See Fine, Know Clear: Improving Video Object Segmentation via Fine-Grained Matching and MambaVision-Powered Semantics

Dongpei Dong[1], Bo Li[1*], Zhiheng Zhou[1], Delu Zeng[1]
[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China
eedpdong@mail.scut.edu.cn, *leebo@scut.edu.cn, zhouzh@scut.edu.cn, dlzeng@scut.edu.cn

***Abstract*—Memory-bank-based methods have attained promising performance in semi-supervised video object segmentation (SVOS). In most existing methods, the memory bank only stores single-scale coarse-grained features, thus leading to suboptimal performance in fine-grained segmentation. Meanwhile, memory matching is only conducted at the pixel level, which lacks high-level semantic modeling of objects as holistic entities, thus causing ambiguity in distinguishing similar objects. In this work, we propose an SVOS framework based on multi-scale memory representation and semantic enhancement. It stores multi-scale fine-grained features in the memory bank and fuses instance-level semantic information extracted from MambaVision, thereby boosting sensitivity to fine-grained object details while mitigating ambiguity in distinguishing similar objects. We thus name the proposed model See Fine, Know Clear (SFKC). This design faces a key challenge: elevated computational costs induced by fine-grained feature matching. To mitigate this issue, we design a Top-K-based filtering mechanism that optimizes the matching process and reduces computational overhead. Extensive experiments verify that our method enables accurate and robust foreground object segmentation, while retaining high efficiency when incorporating fine-grained memory features.**

***Index Terms*—Video object segmentation, Semantic enhancement, Fine-grained feature, Memory bank.**

I. Introduction

Video object segmentation (VOS) is a fundamental task in computer vision that aims to accurately segment target objects throughout a video sequence. Existing VOS methods are typically categorized into unsupervised, semi-supervised, language-guided, and interactive approaches [1]. In this work, we focus on semi-supervised VOS (SVOS), where the model is provided with one or more annotated frames and required to segment the target objects in the remaining frames accordingly.

In recent years, memory-based SVOS methods have achieved remarkable performance and become the dominant paradigm in this field. These approaches maintain a memory bank that stores features from previous frames, enabling the model to reuse historical information and improve segmentation accuracy. The Space-Time Memory Network (STM) [2] is a pioneering work in SVOS. STM and its subsequent variants [3]–[5] adopt an encoder–decoder framework to encode image and mask features into the memory bank and retrieve relevant information for the current query frame via space–time attention.

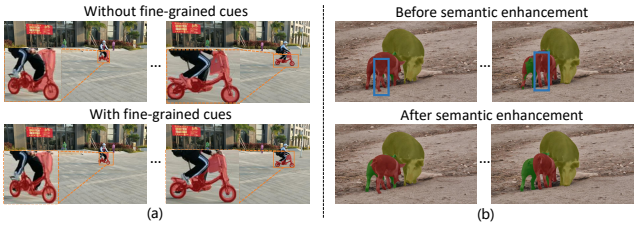

Fig. 1. (a) Coarse-grained mode results in insufficient segmentation on wheels of the bike, while fine-grained mode achieves a finer wheel segmentation. (b) Attention matching with original features leads to a false matching on similar objects (the segmentation of two pigs confused), while instance-level semantic information resolves the issue.

Despite their success, memory-based methods still suffer from two major limitations. First, memory matching is typically performed at a single coarse feature level, such as the `res4` stage of a ResNet [6] backbone, whose resolution is only $1/16$ of the original image. Consequently, the memory bank stores only coarse-grained features, leading to the loss of fine details that are crucial for segmenting small or subtle object parts, as illustrated in Figure 1(a). Second, space–time attention relies solely on pixel-level feature similarity and lacks high-level semantic modeling of objects as holistic entities. When distinct objects exhibit similar textures or appearances, their backbone features become ambiguous, making attention matching prone to errors. As a result, distinguishing visually similar objects becomes difficult, leading to incorrect segmentation, as shown in Figure 1(b).

