Training-Free Open-Ended Object Detection and Segmentation via Attention as Prompts

Zhiwei Lin   **Yongtao Wang***   **Zhi Tang**
Wangxuan Institute of Computer Technology, Peking University, China
{zwlin, wyt, tangzhi}@pku.edu.cn

Abstract

Existing perception models achieve great success by learning from large amounts of labeled data, but they still struggle with open-world scenarios. To alleviate this issue, researchers introduce open-set perception tasks to detect or segment unseen objects in the training set. However, these models require predefined object categories as inputs during inference, which are not available in real-world scenarios. Recently, researchers pose a new and more practical problem, *i.e.*, open-ended object detection, which discovers unseen objects without any object categories as inputs. In this paper, we present VL-SAM, a training-free framework that combines the generalized object recognition model (*i.e.,* Vision-Language Model) with the generalized object localization model (*i.e.,* Segment-Anything Model), to address the open-ended object detection and segmentation task. Without additional training, we connect these two generalized models with attention maps as the prompts. Specifically, we design an attention map generation module by employing head aggregation and a regularized attention flow to aggregate and propagate attention maps across all heads and layers in VLM, yielding high-quality attention maps. Then, we iteratively sample positive and negative points from the attention maps with a prompt generation module and send the sampled points to SAM to segment corresponding objects. Experimental results on the long-tail instance segmentation dataset (LVIS) show that our method surpasses the previous open-ended method on the object detection task and can provide additional instance segmentation masks. Besides, VL-SAM achieves favorable performance on the corner case object detection dataset (CODA), demonstrating the effectiveness of VL-SAM in real-world applications. Moreover, VL-SAM exhibits good model generalization that can incorporate various VLMs and SAMs.

1   Introduction

Deep learning has achieved remarkable success in perception tasks, with autonomous driving as a typical practical application. Existing deep learning based perception models rely on extensive labeled training data to learn to recognize and locate objects. However, training data cannot cover all types of objects in real-world scenarios. When faced with out-of-distribution objects, existing perception models may fail to recognize and locate objects, which can lead to severe safety issues [24].

Many open-world perception methods [15, 48] are proposed to address this issue. Open-world perception tries to give precise results in dynamic and unpredictable environments, which contain novel objects and involve scene domain shifting. Current open-world perception methods can be roughly divided into two categories: *open-set* and *open-ended*. Open-set methods [52, 43, 6] often calculate the similarity between image regions and category names with a pretrained CLIP [35] model. Thus, they require predefined object categories as inputs for the CLIP text encoder during
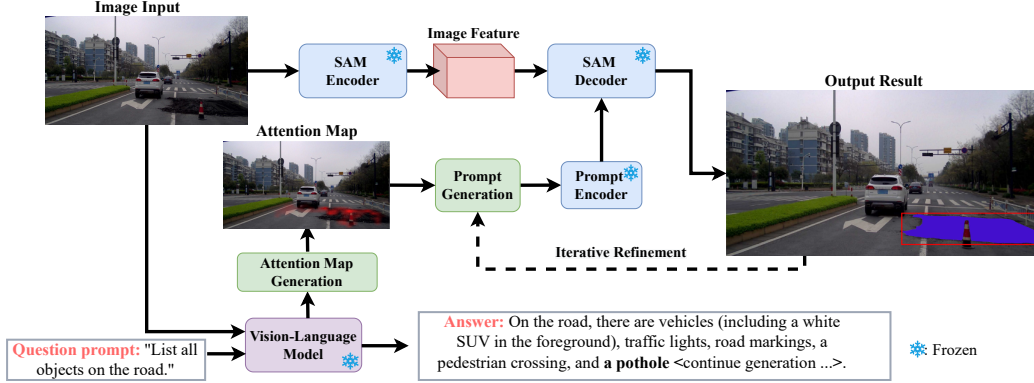
---
*Corresponding author

Figure 1: **Illustration of VL-SAM.** Without additional training, we connect the vision-language and segment-anything models with attention maps as the intermediate prompts.

inference. However, in many real-world application scenarios, we do not have the exact predefined object categories. For instance, in autonomous driving, self-driving vehicles may meet unexpected objects, including various rare animals. Besides, some objects cannot be presented by a simple category name, such as a human in an animal costume, which may look like an animal but is actually a human. Some methods use generic obstacle detection to handle unknown objects. However, many things do not have a significant 3D shape, like pits or grains on the ground. Thus, open-set methods cannot handle all situations. In contrast, open-ended methods [26, 48] are more general and practical since they can predict the object categories and locations themselves.

