
VL-SAM-V2: Open-World Object Detection with General and Specific Query Fusion

Zhiwei Lin Yongtao Wang*

Wangxuan Institute of Computer Technology, Peking University, China
{zwlin, wyt}@pku.edu.cn

Abstract

Current perception models have achieved remarkable success by leveraging large-scale labeled datasets, but still face challenges in open-world environments with novel objects. To address this limitation, researchers introduce open-set perception models to detect or segment arbitrary test-time user-input categories. However, open-set models rely on human involvement to provide predefined object categories as input during inference. More recently, researchers have framed a more realistic and challenging task known as open-ended perception that aims to discover unseen objects without requiring any category-level input from humans at inference time. Nevertheless, open-ended models suffer from low performance compared to open-set models. In this paper, we present VL-SAM-V2, an open-world object detection framework that is capable of discovering unseen objects while achieving favorable performance. To achieve this, we combine queries from open-set and open-ended models and propose a general and specific query fusion module to allow different queries to interact. By adjusting queries from open-set models, we enable VL-SAM-V2 to be evaluated in the open-set or open-ended mode. In addition, to learn more diverse queries, we introduce ranked learnable queries to match queries with proposals from open-ended models by sorting. Moreover, we design a denoising point training strategy to facilitate the training process. Experimental results on LVIS show that our method surpasses the previous open-set and open-ended methods, especially on rare objects.

1 Introduction

Deep learning has made significant breakthroughs in computer vision tasks, enabling substantial progress in object detection tasks. Traditional object detection approaches [34, 16] typically employ a closed-set paradigm, in which detection models are restricted to recognizing and locating objects that are seen in the training set. However, closed-set approaches have notable limitations, particularly in real-world scenarios where novel or unseen objects appear. For instance, in an autonomous driving system, if its object detector trained to identify vehicles and pedestrians encounters a novel object [22], such as a drone or a wild animal, it may either misclassify the object or fail to detect it, which can even lead to serious traffic accidents.

To address this challenge, open-world methods [12, 9, 14] have been proposed, allowing models to be more flexible by enabling them to handle novel and unseen objects. Open-world methods generally can be divided into two categories: open-set (or open-vocabulary) [14, 29, 11] and open-ended [25, 27]. Open-set methods can identify objects within a predefined object category list, which is provided by humans during the inference phase, even if unseen objects are present in the list. However, these open-set models cannot predict objects outside the predefined object category list, which limits their ability to discover new object categories automatically. For instance, if the

*Corresponding author.

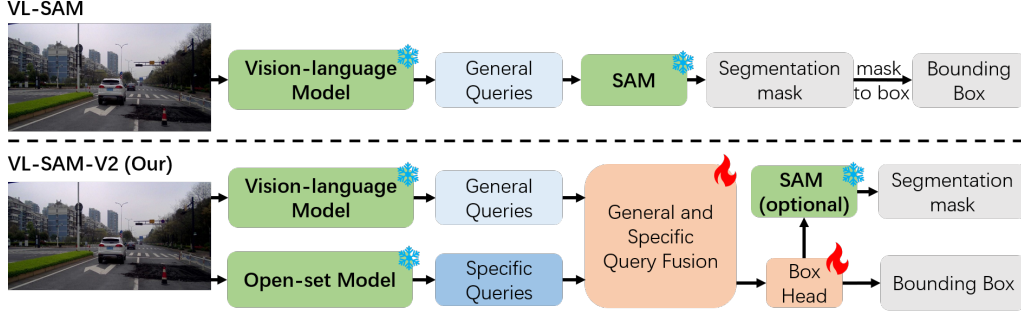


Figure 1: **Illustration of VL-SAM-V2.** VL-SAM-V2 combines the general queries from VL-SAM and the specific queries of an open-set model with a query fusion module.

predefined object category list only contains vehicles and pedestrians, the open-set models still cannot detect drones or wild animals in the input images. Therefore, they need humans to intervene by providing a more comprehensive list including these objects. Additionally, although open-set models perform well on frequent objects, they often struggle with rare objects [29]. In contrast, open-ended models aim to identify and categorize all objects in given images without requiring a predefined object category list provided by humans [25]. This capability is usually achieved by incorporating large vision-language models. However, open-ended models suffer from low performance [43, 27], especially when dealing with crowded objects, which typically appear in frequent object categories.

