TinySAM: Pushing the Envelope for Efficient Segment Anything Model

Han Shu[1,2], Wenshuo Li[2], Yehui Tang[2], Yiman Zhang[2], Yihao Chen[2], Houqiang Li[1], Yunhe Wang[2*], Xinghao Chen[2*]

[1]University of Science and Technology of China
[2]Huawei Noah's Ark Lab
{han.shu, xinghao.chen, yunhe.wang}@huawei.com

Abstract

Recently segment anything model (SAM) has shown powerful segmentation capability and has drawn great attention in computer vision fields. Massive following works have developed various applications based on the pre-trained SAM and achieved impressive performance on downstream vision tasks. However, SAM consists of heavy architectures and requires massive computational capacity, which hinders the further application of SAM on computation constrained edge devices. To this end, in this paper we propose a framework to obtain a tiny segment anything model (TinySAM) while maintaining the strong zero-shot performance. We first propose a full-stage knowledge distillation method with hard prompt sampling and hard mask weighting strategy to distill a lightweight student model. We also adapt the post-training quantization to the prompt-based segmentation task and further reduce the computational cost. Moreover, a hierarchical segmenting everything strategy is proposed to accelerate the everything inference by $2\times$ with almost no performance degradation. With all these proposed methods, our TinySAM leads to orders of magnitude computational reduction and pushes the envelope for efficient segment anything task. Extensive experiments on various zero-shot transfer tasks demonstrate the significantly advantageous performance of our TinySAM against counterpart methods.

Code — https://github.com/xinghaochen/TinySAM

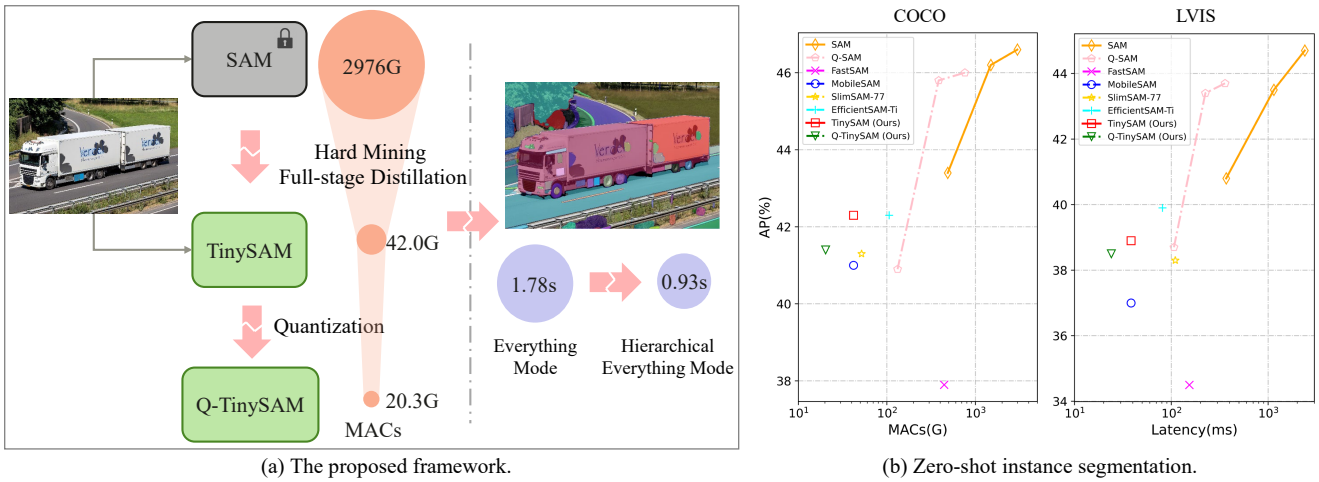
Introduction

Object segmentation is an important and foundational task in computer vision fields. Extensive visual applications such as object localization and verification rely on accurate and fast object segmentation. Tremendous prior works have focused on segmentation tasks which include semantic segmentation (Long, Shelhamer, and Darrell 2015; Strudel et al. 2021), instance segmentation (Bolya et al. 2019; Liu et al. 2018) and panoptic segmentation (Cheng et al. 2022; Kirillov et al. 2019). Recently, Kirillov *et al*. (Kirillov et al. 2023) introduce a powerful segment anything model (SAM), together with a massive segmentation dataset SA-1B that contains over 1 billion masks on 11 million images. With the strong capability to segment objects with arbitrary shapes and categories, SAM has become a foundation framework for many downstream tasks such as object tracking (Cheng et al. 2023), image inpainting (Yu et al. 2023) and 3D vision (Cen et al. 2023) *etc*. Moreover, the powerful zero-shot segmentation ability of SAM has benefited research area with less data like medical imaging (Ma and Wang 2023).

Although SAM has achieved impressive performance on downstream vision tasks, complicated architecture and huge computational cost make SAM difficult to be deployed on resource constrained devices. The inference time of SAM model for a 1024×1024 image could take up to 2 seconds on a modern GPU (Zhao et al. 2023). Some recent attempts have tried to obtain a more computation efficient segment anything model. For example, MobileSAM (Zhang et al. 2023) tries to replace the heavy component of image encoder with a lightweight architecture of TinyViT (Wu et al. 2022). However, it only accesses the image encoder network with a decoupled knowledge distillation strategy by training the compact image encoder network with the supervision of image embeddings from the teacher network. This partial training strategy inevitably causes performance decay without the supervision of final mask prediction. FastSAM (Zhao et al. 2023) transfers the segment anything task to an instance segmentation task with only one foreground category with YOLOv8 (Jocher, Chaurasia, and Qiu 2023). To fulfill the function of prompt-based segmentation, FastSAM applies a post-process strategy together with the instance segmentation network. However, this reformulated framework could not achieve comparable performance as SAM on downstream zero-shot tasks.