To address the first issue, an intuitive solution is to incorporate fine-grained memory features into the memory bank. However, the high resolution of fine-level feature maps leads to prohibitively high computational cost, with complexity on the order of $\mathcal{O}(TH^2W^2)$. Inspired by [7], [8], we introduce an efficient multi-scale memory reading mechanism in which coarse-grained memory matching guides fine-grained matching through a Top-K selection strategy. This reduces the complexity of fine-grained memory reading to $\mathcal{O}(KHW)$ while preserving the detailed information required for segmenting small-scale objects.

To tackle the second issue, we enhance the model's ability

* is the corresponding author.

to distinguish between different instances of the same category and visually similar objects by injecting instance-level semantic information into the pixel-level memory features. Inspired by [9], [10], we design a semantic enhancement module that integrates high-level semantic cues capturing global shape, structural patterns, and instance-specific characteristics. However, extracting such semantic information poses two challenges: (1) SVOS datasets do not provide category labels and thus lack direct semantic supervision, and (2) large pre-trained models such as Vision Transformers (ViT) [11] offer rich semantics but incur high computational cost during inference. Recent Mamba-based architectures [12] provide an efficient alternative with strong representational capacity. Accordingly, we adopt MambaVision [13] as the semantic extractor to derive global instance-level semantics from query frames and fuse them with pixel-level memory features, thereby disambiguating matching among similar-looking objects.

In summary, the proposed framework equips the model with two complementary capabilities: capturing fine-grained object details (See Fine) and robustly distinguishing target instances (Know Clear).

Extensive experiments on challenging benchmarks, including the complex-scene dataset MOSE [14] and the long-video dataset LVOS [15], demonstrate the effectiveness and robustness of the proposed method. Our main contributions are summarized as follows:

1) We propose an efficient multi-scale memory reading module, in which Top-K guided coarse-grained matching directs fine-grained retrieval, reducing the computational complexity from $\mathcal{O}(TH^2W^2)$ to $\mathcal{O}(KHW)$.
2) We introduce a semantic enhancement module that injects instance-level semantic cues into pixel-level memory read-out features via attention-based fusion, improving semantic discriminability among visually similar objects in complex scenes.
3) We design a local attention correction module to refine spatial correspondence, enabling more accurate matching and precise object boundary segmentation.

II. Related Works

Semi-supervised video object segmentation methods can be broadly categorized into four paradigms: online learning, mask propagation, matching-based, and memory-based approaches. Online learning methods [16] pre-train a category-agnostic model and fine-tune it online for specific objects, achieving strong adaptability at the expense of substantial inference overhead. Mask propagation methods [17] exploit the smooth motion assumption to propagate masks across frames. Matching-based methods [18] perform pixel-level or region-level cross-frame matching, which provides the foundation for subsequent memory-based methods.

STM [2], a seminal memory-based VOS method, significantly improves segmentation accuracy and inspires a large body of follow-up work. Subsequent studies mainly advance along four directions: (1) optimizing memory construction and update strategies [19]; (2) improving memory bank design to enhance computational efficiency [20]; (3) strengthening spatio-temporal correspondence [21]; and (4) alleviating semantic confusion among similar objects by incorporating stronger semantic cues [9].

III. Methodology

A. Overview

The proposed SFKC is illustrated in Figure 2. The framework follows a recurrent memory-based segmentation paradigm. The first annotated frame is encoded and stored in the memory bank as an initial reference. For each subsequent frame, the query image is encoded and matched against the stored multi-scale memory features, and the resulting attention maps are refined by the local attention correction module to suppress mismatches and background interference.

Then the memory readout is augmented with instance-level semantic cues extracted by MambaVision through the semantic enhancement module to improve instance discrimination. Subsequently, the segmentation mask is predicted by decoding multi-scale memory readouts together with image features. Finally, the newly segmented frame is re-encoded and selectively inserted into the memory bank, which is updated based on frame importance and temporal proximity.

B. Encoder and Reading of Memory Frames

Both the query-frame encoder and the memory-frame encoder adopt ResNet-18 as the backbone. The `res4` features are further processed to generate the query keys and memory values. Since the two encoders process different input modalities, their parameters are not shared. When a processed frame is inserted into the memory bank, its query key is reused as the corresponding memory key. All encoder backbones are initialized with ImageNet [22]-pretrained weights.

