RMem: Restricted Memory Banks Improve Video Object Segmentation

Junbao Zhou* Ziqi Pang* Yu-Xiong Wang
University of Illinois Urbana-Champaign
{junbaoz,ziqip2,yxw}@illinois.edu

Abstract

With recent video object segmentation (VOS) benchmarks evolving to challenging scenarios, we revisit a simple but overlooked strategy: restricting the size of memory banks. This diverges from the prevalent practice of expanding memory banks to accommodate extensive historical information. Our specially designed "memory deciphering" study offers a pivotal insight underpinning such a strategy: expanding memory banks, while seemingly beneficial, actually increases the difficulty for VOS modules to decode relevant features due to the confusion from redundant information. By restricting memory banks to a limited number of essential frames, we achieve a notable improvement in VOS accuracy. This process balances the importance and freshness of frames to maintain an informative memory bank within a bounded capacity. Additionally, restricted memory banks reduce the training-inference discrepancy in memory lengths compared with continuous expansion. This fosters new opportunities in temporal reasoning and enables us to introduce the previously overlooked "temporal positional embedding." Finally, our insights are embodied in "RMem" ("R" for restricted), a simple yet effective VOS modification that excels at challenging VOS scenarios and establishes new state of the art for object state changes (on the VOST dataset) and long videos (on the Long Videos dataset). Our code and demos are available at https://restricted-memory.github.io/.

1. Introduction

The rapid progress of video object segmentation (VOS) algorithms has motivated the creation of more challenging benchmarks, as exemplified by VOST [34] on more *complicated* videos with significant *object state changes* and the Long Videos dataset [25] featuring extremely *long* duration. These benchmarks elevate the spatio-temporal modeling and prompt us to reassess conventional VOS designs: *can learning-based VOS modules effectively decipher historical information in such challenging scenarios*?

To delve into this issue, it is essential to focus on *memory banks*, which are central to storing past features and feeding input to VOS modules, and are fundamental in memory-based VOS framework [9, 11, 45]. Typically, the memory banks are managed via the simple intuition of *expansion*, continuously *appending* newly sampled frames as the video progresses. While this approach is intended to encompass all historical information, thereby enhancing VOS, we realize its potential limitation: as videos become longer or more complex, these expanding memory banks may overwhelm the capability of VOS modules to discern reliable features.

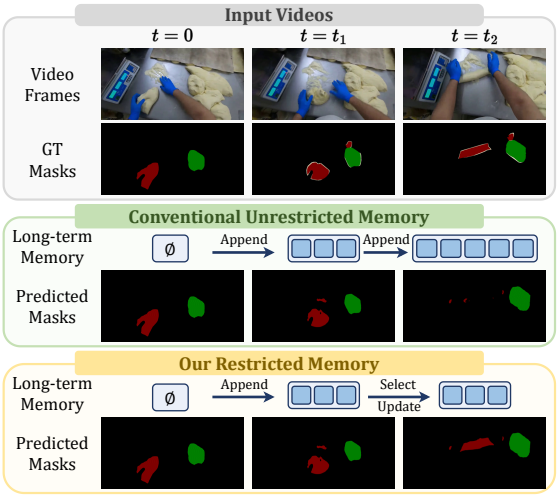

Figure 1. In light of challenging object state changes [34, 41, 47], we rethink the conventional approach of continuously accumulating the features into memory banks: despite capturing all the information, it complicates the deciphering of relevant features. Conversely, restricting the memory significantly enhances VOS.

We investigate this hypothesis by conducting a pilot study, named "*memory deciphering*," to quantify the decoding capability of VOS modules. In our analysis, we continue to use object segmentation as the proxy to VOS, but shift the prediction target to decoding *the object mask at the initial frame (frame 0)* from the memory bank. This choice is deliberate based on the principle of controlling variables: (1) In the VOS framework, the information of frame 0 is implicitly propagated to subsequent frames, ensuring the presence of relevant information for decoding; (2) This prediction target is consistent across frames and allows for a fair comparison of decoding efficacy under varying memory sizes. Intuitively, the later frames have *rigor-*

*Equal contribution.

ously richer information than the earlier frames because of a larger memory bank, and are thus expected to produce better decoding results. However, our observation shows the opposite: *the effectiveness of VOS modules in deciphering information diminishes with increasingly large memory banks*. Intriguingly, this degradation can be mitigated by selecting a small number of relevant frames in the memory bank, and we observe a significantly better concentration of attention scores on relevant frames and regions. Therefore, our systematic study reveals a pivotal insight: *the expansion of memory banks complicates the deciphering of VOS modules* primarily due to *redundant information*.

Inspired by such an insight, we validate its practical significance through a simple approach: *restricting memory banks to a fixed number of frames*. Our concise memory bank facilitates better spatio-temporal modeling and adaptation to object transformation according to the analysis of complex object state changes [34], as illustrated in Fig. 1. The effectiveness of our method stems from a curated memory concisely focusing the attention of VOS modules on relevant information. Based on this, we delve into the updating process when new features arrive. Our strategy balances the relevance and freshness of frame features, drawing inspiration from the upper confidence bound (UCB) algorithm [3] from multi-arm bandit problems.

In addition to enhancing the accuracy, restricted memory banks also reduce *discrepancies in memory lengths* between training and inference when compared with conventional methods. Typically, VOS modules are trained on short clips with a few memory frames, so our restricted memory better aligns with this setup, even when handling significantly longer videos during inference. This alignment opens up opportunities to revisit techniques relying on temporal synchronization between training and inference. As a compelling example, we introduce *temporal positional embedding* to explicitly capture the ordering of memory features – a critical aspect often overlooked by previous methods – leading to superior temporal reasoning.

