DC-SAM: In-Context Segment Anything in Images and Videos via Dual Consistency

Mengshi Qi, Pengfei Zhu, Xiangtai Li, Xiaoyang Bi, Lu Qi, Huadong Ma, Ming-Hsuan Yang

Abstract—Given a single labeled examples, in-context segmentation aims to segment corresponding objects. This setting, known as one-shot segmentation in few-shot learning, explores the segmentation model's generalization ability and has been applied to various vision tasks, including scene understanding and image/video editing. While recent Segment Anything Models (SAMs) have achieved state-of-the-art results in interactive segmentation, these approaches are not directly applicable to in-context segmentation. In this work, we propose the Dual Consistency SAM (DC-SAM) method based on prompt-tuning to adapt SAM and SAM2 for in-context segmentation of both images and videos. Our key insights are to enhance the features of the SAM's prompt encoder in segmentation by providing high-quality visual prompts. When generating a mask prior from support images, we fuse the SAM features to better align the prompt encoder rather than relying solely on a pre-trained backbone. Then, we design a cycle-consistent cross-attention on fused features and initial visual prompts. This design leverages coarse masks from the SAM mask decoder to ensure consistency between features and visual prompts. Next, a dual-branch design is provided by using the discriminative positive and negative prompts in the prompt encoder. Furthermore, we design a simple mask-tube training strategy to adopt our proposed dual consistency method into the mask tube. Although the proposed DC-SAM is primarily designed for images, it can be seamlessly extended to the video domain with the support of SAM2. Given the absence of in-context segmentation in the video domain, we manually curate and construct the first benchmark from existing video segmentation datasets, named *In-Context Video Object Segmentation (IC-VOS)*, to better assess the in-context capability of the model. Extensive experiments demonstrate that our method achieves 55.5 (+1.4) mIoU on COCO-20^i, 73.0 (+1.1) mIoU on PASCAL-5^i, and a $\mathcal{J}\&\mathcal{F}$ score of 71.52 on the proposed IC-VOS benchmark. Our source code and benchmark are available at https://github.com/zaplm/DC-SAM.

Index Terms—Segment Anything Model, In-context Segmentation, Prompt Generation, Efficient Parameter Tuning

1 Introduction

Recent visual foundation models such as Segment Anything models (SAM and SAM2) [1], [2] have attracted significant attention in recent years. Leveraging tens of millions of images and videos along with over a billion masks, the SAM series demonstrates strong performance in interactive segmentation and proves to be a valuable tool across a wide range of applications, including medical image segmentation [3], open-vocabulary segmentation [4], [5], and more. In particular, numerous efforts [6], [7], [8] have been made to exploit the versatile segmentation capabilities of SAMs in specific domains, such as reasoning [6], extending the recognition ability, or combining with large language models [9].

Despite the state-of-the-art segmentation capabilities, SAMs lack the inherent ability to segment instances of the same category across multiple images given a single instance prompt, an ability we refer to as in-context segmentation inspired by the NLP domain [10], [11]. In this task, the given image and object masks are called support or in-context examples, while the input images are called query images. Previous works explore such abilities with few-shot learning approaches [12], [13], [14], [15], [16], [17], [18], such as calculating the matching distance between query images and support images (known as visual prompts for in-context learning) or modeling object prototypes for better alignment. However, these methods explore in-context learning ability with only a few examples and fail to generalize across diverse domains. Most recently, several models [19], [20], [21], [22] based on in-context learning have been developed where the prompts consist of input-output pairs of visual tasks. Specifically, SegGPT [20] explores in-context segmentation by co-training massive image-mask pairs, leading to good generalization ability across various one-shot segmentation benchmarks. Although this work achieves promising performance, substantial computational resources and extensive annotated segmentation data are required to build such a system. In contrast, prompt tuning [23], [24], [25] offers an effectively alternative adaptive approach that has shown impressive generalization capabilities across diverse domains. This work proposes an adaptive SAM model for in-context image and video segmentation based on prompt tuning.

We note that the existing approaches developed based on SAM [24], [25] mainly depend on the characteristics extracted from the backbone networks and do not fully take advantage of the distinctive properties of the SAM-derived representations, as shown in Figure 1(a). This limitation leads to an oversight of the differences between SAM and backbone network features during the prompt generation process, significantly impacting the accuracy of the generated results. Solely relying on SAM-extracted features for prompt generation often lacks sufficient utilization of the semantic priors of target categories, another critical factor contributing to suboptimal performance. In addition, no suitable benchmarks currently exist to evaluate the in-context segmentation capabilities of video data. Existing benchmarks [26], [27], [28], [29], [30] for video segmentation focus mainly on segmenting and tracking pixels over time. To the best of our knowledge, there are no benchmarks for evaluating this ability using in-context

- *M. Qi, P. Zhu, X. Bi and H. Ma are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China.*
- *X. Li is with Nanyang Technological University, Singapore.*
- *Lu. Qi and M.-H. Yang are with the UC Merced, US.*

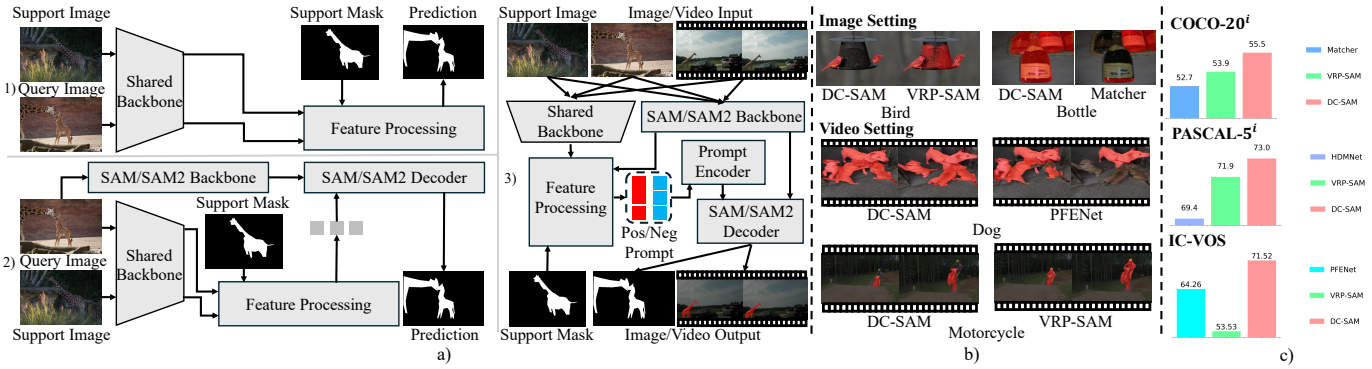


Fig. 1: Overview of the proposed DC-SAM method and IC-VOS benchmark. a) Comparison of the previous few-shot segmentation methods in 1), existing methods based on SAM/SAM2 in 2), and DC-SAM in 3). DC-SAM leverages multi-source features and generates positive and negative prompts by ensuring prompt consistency, integrating with SAM/SAM2 to achieve in-context segmentation for both images and videos; b) Visualization of image and video settings by DC-SAM; c) Quantitative comparison of DC-SAM with state-of-the-art approaches in terms of mIoU on COCO-20^i and PASCAL-5^i, $\mathcal{J}\&\mathcal{F}$ on the IC-VOS benchmark.

examples.

To address the aforementioned problems, we construct the first *In-Context Video Object Segmentation (IC-VOS)* benchmark. We collect examples from existing video segmentation benchmarks and in-context examples from COCO dataset, visually inspecting each example. The IC-VOS dataset comprises 369 videos, averaging 270 frames per video, totaling 99,549 frames across 30 semantic classes. We then benchmark representative methods with SAM2 and propagate the masks from SAM2 to establish the first in-context video benchmarks.

For model design, we propose a novel feature extraction strategy for the prompt generation of SAM, termed *prompt consistency generation*. The meta-architecture is shown in Figure 1(b). Our approach first fuses features extracted by the SAM encoder with those obtained from the backbone network to generate a prior mask for the query image. Experimental results show that combining these two types of features significantly improves model performance. In addition, we design two improvements involving positive and negative features of the SAM's prompt encoder. First, we employ a dual-branch strategy that generates positive and negative prompts using foreground and background masks, respectively. The main idea is to utilize the positive and negative prompt embeddings within SAM's prompt encoder to assign labels to the automatically generated prompts. By leveraging the interaction between positive and negative prompts, we achieve fine-grained control over the generated masks, since more confidential visual cues are provided. Second, we incorporate a Cyclic Consistent Cross-Attention mechanism into the prompt generation process. This mechanism ensures semantic label consistency between input features and queries by aligning highly relevant support pixels with their corresponding categories. It effectively suppresses the propagation of conflicting semantic information. Consequently, this approach ensures that the generated prompts accurately focus on the most critical regions. The proposed prompt consistency method can be easily extended to the video domain, particularly for the SAM2 architecture. Specifically, we design a simple mask tube supervision by extending prompt consistency to the tube mask format. Since our work can be applied to both SAM and SAM2, we term our method DC-SAM, an extension of the SAM series for in-context segmentation of images and videos.

The main contributions of this work are:

- We propose a novel prompt-consistency method based on SAM, called Dual-Consistency SAM (DC-SAM), tailored for one-shot segmentation tasks. It exploits the positive and negative features of the visual prompts, leading to high-quality prompts for in-context segmentation. Furthermore, this design can be easily extended to video tasks by combining the SAM and a new mask tube design.
- We introduce a novel cyclic consistent cross-attention mechanism that ensures the final generated prompts better focus on the key regions requiring prompting. When combined with SAM, this mechanism effectively filters out potentially ambiguous components in the features, further enhancing the accuracy and specificity of in-context segmentation.
- We collect a new video in-context segmentation benchmark, IC-VOS (In-Context Video Object Segmentation), featuring manually curated examples sourced from existing video benchmarks. In addition, we benchmark several representative works in IC-VOS.
- With extensive experiments and ablation studies, the proposed method achieves state-of-the-art performance on various datasets and our newly proposed in-context segmentation benchmarks. DC-SAM achieves 55.5 (+1.4) mIoU on COCO-20^i, 73.0 (+1.1) mIoU on PASCAL-5^i, and a $\mathcal{J}\&\mathcal{F}$ score of 71.52 on the IC-VOS benchmark.

