Global Spectral Filter Memory Network for Video Object Segmentation

Yong Liu[1*], Ran Yu[1], Jiahao Wang[1], Xinyuan Zhao[3], Yitong Wang[2], Yansong Tang[1], and Yujiu Yang[1†]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] ByteDance Inc.
[3] Northwestern University
{liu-yong20, yu-r19}@mails.tsinghua.edu.cn, {tang.yansong, yang.yujiu}@sz.tsinghua.edu.cn

Abstract. This paper studies semi-supervised video object segmentation through boosting intra-frame interaction. Recent memory network-based methods focus on exploiting inter-frame temporal reference while paying little attention to intra-frame spatial dependency. Specifically, these segmentation model tends to be susceptible to interference from unrelated nontarget objects in a certain frame. To this end, we propose Global Spectral Filter Memory network (GSFM), which improves intra-frame interaction through learning long-term spatial dependencies in the spectral domain. The key components of GSFM is 2D (inverse) discrete Fourier transform for spatial information mixing. Besides, we empirically find low frequency feature should be enhanced in encoder (backbone) while high frequency for decoder (segmentation head). We attribute this to semantic information extracting role for encoder and fine-grained details highlighting role for decoder. Thus, Low (High) Frequency Module is proposed to fit this circumstance. Extensive experiments on the popular DAVIS and YouTube-VOS benchmarks demonstrate that GSFM noticeably outperforms the baseline method and achieves state-of-the-art performance. Besides, extensive analysis shows that the proposed modules are reasonable and of great generalization ability. Our source code is available at https://github.com/workforai/GSFM.

Keywords: video object segmentation, spectral domain

1 Introduction

Video Object Segmentation (VOS) [36,37,60,65] aims at identifying and segmenting objects in videos. It is one of the most challenging tasks in computer vision with many potential applications, including interactive video editing, augmented reality [32], and autonomous driving [73]. In this paper, we focus on the semi-supervised setting where target objects are defined by the given masks of

*This work was done during an internship at ByteDance Inc.
†Corresponding author

Fig. 1: Illustration of the disadvantages of lacking semantic global information. The highlight red pixels in the first column are target pixels. The second column shows that previous method [6] would incorrectly match similar pixels of other objects. In the third column, our model relieves the confusion problem by enhancing low-frequency components and updating features from spectral domain.

the first frame. It is crucial for semi-supervised VOS to fully utilize the available reference information to distinguish targets from background objects.

Since the critical problem of this task lies in how to make full use of the spatial-temporal dependency to recognize the targets, matching-based methods, which perform pixel-level matching with historical reference frames, have received tremendous attention. The Space-Time Memory Network [34] memorizes intermediate frames with segmentation masks as references and performs pixel-level matching between them with the current frame to segment target objects in a bottom-up manner, which has been proved effective and has served as the current mainstream framework. Some works [39,23,5,15,58,40,50,6,61,45,25,58] further develop STM and have achieved excellent performance.

Although these methods have made great progress in the field of VOS, they pay little attention to excavating intra-frame dependency and only utilize local representation for matching and prediction due to the inductive bias of convolution. Lacking global dependency would cause low efficacy in distinguishing similar pixels, *e.g.*, pixels of similar color or objects of the same category. We take the typical method STCN [6] for illustration. In Fig. 1 (b), some pixels belonging to background objects are mismatched with the target pixel due to their similar local features. Ignoring long-range dependency for matching would lead to a high risk of interference from other objects. Since the matching-based approaches rely on the matching process to identify the targets, incorrectly matched pixels would negatively affect the final segmentation and even lead to error accumulation. Therefore, it is necessary to excavate the intra-frame spatial dependency to enhance the representation of features.

According to the Fourier theory [19], FFT function generates outputs based on pixels from all spatial locations when processing input feature. Thus, the spec-

tral domain representation contains rich global information. Inspired by this, we introduce a Global Spectral Filter Memory network (GSFM), which fuses global dependency from spectral domain and distinguishes the high-frequency and low-frequency components for targeted enhancement. In GSFM, we propose the Low Frequency Module (LFM) and High Frequency Module (HFM) to enhance different representation according to the characteristics of the encoder-decoder network structure.

The role of encoder is to extract deep features for subsequent modules, and the encoded features need to contain rich semantic information. Intuitively, low-frequency components correspond to high-level semantic information while ignoring details. Some theoretical researches on CNN from spectral domain [63,51,69] also point out similar observations. Inspired by the above analysis, we propose a Low-Frequency Module (LFM) for the encoding process to update the features in the spectral domain and emphasize their low-frequency components. Fig. 1 (c) illustrates that with LFM enhancing global semantic information, the distinguishability of similar pixels is greatly improved. Extensive experiments also demonstrate the rationality of emphasizing low-frequency in the encoder.

