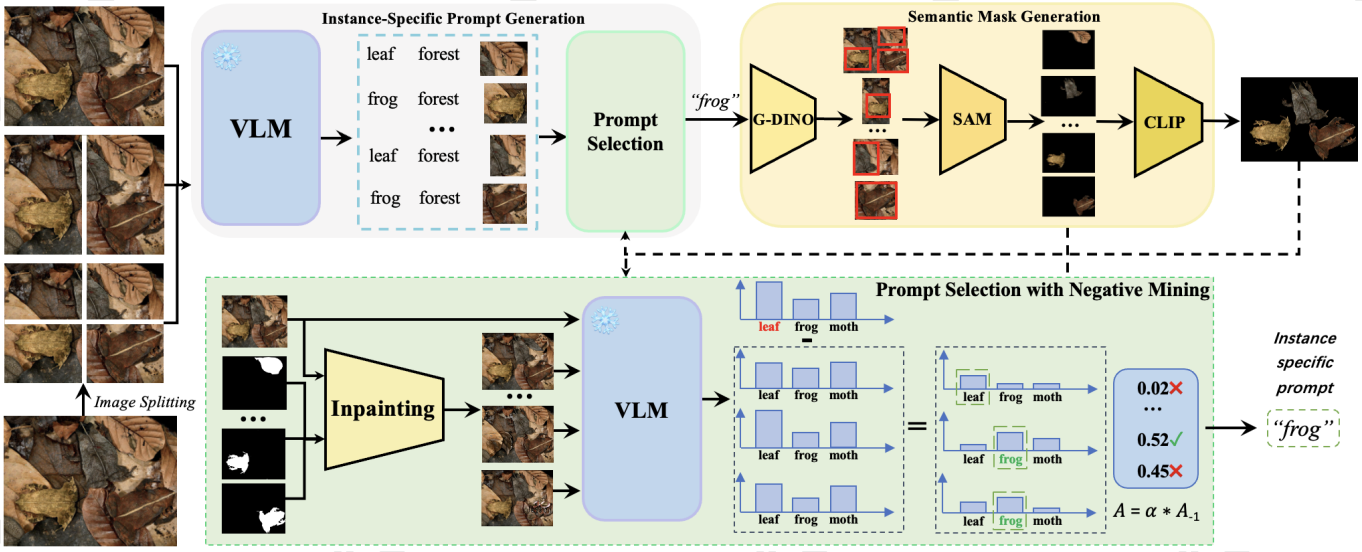

Figure 2: INT consists of two main components: instance-specific prompt generation and semantic mask generation. Initially, the former uses VLMs to generate candidate instance-specific prompts. A prompt selection module then selects the prompt with the highest VLM output contrast, refined through progressive negative mining. This selected prompt is passed to the semantic mask generation module, which employs GroundingDINO to ensure that all task-relevant samples in the image are collected as comprehensively as possible. Simultaneously, SAM and CLIP work together to ensure that the generated masks are semantically aligned with the task.

P_g." For seamless Integration with the existing background A_i^{back}, a positive prompt P_p is used: "*A_i^{back}, high quality, detailed, and well-Integrated with the original image.*" $X_k^{'}$ is formalized as:

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

where F_{in} represents the inpainting module, implemented using Stable Diffusion. Using this method, we construct a set $D(y) = \{D(y_i^k)\}_{k=1}^K$, where K represents the number of patch. The accuracy of the predicted category is directly related to the magnitude of change in unique information about the correct category before and after masking. By evaluating these changes, we identify the most reasonable foreground A_i^u to serve as the instance-specific prompt for the i-th iteration. Specifically, we compare the variations in $D(y_i^k)$ across patches and select the candidate with the largest change in output as the final prediction:

$$A_i^u = \text{argmax}_k D(y_i^k), \quad k = 1, 2, \ldots, K, \tag{5}$$

This approach leverages the variations in softmax outputs to refine the instance-specific prompt A_i^u, ensuring they are closely aligned with task-relevant information and enhancing the overall prediction accuracy.