To further push the envelope for efficient segment anything model, in this paper we propose a full framework to obtain TinySAM that greatly reduces the computational cost while maintaining the zero-shot segmentation ability to maximum extent. Specifically, we propose a hard mining full-stage knowledge distillation method to improve the capability of the compact student network. The student network is distilled in an end-to-end manner with the supervision of teacher network from different network stages. A mask-weighted distillation loss is proposed to efficiently transfer the information from teacher to student through massive various SA-1B masks. Besides, an online hard prompt sampling strategy is proposed to make the distillation process attend more to hard examples and thus im-

*Corresponding authors.



(a) The proposed framework.

(b) Zero-shot instance segmentation.

Figure 1: (a) The overall framework of our proposed method. Consisting the modules of the hard mining full-stage knowledge distillation, the post training quantization and the hierarchical everything inference, the computation cost is down-scaled by magnitudes. (b) The proposed TinySAM can save considerable computation cost while maintaining the performance. The latency is tested with TensorRT on NVIDIA T4 GPU.

proves the final performance. We also adapt the post-training quantization to the prompt-based segmentation task and further reduce the computational cost. Moreover, we find that it takes tremendous computational cost for segmenting everything in an image since massive masks have to be generated from grid prompt points. To this end, a hierarchical segmenting everything strategy is proposed to accelerate the everything inference by $2\times$ with almost no performance degradation. With all these proposed methods, our TinySAM leads to orders of magnitude computational reduction and pushes the envelope for efficient segment anything task. For example, TinySAM can achieve $100\times$ acceleration for segment anything task compared with the original SAM. Extensive experiments on various zero-shot transfer tasks demonstrate the significantly advantageous performance of our TinySAM against counterparts.

Related Work

Segment Anything Model

Recently proposed segment anything model (SAM) (Kirillov et al. 2023) proves its generalization on object segmentation and downstream vision tasks. SAM consists of three subnetworks, *i.e.*, image encoder, prompt encoder and mask decoder. The image encoder is a heavy vision transformer-based network (Dosovitskiy et al. 2020), which extracts the input image into image embedding. The prompt encoder is designed to encode input points, boxes, arbitrary-shaped masks and free-form text with positional information. The geometric prompt and text prompt are processed with different networks. The mask decoder, which contains a two-way transformer, takes the output of image encoder and prompt encoder to generate the final mask prediction. Together with the proposed SA-1B dataset, which contains 11 million high-resolution images and more than 1 billion high-quality segmentation masks, SAM shows impressive high quality segmentation ability for objects of any category and shape. Moreover, SAM demonstrates powerful generalization on zero-shot downstream vision tasks including edge detection, object proposal, instance segmentation and text-to-mask prediction. Due to the flexible prompt mode and high quality segmentation capability, SAM has been regarded as a foundation model for vision applications. However, SAM, especially the image encoder network, consists of large parameters and requires high computation capacity for deployment. Therefore, it is not easy to apply SAM on edge devices with constrained resources. The compression and acceleration of SAM becomes an important research topic (Zhao et al. 2023; Zhang et al. 2023; Chen et al. 2024).

Knowledge Distillation

Hinton *et al.* (Hinton et al. 2015) propose the knowledge distillation method to supervise the training of lightweight student network via the output of teacher network. Since then knowledge distillation has been an important approach to improve the performance of compact networks during training process. Knowledge distillation methods can be roughly divided into two categories, *i.e.* distillation for network outputs (Hinton et al. 2015) and for intermediate features (Romero et al. 2014). Majority of research of knowledge distillation methods have focused on image classification task (Park et al. 2019; Peng et al. 2019; Dong et al. 2023; Li et al. 2022b). Subsequent works (Chen et al. 2017; Liu et al. 2019; Guo et al. 2021; Chen et al. 2020; Deng, Kong, and Murakami 2019) propose knowledge distillation methods for high-level computer vision tasks such as object detection and semantic segmentation. Zhang *et al.* (Zhang et al. 2023) propose to use the distillation method to obtain an efficient segment anything model (MobileSAM). However, MobileSAM only accesses the image encoder network with the supervision of corresponding image embeddings from original SAM. This partial distillation strategy could cause considerable performance decay since there is no guidance of mask-level information for lightweight student network from either teacher network or labeled data.

Quantization

Model quantization is also one of the commonly used model compression methods, which quantizes weights or activations from higher bit-width to lower bit-width to reduce both storage requirements and computational complexity with limited accuracy degradation. There are two types of model quantization methods, quantization-aware training (QAT) (Choi et al. 2018; Esser et al. 2019) and post-training quantization (PTQ) (Choukroun et al. 2019). QAT methods require a labeled training dataset and extensive training cost, while PTQ methods only need a small unlabeled calibration dataset and thus are more efficient. Many prior PTQ methods (Liu et al. 2023; Nagel et al. 2020) have proposed to search for appropriate quantization parameters for convolutional neural networks. As vision transformers (Dosovitskiy et al. 2020; Liu et al. 2021a) achieve remarkable performance on various visual tasks, recent works (Liu et al. 2021b; Yuan et al. 2022; Tai, Lin, and Wu 2023; Li et al. 2022c) investigate how to apply post-training quantization for vision transformers and have achieved good performance with 8-bit quantization configuration. However, there is rare exploration for quantization of prompt-based segmentation task, especially for segment anything models.