2 Related Work

Segmentation Anything Model. SAM models [1], [2] are proposed to segment objects in both image and video interactively. Using large-scale data co-training, SAM [1] presents a novel data engine and a portable model for segmentation. Subsequent research has utilized SAM as an interactive segmentation tool for various vision tasks, including visual grounding [31], tracking [32], distillation and efficiency modeling [33], medical analysis [34], [35], scene understanding [36], [37], [38], [39], [40] and image generation [23]. To adapt SAM for few-shot and in-context segmentation, PerSAM [23] and Matcher [24] employ patch cosine similarity to identify similar regions for subsequent tasks. VRP-SAM [25] employs a comparable feature extraction method to identify similar regions for few-shot segmentation. It proposes a query-based approach based on cross-attention to generate prompts. In contrast, our method emphasizes the prompt

encoding process in SAM, treating both foreground and background masks as essential constraints. By leveraging positive and negative prompt branches along with a Cyclic Consistent Cross-Attention mechanism, DC-SAM efficiently utilizes SAM's prompt encoder and mask decoder to improve its in-context segmentation performance.

Few-shot Segmentation. The goal of few-shot segmentation [12] is to segment novel semantic categories by giving only a few annotated examples, including few-shot instance segmentation [41], incremental few-shot instance segmentation [42], [43], and generalized few-shot segmentation [43], [44]. Our method primarily focuses on one-shot semantic segmentation, also known as in-context segmentation. This task aims to segment query images using only a single support sample. Recent methods [12], [45], [46], [47], [48] are typically developed based on metric learning by matching spatial location features with semantic centroids. In particular, PFENet [12] employs the last-layer features to generate prior masks and utilizes the mid-level features for segmentation. CycTR [14] introduces cyclic consistency between query and support features within the same class mask regions. Then, two-branch conditional networks [49], [50], [51], 4D dense convolution networks [52], [53], data augmentation methods [54], [55]. Recently, transformer-based models [56], [57], [58], [59] have also been applied to solve the problem. The core idea behind these works is to learn the correspondence in the feature space. In addition, several few-shot segmentation methods are developed based on in-context mask generation [19], [20], [60] and latent diffusion models [61]. However, existing approaches focus on directly designing a model to complete the entire few-shot segmentation task, relying on complex computational frameworks (*e.g.*, Transformer or diffusion models). As a result, the generalizability of these models is limited. Moreover, these methods do not fully utilize the general segmentation prior knowledge inherent in pre-trained foundational models such as SAM [1]. By integrating rich knowledge extracted from SAM and pre-trained backbones (such as ResNet [62] or the vision transformer-based DINO-v2 [63]), our approach generalizes well across various benchmarks while incurring lower computational costs.

Prompt Tuning in Vision. Due to the limited computation resources, most current work cannot re-train the foundation models from scratch. Inspired by prompt tuning in NLP, several methods [7], [64], [65], [66], [67], [68], [69], [70] fine-tune only a tiny portion of parameters to adapt the pre-trained foundation models to various downstream tasks, including image classification, segmentation, image generation and editing. The primary objective of these methods is to optimize the prompting process using the inherent knowledge in the model and then use the refined prompts to enhance performance. This approach preserves the original capabilities of the model and improves its effectiveness in specific tasks. Our work belongs to the prompt tuning approach on SAM, where only the parameters in the prompt encoder are learned. In particular, we aim to learn better correspondence between objects in query and support images, facilitating the model to achieve strong performance in in-context segmentation.

Video Segmentation Benchmarks. Large-scale video object segmentation (VOS) datasets, such as DAVIS [27], MOSE [30], and LVOS [28], are widely used for numerous tasks. In these datasets, the mask of the target object is provided in the first frame, and subsequent frames require the segmentation of the corresponding objects throughout the video. However, in practical applications, when a specific semantic mask (provided along with a support image) needs to be segmented throughout the entire video, existing methods cannot inherently perform this task without requiring annotations in the first frame. Even with the assistance of recent models such as SAM [1], manual clicks or selections of the target object and semantics still require corrections to the edges or interiors, which is cumbersome and labor-intensive. To address these limitations, in this work, we collect the first in-context VOS benchmark to utilize a small number of images as prompts to achieve video semantic segmentation.

TABLE 1: Comparison of the video portion of our proposed benchmark with other well-known VOS datasets. Annotation type indicates whether a mask **M** or a bounding box **B** is provided.

Dataset	Videos	Mean Frames	Total Frames	Classes	Annotations Type
DAVIS [27]	90	69	6298	-	**M**
MOSE [30]	2149	73	159,600	36	**M**
LVOS [28]	220	576	126,280	27	**M**
LVOS v2 [29]	720	412	296,401	44	**M**
YouTube-VOS [26]	4453	27	120,532	94	**M**
UAV20L [71]	20	2934	59,000	5	**B**
IC-VOS (Ours)	369	270	99,549	30	**M**

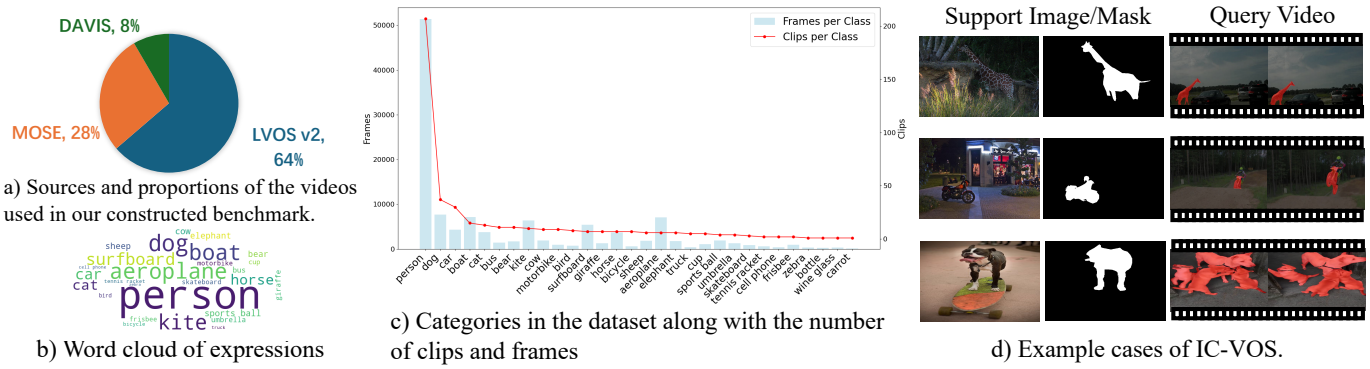
3 In-Context VOS Benchmark

We propose a new in-context video segmentation dataset. The goal is to enable segmentation models to automatically identify and segment target semantics in videos, eliminating the need for manual annotation of the first frame.

3.1 Problem Setting

We first introduce the previous image setting. The *in-context image segmentation* task takes query images, support images, and support masks as inputs. Our model is trained on a dataset D_{train} and evaluated on D_{test}, where D_{train} and D_{test} contain mutually exclusive categories C_{train} and C_{test}, respectively (*i.e.*, $C_{\text{train}} \cap C_{\text{test}} = \emptyset$). Similar to prior work [12], [14], [15], the original datasets are partitioned into multiple folds, each of which comprises a training set D_{train} and a test set D_{test}. For each fold, there exists a query set $Q = \{(I_i^q, M_i^q)\}_{i=1}^{c}$ and a support set $S = \{(I_i^s, M_i^s)\}_{i=1}^{c}$, where c denotes the total number of categories in the training set D_{train} or the test set D_{test}. Here, I and M represent RGB images and their corresponding segmentation masks. The objective is to produce a mask output $\hat{M} \in \mathbb{R}^{H\times W\times 1}$, given a query image $I^q \in \mathbb{R}^{H\times W\times 3}$ and a support set S, where H and W denote the height and width dimensions.

For *in-context video segmentation*, a given image and its corresponding semantic mask are used to segment the associated semantics within a video clip. The image and its mask play the same role as the support image and support mask in in-context image segmentation. Given a support image $I^s \in \mathbb{R}^{H\times W\times 3}$ and its corresponding mask $M^s \in \mathbb{R}^{H\times W\times 1}$, the task involves segmenting each frame of the query video $V^q \in \mathbb{R}^{T\times H\times W\times 3}$ to obtain the associated semantic masks, resulting in a mask tube $M^q_{pred} \in \mathbb{R}^{T\times H\times W\times 1}$, where T means the frame numbers of the video. Thus, this task requires the model to segment and track objects with the same semantics as the given support object, posing greater challenges compared to in-context image segmentation.



a) Sources and proportions of the videos used in our constructed benchmark.

b) Word cloud of expressions

c) Categories in the dataset along with the number of clips and frames

d) Example cases of IC-VOS.

Fig. 2: Overview of our constructed IC-VOS benchmark. a) Distribution of video sources and their proportions. b) Word cloud of expressions. c) Categories in the dataset with the number of clips and frames for each category. d) Example cases illustrating the support image, support mask, and query video.

3.2 Dataset Construction

The benchmark consists of image data and video datasets. The image dataset is divided into a training set and a validation set. We use the training set of the COCO semantic segmentation dataset [72] as the training set for the proposed benchmark. The COCO dataset is renowned for its richness and diversity. It contains a vast number of images and detailed semantic annotations, which provide a solid foundation for training our model. Additionally, the COO validation set serves as the source of image prompts during our validation process.

To reduce annotation costs and leverage high-quality annotations from existing VOS datasets, we use the DAVIS [27], MOSE [30], and LVOS v2 [29] datasets and their annotations as a template to construct our benchmark. However, these VOS datasets use instance-level annotations, which differ from our requirements for semantic-level segmentation. As such, we manually screen the video data from these three datasets to meet the following rules. First, all instances of a given category that appear in the video are annotated in the VOS dataset, at least in the first frame. Second, the categories of the instances must belong to one of the 80 classes in the COCO instance segmentation dataset.

We overlay the annotations from all the videos and save them as GIFs to facilitate manual selection. Only videos satisfying the above criteria will be selected. We archive the annotations for each instance into n binary mask tubes, where n represents the number of qualifying categories in the video.

3.3 Dataset Statistics

The statistics of the proposed benchmark are shown in Table 1. In our proposed benchmark, we collect a total of 369 videos, with an average of 270 frames per video (99,549 frames in the whole set). The videos contain annotations for 30 semantic categories. All videos are used in the validation process.