Progressive Negative Mining While this method effectively identifies correct categories when task-related objects are easily distinguishable, it struggles with more ambiguous samples. In such cases, incorrect categories may occasionally exhibit large score differences before and after masking during certain iterations but show little to no change in others. In contrast, the correct category consistently demonstrates stable and significant differences across all iterations. Building on this observation, we design a progressive negative mining technique to iteratively refine A_i^u. The idea is to accumulate and reinforce consistent patterns by iteratively multiplying scores from each iteration, reducing the influence of unstable changes caused by incorrect categories. To normalize the differences from each iteration for comparability, we use:

$$D_{\text{norm}}(y_i^k) = \frac{D(y_i^k)}{\sum_{k=1}^K D(y_i^k)}, \tag{6}$$

where $D_{\text{norm}}(y_i^k)$ represents the normalized difference for category y^k at iteration i, and K is the total number of patches. We then iteratively update the differences for the next iteration by applying cumulative multiplication:

$$D(y_{i+1}^k) = D(y_{i+1}^k) \cdot D_{\text{norm}}(y_i^k), \tag{7}$$

where $D(y_{i+1}^k)$ is the updated difference for category y^k in iteration $i+1$, incorporating the influence of the current iteration's normalized differences. By iteratively applying negative mining, we amplify consistent patterns in the score differences associated with the correct category while suppressing the sporadic and unstable changes caused by incorrect categories, effectively enhancing the model's ability to accurately refine A_i^u over successive iterations.

3.2 Semantic Mask Generation

After obtaining the instance-specific prompt A_i^u, we aim to produce a mask that accurately delineates task-related objects without overlooking any targets. To achieve this, we first process A_i^u through Grounding DINO [Liu *et al.*, 2023c] to gather all potential bounding boxes across various patches:

$$B_i^k = \text{GroundingDINO}(X^k, A_i^u) \tag{8}$$

Subsequently, both B_i^k and A_i^u are input into SAM:

$$m_i^k = \text{SAM}(\text{Spatial CLIP}(A_i^u, X_i^k), B_i^k, X_i^k), \tag{9}$$

where Spatial CLIP [Hu *et al.*, 2024a] maps the text prompt A_i^u to regions within the image X_i as the visual prompts. These visual prompts, along with the corresponding bounding box B_i^k are fed into SAM during the i-th iteration to generate the mask m_i^k. These processed masks, along with the generated instance-specific text prompts A_i^u, are subsequently input into CLIP to evaluate semantic similarity.

$$s(m_i^k) = \text{CLIP}(m_i^k \odot X_i, A_i^u), \tag{10}$$

where the operation $\odot$ retains only those parts of X_i covered by the predicted mask. $s(m_i^k)$ quantifies the similarity between the masked image and A_i^u. Similarity scores from various patches are represented as $S_i = [s(m_i^1), s(m_i^2), \ldots, s(m_i^k)]$. After normalizing these elements within S_i, a normalized $s(m_i^k)$ closer to 1 indicates a higher semantic alignment of m_i^k with the instance-specific text prompt A_i^u. The weighted sum of the normalized $s(m_i^k)$ and m_i^k is then computed as follows:

$$M_i = \sum_{k=1}^{K} (s(m_i^k) * m_i^k), \tag{11}$$

where M_i is the resultant mask from the i-th iteration of X. This mask, generated using SAM's capabilities, ensures highly detailed mask production. Concurrently, this mask semantic alignment process guarantees that the output mask is consistent with the task's semantic requirements, overcoming the limitations of SAM's mask prediction.

The mask is then applied to the original image as a weighting factor to generate the next iteration image X_i for segmentation. This helps to exclude irrelevant regions and reduce Interference during segmentation:

$$X_{i+1} = w \cdot (X_i \odot M_i) + (1 - w) \cdot X_i, \tag{12}$$

where w is a hyperparameter set to 0.3. The mask generated in the last iteration serves to guide the prompt generator in the subsequent iteration, focusing on potential task-related regions, mitigating the impact of irrelevant hallucinations, and yielding more precise instance-specific prompts. These prompts, in return, aid the mask generator in producing improved masks. Through iterative cycles of prompt and mask generation, both elements enhance significantly. Ultimately, masks from different iterations are averaged, and the mask closest to this mean is selected as the final output:

$$\mathrm{i}^* = \arg\min_i \left(\left| M_i - \frac{\sum_i (M_1, \ldots, M_{\mathbf{I}})}{i_{\text{result}}} \right| \right). \tag{13}$$

Here, $\mathbf{I}$ represents the number of adaptation epochs, and M_{i^*} is the definitive mask for image X.