To this end, we introduce VL-SAM-V2, an open-world object detection framework that combines the strengths of open-set and open-ended methods to achieve favorable performance in both rare and frequent object categories while equipped with the ability to discover novel objects, as shown in Figure 1. Specifically, we treat object queries from open-set models as specific queries, since they specialize in frequent objects. Meanwhile, we adopt the VL-SAM pipeline to generate object queries in an open-ended manner and view them as general queries, because of the generalization of large vision-language models. Then, we propose a general and specific query fusion module to fuse two distinct types of queries with the attention mechanism in a lightweight manner. Since the VL-SAM only provides point prompts, we present ranked learnable queries by sorting the general point prompts based on their scores and matching them with the learnable queries to compose general queries. In addition, to facilitate the training process, we introduce the denoising points strategy by adding both positive and negative points as the noisy general queries. This denoising process helps models to learn to distinguish positive and high-accuracy general queries and to filter out negative queries or point prompts from VL-SAM. During inference, VL-SAM-V2 can be dynamically adapted as an open-set or open-ended model based on the predefined object category list. When the list is empty, VL-SAM-V2 is viewed as an open-ended model. Experimental results show that VL-SAM-V2 significantly improves detection performance on rare objects compared to existing open-set models by discovering novel objects with general queries. Meanwhile, VL-SAM-V2 compensates for the poor performance of open-ended models.

The main contributions of this work are summarized as follows:

- We introduce an open-world object detection framework, VL-SAM-V2, to combine the strengths of open-set and open-ended methods with the general and specific query fusion module. VL-SAM-V2 can be dynamically adapted as an open-set or open-ended model.
- We design ranked learnable queries to transform point prompts from VL-SAM into object queries and present a denoising points strategy to facilitate the training process.
- Experimental results show that VL-SAM-V2 achieves both state-of-the-art open-set and open-ended object detection performance on the LVIS dataset in a zero-shot manner.

2 Related work

2.1 Open-set Object Detection

Open-set or open-vocabulary object detection methods aim to detect objects in a predefined object category list given by humans [29]. To achieve this, current open-set object detection methods

often transform the open-set detection task into a vision-language alignment task, allowing detectors to match unseen object categories with visual information through the alignment process. With the advent of contrastive learning [15, 3], vision-language alignment models (*e.g.*, CLIP [33]) achieve favorable zero-shot alignment performance for unseen objects. Based on this, many open-set object detection methods 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. However, CLIP is trained to align the whole image and texts, while the open-set object detection task favors region-level feature alignment.

To address this, GLIP [24] proposes an object-aware vision-language region-level alignment task with phrase grounding to pre-train open-world object detectors. DetCLIP [40] separates the texts in GLIP into a parallel concept and regards each object category as an individual input. GLIPv2 [44] further improves on GLIP by introducing a deep fusion block for better vision-language fusion and intra- and inter-image-text contrastive losses. GroundingDINO [29] follows the idea of GLIP to present image-text cross-modality fusion, and proposes object queries by text information. OWL [32] scales the training dataset to 10M examples with pseudo-annotation and a self-training strategy. YOLO-World [7] adopts the efficient YOLO-series backbone to achieve real-time open-set object detection. More recently, due to the emergence of large language models and vision-language models, many open-set methods try to adopt them for additional supervision. DetCLIPv3 [41] combines the object-level and image-level generation tasks with the open-set object detection task. LLMDet [11] explores two-stage co-training by generating image-level and region-level detail captions with a pre-trained vision-language model. In addition, some works [19, 43] directly fine-tune a pre-trained vision-language model to address the open-set object detection by following pix2seq [4].

However, the above methods require a predefined object category list given by humans and cannot discover novel objects by themselves. In this paper, we propose prompting the open-set method using point priority from large vision-language models and enabling the proposed model to discover novel objects not in the predefined object category list.

2.2 Open-ended Object Detection

Open-ended object detection methods can directly predict all objects in given images without any predefined object category list from humans. GenerateU [25] first introduces the open-ended problem and proposes a generative framework with a large language model to generate object categories from object queries. In addition, it constructs a large dataset with pseudo bounding boxes and caption pairs to fine-tune the whole model. To alleviate the training costs, VL-SAM [27] presents a training-free open-ended object detection and segmentation framework that connects a large vision-language model and the segment-anything model (SAM) [18] with attention maps as the intermediate prompts. Moreover, some large vision-language models [19, 43, 20] introduce specific tokens, like '<Det>', to allow models to predict objects in an open-ended manner. Though good grounding performance of these vision-language models is achieved, the accuracy of detecting all objects is very low.