In a separate line of research, large vision-language models (VLMs) [29, 23, 55] show a strong generalized ability to recognize objects, *e.g.*, it can recognize rare objects for corner cases in autonomous driving scenarios [44]. However, VLM's localization ability is less accurate than that of specific perception models [51], sometimes missing objects or giving wrong localization results. On the other hand, as a pure vision model, segment-anything model (SAM) [20] exhibits good generalized segmentation capabilities for images from many different domains. However, SAM is unable to provide categories for segmented objects [49] and may yield numerous irrelevant segmentation results.

In this paper, we propose to combine the existing generalized object recognition model, *i.e.*, VLM, with the generalized object localization model, *i.e.*, SAM, to address the open-ended object detection and segmentation task. We present VL-SAM, a training-free framework that connects two generalized models with attention maps as the intermediate prompts, as illustrated in Figure 1. Specifically, we utilize the attention maps generated by VLM when describing the whole driving scene to prompt the segmentation of SAM. Firstly, given the generated token of VLM, we use the token as the query to obtain the attention maps from all layers and heads of VLM. Then, in the attention map generation module, we introduce the head aggregation and attention flow mechanism to aggregate and propagate global attention maps through all heads and layers. Besides, to alleviate the collapse problem caused by causal masks when propagating with attention flow, we adopt a regularization term to constrain the attention flow propagation process. After that, to better guide SAM to segment with the attention maps, we present a prompt generation module by grouping and sampling positive and negative points as the point prompts for SAM. Furthermore, to reduce the number of missing objects, we further use the segmentation results from SAM to sample positive and negative points from attention maps iteratively until convergence.

The main contributions of this work are summarized as follows:

- We present VL-SAM, a training-free open-ended object detection and segmentation framework that connects the generalized object recognition model and the generalized object localization model with attention maps as the prompts.

- We introduce a head aggregation and regularized attention flow mechanism to aggregate and propagate attention maps with the causal masks through all heads and layers.

- We propose an iterative refinement pipeline with a positive and negative point sampling strategy for attention maps.

2

- VL-SAM outperforms the *open-ended* method GenerateU and obtains competitive results compared with existing *open-set* methods on the long-tail instance segmentation dataset LVIS [14]. In autonomous driving applications, VL-SAM achieves favorable corner case object detection performance on the CODA [24].

2 Related work

2.1 Vision Language Model

Large language models (LLMs), including GPT-3 [3], GLM [11], and LLaMA [40], have shown human-like dialogue and reasoning skills. However, the limitation of LLM's ability to process and understand visual data restricts its application to more real scenarios. To overcome this, a cutting-edge Vision-Language Model (VLM) is introduced to open up new vistas for application. Recently, BLIP-2 [23] proposes Q-Former to connect and fuse image and text embeddings with three alignment pretrain losses. LLaMA-Adapter [53, 12], LLaVA [29], and MiniGPT [55] introduce an adapter or projection layer to align the embedding space from image and text. CogVLM [41] presents visual expert modules to transform the image features to align with text features in different transformer heads. SPHINX [28] utilizes several mixing techniques for multiple visual tasks. Furthermore, CogAgent [17] and LLaVA-Phi [57] view VLM as an agent or assistant to complete various tasks.

Existing VLMs, especially GPT-4V [2], exhibit strong generalization capability for understanding and reasoning new or rare situations, *e.g.,* it can deal with corner cases for autonomous driving [44]. However, the localization ability of VLMs is weaker than specific perception models, like SAM.

In this paper, we equip VLM with generalized segmentation models, *i.e.*, SAM, to address the localization limitation of VLM for open-ended object detection and segmentation. We achieve this by connecting two models with attention maps as the prompts without additional training.

