Joint Modeling of Feature, Correspondence, and a Compressed Memory for Video Object Segmentation

Jiaming Zhang Yutao Cui Gangshan Wu Limin Wang [⊠] State Key Laboratory for Novel Software Technology, Nanjing University, China

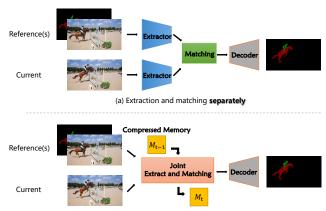
jiamming.zhang@gmail.com cuiyutao@smail.nju.edu.cn {gswu,lmwang}@nju.edu.cn

Abstract

Current prevailing Video Object Segmentation (VOS) methods usually perform dense matching between the current and reference frames after extracting their features. One on hand, the decoupled modeling restricts the targets information propagation only at high-level feature space. On the other hand, the pixel-wise matching leads to a lack of holistic understanding of the targets. To overcome these issues, we propose a unified VOS framework, coined as Joint-**Former**, for joint modeling the three elements of feature, correspondence, and a compressed memory. The core design is the Joint Block, utilizing the flexibility of attention to simultaneously extract feature and propagate the targets information to the current tokens and the compressed memory token. This scheme allows to perform extensive information propagation and discriminative feature learning. To incorporate the long-term temporal targets information, we also devise a customized online updating mechanism for the compressed memory token, which can prompt the information flow along the temporal dimension and thus improve the global modeling capability. Under the design, our method achieves a new state-of-art performance on DAVIS 2017 val/test-dev (89.7% and 87.6%) and YouTube-VOS 2018/2019 val (87.0% and 87.0%) benchmarks, outperforming existing works by a large margin.

1. Introduction

Semi-supervised Video Object Segmentation (VOS) is a fundamental and challenging task in computer vision with many potential applications [29,55], *i.e.*, interactive video editing, video inpainting, and autonomous driving, which aims to track and segment the object(s) across a video sequence based on the mask annotation given in the first frame only. Due to the limited targets information provided, there comes out a core problem of *how to capture discriminative*



(b) Joint Modeling of Feature, Correspondence, and our Compressed Memory

Figure 1. The pipeline of existing VOS works (a) and ours (b). (a) Existing works perform feature extraction and matching separately. (b) Our framework jointly models features, correspondence, and the compressed memory.

representation and propagate the information both at finegrained level and holistic level.

The propagation-based methods [9, 19, 30, 32, 38, 46, 48] give their answer by iteratively propagating the masks with temporal correlations frame by frame. Besides, the prevailing matching-based methods [4, 16, 40, 49] pursue a different direction, that is, performing dense matching between the top-level feature of current and reference frames by calculating the correspondence map. Furthermore, several memory-based methods [6, 8, 22, 31, 35] leverage a memory bank to store the multiple reference frames with masks as spatially fine-grained memory. Despite the significant success they have achieved, there still exist the following drawbacks. i) They all follow a fixed pipeline of extractthen-matching as shown in Fig 1(a), i.e., first extracting the feature of current and reference frames independently, and then performing integration only at high-level feature space to propagate the target information. On one hand, this inevitably makes the model struggle to capture targetspecific feature at the lower levels, which is of vital importance for fine-grained segmentation and discriminative learning. On the other hand, the decoupled matching mod-

^{⊠:} Corresponding author (lmwang@nju.edu.cn).

ule lacks the flexibility of enjoying the development of large-scale pre-training, such as the Masked Image Modeling (MIM) [14,15]. ii) Typically, the matching is processed in a pixel-wise way, *i.e.*, performing dense propagation between all elements in reference features and the current feature. The modeling of holistic targets, however, tends to be overlooked, which may lead to deficient discriminative capability, especially when facing distractors.

To address the above issues, we bring a new perspective to VOS that the modeling of feature, fine-grained correspondence and compressed memory should be coupled in a compact transformer architecture (refer to Fig. 1). First, the joint modeling can unlock the potential of capturing extensive and discriminative target-specific feature, unleash the strong power of MIM pre-training for all the three processes and also help each other. Second, unlike existing matching which provides spatially fine-grained features at the pixel level, our compressed memory treats each object as a whole instance, so as to provide a comprehensive and discriminating understanding for the objects.

Driven by the analysis, we propose a unified framework that jointly models feature, matching, and the compressed memory, coined as JointFormer. Specifically, we concatenate the flatten current frame, the reference frames with masks, and the compressed memory embedding into token sequences, and feed them into stacks of transformerbased Joint Block. With the help of the attention mechanism, the joint blocks perform iterative modeling to propagate the target information in multiple flows, hence achieving the above goals. We also in-depth investigate the impact of various methods of information propagation. Especially, the presented *compressed memory* stores only one token for each target, so as to represent it at the instance level. To incorporate the long-term temporal targets information, we also design a customized online updating mechanism for the compressed memory token. In detail, we take the multilevel memory tokens, generated from the backbone or decoder in previous frame, as the temporal memory for the current compressed token, which enables the targets information to propagate along the temporal dimension. Under these designs, we formulate a compact, unified and concise VOS framework, allowing for accurate and robust video object segmentation.

Our contributions can be summarized as follows:

- We propose a unified network, termed as *JointFomer*, to jointly model feature, correspondence, and the compressed memory. The compact and concise framework enables extensive information propagation and discriminative learning.
- We develop a customized online updating mechanism for the compressed memory, which helps to prompt the information flow along the temporal dimension and

- thus improving the global modeling capability.
- Comprehensive experiments show that our *JointFomer* achieves state-of-the-art performance with 89.7% and 87.6% on DAVIS 2017 [34] validation and test-dev split, 87.0% and 87.0% on YouTube-VOS [47] 2018 & 2019 validation split.