Different from encoding, features in the decoding process need to contain more fine-grained information for accurate prediction. And high-frequency components correspond to the image parts that change drastically, *e.g.*, object boundaries and texture details. Combined with the above analysis, we believe that focusing on high-frequency components would help to rich the fine-grained representation of features and make more accurate predictions of boundaries or ambiguous regions. Therefore, we introduce a High-Frequency Module (HFM) in the decoding process, which enhances the high-frequency components of features to better capture detailed information. Besides, to take full advantage of HFM, we combine it with an additional boundary prediction branch to provide better localization and shape guidance.

Experiments show that the proposed model noticeably outperforms the baseline method and achieves state-of-the-art performance on DAVIS [36,37] and YouTube-VOS [60] datasets. The contribution of this paper can be summarized as follows. Firstly, we propose to leverage the spectral domain to enhance the global spatial dependency of features for semi-supervised VOS. Secondly, considering the differences between the process of encoding and decoding, we propose LFM and HFM to perform targeted enhancement, respectively. Thirdly, we combine object boundaries and high-frequency to provide better localization and shape information while keeping the decoding features are fine-grained.

2 Related Work

Semi-supervised video object segmentation. Since the masks for the first frame are given, early methods [3,48,29,49,56] take the strategy that online fine-tune the network according to the object mask of the first frame, which suffers from slow inference speed. Propagation-based methods [46,8,7,62,16,1,18,21,13] forward propagate the segmentation masks as a reference to the next frame,

and they are difficult to handle complicated scenarios. Some other researchers have decoupled VOS into three independent subtasks of detection, tracking, and segmentation [28,22,17,44]. Although this approach balances running time and accuracy, it is extremely dependent on the performance of the detectors and makes the entire pipeline complex.

In recent years matching-based methods have received great attention for excellent performance and robustness. FEELVOS [47], CFBI [66] and CFBI+ [68] perform global and local matching with the first frame and the previous adjacent frame, respectively. AOT [67] associates multiple target objects into the same embedding space by employing an identification mechanism. STM [34] leverages the memory network to memorize intermediate frames as references, which has been proved effective and has served as the current mainstream framework. Based on STM, KMN [39] and RMNet [58] perform local-to-local matching by using the Gaussian kernel and hard crop strategy. SwiftNet [50] and AFB-URR [25] reduce memory redundancy by calculating the similarity between query and memory. LCM [15] and SCM [72] proposes spatial constraint to enhance spatial location information. EGMN [27] employs an episodic memory network to memorize frames as nodes and capture cross-frame correlations by edges. MiVOS [5] further developed KMN [39] by utilizing the top-k strategy to reduce noise information in the memory read block. STCN [6] improves the feature extraction and performs more reasonable matching by decoupling the image and masks.

Despite the great performance achieved by these methods, they ignore the importance of fully excavating the intra-frame global information, which may lead to a high risk of interference by pixels with similar local features.

Spectral domain learning. Recent years have witnessed increasing research enthusiasm on combining spectral domain and deep learning [41,10,38,51,69,57]. Among them, some researches [63,51,69] attempt to explain the behavior of convolution neural network from the perspective of spectral domain. They point out that the features of different frequency bands represent different types of information and observe some properties of deep neural networks related to it. With the guidance of these works and rethinking about the characteristics of the encoder-decoder structure, we propose separating the high-frequency and low-frequency components for reasonably utilizing them. In this paper, we introduce a low-frequency module (LFM) and a high-frequency module (HFM). LFM enhances the low-frequency components during encoding to fuse global semantic features, while HFM enhances the high-frequency components in the decoder to make features contain more fine-grained details.

Some previous methods [25,75,64] applying spatial prior filter or introducing boundary to features can also be explained from the perspective of spectral domain. Applying filter kernels or highlighting boundaries in the spatial domain is essentially a special way to distinguish between high and low frequencies. While this approach can also serve the purpose of targeted enhancement, it loses the advantage of global perception in the spectral domain. Therefore, our approach that updates features in the spectral domain is more generalized and effective.

