# EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything

Yunyang Xiong, Bala Varadarajan*, Lemeng Wu*, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu,
Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, Vikas Chandra
Meta AI Research
https://yformer.github.io/efficient-sam/

## Abstract

*Segment Anything Model (SAM) has emerged as a powerful tool for numerous vision applications. A key component that drives the impressive performance for zero-shot transfer and high versatility is a super large Transformer model trained on the extensive high-quality SA-1B dataset. While beneficial, the huge computation cost of SAM model has limited its applications to wider real-world applications. To address this limitation, we propose EfficientSAMs, light-weight SAM models that exhibits decent performance with largely reduced complexity. Our idea is based on leveraging masked image pretraining, SAMI, which learns to reconstruct features from SAM image encoder for effective visual representation learning. Further, we take SAMI-pretrained light-weight image encoders and mask decoder to build EfficientSAMs, and finetune the models on SA-1B for segment anything task. We perform evaluations on multiple vision tasks including image classification, object detection, instance segmentation, and semantic segmentation, and find that our proposed pretraining method, SAMI, consistently outperforms other masked image pretraining methods. On segment anything task such as zero-shot instance segmentation, our EfficientSAMs with SAMI-pretrained lightweight image encoders perform favorably with a significant gain (e.g., ∼4 AP on COCO/LVIS) over other fast SAM models. Our EfficientSAM code and models are available at here.*

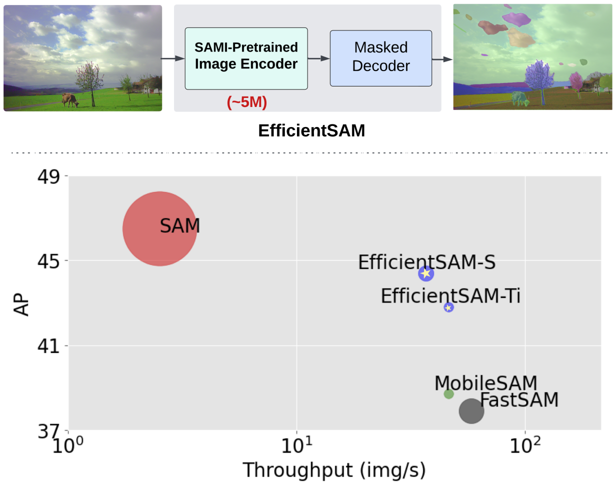

Figure 1. The comparative analysis result. (Top) The overview of EfficientSAM model by taking a well-pretrained light-weight image encoder for instance segmentation with largely reduced complexity. (Bottom) Throughput/Parameter/Performance comparison of EfficientSAM, MobileSAM, FastSAM, and SAM for zero-shot instance segmentation on COCO. We benchmark throughput (image per second) of all models on a single NVIDIA A100 with one box prompt. The input image resolution is 1024 × 1024. Our EfficientSAMs outperform MobileSAM and FastSAM by a large margin, ∼4 AP, with comparable complexity. Our EfficientSAM-S reduces the inference time of SAM by ∼20x and the parameter size by ∼20x with a small performance drop, 44.4 AP vs 46.5 AP.

## 1. Introduction

Segment Anything Model (SAM) [31] has been very successful in the vision field, achieving state-of-the-art performance in a variety of image segmentation tasks such as zero-shot edge detection[1, 31], zero-shot object proposal generation[31, 54], and zero-shot instance segmentation[31], and many other real-world applications[24, 37, 41, 50–52]. The key feature of SAM is a prompt-based Vision Transformer (ViT)[19]model trained on a large-scale visual dataset with more than 1B masks from 11M images, SA-1B[31], which allows segmenting any object on a given image. This ability of *Segment Anything* makes SAM a foundation model in vision and enables its applications even beyond vision.

Despite the forgoing advantages, the model of SAM turns out to be a major efficiency bottleneck for practical deployment since the architecture of SAM, especially, the image encoder (e.g., ViT-H) is very expensive. Note that ViT-H image encoder in SAM has 632M parameters while the

*Joint second author.

prompt-based decoder only take 3.87M parameters. As a result, it leads to high computation and memory costs when using SAM to perform segment anything tasks in practice, which makes it challenging for real-time applications.

To address this challenge, several recent works have proposed strategies that avoid incurring the huge cost when applying SAM for the prompt-based instance segmentation. For example, [68] suggests distilling the knowledge from the default ViT-H image encoder to a tiny ViT image encoder. In [71], the computation cost can be reduced with a real-time CNN-based architecture for *Segment Anything* task.

In this paper, we propose using a well-pretrained lightweight ViT image encoder (e.g., ViT-Tiny/-Small[53]) to reduce the complexity of SAM while maintain decent performance. Our method, SAM-leveraged masked image pertraining (SAMI), produces desired pretrained lightweight ViT backbones for segment anything task. This is achieved by leveraging the celebrated MAE[26] pretraining method with SAM model to obtain high-quality pretrained ViT encoders. Specifically, our SAMI makes use of SAM encoder, ViT-H, to generate feature embedding and train a masked image model with lightweight encoders to reconstruct features from ViT-H of SAM instead of image patches. This leads to generalized ViT backbones, which can be used for downstream tasks such as image classification, object detection, and segment anything. Then we finetune the pretrained lightweight encoders with SAM decoders for segment anything task[31].

To evaluate our method, we consider a transfer learning setting for masked image pretraining, where models are first pretrained with a reconstructive loss on ImageNet with image resolution 224 × 224, and then finetuned on target tasks using supervised data. Our SAMI learns lightweight encoders that generalize well. With SAMI pretraining, we can train models like ViT-Tiny/-Small/-Base on ImageNet-1K with improved generalization performance. For a ViT-Small model, we achieve 82.7% top-1 accuracy when finetuned on ImageNet-1K with 100 epochs, which outperforms other state-of-the-art image pretraining baselines. We also finetune our pretrained models on object detection, instance segmentation, and semantic segmentation. Across all these tasks, our pretraining method achieves better results than other pretraining baselines, and more importantly, we observe significant gains for small models. Further, we evaluate our models on Segment Anything task. On zero-shot instance segmentation, our model performs well compared to recent lightweight SAM methods, including FastSAM, by a margin of 4.1 /5.2 AP on COCO/LVIS.