In conclusion, we make the following contributions:

1. We introduce the novel *memory deciphering* analysis to systematically reveal the drawbacks of expanding memory banks for VOS modules in decoding information.
2. Our revisit of *restricting memory banks* notably enhances VOS accuracy for challenging cases, cooperated with a memory update strategy balancing the relevance and freshness of frames.
3. Benefiting from smaller training-inference gaps, we introduce the previously overlooked *temporal positional embedding* to capture the order of memory frames.

Collectively, our insights lead to a simple yet strong VOS method: "RMem." Our extensive experiments show its strengths and establish new state of the art on VOST [34] for object state changes and the Long Videos dataset [25].

2. Related Work

VOS benchmarks. VOS has evolved through several benchmarks. DAVIS [30, 31] is the first exhibiting diversity and quality, surpassing early benchmarks [5, 23, 35]. YoutubeVOS [40] further scales up by collecting more videos. Although they have enabled great progress in VOS, their limited difficulty and video lengths have spurred more challenging datasets. For example, the average duration in LVOS [20] is more than 500 frames and the Long Videos dataset [25] further extends it to over 1,000 frames, and MOSE [15] increases the difficulty by selecting videos with crowds and occlusions. To evaluate our insight on the most demanding scenarios, we highlight *object state changes* involving noticeable transformations in the existence, appearance, and shapes. Studies on state changes, *e.g.*, VSCOS [47], mostly utilize ego-centric datasets [13, 14, 18]. In this paper, we primarily select the recent VOST [34]. It combines multiple datasets and provides accurate annotations. Notably, VOST shows greater complexity and longer duration than previous YoutubeVOS and DAVIS. We mainly concentrate on the challenging benchmarks.

Memory-based VOS. Memory banks are fundamental for VOS. Earlier approaches [4, 6, 26, 32, 37] treat VOS as online learning and finetune networks with memorized features. Some others [7, 21, 38, 42, 44, 46] approach VOS as template matching but struggle with occluded or dynamically changing objects. Consequently, recent methods mostly focus on memory reading via either pixel-level or object-level attention [36]. Object-level memory reading [1, 2, 12], inspired by Mask2Former [8], excels at efficiency. However, it is less effective for delicate masks or complex scenarios, *e.g.*, VOST [34], where the objects are frequently small or cluttered. In comparison, pixel-wise memory reading [9, 11, 17, 25, 28, 33, 39, 43, 45] is more adopted for its reliable segmentation and it typically associates the current frame to memory features with attention. Our work differs from previous studies by focusing more into the general insights of *drawbacks of expanding memory banks* and plug-and-play strategies to mitigate such issues, instead of dedicated memory reading architectures.

Restricted memory banks in VOS. Previous studies approach restricting memory banks mostly from the efficiency aspect [9, 24, 25]. A notable representative, XMem [9], adopts a hierarchical architecture with customized modifications like similarity computation and memory potentiation. In contrast to these prior efforts, our work stands out by explicitly revealing and highlighting the *accuracy* benefits of restricted memory banks through reducing redundant information, rather than emphasizing *efficiency*. Moreover, our RMem demonstrates such an insight with a *simple plug-and-play* enhancement to the VOS framework, avoiding any noticeable increase or reliance on special operators as in XMem.

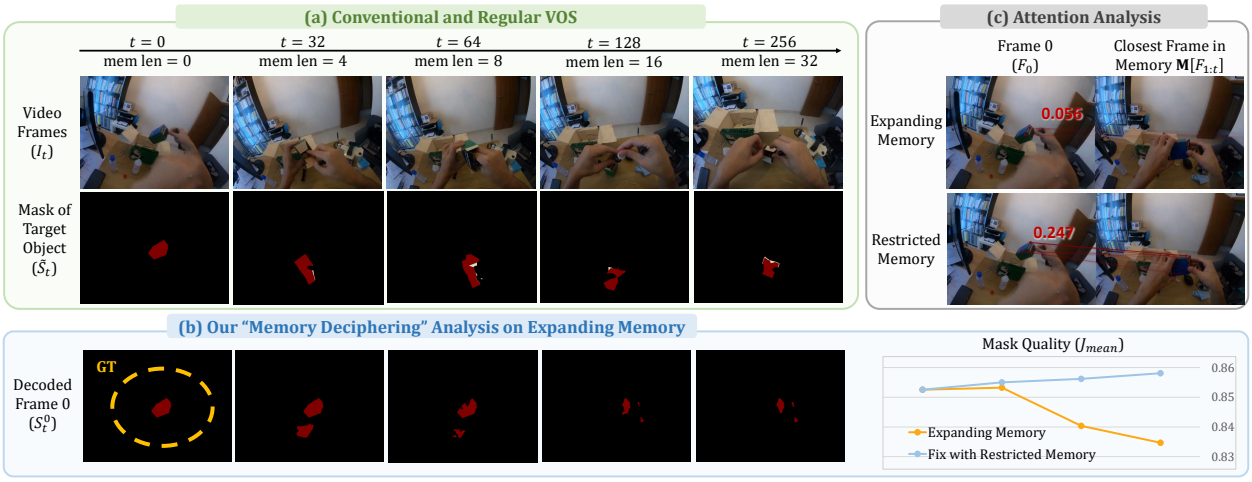

Figure 2. Sketch of Pilot Study. Our *memory deciphering* analysis emulates *decoding the mask on frame 0 from the memory bank features* to quantify the impact of growing memory on VOS modules, where the "desired results" in the figure is the ground truth. For a video shown in Block **(a)**, we visualize its decoding results in Block **(b)**: the masks degrade both quantitatively (yellow curve) and qualitatively, deviating from the desired results. However, selecting a set of concise frames mitigates this issue (blue curve in Block **(b)**). Therefore, we conjecture the drawback of growing memory lies in confusing the attention of VOS modules. In Block **(c)**, we use red lines to indicate highly weighted associations in attention, with thickness denoting the attention score values. As illustrated, the query F_0 focuses less on its most relevant frame after the memory bank expands, with the attention score dropping from 0.247 to 0.056. (2^{nd} row shows ground-truth masks $\widetilde{S}_t$ as context. $\mathcal{J}_{mean}$ is the average Jaccard between S_0^t and $\widetilde{S}_0$ over all videos.)