Methodology

Overview of TinySAM

This paper proposes a framework to get a highly efficient SAM, as described in Figure 1. Firstly, we introduce a hard mining full-stage knowledge distillation specifically designed for SAM. To further activate the distillation process, the proposed hard mask weighting and hard prompt sampling strategy are utilized to mine the essential knowledge from the teacher network to the student network. Secondly, a post-training quantization method is adapted to prompt-based segmentation task and applied to the lightweight student network. Thirdly, a hierarchical everything inference mode is designed for segmenting everything task, which can avoid massive redundant computation only with negligible accuracy loss and speedup the inference time by $2\times$.

Hard Mining Full-Stage Knowledge Distillation

SAM consists of three subnetworks, *i.e.* image encoder, prompt encoder and mask decoder. The image encoder network is based on vision transformer (Dosovitskiy et al. 2020) and consumes great computation cost. Inspired by MobileSAM (Zhang et al. 2023), we use the lightweight TinyViT (Wu et al. 2022) to replace the original heavy image encoder network. Considerable performance decay exists for this simple substitution. Therefore, we propose a hard mining full-stage knowledge distillation strategy to guide the lightweight image encoder during learning procedure from multiple knowledge levels.

Besides the conventional loss between the predicted results and ground-truth labels, we introduce multiple distillation losses on different stages as described in Figure 2. Specifically, we select several nodes of teacher network to guide the learning of student network from multiple level of knowledge. Firstly, we choose the output feature of image

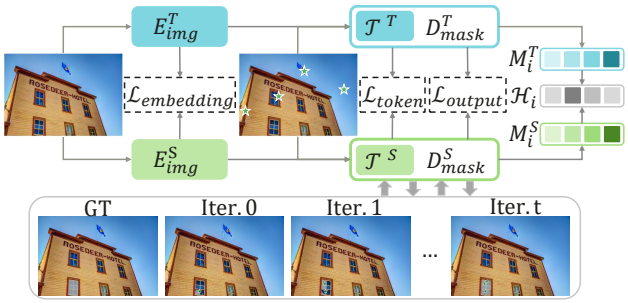


Figure 2: The framework of the hard mining full-stage knowledge distillation. For the massive masks of SA-1B dataset, we design the hard prompt sampling for prompts and hard mask weighting for distillation loss. For sampling process, the stars represent sampling point with different iterations. With the increase of iterations, the sampling region is more closed to the edge of the target mask, which makes the prompt relatively harder for student network to learn. Moreover, according to the gap between student and teacher network, different weight is assigned to each mask when calculating the distillation loss.

encoder, *i.e.* image embedding, as a distillation information. Image embedding concentrates the information from input image, which is the fundamental knowledge during the prediction. For an input image of I, the distillation loss function for image embedding can be expressed as,

$$\mathcal{L}_{embedding} = \mathcal{L}(E^T_{img}(I), E^S_{img}(I)), \tag{1}$$

where E^S_{img} and E^T_{img} denote the image encoder for student and teacher network, respectively. Since image level information does not directly relate to the mask prediction, features more close to the final output are essential for this segmentation task. Naturally, the final output of the teacher network is chosen to be a distillation point. The output distillation loss $\mathcal{L}_{output}$ can be described as,

$$\mathcal{L}_{output} = \mathcal{L}(D^T_{mask}(E^T_{img}(I), q), D^S_{mask}(E^S_{img}(I), q)), \tag{2}$$

where D^S_{mask} and D^T_{mask} are mask decoders for student and teacher, respectively. q denotes the query of the mask decoder, which is the concatenation of prompt embedding and output tokens. Since the structure of SAM is rather complicated, the previously mentioned two distillation losses could be inconsistent and thus hard for lightweight student to learn. We further propose to distill the output tokens from the two-way transformer of the mask decoder, which interacts information from prompt embedding and image embedding. It captures the target mask information in a more abstract way. The corresponding distillation losses $\mathcal{L}_{token}$ can be described as,

$$\mathcal{L}_{token} = \mathcal{L}(\mathcal{T}^T(E^T_{img}(I), q), \mathcal{T}^S(E^S_{img}(I), q)), \tag{3}$$

where $\mathcal{T}^S$ and $\mathcal{T}^T$ are the two-way transformer module of mask decoder and $\mathcal{L}$ denotes the loss function. We empirically find that the numerical values of feature difference could make the conventionally used MSE loss (ℓ_2 distance) too small to be well optimized. Thus we use ℓ_1 distance function instead. The overall distillation loss function

$\mathcal{L}_{distill}$ can be expressed as,

$$\mathcal{L}_{distill} = \alpha * \mathcal{L}_{embedding} + \beta * \mathcal{L}_{token} + \gamma * \mathcal{L}_{output}, \quad (4)$$

where α, β, γ represent the hyper-parameters for each distillation loss. The total training loss is a linear combination of distillation loss, ground truth loss for mask prediction $\mathcal{L}_{mask}$ and IoU prediction $\mathcal{L}_{ious}$, where $\mathcal{L}_{mask}$ is a combination of focal loss (Lin et al. 2017) and dice loss (Milletari, Navab, and Ahmadi 2016), $\mathcal{L}_{ious}$ is ℓ_1 loss function between predicted IoUs and calculated IoUs.

$$\mathcal{L}_{total} = \mathcal{L}_{distill} + \mathcal{L}_{mask} + \mathcal{L}_{ious}. \quad (5)$$

