EfficientViT-SAM: Accelerated Segment Anything Model Without Performance Loss

Zhuoyang Zhang[1,3*], Han Cai[2,3*], Song Han[2,3]
[1]Tsinghua University, [2]MIT, [3]NVIDIA
https://github.com/mit-han-lab/efficientvit

Abstract

We present EfficientViT-SAM, a new family of accelerated segment anything models. We retain SAM's lightweight prompt encoder and mask decoder while replacing the heavy image encoder with EfficientViT. For the training, we begin with the knowledge distillation from the SAM-ViT-H image encoder to EfficientViT. Subsequently, we conduct end-to-end training on the SA-1B dataset. Benefiting from EfficientViT's efficiency and capacity, EfficientViT-SAM delivers 48.9× measured TensorRT speedup on A100 GPU over SAM-ViT-H without sacrificing performance. Our code and pre-trained models are released at https://github.com/mit-han-lab/efficientvit.

1. Introduction

Segment Anything Model (SAM) [1] is a family of image segmentation models pretrained on a high-quality dataset with 11M images and 1B masks. SAM provides astounding zero-shot image segmentation performance and has many applications, including AR/VR, data annotation, interactive image editing, etc.

Despite the strong performance, SAM is highly computation intensive, restricting its applicability in time-sensitive scenarios. In particular, SAM's main computation bottleneck is its image encoder, which requires 2973 GMACs per image at the inference time.

To accelerate SAM, numerous efforts have been made to replace SAM's image encoder with lightweight models. For example, MobileSAM [2] distills the knowledge of SAM's ViT-H model into a tiny vision transformer. EdgeSAM [3] trains a purely CNN-based model to mimic ViT-H, employing a meticulous distillation strategy with the prompt encoder and mask decoder involved in the process. EfficientSAM [4] leverages the MAE pretraining method to improve the performance.

While these methods reduce the computation cost, they all suffer from significant performance drops (Figure 1). This work introduces **EfficientViT-SAM** to address this limitation by leveraging EfficientViT [7] to replace SAM's image encoder. Meanwhile, we retain the lightweight prompt encoder and mask decoder architecture from SAM. Our training process consists of two phases. First, we train the image encoder of EfficientViT-SAM using SAM's image encoder as the teacher. Second, we train EfficientViT-SAM end-to-end on the whole SA-1B dataset [1].

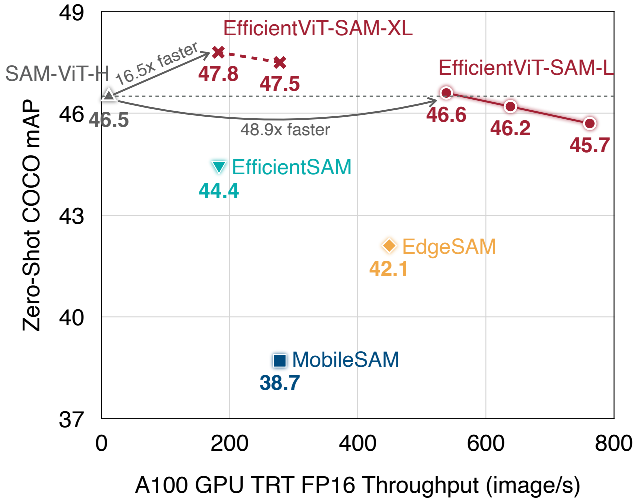

Figure 1. **Throughput vs. COCO Zero-Shot Instance Segmentation mAP.** As far as we know, EfficientViT-SAM is the first accelerated SAM model that matches/outperforms SAM-ViT-H's [1] zero-shot performance, delivering the SOTA performance-efficiency trade-off.

We thoroughly evaluate EfficientViT-SAM on a series of zero-shot benchmarks, including point-prompted segmentation, box-prompted segmentation, and in-the-wild segmentation. EfficientViT-SAM provides a significant performance/efficiency boost over all prior SAM models. In particular, on the COCO dataset [8], EfficientViT-SAM achieves 48.9× higher throughput on A100 GPU without mAP drop compared with SAM-ViT-H [1].