For the query-frame encoder, the query frame $F_Q \in \mathbb{R}^{3\times H_{\text{ori}}\times W_{\text{ori}}}$ is fed into the ResNet-18 backbone to obtain the `res4` feature map $f^Q_{\text{res4}} \in \mathbb{R}^{C\times H\times W}$, where $H = H_{\text{ori}}/16$ and $W = W_{\text{ori}}/16$. A 1×1 convolution is then applied to compress the channel dimension, producing the query key K^Q.

For the memory-frame encoder, the input is formed by concatenating the video frame $F_M \in \mathbb{R}^{3\times H_{\text{ori}}\times W_{\text{ori}}}$ with its corresponding mask $M \in \mathbb{R}^{1\times H_{\text{ori}}\times W_{\text{ori}}}$ along the channel dimension. The concatenated tensor $[F_M, M] \in \mathbb{R}^{4\times H_{\text{ori}}\times W_{\text{ori}}}$ is encoded by the ResNet-18 backbone followed by a channel-compression layer, yielding the memory value features.

To retrieve relevant temporal information from the memory bank, we adopt the anisotropic L_2-attention mechanism proposed in [5] for memory reading, which enables efficient and robust feature matching across frames.

C. Semantic Enhancement Module

In this section, we design a semantic enhancement module (SEM), which incorporates instance-level semantic information into the memory readout features to strengthen the model's discriminative capability in complex scenes. As shown in Figure 2, we use a pretrained and frozen MambaVision model to extract semantic features from the query frame,

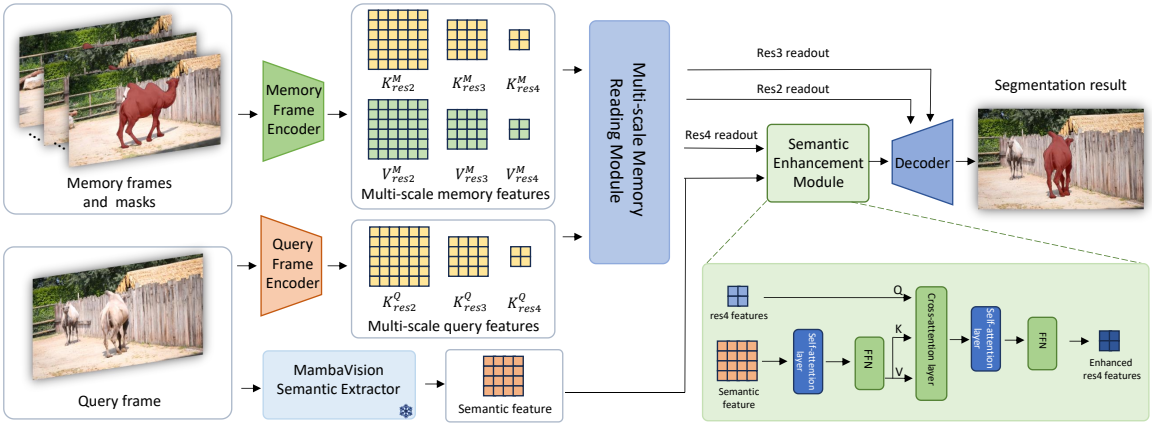

Fig. 2. Overview of SFKC. The query frame and memory frames are encoded at multiple scales by their respective encoders. After acquiring the multi-scale encoded features, they are fed into the multi-scale memory reading module for three-scale (`res2`, `res3`, `res4`) memory reading, where the `res4` reading result guides the reading operations at the `res2` and `res3` scales. Subsequently, the semantic enhancement module injects instance-level semantics extracted by MambaVision into the preliminary `res4` read features via cross-attention for semantic refinement. Finally, the features read at `res2` and `res3`, combined with the enhanced `res4` features, are fed into the decoder to generate the final segmentation mask, which is then stored in the memory bank for updating.

and progressively inject them into the preliminary `res4` memory readout P_0. Specifically, we cascade L SEM blocks to iteratively refine the pixel-level features P_0 with semantic cues, suppressing noise while enhancing instance-level distinctiveness. This design improves segmentation robustness in scenes containing visually similar objects.