Hard Mask Weighting. To make the knowledge distillation more effective, we design a hard mask weighting strategy when calculating the losses. There is an observation that masks could be extremely various in a single image of SA-1B dataset since the fine-grained granularity and no semantic constraints. As shown in Figure 2, segmenting the flag with complex boundary could be difficult while segmenting the rectangular window with high contrast color could be easy. The hard mask should reasonably be assigned with larger weight for student to learn. Specifically, we calculate the gap of student and teacher network output to indicate the mask hardness $\mathcal{H}_i$.

$$\mathcal{H}_i = \text{sigmoid}(\frac{\text{IoU}(M_i^T, M_i^{GT})}{\text{IoU}(M_i^S, M_i^{GT}) + \epsilon} - 1), \quad (6)$$

where M_i^T, M_i^S, M_i^{GT} represent the mask prediction of student network, the mask prediction of teacher network and the ground truth for ith mask, respectively. Thus the distillation loss could be updated with

$$\mathcal{L}^*_{distill} = \alpha * \mathcal{L}_{embedding} + \beta * \mathcal{L}_{token} + \gamma * \sum_{i=1}^{N} \mathcal{H}_i * \mathcal{L}^i_{output}. \quad (7)$$

Hard Prompt Sampling. Generally, random sampling from labeled training data could be adopted to generate the prompts to drive the end-to-end training of prompt-based mask prediction network as SAM. To further ease the learning process of the distillation between teacher and lightweight student network, we propose a hard prompt sampling strategy, which makes the training samples concentrate in the difficult area for prediction. Taking points prompt as an example, points P_0 are initially sampled inside the labeled mask region M_{gt}. These initial points are fed into the network with input image to get the predicted mask region M_0. Then we sample the prompt points from the difference set of M_{gt} and M_0, and we conduct the procedure iteratively. The $(i+1)$-th round sampling points P_i are sampled from the difference set of M_{gt} and M_i, *i.e.*

$$P_{i+1} \in M_{gt} - M_i, i = 0, 1, 2, ... \quad (8)$$

where

$$M_i = D_{mask}(E_{prompt}(P_i), E_{img}(I)). \quad (9)$$

When applied on the training process, the i-th iteration is random sampled from 0 to 9, which makes the difficulty of sampled prompts in a constrained range. The bottom of Figure 2 shows the location change of the sampling prompts with iterations, the green stars denote the sampled point prompts with online hard prompt sampling strategy. With more iterations, the sampling points are more close to the edge region of the ground truth mask.

Quantization

Quantization aims to project floating point tensor x to b-bit integer tensor x_q with a scaling factor s. The uniform symmetric quantization could be formulated as follows,

$$x_q = Q(b, s) = \text{clip}(\text{round}(\frac{x}{s}), -2^{b-1}, 2^{b-1} - 1). \quad (10)$$

For a matrix multiplication $O = AB$, it can be quantized with two scaling factors s_A and s_B, and the quantized matrix is denoted as $\hat{O} = \hat{A}\hat{B}$. The metric for measuring the distance between $\hat{O}$ and O is vitally important for optimizing $\hat{A}$ and $\hat{B}$. Following the successful practice of quantization methods in image classification models (Tai, Lin, and Wu 2023; Yuan et al. 2022; Frantar et al. 2022; Wu et al. 2020), we perform hessian guided metric as the distance to solve the scaling factors, which is more consistent with task loss. Different from classification tasks, the prompt-based segmentation task of SAM outputs segmentation predictions which contains fine-grained masks. Thus we use the Kullback-Leible (KL) divergence of masks and IoUs as the task loss and use some calibration data to calculate the hessian matrix, the task loss is formulated as,

$$L = \text{KL}(\hat{y}_{pred}, y_{pred}) + \text{KL}(\hat{y}_{iou}, y_{iou}), \quad (11)$$

where y_{pred} and y_{iou} are the outputs of the floating point model, $\hat{y}_{pred}$ and $\hat{y}_{iou}$ are the outputs after quantization.

After specifying the distance metric, we could solve s_A and s_B as an alternate iterative grid search problem. With calibration data we get the maximum value of A and B, which is A_{max} and B_{max} respectively, and use two parameters θ_l and θ_u to specify the search range for s_A and s_B, $[\theta_l \frac{A_{max}}{2^{b-1}}, \theta_u \frac{A_{max}}{2^{b-1}}]$ and $[\theta_l \frac{B_{max}}{2^{b-1}}, \theta_u \frac{B_{max}}{2^{b-1}}]$. These two search ranges are linearly divided into n candidate options separately. $\hat{A}$ and $\hat{B}$ are optimized in an alternate manner.

The input of matrix multiplication after softmax is unevenly distributed at both ends of the interval [0,1], while the feature after GELU varies greatly between the positive and negative ranges. These two circumstances go far from the assumption of uniform quantization, *i.e.*, the activation in neural networks obeys Gaussian distribution. The violation will result in high quantization error. Thus we split feature into two groups and use two scaling factors to reduce the quantization error.

Hierarchical Segmenting Everything

SAM proposes an automatic mask generator which samples points as a grid to segment everything. However, we find that dense point grid leads to over fine-grained segmentation results and also occupies massive computing resources. On the one hand, for a complete object, too many sampling points may cause slightly different parts of the object to be incorrectly segmented as separate masks. On the other hand, since the image encoder has been largely shrunk by the proposed method, the time cost of everything mode inference is mainly in the mask decoder part. For the default setting of SAM automatic mask generator, it samples $32 \times 32 = 1024$ points as the prompts, which means the mask decoder is inferred by 1024 times. It costs 16ms for image encoder and 894ms for mask decoder on a single V100 GPU.