3. Pilot Study: Memory Deciphering Analysis

This section devises our pilot experiments on how an expanding memory bank influences the decoding capability of VOS modules. Our design emulates the task of VOS but makes several modifications guided by the principle of controlling variables: aligned prediction targets and VOS modules across our pilot experiments, while only the frames in the memory bank vary. Such a comparison enables a clean analysis and reveals the core insight: VOS modules have limited capability to decode a growing memory bank.

Notation and Formulation of VOS. We consider the existing VOS framework as a memory-based encoder-decoder network: the encoder $\mathbf{E}(\cdot)$ is a visual backbone encoding the image I_t at frame t into the feature F_t; and then, the decoder $\mathbf{D}(\cdot)$ converts F_t into the segmentation S_t via reading the features stored in the memory $\mathbf{M}[F_{0:t-1}]$, as below,

$$F_t = \mathbf{E}(I_t), \quad S_t = \mathbf{D}(F_t, \mathbf{M}[F_{0:t-1}]). \tag{1}$$

Here, $\mathbf{M}[F_{0:t-1}]$ generally comes from saving the features at a certain frequency [11, 24, 45], and the VOS decoder is usually special transformers [36], *e.g.*, LSTT in AOT [45]. The final objective of VOS is to minimize the difference between the predicted mask S_t and ground truth $\widetilde{S}_t$.

Design of Our Memory Deciphering Analysis. Our pilot study separates the variables of VOS module $\mathbf{D}(\cdot)$ and the prediction target $\widetilde{S}_t$ to clearly analyze the influence of the memory bank $\mathbf{M}[F_{0:t-1}]$ under a controlling variable setting. Therefore, we purposefully design our memory deciphering analysis as *decoding the mask of the initial frame (frame 0) from the features stored in the memory bank*.

More precisely, our pilot study is formulated as Eqn. 2,

$$S_t^0 = \mathbf{D}'(F_0, \mathbf{M}[F_{1:t}]), \tag{2}$$

where $\mathbf{D}'(\cdot)$ is an additional VOS decoder trained for the objective in Eqn. 2. In practice, we let the original VOS decoder $\mathbf{D}(\cdot)$ to conduct regular VOS as Eqn. 1, then use $\mathbf{D}'(\cdot)$ only for deciphering the mask S_0^t for frame 0, to avoid influencing the original VOS. $\mathbf{M}[F_{1:t}]$ contains the stored feature between frame 1 to t. Please note that the feature of frame 0 are excluded from the input $\mathbf{M}[F_{1:t}]$ to avoid $\mathbf{D}'$ from trivially relying on single-frame memory.

Before delving into the experiment, we clarify and emphasize our reasons for choosing the formulation as Eqn. 2. (1) *Presence of relevant information*. The procedure in Eqn. 1 resembles propagating the masks from historical frames to the current frame t, indicating that $\mathbf{M}[F_{1:t}]$ already contains the information about the mask at frame 0. Therefore, decoding the mask on frame 0 from $\mathbf{M}[F_{1:t}]$ is not a random guess, but should have high-quality results. (2) *Identical prediction target*. Our prediction target remains identical for every frame and varying memory size. (3) *Cooperating with regular VOS*. We utilize $\mathbf{D}'(\cdot)$ as a stand-alone VOS decoder for the analytical deciphering so that the original VOS process remains unchanged and our pilot study can utilize the same memory bank.

Implementation. We select the recent VOST [34] to highlight challenging *object state changes*. Its long video duration and complex scenarios push the limits of VOS decoders in deciphering memory. Then we adopt AOT [45] as VOS encoder-decoder, a popular baseline and the top method on

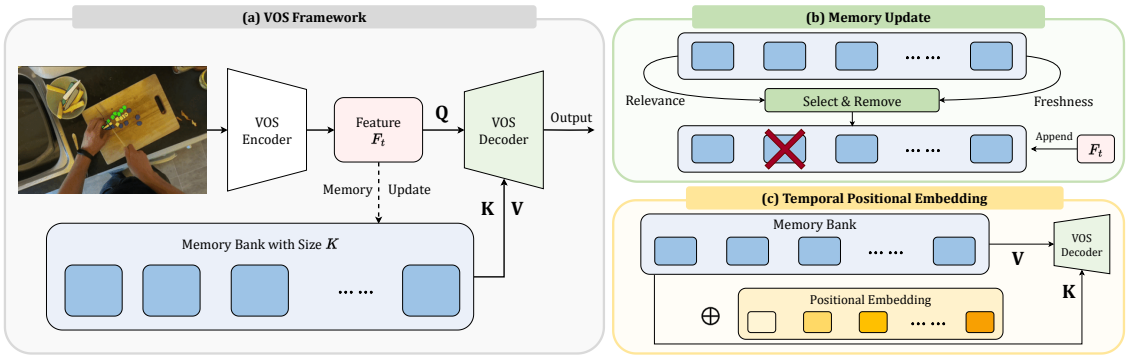

Figure 3. RMem Overview. **(a)** RMem revisits *restricting memory banks* to enhance VOS (Sec. 4.1), motivated by the insight from our pilot study. **(b)** To maintain an informative memory bank, we balance both the relevance and freshness of frames when updating the latest features (Sec. 4.2). **(c)** Benefiting from smaller memory size gaps between training and inference, we introduce previously overlooked temporal positional embedding to encode the orders of frames explicitly (Sec. 4.3), which enhances spatio-temporal reasoning.