Semantic Feature Extraction and Refinement. For each query frame, the original image F_Q is fed into the MambaVision to extract semantic features. From the fourth stage of MambaVision, we extract a deep local feature map $f^Q_{\text{Local}} \in \mathbb{R}^{C' \times h \times w}$ and a global average pooling feature $f^Q_{\text{Global}} \in \mathbb{R}^{C' \times 1}$. To handle the inconsistent input sizes between training and inference, we apply adaptive average pooling to downsample f^Q_{Local} to 15×15, and then flatten it to obtain $f^{Q'}_{\text{Local}} \in \mathbb{R}^{C' \times 225}$. By concatenating it with the global semantic descriptor along the channel dimension, we form the instance-level semantic representation: $f^Q_{\text{Semantic}} \in \mathbb{R}^{C' \times (225+1)}$.

To further refine the semantic representation for downstream attention fusion, we apply a Transformer-style self-attention and feed-forward network:

$$f^Q_{\text{Attn}} = f^Q_{\text{Semantic}} + \text{Dropout}\Big(\text{MHSA}(\text{LN}(f^Q_{\text{Semantic}}), pe)\Big), \quad (1)$$

$$f^Q_{\text{Enhanced}} = f^Q_{\text{Attn}} + \text{FFN}(f^Q_{\text{Attn}}), \quad (2)$$

where MHSA denotes multi-head self-attention, pe denotes positional encoding, and LN denotes layer normalization. This refinement strengthens the semantic robustness and improves generalization to complex video scenes.

Semantic Information and Memory Readout Incorporation. The refined semantic features f^Q_{Enhanced} are then fused with the preliminary memory readout P_0 using multi-head cross-attention. Treating P_0 as the query and f^Q_{Enhanced} as both key and value, we obtain the semantically enriched memory readout:

$$P^Q_{\text{Enhanced}} = P_0 + \text{Dropout}\Big(\text{MHCA}\big(P_0, W^K f^Q_{\text{Enhanced}}, W^V f^Q_{\text{Enhanced}}\big), pe_q, pe_{kv}\Big), \quad (3)$$

Here, MHCA denotes multi-head cross-attention, W^K and W^V denote the projection matrices for the key and value, and pe_q and pe_{kv} are the positional encodings for the query and key-value inputs, respectively.

Enhanced Feature Generation. After obtaining the semantically enriched memory readout P^Q_{Enhanced}, we employ a self-attention layer followed by a FFN to further enhance the feature representation, obtaining the final memory readout P^Q_{out}.

D. Local Attention Correction Module

In previous works, memory-based methods rely on global attention, where each pixel in the query frame is matched against all pixels in the memory frames. However, global matching is easily degraded by visually similar objects and background clutter, leading to mis-classification. To mitigate these issues, we propose a local attention correction module (LACM), converting global matching into a spatially constrained local matching process. The module consists of two components: *Top-K filtering* and *Gaussian kernel guidance*.

Global attention weights usually contain substantial noise. Top-K filtering is employed to remove a large number of low-confidence matches (commonly originating from background or ambiguous regions), thereby reducing interference in subsequent steps. Nevertheless, although the retained Top-K points are relatively reliable, their weights may lack spatial coherence. In particular, pixels belonging to the same local region of the target may exhibit overly diverse weight magnitudes. To address this issue, we introduce a Gaussian kernel that adjusts attention weights according to spatial distance: nearby pixels are assigned higher weights, while distant pixels are suppressed. This restores spatial smoothness and enforces local structural consistency.

Concretely, given the original attention matrix $W_{\text{ori}} \in \mathbb{R}^{(THW) \times HW}$, we first reshape it into $W_{\text{ori}} \in \mathbb{R}^{T \times HW \times HW}$, and apply Top-$K$ filtering along the second dimension (i.e., for each memory frame). Only the largest K attention values

are preserved while the rest are set to zero, yielding the filtered matrix $W_{\text{topk}} \in \mathbb{R}^{T\times HW\times HW}$.