2.2 Open-World Object Detection and Segmentation

With the advent of the CLIP models [35], open-world classification, object detection, and instance segmentation have made great progress at the same time. Open-world methods try to discover and recognize unseen objects in the training set during inference. Current open-world methods can be roughly classified into two types: *open-set* [37] and *open-ended* [26]. Open-set methods require redefined object categories, including seen objects and unseen objects in the training set, as inputs during inference. By contrast, open-ended methods can locate seen and unseen objects and generate their names simultaneously, as the current VLM does. In real-world applications, the exact categories may remain unknown for the perception models. For instance, in autonomous driving, self-driving vehicles often encounter unknown objects on the road, including overturned cars and construction vehicles with various shapes. Thus, the open-ended problem is more general and practical.

Open-Set Methods. With the powerful text-image embedding matching with CLIP, current open-set object detection methods mainly use a proposal network to obtain foreground object bounding boxes and embeddings, and then use CLIP as the open-set classification module to predict their categories. More recently, GLIP [25] proposes to use phrase grounding to pre-train open-world object detectors. GroundingDINO [30] presents cross-modality fusions to introduce text information to the image encoder for object grounding. SWORD [45] designs a novel contrastive method to learn the discrimination between foreground and background for instance segmentation. YOLO-World [7] introduces a prompt-then-detect paradigm for real-time open-world object detection. However, the above methods require predefined object categories as inputs for the text encoder.

Open-Ended Methods. GenerateU [26] first proposes the open-ended problem. Concurrently, DetCLIPv3 [48] introduces a similar concept with open-ended. They present a generative framework with language models to generate object categories and bounding boxes at the same time. To achieve better generalization capabilities, they construct a large dataset with bounding box and caption pairs and finetune the whole network on the constructed dataset.

In contrast, we propose a training-free open-ended framework, VL-SAM, that combines generalized recognition and segmentation models. VL-SAM can generate object categories with the generalized recognized model and then localize objects with the generalized segmentation models.
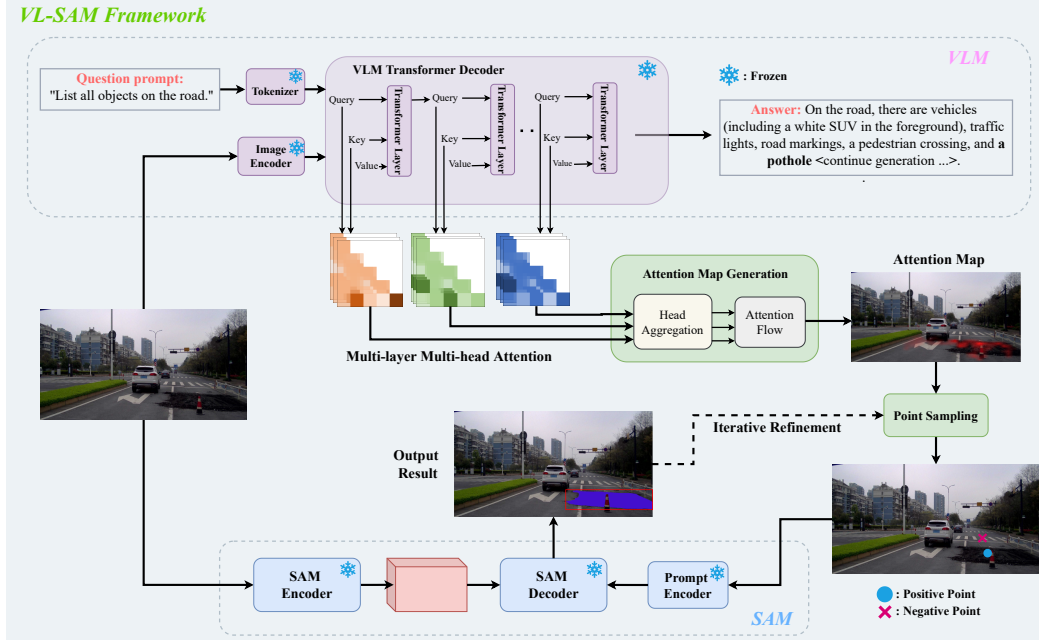
Figure 2: **An overview of VL-SAM framework.** We first use VLM to describe the input image and generate all possible objects' names. Then, for each object name, we obtain the corresponding attention map with the attention map generation module. Finally, we sample point prompts from the attention map and send them to SAM to predict detection and segmentation results.