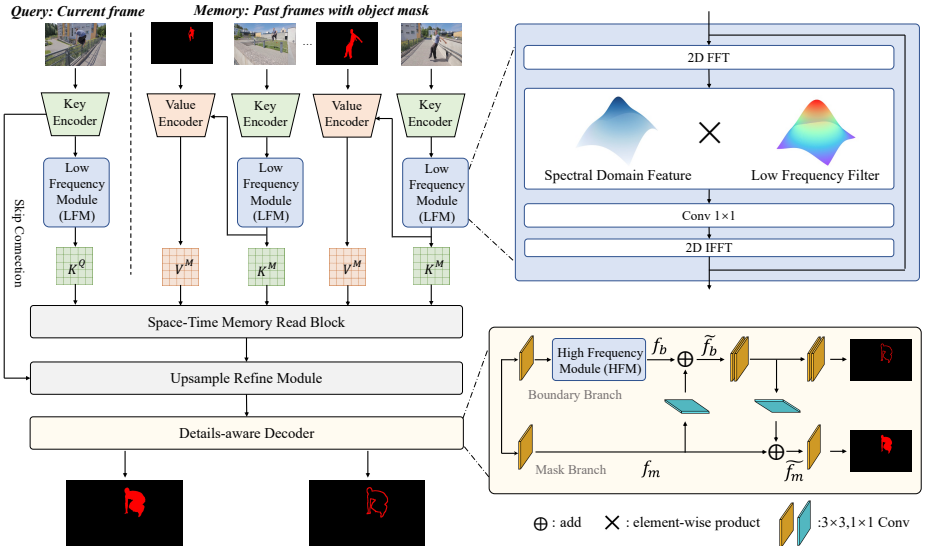


Fig. 2: Overview of GSFM. The network takes both query (current frame) and memory (past frames with masks) as input. LFM enhances low-frequency components of features and fuses global information in the spectral domain. Having K^M, K^Q and V^M extracted from the encoder, the memory read block calculates similarity between query and memory. The refine module upsamples the features and outputs to the decoder. With HFM enhancing high-frequency components, the decoder jointly predicts object masks and boundaries.

3 Method

3.1 Overview

The overall architecture of our GSFM is shown in Fig. 2. Given a video sequence and the annotation of the first frame, we process it frame by frame. During processing, the current frame is considered a query, and the past reference frames with segmentation masks are memory. Following the baseline method STCN [6], a Key Encoder extracts key features for each frame, and a Value Encoder extracts value features only for memory frames. By performing matching between query and memory in Space-Time Memory Read Block, the decoder identifies and segments the target object in a bottom-up manner. In the encoder, for exploiting the intra-frame semantic information to improve the representative capacity of features and promote the effectiveness of matching, a low-frequency module (LFM) enhances the low-frequency components of the features and performs global information updating from the spectral domain. In the decoder, the high-frequency module (HFM) enhances high-frequency components to highlight fine-grained information for accurate prediction. Besides, we take the strategy that jointly learning object boundaries and masks in an

end-to-end manner [74,26,9,43]. With the interaction between mask branch and boundary branch features, the network can better perceive the localization and shape information, which also helps identify the target objects.

3.2 Frequency Modules

According to the spectral convolution theorem [2] in Fourier theory, updating a single value in the spectral domain affects globally all original data, which sheds light on design operations with the non-local receptive field. Intuitively, the high-frequency components correspond to the pixels varying drastically, such as object boundaries and textures, while the low-frequency components correspond to the general semantic information. Some previous theoretical studies [63,51,69] on spectral-domain and deep learning also point similar observation. Besides, to show the information represented by different frequency components more vividly, we take Fig. 3 as an example (for convenience, we use the grayscale image). In Fig. 3, the remaining information after high-pass filtering is the edges and details of objects. After low-pass filtering, the result is an image that retains the general semantic information (some details and noise are blurred). Considering that the role of the encoder is to extract high-level semantic information, while the decoder pursues focus on detailed features, we believe that this difference is similar to the difference between the high and low-frequency components. Thus, we propose a low-frequency module (LFM) for the encoder and a high-frequency module (HFM) for the decoder.

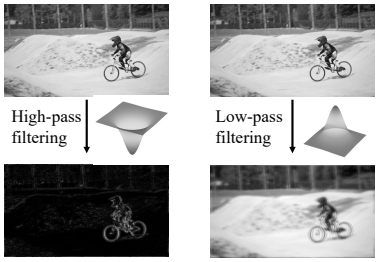


Fig. 3: Illustration of different frequency components. The top line is the original image.