2. Related Works

2.1. Semi-supervised VOS

The Semi-supervised VOS task aims to track and segment object(s) in the video sequence based on its mask given in the first frame. Early online learning-based methods [2, 17, 26, 41, 45] rely on fine-tuning networks during testing leads to inefficiencies. Propagation-based methods [9, 19, 30, 32, 38, 46, 48] iteratively propagate the segmentation masks with temporal correlations, but they are prone to drifting and struggle with occlusions. Matchingbased methods [4, 16, 40, 49, 51] calculate the correspondence pixel map between the current and reference frames. Recent memory-based methods [6-8,22,27,31,35] propose an external memory bank to store the past frames in order to address the context limitation. Some works [13, 28, 50, 52] adopt transformer blocks [39] for better matching, but train the blocks from scratch without pre-training. Despite the promising results, they all follow a fixed pipeline that extracts features of the current and reference frames separately and then matches them, limiting the target information of current feature. Unlike them, we jointly model features and correspondence inside the full vision transformer [12, 14], allowing them to help each other, and minimize its modification to leverage the pre-training of backbone.

2.2. Memory Design

How to design the memory bank and what needs to be stored is critical for memory-based methods. But most works [8,31] choose to only memorize the reference frames with masks to provide fine-grained information in pixellevel. AFB-URR [22] and QDMN [24] dynamically manage them according to the similarity or segmentation quality. However, these works lack a comprehensive understanding of the target, making it difficult to solve the large deformation problem and distinguish between similar objects at the pixel level. XMem [6] introduces multi-store memory and dynamic update its sensory memory with GRU [10], but it can only be updated using top-level feature and the additional temporal module needs to be trained from scratch. HODOR [1] and ISVOS [42] encode object feature into descriptors (or object queries) with a Transformer Decoder. Again, they only interact with the top-level feature in one frame and the extra module brings more computations and parameters, and the latter joins another dataset as supervised signals for the module. Unlike them, our compressed

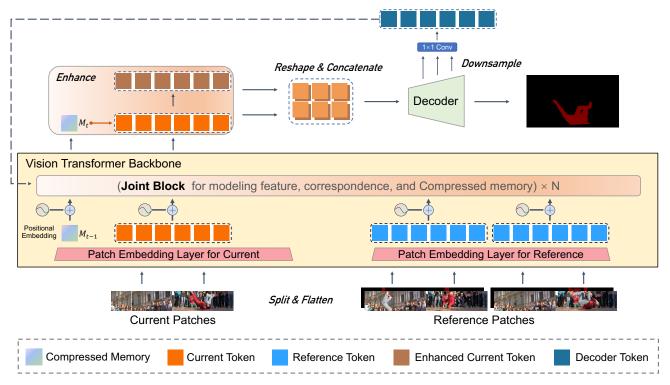


Figure 2. Overview of our **JointFormer**. The current and reference frames with masks are split and flattened into patches, then fed with our compressed memory into the Vision Transformer backbone consisting of **Joint Blocks**. Finally, we enhance the current tokens with compressed memory and sent them into the decoder to predict the object mask.

memory views each target as a whole instance during the entire video. We unify the framework to jointly model our instance-level compressed memory and pixel-level memory inside the backbone and make two memories conceptually justified and compensate for each other. On the one hand, since the updating is inside the backbone, our compressed memory can learn multi-level target features to overcome the object deformation problem. On the other hand, our compressed memory provides long-term features passed between frames through a customized online updating mechanism, which is more suitable for online video tasks.

2.3. Joint Learning and Relation in SOT task

In the single object tracking (SOT) task, the dominant tracking framework follows the two-stage pipeline that extracts the features of the template and search region separately and then performs relation modeling. In order to simplify this pipeline, some works [3,11,53] unify the process of feature extraction and target information integration simultaneously inside the backbone. With the help of the transformer structure and its attention operation, their unified one-stream model structure has achieved excellent performance for single object tracking task. Note that they almost work along the bidirectional information flow, which means they concatenate the template frame and search frame as a whole sequence and perform self-

attention to make them share information in a bilateral way during modeling. DropMAE [44] redesign MAE pre-train on videos to facilitate temporal correspondence learning for SOT task and build a simple VOS baseline following the bidirectional flow with its pre-training. Nevertheless, since the granularity of object labels and the number of targets given in the first frame are different from the SOT task, it is sub-optimal for preserving target details during joint modeling to simply follow the bi-directional flow structure in the VOS task, and we need a holistic understanding of the target to distinguish between similar objects. We redesign the flow and discuss this motivation in detail in Section. 3.4.

3. Method

3.1. Overall Architecture

In this section, we present a neat, unified network **Joint-Former** as illustrated in Fig. 2. For simplicity, we take a single object as an example here. We first concatenate T reference frames with their masks along the RGB channels to get reference pairs, and split and flatten them with the current frame into patches $x_p \in \mathbb{R}^{N_x \times (3 \cdot P^2)}$ and $r_p \in \mathbb{R}^{N_r \times (4 \cdot P^2)}$, where (P, P) is patch size, and N_x and N_r are the length of current and reference patches respectively. Then, we project them into tokens with two patch embedding layers, add a learnable positional embedding,

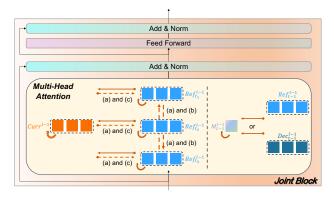


Figure 3. Detailed view of the Joint Block, for joint modeling of features, correspondence, and compressed memory. Specifically, we utilize attention operation to model target information flow (k/v to q) which is indicated by the orange line, and the dotted line indicates that the flow exists only in the corresponding mode.