Though some progress has been made on the open-ended detection task, the performance of current open-ended methods is significantly lower than that of open-set methods. To address this issue, we propose a general and specific fusion pipeline by fusing the queries from open-set and open-ended models. The proposed pipeline preserves the strengths of open-set methods' high performance and can discover novel objects like open-ended methods.

3 Method

3.1 Preliminary of VL-SAM

VL-SAM [27] is a training-free open-ended object detection and segmentation framework that connects vision-language and segment-anything models with attention maps as the intermediate prompts. It utilizes a vision-language model to generate the categories of all objects in the given image and stores the corresponding attention maps. Then, VL-SAM adopts the head aggregation and attention flow mechanism to aggregate and propagate attention maps through all heads and layers. After that, a sampling strategy is applied to the refined attention maps to generate positive and negative points as the point prompts for SAM. In addition, VL-SAM uses several ensemble strategies to obtain more comprehensive object points.

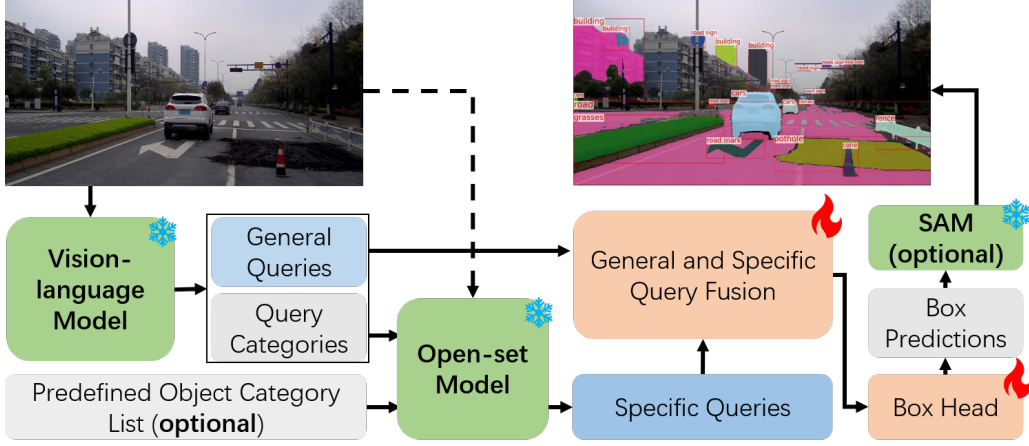


Figure 2: **The overall pipeline of VL-SAM-V2.** VL-SAM-V2 utilizes a vision-language model to generate general queries and a standard open-set detection model to generate specific queries. Then, the two distinct queries are sent to the general and specific query fusion module for interaction. Finally, a box head and an optional SAM are applied to predict the perception results. During the training, we only fine-tune the general and specific query fusion module and the box head. In addition, by controlling the predefined object category list, VL-SAM-V2 can operate in open-ended mode.

In this paper, we adopt VL-SAM to generate point prompts as proposals for the general queries in an open-ended manner.

3.2 VL-SAM-V2

As shown in Figure 2, we provide an overview of the proposed framework. Specifically, given an image input, we first use VL-SAM to discover all objects in the image and sample their point coordinates according to the attention maps. Then, we combine the discovered objects with the predefined object set to form the final object set for the open-set model to generate specific queries. Meanwhile, the sampled points are converted into initial bounding boxes and matched with learnable queries to compose general queries. After that, the general and specific queries are sent to the general and specific query fusion module to update the queries. Finally, the box head and optional SAM are applied to predict the final results.

Denoising Points. VL-SAM-V2 employs VL-SAM to generate point prompts to compose general queries. However, during training, the inference of VL-SAM will incur additional training costs due to the use of a large vision-language model. In addition, directly training with a few points from VL-SAM can lead to unstable training and poor convergence. Inspired by the denoising anchor boxes trick in the training of DETR [21, 42], we propose denoising points to address this issue. Specifically, we randomly sample positive points in the ground-truth bounding boxes and negative points outside the boxes to replace the point proposals generated by VL-SAM. The sampled noisy points are sent to the decoder, and the models are trained to denoise the noisy points into corresponding bounding boxes and labels. More concretely, we follow DINO [42] to sample points by adding coordinate noise ΔX and ΔY with two hyper-parameters λ_1 and λ_2 :

$$\begin{aligned} |\Delta X_p| &< \lambda_1 \times w/2 < |\Delta X_n| < \lambda_2 \times w/2, \\ |\Delta Y_p| &< \lambda_1 \times h/2 < |\Delta Y_n| < \lambda_2 \times h/2, \end{aligned} \quad (1)$$

where p and n denote positive and negative points, w and h indicate the width and height of the ground-truth bounding box for denoising. The ΔX and ΔY are sampled from the uniform distribution according to the range in Eq. 1.