3 Method

As shown 2, we provide an overview of our proposed framework. We use VLM and SAM as the generalized object recognition model and object localization model, respectively. Given an image input, we first use VLM to describe the scene and list all possible objects in the image. Then, for each object, we use the attention generation module with head aggregation and attention flow to obtain the high-quality attention map from VLM. Finally, we generate point prompts from the attention map and send them to SAM to get the location prediction iteratively.

3.1 Preliminary

Segment Anything Model. SAM is a prompt-based segmentation model with excellent data generation capability. It consists of three components: an image encoder, a mask decoder, and a prompt encoder. SAM takes an image and a set of prompts, including points, a box, and a mask, as the inputs. To segment objects with the prompts, SAM first extracts image features with the image encoder. Concurrently, the set of prompts is sent to the prompt encoder to transform into the prompt tokens. Then, the image features, prompt tokens, and mask tokens interact in the mask decoder with the two-way transformers. Finally, the mask tokens are transformed into multi-scale segmentation masks by multiplying mask tokens with the image features following MaskDINO [22].

Auto-Regressive Based Vision-Language Model. Current Auto-Regressive based VLMs have yielded surprising performance in various vision-language tasks. The mainstream framework of current VLMs comprises four parts, *i.e.*, an image encoder, a text tokenizer, projection layers, and a language decoder. Given an image and text as inputs, VLMs extract image tokens and text tokens with the image encoder and text tokenizer, respectively. Then, the image tokens are aligned with text tokens with projection layers. After that, the tokens from two modals are concatenated and sent to the language decoder to generate text outputs. The language decoder adopts the next-token prediction paradigm that the probability of the current generated token x_t depends on all previous tokens $(x_1, x_2, ..., x_{t-1})$.
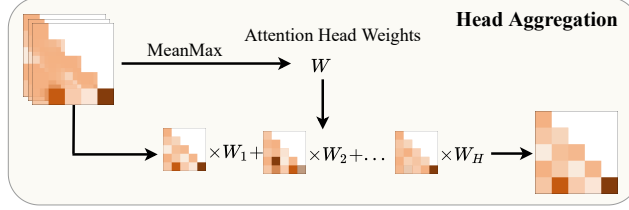
Figure 3: **Head aggregation.** We aggregate information from all attention heads with head weights.

Figure 4: **Attention flow.** We propagate attention from the first layer to last layer with attention flow.

Figure 5: **Illustration of attention collapse.** For each column, from left to right, we show image inputs, attention flow (collapse), regularized attention flow, and generated answers from VLM.

3.2 Attention Map Generation

The main idea of VL-SAM is to use attention maps of objects as the prompts for SAM to segment. Thus, how to generate a high-quality attention map for an object is critical. To achieve this, we introduce attention flow to aggregate and propagate attention maps through all transformer heads and layers in VLM.

Specifically, given an image input, we ask VLM to give all possible objects in the image. During this process, we cache all queries and keys from VLM. Then, we multiply queries and keys with causal masks and SoftMax normalization to obtain similarity matrix $S \in N \times N \times H \times L$, where N is the length of queries and keys, H is the number of transformer heads, and L denotes the number of transformer layers. $S_{i,j}^{h,l}$ represents the similarity between query i and key j in the head h, layer l. After that, we aggregate information from all transformer heads with mean-max attention head weights, as shown in Figure 3. In particular, we choose the maximum similarity weights of matrix S in dimension j and average them in dimension i to obtain the attention head weights $W \in 1 \times 1 \times H \times L$:

$$W = \texttt{Mean}(\texttt{Max}(S,\ dim=1),\ dim=0). \tag{1}$$

Obviously, the attention head weight indicates the importance of each head in each layer. Then, we pointwise multiply attention head weight W with similarity matrix S and average all heads as follows:

$$S' = \texttt{Mean}(S \odot W,\ dim=2). \tag{2}$$

After aggregating all information from all heads, we present attention flow to further aggregate attention from all layers, as illustrated in Figure 4. Concretely, we use the attention rollout method [1]

to compute the attentions from layer $l-1$ to layer l as follows:

$$\bar{S}'^{l}_{i,j} = \sum_{k=1}^{N}(I_{i,k} + S'^{l}_{i,k}) \times (I_{k,j} + \bar{S}'^{l-1}_{k,j}), \tag{3}$$

where I is the identity matrix. After the attention rollout, we only need the attention map from the last layer. To obtain the image attention map of the generated token, we select the corresponding line and columns from $\bar{S}'^{L}$.