We also report the source distribution of the source videos. As shown in Figure 2(a), we gather a total of 369 videos, with 63.7% originating from the LVOS v2 [29] dataset, 27.9% from MOSE [30], and 8.4% from DAVIS [27]. The LVOS v2 dataset predominantly features longer videos, whereas shorter videos are more prevalent in MOSE and DAVIS. This distribution results in the average length of our collected videos falling at a moderate level in Table 1. Figure 2(b) and (c) show the word cloud illustrating the proportion of frames for each category in our dataset, as well as the number of clips and frames per category. The categories with a relatively higher number of clips in our dataset are "person," "dog," and "cat." Meanwhile, some categories, despite having fewer clips, contain a large number of frames, such as "kite," "surfboard," and "aeroplane." Figure 2(d) presents multiple test cases from the benchmark, including category examples such as "giraffe," "motorcycle," and "dog." Given a support image (e.g., one containing a "dog") and its corresponding semantic mask, the model must segment the same semantic regions in the video (*e.g.*, track all semantic pixels of "dog" across the video). This process assesses the model's ability to transfer semantic understanding from static images to dynamic videos under one-shot learning conditions.

3.4 IC-VOS Benchmark

We evaluate state-of-the-art few-shot segmentation methods on the proposed dataset to establish the IC-VOS benchmark. The evaluated methods encompass large-model-based approaches (*e.g.,* PerSAM [23], Matcher [24], VRP-SAM [25]) and traditional few-shot segmentation models (PFENet [12], HDMNet [15], AMNet [16]). Since these methods are primarily designed for image-level tasks, we implement modifications to ensure fair and meaningful comparisons. Specifically, these models are integrated with SAM2 [2] to leverage its mask propagation capabilities. The specific modifications are detailed below.

For large-model-based approaches (*e.g.,* PerSAM, Matcher, VRP-SAM), the intermediate outputs of the first frame (*e.g.,* prompts) are fed into SAM2 for inference and propagation. For conventional few-shot segmentation methods (*e.g.,* PFENet, HDMNet, AMNet), the final binary mask prediction from the first frame is used by SAM2 for propagation.

Trainable models (*i.e.,* VRP-SAM, PFENet, HDMNet, AMNet) are retrained on the COCO few-shot segmentation dataset (all four folds combined) for 50 epochs, following the learning rates specified in their original works. For non-trainable models (*i.e.,* Matcher, PerSAM), inference is performed directly without additional training or fine-tuning.

4 Proposed Method

4.1 Overview

As illustrated in Figure 3 and Figure 6, our proposed method builds upon SAM and SAM2. SAM is an interactive segmentation model capable of generating high-quality masks from given

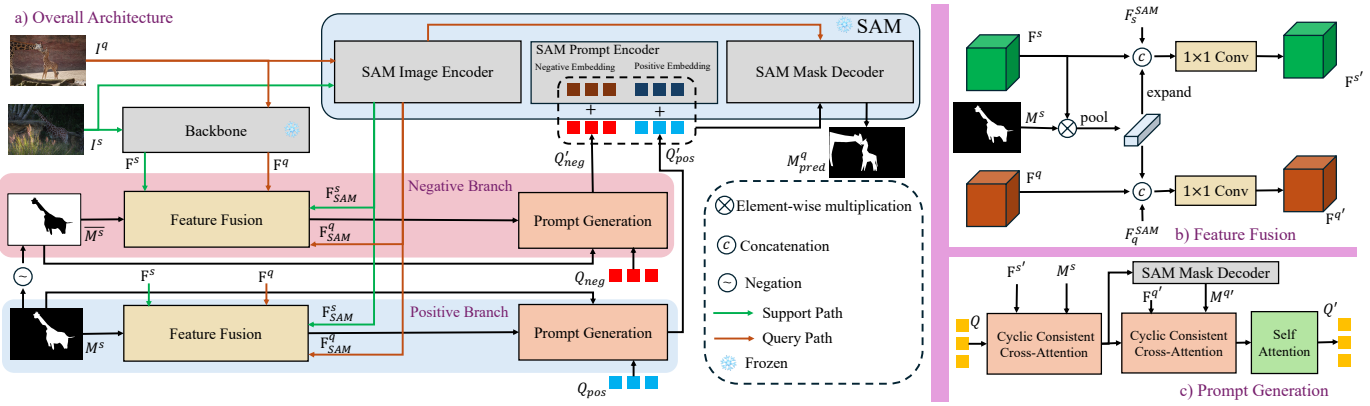


Fig. 3: Overview of the proposed DC-SAM framework. We use positive and negative branches to generate respective prompts, thereby refining the scope of the final generated mask. Additionally, we incorporate SAM features during the prompt generation process to better capture the characteristics of SAM, resulting in more accurate prompt boundaries. During the prompt generation process, we introduce cyclic consistent cross-attention to filter out non-cycle-consistent feature points, enhancing the precision of the prompts.

prompts. The model consists of three primary components: the image encoder, the prompt encoder, and the mask decoder. The image encoder extracts high-quality features for segmentation, and the prompt encoder processes visual prompts (such as boxes, points, masks, or text) to generate the corresponding tokens. The mask decoder receives the image features and the tokens encoded from the prompts, utilizing a bidirectional transformer decoder to generate semantic masks. SAM2 further improves SAM by extending interactive segmentation to video. It also has a memory module, including a memory encoder and a memory bank, to track each pixel in time. Notably, both SAM and SAM2 share an identical prompt encoder design. Our approach efficiently leverages the inherent capabilities of each SAM component during the prompt generation process, particularly for positive and negative points. Thus, our method can be applied to both architectures, resulting in unified modeling of in-context segmentation in both images and videos.

As depicted in Figure 3 (a), our proposed DC-SAM primarily models the SAM prompt encoder by integrating in-context information into the prompt generation process. Specifically, it employs dual branches to generate positive and negative prompts. For each branch, the prompt generation process is bifurcated into feature fusion and consistent prompt generation. This integration ensures that the prompt generation process accounts for SAM features while leveraging complementary insights from mask priors. To refine query-based prompt generation, we introduce a new cycle-consistent cross-attention mechanism to exclude regions irrelevant to the desired prompts. This mechanism is applied twice to produce refined prompts.

4.2 Feature Extraction and Fusion

Similar to the prior design in few-shot segmentation [12], we extract features from both the query image and the support image before the prompt generation process. Specifically, given the input support image I^s and the query image I^q, the backbone initially extracts features from these images. The backbone network can be the pre-trained ResNet-50 [62], VGG-16 [73], or DINOv2 [63]. Following the conventional design [12], [14], we utilize intermediate layer features (from the third and fourth stages of the backbone) and apply convolution operations to reduce dimensions, yielding initial features F_b^s and F_b^q. The prior masks are computed using the support image's corresponding mask M^s and high-level features (from the fifth stage). This mask determines the pixel-wise similarity between the query and support features, retaining the maximum similarity at each pixel and normalizing the similarity map to the range $[0, 1]$ using min-max normalization.

As illustrated in Figure 3, we concatenate the mask-averaged support feature with the query and support features, as well as the features extracted by SAM (F_S^s and F_S^q), which are the same size as the query and support features. These concatenated features are then processed by a 1×1 convolution before being fed into the transformer, ultimately yielding F^s and F^q:

$$F^{s'} = \text{Conv}_{1\times1}\left(\text{Concat}(F^s, M^s \otimes F^s, F^s_{SAM})\right), \quad (15)$$

$$F^{q'} = \text{Conv}_{1\times1}\left(\text{Concat}(F^q, M^s \otimes F^s, F^q_{SAM})\right). \quad (16)$$

By integrating SAM's features, both F^s and F^q become better aligned for the prompt generation process.

4.3 Consistent Prompt Generation

Motivated by CycTR [14], our method includes the cyclic consistent cross-attention mechanism and the self-attention mechanism. However, unlike CycTR, which primarily enforces pixel-level attention consistency between query and support features, our goal is to ensure cyclic consistency for visual prompts compatible with SAM.

Specifically, the initialized random visual prompts are regarded as query Q in Figure 3(c). We compute cross-attention between query and support features, with the fused features $F^{s'}$ and $F^{q'}$ serving as the key and value inputs, respectively, as shown in Figure 3(c). First, we compute the affinity map $A = \frac{QK^T}{\sqrt{d}}, A \in \mathbb{R}^{N \times H_s W_s}$ to evaluate the similarity between the query and each support feature. For each pixel j in the support features, where $j \in \{0, 1, \cdots, H_s W_s - 1\}$, H_s and W_s represent the height and width of the support features respectively, the affinity map A is used to select the index i^* of the query with the highest similarity:

$$i^* = \arg\max_i A(i, j), \quad (17)$$

where $i \in \{0, 1, \cdots, N-1\}$ and N is the number of specified queries. Applying the same method, we identify the pixel index j^* of the support feature that has the highest similarity to the selected query:

$$j^* = \arg\max_j A(i^*, j). \quad (18)$$

Algorithm 1: DC-SAM

Input:
- Support image I^s with mask M^s
- Query image I^q
- Feature extractor f_θ
- SAM Image Encoder f_θ^{SAM}
- SAM Positive/Negative Embeddings E_{pos}, E_{neg}

Output: Predicted mask M^q_{pred}

Step 1: Feature Extraction

$$F^s \leftarrow f_\theta(I^s), \quad \text{Support Feature:} \tag{1}$$
$$F^q \leftarrow f_\theta(I^q), \quad \text{Query Feature:} \tag{2}$$
$$F^s_{SAM} \leftarrow f_\theta^{SAM}(I^s), \quad \text{Support SAM Feature} \tag{3}$$
$$F^q_{SAM} \leftarrow f_\theta^{SAM}(I^q) \quad \text{Query SAM Feature} \tag{4}$$

Step 2: Feature Fusion

$$F^{s'} \leftarrow \text{Conv}_{1\times1}\left(\text{Concat}(F^s, M^s \otimes F^s, F^s_{SAM})\right) \tag{5}$$
$$F^{q'} \leftarrow \text{Conv}_{1\times1}\left(\text{Concat}(F^q, M^s \otimes F^s, F^q_{SAM})\right) \tag{6}$$

Step 3: Prompt Initialization

$$Q_{pos} \leftarrow \text{RandomInit}(), \tag{7}$$
$$Q_{neg} \leftarrow \text{RandomInit}() \tag{8}$$

Step 4: Mediate Prompt Generation

$$Q^{med}_{pos} \leftarrow \text{QCycAttn}(Q_{pos}, F^{s'}, M^s), \tag{9}$$
$$Q^{med}_{neg} \leftarrow \text{QCycAttn}(Q_{neg}, F^{s'}, M^s) \tag{10}$$

Step 5: Pseudo Query Mask Generation

$$M^{q'} \leftarrow \text{SAMMaskDecoder}(Q^{med}_{pos} + E_{pos}, Q^{med}_{neg} + E_{neg}) \tag{11}$$

Step 6: Final Prompt Refinement

$$Q^{'}_{pos} \leftarrow \text{SelfAttn}(\text{QCycAttn}(Q^{med}_{pos}, F^{q'}, M^{q'})), \tag{12}$$
$$Q^{'}_{neg} \leftarrow \text{SelfAttn}(\text{QCycAttn}(Q^{med}_{neg}, F^{q'}, M^{q'})) \tag{13}$$