[*] Work done during an internship at NVIDIA.

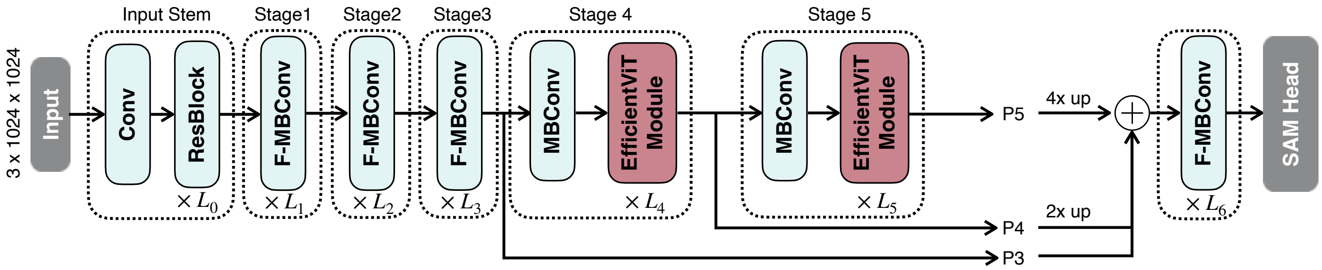

Figure 2. **Macro Architecture of EfficientViT-SAM-XL.** 'ResBlock' refers to the basic building block from ResNet34 [5]. 'F-MBConv' refers to the fused MBConv block from [6]. 'EfficientViT Module' is the building block from [7].

2. Related Work

2.1. Segment Anything Model

SAM [1] has gained widespread recognition as a milestone in the field of computer vision, showcasing its exceptional performance and generalization in image segmentation. SAM defines image segmentation as a promptable task, that aims to generate a valid segmentation mask given any segmentation prompt. To achieve this objective, SAM utilizes an image encoder and a prompt encoder to process the image and provided prompts. The outputs from both encoders are then fed into a mask decoder, which generates the final mask prediction. SAM is trained on a large-scale segmentation dataset comprising over 11 million images with more than 1 billion high-quality masks, enabling robust zero-shot open-world segmentation. SAM has shown its high versatility in a wide range of downstream applications, including image in-painting [9], object tracking [10, 11], and 3D generation [12, 13]. Nevertheless, the image encoder component of SAM imposes significant computational costs, leading to high latency that restricts its practicality in time-sensitive scenarios. Recent works [2–4, 14] have been focused on improving the efficiency of SAM, aiming to address its computational limitations.

2.2. Efficient Deep Learning Computing

Improving the efficiency of deep neural networks is critical when deploying them in real-world applications on both edge and cloud platforms. Our work is related to efficient model architecture design [15, 16] that aims to improve the performance-efficiency trade-off by replacing inefficient model architectures with efficient ones. Our work is also related to knowledge distillation [17] that uses pretrained teacher models to guide the training of student models. Additionally, we can combine EfficientViT-SAM with other parallel techniques to further boost efficiency, including pruning [18], quantization [19], and hardware-aware neural architecture search [20].

3. Method

We propose EfficientViT-SAM, which harnesses EfficientViT [7] to accelerate SAM. In particular, our approach preserves the prompt encoder and mask decoder architecture from SAM while replacing the image encoder with EfficientViT. We design two series of models, EfficientViT-SAM-L and EfficientViT-SAM-XL, offering a balanced trade-off between speed and performance. Subsequently, we train EfficientViT-SAM using the SA-1B dataset [1] in an end-to-end fashion.

3.1. EfficientViT

EfficientViT [7] is a family of vision transformer models for efficient high-resolution dense prediction. Its core building block is a multi-scale linear attention module that enables the global receptive field and multi-scale learning with hardware-efficient operations. Specifically, it substitutes the inefficient softmax attention with lightweight ReLU linear attention to have the global receptive field. By leveraging the associative property of matrix multiplication, ReLU linear attention can reduce the computational complexity from quadratic to linear while preserving functionality. In addition, it enhances the ReLU linear attention with convolution to mitigate its limitation in local feature extraction. More details are available in the original paper [7].