However, since VLM uses causal masks for auto-regressive generation, simply adopting the attention rollout method causes attention collapse, as shown in Figure 5. Fortunately, we find a simple regularization term that can alleviate this problem efficiently. Specifically, for each column, assuming the unmasked length is L_0, we multiply each value in this column with $1 - (L_0 - 1)/L$. With this regularization term, the attention value in the top left corner will be constrained.

3.3 SAM Prompt Generation

The attention map generated in Section 3.2 has some unstable false positive peaks. To filter these false positive areas, we first use a threshold to filter weak activated areas and find the maximum connectivity area as the positive area [5]. The remaining area serves as a negative area. After that, we sample a positive point from the positive area with the maximum activated value and a negative point from the negative area with the weakest activated value. The positive and negative points serve as the point prompt pair for SAM.

3.4 Iterative Refinement

The segmentation results from the SAM decoder may include rough edges and background noises. We adopt two iterative strategies to further refine the segmentation results. In the first iterative strategy, we follow the cascaded post-refinement in PerSAM [54] to take the initial segmentation masks generated with the positive and negative pairs as the additional prompt input for the SAM decoder. For the second iterative strategy, we use the segmentation masks in the first iterative strategy to mask the attention map $\bar{S}'$. Then, we iteratively generate positive and negative pairs with Prompt Generation in Section 3.3 from the masked attention map and send them to the SAM decoder. Finally, we aggregate the results with NMS [13].

3.5 Multi-scale Ensemble

Due to the low-resolution image input of the image encoder in VLM, VLM may fail to recognize small objects. For instance, it may generate an answer: '*On the road, there are vehicles (a red truck and a blue bus), road signs, a pedestrian crossing, a white barrier, and **a few other smaller objects that are not clearly identifiable from the image.***'. To alleviate this issue, we follow SPHINX [28] to split an image ($H \times W$) into four sub-images ($H/2 \times W/2$) from the four corners and send each sub-image to VL-SAM independently. Finally, we ensemble the output of VL-SAM for four sub-images and the whole image.

3.6 Question-prompt Ensemble

The output of VLM is sensitive to the input prompt. To obtain a more comprehensive description of the input image, we ask VLM to generate ten question prompts for scene description with the sentence:'*If we want you to list all possible objects in the given image, what questions should we ask? Please give 10 questions you prefer.*' Then, we use the generated question prompts for VL-SAM to segment objects and ensemble the outputs of all question prompts.

4 Experiments

4.1 Implementation Details

We chose CogVLM-17B [41] with EVA2-CLIP-E [39] and Vicuna-7B-v1.5 [8] as the vision-language model. CogVLM-17B divides an image with 490×490 into 35×35 patches. We set the temperature

Table 1: **Comparison of object detection and segmentation results on LVIS minival.** 'Open-Ended' denotes that we do not have exact object categories during inference [26]. We report *fixed* AP [9] for rare objects. * denotes using the external data.

Method	Type	Training	LVIS	
			box AP_{rare}	mask AP_{rare}
Mask R-CNN [16]	Close-Set	✓	26.3	25.1
Deformable DETR [56]		✓	24.2	-
GLIP [25]		✓	20.8	-
GroundingDINO [30]		✓	27.4	-
DetCLIP [47]	Open-Set	✓	26.9	-
YOLOWorld [7]		✓	27.1	-
OWLv2* [32]		✓	39.0	-
GenerateU [26]	Open-Ended	✓	20.0	-
VL-SAM (Ours)		×	23.4	22.7

to 0.8 and top-p for nucleus sampling to 0.1 for CogVLM-17B. For the generated localization model, we use SAM with ViT-Huge [10].