Step 7: Final Mask Prediction

$$M^q_{pred} \leftarrow \text{SAMMaskDecoder}(Q^{'}_{pos} + E_{pos}, Q^{'}_{neg} + E_{neg}) \tag{14}$$

return M^q_{pred};

Given the flattened mask $M_s \in \mathbb{R}^{H_s W_s}$ corresponding to the support image, cyclic consistency can be determined using the identified pixel indices. Cyclic consistency is satisfied when $M_{s(j)} = M_{s(j^*)}$, and we incorporate this constraint into the cross-attention calculation to encourage the model to learn more cyclically consistent results. Specifically, we calculate a bias $B \in \mathbb{R}^{H_s W_s}$ using the following formula:

$$B_j = \begin{cases} 0, & \text{if } M_{s(j)} = M_{s(j^*)} \\ -\infty, & \text{if } M_{s(j)} \neq M_{s(j^*)} \end{cases}, \tag{19}$$

In this manner, the attention weights for cyclically inconsistent features are set to zero, thereby effectively filtering out features that should not be included in the prompt. For each query $Q_i \in \mathbb{R}^d$, we compute the result of the cross-attention as follows:

$$QCycAttn(Q_i, K, V) = \text{Softmax}(A_i + B) \cdot V, \tag{20}$$

where A denotes the affinity matrix without masking. Since we do not have the masks for the query features as in Equation 19, it is challenging to apply the aforementioned Cyclic Consistent Cross-Attention. Alternatively, the queries can be fed into the SAM mask decoder to generate a pseudo-mask $\hat{M}^q$ for the query features. This also allows us to apply the Cyclic Consistent Cross-Attention to the query features, as shown in the middle of Figure 3(c). The above process is repeated with the refine query Q' and query feature $F^{q'}$ as input. After these two Cyclic Consistent Cross-Attention layers, we further perform a self-attention operation on all queries, forcing the global view of query features. This finally yields the generated prompts P.

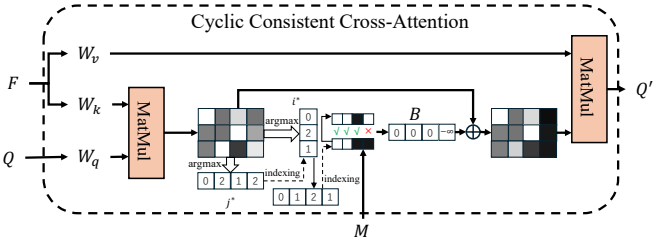


Fig. 4: Illustration of our proposed cyclic consistent cross-attention mechanism. This figure shows the version applied to query features with one head. The "Cyc" operation represents the process described in Equation 19, which ultimately generates a bias to filter out features that are not cycle-consistent.

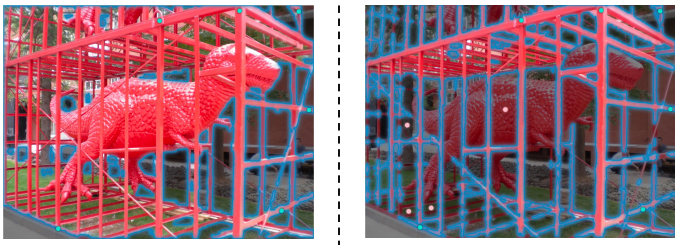


Fig. 5: Comparison of SAM segmentation results with and without negative prompts. (a) Segmentation of the cage using only positive prompts. (b) Segmentation of the cage using both positive and negative prompts. Although not achieving optimal segmentation results, adding negative prompts allowed for better differentiation between the background, the dinosaur, and the cage, resulting in a significantly improved result.

4.4 Dual Prompt-aware Mask Prediction

SAM uses positive and negative prompts during the training process to achieve flexible fine-grained mask control. As shown in Figure 5, using only positive sample points as prompts results in segmentation results with coarse and imprecise mask edges. However, the addition of a negative sample point as a prompt significantly improves the mask edges. Thus, we leverage this inherent characteristic of SAM by combining positive and negative prompts to achieve superior segmentation results.

Specifically, we employ two branches to generate positive and negative prompts, respectively. For the positive branch, we use the support mask M^s to participate during the feature extraction process for the target category and then to generate the positive prompt P_{pos}. For the negative branch, we invert the support mask to obtain the background mask $\overline{M^s} = 1 - M^s$, which indicates the region where the negative prompt should be located, and use it to generate the negative prompt P_{neg}. Furthermore, we utilize SAM's inherent method of encoding positive and negative sampling points, by leveraging SAM's prompt encoder to enable it

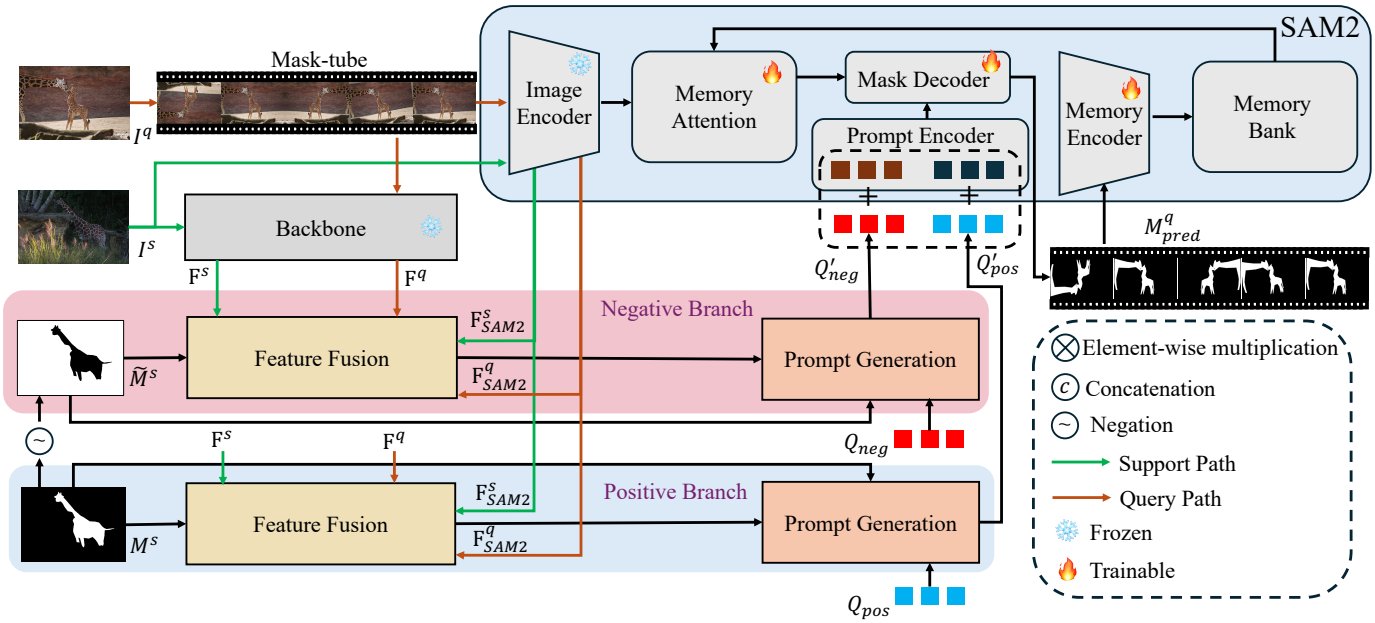


Fig. 6: Illustration of our proposed DC-SAM framework with SAM2. Unlike the image-level framework, we train the entire model for the video to acquire the image-to-video prompt ability. We apply different data augmentation techniques to the query image, and the augmented images compose a mask tube for training.

to perceive the differences between positive and negative prompts. In particular, SAM uses positive and negative embeddings, E_{pos} and E_{neg}, to annotate the input prompts. We adopt a similar approach by labeling the generated positive and negative samples, which result in $P^{'}_{pos} = P_{pos} + E_{pos}$ and $P^{'}_{neg} = P_{neg} + E_{neg}$. These labeled prompts can be fed into SAM's mask decoder to obtain the predicted mask M^q_{pred}.

More importantly, our work can be easily extended to SAM2 for in-context video segmentation, since we only improve the prompt encoder parts. As shown in Figure 6, we design a mask tube prediction for the input query video during training. The mask tube is created by stacking enhanced image masks into a video, encapsulating semantic-level spatiotemporal information. Each mask tube represents the semantics traced in the temporal domain. Without bells and whistles, this simple design further boosts the performance of in-context video segmentation on our proposed dual consistency baseline. In particular, we jointly fine-tune the SAM2 decoder, memory modules, and our proposed dual consistency prompt generation module.

4.5 Optimization and Inference

Training. We adopt the common segmentation training setup for both SAM and SAM2. We directly use two loss functions to guide the segmentation training process. The Binary Cross-Entropy Loss (BCE Loss) supervises the pixel-level binary mask output by the model, as follows:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)], \quad (21)$$

where N denotes the total number of pixels in the image, y_i represents the true label value of pixel i, and p_i is the probability value predicted by the SAM model at pixel i. Additionally, the Dice Loss is employed to provide additional context for pixel-level segmentation, by addressing class imbalance issues, as follows:

$$\mathcal{L}_{Dice} = 1 - \frac{2 \cdot \sum_{i=1}^{N}(p_i \cdot y_i)}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} y_i^2} \quad (22)$$

Consequently, the total loss function used in our model is a combination of the BCE Loss and the Dice Loss:

$$\mathcal{L} = \mathcal{L}_{BCE} + \mathcal{L}_{Dice} \quad (23)$$

Inference. We process the query and support images and the support mask using DC-SAM for the *image in-context segmentation* task. The resulting positive and negative prompts are then combined with the pre-trained positive and negative embeddings to generate point prompts. These combined embeddings, enriched with contextual information, are then fed into the SAM/SAM2 mask decoder to produce final masks, ensuring accurate and contextually relevant segmentation outcomes.

However, for the *video in-context segmentation* task, we apply DC-SAM to the first frame of the input video using the same procedure as in the image setting, thereby obtaining the mask output for the first frame. This mask is then inputted into SAM2 to generate memory embeddings, which are propagated across subsequent video frames utilizing the SAM2 mask decoder. Thus, we can obtain the final mask tube of the entire video.