and feed current tokens $\mathcal{F}_x^0 \in \mathbb{R}^{N_x \times C}$ and reference tokens $\mathcal{F}_r^0 \in \mathbb{R}^{N_r \times C}$ with our compressed memory $\mathcal{F}_M^0 \in \mathbb{R}^{1 \times C}$ into the backbone. The Vision Transformer backbone consists of N-layer **Joint Blocks** (Section 3.2) to propagate the target information, achieving jointly model feature, correspondence, and our compressed memory (Section 3.3) simultaneously inside the backbone. In the case of modeling the feature and correspondence, we propose four propagation modes to analyze the joint modeling and find a propagation way to fit the fine-grained target information. As for modeling the compressed memory, we design a *customized* online updating mechanism to make it provide a long-term and stable object feature during the online process. After the joint modeling, we take the current tokens and compressed memory and utilize the comprehensive feature of compressed memory $\mathcal{F}_M^N \in \mathbb{R}^{1 \times C}$ to enhance the current tokens. Finally, we reshape and concatenate two current token sequences $\mathcal{F}_x^N, \mathcal{F}_{x'}^N \in \mathbb{R}^{N_x \times C}$ and input them into the decoder to predict the target mask. In addition, the decoder also outputs decoder token $\mathcal{F}_d \in \mathbb{R}^{N_r \times C}$ used to update the compressed memory token in the next frame.

3.2. Joint Block

The previous works perform feature extraction and relation matching separately, making it difficult to capture lower-level target representations and handle the large-scale pre-training. Unlike them, the core module of our Joint-Former is **Joint Block**, which joint models feature, correspondence, and our compressed memory. As shown in Fig. 3, to extract features and propagate object information simultaneously, we apply the Multi-Head attention [39] of each block to model the information flow (k/v to q).

As all tokens input to the same block represent the same level of feature, a straightforward thought is to concatenate the current tokens and reference tokens as a whole token sequence and perform self-attention in the sequence, *i.e.*,

the information will be fully shared with all tokens. Nevertheless, it should be noted that the reference tokens are generated from the reference frames and masks, and this simple full-sharing approach introduces too much influence on the reference markers during the modeling process. This may harm the target details contained in the mask, leading to incorrect relational matches for the current tokens.

To further analyze the impact of information propagation, we propose four various modes for propagating target information. Specifically, we flexibly change the sender of the information received by each token, *i.e.*, k/v in attention which can be written as:

$$\begin{aligned} & \text{Attention}(\mathcal{F}) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \\ & K, V_x = W^{K,V}[\mathcal{F}_x, \mathcal{F}_r, \mathcal{F}_{r'}] \\ & K, V_r = \begin{cases} W^{K,V}[\mathcal{F}_r, \mathcal{F}_x, \mathcal{F}_{r'}], & \text{(a)} & \text{(1)} \\ W^{K,V}[\mathcal{F}_r, \mathcal{F}_{r'}], & \text{(b)} \\ W^{K,V}[\mathcal{F}_r, \mathcal{F}_x], & \text{(c)} \\ W^{K,V}\mathcal{F}_r, & \text{(d)} \end{cases}$$

Except that each token receives information from its own, the current token accepts target features from all reference tokens, which are the same in all four modes, but the reference tokens are different. As shown in Eq. 1, in modes (a) and (b), the reference tokens receive target information from other references, but modes (c) and (d) do not. In modes (a) and (c), the reference tokens receive information from current tokens, but modes (b) and (d) do not. In Figure. 2, the orange lines indicate the information flows, while the dotted lines indicate that these information flows exist only in their corresponding mode. We compare the performance of the four modes in Section 4.3, and provide extensive visualizations for a more comprehensive analysis.

3.3. Compressed Memory

Most of the existing memory-based works rely only on reference masks, which are pixel-level memories. However, they ignore the holistic modeling of the target, which resulted in them failing to distinguish similar objects well at the pixel level and not having a long-term understanding during the online segmentation process. To fix this issue and further provide discriminative features for targets, inspired by the "classification token" in ViT [12] that can represent the entire image feature, we present **compressed memory**, just one token for each target to represent it as a whole instance. Specifically, we design a customized online updating mechanism that inputs the compressed memory with the current tokens and reference tokens together to joint model them inside the backbone rather than using an additional temporal module. After updating during joint modeling, we enhance the target-specific information of current tokens at the instance level by injecting the comprehensive feature from the compressed memory into current tokens.

Customized online updating mechanism. Our goal is to provide a long-term and adaptable feature for targets, so we choose to model the compressed memory inside the backbone to utilize the multi-level features, instead of using an additional temporal module that only leverages the top-level features. Moreover, we do not inject the information into the reference features in order to retain their target details as much as possible. In that way, we also model the compressed memory inside Joint Block where it receives information from decoder tokens (we will explain later) or reference tokens from the previous frame except itself. We use the decoder tokens as k/v in the first two blocks and use the reference tokens in all other blocks, like $K, V_M = [\mathcal{F}_M, \mathcal{F}_r \text{ or } \mathcal{F}_d]$. Meanwhile, the reference tokens and decoder tokens perform feature exaction.

Enhanced utilization. After the updating mechanism, the compressed memory contains the comprehensive target feature. We utilize it to enhance current tokens and also employ a Transformer Block but with a small modification. We first adopt current tokens and the compressed memory token as query and key for the attention operation, respectively, to get similarity matrix $A \in \mathbb{R}^{N_r \times 1}$. Since we have only one key for each query, we modify softmax to sigmoid to increase the distance between the responses from compressed memory at different locations of the current feature. Then, we adopt the compressed memory token as value and multiply it with the similarity matrix.

Finally, we reshape and concatenate the current tokens and enhanced current tokens, then input them into the decoder to predict the target mask. Meanwhile, we fuse features at the 1/16, 1/8, and 1/4 scales within the decoder to obtain the decoder tokens, then apply them to update the compressed memory of the next frame, as shown on the top side of Fig. 2. Over a long period of frame-by-frame updating, our compressed memory is modeled with all levels of reference, resulting in a stable and adaptable feature.