We evaluate VL-SAM in a *training-free zero-shot* manner for all datasets. To obtain object categories from the generated sentence of VLM, we follow Tag2Text [18] to parse tags from the given sentence. To evaluate the open-ended performance on datasets with predefined object category names, we follow GenerateU [26] to adopt CLIP [35] text encoder and map the generated object categories to predefined categories in datasets for evaluation. Specifically, we use the text prompt 'a {object category}' for CLIP text encoder to calculate the similarity between generated object categories and predefined categories for mapping. All models are inferred on an 80G A800 machine.

4.2 Main Results

LVIS Dataset. We evaluate VL-SAM on the LVIS dataset [14], which has a long tail of categories and annotations for over 1000 object categories. Following previous works [26, 7], we mainly evaluate VL-SAM on LVIS minival and report the fixed AP [9] for rare objects.

As shown in Table 1, we list the performance for three types of perception methods, *i.e.,* close-set, open-set [15], and open-ended. The different between open-set and open-ended is that open-set requires exact prior knowledge of object categories as inputs, while open-ended can generate them during inference in a zero-shot manner [26]. In a real scenario, we often do not have predefined object categories for a scene. Thus, open-ended methods are more general and practical. As we can see, VL-SAM outperforms GenerateU by 3.4 AP_{rare}. Notably, VL-SAM is a training-free framework and can simultaneously obtain boxes and segmentation masks. In contrast, GenerateU needs to fine-tune both the image encoder and language model on VG [21] and GRIT [33] datasets, requiring significant training costs, and can only predict bounding boxes. Besides, VL-SAM achieves competitive detection and segmentation performance compared to open-set detection methods and close-set segmentation methods, respectively.

CODA Dataset. To further demonstrate the effectiveness of the proposed method in the real-world application, we present the results of VL-SAM on corner case object detection dataset CODA for autonomous driving in Table 2. Specifically, as we can see, RPN only achieves 10.6 mAR, indicating that current open-set detectors relying on object proposals have difficulty dealing with corner cases. For more recent open-set detectors, they achieve higher mAR with CLIP as the object category predictor. For the open-ended method, LLaVA-Grounding ensembles VLM and grounding models into one model and achieves better performance than open-set methods. However, aggregating VLM and grounding models to one model requires joint training of two models, introducing additional training costs. By contrast, VL-SAM is a training-free framework and obtains significant performance improvement over LLaVA-Grounding from 18.4 mAR to 40.1 mAR.

In addition, we evaluate the performance upper bound of the current SAM. We utilize ground-truth boxes as the box prompt for SAM decoder to segment objects. We can observe that, in this setting, SAM achieves 54.1 mAR and 94.1 AR_{50} since SAM has its limitations on segmentation tasks.

Table 2: **Comparison of object detection results on CODA.** We chose the best performance for * results from CODA. † denotes few-shot object detectors in the one-shot setting. 'Oracle' represents utilizing ground-truth boxes as the box prompt for SAM.

Method	Type	VLM	Training	CODA mAR	AR_{50}	AR_{75}
RetinaNet* [27]		×	✓	12.8	23.2	11.9
Faster R-CNN* [36]		×	✓	10.7	19.2	10.2
Cascade R-CNN* [4]		×	✓	10.4	18.5	9.7
Deformable DETR* [56]	Close-Set	×	✓	9.0	22.2	5.6
Sparse R-CNN* [38]		×	✓	10.1	19.6	9.0
Cascade Swin* [31]		×	✓	9.9	17.2	9.7
RPN* [36]		×	✓	10.6	20.0	10.2
ORE* [19]		×	✓	8.3	16.4	7.4
FsDet† [42]		×	✓	4.2	7.7	4.0
DeFRCN† [34]	Open-Set	×	✓	4.5	8.9	4.2
GroundingDINO [30]		✓	✓	12.6	21.7	13.3
YOLOWorld [7]		✓	✓	16.1	26.2	19.6
LLaVA-Grounding [51]	Open-Ended	✓	✓	18.4	30.5	22.0
VL-SAM (Ours)		✓	×	40.1	90.1	50.5
GT+SAM (Oracle)	−	−	−	54.1	94.1	64.9

Table 3: **Ablation of main components.** 'Attn' is short for 'attention'. Each component improves the detection performance consistently.

