Efficient Track Anything

Yunyang Xiong, Chong Zhou, Xiaoyu Xiang, Lemeng Wu, Chenchen Zhu, Zechun Liu,
Saksham Suri, Balakrishnan Varadarajan, Ramya Krishna Akula, Forrest Iandola,
Raghuraman Krishnamoorthi, Bilge Soran, Vikas Chandra
Meta AI Research
https://yformer.github.io/efficient-track-anything/

Abstract

Segment Anything Model 2 (SAM 2) has emerged as a powerful tool for video object segmentation and tracking anything. Key components of SAM 2 that drive the impressive video object segmentation performance include a large multistage image encoder for frame feature extraction and a memory mechanism that stores memory contexts from past frames to help current frame segmentation. The high computation complexity of image encoder and memory module has limited its applications in real-world tasks, e.g., video object segmentation on mobile devices. To address this limitation, we propose EfficientTAMs, lightweight end-to-end track anything models that produce high-quality results with low latency and small model size. Our idea is based on adopting lightweight Vision Transformer (ViT) as an image encoder for video object segmentation, and introducing an efficient memory module, which reduces the complexity for both frame feature extraction and memory computation for current frame segmentation. We take vanilla lightweight ViTs and efficient memory module to build EfficientTAMs, and train the models on SA-1B and SA-V datasets for video object segmentation and track anything tasks. We evaluate on multiple video segmentation benchmarks including semi-supervised VOS and promptable video segmentation, and find that our proposed EfficientTAM with lightweight ViT performs comparably to SAM 2 model (SAM 2-HieraB+) with ~1.6x speedup on A100 and ~2.4x parameter reduction. On segment anything image tasks, our EfficientTAMs also perform favorably over original SAM with ~20x speedup on A100 and ~20x parameter reduction. On mobile devices such as iPhone 15 Pro Max, our EfficientTAM can run at ~28 FPS for near real-time video object segmentation with reasonable quality, highlighting the capability of small models for on-device video object segmentation applications. Our EfficientTAM code and models are available at here.

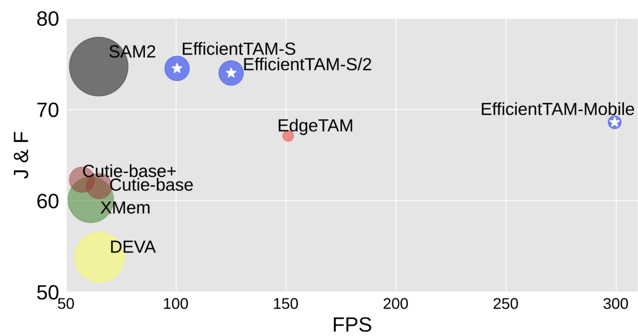

Figure 1. Comparative analysis. FPS/Parameter/Performance comparison of EfficientTAM and other models for zero-shot video object segmentation on SA-V test. We benchmark FPS of all models on a single NVIDIA A100. Our EfficientTAM performs comparably to SAM 2 model with ~1.6x speedup and ~2.4x parameter reduction, and outperforms other models by a large margin with comparable complexity. Our EfficientTAM-Mobile reduces the inference time of SAM 2 by ~4.6x and parameter size by ~4.5x. Comparing to EdgeTAM, our EfficientTAM-Mobile is more accurate while being ~2x efficient on A100 and achieving ~1.8x speedup on iPhone 15 Pro Max.

1. Introduction

Segment Anything Model 2 (SAM 2) [55] is a foundational model for unified image and video object segmentation, achieving state-of-the-art performance in various segmentation tasks such as zero-shot image segmentation [6, 7, 18, 35], semi-supervised video object segmentation [1, 9, 13, 40, 48, 52, 56, 66, 70, 75, 77, 78, 84], interactive video segmentation [3, 10, 13, 14, 17, 30, 31, 54, 76], and other real-world applications [20, 53, 58, 60, 71, 85, 87, 90]. SAM 2 uses a multistage image encoder to extract hierarchical frame features and introduces a memory module to cross-attend to both current frame features and stored memories from observed frames for consistent object segmentation across frames and interactive tracking in videos.

Despite these advantages, SAM 2 is not efficient for mobile deployment, particularly because the large image encoder (e.g., HieraB+) and memory module are expensive. The default image encoder of SAM 2, HieraB+ [57], is parameter inefficient, e.g., ∼80M parameters. While SAM 2 provides a tiny version, it has a running time of only ∼1 FPS. Additionally, the memory tokens (e.g., a concatenation of spatial memory tokens and object pointer tokens) are long, e.g., ∼30K, which hurts the efficiency of the memory module with cross-attention.

In this paper, we revisit using a vanilla lightweight ViT image encoder (e.g., ViT-Tiny/-Small [64]) as EfficientSAMs [73] did to reduce the complexity of SAM 2 while maintaining decent performance. Further, we propose an efficient cross-attention method for accelerating the memory module. This is achieved by leveraging the underlying structure of memory spatial tokens. We observed that the memory spatial tokens have strong locality and a coarser representation of memory spatial tokens can be a good proxy for performing cross-attention. We show that this yields a good alternative to the original memory module.

To evaluate our method, we conduct extensive experiments across video and image segmentation benchmarks, including MOSE, DAVIS, LVOS, and SA-V for video segmentation, and SA-23 for image segmentation. Our EfficientTAM outperforms strong semi-supervised video object segmentation methods such as EdgeTAM, Cutie-base, XMem, and DEVA while being more efficient. Compared with SAM 2, our EfficientTAM is comparable, e.g., 74.5% vs 74.7% on SA-V test dataset, with ∼ 1.6x speedup. On image segmentation benchmark, SA-23, our EfficientTAM achieves 60.7% accuracy for zero-shot image segmentation compared to 59.1% accuracy for SAM and and 61.9% for SAM 2. We also benchmarked our EfficientTAM model on iPhone 15 Pro Max, which can achieve near real-time video segmentation and outperform EdgeTAM with ∼1.8x speedup.

Our main contributions can be summarized as follows:

- We explore using lightweight vanilla ViT image encoder, ViT-Tiny/-Small, for video object segmentation and show that lightweight ViT encoder can achieve competing performance comparing to SAM 2.
- We propose an efficient memory cross-attention by exploiting the underlying memory spatial token structure and demonstrate the favorable performance.
- We deliver EfficientTAMs, lightweight video object segmentation and track anything models with state-of-the-art quality-efficiency tradeoffs (Fig. 1), which is complementary to SAM 2 for practical deployment.