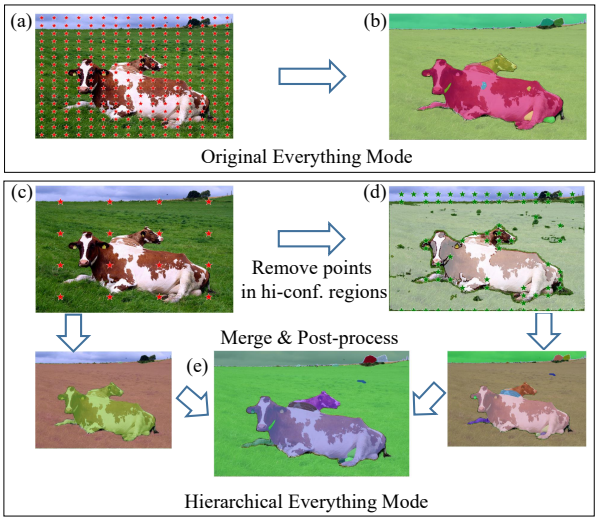


Figure 3: Comparison between our hierarchical strategy and the original strategy. (a) Points sampling (take *points_per_side=16* as an example) of original everything mode. (b) Segmentation results of original strategy. (c) First step of our hierarchical strategy, only $1/16$ points are sampled. (d) Get high confidence area from (c) and ignore points in this area. The high confidence area is shown as white mask. (e) Segmentation results of our hierarchical strategy.

To reduce the time cost of everything mode, we propose a hierarchical mask generating method. The comparison between our hierarchical strategy and the original one is shown in Figure 3. Different from original everything mode, in the first step we only use $1/4$ points in each side so the total points is $1/16$ of the original settings, as shown in Figure 3(c). Then we infer the prompt encoder and mask decoder with these prompts and get the results.

Then we filter out some masks with confidence exceeding a threshold τ, and mark the corresponding regions as areas that could be considered as final predictions. For these areas, since they are considered as the segmentation results of instances with high confidences, there is no need to regenerate point prompts. Thus we sample points as the same density with original setting but ignore points in the above area. As shown in Figure 3(d), most points on the grass and body of the front cow are ignored. Meanwhile, the points on the back cow and the sky are kept to be further segmented. Specifically, the back cow is incorrectly segmented as the same object with the front cow in the initial round. This strategy can avoid redundant cost of inference time and over fine-grained segmentation of the object. Then we utilize the point prompts sampled in the second round to get the mask predictions. Finally, the results of these two round are merged and post-processed to get the final masks. More than 50% points are ignored by our method thus brings in significant latency reduction.

Experiments

Implementation Details

We utilize the TinyViT-5M (Wu et al. 2022) as the lightweight student image encoder and SAM-H as the teacher model, following prior work (Zhang et al. 2023). 1% of SA-1B dataset is used as the training data for full-stage distillation. We adopt Adam optimizer and train the student network for 8 epochs. For each iteration, we sample 64 prompts according to hard prompt sampling strategy. To accelerate the distillation process, the image embeddings from the teacher network have been computed and stored in advance. Therefore, the heavy image encoder of teacher network is not necessary to compute repeatedly during training time. For post training quantization, we set $\theta_l = 0.01, \theta_u = 1.2, n = 100, rounds = 3$ for iterative search. We calibrate quantized model on SA-1B dataset using 8 images. We conduct zero-shot evaluation on downstream tasks like instance segmentation and point prompt segmentation. Following the suggestions by SAM (Kirillov et al. 2023), the multi-output mode is adopted and the final mask prediction is the one with highest IoU prediction.

Zero-Shot Instance Segmentation

For zero-shot instance segmentation task, we strictly follow the experimental settings of SAM and use the object detection results of ViTDet-H (Li et al. 2022a) as the box prompt for instance segmentation. We evaluate the zero-shot instance segmentation task for models on the benchmark of COCO (Lin et al. 2014) dataset and LVIS v1 (Gupta, Dollar, and Girshick 2019). We compare our TinySAM with different variants of SAM (Kirillov et al. 2023), and also with prior efficient models like FastSAM (Zhao et al. 2023), MobileSAM (Zhang et al. 2023), EfficientSAM (Xiong et al. 2024) and SlimSAM (Chen et al. 2024). As shown in Table 1, the proposed TinySAM obtained superior performance when compared with prior methods. Specifically, our TinySAM outperforms FastSAM (Zhao et al. 2023) in terms of MACs and instance segmentation accuracy, *i.e.*, about 4% AP improvement with only 9.5% MACs and 25% latency. With the same computational cost, our TinySAM also achieves $1.3\%+$ AP on COCO dataset than MobileSAM (Zhang et al. 2023) and $1.9\%+$ AP on LVIS v1 dataset, respectively. With similar performance on COCO dataset, TinySAM is $2\times$ faster than EfficientSAM (Xiong et al. 2024). Our W8A8 quantized variant of TinySAM (Q-TinySAM) also obtains competitive performance across different methods. Specifically, Q-TinySAM achieves $0.1\%+$ AP on COCO and $0.2\%+$ on LVIS v1 dataset than SlimSAM (Chen et al. 2024), with only 39% MACs and 21.8% latency. Visual results on COCO validation set and LVIS dataset shows that our proposed TinySAM captures more clear and smooth boundaries compared with other efficient variants of SAM.