The architecture of LFM and HFM is the same, and their difference is the frequency domain filter (LFM is a low-pass filter, and HFM is a high-pass filter). In our experiments, the filter is set in the form of Gaussian. Here we take LFM as an example to introduce the process. As shown in Fig. 4, having the image feature tensor x, LFM first transfers it to the spectral domain by FFT. Then the spectral features y will be passed through a low-pass filter to enhance low-frequency components, which helps to make features rich in global semantic information. Specifically, we generate a coefficient map g with the same spatial size of the feature y and perform element-wise multiplication between them with the help of broadcast mechanism. For LFM, the center of the coefficient map has the value of 0 and increases around in the form of Gaussian (without spectrum centralization, the center of the spectrum after FFT is high frequency, and the surrounding is low frequency). Before updating the spectral domain, note that the spectral features are complex numbers for the FFT operation. To make the complex number features compatible with the neural layers, we split the complex

Algorithm: Pseudocode of LFM

```
# x: input feature
# B: batchsize, C: dimension of channel, H, W: spatial size of input feature
# y_r, y_i: the real and imaginary part of the spectral features, respectively

# generate the Gaussian frequency filter
g = Make_gaussian_filter(H, W)   # g: (B, 1, H, W)
y = FFT(x)                       # y: (B, C, H, W)
y = y * g
# convert complex number features to real number
y_r, y_i = y.real, y.imag        # y_r, y_i: (B, C, H, W)
y = Concatenate([y_r, y_i], dim=1)   # y: (B, 2*C, H, W)
y = ReLU(Conv(y))
# convert back to complex number
y_r, y_i = Split(y, dim=1)       # y_r, y_i: (B, C, H, W)
y = Complex(y_r, y_i)
y = iFFT(y)                      # y: (B, C, H, W)

return x + y
```

Fig. 4: The Pseudocode of Low Frequency Module (LFM)

number into a real part y_r and an imaginary part y_i. For ease of computation, we append the imaginary part to the real part by concatenating them along the channel dimension, forming a new tensor with double channels. Essentially, the resultant tensor is treated as a vanilla real number tensor, and we can perform a series of neural layers on it. To update features in the spectral domain, we utilize 1×1 convolution with ReLU activation function. According to the convolution theorem [19], convolution in one domain equals point-wise multiplication in the other domain, which implies that the 1×1 convolution in spectral-domain incurs a global update in the spatial domain. After that, the results are converted back to complex numbers by splitting them into real and imaginary parts along the channel dimension. Inverse 2-D FFT operation transfers the spectral features back to the spatial domain. Finally, LFM outputs the enhanced features by adding the updated features y with initial tensor x.

3.3 Details-aware Decoder

Only utilizing local information for pixel-level mask prediction may lead to a lack of overall perception of the objects and an over-reliance on pixel appearance information such as pixel color. Intuitively, object boundaries and object masks have a close relation. It would be helpful to locate and identify target objects from the background if the model has some sense of the shape or boundary of the objects, especially with HFM highlighting detailed information. Besides, since semi-supervised VOS is a pixel-level tracking task, accurate boundary segmentation is significant. Otherwise, it is easy to cause error accumulation. Therefore,

we propose to combine HFM and object boundaries to provide localization and more detailed guidance.

The architecture of the decoder is shown in Fig. 2. Compared to the vanilla mask decoder of other memory network-based approaches, we add a branch dedicated to predicting object boundaries so that the model gives more attention to the object boundaries and shapes. The input feature of the boundary branch is first processed by HFM to enhance its high-frequency components, which helps to better perceive fine-grained information for accurate prediction. In addition, due to the special relationship between object boundaries and object masks, there is a lot of mutually exploitable information between their features. Specifically, features from the mask branch can provide basic information for localizing boundaries. After making sense of object boundaries, the shape and location information in boundary features is also conducive to guiding more precise mask predictions. To take full advantage of the special relationship between them, we take a fusion module [9] for the interaction between the mask branch and the boundary branch. Take the Mask $\rightarrow$ Boundary (M2B) Fusion as example, the fusion process can be formulated as follows:

$$\widetilde{f}_b = \mathcal{F}(f_m) + f_b, \tag{1}$$

where $\widetilde{f}_b$ denotes the fused boundary features, f_m is the mask branch feature, and f_b is the boundary branch feature. $\mathcal{F}$ is a 1×1 convolution with ReLU function. The fusion block is the same for the boundary $\rightarrow$ Mask (M2B) Fusion.

Boundary Ground Truth. Following previous works [59,70], we take the boundary prediction as a pixel-level classification problem. Since only the ground truth of the mask is available in the video object segmentation dataset, we use the Laplacian operator to generate the boundary ground truth. The Laplacian operator is a second-order gradient operator. As it is regarded as a classification problem, the resultant boundaries need to be converted into binary maps, and we binarize them with a threshold of 0.1.