3.4. Discussion

The SOT and VOS tasks perform coarse- and fine-grained tracking of targets in the form of boxes and masks, respectively, so both of them emphasize target feature extraction and relationship modeling. However, these two tasks have many differences. i) Since the SOT task only needs to localize the target roughly, the non-target region can be simply filtered by cropping so that the template and search tokens are both from RGB frames. But the reference masks provide more details to segment the target precisely which masks the reference tokens are from RGB frames and masks which are different from current tokens. ii) The VOS task pays more attention to subtle segmentation but neglects the integrity of the target. iii) Unlike the SOT task, which

tracks only one target, most of the videos in the VOS task are multi-target, so it is more important to provide a holistic and discriminating feature for each target to distinguish similar objects. Combining the above reasons, the bidirectional flow structure from SOT works is unsuitable for the VOS task, and we need a holistic understanding of objects.

4. Experiments

4.1. Implementation Details

Networks. We apply ConvMAE-Base¹ [14] as our backbone with its MAE pre-training [15]. The decoder first concatenates the two current features, then iteratively upsamples by $2 \times$ at a time until 1/4 scale while refining with the skip connections from the backbone. Finally, we generate the single-channel logit with a 3×3 convolution and bilinearly upsample it to the input resolution. We copy the last two blocks of the backbone to update compressed memory with decoder tokens and copy the last block to enhance the current tokens. In the multi-object scenario, we apply the soft-aggregation operation [31] to merge the prediction.

Training. Following [6,8,31,35,50], we first pre-train our network on synthetic video sequences generated from static image datasets [5,20,36,43,54]. Then, we perform the main training on DAVIS [34] and YouTube-VOS [47] datasets with curriculum sampling [31]. We also provide results that are pre-trained on BL30K [7] optionally to further improve the performance. The models pre-trained with additional datasets are denoted with * and †. The sample sequence length is set to four, and a maximum of two past frames are randomly selected to be reference frames. For each training sample sequence, we randomly choose at most three objects for training. We use bootstrapped cross entropy loss and dice loss with 0.5:0.5 combination following [50].

For optimization, we use the AdamW [18, 25] optimizer with a learning rate of 5e-5 and a weight decay of 0.05. To avoid over-fitting, the initial learning rate of the backbone is reduced to 0.1 of other network parts. We pre-train our model for 150K iterations with batch size 24 on static image datasets, and the main training lasts 160K iterations with batch size 12. We drop the learning rate by a factor of 10 after the first 100K iterations on main training.

Inference. For DAVIS dataset, we only keep the first and the previous frames. For longer YouTube-VOS dataset, we simply implement a first-in-first-out queue by memorizing every 5th frame following previous work [8, 31] and set the maximum frame size to 3 in addition to the previous frames.

We use videos in 480p and top-k filtering [7] in all blocks by default. Specifically, the current tokens receive information from themselves and all reference tokens with the attention matrix $A \in \mathbb{R}^{N_x \times (N_x + N_r)}$, we keep all from

¹ConvMAE is the replacement of the ViT patch embedding layer with fewer lightweight CNN blocks.

Method	DAVI	S 2017	7 val	DAVI	S 2017	' test	Y	ouTube	e-VOS	2018 v	/al	Yo	ouTube	-VOS	2019 v	/al
Wictiod	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U
STM * [31]	81.8	79.2	84.3	-	-	-	79.4	79.7	84.2	72.8	80.9	-	-	-	-	_
AFB-URR * [22]	76.9	74.4	79.3	-	-	-	79.6	78.8	83.1	74.1	82.6	-	-	-	-	-
CFBI * [49]	81.9	79.1	84.6	74.8	71.1	78.5	81.4	81.1	85.8	75.3	83.4	81.0	80.6	85.1	75.2	83
SST [13]	82.5	79.9	85.1	-	-	-	81.7	81.2	-	76.0	-	81.8	80.9	-	76.6	-
MiVOS *† [7]	84.5	81.7	87.4	78.6	74.9	82.2	82.6	81.1	85.6	77.7	86.2	82.4	80.6	84.7	78.1	86.4
JOINT [27]	83.5	80.8	86.2	-	-	-	83.1	81.5	85.9	78.7	86.5	82.8	80.8	84.8	79.0	86.6
STCN * [†] [8]	85.3	82.0	88.6	77.8	74.3	81.3	84.3	83.2	87.9	79.0	87.3	84.2	82.6	87.0	79.4	87.7
SwinB-AOT-L * [50]	85.4	82.4	88.4	81.2	77.3	85.1	85.1	85.1	90.1	78.4	86.9	85.3	84.6	89.5	79.3	87.7
SwinB-DeAOT-L * [52]	86.2	83.1	89.2	82.8	78.9	86.7	86.3	85.4	90.7	80.1	89.0	<u>86.4</u>	<u>85.4</u>	<u>90.3</u>	80.5	<u>89.3</u>
XMem [6]	84.5	81.4	87.6	79.8	76.3	83.4	84.3	83.9	88.8	77.7	86.7	84.2	83.8	88.3	78.1	86.7
XMem * [6]	86.2	82.9	89.5	81.0	77.4	84.5	85.7	84.6	89.3	80.2	88.7	85.5	84.3	88.6	80.3	88.6
XMem *† [6]	87.7	84.0	91.4	81.2	77.6	84.7	86.1	85.1	89.8	80.3	89.2	85.8	84.8	89.2	80.3	88.8
ISVOS * [42]	87.1	83.7	90.5	82.8	79.3	86.2	86.3	85.5	90.2	80.5	88.8	86.1	85.2	89.7	80.7	88.9
ISVOS *† [42]	88.2	<u>84.5</u>	<u>91.9</u>	84.0	<u>80.1</u>	<u>87.8</u>	86.7	<u>86.1</u>	<u>90.8</u>	<u>81.0</u>	<u>89.0</u>	86.3	85.2	89.7	<u>81.0</u>	89.1
Ours	89.1	85.9	92.2	87.0	83.4	90.6	86.0	86.0	91.0	79.5	87.5	86.2	85.7	90.5	80.4	88.2
Ours *	89.7	86.7	92.7	87.6	84.2	91.1	87.0	86.2	91.0	81.4	89.3	87.0	86.1	90.6	82.0	89.5
Ours *†	90.1	87.0	93.2	88.1	84.7	91.6	87.6	86.4	91.0	82.2	90.7	87.4	86.5	90.9	82.0	90.3