Naive Attn	Attn Generation	Prompt Generation	Iterative Refine	Multi-scale	Question ensemble	mAR
✓						2.2
✓				✓	✓	5.0
	✓					10.1
	✓	✓				12.3
	✓	✓	✓			14.1
	✓	✓	✓	✓		27.3
	✓	✓	✓	✓	✓	40.1

It sometimes over- or under-segments an object and cannot obtain perfect segmentation results. Nevertheless, VL-SAM achieves 74.1% mAR performance of this upper bound, demonstrating the effectiveness of the proposed framework. Overall, VL-SAM achieves favorable performance on the CODA dataset.

4.3 Ablation Study

Main Components. As shown in Table 3, we conduct ablation studies on CODA to analyze the effectiveness of each component of VL-SAM. For the baseline Naive Attention method, we use the attention map from the last layers and average all attention heads. We can see that the Naive Attention baseline obtains unsatisfactory results even with multi-scale and question ensemble techniques. With the proposed attention generation module, we improve the baseline by 7.9 mAR. Adding points pairs with prompt generation brings 2.2 mAR improvement. Besides, refining the segmentation maps with the iterative refinement module improves the detection performance from 12.3 mAR to 14.1 mAR. Furthermore, ensembling with multi-scale image input and question prompt obtains 13.2 mAR and 12.8 mAR, respectively. Though multi-scale and question prompt ensembles greatly improve performance, these two ensemble techniques do not show effectiveness without the proposed components. In summary, the results show the effectiveness of each component proposed in VL-SAM.

Attention Generation. To obtain high-quality attention maps from VLM, we introduce head weights to fuse transformer heads and a regularization term for attention flow. As shown in Table 4, simply using attention flow [1] almost fails to recognize objects for SAM due to the attention collapse caused by causal masks (Figure 5). With the regularization term, the attention flow mechanism shows its

Table 4: **Ablation of attention generation.** We can obtain high-quality attention maps with the proposed modules.

Naive Attention Map	Attention Flow		Head Weight	mAR
	No Regularization	Regularization		
✓				2.2
	✓			0.1
		✓		8.5
		✓	✓	10.1

Table 5: **Ablation of model generalization.** VL-SAM can adopt various vision-language models and segmentation models.

Vision-Language Model	Segmentation Model	mAR
CogVLM	SAM	40.1
MiniGPT-4	SAM	34.7
LLaVA	SAM	37.2
CogVLM	MobileSAM	29.2

superiority over naive attention by improving 6.3 mAR. Moreover, fusing with head weights leads to a 1.6 mAR improvement.

Model Generalization. To demonstrate the model generalization ability of the VL-SAM framework, we adapt two additional popular VLMs, MiniGPT-4 [55] and LLaVA [29] to replace CogVLM and use MobileSAM [50] to replace SAM. In Table 5, we present the results of using these models in the VL-SAM framework. Empirical results show that replacing CogVLM with MiniGPT-4 or LLaVA may reduce the object localization performance in corner cases as CogVLM shows more powerful multimodal chat and reasoning ability than MiniGPT-4 and LLaVA. This indicates that our VL-SAM framework can benefit from more powerful VLMs. Besides, replacing SAM with a more lightweight but less accurate MobileSAM also leads to performance drops. Nevertheless, all these results outperform previous methods (18.4 mAR) in Table 2. This evidences that our framework can generalize to multiple vision-language and segmentation models.

5 Limitations

Since we combine VLM and SAM to address the open-ended object detection and segmentation task, VL-SAM inherits the defects of VLM and SAM. The first defect is the hallucination problem in VLM. VL-SAM also suffers from hallucinations, generating wrong object tokens and attention maps. The second defect is the low inference speed of VL-SAM. However, these defects can be fixed in the future. For example, there are many more efficient SAM variant models, including EfficientSAM [46] and MobileSAM [50]. Our framework can benefit from these new models since we can easily replace CogVLM and SAM in VL-SAM with these more efficient and highly accurate models.