Boundary Loss. Following previous work [9], we use dice loss [31] and binary cross-entropy to optimize the boundary predictions. Dice loss measures the overlap between predicted boundaries and ground truth. More importantly, dice loss can better handle category imbalance and focus on foreground pixels, which is compatible with boundary prediction (the number of boundary points is much less than points of non-boundary). The boundary loss $\mathcal{L}_b$ can be formulated as follows:

$$\mathcal{L}_b = \mathcal{L}_{Dice} + \mathcal{L}_{BCE}. \tag{2}$$

The dice loss is given as follows:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_i p^i q^i}{\sum_i (p^i)^2 + \sum_i (q^i)^2 + \epsilon}, \tag{3}$$

where p and q denote the predictions and ground truth, respectively. i denotes the i-th pixel and ϵ is a smooth term to avoid zero division.

3.4 Other Modules

Encoder. Following STCN [6], we construct a Key Encoder and a Value Encoder. For each frame, the key features are extracted only once. In other words, we would reuse the "query key" as the "memory key" if one frame is memorized into the memory during video sequences. For memory frames, since both memory keys and memory values are extracted from the same image, it is natural to reuse existing key features as the input of value encoder. Specifically, a backbone first extracts memory features from images with segmentation masks and the resultant features are concatenated with the last layer features from key encoder. Then two ResBlocks [14] and a CBAM block [55] process them and output the final memory value features V^M.

Space-Time Memory Read Block. The query frame and T memory frames are encoded into the followings: memory key $K^M \in \mathbb{R}^{C^k \times T \times H/16 \times W/16}$, memory value $V^M \in \mathbb{R}^{C^v \times T \times H/16 \times W/16}$, query key $K^Q \in \mathbb{R}^{C^k \times H/16 \times W/16}$

In the Space-Time Memory Read block, activation weights are computed by measuring the similarities between K^Q and K^M. Then the V^M is retrieved by a weighted summation with the weights to get the output M. This operation can be summarized as:

$$M_i = \frac{1}{Z} \sum_j \mathcal{D}(K_i^Q, K_j^M) V_j^M, \tag{4}$$

where i and j are the index of the query and the memory location, $Z = \sum_j \mathcal{D}(K_i^Q, K_j^M)$ is the normalizing factor. $\mathcal{D}$ denotes similarity measure (following [6], in our experiments we take the L2 distance as measurement).

Refine Module. We use the same refinement module as previous works [33,6,5]. The role of the refinement modules is to process the matched value features and merge the detail information from the shallow layer of the encoder.

4 Implementation Details

Following the training strategy in previous works [6,67,5], we first pretrain our model on static image datasets [52,42,71,4,20] and then perform main training on YouTube-VOS and DAVIS datasets. During pretraining, each image is expanded into a pseudo video of three frames by data augmentation. For main training, we randomly pick three frames in chronological order (with a ground-truth mask for the first frame) from a video to form a training sample. The range of random sampling varies with the training process. In the intermediate period of training, the sampling range is set larger to improve the robustness of the model, while at the end of training, it is set smaller to narrow the gap between training and inference. We use randomly cropped 384×384 patches for training.

Our models are trained with eight 32GB Tesla V100 GPUs with the Adam optimizer using PyTorch. The batch size is set to 16 for each GPU during pretraining and 8 during main training. It takes about 18 hours to perform pretraining and 6 hours for main training. We adopt ResNet50 [14] as backbone for

Table 1: The quantitative evaluation on DAVIS dataset. '*' indicates our re-implementation version. The results of baseline method are underlined