Our main contribution can be summarized as follows:

- We propose a SAM-leveraged masked image pretraining framework called **SAMI**, which trains the model to reconstruct features from SAM ViT-H image encoder. We show that this can substantially improve the performance of image masked pretraining method.
- We demonstrate that SAMI-pretrained backbones can generalize well to many tasks including image classification, object detection, and semantic segmentation.
- We deliver EfficientSAMs, light-weight SAM models with state-of-the-art quality-efficiency trade-offs (Fig. 1), which is complementary to SAM for practical deployment.

## 2. Related Work

We briefly review relevant works on segment anything model, vision transformers, knowledge distillation, and masked image pretraining.

### 2.1. Segment Anything Model

SAM[31] has been considered as a milestone vision foundation model, which can segment any object in the image based on interaction prompts. SAM has shown remarkable zero-shot transfer performance and high versatility for many vision tasks including a variety of segmentation application[7, 8, 10, 17], in-painting[67], image restoration[29], image editing[21], image shadow removal[69], object tracking[14, 65], and 3D object reconstruction[49]. There are many other works attempting to generalize SAM to real-world scenarios, including medical image segmentation[41], camouflaged object detection[51], transparent object detection[24], concept-based explaination[50], semantic communitation[52], and helping people with visual impairments[37]. Due to its wide real-world applications, practical deployment of SAM has also gained increasing attention. Several recent works including [68, 71] have proposed strategies to reduce the computation costs of SAM. FastSAM[68] develops a CNN-based architecture, YOLOv8-seg[30], to segment all objects in an image for efficiency improvement. MobileSAM[71] presents a decoupled distillation for obtaining a lightweight image encoder of SAM. Our work focuses on dealing with this efficiency issue for practical deployment of SAM.

### 2.2. Vision Transformers

ViTs [19] have achieved impressive performance in vision applications[5, 20, 26, 34, 39, 44]. ViTs demonstrate the advantages of and generalization over their CNN counterparts[26]. There are also a number of works on efficient ViTs for deployment. Smaller ViTs such as ViT-Small/Deit-Small and ViT-Tiny/DeiT-Tiny are introduced in [53] for complementing ViT-Huge, ViT-Large, and ViT-Base in [19]. Motivated by the ability of convolution to capture local information with reduced parameters and computation costs, MobileViT[42] explore combining ViT with convolutions, which outperforms light-weight CNN models such as MobileNet-v2/v3[32, 48] with better task-level generalization properties and reduced memory size and computation cost. This trick has been used in many follow-up works including LeViT[22], EfficientFormer[35], Next-ViT[33], Tiny-ViT[61], Castling-ViT[66], EfficientViT[38]. This line of progress for designing efficient ViTs is orthogonal to our EfficientSAM work towards building efficient SAM.

### 2.3. Knowledge Distillation

Knowledge distillation (KD) is a technique to improve the performance of deep learning models without changing their architectures. [27] is a pioneering work to distill the dark knowledge from a larger teacher model to a smaller student model. The learning of a student model is supervised by the hard labels and the soft labels from a teacher model. This practice is followed by multiple works which aim to make better use of soft labels to transfer more knowledge. In [64], the distillation method decouples representation learning and classification. Decoupled knowledge distillation[70] separates the classical KD loss into two parts, target class knowledge distillation and non-target class knowledge distillation, which improves the effectiveness and flexibility of knowledge transfer. Another line of work is to transfer knowledge from intermediate features. FitNet [47] is a pioneering work by distilling the semantic information from the teacher model's intermediate feature directly. In [60], a self-supervised teaching assistant (SSTA) is introduced to guide the learning of a ViT-based student model with a supervised teacher together. [2] studies the potential of knowledge distillation from pre-training MAE model by aligning the intermediate features between the larger MAE teacher model and smaller MAE student model.

### 2.4. Masked Image Pretraining

Self-supervised pretraining approaches [6] have attracted significant attention in computer vision. One line of work is contrastive learning methods[9, 11, 57, 62], which learn augmentation in-variance by imposing high similarity between different augmented views of a given image. While the learned representation show good properties such as high linear separability, contrastive learning methods relies on strong augmentation and negative sampling. Another interesting line of work is masked image modeling (MIM), which helps models learn meaningful representations by reconstructing masked image patches. The MIM pioneering works focus on using denoising autoencoders[56] and context encoders[43] to train vision Transformer with masked prediction objectives. There are various promising works on using MIM for self-supervised image pretraining. BEiT[3] is the first one that adopts MIM for ViT pretraining to predict visual tokens. In BEiTv2[44], a semantic-rich image tokenizer is utilized for a better reconstruction target. In MaskFeat[59], reconstructing the local gradient features generated from HOG descriptor leads to effective visual pretraining. In SimMIM[63] and MAE[26], directly reconstructing the pixel values of the masked image patches achieves effective visual representation learning. There are MAE-based follow-up works that use large teacher models to guide MAE pretraining[2, 28, 60]. Our work is built on MAE and finds that leveraging MAE to reconstruct the features from SAM image encoder enables the pretraining to be highly effective.

## 3. Approach

### 3.1. Preliminary

**Masked Autoencoders.** Masked Autoencoders (MAE) model has two components, an encoder and a decoder. Both encoder and decoder are built on Transformer layers[55]. MAE takes image tokens, i.e., non-overlapping patches from the input images, as input. These input tokens are grouped to unmasked tokens and masked tokens with a given masking ratio. The unmasked tokens will be kept for encoder to extract features and the masked tokens will be set as the learning targets of the MAE decoder that need to be reconstructed during self-superivsed learning (MIM). MAE[26] adopts a high mask ratio (e.g., 75%), which prevents information leakage (e.g., simply extrapolating masked pixels based on the neighbors) in the pretraining stage.

### 3.2. SAM-Leveraged Masked Image Pretraining

We now adapt MAE framework to obtain efficient image encoders for segment anything model. Motivated by the high versatility of SAM [31], we explore latent features from SAM image encoder as the reconstruction target to leverage MAE. Our method emphasizes transferring the knowledge embedded in SAM. Fig. 2 (top) illustrates an overview of the proposed SAM-leveraged masked image pretraining, SAMI. The encoder transforms the unmasked tokens into latent feature representation and the decoder reconstructs the representation of the masked tokens aided by the output feature embedding from the encoder. The representation learning of reconstruction is guided by latent features from SAM.