Table 1. Quantitative comparisons on the DAVIS 2017 [34], YouTube-VOS [47] 2018 & 2019 dataset. The * and † denote pre-trained on image datasets and the large BL30K dataset [7], respectively. The <u>underlined</u> and **bolded** results indicate the best of the previous and all works. We re-run AOT [50] and DeAOT [52] on YouTube-VOS dataset for fair comparison which improves their performance.

Method	$\int \mathcal{J} \& \mathcal{F}$	$\mathcal J$	\mathcal{F}
STM * [31]	89.3	88.7	89.9
CFBI * [49]	89.4	88.3	90.5
STCN *† [8]	91.7	90.4	93.0
SwinB-AOT-L * [50]	92.0	90.7	93.3
SwinB-DeAOT-L * [52]	92.9	91.1	94.7
XMem * [6]	91.5	90.4	92.7
XMem * [†] [6]	92.0	90.7	93.2
ISVOS * [42]	92.6	91.5	93.7
ISVOS *† [42]	92.8	91.8	93.8
Ours *	92.1	90.6	93.6
Ours *†	92.4	90.4	94.4

Table 2. Quantitative evaluation on DAVIS 2016 val split.

current tokens and the top-k from reference tokens to get $A' \in \mathbb{R}^{N_x \times (N_x + N_{topK})}$ before *softmax* to perform robust matching while maintaining the details in the current frame.

4.2. Compare with the State-of-the-art Methods

Datasets and evaluation metrics. We report the results on DAVIS 2016/2017 [33, 34] and YouTube-VOS 2018/2019 [47] datasets. The evaluation metrics include region similarity \mathcal{J} , contour accuracy \mathcal{F} , and their average $\mathcal{J}\&\mathcal{F}$, and also report them from the seen and unseen categories for YouTube-VOS dataset. We evaluate all the results on official evaluation servers or with official tools.

DAVIS 2017 [34] is a multiple objects extension of DAVIS 2016, whose validation split has 30 videos with 59 objects, and test split contains 30 more challenging videos with 89 objects. Table 1 shows our network significantly outper-

forms all existing works both on the validation (89.7%) and test-dev (87.6%) split. Remarkably, our model outperforms them by a large margin **without** synthetic pre-training.

YouTube-VOS [47] is the latest large-scale benchmark for multi-object video segmentation. It has 3471 videos in the training split with 65 categories and 474/507 videos in the validation 2018/2019 split which contain 26 unseen categories which do not exist in the training split to evaluate the generalization ability of algorithms. As shown in Table 1, our network achieves superior performance both on YouTube-VOS 2018 (87.0%) and 2019 (87.0%).

DAVIS 2016 [33] is a single object benchmark for VOS. As shown in Table 2, our network achieves competitive performance (92.1%) with the previous state-of-the-art methods. **Qualitative Results.** We compare our qualitatively with XMem [6] and Swin-DeAOT [52] in Fig. 8, all three models are pre-trained with image dataset. It can be found that our model significantly outperforms them both in terms of distinguishing similar targets and boundary details.

4.3. Exploration Studies

We perform ablations experiments on DAVIS 2017 [34] and YouTube-VOS 2019 [47] datasets. By default, we apply ConvMAE with its self-supervised MAE [15] pre-training, choose propagation mode (d), without using the compressed memory, train only on VOS datasets, and apply the top-k filter in all blocks for simplicity and fairness.

Study on various propagation modes. To verify the ability of joint modeling inside the backbone, we first implement a naïve baseline and compare it with our Joint Block

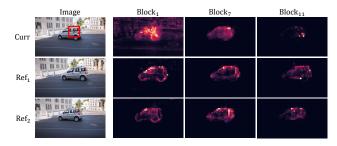


Figure 4. Visualization of attention weights for the current frame, corresponding to itself and all reference frames in mode (d).

	Image	Block ₁	Block ₇	Block ₁₁
Ref ₁		72	(0)	121-
Ref_1		13		·
Ref ₂		13		ie.
Curr	Per Carrier In	- Marie	•	ēr.

Figure 5. Visualization of attention weights for one reference frame. The first row is the mode (d). The last three rows are mode (a), corresponding to itself, another reference, and the current.

in four various propagation modes. Specifically, we extract *key/value* with ConvMAE and ResNet18 following STCN [8], and calculate affinity matrix with dot product instead of L2 similarity, so as to keep consistent with the standard attention of the ConvMAE. As shown in Table 3, 'Refs' and 'Curr' denote whether the reference tokens share information with other references and whether they accept information from the current tokens, respectively.

i) Comparing the naïve baseline with our Joint Block, the results show that joint modeling features and matching in all propagation modes is better than post-matching, which indicates that the decoupling limits the target-specific feature. ii) Comparing (a) to (b) or (c) to (d), the reference tokens only perform feature extraction is better than obtaining information from the current tokens. It demonstrates that the reference tokens can filter out most of the non-target information through the masks, but the current tokens cannot do so, then the feature extraction of the references is corrupted by the non-target information severely from the current tokens before getting sufficient features. iii) Comparing (a) to (c) or (b) to (d), each reference frame performs feature extraction separately without sharing with others is better. It's probably because targets in different reference frames may vary significantly, uniformly performing feature extraction destroys the low-level feature in the early stages.