Method	DAVIS2016			DAVIS2017 val			DAVIS2017 test-dev			FPS
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	
RANet [53]	86.6	87.6	87.1	63.2	68.2	65.7	53.4	56.2	55.3	-
FEELVOS [47]	81.1	82.2	81.7	69.1	74.0	71.5	55.2	60.5	57.8	-
RGMP [33]	81.5	82.0	81.8	64.8	68.6	66.7	51.3	54.4	52.8	-
DMVOS [54]	88.0	87.5	87.8	-	-	-	-	-	-	-
STM [34]	88.7	89.9	89.3	79.2	84.3	81.8	69.3	75.2	72.2	7.9
KMN [39]	89.5	91.5	90.5	80.0	85.6	82.8	74.1	80.3	77.2	7.1
CFBI [66]	88.3	90.5	89.4	79.1	84.6	81.9	71.1	78.5	74.8	3.4
GCM [23]	87.6	85.7	86.6	69.3	73.5	71.4	-	-	-	-
G-FRTM [35]	-	-	84.3	-	-	76.4	-	-	-	-
GIEL [12]	-	-	-	80.2	85.3	82.7	72.0	78.3	75.2	-
SwiftNet [50]	90.5	90.3	90.4	78.3	83.9	81.1	-	-	-	20.6
RMNet [58]	88.9	88.7	88.8	81.0	86.0	83.5	71.9	78.1	75.0	<11.9
SSTVOS [11]	-	-	-	79.9	85.1	82.5	-	-	-	-
LCM [15]	89.9	91.4	90.7	80.5	86.5	83.5	74.4	81.8	78.1	<9.5
MiVOS [5]	87.8	90.0	88.9	80.5	85.8	83.1	72.6	79.3	76.0	6.5
JOINT [30]	-	-	-	80.8	86.2	83.5	-	-	-	3.8
RPCMVOS [61]	87.1	94.0	90.6	81.3	86.0	83.7	75.8	82.6	79.2	-
DMN-AOA [24]	-	-	-	81.0	87.0	84.0	74.8	81.7	78.3	<6.2
HMMN [40]	89.6	92.0	90.8	81.9	87.5	84.7	74.7	82.5	78.6	6.8
AOT-L [67]	89.7	92.3	91.0	80.3	85.7	83.0	75.3	82.3	78.8	15.2
STCN* [6]	90.1	92.2	91.1	81.5	87.9	84.7	72.7	79.6	76.1	11.7
GSFM (Ours)	**90.1**	**92.7**	**91.4**	**83.1**	**89.3**	**86.2**	**74.0**	**80.9**	**77.5**	8.9

key encoders and ResNet18 for value encoder. Bootstrapped cross-entropy loss (hard example mining) is used for mask segmentation. Binary cross-entropy loss and Dice loss are used for boundary prediction. The weight of boundary prediction loss is 0.05. For inference, we adopt top-k filtering [6,5] in our experiment with $k = 50$ in default. We memorize every 3 frame, and no previous temporary frame is used. Unless otherwise specified, we utilize the DAVIS2017 val set for experiment analysis.

5 Experiments

5.1 Comparisons with State-of-the-Art Methods

DAVIS 2016 [36] is a densely annotated video object segmentation benchmark which contains 20 high-quality annotated video sequences. We compare GSFM with state-of-the-art methods in Table 1. Since the scenarios in this dataset are relatively simple and only focus on a single target object, the segmentation results of most of the methods are excellent. Based on the STCN [6], our method

Table 2: Evaluation on YouTube-VOS 2018 validation set. Seen and Unseen denote the presence or absence of these categories in the training set, respectively. $\mathcal{G}$ is the averaged score of all $\mathcal{J}$ and $\mathcal{F}$.

Methods	Seen		Unseen		$\mathcal{G}$
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	
OnAVOS [48]	60.1	62.7	46.6	51.4	55.2
PReMVOS [28]	71.4	75.9	56.5	63.7	66.9
STM [34]	79.7	84.2	72.8	80.9	79.4
AFB-URR [25]	78.8	83.1	74.1	82.6	79.6
GCM [23]	72.6	75.6	68.9	75.7	73.2
KMN [39]	81.4	85.6	75.3	83.3	81.4
G-FRTM [35]	68.6	71.3	58.4	64.5	65.7
SwiftNet [50]	77.8	81.8	72.3	79.5	77.8
SSTVOS [11]	80.9	-	76.6	-	81.8
RMNet [58]	82.1	85.7	75.7	82.4	81.5
LCM [15]	82.2	86.7	75.7	83.4	82.0
MiVOS [5]	80.0	84.6	74.8	82.4	80.4
JOINT [30]	81.5	85.9	78.7	86.5	83.1
HMMN [40]	82.1	87.0	76.8	84.6	82.6
RPCMVOS [61]	83.1	87.7	78.5	86.7	84.0
DMN-AOA [24]	82.5	86.9	76.2	84.2	82.5
AOT-L [67]	82.5	87.5	77.9	86.7	83.7
STCN* [6]	81.8	86.4	77.8	85.6	82.9
GSFM (Ours)	**82.8**	**87.5**	**78.3**	**86.5**	**83.8**

achieves the performance of 91.4 $\mathcal{J}\&\mathcal{F}$.

DAVIS 2017 [37] is a multiple objects benchmark. The validation set contains 59 objects in 30 videos. In the Table 1, GSFM achieves an average score of 86.2, which outperforms baseline methods by 1.5 $\mathcal{J}\&\mathcal{F}$. What's more, we also test our model on the more challenging DAVIS 2017 test-dev split set. It also significantly surpasses the baseline method (1.4 $\mathcal{J}\&\mathcal{F}$).