VOST. Emulating Eqn. 2, we initialize $\mathbf{D}'(\cdot)$ from AOT's pretrained decoder $\mathbf{D}(\cdot)$, and then supervise S_0^t with a segmentation loss between the ground truth $\widetilde{S}_0$. More implementation details are in the supplementary.

Hypothesis and Expectations. With an expanding memory bank, the information in $\mathbf{M}[F_{1:t}]$ only becomes *rigorously richer* at later frames while the prediction target is unchanged. Therefore, we naturally expect the decoded mask S_0^t to illustrate stable or better accuracy at later frames, assuming that the VOS decoder $\mathbf{D}(\cdot)$ is capable of extracting the relevant features from an increasingly large $\mathbf{M}[F_{1:t}]$.

Results and Analysis. Contrasting the expectation above, we observe that masks S_0^t degrade with a growing memory bank, as shown in Fig. 2 (b). To verify that the growing memory bank is indeed the cause of degradation, we empirically bound the memory bank to 8 frames containing the most relevant and latest information, intuitively: first 7 frames and the latest frame in $\mathbf{M}[F_{1:t}]$. According to the blue curve in Fig. 2 (b), *restricting the memory only to store concise features* effectively avoids degradation.

Inspired by addressing the degradation issue, we propose that the *redundant information* is the main negative impact of an expanding memory bank. Otherwise, the degradation should not disappear simply after we select a subset of intuitively relevant frames. More specifically, this closely relates to how VOS methods utilize attention mechanisms to read from memory banks, where the redundant features decrease the attention scores on relevant frames. As direct evidence, we analyze the attention scores for decoding S_0^t in Fig. 2 (c) and observe that the attention scores between F_0 and its most relevant memory feature (first frame in $\mathbf{M}[F_{1:t}]$) have worse concentration on the correct object and become scattered in a longer memory. Therefore, we conclude that restricting the memory banks with a concise set of relevant features potentially benefits the decoding of VOS modules via more precise attention.

4. Method of RMem

Motivated by our insight from the pilot study, we propose a straightforward approach highlighting a concise memory bank: restricting the memory with a constant frame number (Sec. 4.1). We then explore the strategies to update the memory bank to constantly digest incoming features and remove obsolete frames (Sec. 4.2). Finally, the restricted memory bank decrease the gaps between the memory lengths across the training and inference stages. This enables previous overlook techniques, and we propose a compelling example of temporal positional embedding (Sec. 4.3). The overview of our method "RMem" is in Fig. 3.

4.1. Restricting Memory Banks for VOS

Design. As indicated in our pilot study (Sec. 3), VOS modules have limited capability to process large quantities of features and benefit from a concise memory bank with less redundant information. To verify this in actual VOS systems, we develop the simple approach of *restricting the memory bank to a fixed frame number*. In practice, a predefined small constant number K is the maximum number of frames a memory bank can store, as shown in Fig. 3. The simplicity of our approach makes it a *plug-and-play* enhancement for the existing VOS framework.

On an arbitrary frame t, we simplify the notation of memory bank by denoting $\mathbf{M}[F_{0:t-1}]$ as $\mathbf{M}^t$, containing $K_t \leq K$ frames. A natural issue of bounded memory $\mathbf{M}^t$ is that K_t can reach the limit K at sufficiently large t, making the digestion of newly arriving features non-trivial, especially when the quality of information is vital for VOS, according to how we address degradation in the pilot study (Sec. 3). Our baseline adopts an intuitively simple yet effective approach (we explore better strategies in Sec. 4.2): selecting the most reliable frame (frame 0) and temporally most relevant frames (closest frames). Formally, updating the memory bank is as Eqn. 3 when $K_t = K$,

$$\mathbf{M}^{t+1} = \text{Concat}(\mathbf{M}_0^t,\ \mathbf{M}_{2:K_t-1}^t,\ F_t), \tag{3}$$

where $\mathbf{M}^t_{2:K_t-1}$ and F_t are the closest frames, and $\mathbf{M}^t_1$ is removed to create an available slot, just like Fig. 3 (b).

Discussion. Our restricted memory is a revisit to previous methods [24, 25]. However, we are distinct in emphasizing *accuracy* instead of *efficiency*. In addition, our RMem also simplifies them [9, 24, 25] by treating each frame as a constituent feature map instead of breaking it into smaller regions or pixels [9]; thus, our strategy can directly apply to a wider range of models. Although more sophisticated strategies might further improve our accuracy, a simple approach is already effective (Sec. 5.3).

4.2. Memory Update

Updating the incoming frames to the memory bank provides informative cues for VOS modules to decode. Although our baseline strategy (Eqn. 3) has already cooperated with the bounded memory bank, we investigate its deeper principles.

Challenges of Memory Update. As proved in our pilot study (Sec. 3), the conciseness of information heavily influences the decoding efficacy of VOS modules. Therefore, naive heuristics of random selection or keeping the latest frames are unreliable (as in Sec. 5.4, memory update analysis) since they fail to consider the relevance of frames (random) or suffer from drifting of knowledge (latest). Therefore, we propose the principles that consider both *relevant* prototypical features and *fresh* incoming information.