4 Experiments

4.1 Experimental Setup

Baselines. Our study evaluates the ProMaC model's effectiveness in handling complex segmentation challenges such as Camouflaged Object Detection (COD), Medical Image Segmentation (MIS), and Transparent Object Detection (TOD), where traditional SAM models often fall short [Ji *et al.*, 2023]. In our COD assessments, ProMaC is benchmarked against various weakly supervised segmentation methods [Kirillov *et al.*, 2023; Zhang *et al.*, 2020; Yu *et al.*, 2021; Zhang *et al.*, 2020; He *et al.*, 2023c; He *et al.*, 2023c; Hu *et al.*, 2019; Hu *et al.*, 2022; He *et al.*, 2023a]. We employ two levels of supervision for comparison: scribble supervision, where the main structures of foreground and background are delineated during training, and point supervision, where distinct points are annotated for both. For task-generic prompt settings, we introduce a demanding scenario by relying solely on a task description as a generic prompt for segmentation. Here,INT incorporates LLaVA1.5 [Liu *et al.*, 2023b] with SAM [Kirillov *et al.*, 2023]. Further experimentation in MIS and PIS tasks aims to validate our method's superiority using task-generic prompts versus traditional techniques. We test combinations like GPT4V+SAM and LLaVA1.5+SAM to highlight the limitations of current VLM models in this context. Our INT is also evaluated with leading state-of-the-art promptable segmentation methods to underline its effectiveness. Our results are the average of three trials.

Metric. To assess performance in the first three tasks, we employ the following metrics: Mean Absolute Error (M), adaptive F-measure (F_β) [Margolin *et al.*, 2014], mean E-measure (E_ϕ) [Fan *et al.*, 2021b], and structure measure (S_α) [Fan *et al.*, 2017]. A lower M or higher F_β, E_ϕ and S_α indicate superior performance.

PyTorch Implementation Details. For the VLM models, we employ LLaVA-1.5-13B for evaluation. For image processing, we use the CS-ViT-B/16 model pre-trained with CLIP, and for image inpainting module, we deploy stable-diffusion-2-inpainting. The task-generic prompts for the COD task are specified as "camouflaged animal." The MIS task includes two sub-tasks: polyp image segmentation and skin lesion segmentation, each prompted by "polyp" and "skin lesion" respectively. All tasks undergo training-free test-time adaptation, iterating for four epochs, except for the polyp image segmentation task, which extends to six epochs. We utilize the ViT-H/16 model for promptable segmentation methods. Our experiments are conducted on a single NVIDIA A100 GPU, with further details provided in the appendix.

4.2 Results and Analysis

Results on COD Task. The COD tasks are designed to detect animals camouflaged within complex environments. We tested INT on three benchmark datasets: CHAMELEON [Skurowski *et al.*, 2018], CAMO [Le *et al.*, 2019], and COD10K [Fan *et al.*, 2021a]. The CHAMELEON dataset includes 76 images collected from the Internet specifically for testing purposes. The CAMO dataset contains 1,250 images, divided into 1,000 training images and 250 testing images. The COD10K dataset comprises 3,040 training samples and 2,026 testing samples in total. As indicated in Table 1, we compared INT against other methods that apply different supervision levels. Generally, methods with scribble supervision outperformed those with point supervision. Notably, our INT, utilizing only a single generic task prompt, outshines all point-supervised methods and scribble-supervised methods on all three datasets in terms of all the metrics. This highlights