To convert sampled noisy points into bounding boxes and general queries for subsequent fusion and decoding, we utilize multi-level image features to predict the initial bounding boxes according to the coordinates of noisy points and introduce ranked learnable queries. Specifically, given the coordinates of a noisy point (X_{noise}, Y_{noise}) and the multi-level image features f , we interpolate the image

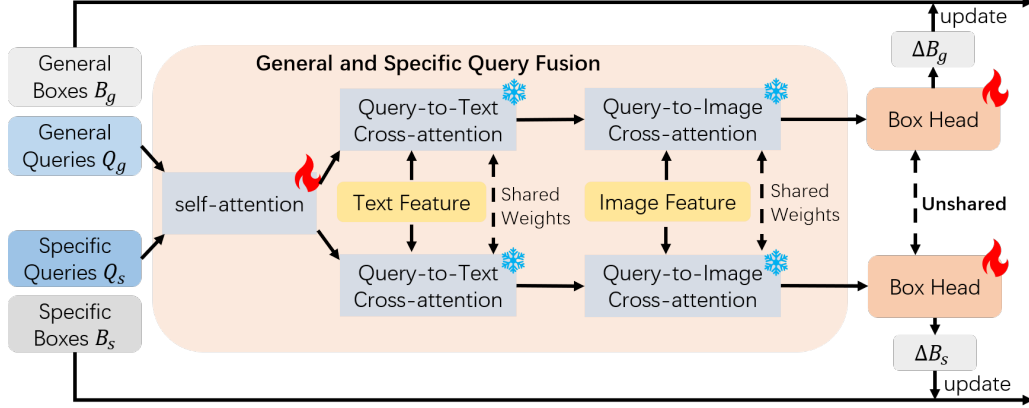


Figure 3: **Illustration of general and specific query fusion module.** General and specific queries interact with a self-attention mechanism. Then, the shared query-to-text and query-to-image cross-attention are applied for the two queries independently. Finally, the unshared box heads predict the offset of corresponding bounding boxes. During the training, we only update the parameters in the self-attention and box heads.

features in (X_{noise}, Y_{noise}) for all levels and obtain sampled point feature $f_{X_{noise}, Y_{noise}}^l$, where l denotes the l -th level of the image features. After that, we send the point feature $f_{X_{noise}, Y_{noise}}^l$ to a linear layer and a layer normalization, followed by a box head in the open-set model to obtain the initial bounding box. The initial bounding box for each point is combined with the ranked learnable query to compose general queries and sent to the fusion module and box head.

Ranked Learnable Queries. Each point prompt or noisy point should be matched with a learnable query for subsequent fusion and decoding. However, point prompts or noisy points are sampled randomly without any order. A simple way is to set the same learnable query for each point, without considering their difference in coordinates or features. Intuitively, different queries are expected to handle different objects. Therefore, we introduce ranked learnable queries by matching different points with different queries. Specifically, for noisy points, we send $f_{X_{noise}, Y_{noise}}^l$ to the classification head to obtain the score of each noisy point. Then, we rank the noisy points from largest to smallest based on their scores. Meanwhile, we set N additional learnable queries and fix their order after initialization. Finally, the ranked noisy points are matched with the learnable queries one by one. For example, the noisy point with the largest score matches the first learnable query. Since the number of noisy points may not match N , we simply discard the redundant noisy points or ranked learnable queries.

Similarly, during inference, the points sampled by VL-SAM can be ranked by scores and matched with ranked learnable queries.