Memory Update Inspired by Multi-arm Bandits. Our memory update problem can be stated as *how to select and delete the most obsolete frame k_d from K candidates* to make space for incoming features. Although not exactly identical, this problem analogizes *multi-arm bandit*, which also concerns optimizing the reward by selecting from a fixed number of candidates. Its most inspiring insight for us is balancing the exploitation and exploration with the upper confidence bound (UCB) algorithm [3], whose maximization objective O_j for an option j is as Eqn. 4,

$$O_j = R_j + \sqrt{(2\log T)/t_j}, \tag{4}$$

where R_j is option j's average reward, T is the total timestamps, and t_j is the number of timestamps selecting j. When migrating to our VOS, we re-define R_j as the relevance of a frame for reliable VOS and consider $\sqrt{(2\log T)/t_j}$ as the freshness of memory, intuitively. Then, the deleted frame k_d is chosen according to the smallest $O_{1:K}$. In practice, we define the relevance term R_k using the attention scores between frame $\mathbf{M}^t_k$ and current VOS target F_t, to quantify the contribution of features from the memory. Under the context of transformers, we assume decoding the memory bank is as Eqn. 5,

$$F^D_t = \text{Attn}(\text{Q} = F_t, \text{K} = \mathbf{M}^t, \text{V} = \mathbf{M}^t), \tag{5}$$

and assume that $\mathbf{S}^t$ is the scores (after softmax) between F_t and $\mathbf{M}^t$, computed inside the attention. Then, we treat the sum of scores as the relevance of a frame in the memory: $R_k = \text{sum}(S^t_k)$, where S^t_k is the slice of attention scores corresponding to $\mathbf{M}^t_k$. Compared to XMem [9], which also uses attention scores for selection, our design differs in selecting at the frame level instead of the pixel level, which is simpler and already effective (as in Sec. 5.4).

As for the second term in UCB, $\sqrt{(2\log T)/t_j}$, we modify it by defining t_j as the times a frame has stayed in the memory bank and T as the sum of all the frames' staying time. This freshness term penalizes long-staying frames and allows refreshing from the latest information. Finally, O_k combines it with the relevance term R_k via a weight α balancing their numerical scales.

4.3. Memory with Temporal Awareness

Motivation. In addition to accommodating the decoding capability of VOS modules, restricting the memory bank systematically decreases the training-inference discrepancies in memory lengths. Specifically, the VOS algorithms are generally trained on short video clips with a few frames in the memory, while the videos are much longer during inference time. Therefore, the number of frames in the memory bank diverges more significantly without our restriction.

Such temporal alignment between training and inference opens new opportunities for VOS. As a compelling example, we introduce temporal positional embedding (PE) to enhance spatio-temporal reasoning. Specifically, we notice that previous approaches [9, 11, 45] overlook the order of frames in the memory, *i.e.*, the temporal relationship among the frames are not explicitly considered, while spatial PE is widely adopted. Considering the vital role of orders in temporal modeling, which is commonly addressed with temporal PE in video-based tasks, we conjecture that the distinction of memory sizes between training and inference hinders previous methods from employing temporal PE.

Design. The objective of temporal PE is to embed explicit temporal awareness into memory and guide the attention in Eqn. 5. Although restriction on the memory bank alleviates the training-inference shift, the challenges of temporal PE still exist: the optimal memory size K, though much smaller than expanding, can still be larger than the training-time memory size K_{train}; (2) the frames in the memory are varying from 1 to K. To address them, our solution is inspired by how ViT [16] uses learnable PE and interpolation to address different image resolutions. Similarly, we initialize the PE according to K_{train}, denoted as $\widetilde{P}_{0:K_{\text{train}}-1}$, and the query F_t having a dedicated PE P_q. Then, the temporal PE for the memory bank $\mathbf{M}^t_{0:K_t-1}$ is $P^t_{0:K_t-1}$.

$$P^t_{0:K_t-1} = \begin{cases} \widetilde{P}_{0:K_t-1}, & K_t \le K_{\text{train}} \\ \text{Interp}(\widetilde{P}_{0:K_{\text{train}}-1}, K_t), & K_t > K_{\text{train}} \end{cases} \tag{6}$$

where "Interp($\cdot$)" interpolates $\widetilde{P}_{0:K_{\text{train}}-1}$ to K_t via nearest neighbor. Finally, temporal PE enhances the original attention in Eqn. 5 by augmenting the key and values, identical

to our conceptual illustration in Fig. 3 (c):

$$F_t^D = \text{Attn}(\text{Q} = F_t + P_q, \quad \text{K} = \text{M}^t_{0:K_t-1} + P^t_{0:K_t-1}, \quad \text{V} = \text{M}^t_{0:K_t-1}). \tag{7}$$

The above design contains two critical choices. (1) We use the relative index $\{k = 0, ..., K_t-2\}$ inside the memory instead of the frame index t to avoid the shift between training and inference. (2) Using learnable PE instead of Fourier features fits better to a limited training length, K_{train}.

5. Experiments

5.1. Datasets and Evaluation Metrics

VOST. We primarily utilize the recent VOST [34] that concentrates on challenging object state changes. It curates over 700 videos covering diverse object state changes, *e.g.*, changing appearance, occlusions, crowded objects, and fast motions. In VOST, the evaluation metrics are $\mathcal{J}$ and $\mathcal{J}_{tr}$, resembling the average Jaccard over *all the frames* and the harder *last 25% frames* corresponding to state changes.

Long Videos Dataset. We use the Long Videos dataset [25] to evaluate long-term understanding, similar to XMem [9]. It contains 3 validation videos with more than 1k frames. Both $\mathcal{J}$ and $\mathcal{F}$ are considered for evaluation.

LVOS. We also experiment with the recent LVOS [20] and include the results in the supplementary materials.