Zero-shot Points Valid Mask Evaluation

In this section, we evaluate the performance of our TinySAM for segmenting an object from several points as the prompts. We use the same points selection metric as previous work (Kirillov et al. 2023; Gupta, Dollar, and Girshick 2019), which calculates the distance transform of false positive and false negative masks, and then sample points at a maximal value. We calculate the mIoU of each dataset to evaluate the performance of different models.

Method	MACs	Lat.(ms)	COCO AP	COCO AP^S	COCO AP^M	COCO AP^L	LVIS v1 AP	LVIS v1 AP^S	LVIS v1 AP^M	LVIS v1 AP^L
ViTDet-H (Li et al. 2022a)	-	-	51.0	32.0	54.3	68.9	46.6	35.0	58.0	66.3
zero-shot transfer methods (segmentation module only):										
SAM-H (Kirillov et al. 2023)	2976G	2392	46.6	30.8	51.0	61.7	44.7	32.5	57.6	65.5
SAM-L (Kirillov et al. 2023)	1491G	1146	46.2	30.2	50.1	60.5	43.5	31.1	56.3	65.1
SAM-B (Kirillov et al. 2023)	487G	368.8	43.4	28.5	45.5	53.4	40.8	29.1	52.8	60.7
FastSAM (Zhao et al. 2023)	443G	153.6	37.9	23.9	43.4	50.0	34.5	24.6	46.2	50.8
EfficientSAM-Ti (Xiong et al. 2024)	106G	81.0	42.3	26.7	46.2	57.4	39.9	28.9	51.0	59.9
SlimSAM-77 (Chen et al. 2024)	51.7G	110	41.3	25.7	44.9	57.4	38.3	26.7	49.7	59.0
MobileSAM (Zhang et al. 2023)	42.0G	38.4	41.0	24.4	44.5	58.6	37.0	24.7	47.8	59.1
TinySAM (Ours)	**42.0G**	**38.4**	**42.3**	26.3	45.8	58.8	**38.9**	27.0	50.3	60.2
Q-TinySAM (Ours)	**20.3G**	**24.0**	**41.4**	25.6	45.1	57.9	**38.5**	26.6	49.8	59.8

Table 1: Zero-shot instance segmentation results on COCO and LVIS v1 dataset. Zero-shot transfer methods are prompted with the detection boxes from fully-supervised ViTDet model. TinySAM and quantized Q-TinySAM demonstrate advantageous performance on average precision. The latency is tested on NVIDIA T4 GPU.

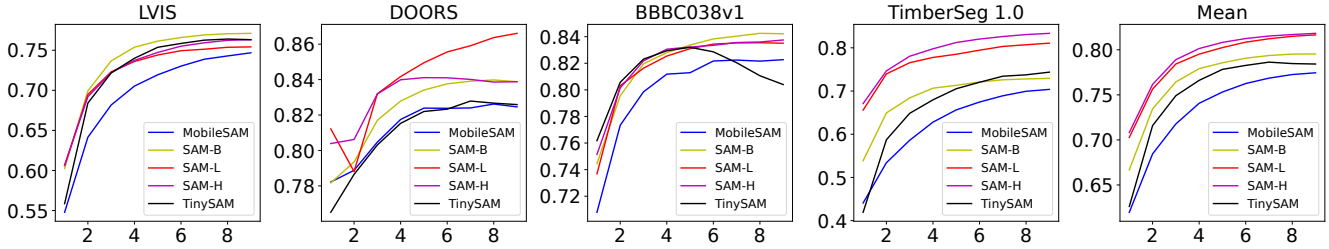


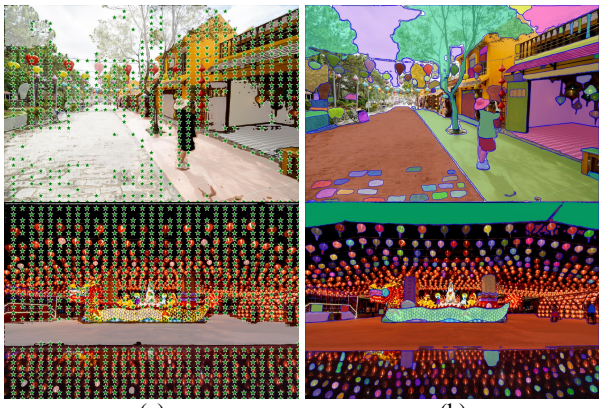
Figure 4: Results of zero-shot points valid mask evaluation. X-axis represents the number of prompts points and Y-axis represents the mIoU across all masks. The proposed TinySAM outperforms MobileSAM and achieves results close to SAM-B.

Strategy	Model	mIoU	Time (s)
Original	MobileSAM	0.5963	1.6719
Hierarchical (Ours)	MobileSAM	0.5958	0.8462
Original	SAM-H	0.7047	2.4549
Hierarchical (Ours)	SAM-H	0.7055	1.3537
Original	TinySAM	0.6137	1.7790
Hierarchical (Ours)	TinySAM	0.6061	0.9303

Table 2: Comparison of original point grid strategy and our hierarchical strategy. Evaluation on the first 100 images of COCO val2017 set.

We choose a subset of total 23 datasets used in (Kirillov et al. 2023) for efficient evaluation, which contains BBBC038v1 (Caicedo et al. 2019), DOORS (Pugliatti and Topputo 2022), TimberSeg (Fortin et al. 2022) and LVIS (Gupta, Dollar, and Girshick 2019). To make fair comparison, we follow the settings of SAM (Kirillov et al. 2023) to sample the images and masks, and the first N masks in the corresponding split are used in the evaluation.