YouTube-VOS [60] is the largest benchmark available for video object segmentation. It contains 3471 videos in the training set (65 categories), 507 videos in the valid set (additional 26 categories not in the training set), and 541 videos in the test set. As shown in Table 2, our method achieves competitive results (83.8) on YouTube-VOS and outperforms the baseline methods by 0.9 $\mathcal{J}\&\mathcal{F}$.

Qualitative Results. Fig. 5 shows some comparison examples between ours GSFM and STCN [6]. In the first example, similar pixels of the dogs are easily mis-segmented by STCN because only local information is used for matching. While with LFM enhancing global semantic information, GSFM can identify

Fig. 5: Visualization results of our proposed method. (a) denotes the segmentation results of our baseline method [6]. (b) is the results of our GSFM. The first example shows that our model can better perceive the overall semantic information of the target and thus identify similar objects. The second example shows that our approach makes a better determination of the ambiguous areas

targets more robustly. This is also illustrated in Fig. 1. The second example shows that with HFM enhancing fine-grained information, the proposed model has a better judgment for details and ambiguous areas.

5.2 Ablation Study

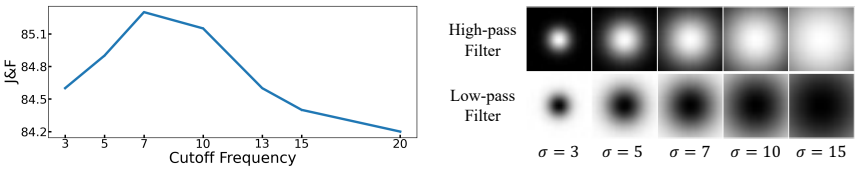
Table 3: Enhancing different frequency. freq_L, freq_H, and freq_F denotes enhancing low, high, and full-frequency, respectively

LFM	HFM	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
freq_F	freq_H	81.6	88.0	$84.8^{\downarrow 0.5}$
freq_H	freq_H	80.8	87.6	$84.2^{\downarrow 1.1}$
Attn.	freq_H	81.6	88.2	$84.9^{\downarrow 0.4}$
freq_L	freq_F	81.4	87.9	$84.7^{\downarrow 0.6}$
freq_L	freq_L	81.2	87.7	$84.5^{\downarrow 0.8}$
freq_L	**freq_H**	**81.9**	**88.7**	**85.3**

Table 4: The quantitative results of generalization effect. FM denotes the proposed LFM and HFM and ✓ indicates deployed

Method	FM	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
STM [34]		78.8	84.2	81.5
	✓	80.8	86.2	**83.5**$^{\uparrow}$
KMN [39]		79.7	85.5	82.6
	✓	81.6	87.8	**84.7**$^{\uparrow}$
MiVOS [5]		79.8	85.6	82.7
	✓	81.7	87.4	**84.6**$^{\uparrow}$

Analysis on LFM and HFM. In addition to the observation in some theoretical works [51], we conduct experiments to verify the rationality of enhancing low-frequency in encoder and high-frequency in decoder. The results are shown in Table 3. Note that enhancing full frequency is different from removing the



(a) Quantitative results (b) Visualization of filter kernels

Fig. 6: Analysis on the selection of frequency filters

module since it still updates the features in the spectral domain. From the table we can see that, when the high-frequency components are enhanced in the encoder, there is a significant decrease on performance (1.1 $\mathcal{J}\&\mathcal{F}$), which illustrates the encoded features need contain enough high-level semantic information. Conversely, decoder features need have fine-grained detail information. Besides, we have also tried other strategy that fusing global information, *e.g.*, attention, and LFM works better.

Generalizability Analysis. To demonstrate the generalization ability of our frequency modules and prove that the lack of intra-frame global dependency is a common problem of memory network-based methods, we conduct experiments by applying our modules on some other methods as well. As shown in Table 4, the effectiveness of these methods is significantly improved by adding the frequency modules, which further shows the rationality of enhancing different frequency components separately in different parts of the network.

Selection of Frequency Filters. When performing frequency enhancement, we need to choose the cutoff frequency σ that distinguishes high and low frequency (the value of the cutoff frequency affects the frequency filter). After visualizing and experimenting with Gaussian filter kernels of different cutoff frequencies, finally, we choose $\sigma = 7$ as the cutoff frequency in default. Fig. 6(a) shows that too large or too small cutoff frequency will have a bad effect. From Fig. 6(b) we can see that if σ is set too large, the high-pass filter will pass almost all frequencies while the low-pass filter will filter out all frequencies, which losses the function of selective enhancement. Same thing if σ is set too small.

Component Analysis. We experiment with the effectiveness of the proposed LFM, HFM, and Boundary Decoder. As shown in Table 5, all of them bring performance improvement and their combination works better (upgraded 1.2).