Regular and Short Datasets. YoutubeVOS [40] and DAVIS [30, 31] are two earlier VOS datasets with *shorter* duration and *easier* scenarios compared to VOST. In this paper, we use them as the pretraining datasets for VOST and the Long Videos dataset following standard practice [9, 34], and conduct analysis in addition to the challenging datasets.

5.2. Baseline and Implementation Details

Our proposed RMem is a simple and plug-and-play enhancement for the VOS framework. Without loss of generality, we select AOT [45] as the main baseline because of its top performance on VOST (as in Table 1) and simplicity. It adopts ResNet-50 [19] as its encoder and a specially designed "long short term-transformer" (LSTT) as its decoder.For the memory bank, the original AOT digests the latest frame and expands the memory continuously, while RMem restricts its size to 8 frames. We also employ RMem on other VOS methods besides AOT. More details on models and implementation in the supplementary.

5.3. State-of-the-art Comparison

VOST. In Table 1, we compare RMem with previous methods on VOST. Our approach establishes new state-of-the-art on this challenging benchmark with significant improvement. Notably, our simple strategy increases the VOS quality for the whole video ($\mathcal{J}$) and maintains better robustness for the state-changing frames ($\mathcal{J}_{tr}$). This is especially clear when compared to AOT [45], which is both the previous top-performing method and our baseline: the improvement is over ∼3% with our plug-and-play modifications.

	$\mathcal{J}_{tr}$	$\mathcal{J}$
OSMN Match [42]	7.0	8.7
OSMN Tune [42]	17.6	23.0
CRW [22]	13.9	23.7
CFBI [44]	32.0	45.0
CFBI+ [46]	32.6	46.0
XMem [9]	33.8	44.1
HODOR Img [1]	13.9	24.2
HODOR Vid [1]	25.4	37.1
AOT [45]	36.4	48.7
AOT$^\Psi$	37.0	49.2
AOT$^\Psi$ + RMem (Ours)	39.8	50.5
DeAOT$^\Psi$	37.6	50.1
DeAOT$^\Psi$ + RMem (Ours)	**40.4**	**51.8**

Table 1. Comparison with previous methods on VOST [34]. Our RMem shows advantages on both overall quality ($\mathcal{J}$) and addressing object state changes ($\mathcal{J}_{tr}$). (Without mention, the results are from VOST's implementation. Ψ denotes our implementation.)

	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CFBI [44]	53.5	50.9	56.1
CFBI+ [46]	50.9	47.9	53.8
STM [29]	80.6	79.9	81.3
MiVOS [10]	81.1	80.2	82.0
AFB-URR [25]	83.7	82.9	84.5
STCN [11]	87.3	85.4	89.2
XMem [9]	89.8	88.0	91.6
AOT [45]	84.3	83.2	85.4
AOT$^\Psi$	86.7	85.5	87.9
AOT$^\Psi$ + RMem (Ours)	90.3	88.5	92.1
DeAOT$^\Psi$	89.4	87.4	91.4
DeAOT$^\Psi$ + RMem (Ours)	**91.5**	**89.8**	**93.3**

Table 2. Comparison with previous methods on Long Videos dataset [25]. For both baselines of AOT and DeAOT, our RMem shows significant improvement. (Without mention, the results are from XMem [9], Ψ denotes our implementation.)

Long Videos Dataset. As our RMem limits memory capacity, a natural suspicion is that our memory bank performs worse in storing information and struggles with long-term modeling. However, our comparison in Table 2 shows the opposite. On the Long Videos dataset, our RMem not only improves upon the baseline AOT and DeAOT model but also performs comparably with the state-of-the-art XMem [9] model, which utilizes specially designed hierarchical memory banks and memory manipulation operators. Therefore, this further supports our insight on keeping a concise memory bank to accommodate the limited capability of VOS modules to address expanding memory banks.

5.4. Ablation Studies

Effect of RMem Components. We analyze each RMem component with the baseline of AOT and DeAOT, as in Table 3. **(1)** *Restricting memory banks.* The most important insight from our pilot study (Sec. 3) is to maintain a concise memory bank with relevant information, which motivates our revisit of restricting memory banks (Sec. 4.1). Accord-

Index	RM	TPE	MU	AOT $\mathcal{J}_{tr}$	AOT $\mathcal{J}$	DeAOT $\mathcal{J}_{tr}$	DeAOT $\mathcal{J}$
1	Baseline			37.0	49.2	37.6	50.9
2	✓			38.6	50.2	38.8	51.0
3	✓	✓		39.7	50.3	40.0	51.7
4	✓		✓	39.4	50.3	39.0	51.4
5	✓	✓	✓	**39.8**	**50.5**	**40.4**	**51.8**

Table 3. Ablation studies of RMem components on VOST. Starting from the AOT and DeAOT baseline, all of the components improve the performance, especially the harder object state-changing frames ($\mathcal{J}_{tr}$). **RM**: restricting memory banks. **TPE**: temporal positional embedding. **MU**: memory update with UCB algorithm.

Method	Variants	$\mathcal{J}_{tr}$	$\mathcal{J}$
Remove	0-th	35.9	48.9
	1-st	38.6	50.2
	Middle	38.3	50.2
	Latest	35.7	48.5
	Random	38.0	50.0
UCB	Relev	39.1	50.1
	Relev + Fresh	**39.4**	**50.3**

Table 4. Ablation study of different memory updating strategies on VOST. We analyze deleting a frame in the memory based on heuristics ("Remove") or guided by the relevance and freshness of the UCB algorithm ("UCB"). Our final memory updating strategy using both relevance and freshness achieves the best performance.

ing to Table 3 (row 1 and 2), a bounded memory bank leads to significant enhancement in the long and complex VOST videos. **(2)** *Temporal positional embedding.* In Table 3, we illustrate that adding positional embedding (Sec. 4.3) greatly benefits the spatio-temporal modeling, especially the harder $\mathcal{J}_{tr}$ for state changes. **(3)** *Memory update.* We refresh the memory banks by balancing the relevance and freshness of frames (Sec. 4.2), inspired by the UCB algorithm [3]. In row 4 and row 5 of Table 3, such a strategy effectively boosts the overall performance.