General and Specific Query Fusion. As shown in Figure 3, after obtaining the initial bounding box and ranked learnable query of each point, we combine them with the bounding boxes and queries generated by the open-set model and send them into the general and specific query fusion module, which is distributed in each transformer decoder layer of the open-set model. The boxes and queries of sampled points are treated as general boxes B_g and queries Q_g since they are prompted by the large vision-language model, while the boxes and queries from the open-set model are denoted as specific boxes B_s and queries Q_s . Since the number of sampled points varies, we adopt the self-attention module for the interaction between the general and specific queries, as shown in Figure 3. Specifically, Q_g and Q_s are first concatenated and sent to a standard self-attention module to obtain the fused queries. Then, the fused queries are separated and sent to a shared query-to-text cross-attention, a shared query-to-image cross-attention, and a shared FFN to obtain updated queries \hat{Q}_g and \hat{Q}_s . After that, two unshared box heads are applied to \hat{Q}_g and \hat{Q}_s to predict the coordinate offsets of bounding boxes ΔB_g and ΔB_s , respectively. Finally, we update the B_g and B_s as follows:

$$\begin{aligned} B_g &\leftarrow B_g + \Delta B_g, \\ B_s &\leftarrow B_s + \Delta B_s. \end{aligned} \quad (2)$$

During training, we only update the parameters in the self-attention and box heads.

Loss Function. We adopt the grounding losses $\mathcal{L}_{grounding}$ and the generation losses $\mathcal{L}_{generation}$ in LLMDet [11] as the supervision to fine-tune VL-SAM-V2. The grounding losses contain vision-language alignment and bounding box regression losses, while the generation losses comprise region-level and image-level caption losses. In VL-SAM-V2, the alignment process between ground-truth and general/specific boxes is performed independently. The final loss can be calculated as follows:

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{grounding}(\hat{B}, B_g) + \mathcal{L}_{grounding}(\hat{B}, B_s)) + \mathcal{L}_{generation}(\hat{T}, T), \quad (3)$$

where \hat{B} denotes the ground-truth bounding boxes, \hat{T} and T are the ground-truth captions and generated text predictions from a vision-language model in LLMDet [11], respectively.

4 Experiments

4.1 Implementation Details

For VL-SAM, we choose InternVL-2.5-8B with InternViT-300M [6] and InternLM2.5-7B [5] as the vision-language model. We set the temperature to 0.8 and top-p for nucleus sampling to 0.8 for InternVL-2.5-8B. For the open-set model, we select LLMDet [11] as the baseline model because of its SOTA performance. The number N of additional learnable queries is set to 900. For denoising points, both hyper-parameters λ_1 and λ_2 are set to 1.

The whole model of VL-SAM-V2 is fine-tuned with GroundingCap-1M dataset [11] following the training protocol of LLMDet [11]. During training, only the self-attention modules in general and specific query fusion and box heads are fine-tuned, while others are frozen. We fine-tune VL-SAM-V2 for 150k iterations using automatic mixed-precision with a batch size of 16. All training can be done on 8 NVIDIA A800 GPUs within two days.

4.2 Main Results

We mainly evaluate the proposed method on the LVIS dataset, which contains 1203 categories. We adopt the fixed AP [8] as the evaluation metric on frequent, common, and rare classes.

Open-set Object Detection. As shown in Table 1, VL-SAM-V2 beats all previous open-set models and achieves the new state-of-the-art zero-shot open-set object detection results. Specifically, with general and specific query fusion, VL-SAM-V2 outperforms the baseline LLMDet by 1.0 AP and 0.8 AP on LVIS minival with Swin-T [30] and Swin-L as backbones, respectively. Notably, for rare objects, VL-SAM-V2 obtains a significant performance improvement, from 37.3 AP_r to 41.2 AP_r with Swin-T and from 45.1 AP_r to 47.2 AP_r with Swin-L. These results demonstrate our motivation that the prior knowledge from large vision-language models enables the open-set model to discover rare objects.

In addition, we notice that DetCLIPv3 [41] achieves higher AP_r and AP_c. The reason is that DetCLIPv3 collects more balanced data and noun concept corpora. Nevertheless, we believe that the performance of DetCLIPv3 can be further improved by the fusion pipeline of VL-SAM-V2.

Moreover, we evaluate VL-SAM-V2 on other datasets, including COCO [26] and CODA [22]. As shown in Table 3, our method achieves the best results on various datasets, especially on the CODA dataset.

Open-ended Object Detection. For open-ended object detection evaluation, we set the predefined object category list to empty and follow GenerateU [25] and VL-SAM [27] to match the generated object categories with the category list in LVIS by CLIP [33]. We present the open-ended object detection results in Table 2. The results show that VL-SAM-V2 achieves the best open-ended object detection performance on AP and AP_r. Specifically, with Swin-T as the backbone, VL-SAM-V2 outperforms GenerateU by 2.7 AP and 9.8 AP_r. For Swin-L, VL-SAM-V2 also improves AP_r by a large margin compared to previous methods, from 22.3 AP_r and 23.4 AP_r to 30.5 AP_r. Moreover, though VL-SAM obtains a higher AP_r than GenerateU, the overall AP is lower than GenerateU. This indicates that the priority of vision-language models favors rare objects but performs unsatisfactorily

Table 1: **Comparison of zero-shot open-set object detection results on LVIS val and minival [13].** We report *fixed* AP [8]. Grey results denote using additional private data.