3.2. EfficientViT-SAM

Model Architecture. The macro architecture of EfficientViT-SAM-XL is demonstrated in Figure 2. Its backbone consists of five stages. Similar to EfficientViT [7], we use convolution blocks in the early stages while using EfficientViT modules in the last two stages. We fuse the features from the last three stages by upsampling and addition. The fused feature is fed to the neck comprising several fused MBConv blocks and then fed to the SAM head.

Training. To initialize the image encoder, we begin by distilling the image embedding of SAM-ViT-H into EfficientViT. We employ the L2 loss as the loss function. For

	#Params(M)	#MACs(G)	Throughput (image/s)	COCO mAP
SAM-ViT-H [1]	641.1	2973	11	46.5
MobileSAM [2]	9.8	39	278	38.7
EdgeSAM [3]	9.6	20	449	42.1
EfficientSAM [4]	25.3	247	183	44.4
EfficientViT-SAM-L0	34.8	35	762	45.7
EfficientViT-SAM-L1	47.7	49	638	46.2
EfficientViT-SAM-L2	61.3	69	538	46.6
EfficientViT-SAM-XL0	117.0	185	278	47.5
EfficientViT-SAM-XL1	203.3	322	182	47.8

Table 1. **Runtime Efficiency Comparison**. We benchmark the throughput on a single NVIDIA A100 GPU with TensorRT, fp16.

	COCO			LVIS		
	1 click	3 click	5 click	1 click	3 click	5 click
SAM-ViT-H [1]	58.4	69.6	71.4	59.2	66.0	66.8
EfficientViT-SAM-XL1	59.8	71.3	75.3	56.6	67.0	71.7

Table 2. **Zero-Shot Point-Prompted Segmentation Results.**

	COCO				LVIS			
	mIoU	mIoU^{S}	mIoU^{M}	mIoU^{L}	mIoU	mIoU^{S}	mIoU^{M}	mIoU^{L}
SAM-ViT-H [1]	77.4	72.3	80.4	81.8	77.0	70.6	87.5	89.9
EfficientViT-SAM-XL1	79.9	75.8	82.2	83.8	79.9	74.4	88.4	91.6

Table 3. **Zero-Shot Instance Segmentation Results, Prompted with Ground Truth Bounding Box.**

	COCO				LVIS			
	mAP	AP^{S}	AP^{M}	AP^{L}	mAP	AP^{S}	AP^{M}	AP^{L}
SAM-ViT-H [1]	46.5	30.8	51.0	61.7	44.2	31.8	57.1	65.3
MobileSAM [2]	38.7	23.7	42.2	54.3	37.0	24.7	47.8	59.1
EdgeSAM [3]	42.1	26.6	46.7	56.9	39.8	28.6	51.3	59.3
EfficientSAM [4]	44.4	28.4	48.3	60.1	41.5	29.7	53.4	62.2
EfficientViT-SAM-L0	45.7	28.2	49.5	63.4	41.8	28.8	53.4	64.7
EfficientViT-SAM-L1	46.2	28.7	50.4	64.0	42.1	29.1	54.3	65.0
EfficientViT-SAM-L2	46.6	28.9	50.8	64.2	42.7	29.4	55.1	65.5
EfficientViT-SAM-XL0	47.5	30.0	51.5	64.6	43.9	31.2	56.2	65.9
EfficientViT-SAM-XL1	47.8	30.5	51.8	64.7	44.4	31.6	57.0	66.4

Table 4. **Zero-Shot Instance Segmentation Results, Prompted with ViTDet Boxes.**

the prompt encoder and mask decoder, we initialize them by loading the weights from SAM-ViT-H. Then, we train EfficientViT-SAM on the SA-1B dataset in an end-to-end manner.