2. Related Work

Video Object Segmentation (VOS) is a fundamental task in computer vision, segments objects of interest from the background and tracks target objects in a video. In the unsupervised setting [2, 24, 26, 28, 38, 39, 49–51, 63, 68, 74, 83], VOS models segment salient objects without a reference mask. In the semi-supervised setting [1, 9, 13, 40, 48, 52, 56, 66, 70, 75, 77, 78, 84], VOS requires tracking and segmenting objects based on a first-frame mask of target objects. For interactive video object segmentation (iVOS) [3, 10, 13, 14, 17, 30, 31, 54, 76], iVOS models perform object segmentation in videos (masklets) with user guidance, e.g., clicks, bounding boxes, and scribbles. In SAM 2 [55], semi-supervised VOS and iVOS have been extended to promptable visual segmentation (PVS), where the model can be interactively prompted with different types of inputs such as clicks, boxes, and masks on any frame in a video for segmenting and tracking a valid object.

Vision Transformers (ViTs) have achieved huge success on various vision tasks including image classification [21], object detection [41], image segmentation [8, 35], video classification [25], and video object segmentation [22, 76]. The original ViT family scales from the efficient ViT-Tiny up to ViT-Huge, with a plain, non-hierarchical architecture. There are also hierarchical vision transformers that combine transformers with convolutions [37], such as Swin [45], MViT [25, 42], PViT [69], and Hiera [57]. While being successful, hierarchical models are usually slower than their plain ViT counterparts for practical deployment [57].

Efficient Attention. The field has developed methods to reduce the quadratic cost of standard self-attention with respect to input sequence length [65]. Works in this direction include Linformer [67], Nyströmformer [72], and Performer [15]. The approach of [44, 79] leverages the associative property of matrix multiplication for efficient attentions in vision transformers. However, in preliminary experiments we found that these methods underperformed in a memory cross-attention module when adapted for efficiency improvement.

Segment Anything Model. SAM [35] is a vision foundation model that can segment any object in an image using interactive prompts such as points and bounding boxes. SAM has demonstrated remarkable zero-shot transfer performance and high versatility for many vision tasks including a broad range of segmentation applications [5–7, 18, 82, 88], inpainting [80], image restoration [34], image editing [27], image shadow removal [86], medical image segmentation [47], camouflaged object detection [61], transparent object detection [29], concept-based explanation [59], semantic communication [62], and object tracking [14, 76]. The strong ability on image segmentation with flexible prompts motivates the extension of SAM for video object segmentation and track anything. Track Anything Model (TAM) [76] combines SAM and XMem [9] for interactive video object tracking and segmentation with SAM for frame segmentation and XMem for tracking. SAM-Track [14] perform object tracking and segmentation in videos by combining SAM [35], DeAOT [77], and Grounding-Dino [43]. The latest SAM 2 [55] extended SAM for video segmentation through a hierarchical image encoder for frame embeddings and a memory module that conditions current frame embeddings on past frames. Motivated by mobile app use-cases and computationally-constrained applications, one concurrent work, EdgeTAM [89], leverages CNN encoder and Perceiver [33] to reduce the computational cost of SAM 2. Our work focuses on improving the efficiency of SAM 2 for practical deployment of video object segmentation and track anything.

3. Preliminaries

Segment Anything. SAM [35] contains a ViT image encoder and a prompt-guided mask decoder. The encoder takes an image and outputs image embeddings. Then the decoder takes the image embeddings and a prompt, which allows cutting out any object from the background in an image. SAM is trained on an image dataset of over 1B masks.

Segment Anything 2. The architecture of segment anything 2 (SAM 2) [55] largely follows SAM, which consists of a hierarchical image encoder, a prompt-guided lightweight mask decoder, and a new memory mechanism. SAM 2 uses a hierarchical image encoder, Hiera [57], to produce image embeddings for each frame. The stride 16 and 32 features from Stage 3 and 4 are used for the memory module. The stride 4 and 8 features from Stage 1 and Stage 2 are not used in the memory module but are fed to upsampling layers in the mask decoder for generating segmentation masks. For stable object tracking, SAM 2 employs a memory mechanism consisting of a lightweight memory encoder, a lightweight memory bank, and a memory attention module. It stores information from past frames and uses the memory attention module to perform cross-attention between the stored memory in the memory bank and current frame features, thereby understanding temporal dependencies in video.

The memory attention module consists of a stack of transformer blocks. Each block contains self-attention, cross-attention, and MLP. The first transformer block takes the image embedding from the current frame as an input. The core component of each transformer block, cross-attention, integrates the current frame embedding and the memory stored in memory bank to produce an embedding with temporal correspondence information. For memory tokens, it includes two parts, the spatial embedding tokens from memory encoder and the object-level pointer tokens from mask decoder. Let us assume the number of spatial tokens is n, the number of object-level pointer tokens is P, and d_m is the channel dimension, memory tokens can be formulated as $M_b = \begin{bmatrix} M_s \in \mathbb{R}^{n\times d_m} \\ M_p \in \mathbb{R}^{P\times d_m} \end{bmatrix}$.

Let L be the number of tokens and d_q be the dimension of each token for input frame features after self-attention, $X \in \mathbb{R}^{L\times d_q}$. The input sequence $X \in \mathbb{R}^{L\times d_q}$ is linearly projected to input queries $Q \in \mathbb{R}^{L\times d}$, and the memory tokens, $M_b \in \mathbb{R}^{(n+P)\times d_m}$ are linearly projected to keys $K \in \mathbb{R}^{(n+P)\times d}$, and values $V \in \mathbb{R}^{(n+P)\times d}$ respectively, where d is the embedding dimension of queries, keys, and values. The scaled dot-product cross attention mechanism applied on the queries Q, keys K, values V can be formally written as,

$$\mathsf{C}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

where the softmax operation is applied row-wise. A single head cross attention is used in the memory module. In later discussion, we also consider keys and values as memory tokens for simplification.

Efficiency Bottleneck. Despite the advantages of the hierarchical image encoder for multiscale frame feature extraction and cross-attention for integrating current frame features with stored memory, it poses the challenges for practical deployment of SAM 2. For example, the smallest SAM 2 model runs only ∼1 FPS on iPhone 15 Pro Max. Moreover, the number of tokens in keys and values for performing cross-attention in the memory module are super long, e.g., ∼30K. It leads to a large computation and memory cost when performing cross-attention, which becomes the efficiency bottleneck of the memory module for real-world deployment.

4. Efficient Video Object Segmentation and Track Anything

Motivated by the high quality image segmentation performance of EfficientSAM, we revisit using lightweight ViT image encoders such as ViT-Small/ViT-Tiny, for efficient frame feature extraction. Further, we introduce an efficient memory module to reduce the computation and memory cost by proposing an efficient cross-attention operation. Based on these two designs, we build efficient video object segmentation and track anything model by largely following SAM2. Fig. 2 illusrates an overview of our proposed EfficientTAM.