5 Experiments

Datasets. We first evaluate the proposed method on the IC-VOS benchmark. In addition, we use the setting of [25] and evaluate the proposed DC-SAM on two other widely adopted datasets: PASCAL-5^i [49] and COCO-20^i [74]. The PASCAL-5^i is derived from PASCAL VOC 2012 [75] with additional annotations from SDS [76], encompassing 20 classes. COCO-20^i is based on MS-COCO [72] and includes 80 categories. For each dataset, we perform cross-validation by evenly dividing all classes into four

folds. We adhere to the same class splits specified in [49], [74] for PASCAL-5^i and COCO-20^i, respectively. Specifically, three folds are used for training, while the remaining fold is reserved for testing.

Evaluation Metrics. We use the mean intersection over union (mIoU) as our primary evaluation metric. We denote mIoU $= \frac{1}{C}\sum_{i=1}^{C}$ IoU$_i$, where C represents the total number of classes to be evaluated in each fold, and IoU$_i$ represents the Intersection-over-Union for the i-th class. This metric disregards class-specific distinctions and computes the average value across all classes. For the IC-VOS benchmark, two commonly-used metrics were adopted: region similarity ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$). These metrics are also used by the video sources of our dataset: MOSE [30], LVOS v2 [29], and DAVIS [27]. We calculate the mean of these values as the final score.

Image Benchmarks. We implement DC-SAM using the PyTorch [77] framework, employing various encoders to generate prior masks for support and query images, including VGG-16 [73], ResNet-50 [62], Swin-B [78], and DINO v2-B/14 [63]. We used AdamW [79] as an optimizer and employed a cosine learning rate decay strategy for training. For the COCO-20^i dataset, the model is trained for 50 epochs with a learning rate of 1×10^{-4} and a batch size of 8. For the PASCAL-5^i dataset, training was conducted for 100 epochs with a learning rate of 2×10^{-4} and a batch size of 8. A weight decay of 1×10^{-5} was applied to both datasets during training. The input image size was fixed at 512×512, and no data augmentation techniques were applied. Note that due to the patch size of 14 in DINO v2-B/14, we scale the image size to 896×896 when using DINO v2-B/14 as the prior masks generator to ensure that its output size matches the feature size of SAM. The number of queries in both the positive and negative branches was set to 25.

IC-VOS Benchmark. For the IC-VOS benchmark, our prompt generator is connected with SAM2 to provide prompts for the first image, enabling SAM2 to propagate these prompts throughout the video sequence. The structure and optimizer of the prompt generator are identical to those used in the image benchmark. The differences primarily involve the number of training iterations, training methods, and learning rate configurations. In the first training step, we pre-train our entire framework on COCO images, specifically training the prompt generator and the SAM2 decoder while freezing all other parameters. This step involves $40,000$ iterations with a learning rate of 1×10^{-4}. In the second step, we generate mask tubes from the query image using various image augmentation techniques to enhance the performance of our framework in the video domain. We also unfreeze the memory encoder and memory attention parameters, and this step involves $10,000$ iterations with a learning rate of 1×10^{-5}.

5.1 Main Results

Comparison with SAM-based models and visual foundation models. We evaluate our proposed DC-SAM against other SAM-based methods, including PerSAM [23], Matcher [24], and VRP-SAM [25], on the COCO-20^i dataset. As shown in Table 2, DC-SAM outperforms other current SAM-based methods. We also compare our approach with methods based on visual foundational models, such as Painter [19] and SegGPT [20]. Table 2 shows that the method based on DC-SAM and DINO v2-B surpasses SegGPT by 6%, wdespite SegGPT being trained on large-scale in-domain datasets. These results demonstrate the effectiveness of our approach with less data tuning.

TABLE 2: Comparison with other few-shot segmentation models with foundational models on the COCO-2020^i dataset. Methods marked with * indicate using external data. Methods marked with a † symbol indicate SAM-based models.

Method	F-0	F-1	F-2	F-3	Means
Painter* [19]	31.2	35.3	33.5	32.4	33.1
SegGPT* [20]	56.3	57.4	58.9	51.7	56.1
PerSAM-F [23]†	22.3	24.0	23.4	24.1	23.5
Matcher [24]†	52.7	53.5	52,6	52.1	52.7
VRP-SAM [25]†					
- ResNet50	48.1	55.8	60.0	51.6	53.9
- DINO v2-B	**56.8**	61.0	64.2	59.7	60.4
DC-SAM†					
- ResNet50	50.4	56.0	61.0	54.4	55.5
- DINO v2-B	**56.8**	**62.0**	**67.3**	**61.9**	**62.0**

Comparison with few-shot segmentation methods. We present the evaluation results of our model and recent few-shot segmentation methods on the COCO-20^i and PASCAL-5^i datasets in Table 3. We use different backbones to generate prior masks, including VGG16 [73], ResNet50 [62], and DINO v2-B/14 [63], each representing different architectures and feature extraction capabilities. As illustrated in Table 3, DC-SAM performs favorably across various backbone configurations. Specifically, our model outperforms existing state-of-the-art few-shot segmentation models in every fold of each dataset and under all backbone settings. These results highlight significant performance advantages and the exceptional generalization capabilities of DC-SAM in different scenarios and categories.

Quantitative Results on the IC-VOS Benchmark. As shown in Table 4, DC-SAM surpasses all other evaluated models, achieving a $\mathcal{J}\&\mathcal{F}$ score of 71.52 on the IC-VOS benchmark. This represents an 11.3% improvement over the second-best performing method, PFENet [12]. Our model can effectively segment the corresponding semantic regions in videos based on a few categorical examples.

5.2 Visual Comparisons

Images. Figure 11 shows sample segmentation results of the SAM-based methods on the PASCAL-5^i dataset. For the "bottle" example shown in the first row, DC-SAM segments the objects accurately with the complete contours. For the "bird" example in the second row, DC-SAM segments the objects with fine details. Similarly, in the "bicycle" and "aeroplane" examples presented in the third and fourth rows, DC-SAM consistently performs effectively with no false positives in background regions and accurately captures the complex contours of target objects. Overall, DC-SAM can segment complex objects in the PASCAL-5^i dataset with fine details.

Videos. Figure 8 presents sample results of DC-SAM against PFENet and VRP-SAM on few-shot video semantic segmentation using the proposed benchmark. We show the results from a single video clip in the dataset, and all three models receive the same support image and mask, indicating the target semantic category of motorcycle. DC-SAM accurately identifies the motorcycle and maintains good performance in subsequent segmentation. In contrast, PFENet overlooks the wheels of the motorcycle and also segments the person together with the motorcycle. Similarly, VRP-SAM segments both the person and the motorcycle in the earlier frames.

TABLE 3: Comparison with the state-of-the-art few-shot segmentation methods on COCO-20^i [74] and PASCAL-5^i [75]. The best results are highlighted in **bold**, while the second-best results are underlined.

Method	Image encoder	COCO-20^i					PASCAL-5^i				
		F-0	F-1	F-2	F-3	Mean	F-0	F-1	F-2	F-3	Mean
PFENet [12]	VGG-16	35.4	38.1	36.8	34.7	36.3	56.9	68.2	54.5	52.4	58.0
BAM [13]		36.4	47.1	43.3	41.7	42.1	63.2	70.8	66.1	57.5	64.4
HDMNet [15]		40.7	50.6	48.2	44.0	45.9	64.8	71.4	67.7	56.4	65.1
VRP-SAM [25]		43.6	51.7	50.0	46.5	48.0	70.0	74.7	68.3	61.9	68.7
DC-SAM	VGG-16	44.7	50.2	59.1	50.6	51.2	71.7	77.2	69.0	63.8	70.4
PFENet [12]	ResNet-50	36.5	38.6	34.5	33.8	35.8	61.7	69.5	55.4	56.3	60.8
HSNet [80]		36.3	43.1	38.7	38.7	39.2	64.3	70.7	60.3	60.5	64.0
CyCTR [14]		38.9	43.0	39.6	39.8	40.3	65.7	71.0	59.5	59.7	64.0
SSP [81]		35.5	39.6	37.9	36.7	37.4	60.5	67.8	66.4	51.0	61.4
NTRENet [82]		36.8	42.6	39.9	37.9	39.3	65.4	72.3	59.4	59.8	64.2
DPCN [83]		42.0	47.0	43.3	39.7	43.0	65.7	71.6	69.1	60.6	66.7
VAT [52]		39.0	43.8	42.6	39.7	41.3	67.6	72.0	62.3	60.1	65.5
BAM [13]		39.4	49.9	46.2	45.2	45.2	69.0	73.6	67.6	61.1	67.8
HDMNet [15]		43.8	55.3	51.6	49.4	50.0	71.0	75.4	68.9	62.1	69.4
AMNet [16]		44.9	55.8	52.7	50.6	51.0	71.1	75.9	69.7	63.7	70.1
ABCB [17]		44.2	54.0	52.1	49.8	50.0	72.9	76.0	69.5	64.0	70.6
VRP-SAM [25]		48.1	55.8	60.0	51.6	53.9	73.9	78.3	70.6	65.0	71.9
DC-SAM	ResNet-50	**50.4**	56.0	61.0	54.4	55.5	74.8	**79.1**	**71.4**	66.5	73.0
DC-SAM (SAM2)		49.7	**56.4**	**63.1**	**56.2**	**56.4**	**77.4**	78.5	70.5	**69.4**	**74.0**

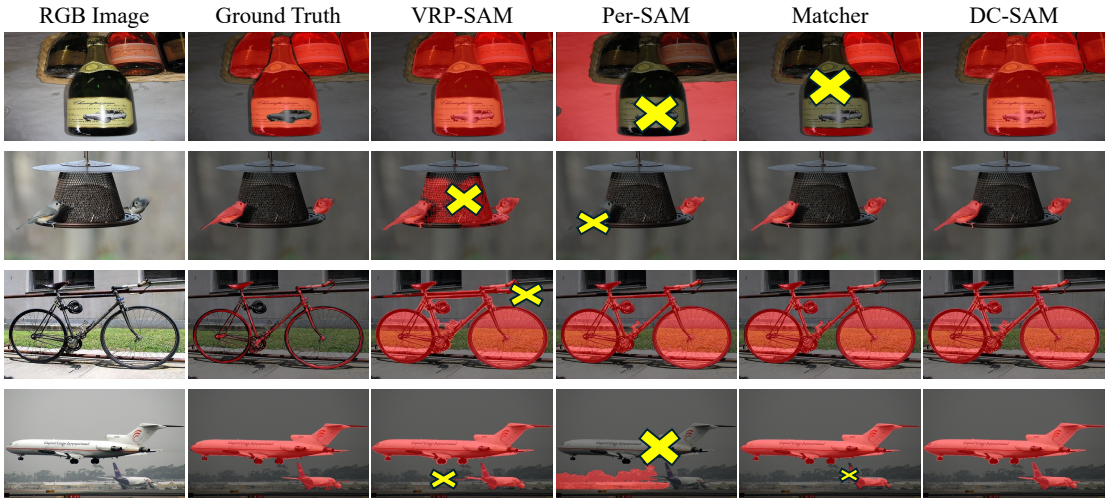


Fig. 7: Comparison of segmentation results from different methods on the PASCAL-5^i. Each row displays an RGB image along with its corresponding ground truth segmentation and the results of the four methods. Notable errors are marked with a yellow "×".