To further impose spatial locality, we design a Gaussian-kernel-guided correction mechanism. Specifically, we locate the Gaussian kernel center for each query pixel by identifying the position with the maximum weight in W_{topk} along the memory frame pixel dimension (dim=2), and its coordinates form the matrix:

$$I = \text{Argmax}(W_{\text{topk}},\ \text{dim} = 2) \in \mathbb{R}^{T\times 2\times HW}. \tag{4}$$

Then, let $X \in \mathbb{R}^{T\times 2\times HW}$ represent the 2D coordinate grid of all pixels in each memory frame, where $X[t,:,j]$ denotes the 2D coordinates of the j-th pixel in the t-th memory frame. The Gaussian weight $\kappa[t,j,i]$ for each pair of query pixel i and memory pixel j in the t-th memory frame is then computed as:

$$\kappa[t,j,i] = \exp\left(-\frac{\|X[t,:,j] - I[t,:,i]\|}{2\sigma^2}\right), \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean distance, σ is the kernel bandwidth, and $\kappa \in \mathbb{R}^{T\times HW\times HW}$.

The Gaussian weight matrix exhibits a peak weight at the kernel center and exponential decay with increasing distance from the center, effectively confining the matching range to the local window. Finally, the corrected attention matrix is obtained by element-wise multiplication:

$$W_{\text{out}} = W_{\text{topk}} \odot \kappa. \tag{6}$$

E. Multi-scale Memory Reading Module

Most existing methods adopt a single-scale memory reading strategy, which often leads to inaccurate correspondence after decoder upsampling and results in the loss of fine spatial details, especially for small objects and object boundaries. To address this limitation, we propose a Multi-scale Memory Reading Module (MMRM) that jointly performs coarse-grained and fine-grained matching, enabling the model to capture rich spatial cues while maintaining computational efficiency.

A major challenge in multi-scale reading lies in reducing the computational cost of fine-grained matching, since high-resolution feature maps contain substantially more pixels than coarse ones. Observing that object locations across different resolutions are spatially consistent, we use coarse-grained memory readout to guide fine-grained matching. Specifically, we introduce a Top-K-based guidance mechanism that significantly reduces computation while preserving detailed information. We illustrate the process using `res4`-guided `res3` matching as an example.

Top-K-guided Multi-scale Reading and Matching. After obtaining the `res4` memory readout and its attention weight matrix, we select the top K positions with the highest attention responses as candidate locations for fine-grained matching. These positions are then upsampled to the `res3` resolution, yielding $4K$ guided points, since each `res4` pixel corresponds to four pixels in `res3`. During `res3` memory reading, the model only retrieves memory information at these guided locations. Since small spatial misalignments may arise from downsampling between `res3` and `res4`, the corresponding object positions across scales are not strictly rigid. To improve robustness, each `res3` pixel aggregates memory information from all $4K$ guided points rather than only K points, as shown in Figure 3.

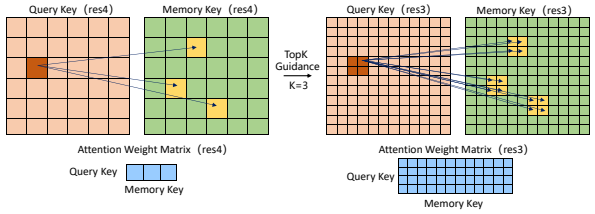

Fig. 3. An example of Top-K-guided memory reading, where $K = 3$. For simplicity, only the matching process between a single query key and memory keys in `res3` is shown. The corresponding attention weight matrix for memory reading of the brown query key is displayed at the bottom.

Extension to the `res2` Stage. The same strategy is applied to the `res2` feature map. To maintain computational consistency with `res3` reading, we select the top $K/4$ points from the `res4` weight matrix. After mapping these positions to the higher-resolution `res2` feature map, the total number of guided positions again becomes $4K$, ensuring uniform computational complexity across scales.

Fusion and Output. The memory readouts from `res2` and `res3` are first filtered using dropout to suppress noisy responses. Then, in the decoder, they are fused with the `res4` memory readout results via an adaptive gating fusion network to generate the segmentation results.