Efficient Image Encoder. The image encoder's role is to produce feature embeddings for each high-resolution frame. We use a SAMI [73] pretrained vanilla ViT image encoder [21, 64] to extract frame features. Differing from the image encoder of SAM 2, our image encoder provides a single-scale feature map and no other features in the mask decoder are added to the upsampling layers during decoding for segmentation mask generation. We adopt the lightweight image encoders, ViT-Small and ViT-Tiny, with a 16×16 patch size. Following [41], we use 14×14 non-overlapping windowed attention and 4 equally-spaced global attention blocks to efficiently extract features from high-resolution frames. Our image encoder outputs a single-scale feature embedding with a 16x reduced resolution, which takes high-resolution (e.g., 1024×1024) frames as input and transforms it into a dense embedding of downscaled size 64×64.

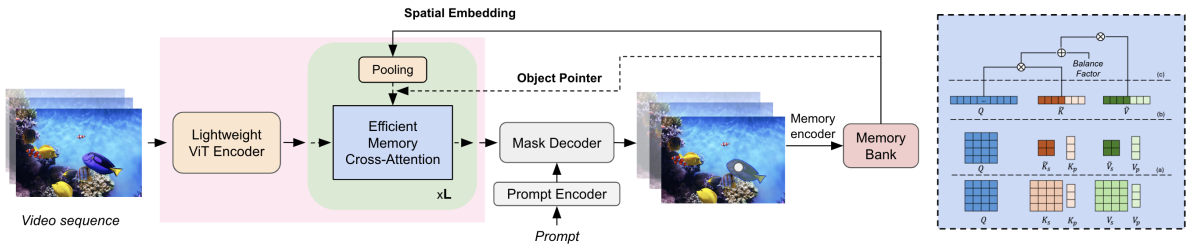

Figure 2. EfficientTAM architecture. Our proposed EfficientTAM takes a vanilla lightweight ViT image encoder for frame feature extraction. An efficient memory cross-attention is proposed to further improve the efficiency of EfficientTAM by leveraging the strong locality of memory spatial embeddings. Our efficient memory cross-attention contains 3 steps: (a) average pooling on spatial keys and values; (b) flatten and concatenate with object pointer; (c) perform cross-attention computation using Eq. (5) or Eq. (6), marked in the dotted box.

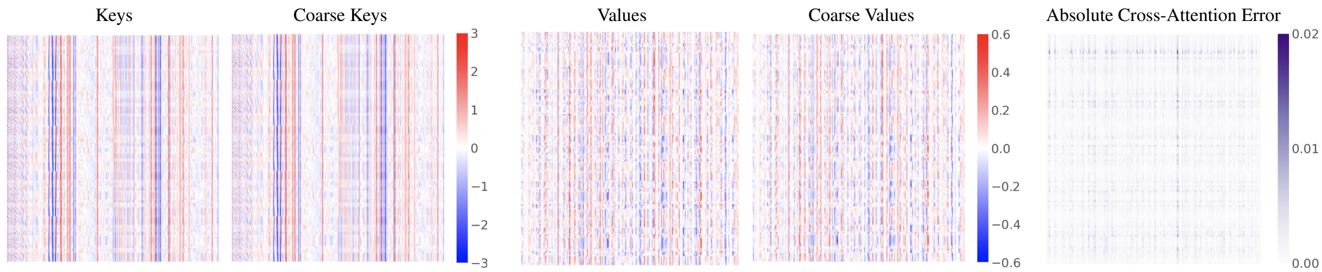

Figure 3. An example to show strong locality of the Keys and Values in the cross-attention of the memory module. Keys and Values are a matrix of size 28700×256. Cross-attention is a matrix of size 4096×256. For simplicity of visualizing and comparison, we only draw the top matrix of size 320×256. We use a single averaged token to represent other tokens in the homogeneous window with a 2×2 size, for Keys and Values to obtain coarse Keys and Values. At right, we visualize the difference between original cross-attention of Eq. (1) and efficient cross-attention of Eq. (5); the relative error w.r.t original cross-attention is 0.03 under Frobenius norm.

Efficient Memory Module. The memory module leverages information from previous frames to facilitate consistent object tracking. Cross-attention is a major efficiency bottleneck of the memory module in SAM 2 [55] due to its long memory token sequence. We now discuss how exploiting the underlying structure of memory tokens — local smoothness (strong locality) within spatial memory tokens — can yield a more efficient cross-attention.

Consider two consecutive memory spatial tokens, k_i and k_{i+1}, local smoothness implies that $||k_i - k_{i+1}||_2^2 \leq \frac{c_K}{n^2}$, for $i = 1, \dots, n-1$, where c_K is a positive constant. This suggests that given a sufficient small local window, $l_w \times l_h$, using a single token to represent other tokens in the homogeneous window may provide a coarser representation of the full set of memory spatial tokens K_s as $\tilde{K}_s$. We can construct a good surrogate of K_s with the same size, $\bar{K}_s$, from $\tilde{K}_s$ by repeating the single token in each window $l_w \times l_h$ times. Under the smoothness assumption, $\bar{K}_s$ will not be far from K_s. Empirically, we observed that a coarser representation of spatial memory tokens is good surrogate of the full spatial memory tokens. Fig. 3 confirms the coarser representation of input keys and values are close to the original keys and values of cross-attention in the memory module.

Utilizing highly correlated neighboring tokens in cross-attention, we perform average pooling to efficiently compute a coarser representation for keys K and values V in our model. For input spatial tokens $K_s = [k_{11}, \dots, k_{1h}; \dots; k_{w1}, \dots, k_{wh}]$ where $w \times h$ is the resolution size, we divide the $n = w \times h$ tokens into $k = \tilde{w} \times \tilde{h}$ rectangular pooling regions and compute the average token of each region. For simplicity, we assume w is divisible by $\tilde{w}$ and h is divisible by $\tilde{h}$. Denote $l_w = \frac{w}{\tilde{w}}, l_h = \frac{h}{\tilde{h}}$. $\tilde{K}_s$ and $\tilde{V}_s$ can be computed by averaging each region as,

$$\tilde{k}_{ij} = \sum_{p=i\times l_w+1}^{(i+1)\times l_w} \sum_{q=j\times l_h+1}^{(j+1)\times l_h} \frac{k_{pq}}{l_w \times l_h},$$
$$\tilde{v}_{ij} = \sum_{p=i\times l_w+1}^{(i+1)\times l_w} \sum_{q=j\times l_h+1}^{(j+1)\times l_h} \frac{v_{pq}}{l_w \times l_h}, \tag{2}$$

where $i = 1, \cdots, \tilde{w}, j = 1, \cdots, \tilde{h}$. This token-pooling scheme requires a single scan of the tokens leading to an efficient coarse token generation. We find that using averaging pooling with window size, 2×2, is sufficient to ensure a good approximation for spatial memory tokens.