To get a more intuitive sense of the impact of the propagation, we visualize the attention weights from the first,

Setting	Refs	Current	DAVIS 2017 test			YouTube-VOS 2019 val				
Setting		Current	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U
naïve	l —	_	76.2							
(a)	✓	\checkmark	83.6							
(b)	✓		83.8	80.3	87.3	85.2	84.1	88.5	80.3	87.7
(c)		\checkmark	83.9	80.3	87.5	85.0	84.9	89.3	79.1	86.6
(d)			86.4	82.7	90.0	85.7	85.6	90.2	79.8	87.2

Table 3. Ablation on propagation modes in Joint Block. 'Refs' and 'Current' denote whether the reference tokens accept information from other references and the current tokens, respectively.

	DAVI								
Setting	$\mathcal{J}\&\mathcal{F}$	$\mathcal J$	${\mathcal F}$	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U	
w/o + GRU [6] + compressed	88.1	85.3	90.9	85.7	85.6	90.2	79.8	87.2	
+ GRU [6]	88.1	85.1	91.1	85.9	85.7	90.4	80.0	87.4	
+ compressed	89.1	85.9	92.2	86.2	85.7	90.5	80.4	88.2	

Table 4. Ablation on our compressed memory.



Figure 6. Visualization for attention weights of current tokens corresponding to the compressed memory. The first row is a single car moving fast between frames. The second and third rows are multiple targets highly similar or obscured in the same frame.

seventh, and last block, and keep only the current part inside the red box. We first provide the attention weights of the current tokens corresponding to themselves and all reference tokens for mode (d) in Fig. 4. The results show that our model can accurately match the target location of the reference frames and pay more attention to the boundaries in the early stages, and then gradually focus on details and critical points. In Fig. 5, we visualize the attention weights of one reference frame. The first row is the feature extraction in mode (d) and the last three rows are corresponding to itself, another reference frame, and the current frame in mode (a). The results show that the reference tokens focus on the non-target region of the current a lot in mode (a) which corrupts the feature extraction of the target.

Study on compressed memory. To verify the generalization of compressed memory, we compare it to the GRU sensory memory from XMem [6]. As shown in Table 4, The experimental results show that it works well both on the short- and long-term datasets, indicating it can provide a more robust target feature by leveraging the multi-level feature from reference inside backbone and also distinguish

Block	DAVIS	S 201	7 val	Youtube-VOS 2019 val					
Locations	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U	
first 6 blocks	87.3	84.4	90.1	84.9	84.9	89.3	78.9	86.6	
last 6 blocks	87.1	84.4	89.9	84.6	85.1	89.6	78.1	85.5	
evenly 2 blocks	86.9	83.8	89.9	85.0	85.0	89.6	78.8	86.8	
all blocks	88.1	85.3	90.9	85.7	85.6	90.2	79.8	87.2	

Table 5. Ablation on locations of the Joint Blocks.

Backbone	Pre-train	DAVIS	3 201	7 test	You	Tube-	-VOS	2019	9 val
Dackbone	Methods	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U
	Scratch	67.8	63.9	71.8	72.7	75.0	78.4	64.8	72.7
ViT - 4	DeiT Sup.	77.9	74.0	81.9	81.1	81.5	85.5	75.0	82.4
VII - 4	MAE Ft.	81.0	77.1	85.0	83.0	83.3	87.3	77.1	84.5
	MAE Pre.	81.7	77.7	85.8	84.6	84.6	89.1	78.2	86.5
	Scratch	66.7	63.0	70.5	73.1	75.6	79.0	65.2	72.7
ViT - last	DeiT Sup.	78.4	74.4	82.3	82.1	81.2	85.5	76.7	85.2
VII - last	MAE Ft.	80.5	76.5	84.6	82.7	83.0	87.5	76.4	83.9
	MAE Pre.	82.5	78.5	86.5	85.1	84.5	89.0	79.3	87.4
	Scratch	65.6	61.7	69.4	73.3	75.2	78.5	65.8	73.6
	MAE Ft.	79.7	75.8	83.7	82.5	82.8	87.1	76.0	84.1
	MAE Pre.	86.4	82.7	90.0	85.7	85.6	90.2	79.8	87.2

Table 6. Ablation on pretraining methods. 'ViT-4' and 'ViT-last' represent building multi-scale features with the last of each four blocks and the last block of the whole backbone, respectively.

the target from similar others at the instance level by simply storing one token for each target. We visualize the attention weights of current tokens corresponding to the compressed memory after *sigmoid* in Fig. 6 to further explore how the compressed memory works. It can be noticed that it highly responds at the exact position for both deformed long-term single target and obscured multiple targets. Moreover, the lower response values for the target boundaries indicate that it can also correct the boundaries.

Study on locations of Joint Blocks. In Table 5, we compare the impact from the location of Joint Blocks. We set the first six, last six, evenly two, or all blocks as Joint blocks, otherwise it means that the block only performs feature extraction without interaction. We can find that setting all blocks to Joint Blocks has the best performance, indicating that both low-level and high-level interactions are necessary. In addition, setting the first six blocks as Joint Blocks is better than setting the last six, indicating that interaction at the lower level is important for fine-grained segmentation and discriminative learning, which is missing in previous methods that only match at the top-level.

Study on pre-training methods and multi-scale features.

Since most works leverage supervised pre-training or transformer blocks without pre-training, we further investigate the effect of pre-training methods. Since ConvMAE does not provide supervised pre-training, we leverage ViT with DeiT-3 [37] which supervised pre-training on ImageNet-

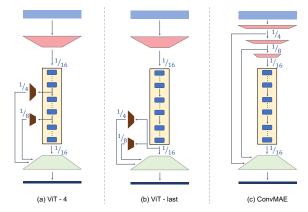


Figure 7. Multi-scale Structure Design of ViT and ConvMAE.