Methods	Venue	Camouflaged Object Detection											
		CHAMELEON [2018]				CAMO [2019]				COD10K [2021a]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
Scribble Supervision Setting													
WSSA[Zhang *et al.*, 2020]	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[Yu *et al.*, 2021]	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[Zhang *et al.*, 2020]	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[He *et al.*, 2023c]	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[Kirillov *et al.*, 2023]	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[He *et al.*, 2023b]	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[Zhang *et al.*, 2020]	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[Yu *et al.*, 2021]	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[Zhang *et al.*, 2020]	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[He *et al.*, 2023c]	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[Kirillov *et al.*, 2023]	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[Kirillov *et al.*, 2023]	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[He *et al.*, 2023b]	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 [OpenAI, 2024a; Kirillov *et al.*, 2023]	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 [Liu *et al.*, 2023b; Kirillov *et al.*, 2023]	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 [Zou *et al.*, 2023a]	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 [Zou *et al.*, 2023b]	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 [Kirillov *et al.*, 2023; Liu *et al.*, 2023c]	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 [Hu *et al.*, 2024a]	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[Hu *et al.*, 2024b]	NeurIPS24	0.044	0.790	0.899	0.833	0.090	0.725	0.846	0.767	0.042	0.716	0.876	0.805
INT	Ours	**0.039**	**0.801**	**0.906**	**0.842**	**0.086**	**0.734**	**0.853**	**0.772**	**0.037**	**0.722**	**0.883**	**0.808**

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

Methods	Venue	Polyp Image Segmentation								Skin Lesion Segmentation			
		CVC-ColonDB [2015]				Kvasir [2020]				ISIC [2019]			
		$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
GPT4V+SAM [OpenAI, 2024a; Kirillov *et al.*, 2023]	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 [Liu *et al.*, 2023b; Kirillov *et al.*, 2023]	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 [Zou *et al.*, 2023a]	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 [Zou *et al.*, 2023b]	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 [Kirillov *et al.*, 2023; Liu *et al.*, 2023c]	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 [Hu *et al.*, 2024a]	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 [Hu *et al.*, 2024b]	NeurIPS24	0.176	0.243	0.583	0.530	0.166	0.394	0.726	0.573	0.160	0.728	0.766	0.703
INT	Ours	**0.172**	**0.250**	**0.589**	**0.537**	**0.161**	**0.401**	**0.732**	**0.5739**	**0.152**	**0.733**	**0.771**	**0.708**

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

I	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
1	0.079	0.692	0.861	0.794
2	0.053	0.748	0.897	0.816
3	0.050	0.752	0.902	0.823
4	0.045	0.792	0.903	0.829
5	0.039	**0.801**	**0.906**	0.842
6	**0.038**	0.800	**0.906**	**0.844**

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

Scale	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
Original	0.071	0.541	0.758	0.668
Havel	0.059	0.582	0.780	0.692
Quarters	0.062	0.482	0.685	0.601
Original+Havel	0.043	0.793	0.891	0.835
Original +Havel+Quarters	**0.039**	**0.801**	**0.906**	**0.842**

(b) Image preprocess strategy.

Method's Variants				CHAMELEON [2018]			
HCG	PSNM	PNM	SMG	$M\downarrow$	$F_\beta\uparrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$
	✓	✓	✓	0.071	0.541	0.758	0.668
✓		✓	✓	0.083	0.703	0.811	0.746
✓	✓		✓	0.060	0.732	0.836	0.768
✓	✓	✓		0.053	0.772	0.895	0.818
✓	✓	✓	✓	**0.039**	**0.801**	**0.906**	**0.842**

(c) module ablation study.

Table 3: Ablation study on CHAMELEON dataset.

INT's effectiveness. Additionally, our method consistently surpasses SAM, SAM-P, SAM-S, and CLIP Surgery+SAM, demonstrating that the enhancements provided by our approach are derived from its Intrinsic qualities and not merely from leveraging the advanced segmentation capabilities of SAM.

Results on MIS Tasks. The MIS task involves identifying pathological tissues in medical images. We utilized datasets such as CVC-ColonDB [Tajbakhsh *et al.*, 2015] and Kvasir [Jha *et al.*, 2020] for polyp image segmentation, and ISIC [Codella *et al.*, 2019] for skin lesion segmentation. Comparisons from Table 2 show that we conducted experiments in the task-generic promptable segmentation setting. Since most VLMs have not been specifically trained on medical images, directly applying a VLM to medical tasks results in significantly lower performance compared to natural image tasks. In contrast, our proposed INT, through extensive candidate sample inference and negative mining, effectively explores task-relevant information and progressively filters out incorrect samples. This ensures the accuracy of the generated instance-specific prompts, greatly enhancing INT's segmentation performance on the MIS task.

