RMP-SAM: TOWARDS REAL-TIME MULTI-PURPOSE SEGMENT ANYTHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent segmentation methods, which adopt large-scale data training and transformer architecture, aim to create one foundation model that can perform multiple tasks. However, most of these methods rely on heavy encoder and decoder frameworks, hindering their performance in real-time scenarios. To explore real-time segmentation, recent advancements primarily focus on semantic segmentation within specific environments, such as autonomous driving. However, they often overlook the generalization ability of these models across diverse scenarios. Therefore, to fill this gap, this work explores a novel real-time segmentation setting called real-time multi-purpose segmentation. It contains three fundamental sub-tasks: interactive segmentation, panoptic segmentation, and video instance segmentation. Unlike previous methods, which use a specific design for each task, we aim to use only a single end-to-end model to accomplish all these tasks in real-time. To meet real-time requirements and balance multi-task learning, we present a novel dynamic convolution-based method, Real-Time Multi-Purpose SAM (RMP-SAM). It contains an efficient encoder and an efficient decoupled adapter to perform prompt-driven decoding. Moreover, we further explore different training strategies and one new adapter design to boost co-training performance further. We benchmark several strong baselines by extending existing works to support our multi-purpose segmentation. Extensive experiments demonstrate that RMP-SAM is effective and generalizes well on proposed benchmarks and other specific semantic tasks. Our implementation of RMP-SAM achieves the optimal balance between accuracy and speed for these tasks. Code and model will be available.

1 INTRODUCTION

Recent advancements in computer vision, driven by transformer architectures (Dosovitskiy et al., 2021; Carion et al., 2020; Wang et al., 2023b), focus on creating a single large-scale model versatile enough to handle different tasks like detection and segmentation with a general-purpose design. Several studies utilizing the same architectures (Wang et al., 2023a; Yan et al., 2023; Jain et al., 2023) have shown superior performance across various tasks. Enjoying the benefit of the large-scale model capacity and training data, the general-purpose models demonstrate strong performance on versatility for specific applications (Kirillov et al., 2023; Jia et al., 2021). For image and video segmentation at the semantic level, numerous studies (Cheng et al., 2022; Zhang et al., 2021; Li et al., 2022d; 2023e; Jain et al., 2023) employing a universal design have surpassed previous models tailored for specific image and video segmentation tasks. The Segment Anything Model (SAM) (Kirillov et al., 2023) is a leading approach in interactive segmentation, offering a unified model for various interactions such as point, bounding box, mask, and text input.

However, most of these studies face challenges in achieving real-time performance on devices with limited computing resources due to their *heavy encoders* or *cascaded decoders*. Despite their high-performance capabilities and versatility, this limitation hinders the practical use of these advanced architectures in vision applications, such as editing tools on edge devices. In contrast, several studies (Zhao et al., 2018; Hu et al., 2023) focus on real-time segmentation using specific designs (e.g., efficient decoder fusion) but are often tailored to a specific application trained on a particular dataset (e.g., COCO) or restricted to similar tasks like video semantic segmentation. To our knowledge, no study investigates real-time, multi-purpose segmentation, where a single model can perform universal segmentation tasks, including image, video, and SAM-like interactive segmentation.

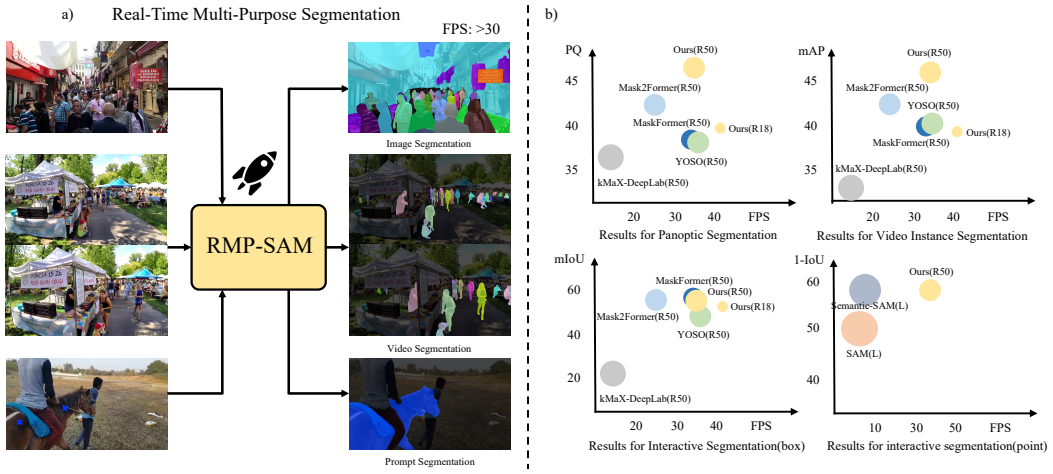

Figure 1: We present real-time multi-purpose segmentation to segment and recognize objects for image, video, and visual prompt inputs. In addition to benchmarking, we propose a simple yet effective baseline, named RMP-SAM, which achieves the best performance and speed trade-off among three different tasks. The larger dot indicates more parameters.

For example, given a video input, we want to create a tool to perform various segmentation tasks on the device, including interactive segmentation with visual cues, labeling the masks, and tracking the masks.

Inspired by the abovementioned challenges, we focus on real-time multi-purpose segmentation and efficiently investigate the novel issue of training versatile models. We pose a critical inquiry: Given the limited computational resources and model capacity, how can we develop an efficient, multi-purpose segmentation model? This entails creating a single model capable of segmenting, tracking, and classifying each pixel in real-time, similar to performing interactive segmentation similar to SAM. We present an illustration figure in Fig. 1(a).

First, we explore this problem by benchmarking existing methods. Specifically, we use the COCO (Lin et al., 2014) and YouTube-VIS (Yang et al., 2019) for joint co-training, employing identical hyperparameters. Beyond the object queries used for image and video segmentation in prior research, our approach incorporates visual prompt queries for interactive segmentation. However, none of these works can achieve the best accuracy and speed trade-off on three tasks using one model. This is because: 1) No previous works explore joint co-training for segmentation for different purposes. Thus, extensive experiments are needed to gain a better understanding of meta-architecture. 2) Interactive and semantic-level segmentation needs better strategies to balance each other. 3) Under the real-time setting, several previous architectures (Yan et al., 2023; Li et al., 2022c) for multi-task learning may not be practical.

To solve these issues, we present a real-time multi-purpose segment anything model (RMP-SAM). It contains an efficient encoder and shared decoder with multiple queries as inputs. Unlike previous works, we avoid cascaded transformers encoder layers and pixel-wise cross-attention decoder (Cheng et al., 2022). Instead, we propose a pooling-based dynamic convolution to replace per-pixel cross-attention to achieve better accuracy and speed trade-off. We represent all three tasks using queries with shared dynamic convolution operation. According to different outputs, we can divide the segmentation tasks into two types: one for class-agnostic masks (SAM-like) and the other for class-specific masks (contains one mask and one class label). Two types have different goals and will conflict during the multi-task learning process. Thus, we further explore four different architectures in the case of various decoder designs. In particular, we find a single decoder with multiple queries for different tasks can achieve good enough results and best speed and accuracy trade-off. Moreover, we present an effective dual adapter design that better adapts shared knowledge of the same decoder. The dual adapter contains an asymmetric design to better transfer shared features. To the best of our knowledge, RMP-SAM is the *first* real-time multi-purpose segmentation model that can achieve the best speed and accuracy trade-off on four different tasks, as shown in Fig. 1(b). Moreover, due to the

Table 1: Real-Time Segmentation Settings. Our multi-purpose segmentation supports more tasks in one framework.

Property	Image Seg	Video Seg	SAM-Like	SAM-2-Like	Ours: Real-Time Multi-Purpose Seg
Image Masks	✓	✗	✓	✓	✓
Video Masks	✗	✓	✗	✓	✓
Interactive	✗	✗	✓	✓	✓
Semantic Labels	✓	✓	✗	✗	✓
Multitasks	✗	✗	✗	✗	✓

Table 2: Comparison of Segmentation Methods. Our proposed RMP-SAM supports various segmentation tasks and performs them in real-time.

Methods	SS	PS	VIS	Interactive	Multi-Task in One Model	Real Time
ICNet (Zhao et al., 2018)	✓	✗	✗	✗	✗	✓
Bi-Seg (Yu et al., 2018)	✓	✗	✗	✗	✗	✓
YOSO (Hu et al., 2023)	✓	✓	✗	✗	✗	✓
Mobilie-VIS (Zhang et al., 2024)	✗	✗	✓	✗	✗	✓
SAM (Kirillov et al., 2023)	✗	✗	✗	✓	✗	✗
Mask2Former (Cheng et al., 2022)	✓	✓	✗	✗	✗	✗
Video K-Net (Li et al., 2022d)	✗	✓	✓	✗	✗	✗
OneFormer (Jain et al., 2023)	✓	✓	✗	✗	✓	✗
RMP-SAM (Ours)	✓	✓	✓	✓	✓	✓

semantic level and interactive level segmentation ability, our model can create more new applications, such as interactive video segmentation. In addition, we also verify our model in the SAM-like setting by co-training with SAM-data, it also achieves comparable results. Extensive experiments show the effectiveness of our design. The main contributions of this work are:

- We introduce real-time multi-purpose segmentation, a novel multi-task segmentation aiming to segment objects for real-time image, video, and interactive inputs.
- We benchmark several real-time transformer-based segmentation approaches for the new settings.
- We present a simple yet fast baseline named RMP-SAM. It contains a lightweight feature extractor, a unified decoder, and two asymmetric adapters.
- Extensive experiments demonstrate that RMP-SAM achieves the best speed and accuracy trade-off in the proposed benchmark and regular real-time semantic and panoptic segmentation benchmarks. We also show scalability across datasets and application demos.