**Cross-Attention Decoder.** With the supervision of SAM features, we observe that only masked tokens need to be reconstructed via decoder while the encoder's output can serve as anchors during the reconstruction. In the cross-attention decoder, queries come from masked tokens, and keys and values derive from both unmasked features from encoder and masked features. We merge the output features of masked tokens from cross-attention decoder and the output features of unmasked tokens from encoder for the MAE output embedding. Then, this combined features will be reordered to the original positions of input image tokens for the final MAE outputs.

**Linear Projection Head.** We obtain the image output through our encoder and cross-attention decoder. Then we feed such features into a small project head for aligning the features from SAM image encoder. For simplicity, we just use a linear projection head to address the feature dimension

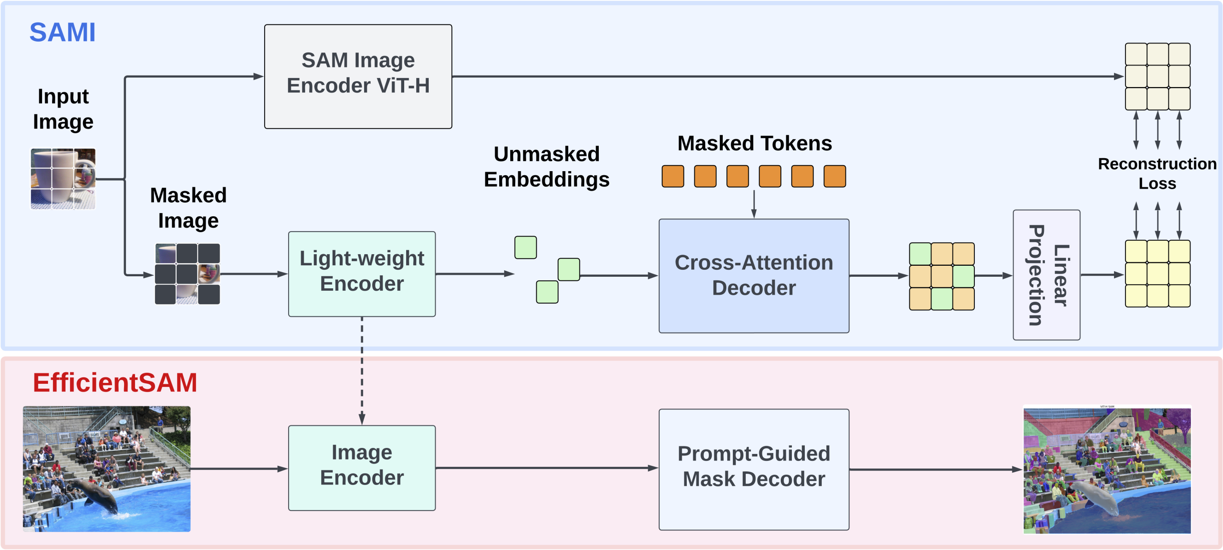

Figure 2. The overview of EfficientSAM framework. Our proposed EfficientSAM contains two stages: SAMI pretraining (top) on ImageNet and SAM finetuning (bottom) on SA-1B. For SAMI pretraining, the masked autoencoder takes the feature embeddings from SAM image encoder as the reconstruction target. After SAMI pretraining, the decoder is discarded and the light-weight encoder is served as the image encoder of EfficientSAM for finetuning on SA-1B.

mismatch between the output of SAM image encoder and MAE.

**Reconstruction Loss.** At each training iteration, SAMI consists of a feedforward feature extraction from SAM image encoder, and a feedforward and a backpropagation procedure of MAE. The outputs from SAM image encoder and MAE linear projection head are compared to compute the reconstruction loss.

Let us denote the SAM image encoder as $f^{\text{sam}}$, and the encoder and decoder of MAE as $g^e$ with weights $W_e$ and $g^d$ with weights $W_d$, linear projection head as $h^\theta$ with weights $W_\theta$ respectively. Assume the input tokens are denoted as $\{\mathbf{x}_i\}_{i=1}^N$, where $N$ is the number of tokens. The input tokens are randomly grouped into the unmasked tokens, $\{\mathbf{x}_i\}_{i\in\mathcal{U}}$, the masked tokens $\{\mathbf{x}_i\}_{i\in\mathcal{M}}$ with a given masked ratio. Let the feature reordering operator be $\phi$, and the merging operator be $\oplus$.

Our target features from SAM image encoder can be written as $f^{\text{sam}}(\mathbf{x}) = f^{\text{sam}}(\{\mathbf{x}_i\}_{i=1}^N)$, the output from MAE encoder is $g^e(\{\mathbf{x}_i\}_{i\in\mathcal{U}})$, the decoder output is $g^d(\{\mathbf{x}_i\}_{i\in\mathcal{M}})$. The output from linear projection head is $f^h(\mathbf{x}) = h^\theta(\phi(g^e\{\mathbf{x}_i\}_{i\in\mathcal{U}} \oplus g^d\{\mathbf{x}_i\}_{i\in\mathcal{M}}))$. Therefore, our target reconstruction loss can be formulated as,

$$L_{W_e,W_d,W_\theta} = \frac{1}{N} \cdot \sum_{j=1}^{N} ||f^{\text{sam}}(\mathbf{x}) - f^h(\mathbf{x})||^2, \quad (1)$$

where $N$ is the number of input tokens, $||\cdot||$ denotes a norm. We use $\ell_2$ norm for reconstruction loss in our experiments. By minimizing the reconstruction loss, $L_{W_e,W_d,W_\theta}$, our encoder $g^e$ is optimized to serve as an image backbone to extract features as SAM image encoder. Our encoder, decoder, and linear projection head are optimized to learn context modeling ability from SAM image encoder. Optimizing the reconstruction loss on all tokens transfer the knowledge embedded in SAM.