F. Decoder

We adopt a decoder similar to XMem [5] to restore readout memory to the original image resolution and obtain segmentation results. Different from XMem, our decoder explicitly incorporates fine-grained memory features, adaptively fusing them with the previous-stage read-out through a gating mechanism.

IV. Experiment

A. Training Settings

Datasets. DAVIS-2017 is a widely used benchmark dataset for video object segmentation that supports multi-object scenarios. YouTube-2019 is a large-scale multi-object dataset whose validation and test sets contain additional "unseen" categories that are not present in the training set. LVOS is a long-term video object segmentation dataset with an average video length of 1.59 minutes. MOSE includes more challenging scenarios, such as frequent object disappearance and reappearance, severe occlusion, and dense congestion, which better reflect real-world conditions. We evaluate our model on these four datasets and compare it with 12 state-of-the-art methods.

Loss Function. Following [10], we adopt a point-supervision strategy. The training objective is a weighted combination of cross-entropy loss and soft Dice loss with equal weights.

TABLE I

QUANTITATIVE COMPARISON ON DAVIS-2017, YOUTUBE-2019 VAL, MOSE VAL AND LVOS VAL. THE BEST AND SECOND-BEST PERFORMANCE ARE MARKED WITH **BOLDFACE** AND UNDERLINE, RESPECTIVELY. FPS STANDS FOR THE NUMBER OF VIDEO FRAMES PROCESSED PER SECOND.

Method	DAVIS-2017							YouTube-2019-Val					MOSE			LVOS		
	Validation Set			Test-dev Set			FPS	$\mathcal{G}$	Seen		Unseen		Validation Set			Validation Set		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$			$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STM [2]	81.8	79.2	84.3	72.2	69.3	75.2	18.9	79.2	79.6	83.6	73.0	80.6	-	-	-	-	-	-
STCN [4]	85.4	82.2	88.6	76.1	72.7	79.6	69.5	82.7	81.1	85.4	78.2	85.9	52.5	48.5	56.6	45.8	41.1	50.5
RDE [23]	84.2	80.8	87.5	77.4	73.6	81.2	69.6	81.9	81.1	85.5	76.2	84.8	48.8	44.6	52.9	52.9	47.7	58.1
XMem [5]	86.2	82.9	89.5	81.0	77.4	84.5	75.5	85.5	84.3	88.6	80.3	88.6	56.3	52.1	60.6	50.0	45.5	54.4
QDMN [19]	84.6	81.8	87.3	77.7	74.2	81.2	35.9	-	-	-	-	-	52.8	50.0	55.6	48.8	43.7	53.9
AOT [24]	84.9	82.3	87.5	79.6	75.9	83.3	40.7	84.1	83.5	88.1	78.4	86.3	57.2	53.1	61.3	56.9	51.8	61.9
DeAOT [25]	85.2	82.2	88.2	80.7	76.9	84.5	61.0	85.9	84.6	89.4	**80.8**	**88.9**	-	-	-	-	-	-
DeVOS [21]	86.1	83.4	88.8	81.0	77.2	84.7	80.5	85.2	84.5	89.5	79.4	87.4	-	-	-	-	-	-
DEVA [26]	86.8	83.6	90.0	82.3	78.7	85.9	63.3	85.5	85.0	89.4	79.7	88.0	60.0	55.8	64.3	58.3	52.8	63.8
DDMemory [15]	84.2	81.3	87.1	-	-	-	93.8	-	-	-	-	-	-	-	-	**60.7**	55.0	**66.3**
ISVOS [9]	87.1	83.7	90.5	**82.8**	**79.3**	**86.2**	9.8	**86.1**	**85.2**	**89.7**	80.7	88.9	-	-	-	-	-	-
Cutie-R18 [10]	86.3	82.8	89.9	81.1	77.4	84.8	76.9	84.9	84.4	88.7	78.9	87.5	60.6	56.6	64.5	59.2	54.2	64.2
SFKC(Ours)	**87.7**	**84.4**	**90.9**	82.0	78.3	85.7	34.0	85.2	84.2	88.5	80.0	88.2	**61.5**	**57.5**	**65.6**	60.6	**55.6**	65.7

Training Settings. The model is trained for 150K iterations, with the learning rate decayed to 0.1 times its current value at 90K and 130K iterations. The initial learning rate is set to 5×10^{-5}, the batch size is 4, and the AdamW optimizer [27] is used. Mixed training is performed on the DAVIS-2017 [28] and YouTube-2019 [29] datasets. Unlike most previous VOS methods that rely on static image datasets for pre-training [2], we train exclusively on video datasets, since static pre-training provides limited performance gains while substantially increasing training time.