2 Real-Time Multi-Purpose Segmentation

Problem Settings. Our proposed real-time multi-purpose segmentation contains three visual inputs: image, video, and visual prompts (boxes and points). It predicts the corresponding masks, labels, and instance IDs for the content within the given input. Basically, we adopt the definition of panoptic segmentation Kirillov et al. (2019) for image inputs since it covers all concepts in the scene. For video inputs, we segment and track each foreground instance (video instance segmentation). Given visual prompts, we follow the SAM-like Zhang et al. (2023a); Zhou et al. (2023); Shu et al. (2023); Xiong et al. (2023) setting for interactive segmentation, performing class-agnostic segmentation according to the user's inputs. Although recent advancements, such as SAM-2 (Ravi et al., 2024), have attempted to unify the interactive segmentation task for both image and video inputs, our framework emphasizes the importance of object semantics. It incorporates multi-task learning in a single shot. Combined with interactive segmentation and video segmentation, we can also achieve interactive video segmentation as video object segmentation. As shown in Tab. 1 and Tab. 2, we compare previous segmentation methods with and indicate that our method unifies different segmentation tasks within a single framework. This approach enables more flexible usage across various segmentation tasks, particularly regarding input modalities and predictions.

Datasets. For benchmarking, we use the well-known COCO dataset (Lin et al., 2014) for panoptic and interactive segmentation. During the training of interactive segmentation, the boxes derived from the ground truth, along with points randomly sampled from the masks, serve as visual prompts. For

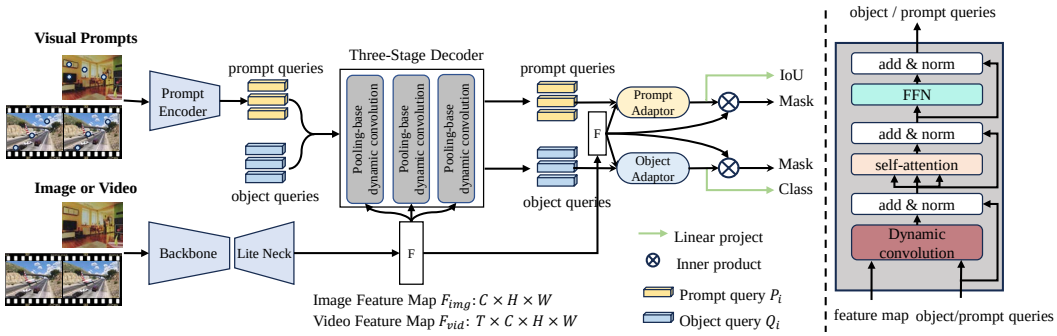

Figure 2: **RMP-SAM overview.** Our method contains three visual inputs: image, video, and visual prompts. Utilizing positional encoding, we generate prompt queries from these visual prompts. The learnable object queries, prompt queries, and the feature map F are directed to the multi-stage decoder. This process generates multi-stage predictions and refined queries. These refined queries engage in cross-attention with F, resulting in the final prediction.

testing, we obtain the center point of the mask as a visual prompt. For video segmentation, we adopt the widely used YouTube-VIS 2019 dataset (Yang et al., 2019) for training. We *do not* use the SAM data for our benchmarking. There are several reasons: (1) SAM data is not as well annotated as the COCO dataset. (2) We only focus on object-level and scene-level segmentation, while SAM data contain more granularity. (3) SAM data has no semantics, while our goal is to predict semantic labels for real applications. To fairly compare with SAM-like models, we adopt the SAM-data training for specific setting, as shown in Sec. 4.1.

3 Proposed Method

First, we revise panoptic segmentation, video instance segmentation, and interactive segmentation under the unified formulation. When combined with a query-based segmentation model, object queries can represent all entities within the scene, encompassing image-level objects, video-level tracked objects, and prompt-based specific objects. For panoptic segmentation (Kirillov et al., 2019), given an input image $I \in \mathbb{R}^{H \times W \times 3}$, the objective is to generate a group of masks $\{y_i\}_{i=1}^{N} = \{(m_i, c_i)\}_{i=1}^{N}$ where c_i denotes the class label of the binary mask m_i and N is the number of queries, $H \times W$ indicates the spatial dimensions. For video instance segmentation (Yang et al., 2019), given a video clip input $V \in \mathbb{R}^{T \times H \times W \times 3}$, where T denotes the number of frames, the process aims to generate an instance mask $\{y_i\}_{i=1}^{N} = \{(m_i, c_i, d_i)\}_{i=1}^{N}$, where N represents the number of the tube masks $m_i \in \{0, 1\}^{T \times H \times W}$, c_i indicates the class label of the tube m_i, and d_i identifies the instance ID for each instance. For SAM-like segmentation (Kirillov et al., 2023), both the image I and visual prompts $P \in \mathbb{R}^{K \times 2}$ (point prompts are used here) are taken as inputs. These prompts, which include points and boxes, lead to the outputs of corresponding binary image masks $\{y_i\}_{i=1}^{K} = \{m_i \in \{0, 1\}^{H \times W}\}_{i=1}^{K}$, where K denotes the number of visual prompts. Each visual prompt is encoded into one object query, which naturally can be the input of the decoder.

3.1 RMP-SAM Architecture

Overall Architecture. As shown in Fig. 2, our RMP-SAM is built upon a simple encoder and decoder architecture that can input images, videos, and visual prompts. Following SAM, we utilize the prompt encoder to convert visual prompts into queries. These visual prompts and object queries share the same decoder for the efficiency of computation and parameter size. However, the goals of the two types of queries are different: the former pays more attention to the local details of the visual prompts, while the latter considers the scene and temporal features. To better balance the results for interactive segmentation and image/video segmentation, we design a prompt adapter and an object adapter at the end of the decoder.

Lite Feature Extractor. Due to computation cost limitations, we avoid using previous methods (Cheng et al., 2022; Li et al., 2024b) that involve large backbones and heavier encoders. As required by real-time constraint, we investigate lightweight backbones such as ResNet18 (He et al.,

2016) and TopFormer (Zhang et al., 2022). We incorporate a feature pyramid network with deformable convolution to enhance representation alignment and fuse multi-scale features in real-time. When dealing with video input, we extract the spatial-temporal features. For simplicity, we use $F_{img} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times d}$ for image inputs and $F_{vid} \in \mathbb{R}^{T \times \frac{H}{4} \times \frac{W}{4} \times d}$ for video inputs.

Unified Dynamic Convolution Decoder. The goal of the decoder is to refine the object query. However, many of these methods (Cheng et al., 2021b; 2022), are not feasible for real-time settings due to their reliance on heavily cascaded layers and pixel-wise cross-attention mechanisms. In contrast, our method uses a pooling-based dynamic convolution framework (Zhang et al., 2021; Li et al., 2022d) to enhance the efficiency of the decoder. Given object query $Q \in \mathbb{R}^{N \times d}$ and prompt query $P \in \mathbb{R}^{K \times d}$, the masks M_i are obtained by dot product with F_{img} or F_{vid}. Here, i represents the decoder stage index, N is the total number of object queries, and K is the number of visual prompts. According to the inputs, the masks can be image masks for panoptic segmentation $M_i^{pan} \in \mathbb{R}^{N \times \frac{H}{4} \times \frac{W}{4}}$ and interactive segmentation $M_i^{iter} \in \mathbb{R}^{K \times \frac{H}{4} \times \frac{W}{4}}$ or tube masks $M_i^{tube} \in \mathbb{R}^{N \times T \times \frac{H}{4} \times \frac{W}{4}}$. Then, we can obtain the corresponding query features $X_i^{pan} \in \mathbb{R}^{N \times d}$, $X_i^{iter} \in \mathbb{R}^{K \times d}$, $X_i^{tube} \in \mathbb{R}^{N \times d}$ via mask pooling:

$$X_i^{pan} = \sum_{u}^{W} \sum_{v}^{H} M_{i-1}^{pan}(u, v) \cdot F_{img}(u, v). \tag{1}$$

For video tasks, we group the spatial-temporal features as follows:

$$X_i^{tube} = \sum_{t}^{T} \sum_{u}^{W} \sum_{v}^{H} M_{i-1}^{tube}(t, u, v) \cdot F_{vid}(t, u, v), \tag{2}$$

where u, v are the indices of spatial location, t is index of input frame. Since the dynamic convolution performs similarly for each task, we use panoptic segmentation X_i^{pan} as an illustration. For interactive segmentation, we adopt the same equation (Equ. 1). In our implementation, the learned query is shared for both Q_i^{pan} and Q_i^{tube}.