Method	Backbone	LVIS ^{minival}				LVIS			
		AP	AP _r	AP _c	AP _f	AP	AP _r	AP _c	AP _f
GLIP [24]	Swin-T	26.0	20.8	21.4	31.0	17.2	10.1	12.5	25.2
GLIPv2 [44]	Swin-T	29.0	—	—	—	—	—	—	—
CapDet [31]	Swin-T	33.8	29.6	32.8	35.5	—	—	—	—
Grounding-DINO [29]	Swin-T	27.4	18.1	23.3	32.7	20.1	10.1	15.3	29.9
OWL-ST [32]	CLIP B/16	34.4	38.3	—	—	28.6	30.3	—	—
Desco-GLIP [23]	Swin-T	34.6	30.8	30.5	39.0	26.2	19.6	22.0	33.6
DetCLIP [40]	Swin-T	35.9	33.2	35.7	36.4	28.4	25.0	27.0	28.4
DetCLIPv2 [39]	Swin-T	40.4	36.0	41.7	40.4	32.8	31.0	31.7	34.8
YOLO-World-L [7]	YOLOv8-L	35.4	27.6	34.1	38.0	—	—	—	—
T-Rex2 [17]	Swin-T	42.8	37.4	39.7	46.5	34.8	29.0	31.5	41.2
OV-DINO [37]	Swin-T	40.1	34.5	39.5	41.5	32.9	29.1	30.4	37.4
LLMDet [11]	Swin-T	44.7	37.3	39.5	50.7	34.9	26.0	30.1	44.3
VL-SAM-V2 (Our)	Swin-T	45.7	41.2	41.1	50.7	35.5	29.3	31.8	44.3
GLIP [24]	Swin-L	37.3	28.2	34.3	41.5	26.9	17.1	23.3	36.4
GLIPv2 [44]	Swin-H	50.1	—	—	—	—	—	—	—
Grounding-DINO [29]	Swin-L	33.9	22.2	30.7	38.8	—	—	—	—
OWL-ST [32]	CLIP L/14	40.9	41.5	—	—	35.2	36.2	—	—
DetCLIP [40]	Swin-L	38.6	36.0	38.3	39.3	28.4	25.0	27.0	31.6
DetCLIPv2 [39]	Swin-L	44.7	43.1	46.3	43.7	36.6	33.3	36.2	38.5
DetCLIPv3 [41]	Swin-L	48.8	49.9	49.7	47.8	41.4	41.4	40.5	42.3
LLMDet [11]	Swin-L	51.1	45.1	46.1	56.6	42.0	31.6	38.8	50.2
VL-SAM-V2 (Our)	Swin-L	51.7	47.2	46.7	56.6	42.5	33.2	39.7	50.2

Table 2: **Comparison of zero-shot open-ended object detection results on LVIS minival [13].** We report *fixed* AP [8]. VL-SAM utilizes ViT-H [10] of SAM as the image encoder for segmentation.

Method	Image Encoder	Vision-Language Model	AP	AP _{rare}
GenerateU [25]	Swin-T	FlanT5-base	26.8	20.0
VL-SAM-V2 (Our)	Swin-T	InternVL-2.5 (8B)	29.5	29.8
GenerateU [25]	Swin-L	FlanT5-base	27.9	22.3
VL-SAM [27]	ViT-H	CogVLM (17B)	25.3	23.4
VL-SAM-V2 (Our)	Swin-L	InternVL-2.5 (8B)	31.8	30.5

on frequent objects. In contrast, VL-SAM-V2 obtains both the best AP and AP_r, demonstrating that the fusion with the general proposals from open-set models and specific proposals from vision-language models helps the models to retain the high performance of open-set models on frequent objects.

4.3 Ablation Study

In this section, we conduct ablation experiments to analyze the main components and model generalization of VL-SAM-V2. We use Swin-T as the backbone and report the results on LVIS minival.