Assume $\tilde{K}_s$ is a coarser representation of memory spatial keys, K_s, we can construct a good surrogate of $K_s \in \mathbb{R}^{n\times d}$ with the same size, $\bar{K}_s \in \mathbb{R}^{n\times d}$ from $\tilde{K}_s \in \mathbb{R}^{\tilde{w}\tilde{h}\times d}$ by stacking each $\tilde{k}_i, i = 1, \dots, \tilde{w}\tilde{h}$, $l_w \times l_h$ times, which can be written as,

$$\bar{K}_s = [\underbrace{\tilde{k}_1; \ldots; \tilde{k}_1}_{l_w \times l_h}; \underbrace{\tilde{k}_2; \ldots; \tilde{k}_2}_{l_w \times l_h}; \ldots; \underbrace{\tilde{k}_{\tilde{w}\tilde{h}}; \ldots; \tilde{k}_{\tilde{w}\tilde{h}}}_{l_w \times l_h}]$$

Similarly, we stack each $\tilde{v}_i, i = 1, \ldots, \tilde{w}\tilde{h}$, $l_w \times l_h$ times to construct $\bar{V}_s \in \mathbb{R}^{n \times d}$ as a good surrogate of values, $V_s \in \mathbb{R}^{n \times d}$, which can be written as,

$$\bar{V}_s = [\underbrace{\tilde{v}_1; \ldots; \tilde{v}_1}_{l_w \times l_h}; \underbrace{\tilde{v}_2; \ldots; \tilde{v}_2}_{l_w \times l_h}; \ldots; \underbrace{\tilde{v}_{\tilde{w}\tilde{h}}; \ldots; \tilde{v}_{\tilde{w}\tilde{h}}}_{l_w \times l_h}]$$

Then we concatenate this coarse spatial tokens with object pointer tokens, $\bar{K} = [\bar{K}_s; K_p] \in \mathbb{R}^{(n+P) \times d}$ and $\bar{V} = [\bar{V}_s; V_p] \in \mathbb{R}^{(n+P) \times d}$, for a good surrogate of original memory tokens, K and V. For queries $Q \in \mathbb{R}^{L \times d}$, and the coarse memory tokens, $\bar{K}$ and $\bar{V}$, we have,

$$\text{softmax}\left(\frac{Q\bar{K}^T}{\sqrt{d}}\right)\bar{V} = \text{softmax}\left(A\right)\tilde{V}, \tag{3}$$

where $A = [\frac{Q\tilde{K}_s^T}{\sqrt{d}} + \ln(l_w \times l_h), \frac{QK_p^T}{\sqrt{d}}] \in \mathbb{R}^{L \times (\tilde{w}\tilde{h}+P)}$, $\tilde{V} = [\tilde{V}_s; V_p] \in \mathbb{R}^{(\tilde{w}\tilde{h}+P) \times d}$. We provide a proof of Eq. (3) in the supplement. Since $\bar{K}$ and $\bar{V}$ are good surrogate of K and V respectively, we obtain a good surrogate of the original cross-attention, softmax $\left(\frac{QK^T}{\sqrt{d}}\right) V$ in Eq. (1),

$$\bar{\mathsf{C}}(Q, K, V) = \text{softmax}\left(\frac{Q\bar{K}^T}{\sqrt{d}}\right)\bar{V}, \tag{4}$$

With Eq. (3), we have an efficient version of cross-attention,

$$\bar{\mathsf{C}}(Q, K, V) = \text{softmax}(A)\tilde{V}, \tag{5}$$

There is a constant for balancing the attention score between coarse spatial tokens and object pointer tokens in Eq. (5), avoiding reducing the attention to spatial tokens after pooling. We find that adding it to keys for regularizing the attention between coarse spatial tokens and object pointer tokens can lead to efficient cross-attention with comparable performance.

$$\tilde{\mathsf{C}}(Q, K, V) = \text{softmax}\left(\frac{Q\tilde{K}^T}{\sqrt{d}}\right)\tilde{V}, \tag{6}$$

where $\tilde{K} = [\tilde{K}_s + \ln(l_w \times l_h), K_p] \in \mathbb{R}^{(\tilde{w}\tilde{h}+P) \times d}$. Fig. 3 illustrates our proposed efficient memory cross-attention.

It is feasible to achieve a good surrogate of the original cross-attention because spatial memory embeddings have strong locality. Our efficient cross-attention is close to the original cross-attention, visualized in Fig. 3.

5. Experiments

5.1. Experimental Setting

Pretraining. The SA-1B dataset consists of 11M diverse, high resolution images with 1.1B high-quality segmentation masks. Similar to [55], we pretrain our EfficientTAM without memory components on SA-1B dataset [35] for 90k steps. Our ViT image encoder is initialized from pre-trained ViTs [73] . We use the AdamW optimizer [46] with a momentum, ($\beta_1 = 0.9$, $\beta_2 = 0.999$), a global batch size of 256, and a initial learning rate of $4e-4$. The learning rate is decayed by a reciprocal square root learning rate schedule [81] with 1k iterations linear warmup and 5k iterations linear cooldown. We set weight decay to 0.1. We do not apply drop path [36] for our image encoder. Layer-wise decay [16] is set to 0.8. We apply horizontal flip augmentation and resize the input image resolution to 1024×1024. The maximum number of masks per image is 64. Our models are pre-trained on 256 A100 GPUs with 80GB GPU memory with a linear combination of focal and dice loss for mask prediction (e.g., a ratio of 20:1). Bfloat16 is used during the training.

Full Training Datasets. Following [55], we train our EfficientTAMs including memory components on SA-V dataset [55] and a 10% subset of SA-1B [35]. SA-V is a large-scale and diverse video segmentation dataset, including 51K videos captured across 47 countries and 600K mask annotations covering whole objects and parts. SA-V video resolution ranges from 240p to 4K and duration ranges from 4 seconds to 138 seconds. Unlike SAM 2, we do not use other open-source datasets or internal datasets during our training for a fair comparison with baselines.