In particular, we adopt the design of previous methods (Zhang et al., 2021; Sun et al., 2021; Li et al., 2022c) to refine input queries Q_i^{pan} with features X_i^{pan} which are grouped from their masks,

$$\hat{Q}_i^{pan} = \mathrm{DynamicConv}(X_i^{pan}, Q_{i-1}^{pan}), \tag{3}$$

where the dynamic convolution uses the query features X_i^{pan} to generate parameters to weight input queries Q_{i-1}^{pan}. Specifically, $\mathrm{DynamicConv}$ uses input query features X_i^{pan} to generate gating parameters via a multilayer perceptron (MLP) and multiply back to the original query input Q_{i-1}^{pan}. We adopt the same design (Zhang et al., 2021; Sun et al., 2021) by learning gating functions to update the refined queries. The $\mathrm{DynamicConv}$ operation is based on:

$$\hat{Q}_i^{pan} = \mathrm{Gate}_x(X_i^{pan}) X_i^{pan} + \mathrm{Gate}_q(X_i^{pan}) Q_{i-1}^{pan}, \tag{4}$$

where Gate is implemented with a fully connected layer followed by Layer Norm and a sigmoid layer. We adopt two different gate functions, including Gate_x and Gate_q. The former is to weigh the query features, while the latter is to weigh corresponding queries. Next, we adopt one self-attention layer with feed-forward layers (Vaswani et al., 2017; Wang et al., 2021) to learn the correspondence among each query and update them accordingly. This operation leads to the full correlation among queries, shown as follows:

$$Q_i = \mathrm{FFN}(\mathrm{MHSA}(\hat{Q}_i) + \hat{Q}_i), \tag{5}$$

where MHSA means Multi Head Self Attention, FFN is the Feed Forward Network commonly used in the current version of Transformers (Carion et al., 2020; Dosovitskiy et al., 2021). We adopt several feed-forward layers on Q_i^{pan} for mask classification and directly output the class scores. Then, we perform the inner product between the learned queries Q_i^{pan}, and image features F_{vid} and F_{img}. The only difference for the video segmentation task is that we predict the tube mask rather than only a single image mask. The entire process (Equ. 1, Equ. 2, Equ. 3 and Equ. 5) is repeated three times in total.

Moreover, we also explore various decoupled designs, as shown in Fig. 3. In particular, as previous works suggest, we also explore decoupled decoders and their corresponding adapters. Via our detailed experiments, we do not find extra gains in the cases of decoupled decoders that introduce more computation and parameter costs. Please refer to Sec. 4.2 for the detailed decoder design.

Light-Weight Decoupled Adapter. After the shared decoder, we also add two lightweight adapters, A_{obj} and A_{prompt} to adapt the shared decoder's knowledge better. In particular, we use the asymmetric design. A_{obj} uses the same dynamic convolution design to refine the object query further, while A_{prompt} uses pixel-wise cross-attention design. Our key insights are: (1) The interaction between the whole scene feature and object queries is necessary for panoptic and video segmentation but not for interactive segmentation. The visual prompts provide strong positional information, and they pay more attention to local details features. (2) Despite the decoder being shared, the details are still missing for interactive segmentation due to the pooling effect in Equ. 1, where we even find inferior results after adopting a pooling-based adapter. More detailed adapter design can be found in Sec. 4.2.

3.2 Training, Inference, and Extension

Joint Image and Video Segmentation Co-training. Our goal is to train all segmentation tasks only once jointly. All training targets are one entity label and mask for all three cases. The entity can be thing, stuff, class-agnostic masks, and corresponding labels. Note that the instance masks with the same ID d form the tube masks. During training, we apply Hungarian matching between the predicted and ground-truth entity masks to assign object queries to video/image entities and then supervise their predicted masks and classification. To unify label taxonomies across datasets, we replace the learnable classifier with CLIP text embeddings following ViLD (Gu et al., 2021).

Following previous works (Cheng et al., 2022; Li et al., 2023e), the final loss function is given as $L = \lambda_{cls} L_{m_cls} + \lambda_{ce} L_{m_ce} + \lambda_{dice} L_{m_dice}$. Here, L_{m_cls} is the Cross-Entropy (CE) loss for mask classification, and L_{m_ce} and L_{m_dice} are mask Cross-Entropy (CE) loss and Dice loss (Milletari et al., 2016; Wang et al., 2020) for segmentation, respectively. By default, we set $\lambda_{cls} = 2, \lambda_{ce} = 5, \lambda_{dice} = 5$.

Inference. We follow the same inference procedure of Mask2Former (Cheng et al., 2022) for image segmentation. In panoptic segmentation, we merget the prediction of things and stuff according to the sorted scores. For video instance segmentation, we use query matching rather than introducing extra tracking components to generate instance ID, following previous work (Li et al., 2022d; Huang et al., 2022; Li et al., 2023e). We adopt a near-online method for inference. We follow the original SAM (Kirillov et al., 2023) for interactive segmentation tasks by providing box and point prompts and obtaining the binary masks. All the parameters are *shared* across three different tasks.

Extension and Application. Since our model can perform various segmentation tasks, Our method can be extended to more datasets and tasks, including semantic segmentation on ADE20k (Zhou et al., 2017) or video panoptic segmentation on VIPSeg (Miao et al., 2022). Moreover, our method can support prompt-driven video object segmentation by combining interactive segmentation with a video instance segmentation framework. We put the details of the extension and application for RMP-SAM in the appendix (Sec. 7.1).

4 Experiments

Datasets. Our benchmark uses COCO (Lin et al., 2014) and YouTube-VIS 2019 (Yang et al., 2019) datasets for benchmarking. COCO-SAM is constructed using ground truth boxes and points randomly sampled from ground truth masks as visual prompt inputs. In addition, to verify the effectiveness and generality of RMP-SAM, we also use other datasets, including ADE-20K (Zhou et al., 2017) and VIP-Seg dataset (Miao et al., 2022) in Sec. 4.1.

Evaluation Metrics and Devices. For image segmentation, we adopt panoptic quality (PQ), which can be further decomposed into the segmentation quality (SQ) and the recognition quality (RQ). For the VIS task, we use mAP as our primary metric. We further report the semantic segmentation results (mIoU) and video panoptic segmentation results (STQ (Weber et al., 2021) and VPQ (Kim et al., 2020)). Speed testing is conducted fairly on one A100 GPU for all models.

Re-implemented Baselines. Since no previous works explore joint co-training for image, video, and interactive segmentation in one model. We re-implement several representative baselines using the same codebase, including non-real-time models (Mask2Former (Cheng et al., 2022), kMaX-DeepLab (Yu et al., 2022)) for reference. We benchmark multiple different methods, including different backbones and decoders.

Table 3: Real-Time Multi-Purpose Segmentation benchmark. Our proposed RMP-SAM achieves the best accuracy and speed trade-off on three tasks.

Method	Backbone	COCO-Panoptic				COCO-SAM	YouTube-VIS 2019	FLOPs	Parameters	FPS
		PQ	SQ	PQ_{th}	PQ_{st}	mIoU	mAP			
Mask2Former	R18	35.6	77.5	39.0	30.3	54.7	38.6	89.8G	18.6M	31.2
MaskFormer	R18	31.0	76.0	33.2	27.7	**55.6**	34.1	79.7G	18.5M	38.0
kMaX-DeepLab	R18	27.8	71.0	30.3	24.1	16.9	23.1	87.1G	18.7M	15.0
YOSO	R18	31.6	76.8	36.4	24.4	45.7	31.4	57.3G	18.7M	41.0
RMP-SAM(Ours)	R18	**39.9**	**78.6**	**43.3**	**34.8**	52.7	**38.7**	60.5G	22.8M	40.3
Mask2Former	R50	42.9	79.8	47.6	35.6	58.0	42.1	153G	45.2M	26.6
MaskFormer	R50	37.4	77.3	41.2	31.8	**58.8**	40.0	143.0G	45.2M	34.3
kMaX-DeepLab	R50	36.9	75.7	42.7	28.4	20.1	26.4	280.0G	51.4M	14.9
YOSO	R50	37.3	76.9	42.7	29.2	49.2	40.7	119.0G	45.1M	36.3
RMP-SAM(Ours)	R50	**46.9**	**80.8**	**51.6**	**39.8**	57.9	**46.2**	123.0G	47.2M	35.1
MaskFormer	TopFormer	31.6	75.9	33.5	28.6	**56.1**	38.2	60.0G	12.8M	29.9
YOSO	TopFormer	31.0	75.9	33.0	26.9	45.1	27.9	36.8G	12.9M	31.1
RMP-SAM(Ours)	TopFormer	**34.6**	**77.0**	**37.8**	**29.8**	53.3	**41.7**	40.1G	15.0M	30.7

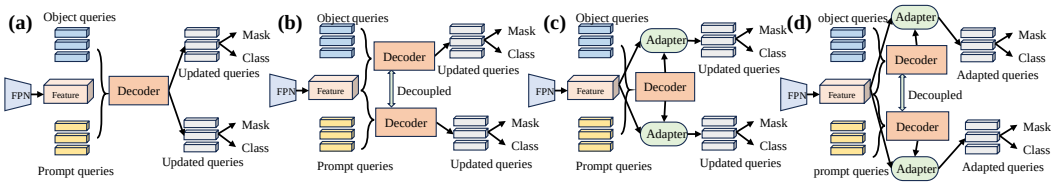

Figure 3: Meta-architecture exploration. (a), Simple shared decoder design. (b), Decoupled decoder design with two heads. (c), Shared decoder with decoupled adapter. (d), Decoupled decoder with the decoupled adapters. Best viewed in color and zoom in.

Implementation Details. We implement our models and all other baselines in PyTorch (Paszke et al., 2019). We use the distributed training framework with 16 A100 GPUs. Each mini-batch has two images per GPU, and each batch contains one data type. In particular, we adopt pseudo video training on COCO by moving image masks with random directions. All the models are trained with 12 epochs. For data augmentation, we adopt large-scale jitter as previous works (Cheng et al., 2022) to build strong baselines. For all models, we adopt the same training steps and optimizers. Refer to the appendix for more details.

4.1 Main Results

Our Benchmark Results. We list our benchmark results in Tab. 3. We benchmark recent state-of-the-art methods using the same training and test settings. From the table, our proposed RMP-SAM achieves the best speed and accuracy trade-off on all three visual segmentation tasks under the various backbones. Mask2Former (Cheng et al., 2022) achieves similar or partially stronger results than our method. However, their speed is still limited, and they are 7-8 FPS slower than our methods. YOSO (Hu et al., 2023) runs as fast as our method. However, the performance gap is still significant. Thus, our proposed RMP-SAM is a simple yet effective baseline for real-time and Multi-Purpose settings.