In the end-to-end training phase, we randomly choose between the box prompt and the point prompt with equal probability. In the case of the point prompt, we randomly select 1-10 foreground points from the ground-truth mask to ensure our model performs effectively for various point configurations. In the case of the box prompt, we utilize the ground-truth bounding box. We resize the longest side to 512/1024 for EfficientViT-SAM-L/XL models and pad the shorter side accordingly. We select up to 64 randomly sampled masks per image. To supervise the training process, we use a linear combination of focal loss and dice loss, with a 20:1 ratio of focal loss to dice loss. Similar to the approach taken in SAM to mitigate ambiguity, we predict three masks simultaneously and only back-propagate the lowest loss.

We train EfficientViT-SAM on the SA-1B dataset for 2 epochs, utilizing a batch size of 256. The AdamW optimizer is employed with a momentum of β_1 = 0.9 and β_2 = 0.999. The initial learning rate is set to 2e^{-6}/1e^{-6} for EfficientViT-SAM-L/XL, which is decayed to 0 using a cosine decay learning rate schedule. Regarding data augmentation, we use the random horizontal flip.

4. Experiment

In this section, we begin by conducting a comprehensive analysis of the runtime efficiency of EfficientViT-SAM in Section 4.1. Subsequently, we evaluate the zero-shot capability of EfficientViT-SAM on the COCO [8] and LVIS [21] datasets, which were not encountered during the training process. Two distinct tasks are performed: point-prompted instance segmentation in Section 4.2 and box-prompted instance segmentation in Section 4.3. These tasks individually assess the effectiveness of the point prompt and box prompt features of EfficientViT-SAM. We also provide results on SGinW benchmark [22] in Section 4.4.

4.1. Runtime Efficiency

We compare the model parameters, MACs, and throughput of EfficientViT-SAM with SAM and other acceleration works. Results are shown in Table 1. We conduct the throughput measurements on a single NVIDIA A100 GPU with TensorRT optimization. Our results show that compared to SAM, we achieve an impressive acceleration of 17 to 69 times. Furthermore, despite having more parameters than other acceleration works, EfficientViT-SAM demonstrates significantly higher throughput due to its effective utilization of hardware-friendly operators.

4.2. Zero-Shot Point-Prompted Segmentation

We assess the zero-shot performance of EfficientViT-SAM in segmenting objects based on point prompts in Table 2. We adopt the point selection method described in [1]. That is the initial point is selected as the point located farthest from the object boundary. Each subsequent point is chosen as the farthest point from the boundary of the error region, which is defined as the area between the ground truth and the previous prediction. The performance is reported using 1/3/5 clicks on COCO and LVIS dataset, with the mIoU (mean Intersection over Union) serving as the metric. Our results demonstrate superior performance compared to SAM, particularly when additional point prompts are provided.

4.3. Zero-Shot Box-Prompted Segmentation

We evaluate the zero-shot performance of EfficientViT-SAM in object segmentation using bounding boxes. We first input ground truth bounding boxes to the model, and the results are presented in Table 3. The mIoU(mean Intersection over Union) is reported for all objects, as well as separately for small, medium, and large objects. Our approach surpasses SAM by a significant margin on the COCO and LVIS dataset. Next, we employ an object detector, ViT-Det [23], and utilize its output boxes as prompts for the model. The results in Table 4 demonstrate that EfficientViT-SAM achieves superior performance compared to SAM. Notably, even the lightest version of EfficientViT-SAM significantly outperforms other acceleration works by a large margin.

Additionally, we evaluate the performance of EfficientViT-SAM on the COCO dataset using YOLOv8 [24] and Grounding DINO [25] as the object detectors. YOLOv8 is a real-time object detector suitable for real-world applications. On the other hand, Grounding DINO is capable of detecting objects using text prompts, allowing us to perform object segmentation based on textual cues. The results presented in Table 5 reveal the outstanding performance of EfficientViT-SAM in comparison to SAM.

4.4. Zero-Shot In-the-Wild Segmentation

The Segmentation in the Wild benchmark [22] consists of 25 zero-shot in-the-wild segmentation datasets. We equip EfficientViT-SAM with Grounding DINO as box prompts and perform zero-shot segmentation. SAM achieves an mAP of 48.7, whereas EfficientViT-SAM achieves a higher score of 48.9.