Backhone	Тор-К	DAVIS	S 2017	test 7	YouTube-VOS 2019 val \mathcal{G} \mathcal{J}_S \mathcal{F}_S \mathcal{J}_U \mathcal{F}_U				val
Dackbone	Blocks	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U
	Last	82.0	78.1	86.0	84.3	84.6	89.1	78.0	85.6
ViT - 4	Half	81.7	77.7	85.8	84.6	84.6	89.1	78.2	86.5
	All	81.5	77.6	85.4	84.5	84.1	88.5	78.7	86.7
	Last	82.5	78.6	86.4	85.0	84.3	88.9	79.5	87.2
ViT - last	Half	82.5	78.5	86.5	85.1	84.5	89.0	79.3	87.4
	All	82.4	78.4	86.4	84.5	84.1	88.7	78.7	86.6
	Last	84.4	80.7	88.0	85.3	85.6	90.1	79.3	86.4
ConvMAE	Half	86.1	82.4	89.7	85.7	85.4	90.1	79.8	87.5
	All	86.4	82.7	90.0	85.7	85.6	90.2	79.8	87.2

Table 7. Ablation on top-k settings.

21K and compare it with self-supervised MAE [15] pre-training on ImageNet-1K. We apply the top-k filter in the first half blocks for ViT backbone. As shown in Table 6, pre-training shows necessity on both two backbones. The self-supervised MAE performs better than supervised DeiT with much less data. Furthermore, we also test MAE Ft checkpoint which means the backbone is fine-tuned with classification on ImageNet-1k after MAE pre-trained. The result shows that MAE pre-training is better than fine-tuning, which means it can alleviate the problem that ViT has fewer inductive biases. We expect the MAE pre-training to contribute to more VOS works in the future.

In addition, we compare various strategies for building multi-scale features. As shown in Fig. 7, we follow ViT-Det [21] and implement two constructions with transposed convolutions for ViT. 'ViT-4' means dividing the network into three stages and building the multi-scale features with the outputs from the last block of each stage, and 'ViT-last' means using only the last block of the whole network. For ConvMAE, we use the outputs of patch embedding layers and the last block. The results show that ConvMAE is better than ViT, indicating that the large stride (16) of the patch embedding layer makes ViT loses spatial details in severe and challenging to recover. Building multi-scale features with the top-level features directly is better than with low-level features which is the default setting of CNNs [23]. ViT

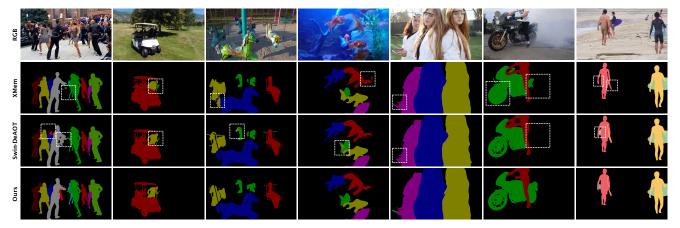


Figure 8. Qualitative comparisons of our model with two state-of-the-art works, XMem [6] and Swin-DeAOT [52]. We mark their failures in the white dashed boxes. Our model outperforms them in terms of detailing and discriminating similarities.

Method	DAVIS	$\frac{\text{DAVIS 2016 val}}{\mathcal{J}\&\mathcal{F} \mathcal{J} \mathcal{F}}$			DAVIS 2017 val			DAVIS 2017 test		
Wicthou	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	$\mathcal{J}\&\mathcal{F}$	\mathcal{J}	\mathcal{F}	
Ours *	92.1	90.6	93.6	89.7	86.7	92.7	87.6	84.2	91.1	
Ours *†	92.4	90.4	94.4	90.1	87.0	93.2	88.1	84.7	91.6	
Ours $^*_{MS}$	92.3	91.2	93.4	90.0	87.2	92.9	88.1	84.8	91.3	
Ours * Ours * Ours $^*_{MS}$ Ours $^*_{MS}$	92.8	91.3	94.4	90.6	87.8	93.5	88.5	85.1	91.8	

Method						YouTube-VOS 2019 val					
Wictiou	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U	\mathcal{G}	\mathcal{J}_S	\mathcal{F}_S	\mathcal{J}_U	\mathcal{F}_U	
Ours *	87.0	86.2	91.0	81.4	89.3	87.0	86.1	90.6	82.0	89.5	
Ours *†	87.6	86.4	91.0	82.2	90.7	87.4	86.5	90.9	82.0	90.3	
Ours $^*_{MS}$	87.2	86.5	91.3	81.8	89.4	87.4	86.4	91.0	82.4	89.8	
Ours *† Ours $^*_{MS}$ Ours $^{*\dagger}_{MS}$	87.6	86.4	91.0	82.3	90.7	87.5	86.5	90.9	82.2	90.4	

Table 8. Multi-scale evaluation on DAVIS and YouTube-VOS.

model only provides single-scale features, thus loses the basis to generate large-scale features from low-level.

Study on the top-k settings. In Table 7, we compare different top-k settings. The results show that ViT gets its best result by applying the top-k filter in the second half of the blocks while ConvMAE requires it in all blocks. It suggests that ViT needs more references to add spatial details due to its more aggressive patching embedding layer, but ConvMAE retains details using the shallow convolutional layers and needs better matching in all blocks.

Multi-scale evaluation. To further improve the performance without modifying the model and training strategy, we employ multi-scale evaluation which is a general trick used in segmentation tasks. Specifically, we adopt scale variation and vertical mirroring and process the inputs at different scales independently, and finally average output probability maps. Evaluation results are shown in Table 8.