Compare with SAM-like Methods. Tab. 4 provides a detailed comparison between our models and previous SAM-like models. We report the results of mask AP of the COCO validation set, the parameter count, and FLOPs of each model. We first generate bounding boxes for each image using Mask R-CNN with ResNet50. Then, we send these bounding boxes to SAM-like models as box prompts. Our method is co-trained using SAM data and demonstrates even better results than most previous efficient SAM-like methods. It has strength in computational efficiency, with an additional function for video instance segmentation and panoptic segmentation.

Comparison with Specific Design Models on VIP-Seg. In Tab. 5(a), we verify the effectiveness of RMP-SAM on a more challenging video segmentation task, video panoptic segmentation (VPS) on the VIP-Seg dataset. Compared with recent works (Li et al., 2023e; Miao et al., 2022), RMP-SAM also archives the best speed and accuracy trade-off, despite RMP-SAM is not specifically designed for VPS.

Table 4: Comparison with SAM-like methods on COCO instance segmentation dataset. We first generate bounding boxes using Faster R-CNN, which are used as visual prompts for segmentation. We report the results of mask AP of the COCO validation set, the parameters, and the FLOPs.

Method	mAP	#Param.	FLOPs(G)	PS	VIS	Interactive
SAM-H (Kirillov et al., 2023)	35.6	641M	3000	✗	✗	✗
TinySAM (Shu et al., 2023)	32.5	10M	42	✗	✗	✗
MobileSAM (Zhang et al., 2023a)	32.1	10M	42	✗	✗	✗
EdgeSAM (Zhou et al., 2023)	31.9	10M	22	✗	✗	✗
EfficientSAM (Xiong et al., 2023)	32.4	10M	-	✗	✗	✗
FastSAM (Zhao et al., 2023)	34.3	68M	344	✗	✗	✗
RMP-SAM	33.2	47M	35	✓	✓	✓

Table 5: Compared with specifically designed segmentation models.

(a) Comparison with video segmentation method on VIP-Seg validation set.

Method	backbone	VPQ	STQ	FPS
Clip-PanoFCN	R50	22.9	31.5	8
Video K-Net	R50	26.1	33.1	10
Tube-Link	STDCv1	30.6	32.0	14
Tube-Link	STDCv2	31.4	32.8	12
Ours	R18	32.5	33.7	30

(b) Comparison with panoptic segmentation method on ADE-20k validation set.

Method	backbone	PQ	Parameters	FLOPs
PanSegFormer	R50	36.4	51M	214G
MaskFormer	R50	34.7	45.2M	181G
Mask2Former	R50	39.7	45.2M	226G
YOSO	R50	38.0	45.1M	176G
RMP-SAM	R50	38.3	47.2M	179G

Comparison with Specific Design Models on ADE-20k. In Tab. 5(b), we further transfer RMP-SAM on ADE-20k datasets. Since our method is a real-time model, we only list several representative works for reference. Again, our method still achieves stronger results than recent work (Hu et al., 2023).

4.2 Ablation Study and Visual Analysis

Ablation on Meta-Architecture Design. In Tab. 6(a), we further explore the meta-architecture design, as shown in Fig. 3. We use R50 as the backbone and keep the adapter unchanged. The table shows that a shared decoder architecture achieves the best parameter and performance trade-off. These findings may conflict with recent multi-task work (Li et al., 2022c) since the decoupled decoders may have more capacity. We argue that since our feature extractors are weaker than those works and the representation power is far from the heavier backbone such as Swin (Liu et al., 2021) or ViT-large (Dosovitskiy et al., 2021), adding more capacity in the decoder may not boost the final performance.

Effectiveness of Joint Co-training. We further explore the impact of each dataset in Tab. 6(b). We find joint co-training with image and video data leads to better performance for video instance segmentation but reduces the performance of panoptic segmentation. Adding COCO-SAM training leads to a few performance drops. This is because the learning targets are different for object queries and prompt queries.

Ablation on Shared Decoder Design. Tab. 6(c) shows using simple pooling-based dynamic convolution performs well under real-time settings, where we adopt ResNet18 as the backbone and keep the adapter unchanged. In particular, there is no difference in panoptic segmentation results and a slight gap for interactive segmentation. However, per-pixel cross-attention introduces extra computation costs.

Ablation on Adapter Design. Then, we explore the adapter design in Tab. 6(d). In particular, we use a pre-trained model without adapters for initialization to see the effect of different adapters. In particular, we find that using asymmetric adapters works well for balanced results for object queries and prompt queries since the goals of the two queries are different. The former needs the spatio-temporal and scene-level context, while the latter only cares about the region context with the input position as the guidance. Note that our adapter can also work with other architecture, as shown in the appendix.

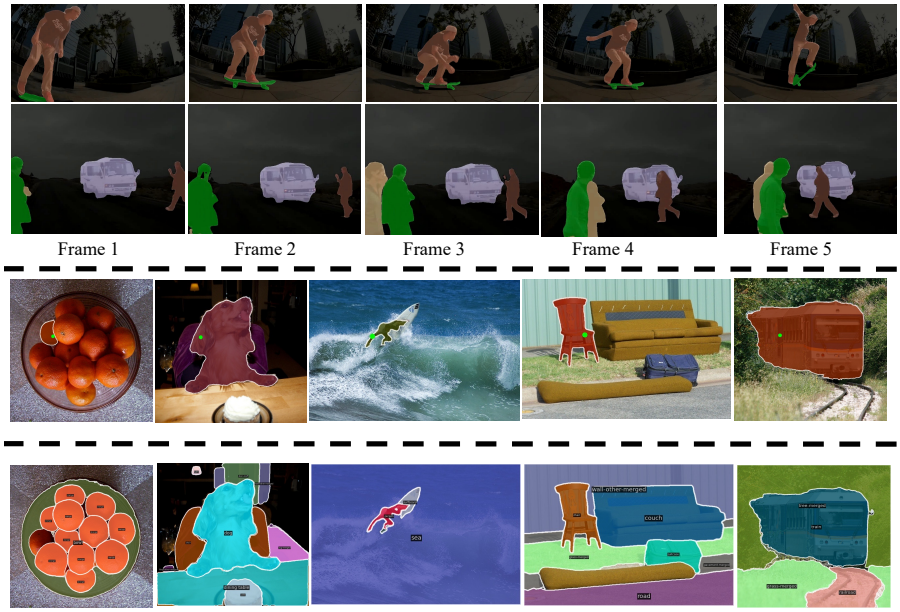

Figure 4: The visualization results of YouTube-VIS 2019 and COCO datasets using RMP-SAM. The first two rows visualize five frames of inputs. The same instances are in the same color. The third row shows the interactive segmentation results with a single-point prompt (green color). The last row shows the panoptic segmentation results.

Table 6: Ablation study on RMP-SAM's component design.

(a) Ablation on meta-architecture.

Setting	COCO-Panoptic	COCO-SAM	Parameters
Fig. 3 (a)	43.8	54.2	46.3M
Fig. 3 (b)	44.0	55.2	53.6M
Fig. 3 (c)	44.6	56.7	47.3M
Fig. 3 (d)	45.2	56.2	54.6M

(b) Ablation on joint co-training. a: COCO-Panoptic. b : YT-VIS 2019. c: COCO-SAM.

Setting	COCO-Panoptic	YT-VIS 2019	COCO-SAM
a	36.6	-	-
b	-	21.5	-
a + b	36.2	36.0	50.7
a + b + c	35.7	35.3	50.9

(c) Ablation on shared decoder design. DC: dynamic convolution. DCG: dynamic convolution with gating.

Settings	COCO-Panoptic	COCO-SAM
Per-Pixel Cross-Attention	45.0	55.3
Pooling + DC	43.8	55.7
Pooling + DCG	44.6	56.7

(d) Ablation on adapter design. CA: cross-attention. DC: dynamic convolution.

A_{obj}	A_{prompt}	COCO-Panoptic	COCO-SAM
-	-	44.2	53.2
CA	CA	42.6	54.3
DC	DC	44.7	52.1
DC	CA	44.6	56.7
CA	DC	44.5	56.2

Result Visualization. In Fig. 4, we give the three different visualization results on the YouTube-VIS 2019 and COCO datasets. We generate these visualization results using only a single model thanks to the joint co-training strategy. The first three rows in Fig. 4 show the video instance segmentation. The fourth row shows the interactive segmentation results with a single-point prompt (shown in green color), and the last row shows the results of panoptic segmentation.

5 Related Work

Universal Segmentation. Prior research has focused on designing segmentation models tailored for specific tasks. Recent developments in this field (Dosovitskiy et al., 2021; Liu et al., 2021; Carion et al., 2020; Zhang et al., 2021; Cheng et al., 2022; Yu et al., 2022; Li et al., 2021b; 2023c; Gu et al., 2023; Yuan et al., 2022; Li et al., 2022c; 2023d; 2024b; Yuan et al., 2024; Yang et al., 2021b; 2024) have shifted towards employing mask classification architectures, coupled with an end-to-

end set prediction objective, to facilitate universal segmentation. This approach has outperformed specialized models (Chen et al., 2019; Kirillov et al., 2019; He et al., 2017; Li et al., 2021a; Huang et al., 2019; Yuan et al., 2020; Zhao et al., 2017; Chen et al., 2018; Li et al., 2022b; 2020a; Chen et al., 2017) across various segmentation tasks, including those involving images, videos, and point clouds (Kim et al., 2022; Li et al., 2022d; 2023e; Cheng et al., 2021a; Xu et al., 2022). Specifically, Mask2Former (Cheng et al., 2022) uses a masked-attention mechanism and surpasses the performance of prior specific image segmentation models. Similarly, Tube-Link (Li et al., 2023e) outperforms earlier models (Li et al., 2022d;b; Liu et al., 2020; Shin et al., 2023; Goel et al., 2021) tailored for specific video segmentation tasks. Semantic-SAM (Li et al., 2023a), built on Mask-DINO (Li et al., 2023b), employs a hybrid query design to facilitate panoptic and interactive segmentation within a single framework. However, despite the strong performance of these works, their complexity, stemming from the heavier encoder or the cascaded decoder designs, poses significant challenges for integration into actual products. Slow processing speeds and substantial parameter overhead primarily hinder these.