Full Training Implementation Details. Similar to [55], we train our EfficientTAM for 300k steps after pretraining. We use the AdamW optimizer [46] with a momentum, ($\beta_1 = 0.9$, $\beta_2 = 0.999$), a batch size of 256, and a initial learning rate of $6e-5$ for image encoder and $3e-4$ for other components of the model. The learning rate is decayed by a cosine schedule with 15k iterations linear warmup. We set weight decay to 0.1. We do not apply drop path [36] for our image encoder. Layer-wise decay [16] is set to 0.8. We apply horizontal flip image augmentation and resize the input image resolution to 1024×1024. For video, we apply horizontal flip augmentation, affine transformation with degree 25 and shear 20, color jittering with brightness 0.1, contrast 0.03, saturation 0.03, gray scale augmentation with a probability of 0.05. The maximum number of masks is 64 per image and 3 per frame for video. Our models are trained on 256 A100-80G GPUs with a linear combination of focal and dice losses for mask prediction, mean-absolution-error loss for IoU prediction, and cross-entropy loss for object prediction. The ratio for the linear combination loss is 20:1:1:1. Bfloat16 is used for training.

Downstream Tasks/Datasets/Models. *Tasks and Datasets.* We consider zero-shot video tasks including promptable video segmentation and semi-supervised video object segmentation, and zero-shot image tasks to demonstrate the competing capabilities of EfficientTAM on image and video segmentation. For zero-shot image tasks, we evaluate EfficientTAM on 37 datasets including 23 datasets of SA-23 [35] and 14 video datasets introduced in [55]. For zero-shot video tasks, we evaluate our EfficientTAM on 9 densely annotated datasets for promptable video segmentation. We use 17 video datasets to evaluate zero-shot accuracy under interactive semi-supervised VOS setting using different prompts. For the standard semi-supervised VOS setting where a ground-truth mask on the first frame is provided, MOSE [19], DAVIS2017 [52], LVOS [32], SA-V [55], and YTVOS [75] are used to measure the VOS accuracy. These datasets cover different resolutions, frame rates, and challenging scenarios like motion blur and low-light conditions. We refer readers to [35, 55] for the details of these datasets. *Models.* We use our EfficientTAM for zero-shot image and video tasks.

Baselines and Evaluation Metrics. *Baselines.* For the standard semi-supervised VOS task, where the first-frame mask is provided, we compare the performance of our EfficientTAM with SAM 2 [55], EdgeTAM [89], Cutie-base [13], DEVA [12], XMem [9], etc. For the zero-shot promptable video segmentation task and the interactive semi-supervised video object segmentation task using different prompts, we compare our method with SAM2 [55], SAM+XMem++ [55], and SAM+Cutie [55]. For zero-shot image segmentation task, we compare with SAM [35] and SAM2 [55]. Note that we use the opensource version of SAM 2 (without training on MOSE/LVOS/YTVOS) for comparison. *Evaluation Metrics.* We evaluate our method and all baselines using the accuracy metrics of the combined $\mathcal{J}$(region similarity)&$\mathcal{F}$(contour accuracy), $\mathcal{G}$(averaged $\mathcal{J}$&$\mathcal{F}$ defined in [23]) for zero-shot video segmentation tasks; mIoU (mean intersection over union) for zero-shot image segmentation tasks. For efficiency metrics, we compare the number of model parameters or inference throughput on GPU (e.g, A100) and latency on mobile devices (e.g., iPhone 15 Pro Max). We follow SAM 2 [55] to report metrics. When providing main results on MOSE, LVOS, and YTVOS, we submit to their benchmarking servers to evaluate on *MOSE val*, *LVOS val*, and *YTVOS2019 val* for final performance. For ablation studies, we evaluate on a MOSE development set, *MOSE dev* with 200 randomly-sampled videos from the MOSE training split [55].

5.2. Main Results

Standard Semi-Supervised Video Object Segmentation. Semi-supervised video object segmentation is the process of object segmentation and tracking in a video based on a ground-truth mask on the first frame. We follow SAM 2 [55] and report accuracy of our methods on this standard semi-supervised video object segmentation task. We also report latency on a single A100 GPU with a batch size of 1. We evaluate EfficientTAMs with different image encoders, ViT-Tiny and ViT-Small, and memory modules, original memory block and efficient memory block with a 2×2 window pooling for a trade-off between efficiency and accuracy. EfficientTAM-S/-Ti denotes EfficientTAM using a ViT-Small/-Tiny image encoder and the original memory block, and EfficientTAM-S/2 denotes EfficientTAM with a ViT-Small image encoder and efficiency memory block with a 2×2 window pooling. EfficientTAM-Mobile denotes EfficientTAM-Ti/2 trained on resolution, 512×512, for near real-time on-device track anything. Tab. 1 compares our EfficientTAM with VOS baselines including SAM 2 [55], Cutie-base [13], and XMem [9]. On SA-V test, our EfficientTAM-S achieves 74.5 $\mathcal{J}$&$\mathcal{F}$, outperforming Cutie-base, Cutie-base+, and XMem by 12.2, 12.9, and 14.4, respectively. On long-term video object segmentation benchmark, LVOS, we can also see that Our EfficientTAM-S outperforms Cutie-base and XMem by a large margin. Notice that our EfficientTAM-S only underperforms SAM 2 by < 2 $\mathcal{J}$&$\mathcal{F}$ or $\mathcal{G}$ across 5 video benchmarks with $\sim$1.6x speedup and $\sim$2.4x fewer parameters. Further, EfficientTAM with efficient memory attention performs slightly worse than the one with original memory attention, but with much speedup, especially on mobile devices, $>$2x reduced latency on iPhone 15. For example, EfficientSAM-S achieves 74.5 $\mathcal{J}$&$\mathcal{F}$ on SA-V test with 1010.8 ms running time per frame on iPhone 15. EfficientSAM-S/2 with efficient cross-memory attention obtain 74.0 $\mathcal{J}$&$\mathcal{F}$ with only 450 ms. For on-device track anything, our EfficientTAM-Mobile runs at $\sim$28 FPS for near real-time video object segmentation, $\sim$1.8x speedup over EdgeTAM on iPhone 15 Pro Max while being more accurate than EdgeTAM across video benchmarks. For example, EfficientTAM-Mobile achieves 68.6 $\mathcal{J}$&$\mathcal{F}$ on SA-V test, 1.5 $\mathcal{J}$&$\mathcal{F}$ improvement over EdgeTAM. We also find that the energy consumption of EfficientTAM-Mobile is quite small. The battery of iPhone 15 Pro Max with $\sim$50 kJ (50×10^6 mJ) of energy, can perform efficient track anything for around 10^5 frames. These results show the extraordinary benefits of EfficientTAMs for semi-supervised video object segmentation and validate the advantages of our methods for practical deployment.