	YOLOv8				GroundingDINO			
	mAP	AP^{S}	AP^{M}	AP^{L}	mAP	AP^{S}	AP^{M}	AP^{L}
SAM-ViT-H [1]	43.8	26.1	48.1	60.4	46.9	31.5	51.8	64.4
EfficientViT-SAM-XL1	44.7	26.0	48.9	62.9	48.2	31.5	52.6	67.3

Table 5. **Zero-Shot Instance Segmentation Results on COCO, Prompted with YOLOv8/Grounding DINO Boxes.**

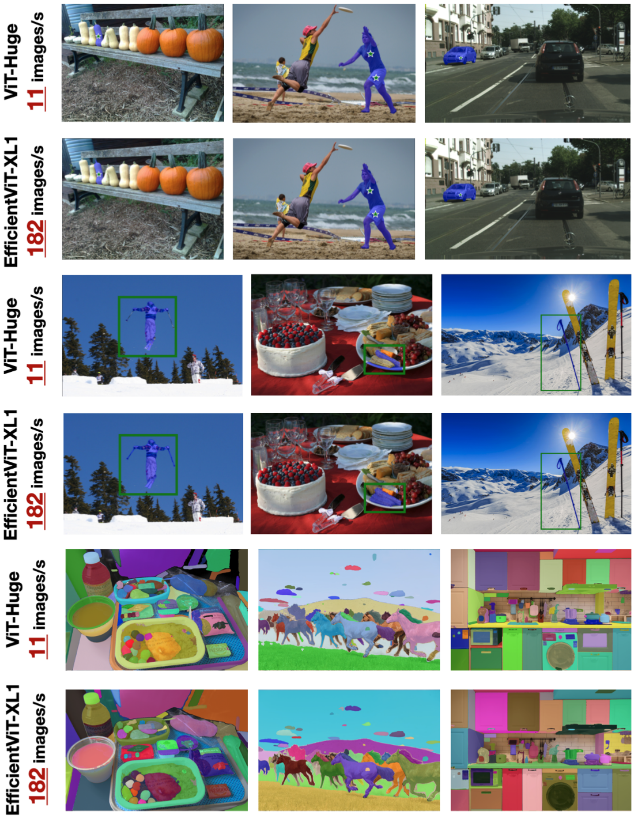

Figure 3. **Qualitative Segmentation Results of EfficientViT-SAM under Point, Box, and Everything Mode.**

4.5. Qualitative Results

Figure 3 showcases the qualitative results of EfficientViT-SAM when provided with point prompt, box prompt, and segment-everything mode. The results demonstrate that EfficientViT-SAM excels in segmenting both large and small objects.

5. Conclusion

In this work, we introduced EfficientViT-SAM, which utilizes EfficientViT to replace the image encoder of SAM. EfficientViT-SAM achieved a significant efficiency boost over SAM without sacrificing performance across various zero-shot segmentation tasks. We have open-sourced our code and pre-trained models on GitHub to the community.

References

[1] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 3, 4

[2] 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. 1, 2, 3

[3] Chong Zhou, Xiangtai Li, Chen Change Loy, and Bo Dai. Edgesam: Prompt-in-the-loop distillation for on-device deployment of sam. *arXiv preprint arXiv:2312.06660*, 2023. 1, 3

[4] 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. 1, 2, 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[6] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 2

[7] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022. 1, 2

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 3

[9] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023. 2

[10] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2

[11] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 2

[12] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023. 2

[13] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2

[14] Han Shu, Wenshuo Li, Yehui Tang, Yiman Zhang, Yihao Chen, Houqiang Li, Yunhe Wang, and Xinghao Chen. Tinysam: Pushing the envelope for efficient segment anything model. *arXiv preprint arXiv:2312.13789*, 2023. 2

[15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[16] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018. 2

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[18] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 2

[19] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2

[20] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 2

[21] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 3

[22] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127, 2023. 3, 4

[23] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 4

[24] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023. 4

[25] 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. 4