Efficient Model Design. This research direction primarily concentrates on the development of efficient CNNs (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019; Iandola et al., 2016; Zhang et al., 2018; Ma et al., 2018; Han et al., 2020), transformers (Maaz et al., 2022; Pan et al., 2022; Mehta & Rastegari, 2022), and hybrid architectures (Zhang et al., 2023b; Ma et al., 2022; Chen et al., 2022b; Li et al., 2022a; Zhou et al., 2022), aimed at advancing visual representation learning. Most research investigates backbone design within the constraints of real-time processing and parameter efficiency. These studies are orthogonal to our work, which primarily explores the impact of data, task, and decoder design. Additionally, our work employs efficient models as encoders and presents detailed benchmarks and analyses of the encoder's effects.

Efficient Segmentation. Previous studies (Zhao et al., 2018; Li et al., 2020b; Yu et al., 2018; Hu et al., 2023; Hong et al., 2021; Wan et al., 2023; Yu et al., 2021; Mehta et al., 2018) on efficient segmentation have predominantly concentrated on closed-set and specific domains. In particular, a significant portion of this research (Li et al., 2020b; Hu et al., 2023; Mehta et al., 2019) is dedicated to driving scenarios. Recently, various studies (Zhang et al., 2022; Wan et al., 2023; Zhou et al., 2023) have developed efficient segmentation techniques that facilitate model execution on mobile devices. Mobile SAM (Zhang et al., 2023a) introduces a streamlined encoder distillation method. Fast SAM (Zhao et al., 2023) employs a single-stage instance segmentation framework that directly decodes class-agnostic masks. Recently, multiple studies have been conducted on efficient panoptic segmentation (Sun et al., 2023; Hu et al., 2023; Mohan & Valada, 2021) and rapid video instance segmentation (Yang et al., 2021a; Zhang et al., 2024). However, these real-time segmentation methods are limited to specific tasks. We contend that a multi-purpose model capable of real-time performance would have wide-ranging applications in editing, tracking, and segmentation functionalities within various products.

Interactive Segmentation. Interactive segmentation (Kirillov et al., 2023; Chen et al., 2022a; Sofiiuk et al., 2022; Lin et al., 2020; 2022) distinguishes objects by actively incorporating user inputs. Recently, SAM (Kirillov et al., 2023) has shown remarkable segmentation outcomes by leveraging multiple user interaction inputs and training on SA-1B data. Inspired by SAM, several studies (Zhang et al., 2023a; Zhou et al., 2023) have concentrated on employing a lighter image encoder to adapt SAM for real-time scenarios.

6 Conclusion

In this work, we explore a new challenging setting, real-time multi-purpose segmentation, to explore multi-task segmentation in one efficient model. To achieve this, we first benchmark existing image segmentation approaches by extending their query for video and interactive segmentation. However, no work can achieve the best speed and accuracy trade-off on all three sub-tasks. To solve this problem, we introduce RMP-SAM, a real-time segmentation model. It has a simple shared decoder with two key designs: shared dynamic convolution and asymmetric adapters. These designs lead to the best trade-off between speed and accuracy on three segmentation tasks. The proposed RMP-SAM model is versatile and effective for various tasks, including image segmentation, video segmentation, and iterative segmentation. To our knowledge, no real-time model can achieve this goal. The proposed benchmark and RMP-SAM baseline can facilitate future research in this new setting.

REFERENCES

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 5, 9

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 10

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 10

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 10

Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *CVPR*, 2022a. 10

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, 2022b. 10

Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv:2112.10764*, 2021a. 10

Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021b. 5

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 9, 10, 17, 18

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 *ICLR*, 2021. 1, 5, 8, 9

Golnaz Ghiasi, Yin Cui, A. Srinivas, Rui Qian, Tsung-Yi Lin, Ekin Dogus Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2020. 17

Vidit Goel, Jiachen Li, Shubhika Garg, Harsh Maheshwari, and Humphrey Shi. Msn: Efficient online mask selection network for video instance segmentation. In *CVPR*, 2021. 10

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021. 6

Xiuye Gu, Yin Cui, Jonathan Huang, Abdullah Rashwan, Xuan Yang, Xingyi Zhou, Golnaz Ghiasi, Weicheng Kuo, Huizhong Chen, Liang-Chieh Chen, et al. Dataseg: Taming a universal multi-dataset multi-task segmentation model. *arXiv:2306.01736*, 2023. 9

Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *CVPR*, 2020. 10

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 10

Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, and Wei Chu. Lpsnet: A lightweight solution for fast panoptic segmentation. In *CVPR*, 2021. 10

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 10

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017. 10

Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You only segment once: Towards real-time panoptic segmentation. In *CVPR*, 2023. 1, 3, 7, 8, 10

De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. In *NeurIPS*, 2022. 6, 17

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 10

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv:1602.07360*, 2016. 10

Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *CVPR*, 2023. 1, 3

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1

Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 6

Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *CVPR*, 2022. 10

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 3, 4, 10

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 *ICCV*, 2023. 1, 3, 4, 6, 8, 10, 18

Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. In *ECCV*, 2023a. 10, 18

Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, 2023b. 10

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:2207.05501*, 2022a. 10

Minghan Li, Shuai Li, Xindong Zhang, and Lei Zhang. Univs: Unified and universal video segmentation with prompts as queries. In *CVPR*, 2024a. 18

Xiangtai Li, Zhao Houlong, Han Lei, Tong Yunhai, and Yang Kuiyuan. Gff: Gated fully fusion for semantic segmentation. In *AAAI*, 2020a. 10

Xiangtai Li, Ansheng You, Zeping Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *ECCV*, 2020b. 10

Xiangtai Li, Hao He, Yibo Yang, Henghui Ding, Kuiyuan Yang, Guangliang Cheng, Yunhai Tong, and Dacheng Tao. Improving video instance segmentation via temporal pyramid routing. *T-PAMI*, 2022b. 10

Xiangtai Li, Shilin Xu, Yibo Yang, Guangliang Cheng, Yunhai Tong, and Dacheng Tao. Panoptic-partformer: Learning a unified model for panoptic part segmentation. In *ECCV*, 2022c. 2, 5, 8, 9

Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *CVPR*, 2022d. 1, 3, 5, 6, 10, 17

Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Guangliang Cheng, Pang Jiangmiao, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *TPAMI*, 2023c. 9

Xiangtai Li, Shilin Xu, Yibo Yang, Haobo Yuan, Guangliang Cheng, Yunhai Tong, Zhouchen Lin, Ming-Hsuan Yang, and Dacheng Tao. Panopticpartformer++: A unified and decoupled view for panoptic part segmentation. *TPAMI*, 2023d. 9

Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. Tube-link: A flexible cross tube baseline for universal video segmentation. In *ICCV*, 2023e. 1, 6, 7, 10, 18

Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *CVPR*, 2024b. 4, 9

Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *CVPR*, 2021a. 10

Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Tong Lu, and Ping Luo. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. *arXiv:2109.03814*, 2021b. 9

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 *ECCV*, 2014. 2, 3, 6

Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *CVPR*, 2020. 10

Zheng Lin, Zhao Zhang, Ling-Hao Han, and Shao-Ping Lu. Multi-mode interactive image segmentation. In *ACM MM*, 2022. 10

Haotian Liu, Rafael A. Rivera Soto, Fanyi Xiao, and Yong Jae Lee. Yolactedge: Real-time instance segmentation on the edge (jetson agx xavier: 30 fps, rtx 2080 ti: 170 fps). *arXiv:2012.12259*, 2020. 10

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 *ICCV*, 2021. 8, 9

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 17

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 17

Hailong Ma, Xin Xia, Xing Wang, Xuefeng Xiao, Jiashi Li, and Min Zheng. Mocovit: Mobile convolutional vision transformer. *arXiv:2205.12635*, 2022. 10

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 10

Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *ECCVW*, 2022. 10, 17

Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 10, 17

Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*, 2018. 10

Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *CVPR*, 2019. 10

Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *CVPR*, 2022. 6, 7

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 6

Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *IJCV*, 2021. 10

Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *ECCV*, 2022. 10

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: an imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 7

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv:2408.00714*, 2024. 3, 17

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 10

Inkyu Shin, Dahun Kim, Qihang Yu, Jun Xie, Hong-Seok Kim, Bradley Green, In-So Kweon, Kuk-Jin Yoon, and Liang-Chieh Chen. Video-kmax: A simple unified approach for online and near-online video panoptic segmentation. *WACV*, 2023. 10

Han Shu, Wenshuo Li, Yehui Tang, Yiman Zhang, Yihao Chen, Houqiang Li, Yunhe Wang, and Xinghao Chen. Tinysam: Pushing the envelope for efficient segment anything model. *arXiv:2312.13789*, 2023. 3, 8

Konstantin Sofiiuk, Ilya A. Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *ICIP*, 2022. 10

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. SparseR-CNN: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 5

Shuyang Sun, Weijun Wang, Qihang Yu, Andrew Howard, Philip Torr, and Liang-Chieh Chen. Remax: Relaxing for better training on efficient panoptic segmentation. *arXiv:2306.17319*, 2023. 10