Analysis on Frame Numbers of Memory Banks. We verify a direct implication of our insight: an expanding memory bank elevates the difficulty of VOS modules to decode information. Specifically, we observe the VOS accuracy under various limits of memory banks. To avoid the influence of hyper-parameter tuning, we utilize the baseline memory update strategy in Sec. 4.1. As in Fig. 4, the performance first improves from richer information. Then both $\mathcal{J}$ and $\mathcal{J}_{tr}$ decrease when the length of memory exceeds the capability of learned AOT modules, until becoming similar to unrestricted memory. Consequently, these results directly support our insight of restricting memory banks.

Memory Update Analysis. Maintaining an informative memory bank is critical for VOS accuracy, and we propose a UCB-inspired algorithm in Sec. 4.2. Table 4 analyzes the key intuition and design choices with AOT. **(1)** The initial frame is critical in keeping the provided ground-truth information: removing the 0-th frame leads to an accuracy drop, and is more profound when scenarios are complex (VOST). **(2)** Freshness of information is critical, as in the worse accuracy of removing the latest frame. **(3)** Randomly removing frames performs surprisingly well but is still worse than our baseline (removing 1-st frame, in Sec. 4.1). **(4)** Using attention scores to reflect the relevance better removes redundant features ("Relev"), and it is further enhanced with the freshness term, where freshness is especially effective to avoid long-staying frames for the Long Video dataset. Finally, the best strategy is our UCB-inspired strategy combining relevance and freshness.

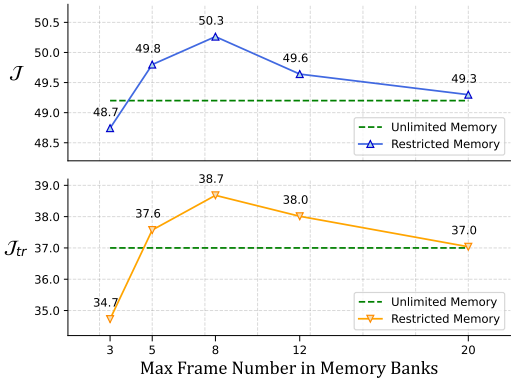

Figure 4. Impact of memory bank size on VOS, tested on VOST. With more frames in the restricted memory, the accuracy first increases and then decreases until it approximates unrestricted memory. This supports the limited deciphering capability of VOS modules and our insight into restricting memory banks.

Method	$\mathcal{J}_{tr}$	$\mathcal{J}$
AOT	37.0	49.2
AOT + RM	38.6	50.2
AOT + SinCos PE	37.2	48.3
AOT + Learnable PE	36.7	49.4
AOT + RM + SinCos PE	37.9	48.9
AOT + RM + Learnable PE	**39.7**	**50.3**

Table 5. Comparison of temporal PE strategies on VOST. Based on restricted memory ("RM"), our learnable temporal PE ("Learnable") is better than using high-frequency Fourier features ("SinCos"). Notably, restricting memory is essential for PE.

Temporal Positional Embedding Strategies. We introduce using learnable temporal PE to address the varied frames in the memory banks of VOS in Sec. 4.3. In Table 5, we analyze another PE strategy of encoding the index into high-frequency features with SinCos functions and find it performs worse. This is because SinCos is commonly used in scenarios of a large number or continuous space of coordinates (*e.g.*, NeRF [27]), while learnable embeddings can better handle a small number of slots (*e.g.*, ViT [16]), as in the limited memory length during the VOS training. Furthermore, we highlight that temporal PE requires restricted memory to function well because of better training-inference temporal alignment in memory lengths. This supports our intuition in Sec. 4.3 and suggests the emerging opportunities from restricting memory banks.

Analysis on Regular Short Video Benchmarks. We

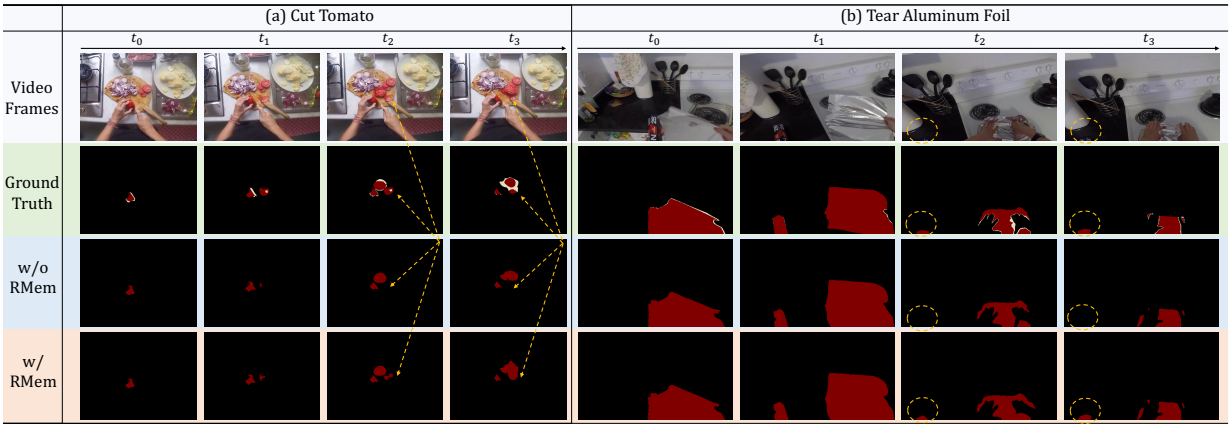

Figure 5. (Best viewed zoom-in with color.) Qualitative VOS results for object state changes on VOST [34]. We provide two examples showing the challenges of object state changes, including slicing, occlusions, distraction from similar objects (other tomatoes), and shape changes. For both scenarios, using RMem shows advantages in robustly maintaining the masks of the target objects, as highlighted. (White pixels are annotated by VOST denoting "ignored" regions for evaluation, which are hard and ambiguous even for human annotators.)