**SAMI for EfficientSAM.** After pretraining, our encoder extract feature representations for various vision tasks and the decoder is discarded. In particular, to build efficient SAM models for the segmentation anything task, we take the SAMI-pretrained lightweight encoder such as ViT-Tiny and ViT-Small as the image encoder and the default mask decoder of SAM for our EfficientSAM, as illustraed in Fig. 2 (bottom). We finetune our EfficientSAM models on SA-1B dataset for the segment anything task. The overview of our EfficientSAM framework is illustrated in Fig. 2.

## 4. Experiments

### 4.1. Experimental Settings

**Pretraining Datasets.** Our masked image pretraining method, SAMI, is conducted on ImageNet-1K training set with 1.2M images. Following masked image pretraining [26], we do not use the label information. We use the SAM ViT-H image encoders from [31] to generate reconstruction features when pretraining our ViT models, ViT-Tiny, ViT-Small, and ViT-Base.

**Pretraining Implementation Details.** Our ViT models are pretrained with a mean squared error (MSE) loss for reconstruction. We use a batch size of 4096, AdamW optimizer

[40] with learning rate 2.4e-3, $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.05, linear learning rate warm-up over the first 40 epochs, cosine learning rate decay to update our models. We only adopt random resize crop to 224x224 resolution, random horiontal flip, and normalization for data augmentation. The mask ratio is set to $75\%$ and the decoder contains 8 Transformer blocks with 512 dimensions as in [26]. We pretrain SAMI for 400 epochs using PyTorch framework on V100 machines. For reference, 1600-epoch pretraining is required for MAE[26].

**Downstream Tasks/Datasets/Models.** *Tasks and Datasets.* We first consider three benchmarking datasets and several representative vision tasks to demonstrate the superiority of the proposed SAMI, including image classification on ImageNet dataset [16] with 1.2 million training and 50K validation images; Object detection and instance segmentation on COCO dataset [36] with 118K training and 5K validation images; Semantic segmentation on ADE20K dataset [72] with 20K/2K/3K images for training, validation, and testing, respectively. Then, we consider segment anything task to further show the advantages of our proposed SAMI. We finetune our pretrained lightweight image encoders for SAM on SA-1B dataset [31] with more than 1B masks from 11M high-resolution images, and test interactive instance segmentation and zero-shot instance segmentation ability of our EfficientSAMs on COCO and LVIS [23]. *Models.* We discard the decoder of SAMI while keeping the encoder as backbone to extract features for different tasks as in MAE[26]. We apply our well-pretrained ViT backbones for different tasks including ViTs for the classification, ViTDet [34] for the detection and instance segmentation, Mask2former [13] for the semantic segmentation task, and SAM for segment anything.

**Finetuning Settings.** *For the classification task,* we use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.05 to finetune ViTs for 100 epochs using 32 V100 GPUs, with each GPU having a batch size of 32. The initial learning rate is 1e-3 with first 5 epochs for linear warm-up and decays to zero by a cosine learning rate scheduler. We set the layer-wise learning rate decay factor to 0.75 for ViT-Small and ViT-Base. We do not apply layer-wise learning rate decay for ViT-Tiny. For data augmentation, we adopt RandAugment [15] and set label smoothing to 0.1, mixup to 0.8. *For the detection and instance segmentation task,* We follow the ViTDet [34] framework by adapting ViT backbones to a simple feature pyramid, for object detection and instance segmentation. We adopt AdamW optimizer with momentum $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay 0.1 to train models on COCO. All models are trained on 64 V100 GPUs for 100 epochs with each GPU having 1 batch size with image resolution $1024 \times 1024$. The initial learning rate is $2e-4$, linearly warmed up for the first 10 epochs, and decayed to 0 by a cosine learning rate schedule. Models are trained for 100 epochs. *For the segmentation task,* Our pretrained ViT models serve as the backbone of Mask2former [13], which is finetuned together with the segmentation layers on ADE20K. We adopt the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a mini-batch size of 16, a weight decay of 0.05, and an initial learning rate of 2e-4. The learning rate is decayed to 0 by a poly learning rate schedule. A learning rate multiplier is set to 0.1 for the backbone. The input image resolution is $512 \times 512$. Models are trained for 160K iterations using 8 V100 GPUs. *For the segmentation anything task,* Following [31], we take our pretrained lightweight ViT models, ViT-Tiny and ViT-Small, as the image encoder of SAM framework and finetune the encoder and decoder together of our EfficientSAM on SA-1B dataset for 5 epochs. We use the AdamW optimizer with a momentum, ($\beta_1 = 0.9$, $\beta_2 = 0.999$), a mini-batch size of 128, and a initial lrearning rate of $4e - 4$. The learning rate is decayed to 0 by a linear learning rate schedule. We set weight decay to 0.1. We do not apply data augmentation. The input image resolution is $1024 \times 1024$. Our EfficientSAMs are trained on 64 A100 GPUs with 40GB GPU memory.

**Baselines and Evaluation Metrics.** *Baselines.* For the classification task, we compare the performance of ViT backbones from different pretraining/distillation methods including MAE[26], SSTA[60], DMAE[2], BEiT[3], CAE[12], DINO[6], iBOT[73], DeiT[53], etc. For the detection and instance semantic task, and semantic segmentation task, we also compare with several pretrained ViT backbones for ViTDet[34] and Mask2former[13]. For the segment everything task, we compare with SAM[31], FastSAM[71], and MobileSAM[68]. *Evaluation Metrics.* We evaluate our method and all baselines in terms of accuracy. Specifically, the accuracy metrics refer to top-1 accuracy for the classification task; $\text{AP}^{\text{box}}$, $\text{AP}^{\text{mask}}$, for the detection and instance segmentation task (AP: average precision); mIoU, for the semantic segmentation task (mIoU: mean intersection over union); mIoU, AP, $\text{AP}^{\text{S}}$, $\text{AP}^{\text{M}}$, $\text{AP}^{\text{L}}$ for segment anything task. For efficiency metrics, we compare the number of model parameters or inference throughput.