Parameter Analysis. Tab. 3(a) investigates how various model metrics evolve over a prolonged number of iterations. all the metrics stabilize after just 5 iterations. Although the results continue to change, the variations remain slight. This indicates that our method converges rapidly and remains stable. It also underscores the importance of using iteration-based termination criteria to establish early stopping conditions. Tab. 3(b) examines the effects of different image pro-

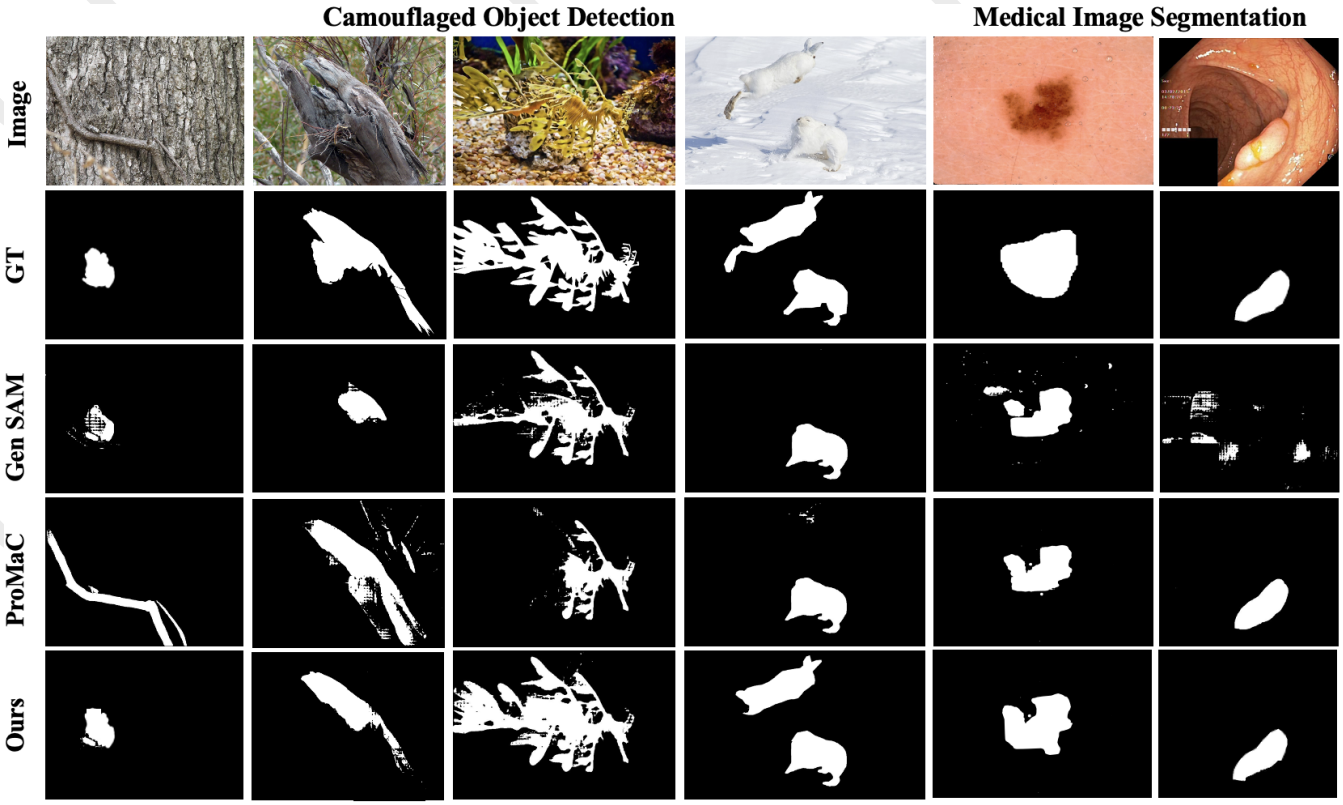

Figure 3: Visualization of various segmentation methods among various segmentation tasks.

cessing strategies. "Original" refers to the unmodified image, "Halve" splits the image horizontally or vertically into two parts, and "Quarters" divides it into four smaller patches. Testing results indicate that combining "Original", "Halve" and "Quarters" achieves the best balance between global and local information, avoiding excessive fragmentation.