Main Components. To evaluate the effectiveness of each component, we successively add components to the LLMDet baseline. As shown in Table 4, we observe that each component consistently improves performance. Specifically, adding general and specific query fusion brings improvements of 0.5 AP, 1.8 AP_r, and 0.8 AP_c, respectively. Then, we adopt ranked learnable queries to replace the single learnable query for each point prompt, obtaining a 1.1 AP_r improvement. Moreover, training with denoising points can further improve AP_r by 1.0. Meanwhile, the overall training cost is reduced by 50% due to the removal of point generation in VL-SAM.

Table 3: **Comparison of open-set object detection results on COCO and CODA.** We report mAP for each dataset.

Method	COCO	CODA
GroundingDINO	48.4	12.6
YOLO-World	45.1	16.1
LLMDet	55.6	38.5
VL-SAM-V2	56.0	42.3

Table 4: **Ablations on main components.** ‘GS Fusion’ denotes the general and specific query fusion module. Each component improves the detection performance consistently.

GS Fusion	Ranked Learnable Queries	Denoising Points	AP	AP _r	AP _c
			44.7	37.3	39.5
✓			45.2	39.1	40.3
✓	✓		45.4	40.2	40.6
✓	✓	✓	45.7	41.2	41.1

Table 5: **Ablations on fusion methods.** ‘GS Fusion’ denotes the general and specific query fusion module.

Fusion methods	AP	AP _r
Naive Concatenate	44.9	38.5
Late Fusion	45.1	40.2
GS Fusion	45.7	41.2

Overall, the results demonstrate the effectiveness of each component proposed in VL-SAM-V2.

Fusion Methods. In addition to the proposed general and specific query fusion, we try two other fusion methods, *i.e.*, Naive Concatenate and Late Fusion. Naive Concatenate simply concatenates two query features into one feature with a linear layer. Late Fusion directly ensembles the detection results from two queries. As shown in Table 5, we can observe that GS Fusion achieves the best fusion performance, outperforming naive concatenate by 2.7 AP_r.

Model Generalization. To demonstrate the model generalization of the VL-SAM-V2, we adapt various popular open-set models and vision-language models, including GroundingDINO [29], LLMDet [11], LLaVA [28], CogVLM [38], and InternVL-2.5 [5].

As shown in Table 6, VL-SAM-V2 consistently improves the performance of different open-set models with different vision-language models, ranging from 2.3 to 4.9 AP_r. Moreover, the empirical results show that stronger vision-language models can obtain better detection performance, aligning with the finding in VL-SAM [27]. This indicates that VL-SAM-V2 can benefit from more powerful open-set models and vision-language models.

4.4 Combined with SAM.

Following Grounded-SAM [35], we combine the proposed VL-SAM-V2 with the segment-anything model (SAM) [18] to achieve open-world instance segmentation.

As shown in Table 7, we compare the proposed method combining SAM with VL-SAM on the open-ended instance segmentation task. We can find that our method shows favorable performance improvement. In addition, we visualize the open-ended instance segmentation results on the corner case object detection dataset, CODA, to demonstrate the effectiveness of the proposed method in real-world scenarios. As depicted in Figure 4, our method can provide dense segmentation masks and category annotations for input images and discover various novel object categories outside of the existing autonomous driving datasets [36, 1], including sacks and cranes. Moreover, since VL-SAM-V2 uses point prompts as inputs, users can provide point prompts by themselves to achieve

Table 6: **Ablation of model generalization.** VL-SAM-V2 can adopt various open-set models and vision-language models for point prompting. † denotes results are reimplemented by MMDetection [2].

Method	Open-set Models	Vision-Language Model	AP	AP _{rare}
GroundingDINO†	-	-	41.4	34.2
LLMDet	-	-	44.7	37.3
VL-SAM-V2 (Our)	GroundingDINO	LLaVA (7B)	42.1	37.3
		CogVLM (17B)	42.7	38.5
		InternVL-2.5 (8B)	43.0	39.1
	LLMDet	LLaVA (7B)	45.2	39.6
		CogVLM (17B)	45.5	40.6
		InternVL-2.5 (8B)	45.7	41.2

Table 7: **Comparison of zero-shot open-ended instance segmentation results on LVIS mini-val [13].**

Method	Vision-Language Model	mask AP	mask AP _{rare}
VL-SAM [27]	CogVLM (17B)	23.9	22.7
VL-SAM-V2 (Our)	InternVL-2.5 (8B)	28.7	27.7



Figure 4: **Visualization results VL-SAM-V2 combining with SAM on CODA [22].** We show input images and detection and segmentation prediction results in the open-ended mode. VL-SAM-V2 can discover various uncommon objects. Best viewed by zooming in.

prompt-based detection and segmentation, demonstrating the diverse applications of the proposed method in real-world scenarios.