## 4.2. Main Results

**Image Classification.** To evaluate the effectiveness of our proposed techniques on the image classification task, we apply the proposed SAMI idea to ViT models and compare their performance over baselines on ImageNet-1K. As shown in Tab. 1, our SAMI is compared with pretraining methods like MAE, iBOT, CAE, and BEiT, and distillation methods including DeiT and SSTA. SAMI-B achieves 84.8% top-1 accuracy, which outperforms the pretraining baselines, MAE, DMAE, iBOT, CAE, and BEiT by 1.2%, 0.8%, 1.1%, 0.9%, and 0.4% respectively. Compared with distillation methods such as DeiT and SSTA, SAMI also shows large improvements. For lightweight models such as ViT-Tiny

| Method | Backbone | Training Data | Acc.(%) |
|---|---|---|---|
| DeiT-Ti[53] | ViT-Tiny | IN1K | 74.5 |
| SSTA-Ti[60] | ViT-Tiny | IN1K | 75.2 |
| DMAE-Ti[2] | ViT-Tiny | IN1K | 70.0 |
| MAE-Ti[26] | ViT-Tiny | IN1K | 75.2 |
| SAMI-Ti (ours) | ViT-Tiny | SA1B (11M) + IN1K | **76.8** |
| DeiT-S[53] | ViT-Small | IN1K | 81.2 |
| SSTA-S[60] | ViT-Small | IN1K | 81.4 |
| DMAE-S[2] | ViT-Small | IN1K | 79.3 |
| MAE-S[26] | ViT-Small | IN1K | 81.5 |
| BEiT-S[3] | ViT-Small | D250M+IN22K+IN1K | 81.7 |
| CAE-S[12] | ViT-Small | D250M+IN1K | 82.0 |
| DINO-S[6] | ViT-Small | IN1K | 82.0 |
| iBOT-S[73] | ViT-Small | IN22K+1N1K | 82.3 |
| SAMI-S (ours) | ViT-Small | SA1B (11M) + IN1K | **82.7** |
| DeiT-B[53] | ViT-Base | IN1K | 83.8 |
| DMAE-B[2] | ViT-Base | IN1K | 84.0 |
| BootMAE[18] | ViT-Base | IN1K | 84.2 |
| MAE-B[26] | ViT-Base | IN1K | 83.6 |
| BEiT-B[3] | ViT-Base | D250M+IN22K+IN1K | 83.7 |
| CAE-B[12] | ViT-Base | D250M+IN1K | 83.9 |
| DINO-B[6] | ViT-Base | IN1K | 82.8 |
| iBOT-B[73] | ViT-Base | IN22K+1N1K | 84.4 |
| SAMI-B (ours) | ViT-Base | SA1B (11M) + IN1K | **84.8** |

Table 1. Image classification results on ImageNet-1K. IN is short for ImageNet. We use IN1K for pretraining/finetuning and add SA1B to the training data to indicate the need for original SAM.

| Method | Backbone | $\text{AP}^{\text{bbox}}$ | $\text{AP}^{\text{mask}}$ |
|---|---|---|---|
| MAE-Ti[26] | ViT-Tiny | 37.9 | 34.9 |
| SAMI-Ti(ours) | ViT-Tiny | **44.7** | **40.0** |
| MAE-S[26] | ViT-Small | 45.3 | 40.8 |
| DeiT-S[53] | ViT-Small | 47.2 | 41.9 |
| DINO-S[6] | ViT-Small | 49.1 | 43.3 |
| iBOT-S[73] | ViT-Small | 49.7 | 44.0 |
| SAMI-S (ours) | ViT-Small | **49.8** | **44.2** |
| MAE-B[26] | ViT-Base | 51.6 | 45.9 |
| SAMI-B (ours) | ViT-Base | **52.5** | **46.5** |

Table 2. Object detection and instance segmentation results on the MS COCO using ViTDet.

| Method | Backbone | mIOU |
|---|---|---|
| MAE-Ti[26] | ViT-Tiny | 39.0 |
| SAMI-Ti(ours) | ViT-Tiny | **42.7** |
| MAE-S[26] | ViT-Small | 44.1 |
| SAMI-S (ours) | ViT-Small | **48.8** |
| MAE-B[26] | ViT-Base | 49.3 |
| SAMI-B (ours) | ViT-Base | **51.8** |

Table 3. Semantic segmentation results on the ADE20K dataset using Mask2former. The input resolution is $512 \times 512$.

| Method | COCO | | | LVIS | | |
|---|---|---|---|---|---|---|
| | box | 1 click | 3 click | box | 1 click | 3 click |
| SAM[31] | 78.4 | 55.6 | 74.1 | 78.9 | 59.8 | 75.2 |
| MobileSAM[68] | 74.2 | 43.7 | 59.7 | 73.8 | 51.0 | 54.4 |
| SAM-MAE-Ti[31] | 74.7 | 43.3 | 65.8 | 73.8 | 50.6 | 65.3 |
| EfficientSAM-Ti (ours) | 75.7 | 45.5 | 67.2 | 74.3 | 52.7 | 66.8 |
| EfficientSAM-S (ours) | 76.9 | 50.0 | 69.8 | 75.4 | 56.2 | 68.7 |

Table 4. Zero-shot single point valid mask evaluation results on COCO and LVIS. Following SAM[31], we uniformly sample random points within ground truth mask for click, and compute the tightest bounding box corresponding ground truth mask for box. SAM-MAE-Ti denotes SAM with pretrained MAE-Ti image encoder.

and ViT-Small, SAMI reports a substantial gain compared to DeiT, SSTA, DMAE, and MAE.

**Object Detection and Instance Segmentation.** We also extend the SAMI-pretrained ViT backbones to the downstream object detection and instance segmentation task and compare it with previous pretraining baseline on COCO dataset to evaluate its efficacy. Specifically, we take the pretrained ViT backbones and adapt them to a simple feature pyramid in the Mask R-CNN framework[25] for constructing the detector, ViTDet[34]. Tab. 2 shows the overall comparison between our SAMI and other baselines. We can see that our SAMI consistently achieves better performance over other baselines. SAMI-B obtains 0.9 $\text{AP}^{\text{bbox}}$ and $0.6^{\text{mask}}$ gains compared with MAE-B. For light-weight backbones, SAMI-S and SAMI-Ti report substantial gains compared to MAE-Ti and MAE-S. Moreover, SAMI-S significantly outperforms DeiT-S by 2.6 $\text{AP}^{\text{bbox}}$ and 2.3 $\text{AP}^{\text{mask}}$. For other pretraining baselines, our SAMI stiil compares favorably to DINO and iBOT. This set of experiments validate the effectiveness of the proposed SAMI for providing pretrained detector backbones in the object detection and instance segmentation task.