Inference Settings. In the SEM, the L parameter for cascaded blocks is set to $L = 3$. In the LACM, the Top-K parameter is set to $K = 30$. In the MMRM, the Top-K values at the `res2` and `res3` stages are set to 10 and 40, respectively.

Metrics. We evaluate performance using the standard metrics: the region accuracy $\mathcal{J}$, the boundary accuracy $\mathcal{F}$, and their average $\mathcal{J}\&\mathcal{F}$ [2].

B. Main Results

The quantitative experimental results are summarized in Table I.

Evaluation on DAVIS-2017. As shown in Table I, our SFKC model achieves a $\mathcal{J}\&\mathcal{F}$ score of 87.7 on the DAVIS-2017 validation set, outperforming all competing methods. On the DAVIS-2017 test-dev set, our model also demonstrates strong performance, indicating its robustness and effectiveness in standard benchmarks.

Evaluation on YouTube-2019. Our model obtains a $\mathcal{J}\&\mathcal{F}$ score of 85.2 on the YouTube-2019 dataset, surpassing most existing baseline methods. Notably, our method achieves superior performance on the "Unseen" category compared with the majority of competitors, suggesting a stronger generalization capability to previously unseen object categories.

Evaluation on MOSE and LVOS. On the MOSE dataset, our model achieves a $\mathcal{J}\&\mathcal{F}$ score of 61.5, outperforming all baseline methods. On the LVOS dataset, it attains a $\mathcal{J}\&\mathcal{F}$ score of 60.6, surpassing most competing approaches and demonstrating its effectiveness in long-term video object segmentation scenarios.

TABLE II

THE ABLATION STUDY OF EACH MODULE.

Index	SEM	MMRM	LACM	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS
1				83.6	80.5	86.6	137.7
2	✓			85.1	81.7	88.6	54.4
3		✓		86.5	82.9	90.0	66.7
4			✓	84.6	81.5	87.6	113.5
5	✓	✓		86.9	83.7	90.1	37.8
6	✓	✓	✓	87.7	84.4	90.9	34.0

C. Ablation Study

We conduct an ablation study on the DAVIS-2017 validation set to evaluate the contribution of each component in the proposed SFKC framework, including the semantic enhancement module (SEM), the multi-scale memory reading module (MMRM), and the local attention correction module (LACM). The quantitative and qualitative results are reported in Table II and Figure 4, respectively.

As shown in Table II, all three modules contribute positively to the overall performance. Among them, MMRM provides the largest performance gain, improving $\mathcal{J}\&\mathcal{F}$ by 2.9 points, while SEM and LACM improve the performance by 1.5 and 1.0 points, respectively. These results demonstrate that each component plays a complementary role in enhancing segmentation accuracy.

The MMRM introduces low-level object features into the matching process, improving sensitivity to fine-grained details. As shown in Figure 4(a), without MMRM the model gradually fails to track small objects and fine structures, leading to foreground misclassification. Incorporating MMRM effectively preserves object details and improves segmentation stability.

The LACM improves the $\mathcal{J}\&\mathcal{F}$ score by 1.0 point. Figure 4(b) shows that without LACM the model tends to segment nearby distractors due to noisy attention. By constraining the matching region, LACM reduces interference from irrelevant areas and enhances segmentation accuracy.