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. In *ICLR*, 2023. 10

Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 5

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 6

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023a. 1

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. In *ICCV*, 2023b. 1

M. Weber, J. Xie, M. Collins, Yukun Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, Aljosa Osep, L. Leal-Taixé, and Liang-Chieh Chen. Step: Segmenting and tracking every pixel. In *NeurIPS*, 2021. 6

Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest N. Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficientsam: Leveraged masked image pretraining for efficient segment anything. In *CVPR*, 2023. 3, 8

Shilin Xu, Xiangtai Li, Jingbo Wang, Guangliang Cheng, Yunhai Tong, and Dacheng Tao. Fashionformer: A simple, effective and unified baseline for human fashion segmentation and recognition. In *ECCV*, 2022. 10

Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, 2023. 1, 2

Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 2, 4, 6

Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021a. 10

Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 2021b. 9

Zongxin Yang, Jiaxu Miao, Yunchao Wei, Wenguan Wang, Xiaohan Wang, and Yi Yang. Scalable video object segmentation with identification mechanism. *TPAMI*, 2024. 9

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018. 3, 10

Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *IJCV*, 2021. 10

Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 6, 9, 17

Haobo Yuan, Xiangtai Li, Yibo Yang, Guangliang Cheng, Jing Zhang, Yunhai Tong, Lefei Zhang, and Dacheng Tao. Polyphonicformer: Unified query learning for depth-aware video panoptic segmentation. *ECCV*, 2022. 9

Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. In *ECCV*, 2024. 9

Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 10

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:2306.14289*, 2023a. 3, 8, 10, 17

Jiangning Zhang, Xiangtai Li, Jian Li, Liang Liu, Zhucun Xue, Boshen Zhang, Zhengkai Jiang, Tianxin Huang, Yabiao Wang, and Chengjie Wang. Rethinking mobile block for efficient attention-based models. In *ICCV*, 2023b. 10

Renhong Zhang, Tianheng Cheng, Shusheng Yang, Haoyi Jiang, Shuai Zhang, Jiancheng Lyu, Xin Li, Xiaowen Ying, Dashan Gao, Wenyu Liu, et al. Mobileinst: Video instance segmentation on the mobile. In *AAAI*, 2024. 3, 10

Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. Dvis: Decoupled video instance segmentation framework. *ICCV*, 2023c. 18

Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *CVPR*, 2022. 5, 10

Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 5, 9, 17

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 10

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 10

Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018. 1, 3, 10

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv:2306.12156*, 2023. 8, 10

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. In *CVPR*, 2017. 6

Chong Zhou, Xiangtai Li, Chen Change Loy, and Bo Dai. Edgesam: Prompt-in-the-loop distillation for on-device deployment of sam. *arXiv:2312.06660*, 2023. 3, 8, 10

Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod: End-to-end video object detection with spatial-temporal transformers. *TPAMI*, 2022. 10

7 APPENDIX

In this appendix, we add a deeper discussion on our RMP-SAM with existing unified methods from different views in Sec. 7.1. Next, we present more detailed results to support the benchmark and our proposed RMP-SAM in Sec. 7.2. Then, in Sec. 7.3, we present a more detailed visual comparison using our model. Finally, in Sec. 7.4, we discuss several failure cases and potential future work. Moreover, we have appended a short video to introduce our work briefly and append the code for reference.

7.1 MORE METHOD DETAILS

More Implementation Details. We implement our proposed RMP-SAM and benchmark methods using PyTorch and train them in the same environment. For kMaX-DeepLab (Yu et al., 2022), we excluded the auxiliary semantic segmentation loss, instance discrimination loss, and preserving PQ-style loss alignment with Mask2former (Cheng et al., 2022) for a fair comparison. We utilize the AdamW (Loshchilov & Hutter, 2019) optimizer with a weight decay of 0.05, and the learning rate is set to 1e-4 for all methods. We warm up the learning rate in the first 500 iterations using a linearly increased strategy and decay the learning rate at 8 and 11 epochs by a factor of 10. For data augmentation, we use the large-scale jittering (LSJ) augmentation (Ghiasi et al., 2020) with a random scale sampled from the range of 0.1 to 2.0 and followed by a fixed size crop to 1024×1024. For joint dataset training, we sample clip images with a ratio of 1: 25: 1 for COCO-panoptic, Youtube-2019, and SAM datasets. For a fair comparison with efficient SAM models, we adopt the SAM dataset for fine-tuning. In particular, we adopt the co-trained checkpoint and fine our visual prompt segmentation branch using 10% SAM dataset. The training follows the previous works Zhang et al. (2023a).

More Inference Details. For video instance segmentation, we adopt simple object query matching (Li et al., 2022d; Huang et al., 2022) for all methods to keep the fair comparison to assign instance ID. We mainly use point prompts for interactive segmentation, which are more challenging than box prompts. For panoptic segmentation, all hyperparameters are the same. The FPS is obtained on one 40GB A100 GPU for the main paper and supplementary.

Detailed Comparison with SAM-2. We present a more detailed comparison with SAM-2 Ravi et al. (2024) here. (1) For functionality, SAM-2 mainly focuses on video object segmentation and interactive segmentation. Compared with SAM, it adds mask tracking ability on given visual prompts, while our method mainly explores multi-purpose and multi-task real-time segmentation. We unify panoptic segmentation, video instance segmentation, and interactive segmentation in one model with the requirements in real time. (2) For data scale and diversity, SAM-2 builds a large scale dataset and mainly has one purpose: Video Object Segmentation. Our RMP-SAM only involves a small set of public data sources and has multiple purposes: segmentation, mask labeling, mask tracking, and panoptic segmentation. (3) For goals, SAM-2 aims at the production level with large-scale datasets (including internal datasets) co-training. Our RMP-SAM aims at efficient model design and performs well under real-time constraints. (4) Last, our work is concurrent with SAM-2 since our work is also inspired by pioneering SAM.

7.2 MORE EXPERIMENT RESULTS

Benchmark More Methods. In Tab. 7, we benchmark more models and backbones (Zhang et al., 2021; Maaz et al., 2022; Mehta & Rastegari, 2022) as the supplementary to main paper. Again, for various backbones and different heads, our model can achieve better or comparable results across three different tasks while running in real-time. In particular, compared with the dynamic kernel-based method, K-Net, our approach achieves better performance on the R18 and R50 backbones and comparable results on the lightweight backbones (EdgeNexT).

Scale Up Joint Co-training. In the bottom of Tab. 7, we further scale up our model on a large backbone, ConvNeXt large (Liu et al., 2022). From the table, we find there are significant gains in all settings while still running with 8 FPS. This means our proposed multi-purposed segmentation still has room to balance the accuracy and inference speed when adopting the large model. This also verifies the scalability of RMP-SAM.

Method	Backbone	COCO-Panoptic				COCO-SAM	YouTube-VIS 2019	FLOPs	Parameters	FPS
		PQ	SQ	PQ_{th}	PQ_{st}	mIoU	mAP			
YOSO	EdgeNexT	31.8	74.5	35.6	26.1	47.4	35.4	40G	15.3M	28.6
K-Net	R18	33.1	75.7	36.8	27.4	53.6	34.8	124G	25.6M	30.4
K-Net	R50	39.2	78.2	43.9	31.6	56.3	41.7	171G	38.5M	20.9
K-Net	EdgeNexT	38.0	76.7	42.5	31.4	55.7	44.0	108G	19.5M	30.5
Mask2Former	EdgeNexT	39.8	78.8	43.6	34.0	54.2	-	73G	12.0M	25.6
RMP-SAM	R18	39.9	78.6	43.3	34.8	52.7	38.7	60.5G	22.8M	40.3
RMP-SAM	R50	46.9	80.8	51.6	39.8	57.9	46.2	123.0G	47.2M	35.1
RMP-SAM	EdgeNexT	38.0	77.7	42.1	31.9	54.8	46.3	44G	14.3M	31.5
RMP-SAM	ConvNeXt-L	52.3	82.5	58.6	42.8	61.1	55.6	700G	239M	8.6

Table 7: Benchmark More Models on Real-Time multi-purpose Segmentation. The GFLOPs are obtained with 1333×800 inputs.

Table 8: Explore adapter Design on More Methods. We apply our dual adapter on the Mask2Former architecture and find it works well.

Method	Backbone	PQ	mIoU
Mask2Former	R50	43.2	57.0
Mask2Former + adapter	R50	43.0	58.1

Table 9: Compared with real-time click-based interactive segmentation model.

Method	1-IoU	FPS	FLOPs
SAM(L)	55.7	1.2	1315.0
Semantic-SAM(L)	57.0	2.3	745.8G
RMP-SAM (ours)	57.9	35.1	123.0G

Explore Dual Adapter Design on Other Methods. We explore the effectiveness of our dual adapter on Mask2Former (Cheng et al., 2022) and add the adapter module to the last decoder layer of Mask2Former. As shown in Tab. 8, Mask2Former with adapter will improve 1.1 mIoU score compared with origin Mask2Former for interactive segmentation and only drop 0.3 PQ for panoptic segmentation. This result indicates that the adapter module applies to our method and can be generalized to other methods.

Single-granularity Interactive Segmentation. In Tab. 9, we evaluate the 1-click mIoU (denoted as 1-IoU) for SAM (Kirillov et al., 2023), Semantic-SAM (Li et al., 2023a) and our model on COCO Val2017. Our model surpasses SAM and Semantic-SAM in 1-IoU performance while also being faster and requiring less computation.

Comparison with Recent Efficient SAM Models. As shown in Tab. 4, we compare our method with SAM-like methods for coco instance segmentation. In Tab. 10, Tab. 11, Tab. 12. Our method can achieve the stronger results than these counterparts, indicating the effectiveness of model design. We also provide visualization in Fig. 5 for easier and direct comparison.