Fair comparison on ConvMAE and limitation. We replace the backbone of [6, 50, 52] with ConvMAE-Base. We

Method	D_{17}^{val}	D_{17}^{td}	Y_{19}^{val}
AOT-L [⋄] DeAOT-L [⋄] XMem [⋄]	80.7	74.5	78.5
DeAOT-L [⋄]	84.3	80.9	83.4
XMem [⋄]	86.4	84.2	85.6
Ours	89.1	87.0	86.2

•	Method	\mathcal{G}	Fps	Mem
-	AOT-L [◊]	78.8	4.9	17847 6144 1877
	$\text{DeAOT-L}^{\diamond}$	83.5	7.6	6144
	XMem [⋄]	85.3	25.5	1877
-	Ours - 1	85.8	6.7	4404 5785
	Ours - 3	86.0	3.0	5785

⁽a) Quantitative comparisons.

(b) YouTube $^{2018}_{val}$ comparison.

Table 9. Fair comparison on ConvMAE backbone. The $^{\diamond}$ indicates we replace their backbone and retrain them on VOS datasets. "Ours - N" means we use N reference frames except the previous.

follow their training settings but train them only on VOS datasets with 160K iterations like us without synthetic image pre-training. The comparisons of performance, efficiency, and memory consumption are shown in Table 9, showing that our model outperforms them substantially with the same backbone. Admittedly, the efficiency issue of JointFormer due to multiple matching is our limitation and needs to be improved in the future. We also measure the impact of reducing the number of reference frames, and show that the performance degradation is small, but significantly improves inference speed.

5. Conclusion

We have proposed JointFormer, a unified framework that jointly models feature, correspondence, and a compressed memory inside the backbone. As the core module of our network, Joint Blocks perform iterative joint modeling to propagate the target information and achieve multi-level and complete interaction. Furthermore, we develop a customized online update mechanism for compressed memory to provide a long-term representation. Extensive experiments show that our JointFormer significantly outperforms previous works on most benchmarks. We expect this work will lead to more VOS works adopting the joint paradigm.

References

- [1] Ali Athar, Jonathon Luiten, Alexander Hermans, Deva Ramanan, and Bastian Leibe. Hodor: High-level object descriptors for object re-segmentation in video learned from static images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3022–3031, June 2022. 2
- [2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. Oneshot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), July 2017. 2
- [3] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, Computer Vision ECCV 2022, pages 375–392, Cham, 2022. Springer Nature Switzerland. 3
- [4] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2
- [5] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very highresolution segmentation via global and local refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 5
- [6] Ho Kei Cheng and Alexander G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, Computer Vision – ECCV 2022, pages 640–658, Cham, 2022. Springer Nature Switzerland. 1, 2, 5, 6, 7, 9
- [7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of* the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5559–5568, June 2021. 2, 5, 6
- [8] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11781–11794. Curran Associates, Inc., 2021. 1, 2, 5, 6, 7
- [9] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2
- [10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014. 2

- [11] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13608–13618, June 2022. 3
- [12] 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*, 2020. 2, 4
- [13] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W. Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5912–5921, June 2021. 2, 6
- [14] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. arXiv preprint arXiv:2205.03892, 2022. 2, 5
- [15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022. 2, 5, 6, 8
- [16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), September 2018. 1, 2
- [17] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. *CoRR*, abs/1703.09554, 2017. 2
- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, 2015. 5
- [19] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2
- [20] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for fewshot segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 5
- [21] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, Computer Vision – ECCV 2022, pages 280–296, Cham, 2022. Springer Nature Switzerland. 8
- [22] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3430–3441. Curran Associates, Inc., 2020. 1, 2, 6

- [23] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), July 2017. 8
- [24] Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, Computer Vision ECCV 2022, pages 468–486, Cham, 2022. Springer Nature Switzerland. 2
- [25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5
- [26] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision ACCV 2018*, pages 565–580, Cham, 2019. Springer International Publishing. 2
- [27] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF Interna*tional Conference on Computer Vision (ICCV), pages 9670– 9679, October 2021. 2, 6
- [28] Jianbiao Mei, Mengmeng Wang, Yeneng Lin, Yi Yuan, and Yong Liu. Transvos: Video object segmentation with transformers. arXiv preprint arXiv:2106.00588, 2021. 2
- [29] King Ngi Ngan and Hongliang Li. *Video segmentation and its applications*. Springer Science & Business Media, 2011.
- [30] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2
- [31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 5, 6
- [32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2
- [33] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 6
- [34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2, 5, 6
- [35] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael

- Frahm, editors, *Computer Vision ECCV 2020*, pages 629–645, Cham, 2020. Springer International Publishing. 1, 2, 5
- [36] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2016. 5
- [37] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 516–533, Cham, 2022. Springer Nature Switzerland. 8
- [38] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 1, 2
- [39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 2, 4
- [40] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 1, 2
- [41] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In Gabriel Brostow Tae-Kyun Kim, Stefanos Zafeiriou and Krystian Mikolajczyk, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 116.1– 116.13. BMVA Press, September 2017. 2
- [42] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Chuanxin Tang, Xiyang Dai, Yucheng Zhao, Yujia Xie, Lu Yuan, and Yu-Gang Jiang. Look before you match: Instance understanding matters in video object segmentation. *arXiv* preprint arXiv:2212.06826, 2022. 2, 6
- [43] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5
- [44] Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B. Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14561–14571, June 2023. 3
- [45] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 2
- [46] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory net-

- work for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1286–1295, June 2021. 1, 2
- [47] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv* preprint arXiv:1809.03327, 2018. 2, 5, 6
- [48] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2
- [49] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision ECCV 2020*, pages 332–348, Cham, 2020. Springer International Publishing. 1, 2, 6
- [50] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2491–2502. Curran Associates, Inc., 2021. 2, 5, 6, 9
- [51] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2
- [52] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022. 2, 6, 9
- [53] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision ECCV 2022*, pages 341–357, Cham, 2022. Springer Nature Switzerland. 3
- [54] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 5
- [55] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1