**Semantic Segmentation.** We further extend the pretrained backbones to the semantic segmentation task to evaluate its effectiveness. Specifically, we use ViT models as the backbone in Mask2former [13] framework to benchmark on the ADE20K dataset. As shown in Tab. 3, Mask2former with SAMI-pretrained backbones achieve better mIoU, i.e., ↑2.5, ↑4.7, and ↑3.7 improvement over backbones with MAE pretraining [26] on ImageNet-1K. This set of experiments validate that our proposed techniques could be well generalized to various downstream tasks.

### 4.3. EfficientSAMs for Segment Anything Task

Segment Anything task is a process of promptable segmentation to produce segmentation masks based on any form of the prompt, including point set, rough boxes or mask, freeform text. We follow SAM [31] and focus on point-based and box-based prompt segmentation on COCO/LVIS. We now test the generalization abilities of our model on segment anything task including zero-shot single point valid mask evaluation and zero-shot instance segmentation. We take the SAMI-pretrained lightweight backbones as the image encoder of SAM for building efficient SAMs, EfficientSAMs. Then we finetune EfficientSAMs on SA-1B dataset and report the performance on zero-shot single point valid mask evaluation and zero-shot instance segmentation.

**Zero-Shot Single Point Valid Mask Evaluation.** Similar

| Method | COCO | | | | LVIS | | | |
|---|---|---|---|---|---|---|---|---|
| | AP | APS | APM | APL | AP | APS | APM | APL |
| ViTDet-H[34] | 51.0 | 32.0 | 54.3 | 68.9 | 46.6 | 35.0 | 58.0 | 66.3 |
| SAM[31] | 46.5 | 30.8 | 51.0 | 61.7 | 44.7 | 32.5 | 57.6 | 65.5 |
| MobileSAM[68] | 38.7 | 23.7 | 42.2 | 54.3 | 34.4 | 23.8 | 44.9 | 53.7 |
| FastSAM[71] | 37.9 | 23.9 | 43.4 | 50.0 | 34.5 | 24.6 | 46.2 | 50.8 |
| EfficientSAM-Ti (ours) | 42.3 | 26.7 | 46.2 | 57.4 | 39.9 | 28.9 | 51.0 | 59.9 |
| EfficientSAM-S (ours) | 44.4 | 28.4 | 48.3 | 60.1 | 42.3 | 30.8 | 54.0 | 62.3 |

Table 5. Zero-shot instance segmentation results on COCO/LVIS. ViTDet boxes are prompted to perform zero-shot segmentation.

to SAM[31], we evaluate segmenting an object from a single foreground point. For general interactive segmentation, we also consider object segmentation from a single box, and multiple points as introduced in [31]. To achieve this, we uniformly sample random points within ground truth mask for click, and compute the tightest bounding box corresponding to ground truth mask for box. Since our models are able to predict multiple masks, we only evaluate the most confident mask as SAM [31].

**Results.** In Tab. 4, EfficientSAMs are compared with SAM, MobileSAM and SAM-MAE-Ti. On COCO, our EfficientSAM-Ti outperforms MobileSAM by 1.9 mIoU on 1 click and 1.5 mIoU on 1 box with comparable complexity. Our EfficientSAM-Ti with SAMI-pretrained weights also performs better than MAE-pretrained wights on COCO/LVIS interactive segmentation. We notice that our EfficientSAM-S only underperforms SAM by 1.5 mIoU on COCO box and 3.5 mIoU on LVIS box with 20x fewer parameters. We find that our EfficientSAMs also show promising performance on multiple click compared with MobileSAM and SAM-MAE-Ti.

**Zero-Shot Instance Segmentation.** Following SAM [31], instance segmentation task is performed by taking the bounding box (bbox) generated by ViTDet[34] as the prompt. The mask with the highest Intersection over Union (IoU) with the bbox as the predicted mask.

**Results.** In Tab. 5, we report AP, $\text{AP}^{\text{S}}$, $\text{AP}^{\text{M}}$, $\text{AP}^{\text{L}}$ for zero-shot instance segmentation. We compare our EfficientSAM with MobileSAM and FastSAM. We can see that EfficientSAM-S obtains more than 6.5 AP on COCO and 7.8 AP on LVIS over FastSAM. For EffidientSAM-Ti, it still outperforms FastSAM by a large margin, 4.1 AP on COCO and 5.3 AP on LVIS, and MobileSAM by 3.6 AP on COCO and 5.5 AP on LVIS. Note that our EfficientSAMs are much light-weight than FastSAM, e.g, 9.8M parameters for efficientSAM-Ti vs 68M parameters for FastSAM. EfficientSAM-S also significantly reduces the gap between SAM with 0.6G parameters, only ∼2 AP reduction. These results demonstrate the extraordinary benefits of EfficientSAMs for zero-shot instance segmentation and validate the advantages of our SAMI pretraining method.

**Qualitative Evaluation.** We now provide the qualitative results for a complementary understanding of instance segmentation capabilities of EfficientSAMs. Some examples can be seen in Fig. 3, Fig. 4, and Fig. 5. Specifically, we report the predicted masks with two types of prompts, point and box as in MobileSAM [68] and also segment everything results. More qualitative results can be found in the supplement. These results demonstrate that our EfficientSAMs have competing capabilities when comparing to SAM. Note that our EfficientSAMs are much lightweight than SAM, and our models can effectively give decent segmentation results. This indicates that our models can be served as a complementary version of SAM for many practical tasks.