Promptable Video Segmentation. Similar to SAM 2 [55], we evaluate promptable video segmentation using two settings, offline evaluation and online evaluation. For offline evaluation, we make multiple passes through a video to annotate frames w.r.t. the largest model error. For online evaluation, we make a single pass through the video to annotate frames. 3 clicks per frame are used for the evaluations on 9 densely annotated video datasets including EndoVis, ESD, LVOSv2, LV-VIS, UVO, VOST, PUMaVOS, Virtual KITTI

Method	$\mathcal{J}\&\mathcal{F}$				$\mathcal{G}$	Parameters (M)	FPS	Latency (ms)
	MOSE val	DAVIS 2017 val	LVOS val	SA-V test	YTVOS 2019 val		A100	iPhone15
STCN [11]	52.5	85.4	-	57.3	82.7	54	62.8	-
RDE [40]	46.8	84.2	-	48.4	81.9	64	88.8	-
XMem [9]	59.6	86.0	-	60.1	85.6	62	61.2	-
DEVA [12]	66.0	87.0	55.9	53.8	85.4	69	65.2	-
Cutie-base [13]	69.9	87.9	66.0	61.6	87.0	35	65	-
Cutie-base+ [13]	71.7	88.1	-	62.3	87.5	35	57.2	-
SAM 2 [55]	72.8	88.9	76.2	74.7	87.9	81	64.8	1513.2
EfficientTAM-Ti/2 (ours)	68.4	88.4	66.1	70.8	87.1	18	156.2	261.4
EfficientTAM-Ti (ours)	69.3	89.1	69.6	70.7	86.7	18	117.7	840.5
EfficientTAM-S/2 (ours)	70.8	88.6	72.1	74.0	87.2	34	124.5	450
EfficientTAM-S (ours)	71.4	89.2	73.4	74.5	87.2	34	100.4	1010.8

Table 1. Standard semi-supervised video object segmentation results across video object segmentation benchmarks. Note that EfficientTAMs are trained on SA-1B and SA-V datasets.

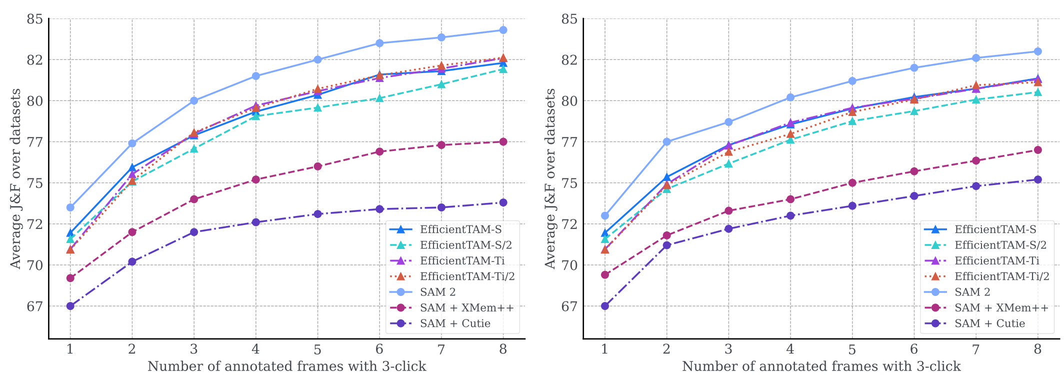

Figure 4. Promptable video segmentation results across 9 video segmentation datasets under interactive offline (left) and online (right) evaluation settings with 3-click. The average $\mathcal{J}\&\mathcal{F}$ over $1, \ldots, 8$ interacted frames is reported.

Method	1-click	3-click	5-click	bounding box	ground-truth mask
SAM+XMem++	56.9	68.4	70.6	67.6	72.7
SAM+Cutie	56.7	70.1	72.2	69.4	74.1
SAM 2	64.3	73.2	75.4	72.9	77.6
EfficientTAM-S/2	60.5	72.8	75.4	71.2	76.8
EfficientTAM-S	63	74.1	75.7	73.2	77.8

Table 2. Interactive semi-supervised video object segmentation results with different prompts. We report averaged $\mathcal{J}\&\mathcal{F}$ zero-shot accuracy across 17 video datasets for each type of prompt.

2, and VIPSeg. Average $\mathcal{J}\&\mathcal{F}$ accuracy over $1, \ldots, 8$ interacted frames is reported. Fig. 4 shows the comparison between our method and strong baselines including SAM 2, SAM + XMem++, and SAM + Cutie. EfficientTAM outperforms SAM + XMem++ and SAM + Cutie for both evaluation settings. EfficientTAM also reduces the gap between SAM 2 for offline and online settings. Specifically, with 8 annotated frames with 3-click, EfficientTAM-S and EfficientTAM-S/2 achieve ~ 82 $\mathcal{J}\&\mathcal{F}$ in average for offline evaluation setting and ~ 81 $\mathcal{J}\&\mathcal{F}$ in average for online evaluation, outperforming SAM + XMem++, and SAM + Cutie by >3 $\mathcal{J}\&\mathcal{F}$ and reducing the gap of SAM 2. This set of experiments further validate the effectiveness of our EfficientTAM on promptable video segmentation.

Interactive Semi-Supervised Video Object Segmentation. We also evaluate our method on the interactive semi-supervised video object segmentation task with click, box, or

Model	SA-23 All	SA-23 Image	SA-23 Video	14 new Video
SAM (ViT-B)	55.9 (80.9)	57.4 (81.3)	54.0 (80.4)	54.5 (82.6)
SAM (ViT-H)	58.1 (81.3)	60.8 (82.1)	54.5 (80.3)	59.1 (83.4)
HQ-SAM (ViT-B)	53.9 (72.1)	56.3 (73.9)	50.7 (69.9)	54.5 (75.0)
HQ-SAM (ViT-H)	59.1 (79.8)	61.8 (80.5)	55.7 (78.9)	58.9 (81.6)
SAM 2	61.9 (83.6)	63.2 (83.8)	60.3 (83.3)	69.9 (85.9)
EfficientTAM-Ti/2 (ours)	58.6 (82.5)	59.6 (82.8)	57.4 (82.1)	63.4 (84.9)
EfficientTAM-Ti (ours)	58.2 (82.6)	59.5 (82.9)	56.5 (82.1)	62.7 (85.0)
EfficientTAM-S/2 (ours)	60.5 (82.9)	61.6 (83.2)	59.1 (82.4)	67.8 (85.4)
EfficientTAM-S (ours)	60.7 (83.0)	61.7 (83.3)	59.5 (82.6)	67.7 (85.4)

Table 3. Segment anything results on SA-23 benchmark [35] and 14 new video benchmark [55]. The average 1-click (5-click) mIoU is reported.

mask prompts provided only on the first frame by following SAM 2. In Tab. 2, we report the average $\mathcal{J}\&\mathcal{F}$ accuracy over 17 video datasets for each type of prompt. We observe that EfficientTAM outperforms SAM + XMem++, and SAM + Cutie with different input prompts. We also notice the reduced gap between EfficientTAM and SAM 2. With 1 click, our EfficientTAM-S obtain 63 $\mathcal{J}\&\mathcal{F}$ accuracy, with a 6 $\mathcal{J}\&\mathcal{F}$ gain over SAM + XMem++ and SAM + Cutie and a slight loss, 1.3 $\mathcal{J}\&\mathcal{F}$ comparing to SAM 2. In summary, EfficientTAM performs favorably on the interactive semi-supervised VOS task using different prompts.