The SEM integrates instance-level semantic information into frame features, improving discrimination in complex scenes. As illustrated in Figure 4(c), without SEM the model

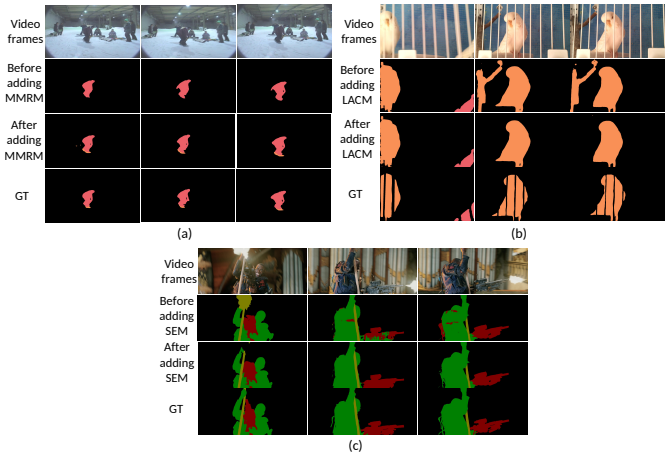

Fig. 4. Qualitative ablation results. (a), (b) and (c) show the effectiveness of MMRM, LACM and SEM, respectively.

often confuses similar objects or splits one object into multiple instances. This issue is largely alleviated after introducing SEM, demonstrating its effectiveness in enhancing object recognition.

V. Conclusion

In this work, we propose SFKC, an efficient memory reading framework for video object segmentation. By introducing fine-grained feature matching and semantic enhancement information, it addresses the suboptimal segmentation performance caused by single-scale coarse-grained memory reading and the ambiguity from similar targets induced by pixel-level matching. Extensive experiments show that the proposed method achieves strong performance in challenging scenarios with similar objects and complex motion. It provides an effective and scalable solution for reliable correspondence modeling in video object segmentation.

References

[1] M. Gao, F. Zheng, J. J. Yu, C. Shan, G. Ding, and J. Han, "Deep learning for video object segmentation: a review," *Artificial Intelligence Review*, vol. 56, no. 1, pp. 457–531, 2023.

[2] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9226–9235.

[3] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 629–645.

[4] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 781–11 794, 2021.

[5] H. K. Cheng and A. G. Schwing, "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *European Conference on Computer Vision*. Springer, 2022, pp. 640–658.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[7] Y. Yi, Y. Gui, and Z. Liu, "Boosting memory network for video object segmentation in complex scenes," *The Visual Computer*, vol. 41, pp. 6571–6585, 2025.

[8] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, "Hierarchical memory matching network for video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 889–12 898.

[9] J. Wang, D. Chen, Z. Wu, C. Luo, C. Tang, X. Dai *et al.*, "Look before you match: Instance understanding matters in video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2268–2278.

[10] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing, "Putting the object back into video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3151–3161.

[11] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First Conference on Language Modeling*, 2024.

[13] A. Hatamizadeh and J. Kautz, "Mambavision: A hybrid mamba-transformer vision backbone," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 25 261–25 270.

[14] H. Ding, C. Liu, S. He, X. Jiang, P. H. Torr, and S. Bai, "Mose: A new dataset for video object segmentation in complex scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 224–20 234.

[15] L. Hong, W. Chen, Z. Liu, W. Zhang, P. Guo, Z. Chen *et al.*, "Lvos: A benchmark for long-term video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 480–13 492.

[16] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 221–230.

[17] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2663–2672.

[18] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. 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*, 2019, pp. 9481–9490.

[19] Y. Liu, R. Yu, F. Yin, X. Zhao, W. Zhao, W. Xia *et al.*, "Learning quality-aware dynamic memory for video object segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 468–486.

[20] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, "Swiftnet: Real-time video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1296–1305.

[21] V. Fedynyak, Y. Romanus, B. Hlovatskyi, B. Sydor, O. Dobosevych, Babin *et al.*, "Devos: Flow-guided deformable transformer for video object segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 240–249.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] M. Li, L. Hu, Z. Xiong, B. Zhang, P. Pan, and D. Liu, "Recurrent dynamic embedding for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1332–1341.

[24] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2491–2502, 2021.

[25] Z. Yang and Y. Yang, "Decoupling features in hierarchical propagation for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 324–36 336, 2022.

[26] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1316–1326.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

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

[29] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang *et al.*, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018.