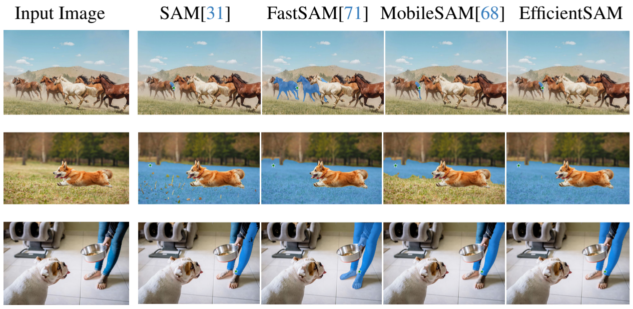

Figure 3. Visualization result on point-prompt input with SAM, FastSAM, MobileSAM, and our EfficientSAM model.

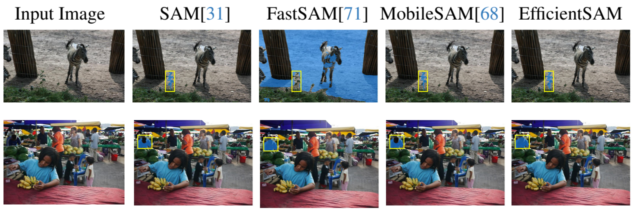

Figure 4. Visualization result on box-prompt input with SAM, FastSAM, MobileSAM, and our EfficientSAM model.

**Salient Instance Segmentation.** Salient object segmentation [4] aims to segment the most visually attractive objects from an image. We extend interactive instance segmentation to salient instance segmentation without manually creating points/boxes. Specifically, we take a state-of-the-art saliency object detection model, $\text{U}^2$-net[45], to predict saliency map and uniformly sample 3 random points (3 click) within

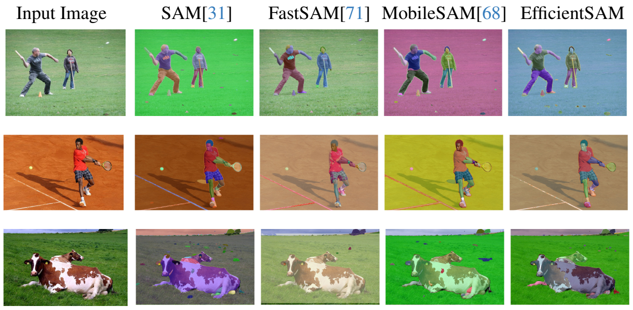

Figure 5. Visualization result on segment everything with SAM, FastSAM, MobileSAM, and our EfficientSAM model.

| Method | Loss | Top-1 Acc.(%) |
|---|---|---|
| SAMI-Ti | 1 - Cosine | 76.1 |
| SAMI-Ti | MSE | **76.8** |
| SAMI-S | 1 - Cosine | 82.3 |
| SAMI-S | MSE | **82.7** |

Table 6. Ablation study on training loss of SAMI. MSE loss gives better classification results on ImageNet-1K.

saliency map to perform instance segmentation with our EfficientSAM. In Fig. 6, we can see that our EfficientSAM can perform salient instance segmentation well. This preliminary exploration opens the potential to help people with hand impairments segment objects of interest in an image.

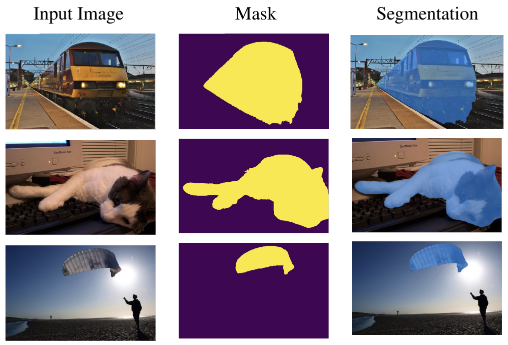

Figure 6. Saliency-based automatic instance segmentation results. With the assistance of saliency map generated from $U^2$-net[45], our EfficientSAM is able to generate mask and perform automatic instance segmentation without manually creating points or boxes.

### 4.4. Ablation Studies

We now analyze SAMI and EfficientSAMs through a series of ablation studies with ViT backbones.

**Reconstruction Loss.** We study the effect of reconstruction loss on the performance of SAMI on ImageNet-1K. We compare our mean square error (MSE) reconstruction loss with cosine similarity loss. We find that MSE reconstruction loss performs better, shown in Tab. 6. This recommends a direct reconstruction of SAM features instead of the target with a high angular similarity.

**Cross-Attention Decoder.** To reconstruct SAM features, we directly use the output tokens from encoder and only take decoder to transform the masked tokens with cross-attention. We study how the performance varies with cross-attention decoder. When changing the decoder module in MAE to our cross-attention decoder in SAMI, we find that SAMI-B improves the performance from 84.4% to 84.8% on ImageNet-1K. The cross-attention decoder of SAMI also shows consistent improvement for ViT-Tiny and ViT-Small. Analogy to anchor points in AnchorDETR[58], the output tokens from encoder are already learned well by directly aligning the SAM features, which can serve as anchor tokens for help masked tokens align via cross-attention decoder.

**Mask Ratio.** We explore how the performance varies with different mask ratio in SAMI. The observations are consistent with MAE [26] that a high mask ratio, 75%, tends to produce good results.

**Reconstruction Target.** We study the impact of reconstruction target. We take a different encoder from CLIP [46] to generate features as the reconstruction target in SAMI. Aligning features from CLIP encoder can also outperform MAE by 0.8% for a ViT-Tiny model on ImageNet-1K. This demonstrates that masked image pretraining benefits from powerful guided reconstruction.

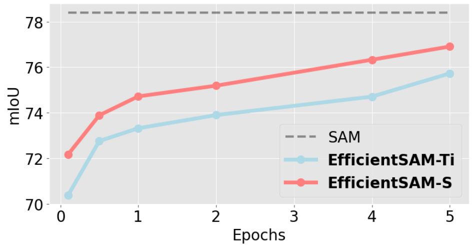

Figure 7. Ablation study on training steps for EfficientSAMs on MS COCO dataset. Zero-shot single point valid mask evaluation with a single box prompt is performed for the ablation.

**Effects of Finetuning Steps for EfficientSAMs.** We explore the effect of fintuning steps for EfficientSAMs. As illustrated in Fig. 7, EfficientSAM-Ti and EfficientSAM-S achieve decent performance even at 0.1 epoch. For 1 epoch, the performance gain is larger than 2.5 mIoU. The final performance of EfficientSAM-S reaches 76.9 mIoU, which is only 1.5 mIoU lower than SAM. These results demonstrate the advantages of SAMI-pretrained image encoders and our EfficientSAMs.