The evaluation results are shown in Figure 4. Our TinySAM outperforms MobileSAM (Zhang et al. 2023) significantly on LVIS and TimberSeg dataset and obtains similar performance on DOORS dataset. Moreover, TinySAM achieves better results on BBBC038v1 when fewer points are utilized as prompts. We also report the mean IoU of all four datasets, as shown in the right of Figure 4. The proposed TinySAM achieves higher mIoU than MobileSAM and obtains close performance to that of SAM-B.



(a) (b)

Figure 5: Visualization for the process hierarchical everything strategy. (a) shows the intermediate result of high-confidence regions after 1st sparse prompt points with white mask and remained 2nd dense prompt points with green stars. (b) shows the final segmentation result and the small objects can be accurately segmented.

Everything Mode Acceleration

We evaluate our proposed hierarchical everything inference strategy on COCO validation set. Latency benchmarks are conducted on a single NVIDIA V100 GPU for everything mode. We sample 100 images with the least *img_id* from val2017 and conduct everything mode inference on these images. The threshold values used in the everything mode are all kept the same as default. The results are shown in

Ind.	Settings	AP (%)
0	Baseline	40.7
1	+ Knowledge Distillation Loss	41.4
2	+ Hard Prompt Sampling	41.9
3	+ Hard Mask Weighting	**42.3**
4	+ Quantization	41.4

Table 3: Effect of distillation loss, online hard prompt sampling and quantization respectively, evaluated on zero-shot instance segmentation on COCO validation dataset.

Embedding Loss	Token Loss	Output Loss	AP (%)
-	-	✓	41.6
✓	-	✓	41.7
✓	✓	✓	41.9
✓	✓	✓(HMW)	**42.3**

Table 4: Ablation study on combinations of knowledge distillation losses for zero-shot instance segmentation on COCO val set.

Table 2. We apply the same threshold and stability score on the same model evaluated with different strategies to make a fair comparison, but they can be different between these models. Our hierarchical strategy achieves comparable results compared with original 32×32 points grid strategy while the cost of inference time is reduced by about 50%. Figure 5 shows the intermediate visual results of the hierarchical strategy. We can see that the 1st round of sparse inference has segmented and removed the large objects, the remained points focus more on the small objects. This self-adaptive hierarchical strategy efficiently reduces the computation redundancy and maintains the high accuracy.

Ablation Studies

In this section, we conduct ablation studies of the proposed method on zero-shot instance segmentation task on COCO validation dataset. The experimental setting is the same as described in zero-shot instance segmentation.

Impacts of different modules. We first evaluate the effects of different modules, *i.e.*, full-stage knowledge distillation loss, hard prompt sampling, hard mask weighting and post quantization, respectively. As shown in Table 3, utilizing our proposed full-stage distillation strategy improve the performance from 40.7% to 41.4%. Incorporated with the online hard prompt sampling strategy, our method could obtain 0.5% AP gain. With the hard mask weighting loss, the performance can further increase to 42.3%. Using post-training quantization results in 0.9% AP degradation but greatly reduces the computational cost.

Impacts of different distillation losses. For detailed full-stage knowledge distillation process, we investigate the necessity of the proposed three-level distillation from the teacher network. Table 4 shows the ablation results with different combinations of distillation losses. The output distillation loss takes important part since it is close to the supervision information and the similarity with teacher network directly reflects in the evaluation metric. Token loss and embedding loss both prove to be beneficial since they are related to key nodes of teacher network, which reflects the image-level information and the interaction of prompts with the image, respectively. Hard mask weighting for output loss can further boost the performance.

Points per side 1st/2nd	Thresh. τ	mIoU	Time (s)
4/16	8.5	0.5521	0.3571
8/32	**8.5**	**0.6061**	**0.9303**
10/32	8.5	0.6078	1.2774
8/32	7.0	0.6018	0.8154
8/32	10.0	0.6067	1.1819
32/-	–	0.6137	1.7790

Table 5: Ablation on point density and threshold for hierarchical strategy.

Model	AP (%)	MACs (G)
MobileSAM	41.0	42.0
+ W8A8	39.8	20.28
+ W6A6	36.3	18.45
TinySAM (Ours)	42.3	42.0
+ W8A8	41.4	20.28
+ W6A6	39.0	18.45

Table 6: Ablation study for different bit width of quantization for zero-shot instance segmentation on COCO val set.

Point density and threshold for hierarchical strategy. In Table 5, we conduct ablation study with different settings of point density and high-confidence mask threshold τ. More points and higher threshold τ lead to more precise results but longer inference time. The point density of 2nd round is more sensitive compared to the 1st one. Considering both accuracy and efficiency, the setting in bold is a good balance and used for other experiments of everything inference.

Different bits for quantization. We here explore the influence of different bit width. Table 6 reports the average precision on COCO dataset. From the results, we can conclude that quantization to 8-bit results in only slight performance drop. We also demonstrate the performance by further reducing the quantization bit width to 6.

Conclusion

In this paper, we propose a framework to push the envelope for segment anything task and obtain a highly efficient model named TinySAM. We first propose a full-stage knowledge distillation method with hard mask weighting and hard prompt sampling strategy to distill a lightweight student model. We also adapt the post-training quantization to the prompt-based segmentation task and further reducing the computational cost. Moreover, a hierarchical segmenting everything strategy is proposed to accelerate the everything inference by $2\times$ with almost no performance degradation. With all these proposed methods, our TinySAM leads to orders of magnitude computational reduction and push the envelope for efficient segment anything task. Extensive experiments on various zero-shot transfer tasks demonstrate the significantly advantageous performance of our TinySAM against counterpart methods. We hope the proposed TinySAM brings beneficial perspective for designing a highly efficient segment anything model.