TABLE 4: Results on IC-VOS benchmark. Bold and underlined texts indicate the best and second-best results, respectively.

Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
PerSAM [23] + SAM2	32.23	36.81	34.52
PerSAM-F [23] + SAM2	31.52	36.83	34.18
Matcher [24] + SAM2	26.88	24.00	20.44
VRP-SAM [25] + SAM2	50.77	56.30	53.53
PFENet [12] + SAM2	62.07	66.45	64.26
HDMNet [15] + SAM2	53.07	57.49	55.28
AMNet [16] + SAM2	53.51	58.36	55.94
DC-SAM	**68.38**	**74.65**	**71.52**

TABLE 5: Ablation study on each innovation of the model. We start with VRP-SAM [25] as the baseline and incrementally add our innovations on the PASCAL-5^i.

Ablation	F-0	F-1	F-2	F-3	Means	Δ
VRP-SAM [25]	73.9	78.3	70.6	65.0	71.9	0
+ Pos-Neg Branch	74.0	78.5	70.3	65.5	72.1	+0.2
+ SAM Feature Fusion	**74.8**	**79.6**	70.7	66.2	72.8	+0.6
+ Cyclic Consistent	**74.8**	79.1	**71.4**	**66.5**	**73.0**	+0.2

5.3 Ablation Studies on DC-SAM

Ablation on Each Component. Table 5 shows the effectiveness of each component of the proposed DC-SAM with ResNet50 [62] as the backbone network in the PASCAL-5^i dataset. Note that we begin by using VRP-SAM [25] as the baseline and progressively integrate DC-SAM. Upon the introduction of positive and negative branches, the model has the ability to refine segmentation outcomes by leveraging positive and negative prompts, thus enhancing overall performance across each fold. During feature extraction and fusion, the incorporation of the SAM feature makes the prompt generation progress more aligned. Finally, as shown in the last row, adding prompt consistency for each branch further

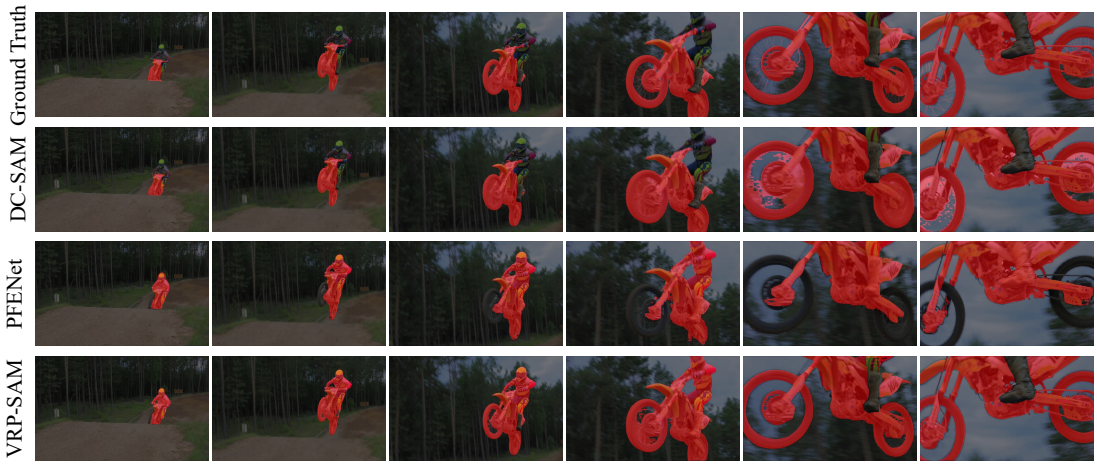


Fig. 8: Visual comparisons of our proposed model with PFENet and VRP-SAM on our proposed benchmark. The support mask in the video indicates the semantic category of motorcycle, and all three models share the same support image and mask.

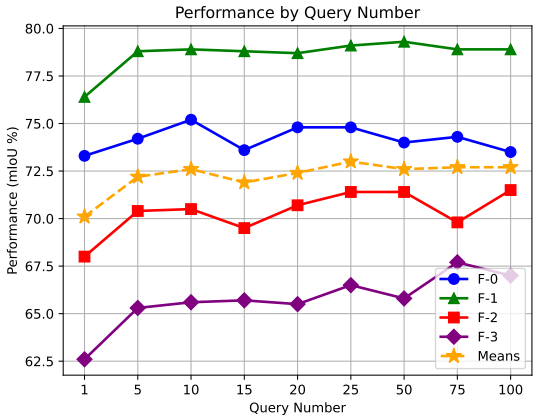


Fig. 9: Ablation study on the number of queries. The x-axis represents the number of queries in one branch. Note that since our DC-SAM consists of both positive and negative branches, the total number of queries is twice the number shown on the x-axis. The y-axis represents the model's performance. These experiments are conducted on the PASCAL-5^i dataset.

TABLE 6: Performance comparison of different pre-trained modules. During fine-tuning, the SAM2 decoder parameters were unfrozen and adjusted. The term "w/o pre-trained" indicates direct training using mask tubes over 50,000 iterations.

Ablation	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
w/o pre-trained	58.72	64.37	61.54
Prompt Generator	64.16	70.30	67.23
Prompt Generator and decoder	**67.57**	**73.62**	**70.59**

leads to improvement over various folders. This guides the model to focus more on the critical areas that require prompts, thus achieving precise segmentation.

Query Number Ablation. We also explore the effect of varying the number of queries Q on the DC-SAM. As shown in Figure 9, the model's performance gradually improves as the number of queries in a single branch increases from 1. However, as the number of queries increases, the model's performance exhibits fluctuations, with improvements in some folds and declines in others. Overall, the model achieves the best average performance when the number of queries in a single branch is set to 25.

TABLE 7: Comparison of fine-tuning with different modules.

decoder	Memory	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
✓		67.57	73.62	70.59
	✓	68.27	74.20	71.23
✓	✓	**68.38**	**74.65**	**71.52**

TABLE 8: Comparison of fine-tuning with and without mask tubes. The total number of training iterations is 50k.

Ablation	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
w/o mask tube	63.85	70.03	66.94
w/ mask tube	**67.57**	**73.62**	**70.59**

5.4 Ablation Study on IC-VOS Benchmark

Pre-training. In Table 6, we show the performance of models trained using different approaches: (1) training the prompt generator and the SAM2 decoder for 50k iterations with data augmentation to generate mask tubes; (2) training only the prompt generator during pre-training; (3) jointly training the prompt generator and SAM2 decoder during pre-training. For fair comparisons, all methods utilizing pre-trained models fine-tune both the prompt generator and the SAM2 decoder. Experimental results demonstrate that pre-training on images followed by fine-tuning with mask tubes achieves the best video segmentation performance. This is because image datasets provide greater diversity, facilitating faster convergence, while fine-tuning on video datasets enhances correspondence learning for mask tubes. Furthermore, training the SAM2 decoder during the image pre-training phase significantly enhances the final video segmentation performance.

Fine-tuning with Ablated Modules. As shown in Table 7, we show the performance variations when fine-tuning different modules of SAM2, including the prompt generator. The experimental results indicate that fine-tuning the memory module (including the memory encoder and memory attention) yields greater improvements compared to fine-tuning only the decoder. Fine-tuning both modules together results in optimal model performance.

Fig. 10: Visualization of two failure cases of our proposed DC-SAM on IC-VOS. We still find missing matching objects due to the occlusion (in the top) and multiple instance inputs with the fast motion (in the bottom).

TABLE 9: Evaluation of DC-SAM in few-shot scenarios on the PASCAL-5^i dataset.

Shots	Backbone	F-0	F-1	F-2	F-3	Means
1	VGG-16 [73]	71.7	77.2	69.0	63.8	70.4
5		76.9 (**+5.2**)	79.6 (**+2.4**)	71.4 (**+2.4**)	69.3 (**+5.5**)	74.3 (**+3.9**)
1	ResNet-50 [62]	74.8	79.1	71.4	66.5	73.0
5		78.2 (**+3.4**)	81.4 (**+2.3**)	72.7 (**+1.3**)	73.8 (**+7.3**)	76.5 (**+3.5**)

TABLE 10: Performance and SAM2 trainable parameters of different fine-tuning versions.

Decoder	Memory	LoRA	# param	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
✓			3.95M	67.57	73.62	70.59
	✓		7.31M	68.27	74.20	71.23
✓	✓		11.26M	**68.38**	**74.65**	**71.52**
✓	✓	✓	0.11M	66.86	72.72	69.79

Mask-tube Fine-tuning. In our proposed DC-SAM, we utilize mask tubes generated by data augmentation to fine-tune the entire model. These mask tubes offer a more diverse set of training examples, thereby enhancing the model's generalization capability across various scenarios. As shown in Table 8, we compare the performance of fine-tuning directly with images versus fine-tuning with mask tubes. Because fine-tuning the memory encoder and memory attention yields substantial improvements for video segmentation, we only train the prompt generator and mask decoder in this comparison to isolate the impact of the mask tubes. These results demonstrate that fine-tuning the model with mask tubes significantly improves video segmentation performance, highlighting the effectiveness of this approach in capturing temporal dynamics and enhancing segmentation consistency.

5.5 More Analysis

TABLE 11: Details of the data split for PASCAL-5^i in the domain shift scenario. Each row represents non-overlapping classes in the training set, corresponding to the respective fold of COCO-20^i.

Fold	Test Classes
0	Aeroplane, Boat, Chair, Dining table, Dog, Person
1	Horse, Sofa, Bicycle, Bus
2	Bird, Car, Potted plant, Sheep, Train, TV/monitor
3	Bottle, Cow, Cat, Motorbike

TABLE 12: Generalization performance on the PASCAL-5^i dataset using Mean IoU (%).

Method	Image Encoder	Means
RPMM [84]	ResNet-50	49.6
PFENet [12]		61.1
RePRI [85]		63.2
VAT-HM [86]		65.1
VRP-SAM [25]		75.9
HSNet [80]	ResNet-101	64.1
DGPNet [87]		70.1
FP-Trans [88]	DeiT-B/16	69.7
DC-SAM	ResNet-50	**76.5**

TABLE 13: Results by the original SAM2 and the fine-tuned SAM2 on the proposed benchmark IC-VOS and the LVOS dataset. Both models were provided with the semantic/instance mask of the first frame of the video.