Comparison with Strong Video Segmentation Models. Several works explore stronger video segmentation designs. We argue the scope of our research is orthogonality to the works. Firstly, since our method is not designed for specific models and our method is trained on multiple purposes tasks, directly comparing our method with previous works Li et al. (2023e); Zhang et al. (2023c) may not be fair. As our core goal is unifying multiple tasks in real-time scenarios and adopting a multi-task co-training strategy, our method may not perform the best on a single task. Secondly, The work Li et al. (2024a) also designs a unified model for video segmentation tasks. However, it cannot perform SAM-like segmentation in real time. We adopt joint image-video data co-training rather on image or video data. Thus, we can keep SAM-like segmentation, panoptic segmentation, and video instance segmentation in one model and run in real time. Thirdly, all these methods focus on achieving strong results rather than real-time design. The work Zhang et al. (2023c) uses extra transformer encoders after heavy Mask2Former architectures, bringing more computation costs. We follow the work Li et al. (2023e); Zhang et al. (2023c), with the modification on replacing pre-training on COCO with our co-training on COCO, COCO-SAM, and YTVIS-2019. Then, we follow the works Li et al. (2023e); Zhang et al. (2023c) and finetune the pre-trained model on YTVIS-2019. As shown in Tab. 14, nevertheless, we achieved an mAP of 47.2 when using the ResNet-50 backbone.

Table 10: Compared with SAM-like methods on COCO instance segmentation. We adopt the Mask R-CNN with ResNet50 as a detector. The bounding box obtained from Mask R-CNN is adopted as a box prompt for SAM-like methods. We report average precision at various thresholds. The AP_s, AP_m, and AP_l mean the mean average precision for small, medium, and large objects.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	#param.	FLOPs(G)
SAM-H	35.6	54.9	38.4	17.2	39.1	51.4	641M	3000
TinySAM	32.7	53.6	34.0	15.5	35.3	48.3	10M	42
MobileSAM	32.3	53.0	33.6	14.7	25.3	48.6	10M	42
EdgeSAM	32.2	53.4	33.3	14.8	35.0	47.0	10M	22
EfficientSAM	32.6	53.3	33.9	16.2	35.2	47.1	10M	-
RMP-SAM(Ours)	33.5	54.6	35.4	14.3	36.3	50.8	47M	35

Table 11: Compared with SAM-like methods on COCO instance segmentation and adopt Mask R-CNN with ResNet101 as a detector.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	#param.	FLOPs(G)
SAM-H	38.4	59.4	41.3	24.1	46.9	56.9	641M	3000
TinySAM	35.4	57.8	36.9	17.6	39.2	51.8	10M	42
MobileSAM	34.9	57.2	36.2	17.0	38.9	52.1	10M	42
EdgeSAM	34.9	57.6	36.2	17.3	38.9	50.3	10M	22
EfficientSAM	35.2	57.6	36.5	18.2	39.0	51.0	10M	-
RMP-SAM(Ours)	35.6	58.0	36.9	17.4	40.2	53.3	47M	35

To provide more comparison, we use ConvNext-L as our backbone and train a model based on the joint co-training described in the paper, achieving a result of 62.2. The results indicate our method can still achieve comparable results with these expert models.

Testing the efficiency across different GPU platforms. We have evaluated several methods from Table 2 across multiple GPU platforms, including A100-40G, A10-22G, and 3090-24G. Unfortunately, we currently do not have access to a V100 GPU and are unable to provide its corresponding results.

All model results for each specific GPU are generated on the same machine. The FPS and GFlops values are calculated using a 1333 x 800 pixels resolution image. We report these results with the ResNet-18 backbone.

7.3 Visual Comparison

Attention Map Visualization. As shown in Fig. 6, we generate the heatmap to illustrate our integrated multi-tasks, which include panoptic segmentation, interactive segmentation, and video instance segmentation. We sample four continuous frames from the YouTube-VIS 2019 validation set and visualize a randomly selected object. The heatmap is the attention score between object/prompt query and image features. The results indicate that despite using a shared decoder across different

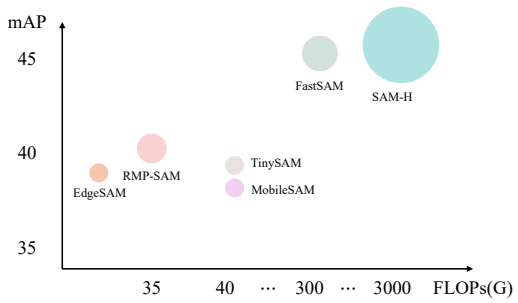

Figure 5: The visualization results of SAM-like methods on COCO instance segmentation. We adopt the ViTDet with ViT-b as the detector. The larger dot means more parameters.

Table 12: Compared with SAM-like methods on COCO instance segmentation and adopt ViTDet with ViT-b as a detector.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	#param.	FLOPs(G)
SAM-H	45.7	69.4	49.8	26.9	49.0	64.1	641M	3000
TinySAM	39.2	64.5	40.4	23.7	42.3	55.7	10M	42
MobileSAM	38.3	63.5	39.3	22.3	41.7	55.3	10M	42
EdgeSAM	39.0	64.8	40.1	23.8	42.3	54.5	10M	22
EfficientSAM	39.4	64.9	40.9	23.8	42.3	55.3	10M	-
RMP-SAM	41.0	65.9	42.5	24.2	44.7	57.3	47M	35

Table 13: Testing the efficiency across different GPU platforms.

Method	GPU	FLOPs	Parameters	FPS
Mask2Former	A100-40G	89.8G	18.6M	31.2
kMaX-DeepLab	A100-40G	87.1G	18.7M	15.0
YOSO	A100-40G	57.3G	18.7M	41.0
RMP-SAM	A100-40G	60.5G	22.8M	40.3
Mask2Former	A10-22G	89.8G	18.6M	10.1
kMaX-DeepLab	A10-22G	87.1G	18.7M	4.3
YOSO	A10-22G	57.3G	18.7M	13.6
RMP-SAM	A10-22G	60.5G	22.8M	14.2
Mask2Former	3090-24G	89.8G	18.6M	25.6
kMaX-DeepLab	3090-24G	87.1G	18.7M	9.0
YOSO	3090-24G	57.3G	18.7M	31.4
RMP-SAM	3090-24G	60.5G	22.8M	32.0

Table 14: Comparison with stronger video instance segmentation methods. Our RMP-SAM can achieve comparable results with more functionalities on image panoptic segmentation and interactive segmentation.

Method	Backbone	Youtube-VIS 2019	COCO-SAM	COCO-Panoptic
UniVS	R50	47.4	-	-
DVIS	R50	51.2	-	-
Tube-Link	R50	52.8	-	-
RMP-SAM	R50	47.2	-	-
UniVS	Swin-L	60.0	-	-
DVIS	Swin-L	63.9	-	-
Tube-Link	Swin-L	64.6	-	-
RMP-SAM	ConvNeXt-L	62.2	60.8	52.0

tasks, each task can effectively learn. This indicates that our design is effective and can significantly reduce the computational complexity of the model.

Comparison on Youtube-VIS dataset. In Fig. 7, we compare our RMP-SAM (right) with a strong baseline Mask2Former (left) with the same ResNet50 backbone. Our methods achieve more consistent tracking and segmentation results in these examples.

Comparison on COCO Panoptic Segmentation dataset. In Fig. 8, We demonstrate that our RMP-SAM model surpasses K-Net in panoptic segmentation tasks. RMP-SAM exhibits enhanced accuracy in processing complex scenes and maintaining fine details, especially in segmenting object edges and managing overlapping objects.

More Interactive Segmentation Results. In Fig. 9, We present further visualizations of interactive segmentation, showcasing segmentation outcomes for 'things' and 'stuff'. In these visualizations, green dots signify manually specified prompts. Our model exhibits precise segmentation abilities

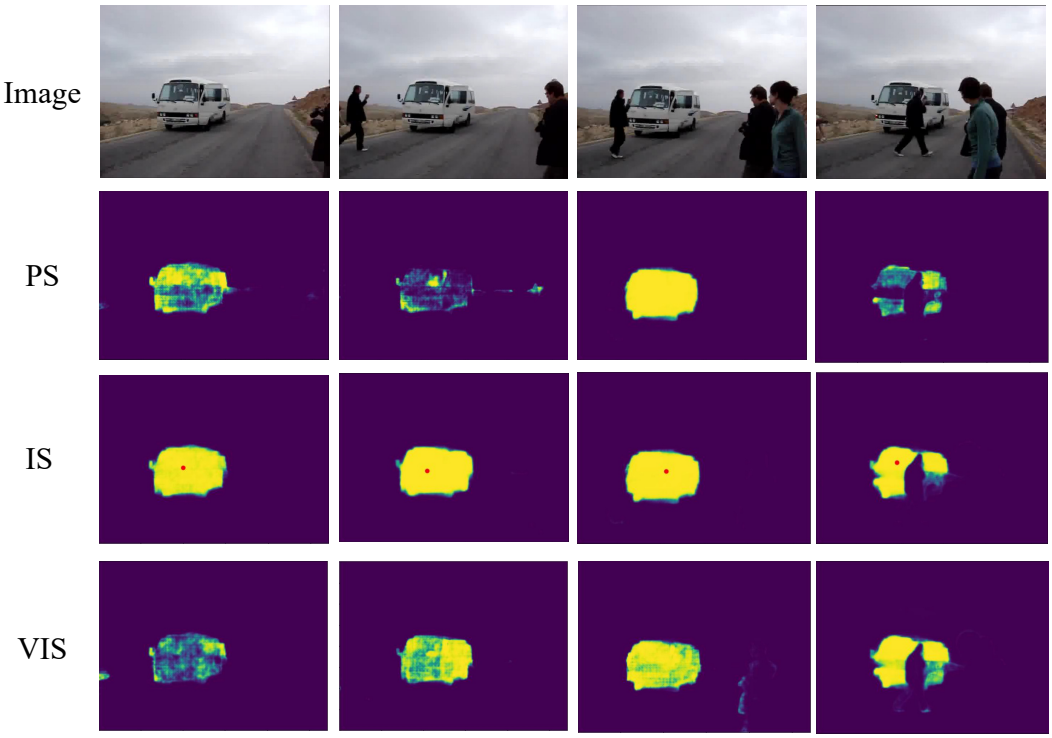

Figure 6: The heatmap shows the results of three distinct tasks: panoptic, interactive, and video instance segmentation, represented by PS, IS, and VIS, respectively. The four test images are sampled from the YouTube-VIS 2019 validation set, and the heatmap is the attention score between object/prompt query and image features. The red point in IS is the point prompt.