References

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9157–9166.

Caicedo, J. C.; Goodman, A.; Karhohs, K. W.; Cimini, B. A.; Ackerman, J.; Haghighi, M.; Heng, C.; Becker, T.; Doan, M.; McQuin, C.; et al. 2019. Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. *Nature methods*, 16(12): 1247–1253.

Cen, J.; Zhou, Z.; Fang, J.; Shen, W.; Xie, L.; Zhang, X.; and Tian, Q. 2023. Segment anything in 3d with nerfs. *arXiv preprint arXiv:2304.12308*.

Chen, G.; Choi, W.; Yu, X.; Han, T.; and Chandraker, M. 2017. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30.

Chen, X.; Zhang, Y.; Wang, Y.; Shu, H.; Xu, C.; and Xu, C. 2020. Optical flow distillation: Towards efficient and stable video style transfer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 614–630. Springer.

Chen, Z.; Fang, G.; Ma, X.; and Wang, X. 2024. SlimSAM: 0.1% Data Makes Segment Anything Slim. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cheng, Y.; Li, L.; Xu, Y.; Li, X.; Yang, Z.; Wang, W.; and Yang, Y. 2023. Segment and track anything. *arXiv preprint arXiv:2305.06558*.

Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P. I.-J.; Srinivasan, V.; and Gopalakrishnan, K. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*.

Choukroun, Y.; Kravchik, E.; Yang, F.; and Kisilev, P. 2019. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 3009–3018. IEEE.

Deng, Z.; Kong, Q.; and Murakami, T. 2019. Towards Efficient Instance Segmentation with Hierarchical Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Dong, M.; Chen, X.; Wang, Y.; and Xu, C. 2023. Improving Lightweight AdderNet via Distillation From ℓ_2 to ℓ_1-norm. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 32: 5524–5536.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Esser, S. K.; McKinstry, J. L.; Bablani, D.; Appuswamy, R.; and Modha, D. S. 2019. Learned step size quantization. *arXiv preprint arXiv:1902.08153*.

Fortin, J.-M.; Gamache, O.; Grondin, V.; Pomerleau, F.; and Giguère, P. 2022. Instance segmentation for autonomous log grasping in forestry operations. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6064–6071. IEEE.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Guo, J.; Han, K.; Wang, Y.; Wu, H.; Chen, X.; Xu, C.; and Xu, C. 2021. Distilling object detectors via decoupled features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2154–2164.

Gupta, A.; Dollar, P.; and Girshick, R. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5356–5364.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Jocher, G.; Chaurasia, A.; and Qiu, J. 2023. YOLO by Ultralytics. https://github.com/ultralytics/ultralytics.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Li, J.; Yang, T.; Ji, W.; Wang, J.; and Cheng, L. 2022a. Exploring denoised cross-video contrast for weakly-supervised temporal action localization. In *CVPR*, 19914–19924.

Li, Y.; Chen, X.; Dong, M.; Tang, Y.; Wang, Y.; and Xu, C. 2022b. Spatial-channel token distillation for vision mlps. In *International Conference on Machine Learning*, 12685–12695. PMLR.

Li, Y.; Xu, S.; Zhang, B.; Cao, X.; Gao, P.; and Guo, G. 2022c. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in Neural Information Processing Systems*, 35: 34451–34463.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, J.; Niu, L.; Yuan, Z.; Yang, D.; Wang, X.; and Liu, W. 2023. Pd-quant: Post-training quantization based on prediction difference metric. In *CVPR*, 24427–24437.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759–8768.

Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; and Wang, J. 2019. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2604–2613.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021b. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34: 28092–28103.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ma, J.; and Wang, B. 2023. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. IEEE.

Nagel, M.; Amjad, R. A.; Van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, 7197–7206. PMLR.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5007–5016.

Pugliatti, M.; and Topputo, F. 2022. DOORS: Dataset fOr bOuldeRs Segmentation. *Zenodo*, 9: 20.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7262–7272.

Tai, Y.-S.; Lin, M.-G.; and Wu, A.-Y. A. 2023. TSPTQ-ViT: Two-scaled post-training quantization for vision transformer. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Wu, D.; Tang, Q.; Zhao, Y.; Zhang, M.; Fu, Y.; and Zhang, D. 2020. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*.

Wu, K.; Zhang, J.; Peng, H.; Liu, M.; Xiao, B.; Fu, J.; and Yuan, L. 2022. Tinyvit: Fast pretraining distillation for small vision transformers. In *European Conference on Computer Vision*, 68–85. Springer.

Xiong, Y.; Varadarajan, B.; Wu, L.; Xiang, X.; Xiao, F.; Zhu, C.; Dai, X.; Wang, D.; Sun, F.; Iandola, F.; et al. 2024. Efficientsam: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16111–16121.

Yu, T.; Feng, R.; Feng, R.; Liu, J.; Jin, X.; Zeng, W.; and Chen, Z. 2023. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*.

Yuan, Z.; Xue, C.; Chen, Y.; Wu, Q.; and Sun, G. 2022. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV*, 191–207. Springer.

Zhang, C.; Han, D.; Qiao, Y.; Kim, J. U.; Bae, S.-H.; Lee, S.; and Hong, C. S. 2023. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. arXiv:2306.14289.

Zhao, X.; Ding, W.; An, Y.; Du, Y.; Yu, T.; Li, M.; Tang, M.; and Wang, J. 2023. Fast Segment Anything. arXiv:2306.12156.