5 Limitations

Since we adopt the VL-SAM framework to utilize vision-language models to generate point prompts, VL-SAM-V2 inherits the defects of vision-language models, including the hallucination problem, incorrect responses, and low inference speed. However, these defects will be gradually overcome with the development of vision-language models. The proposed VL-SAM-V2 framework can benefit from these new models. Moreover, VL-SAM-V2 needs additional SAM to obtain instance segmentation results. In the future, we will integrate segmentation models into our framework to compose an end-to-end open-world perception model.

6 Conclusion

In this work, we introduce VL-SAM-V2, a novel open-world object detection framework that can perform in the open-set or open-ended mode. The core of VL-SAM-V2 is the general and specific query fusion that integrates information from both open-set and open-ended perception paradigms. To further enhance the model’s ability to discover unseen objects, we present ranked learnable queries to promote more diverse and representative query representations. Additionally, we propose the denoising point training strategy to stabilize and facilitate the training process. Experimental results on the LVIS demonstrate that VL-SAM-V2 outperforms previous state-of-the-art methods in both open-set and open-ended settings, particularly excelling in detecting rare objects. Moreover, VL-SAM-V2 exhibits good model generalization that can incorporate various open-set models and vision-language models. By combining with SAM, VL-SAM-V2 shows promise for a new generation of automatic labeling and open-world perception systems.

Broader Impacts Statement. This paper investigates the use of current open-set models and vision-language models for open-world object detection. We do not see potential privacy-related issues. This study may inspire future research on automatic labeling systems and open-world perception models.

Acknowledgments. This work was supported by National Key R&D Program of China (Grant No. 2022ZD0160305).

References

- [1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.
- [4] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.
- [5] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [7] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [8] 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.
- [9] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. *Neural Information Processing Systems (NeurIPS)*, 2018.
- [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 *International Conference on Learning Representations (ICLR)*, 2020.
- [11] Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. LlmDET: Learning strong open-vocabulary object detectors under the supervision of large language models. *arXiv preprint arXiv:2501.18954*, 2025.
- [12] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 2020.
- [13] 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.
- [14] 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.
- [15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [17] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-rex2: Towards generic object detection via text-visual prompt synergy. In *European Conference on Computer Vision (ECCV)*, 2024.
- [18] 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.

- [19] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [21] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [22] 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.
- [23] Liunian Li, Zi-Yi Dou, Nanyun Peng, and Kai-Wei Chang. Desco: Learning object recognition with rich language descriptions. *Neural Information Processing Systems (NeurIPS)*, 2023.
- [24] 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.
- [25] Chuang Lin, Yi Jiang, Lizhen Qu, Zehuan Yuan, and Jianfei Cai. Generative region-language pretraining for open-ended object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.
- [27] Zhiwei Lin, Yongtao Wang, and Zhi Tang. Training-free open-ended object detection and segmentation via attention as prompts. In *Neural Information Processing Systems (NeurIPS)*, 2024.
- [28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Neural Information Processing Systems (NeurIPS)*, 2023.
- [29] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.
- [30] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [31] Yanxin Long, Youpeng Wen, Jianhua Han, Hang Xu, Pengzhen Ren, Wei Zhang, Shen Zhao, and Xiaodan Liang. Capdet: Unifying dense captioning and open-world detection pretraining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [32] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Neural Information Processing Systems (NeurIPS)*, 2023.
- [33] 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.
- [34] 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.
- [35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.
- [36] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

- [37] Hao Wang, Pengzhen Ren, Zequn Jie, Xiao Dong, Chengjian Feng, Yinlong Qian, Lin Ma, Dongmei Jiang, Yaowei Wang, Xiangyuan Lan, et al. Ov-dino: Unified open-vocabulary detection with language-aware selective fusion. *arXiv preprint arXiv:2407.07844*, 2024.
- [38] Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *Neural Information Processing Systems (NeurIPS)*, 2024.
- [39] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [40] 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.
- [41] 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. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [42] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2023.
- [43] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision (ECCV)*, 2024.
- [44] 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.

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 follow 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 partial code 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: We provide the training details and 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: [No]

Justification: Error bars are not reported because it would be too computationally expensive.

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.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs in Section 3.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.