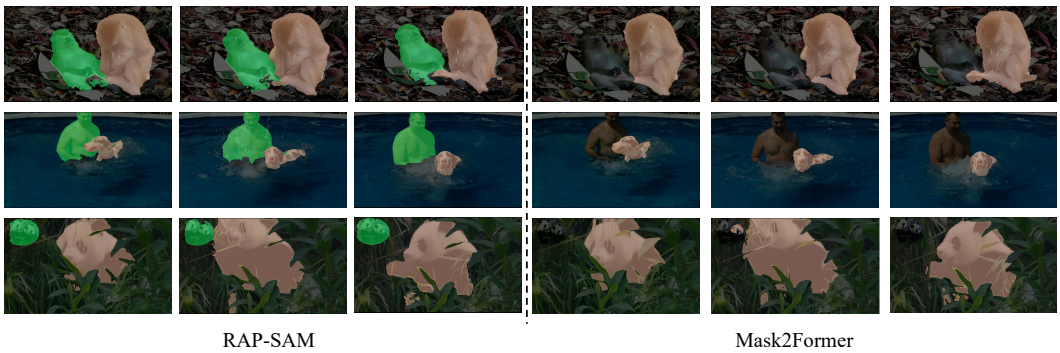

Figure 7: When compared to the Mask2Former(right) on the YouTube-VIS 2019 dataset, our RMP-SAM (left) demonstrates superior performance in recognizing and segmenting objects in certain scenarios, and it also holds an advantage in detailing edges.

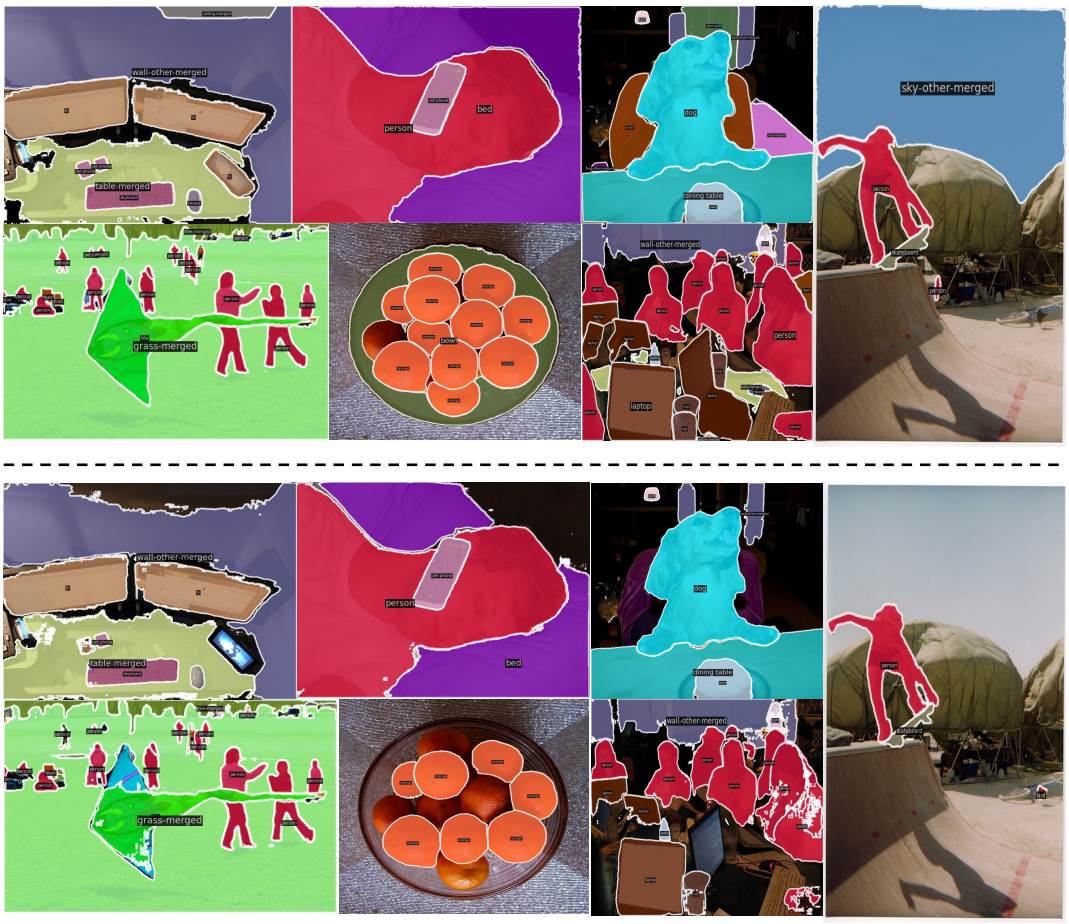

Figure 8: When compared with K-Net (bottom) on the COCO dataset, our RMP-SAM (top) shows a significant advantage, achieving more accurate segmentation of foreground and background, with smoother edges.

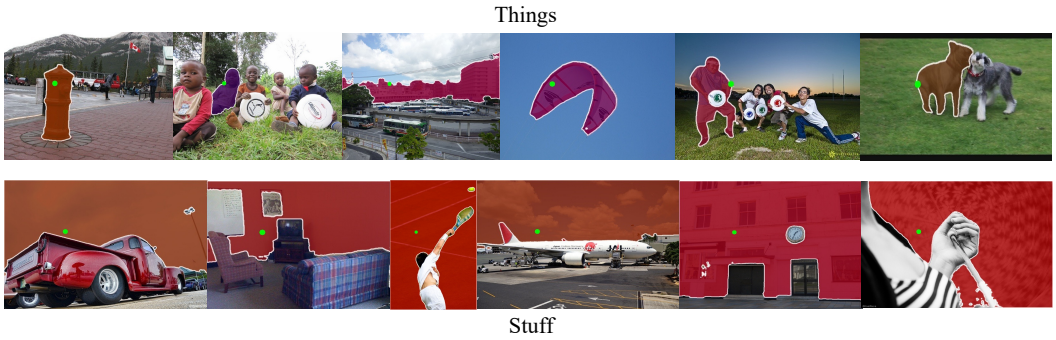

Figure 9: Here are additional results of interactive segmentation. For point prompts or box prompts, RMP-SAM segments out the corresponding object based on the spatial location of the prompt. We have provided both segmentation results for 'things' and 'stuff', demonstrating that RMP-SAM can segment both while maintaining strong performance.

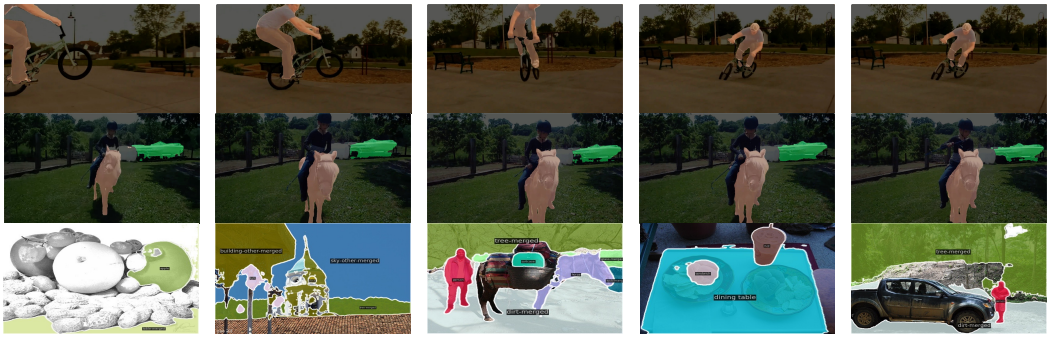

Figure 10: Failure modes of RMP-SAM in image/video segmentation.

for 'things' and 'stuff'. Notably, our model accurately identifies and segments the object even when prompt points are situated on object boundaries.

7.4 Limitations and Future Work

Failure Cases Analysis. We also analyze failure cases in video/image segmentation and identify two typical failure modes of RMP-SAM. Firstly, as illustrated in Fig. 10, when faced with multiple objects with high overlap, RMP-SAM struggles to fully recognize them, resulting in the omission of several masks. Secondly, in crowded scenarios, RMP-SAM faces challenges in recognizing and segmenting all instances with a limited number of instance kernels.

Future Work Discussion. There are several directions to explore for real-time multi-purpose segmentation. The first direction is to achieve better-balanced performance on image, video, and interactive segmentation (since there is still a lot of room for performance, as shown in the last row of Tab. 7). The second direction is to speed up the model and deploy the model on the edge device, such as the iPhone. The third direction is to explore different knowledge distillation approaches to transfer the vision foundation model in real-time multi-purpose models. These will be our future work.