**Efficiency.** We compare the throughput (images per second) of our EfficientSAM with SAM and other models on A100 in Fig. 1. Our EfficientSAM reduces the runtime/parameters of SAM by ∼20x. We also observed similar speedup (∼20x) comparing to SAM w.r.t. FLOPs.

## 5. Conclusion

We proposed a masked image pretraining approach, SAMI, to explore the potential of ViTs under the guidance of SAM foundation model. SAMI improves masked image pretraining by reconstructing the latent features from SAM image encoder to transfer knowledge from vision foundation model to ViTs. Extensive experiments on image classification, object detection and instance segmentation, semantic segmentation, and the segment anything task consistently validate SAMI's advantages. We also demonstrate that SAMI helps build efficient SAMs with pretrained light-weight encoders. Our preliminary work suggests that SAMI has potential applications beyond efficient segment anything task.

## References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 1

[2] Yutong Bai, Zeyu Wang, Junfei Xiao, Chen Wei, Huiyu Wang, Alan L Yuille, Yuyin Zhou, and Cihang Xie. Masked autoencoders enable efficient knowledge distillers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24256–24265, 2023. 3, 5, 6

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3, 5, 6

[4] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational visual media*, 5:117–150, 2019. 7

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3, 5, 6

[7] Jun Cen, Yizheng Wu, Kewei Wang, Xingyi Li, Jingkang Yang, Yixuan Pei, Lingdong Kong, Ziwei Liu, and Qifeng Chen. Sad: Segment any rgbd. *arXiv preprint arXiv:2305.14207*, 2023. 2

[8] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. `https://github.com/fudan-zvg/Semantic-Segment-Anything`, 2023. 2

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[10] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023. 2

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 3

[12] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, pages 1–16, 2023. 5, 6

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 5, 6

[14] 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

[15] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[17] Ruining Deng, Can Cui, Quan Liu, Tianyuan Yao, Lucas W Remedios, Shunxing Bao, Bennett A Landman, Lee E Wheless, Lori A Coburn, Keith T Wilson, et al. Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging. *arXiv preprint arXiv:2304.04155*, 2023. 2

[18] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. In *European Conference on Computer Vision*, pages 247–264. Springer, 2022. 6

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021. 2

[21] Shanghua Gao, Zhijie Lin, Xingyu Xie, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Editanything: Empowering unparalleled flexibility in image editing and generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9414–9416, 2023. 2

[22] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021. 3

[23] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 5

[24] Dongsheng Han, Chaoning Zhang, Yu Qiao, Maryam Qamar, Yuna Jung, SeungKyu Lee, Sung-Ho Bae, and Choong Seon Hong. Segment anything model (sam) meets glass: Mirror and transparent objects cannot be easily detected. *arXiv preprint arXiv:2305.00278*, 2023. 1, 2

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 4, 5, 6, 8

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[28] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 3

[29] Jiaxi Jiang and Christian Holz. Restore anything pipeline: Segment anything meets image restoration. *arXiv preprint arXiv:2305.13093*, 2023. 2

[30] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Yolo by ultralytics. https://github.com/ultralytics/ultralytics, 2023. 2

[31] 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, 5, 6, 7

[32] Brett Koonce and Brett Koonce. Mobilenetv3. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pages 125–144, 2021. 2

[33] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 3

[34] 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. 2, 5, 6, 7

[35] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022. 3

[36] 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*, pages 740–755. Springer, 2014. 5

[37] Ruiping Liu, Jiaming Zhang, Kunyu Peng, Junwei Zheng, Ke Cao, Yufan Chen, Kailun Yang, and Rainer Stiefelhagen. Open scene understanding: Grounded situation recognition meets segment anything for helping people with visual impairments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1857–1867, 2023. 1, 2

[38] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14430, 2023. 3

[39] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[41] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 1, 2

[42] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2021. 2

[43] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 3

[44] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 2, 3

[45] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. page 107404, 2020. 7, 8

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 8

[47] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[48] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[49] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Anything-3d: Towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*, 2023. 2

[50] Ao Sun, Pingchuan Ma, Yuanyuan Yuan, and Shuai Wang. Explain any concept: Segment anything meets concept-based explanation. *arXiv preprint arXiv:2305.10289*, 2023. 1, 2

[51] Lv Tang, Haoke Xiao, and Bo Li. Can sam segment anything? when sam meets camouflaged object detection. *arXiv preprint arXiv:2304.04709*, 2023. 2

[52] Shehbaz Tariq, Brian Estadimas Arfeto, Chaoning Zhang, and Hyundong Shin. Segment anything meets semantic communication. *arXiv preprint arXiv:2306.02094*, 2023. 1, 2

[53] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2, 5, 6

[54] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *2011 international conference on computer vision*, pages 1879–1886. IEEE, 2011. 1

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[56] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 3

[57] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 3

[58] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2567–2575, 2022. 8

[59] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 3

[60] Haiyan Wu, Yuting Gao, Yinqi Zhang, Shaohui Lin, Yuan Xie, Xing Sun, and Ke Li. Self-supervised models are good teaching assistants for vision transformers. In *Proceedings of the 39th International Conference on Machine Learning*, pages 24031–24042. PMLR, 2022. 3, 5, 6

[61] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European Conference on Computer Vision*, pages 68–85. Springer, 2022. 3

[62] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 3

[63] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 3

[64] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2021. 3

[65] 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

[66] Haoran You, Yunyang Xiong, Xiaoliang Dai, Bichen Wu, Peizhao Zhang, Haoqi Fan, Peter Vajda, and Yingyan Celine Lin. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14431–14442, 2023. 3

[67] 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

[68] 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. 2, 5, 6, 7

[69] Xiao Feng Zhang, Tian Yi Song, and Jia Wei Yao. Deshadow-anything: When segment anything model meets zero-shot shadow removal. *arXiv preprint arXiv:2309.11715*, 2023. 2

[70] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022. 3

[71] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 2, 5, 7

[72] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5

[73] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 5, 6