Segment Anything on Images. We now evaluate our model for the segment anything task on images. In Tab. 3, we report 1-click and 5-click mIoU accuracy on both SA-23 bench-

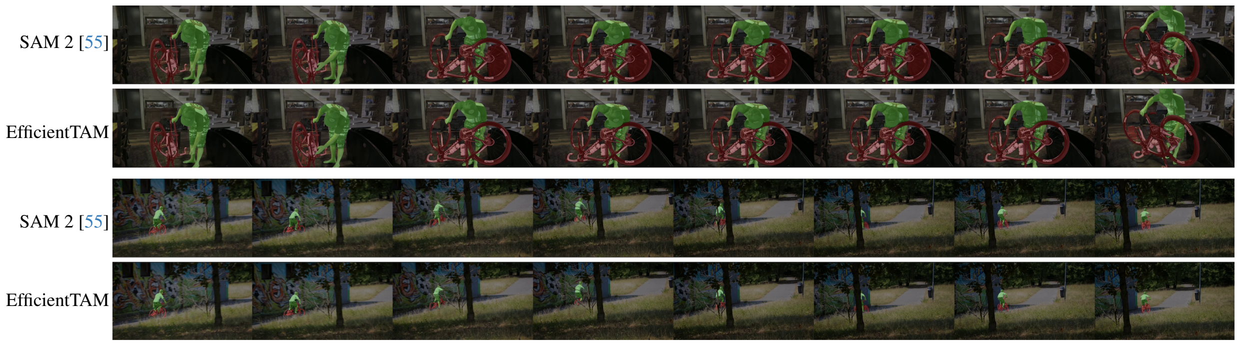

Figure 5. Visualization results on video segmentation and tracking with SAM 2, and our EfficientTAM model. We sampled a subset of frames for visualization. The segmented objects with occlusion, e.g., the person, are colored in green.

mark, plus the new benchmark introduced in SAM 2 [55] with 14 video datasets from video domain. We compare our EfficientTAMs with SAM (ViT-H) and HQ-SAM (ViT-H). Our EfficientTAM-S obtains a 2.6 mIoU improvement over SAM (ViT-H) and 1.6 mIoU improvement over HQ-SAM (ViT-H) on 1-click accuracy. For 5-click, we observe consistent improvement over SAM (ViT-H) and HQ-SAM (ViT-H). We also notice a significant improvement on the video benchmarks of SA-23 and the one with 14 new videos. This indicates our EfficientTAMs are strong for both image and video segmentation.

Qualitative Evaluation. Fig. 5 shows two video examples. We compare EfficientTAM and SAM 2 with a mask in the first frame prompted. We find that our EfficientTAM can generate high-quality masklet for the target object as SAM 2. More challenging examples with small objects and occlusions can be seen in the supplement. These results suggest that our EfficientTAMs have similar abilities to SAM 2, while EfficientTAM is more efficient.

5.3. Ablation Studies

Impact of the object pointer tokens. We study the effect of the object pointer tokens when performing cross-attention in the memory module. We ablate the cross-attention with or without the object pointer tokens. We find that object pointers significantly improve the performance on SA-V test dataset, 74.5 vs 72.1 $\mathcal{J}\&\mathcal{F}$, consistent with SAM 2 [55]. This demonstrates that object pointer tokens need to be cross-attended with spatial tokens from the memory bank.

Structure of memory tokens. We ablate the impact of memory tokens for efficient cross-attention in the memory module. In our efficient cross-attention, we leverage the locality of memory spatial tokens for a coarser representation, and we concatenate the coarser embedding with object pointer tokens. We observe that naively pooling the entire memory tokens instead of only the spatial tokens yields a large performance drop, 2.3 $\mathcal{J}\&\mathcal{F}$ on SA-V test.

Generalization of token locality. We perform zero-shot video segmentation evaluation on Corsican Fire, Virtual KITTI 2, and EndoVis 2018 benchmarks with highly dynamic or non-local object movements. We observe that leveraging token locality within a 2×2 window in efficient cross-attention yields a minor performance drop with $\sim 0.4\%$ $\mathcal{J}\&\mathcal{F}$. These results demonstrate the generalization ability to highly dynamic or non-local object movements scenario.

Impact of window size. We perform window size ablation for our efficient memory cross-attention. We experiment with window sizes 2×2 and 4×4. We find increasing the window from 2×2 to 4×4 for efficient cross-attention will lead to ~ 1 $\mathcal{J}\&\mathcal{F}$ accuracy drop with marginal speed improvement. We also note that a 4×4 window size will introduce a larger performance drops on benchmarks such as Virtual KITTI 2, which suggests that performing an averaging pooling over a large window size on videos with non-local object movements may lose much details. Therefore, our experimental results suggest that a window size of 2×2 achieves a trade-off between accuracy and efficiency.

Linear cross-attention. We explore adapting one representative efficient attention method such as linear attention [4, 79] by leveraging the associative property of matrix multiplication. We find that it leads to significant performance drop, > 10 $\mathcal{J}\&\mathcal{F}$ accuracy on SA-V test, comparing to our proposed efficient cross-attention. Therefore, leveraging the underlying token structure for efficient cross-attention is more effective.

Efficient cross-attention. We compare efficient cross-attention, Eq. (5) and Eq. (6). We observe that Eq. (5) and Eq. (6) achieve comparable performance across video segmentation benchmarks, e.g., ~ 74 $\mathcal{J}\&\mathcal{F}$ on SA-V test.

6. Conclusions

We revisited using vanilla lightweight ViT image encoders and proposed an efficient memory module by leveraging the locality of spatial memory embeddings, for building efficient video object segmentation and track anything models, EfficientTAMs. EfficientTAMs demonstrate competing image and video segmentation capabilities while being more efficient and deployable on mobile devices. Extensive experiments on semi-supervised video object segmentation, promptable video segmentation, and the segment anything tasks consistently validate the advantages of our EfficientTAM. Our preliminary work suggests that EfficientTAM has many potential applications for on-device tracking anything.