Dataset	SAM2 Version	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
IC-VOS	original	87.56	92.30	89.93
	fine-tuned	89.07	94.41	91.74
LVOS v2 [29]	original	75.18	81.97	78.58
	fine-tuned	71.25	80.59	75.92

Extension to the Few-shot Setting. To assess the performance of our proposed DC-SAM in the few-shot scenario, we evaluate the model under the 5-shot setting and compare the results with those from the 1-shot setting. Table 9 shows that the 5-shot model consistently surpasses the 1-shot model in performance across all folds. The results indicate that our DC-SAM can be easily extended to the few-shot settings.

Generalization Capability. To evaluate whether DC-SAM retains its generalization capability under domain-shift scenarios, we adopt the same experimental setups from prior studies [12], [25], [80]. Specifically, our model is trained on the COCO-20^i dataset but tested on the PASCAL-5^i dataset. As shown in Table 11, the categories for each fold of PASCAL-5^i are adjusted based on the scheme in [25] to ensure that there is no overlap between the training and testing sets. As demonstrated in Table 12, our model achieves state-of-the-art performance in generalization evaluation.

LoRA Fine-Tuning of SAM2. We examine the effect of using Low-Rank Adaptation (LoRA) [89] to fine-tune the SAM2 model, focusing on parameter reduction and final performance.

TABLE 14: GFLOPs and learnable parameter analysis of our proposed model DC-SAM, VRP-SAM, and PFENet.

Method	FLOPs (G)	Learnable Parameters
VRP-SAM [25]	218.949	1.6M
PFENet [12]	207.582	10.8M
DC-SAM	278.97	1.9M

As demonstrated in Table 10, applying LoRA to fine-tune part of SAM2 reduces the parameter count by 99% compared to full parameter fine-tuning, while retaining 97.6% of the original model performance in terms of $\mathcal{J}\&\mathcal{F}$.

Performance of SAM2 After Fine-Tuning. To evaluate whether fine-tuning SAM2 in our setup affects its original capabilities, we evaluate both the original and fine-tuned versions of SAM2 on our proposed benchmark and the LVOS v2 [29] validation set. For each video clip, we use the mask of the first frame for the target semantic or instance segmentation, and we use SAM2 to infer the corresponding masks for all subsequent frames. As demonstrated in Table 13, the fine-tuned SAM2 outperforms the original SAM2 on our proposed IC-VOS, demonstrating enhanced capabilities in semantic video segmentation. For the LVOS v2 video instance segmentation benchmark, the fine-tuned SAM2 shows a slight performance loss but retains 96.6% of the original performance based on the $\mathcal{J}\&\mathcal{F}$ metric. With more trained data or co-training with VOS data, DC-SAM is likely to achieve a better performance trade-off on both IC-VOS and VOS, which will be part of our future work.

Failure Cases. Figure 10 shows some failure segmentation results of DC-SAM. The first row shows a scenario where tracking fails due to occlusions, leading to incorrect subsequent segmentation results from the propagation module of SAM2. The second row presents a case where the target semantic category is "dog". During fast motions involving the dog and a toy car, slight tracking errors occur despite initially accurate segmentation. In the third frame, a small portion of the mask erroneously tracks the wheel of the toy car, and the subsequent propagation process accumulates this error.

6 Conclusion

In this paper, we present a prompt tuning-based method to adapt visual foundation models, SAM and SAM2, to better support in-context learners. The core idea is to leverage the features of the SAM prompt encoder to generate more fine-grained visual prompts with dual consistency. Given the fused SAM features, we can use both positive and negative queries to generate visual prompts. During the generation process, we design a cycle-consistent attention in each branch. Furthermore, we propose a new dataset, IC-VOS, to benchmark existing representative methods combined with SAM2. Our proposed DC-SAM performs favorably against existing models on several few-shot segmentation benchmarks, even with SAM2. By adding a simple mask tube to DC-SAM, it also achieves state-of-the-art performance on the IC-VOS benchmark. Extensive analysis shows the effectiveness, efficiency, and generalization of our approach.

Future Work. The proposed model performs well in the few-shot image-to-image segmentation and the few-shot image-to-video segmentation task that we introduced. However, it still has some limitations, such as tracking errors due to occlusion, inaccurate prompts in the first frame, and large motion. We will address these issues in the future work.

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *ICCV*, 2023. 1, 2, 3

[2] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollar, and C. Feichtenhofer, "SAM 2: Segment anything in images and videos," in *ICLR*, 2025. 1, 2, 4

[3] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, 2024. 1

[4] X. Wang, S. Li, K. Kallidromitis, Y. Kato, K. Kozuka, and T. Darrell, "Hierarchical open-vocabulary universal image segmentation," *NeurIPS*, 2024. 1

[5] H. Zhou, T. Shen, X. Yang, H. Huang, X. Li, L. Qi, and M.-H. Yang, "Rethinking evaluation metrics of open-vocabulary segmentaion," in *arXiv*, 2023. 1

[6] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," in *CVPR*, 2024. 1

[7] H. Yuan, X. Li, C. Zhou, Y. Li, K. Chen, and C. C. Loy, "Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively," in *ECCV*, 2024. 1, 3

[8] H. Yuan, X. Li, T. Zhang, Z. Huang, S. Xu, S. Ji, Y. Tong, L. Qi, J. Feng, and M.-H. Yang, "Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos," *arXiv*, 2025. 1

[9] L. Qi, Y.-W. Chen, L. Yang, T. Shen, X. Li, W. Guo, Y. Xu, and M.-H. Yang, "Generalizable entity grounding via assistance of large language model," in *arXiv*, 2024. 1

[10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, 2022. 1

[11] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang *et al.*, "A survey on in-context learning," in *EMNLP*, 2024, pp. 1107–1128. 1

[12] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *TPAMI*, 2020. 1, 3, 4, 5, 8, 9, 11, 12

[13] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *CVPR*, 2022. 1, 9

[14] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," *NeurIPS*, 2021. 1, 3, 5, 9

[15] B. Peng, Z. Tian, X. Wu, C. Wang, S. Liu, J. Su, and J. Jia, "Hierarchical dense correlation distillation for few-shot segmentation," in *CVPR*, 2023. 1, 3, 4, 9

[16] Y. Wang, N. Luo, and T. Zhang, "Focus on query: Adversarial mining transformer for few-shot segmentation," *NeurIPS*, vol. 36, pp. 31 524–31 542, 2023. 1, 4, 9

[17] L. Zhu, T. Chen, J. Yin, S. See, and J. Liu, "Addressing background context bias in few-shot segmentation through iterative modulation," in *CVPR*, 2024. 1, 9

[18] X. Shi, D. Wei, Y. Zhang, D. Lu, M. Ning, J. Chen, K. Ma, and Y. Zheng, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *ECCV*, 2022. 1

[19] X. Wang, W. Wang, Y. Cao, C. Shen, and T. Huang, "Images speak in images: A generalist painter for in-context visual learning," in *CVPR*, 2023. 1, 3, 8

[20] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, "Seggpt: Towards segmenting everything in context," in *ICCV*, 2023. 1, 3, 8

[21] Z. Fang, X. Li, X. Li, J. M. Buhmann, C. C. Loy, and M. Liu, "Explore in-context learning for 3d point cloud understanding," 2024. 1

[22] X. Wang, Z. Fang, X. Li, X. Li, C. Chen, and M. Liu, "Skeleton-in-context: Unified skeleton sequence modeling with in-context learning," *CVPR*, 2024. 1

[23] R. Zhang, Z. Jiang, Z. Guo, S. Yan, J. Pan, H. Dong, Y. Qiao, P. Gao, and H. Li, "Personalize segment anything model with one shot," in *ICLR*. 1, 2, 4, 8, 9

[24] Y. Liu, M. Zhu, H. Li, H. Chen, X. Wang, and C. Shen, "Matcher: Segment anything with one shot using all-purpose feature matching," in *ICLR*. 1, 2, 4, 8, 9

[25] Y. Sun, J. Chen, S. Zhang, X. Zhang, Q. Chen, G. Zhang, E. Ding, J. Wang, and Z. Li, "Vrp-sam: Sam with visual reference prompt," in *CVPR*, 2024. 1, 2, 4, 7, 8, 9, 11, 12

[26] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018. 1, 3

[27] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017. 1, 3, 4, 8

[28] L. Hong, W. Chen, Z. Liu, W. Zhang, P. Guo, Z. Chen, and W. Zhang, "Lvos: A benchmark for long-term video object segmentation," in *ICCV*, 2023, pp. 13 480–13 492. 1, 3

[29] L. Hong, Z. Liu, W. Chen, C. Tan, Y. Feng, X. Zhou, P. Guo, J. Li, Z. Chen, S. Gao *et al.*, "Lvos: A benchmark for large-scale long-term video object segmentation," *arXiv preprint arXiv:2404.19326*, 2024. 1, 3, 4, 8, 11, 12

[30] H. Ding, C. Liu, S. He, X. Jiang, P. H. Torr, and S. Bai, "Mose: A new dataset for video object segmentation in complex scenes," in *ICCV*, 2023, pp. 20 224–20 234. 1, 3, 4, 8

[31] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *ECCV*, 2024. 2

[32] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang, "Segment and track anything," *arXiv preprint arXiv:2305.06558*, 2023. 2

[33] C. Zhou, X. Li, C. C. Loy, and B. Dai, "EdgeSAM: Prompt-in-the-loop distillation for on-device deployment of SAM," *arXiv preprint arXiv:2312.06660*, 2023. 2

[34] J. Wu, Z. Wang, M. Hong, W. Ji, H. Fu, Y. Xu, M. Xu, and Y. Jin, "Medical sam adapter: Adapting segment anything model for medical image segmentation," *Medical Image Analysis*, p. 103547, 2025. 2

[35] C. Chen, J. Miao, D. Wu, A. Zhong, Z. Yan, S. Kim, J. Hu, Z. Liu, L. Sun, X. Li *et al.*, "Ma-sam: Modality-agnostic sam adaptation for 3d medical image segmentation," *Medical Image Analysis*, vol. 98, p. 103310, 2024. 2

[36] C. Lv, S. Zhang, Y. Tian, M. Qi, and H. Ma, "Disentangled counterfactual learning for physical audiovisual commonsense reasoning," in *NeurIPS*, 2023. 2

[37] M. Qi, J. Qin, Y. Yang, Y. Wang, and J. Luo, "Semantics-aware spatial-temporal binaries for cross-modal video retrieval," *IEEE Trans. Image Process.*, vol. 30, pp. 2989–3004, 2021. 2