Module Analysis. As shown in Tab. 3(c), we perform an ablation study on the COD and MIS tasks to assess the effects of different modules. "HCG" refers to hallucination-driven candidate generation, "PSNM" stands for prompt selection with negative mining, "PNM" is progressive negative mining, and "SMG" refers to the semantic mask generator. The first row shows that replacing HCG with just a single original image leads to reduced performance, underscoring the importance of using hallucinations to extract task-relevant information. In the second row, replacing prompt selection with negative mining using the VLM inference result yields worse performance than the full model, highlighting the significance of proper prompt selection. Removing progressive negative mining results in a significant drop in performance, indicating that the instance-specific prompts derived from the initial iteration may contain errors, and our approach effectively corrects these mistakes. The comparison between the last two rows emphasizes the importance of aligning the mask with task semantics. The consistently positive results across tasks confirm the robustness and effectiveness of our approach.

Visualization. Fig. 3 visually compares our method, nNam-Nim, with other approaches across three tasks and also showcases the contrastive images we generated. GenSAM performs well with clear objects but struggles in complex backgrounds. ProMaC produces solid segmentation results across various tasks, but it sometimes misidentifies instance-specific prompts in complex scenes, leading to segmentation results that are unrelated to the task (e.g., the first column in Fig. 3). In contrast, our method introduces a negative mining strategy that not only explores potential candidates to extract task-relevant information from the image for better segmentation but also corrects misidentified instance-specific prompts from early iterations. This approach effectively improves performance. Additionally, our method can segment multiple task-related samples within the same image, something that previous methods could not achieve. It demonstrated the effectiveness of our approach.

5 Conclusion

In this paper, we introduced the Instance-specific Negative Mining for Promptable Segmentation (INT) approach. It leverages the difference in VLM outputs before and after masking as a metric for progressive negative mining. By employing progressive negative mining, INT infers reliable instance-specific prompts from a single task-generic prompt. This allows for effective segmentation of different targets within the same task across various images, even in the absence of annotations. Experiments conducted on six diverse datasets demonstrate the effectiveness of our INT.

References

[Alayrac *et al.*, 2022] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[Codella *et al.*, 2019] 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.

[Fan *et al.*, 2017] 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.

[Fan *et al.*, 2021a] 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.

[Fan *et al.*, 2021b] 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.

[He *et al.*, 2023a] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. 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.

[He *et al.*, 2023b] 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.

[He *et al.*, 2023c] 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.

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

[Hu *et al.*, 2022] 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.

[Hu *et al.*, 2024a] 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.

[Hu *et al.*, 2024b] Jian Hu, Jiayi Lin, Junchi Yan, and Shaogang Gong. Leveraging hallucinations to reduce manual prompt dependency in promptable segmentation. *arXiv preprint arXiv:2408.15205*, 2024.

[Jha *et al.*, 2020] 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.

[Ji *et al.*, 2023] 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.

[Jia *et al.*, 2022] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[Kirillov *et al.*, 2023] 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.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[Le *et al.*, 2019] 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.

[Liu *et al.*, 2021] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*, 2021.

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

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

[Liu *et al.*, 2023c] 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.

[Liu *et al.*, 2023d] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and

downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 417–428, 2023.

[Ma *et al.*, 2023] Chengcheng Ma, Yang Liu, Jiankang Deng, Lingxi Xie, Weiming Dong, and Changsheng Xu. Understanding and mitigating overfitting in prompt tuning for vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[Margolin *et al.*, 2014] 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.

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

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

[OpenAI, 2024b] OpenAI. Hello gpt-4o. 2024. Accessed: 2024-05-20.

[Radford *et al.*, 2021] 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.

[Ramesh *et al.*, 2021] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

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

[Tajbakhsh *et al.*, 2015] 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.

[Wang *et al.*, 2022] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[Wei *et al.*, 2021] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[Xing *et al.*, 2022] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022.

[Yu *et al.*, 2021] 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.

[Zang *et al.*, 2022] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.

[Zhang *et al.*, 2020] 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.

[Zhang *et al.*, 2023a] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[Zhang *et al.*, 2023b] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[Zhou *et al.*, 2022] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[Zhou *et al.*, 2023] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11175–11185, 2023.

[Zou *et al.*, 2023a] 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.

[Zou *et al.*, 2023b] 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.