6 Conclusion

In this paper, we introduce VL-SAM, a framework that cascades VLM and SAM with the attention map to address the open-ended object detection and segmentation task. Without additional training, we adopt attention maps generated by VLM as the prompts for SAM to segment objects. We introduce the attention flow mechanism to aggregate high-quality attention maps. Besides, we present an iterative refinement pipeline with positive and negative points pair sampling strategy to acquire more accurate segmentation masks. Experimental results on the long-tail generic instance segmentation dataset LVIS show that VL-SAM beats the open-ended method GenerateU and achieves competitive performance compared with close-set and open-set methods. Moreover, VL-SAM achieves favorable results on the corner case object detection dataset CODA.

Broader Impacts Statement. This paper studies utilizing VLM and SAM for open-ended object detection and segmentation. We do not see potential privacy-related issues. This study may inspire future research on open-ended perception and potential corner case object detection applications in autonomous driving. However, the proposed model's performance is not yet up to the level of practical application and may pose safety threats when applied directly in practice.

References

[1] Samira Abnar and Willem H. Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[6] Xi Chen, Shuang Li, Ser-Nam Lim, Antonio Torralba, and Hengshuang Zhao. Open-vocabulary panoptic segmentation with embedding modulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:2401.17270*, 2024.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[9] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *arXiv preprint arXiv:2102.01066*, 2021.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[11] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: general language model pretraining with autoregressive blank infilling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[12] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

[13] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[18] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023.

[19] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[20] 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. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal on Computer Vision (IJCV)*, 2017.

[22] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023.

[24] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2022.

[25] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] Chuang Lin, Yi Jiang, Lizhen Qu, Zehuan Yuan, and Jianfei Cai. Generative region-language pretraining for open-ended object detection. *arXiv preprint arXiv:2403.10191*, 2024.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Neural Information Processing Systems (NeurIPS)*, 2023.

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

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[32] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Neural Information Processing Systems (NeurIPS)*, 2023.

[33] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[34] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[35] 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 (ICML)*, 2021.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NeurIPS)*, 2015.

[37] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 2012.

[38] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.

[39] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[41] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[42] Xin Wang, Thomas E. Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning (ICML)*, 2020.

[43] Zhenyu Wang, Yali Li, Xi Chen, Ser-Nam Lim, Antonio Torralba, Hengshuang Zhao, and Shengjin Wang. Detecting everything in the open world: Towards universal object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[44] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, et al. On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving. *arXiv preprint arXiv:2311.05332*, 2023.

[45] Jiannan Wu, Yi Jiang, Bin Yan, Huchuan Lu, Zehuan Yuan, and Ping Luo. Exploring transformers for open-world instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[46] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv preprint arXiv:2312.00863*, 2023.

[47] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Neural Information Processing Systems (NeurIPS)*, 2022.

[48] Lewei Yao, Renjie Pi, Jianhua Han, Xiaodan Liang, Hang Xu, Wei Zhang, Zhenguo Li, and Dan Xu. Detclipv3: Towards versatile generative open-vocabulary object detection. *arXiv preprint arXiv:2404.09216*, 2024.

[49] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. *arXiv preprint arXiv:2401.02955*, 2024.

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

[51] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. *arXiv preprint arXiv:2312.02949*, 2023.

[52] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Neural Information Processing Systems (NeurIPS)*, 2022.

[53] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

[54] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Xianzheng Ma, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023.

[55] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[56] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

[57] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multi-modal assistant with small language model. *arXiv preprint arXiv:2401.02330*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 claim the main contribution of this paper in both the Abstract and Introduction sections.

   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 discuss the limitation of this work in Section 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: This paper does not include theoretical results.

   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 the implementation details in Section 4.1.

   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: [No]

   Justification: We do not provide new datasets and will release the demo after the paper is accepted.

   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: Our method does not require training. We provide the hyperparameters in Section 4.1.

   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: [NA]

   Justification: Error bars are not reported because our method only involves inference.

   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 provide the information for computer resources in Section 4.1.

   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: The research in the paper conforms with 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 provide the discussion of broader impacts in Section 6.

    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: The models in this paper pose no such risks.

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: All owners of models, code, and data we used are properly cited. We compliance all licenses of models, code, and data.

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: The paper does not release new assets.

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: This paper does not involve crowdsourcing or 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: This paper does not involve crowdsourcing or 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.