[38] M. Qi, Y. Wang, A. Li, and J. Luo, "Stc-gan: Spatio-temporally coupled generative adversarial networks for predictive scene parsing," *TIP*, vol. 29, pp. 5420–5430, 2020. 2

[39] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *CVPR*, 2019. 2

[40] M. Qi, Y. Wang, J. Qin, A. Li, J. Luo, and L. Van Gool, "stagnet: An attentive semantic rnn for group activity and individual action recognition," *TCSVT*, vol. 30, no. 2, pp. 549–565, 2020. 2

[41] Z. Fan, J.-G. Yu, Z. Liang, J. Ou, C. Gao, G.-S. Xia, and Y. Li, "Fgn: Fully guided network for few-shot instance segmentation," in *CVPR*, 2020. 3

[42] D. A. Ganea, B. Boom, and R. Poppe, "Incremental few-shot instance segmentation," in *CVPR*, 2021. 3

[43] Y. Han, J. Zhang, Y. Wang, C. Wang, Y. Liu, L. Qi, X. Li, and M.-H. Yang, "Reference twice: A simple and unified baseline for few-shot instance segmentation," *TPAMI*, 2024. 3

[44] Y. Li, H. Zhu, J. Ma, C. S. Teo, C. Xiang, P. Vadakkepat, and T. H. Lee, "Towards generalized and incremental few-shot object detection," *arXiv preprint arXiv:2109.11336*, 2021. 3

[45] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," in *NeurIPS*, 2022. 3

[46] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning non-target knowledge for few-shot semantic segmentation," in *CVPR*, 2022. 3

[47] J. Liu, Y. Bao, G.-S. Xie, H. Xiong, J.-J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *CVPR*, 2022. 3

[48] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *CVPR*, 2021. 3

[49] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017. 3, 7, 8

[50] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Conditional networks for few-shot semantic segmentation," in *ICLR*, 2018. 3

[51] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer," in *ICCV*, 2021. 3

[52] S. Hong, S. Cho, J. Nam, S. Lin, and S. Kim, "Cost aggregation with 4d convolutional swin transformer for few-shot segmentation," in *ECCV*, 2022. 3, 9

[53] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *ICCV*, 2021. 3

[54] N. Tritrong, P. Rewatbowornwong, and S. Suwajanakorn, "Repurposing gans for one-shot semantic part segmentation," in *CVPR*, 2021. 3

[55] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, "Datasetgan: Efficient labeled data factory with minimal human effort," in *CVPR*, 2021. 3

[56] J.-W. Zhang, Y. Sun, Y. Yang, and W. Chen, "Feature-proxy transformer for few-shot segmentation," in *NeurIPS*, 2022. 3

[57] X. Shi, D. Wei, Y. Zhang, D. Lu, M. Ning, J. Chen, K. Ma, and Y. Zheng, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *ECCV*, 2022. 3

[58] G.-S. Xie, H. Xiong, J. Liu, Y. Yao, and L. Shao, "Few-shot semantic segmentation with cyclic memory network," in *ICCV*, 2021. 3

[59] D. Kim, J. Kim, S. Cho, C. Luo, and S. Hong, "Universal few-shot learning of dense prediction tasks with visual token matching," in *ICLR*, 2023. 3

[60] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros, "Visual prompting via image inpainting," in *NeurIPS*, 2022. 3

[61] R.-Z. Qiu, Y.-X. Wang, and K. Hauser, "Alignddiff: aligning diffusion models for general few-shot segmentation," in *ECCV*, 2024. 3

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 3, 5, 8, 9, 11

[63] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *TMLR*, pp. 1–31, 2024. 3, 5, 8

[64] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *ICML*, 2019. 3

[65] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *ACL*, 2021. 3

[66] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022. 3

[67] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *ACL*, 2022. 3

[68] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," in *ACL*, 2021. 3

[69] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *EMNLP*, 2021. 3

[70] J. Wu, X. Li, C. Si, S. Zhou, J. Yang, J. Zhang, Y. Li, K. Chen, Y. Tong, Z. Liu *et al.*, "Towards language-driven video inpainting via multimodal large language models," *CVPR*, 2024. 3

[71] U. Benchmark, "A benchmark and simulator for uav tracking," in *ECCV*, 2016. 3

[72] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 4, 7

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 5, 8, 11

[74] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *ICCV*, 2019. 7, 8, 9

[75] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010. 7, 9

[76] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011. 7

[77] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, 2019. 8

[78] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021. 8

[79] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*. 8

[80] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *ICCV*, 2021. 9, 11

[81] Q. Fan, W. Pei, Y.-W. Tai, and C.-K. Tang, "Self-support few-shot semantic segmentation," in *ECCV*, 2022. 9

[82] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning non-target knowledge for few-shot semantic segmentation," in *CVPR*, 2022. 9

[83] J. Liu, Y. Bao, G.-S. Xie, H. Xiong, J.-J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *CVPR*, 2022. 9
[84] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *ECCV*, 2020. 11
[85] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *CVPR*, 2021. 11
[86] S. Moon, S. S. Sohn, H. Zhou, S. Yoon, V. Pavlovic, M. H. Khan, and M. Kapadia, "Hm: Hybrid masking for few-shot segmentation," in *ECCV*, 2022. 11
[87] J. Johnander, J. Edstedt, M. Felsberg, F. S. Khan, and M. Danelljan, "Dense gaussian processes for few-shot segmentation," in *ECCV*, 2022. 11
[88] J.-W. Zhang, Y. Sun, Y. Yang, and W. Chen, "Feature-proxy transformer for few-shot segmentation," in *NeurIPS*, 2022. 11
[89] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *ICLR*. 11

This supplementary material systematically demonstrates the extensive application effects of DC-SAM in image-to-image in-context segmentation and image-to-video in-context segmentation. To comprehensively validate the effectiveness and robustness of the method, we have carefully selected 20 representative image cases (covering various target categories) and 8 typical video sequences (including challenging scenarios such as dynamic target tracking and complex background changes) for visualization. By comparing the output results of DC-SAM with those of other state-of-the-art methods, the technical advantages of our approach in detail preservation and accurate prompting are intuitively highlighted. These rich visual examples not only corroborate the quantitative analysis conclusions presented in the main text but also provide multidimensional empirical references. They can be read in conjunction with the methodology section of the main text to gain a more comprehensive understanding.

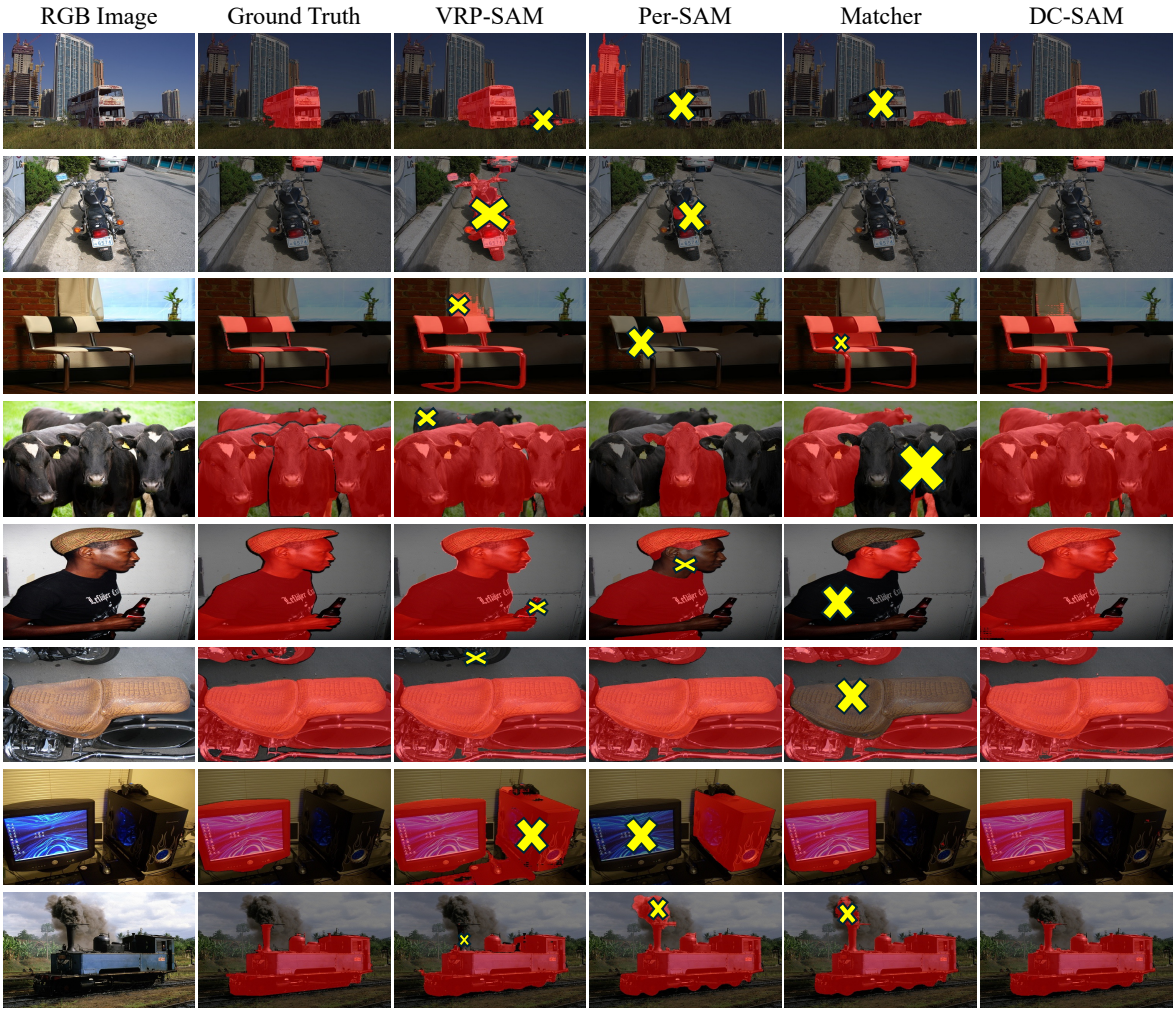


Fig. 11: Comparison of one-shot segmentation results on the PASCAL-5^i dataset.

Fig. 12: Comparison of semantic segmentation results for the “Sheep” category on the IC-VOS dataset.

Fig. 13: Comparison of semantic segmentation results for the “Dog” category on the IC-VOS dataset.

Fig. 14: Comparison of semantic segmentation results for the “Cup” category on the IC-VOS dataset.