References

[1] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 777–794. Springer, 2020. 1, 2

[2] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010. 2

[3] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018. 1, 2

[4] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023. 8

[5] 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

[6] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. `https://github.com/fudan-zvg/Semantic-Segment-Anything`, 2023. 1

[7] 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. 1, 2

[8] 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. 2

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 1, 2, 6, 7

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021. 1, 2

[11] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 7

[12] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023. 6, 7

[13] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024. 1, 2, 6, 7

[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. 1, 2

[15] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 2

[16] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. 5

[17] Thanos Delatolas, Vicky Kalogeiton, and Dim P Papadopoulos. Learning the what and how of annotation in video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6951–6961, 2024. 1, 2

[18] 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. 1, 2

[19] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20224–20234, 2023. 6

[20] Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. Sam2long: Enhancing sam 2 for long video segmentation with a training-free memory tree. *arXiv preprint arXiv:2410.16268*, 2024. 1

[21] 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. In *International Conference on Learning Representations*, 2020. 2, 3

[22] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5912–5921, 2021. 2

[23] Xu et al. Youtube-vos: Sequence-to-sequence video object segmentation. *ECCV'18*. 6

[24] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, page 8, 2014. 2

[25] 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

[26] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1853. IEEE, 2012. 2

[27] 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

[28] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE, 2010. 2

[29] 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. 2

[30] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using global and local transfer modules. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 297–313. Springer, 2020. 1, 2

[31] Namdar Homayounfar, Justin Liang, Wei-Chiu Ma, and Raquel Urtasun. Videoclick: Video object segmentation with a single click. *arXiv preprint arXiv:2101.06545*, 2021. 1, 2

[32] Lingyi Hong, Zhongying Liu, Wenchao Chen, Chenzhi Tan, Yuang Feng, Xinyu Zhou, Pinxue Guo, Jinglun Li, Zhaoyu Chen, Shuyong Gao, et al. Lvos: A benchmark for large-scale long-term video object segmentation. *arXiv preprint arXiv:2404.19326*, 2024. 6

[33] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 3

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

[35] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2, 3, 5, 6, 7

[36] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *International Conference on Learning Representations*, 2017. 5

[37] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2

[38] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *2011 International conference on computer vision*, pages 1995–2002. IEEE, 2011. 2

[39] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE international conference on computer vision*, pages 2192–2199, 2013. 2

[40] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1332–1341, 2022. 1, 2, 7

[41] 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, 3

[42] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4804–4814, 2022. 2

[43] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2

[44] 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. 2

[45] 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

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

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

[48] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9226–9235, 2019. 1, 2

[49] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE international conference on computer vision*, pages 1777–1784, 2013. 2

[50] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE conference on computer vision and pattern recognition*, pages 733–740. IEEE, 2012.

[51] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video

object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 2

[52] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1, 2, 6

[53] Junlong Qiu, Wei Liu, Erzhu Li, Lianpeng Zhang, and Xing Li. Ded-sam: Adapting segment anything model 2 for dual encoder-decoder change detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 1

[54] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*, 2023. 1, 2

[55] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[56] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7406–7415, 2020. 1, 2

[57] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 2, 3

[58] Yiqing Shen, Hao Ding, Xinyuan Shao, and Mathias Unberath. Performance and non-adversarial robustness of the segment anything model 2 in surgical video segmentation. *arXiv preprint arXiv:2408.04098*, 2024. 1

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

[60] George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh: Zero-shot mesh part segmentation via lifting segment anything 2 to 3d. *arXiv preprint arXiv:2408.13679*, 2024. 1

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

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

[63] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4268–4276, 2015. 2

[64] 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, 3

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 2

[66] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Chuanxin Tang, Xiyang Dai, Yucheng Zhao, Yujia Xie, Lu Yuan, and Yu-Gang Jiang. Look before you match: Instance understanding matters in video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2268–2278, 2023. 1, 2

[67] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[68] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, 2015. 2

[69] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 2

[70] Qiangqiang Wu, Tianyu Yang, Wei Wu, and Antoni B Chan. Scalable video object segmentation with simplified framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13879–13889, 2023. 1, 2

[71] Xinyu Xiong, Zihuang Wu, Shuangyi Tan, Wenxue Li, Feilong Tang, Ying Chen, Siying Li, Jie Ma, and Guanbin Li. Sam2-unet: Segment anything 2 makes strong encoder for natural and medical image segmentation. *arXiv preprint arXiv:2408.08870*, 2024. 1

[72] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14138–14148, 2021. 2

[73] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16111–16121, 2024. 2, 3, 5

[74] Chenliang Xu and Jason J Corso. Evaluation of super-voxel methods for early video processing. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1202–1209. IEEE, 2012. 2

[75] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 585–601, 2018. 1, 2, 6

[76] 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. 1, 2

[77] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *Advances in Neural Information Processing Systems*, 35:36324–36336, 2022. 1, 2

[78] Zongxin Yang, Jiaxu Miao, Yunchao Wei, Wenguan Wang, Xiaohan Wang, and Yi Yang. Scalable video object segmentation with identification mechanism. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2

[79] 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. 2, 8

[80] 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

[81] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022. 5

[82] 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

[83] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 628–635, 2013. 2

[84] Jiaming Zhang, Yutao Cui, Gangshan Wu, and Limin Wang. Joint modeling of feature, correspondence, and a compressed memory for video object segmentation. *arXiv preprint arXiv:2308.13505*, 2023. 1, 2

[85] Mingya Zhang, Liang Wang, Limei Gu, Zhao Li, Yaohui Wang, Tingshen Ling, and Xianping Tao. Sam2-path: A better segment anything model for semantic segmentation in digital pathology. *arXiv preprint arXiv:2408.03651*, 2024. 1

[86] 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

[87] Yuxuan Zhang, Tianheng Cheng, Rui Hu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Evf-sam: Early vision-language fusion for text-prompted segment anything model. *arXiv preprint arXiv:2406.20076*, 2024. 1

[88] 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

[89] Chong Zhou, Chenchen Zhu, Yunyang Xiong, Saksham Suri, Fanyi Xiao, Lemeng Wu, Raghuraman Krishnamoorthi, Bo Dai, Chen Change Loy, Vikas Chandra, et al. Edgetam: On-device track anything model. *arXiv preprint arXiv:2501.07256*, 2025. 3, 6

[90] Yuli Zhou, Guolei Sun, Yawei Li, Luca Benini, and Ender Konukoglu. When sam2 meets video camouflaged object segmentation: A comprehensive evaluation and adaptation. *arXiv preprint arXiv:2409.18653*, 2024. 1