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_{tr}$	$\mathcal{J}$	Max Mem ↓	FPS
AOT	85.2	82.5	87.9	4.46G	13.67
AOT + RMem (Ours)	85.2	82.4	88.0	2.34G	15.57
DeAOT	85.2	82.3	88.1	2.24G	25.11
DeAOT + RMem (Ours)	**85.3**	**82.4**	**88.2**	**1.53G**	**27.42**

Table 6. RMem maintains the accuracy on DAVIS2017 while being more efficient, indicating that RMem can be generally applied, not limited to challenging scenarios. This also aligns with the prior works and suggests that not having demanding datasets was potentially why the *accuracy* benefits of memory restriction were not clearly revealed previously.

highlight the improvement on long and complex VOS datasets, but we also supplement our analysis on the regular and short video datasets DAVIS2017. As in Table 6, our RMem has relatively the same performance while being an effective measure to improve efficiency. Compared to our improvement on VOST and the Long Video dataset, we conjecture that the learned VOS modules (AOT and DeAOT) are already capable of handling shorter video duration and less complicated scenarios, even without our concise memory banks. Additionally, this potentially explains that previous studies exploring restricting memory banks [23, 25] have not explicitly discovered its benefits, *probably due to not having longer and more challenging datasets like VOST*.

5.5. Qualitative Results

We visualize on two videos from VOST [34] that demand robust spatio-temporal reasoning. Video (a) is the kitchen behavior of cutting a tomato into slices, and it illustrates the challenges of splitting objects, occlusions from hands, and visual distraction from other tomatoes. Without our RMem, the baseline AOT model fails to maintain the masks for the separated tomato slice, while using RMem correctly remembers this slice at the later stage of the video (columns 3 and 4). Such regions are highlighted with the yellow arrows. The other video (b) illustrates another difficulty of object shape transformation and the splitting between the box and aluminum. Although the baseline model without RMem can correctly segment the box at the beginning of splitting (column 2), it gradually loses track of the box and can only concentrate on the dominant object. However, our model enhanced with RMem robustly segments the small regions of the box, indicating that its attention association with relevant historical frames is still stable thanks to our restricted memory. Therefore, we conclude that the quantitative results reveal the difficulties of object state changes and support the effectiveness of our approach.

6. Conclusions

Limitations and Future Work. Our paper prioritizes the analysis of memory banks and illustrates our insight with a straightforward approach. Therefore, interesting future work is to combine the intuition or techniques from more sophisticated methods, such as XMem [9]. Furthermore, our exploration mainly adapts memory banks to the capability of VOS modules, while how to effectively improve the decoding ability of VOS modules for a huge memory bank is the alternative direction and interesting future work.

Conclusion. This paper reveals the drawbacks of expanding memory banks, a conventional design in VOS. Our insight stems from a novel "memory deciphering" analysis, which suggests that the redundant information in growing memory banks confuses the attention of VOS modules and elevates the difficulty of feature decoding. Then, we propose the simple enhancement for VOS named RMem, whose center is restricting the size of memory banks, accompanied by UCB-inspired memory update strategies and temporal positional embedding to enhance spatio-temporal reasoning. Extensive evaluation of the recent challenging datasets, including VOST and the Long Videos dataset, supports our insight and effectiveness of RMem.

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 *CVPR*, 2022. 2, 6

[2] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. TarVis: A unified approach for target-based video segmentation. In *CVPR*, 2023. 2

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 2002. 2, 5, 7

[4] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 2

[5] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2

[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 2

[7] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 2

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2

[9] Ho Kei Cheng and Alexander G Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 1, 2, 5, 6, 8

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021. 6

[11] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021. 1, 2, 3, 5, 6

[12] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *CVPR*, 2024. 2

[13] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. 2

[14] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR benchmark: Video segmentations and object relations. In *NeurIPS*, 2022. 2

[15] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *ICCV*, 2023. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5, 7

[17] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. SSTVos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, 2021. 2

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[20] Lingyi Hong, Wenchao Chen, Zhongying Liu, Wei Zhang, Pinxue Guo, Zhaoyu Chen, and Wenqiang Zhang. LVOS: A benchmark for long-term video object segmentation. In *ICCV*, 2023. 2, 6

[21] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 2

[22] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 6

[23] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2, 8

[24] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, 2020. 2, 3, 5

[25] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NeurIPS*, 2020. 1, 2, 5, 6, 8

[26] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 41(6):1515–1530, 2018. 2

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 7

[28] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 2

[29] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 6

[30] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 6

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

[32] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020. 2

[33] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020. 2

[34] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *CVPR*, 2023. 1, 2, 3, 6, 8

[35] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M Rehg. Motion coherent tracking using multi-label mrf optimization. *IJCV*, 100:190–202, 2012. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[37] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2

[38] 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 *CVPR*, 2019. 2

[39] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021. 2

[40] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 2, 6

[41] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. *CVPR*, 2024. 1

[42] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 2, 6

[43] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 2

[44] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 2, 6

[45] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 1, 2, 3, 5, 6

[46] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 44(9):4701–4712, 2021. 2, 6

[47] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *ICCV*, 2023. 1, 2