Effect of Small Objects. Although the LFM takes a residual structure to enhance low-frequency components during encoding, it does not result in information loss. To prove that, we analyze the segmentation effect of small objects on YouTubeVOS dataset. Fig. 7 and Table 6 show the qualitative results and quantitative results respectively. In Table 6, we count the results for objects with

Table 5: Ablation study of the proposed modules

LFM	HFM	Boundary Branch	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	FPS
			80.8	87.4	84.1	11.7
✓			81.5	87.9	**84.7**$^{\uparrow 0.6}$	11.3
	✓		81.2	87.8	**84.5**$^{\uparrow 0.4}$	10.9
✓		✓	81.8	88.2	**85.0**$^{\uparrow 0.9}$	9.5
	✓	✓	81.4	87.7	**84.6**$^{\uparrow 0.5}$	9.1
✓	✓	✓	81.9	88.7	**85.3**$^{\uparrow 1.2}$	8.9

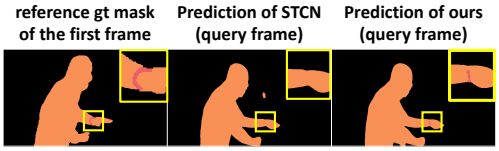


Fig. 7: Qualitative results on small objects.

Area	5%	1%	0.5%
STCN	80.6	76.3	73.5
Ours	81.4	78.1	75.0

Table 6: Quantitative results on small objects.

area less than 5%, 1% and 0.5% of the image. It can be seen that the segmentation results of small objects are not worse, but better due to the enhanced discrimination of features.

6 Conclusions

To fully utilize the intra-frame spatial dependency, we propose a Global Spectral Filter Memory network (GSFM) for semi-supervised video object segmentation in this paper. According to the different characteristics of encoding and decoding, GSFM separately enhances corresponding frequency components. With LFM integrating high-level semantic information and HFM highlighting fine-grained details, GSFM shows excellent performance on the popular DAVIS [36,37] and YouTube-VOS [60]. Besides, extensive experiments also demonstrate the rationality and generalization ability of our frequency modules. We hope that the strategy enhancing low-frequency for encoding and high-frequency for decoding would inspire some research in related fields.

Acknowledgement.

This work was partially supported by the National Natural Science Foundation of China under Grant No. U1903213 and the Shenzhen Key Laboratory of Marine IntelliSense and Computation (NO. ZDSYS20200811142605016.)

References

1. Bao, L., Wu, B., Liu, W.: CNN in MRF: video object segmentation via inference in a cnn-based higher-order spatio-temporal MRF. In: CVPR. pp. 5977–5986 (2018) 3
2. Bergland, G.D.: A guided tour of the fast fourier transform. IEEE spectrum **6**(7), 41–52 (1969) 6
3. Caelles, S., Maninis, K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Gool, L.V.: One-shot video object segmentation. In: CVPR. pp. 5320–5329 (2017) 3
4. Cheng, H.K., Chung, J., Tai, Y., Tang, C.: Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In: CVPR. pp. 8887–8896 (2020) 9
5. Cheng, H.K., Tai, Y., Tang, C.: Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In: CVPR. pp. 5559–5568 (2021) 2, 4, 9, 10, 11, 12
6. Cheng, H.K., Tai, Y., Tang, C.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: NIPS. pp. 11781–11794 (2021) 2, 4, 5, 9, 10, 11, 12
7. Cheng, J., Tsai, Y., Hung, W., Wang, S., Yang, M.: Fast and accurate online video object segmentation via tracking parts. In: CVPR. pp. 7415–7424 (2018) 3
8. Cheng, J., Tsai, Y., Wang, S., Yang, M.: Segflow: Joint learning for video object segmentation and optical flow. In: ICCV. pp. 686–695 (2017) 3
9. Cheng, T., Wang, X., Huang, L., Liu, W.: Boundary-preserving mask R-CNN. In: ECCV (2020) 6, 8
10. Chi, L., Jiang, B., Mu, Y.: Fast fourier convolution. In: NIPS (2020) 4
11. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: Sstvos: Sparse spatiotemporal transformers for video object segmentation. In: CVPR. pp. 5912–5921 (2021) 10, 11
12. Ge, W., Lu, X., Shen, J.: Video object segmentation using global and instance embedding learning. In: CVPR. pp. 16836–16845 (2021) 10
13. Han, J., Yang, L., Zhang, D., Chang, X., Liang, X.: Reinforcement cutting-agent learning for video object segmentation. In: CVPR. pp. 9080–9089 (2018) 3
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 9
15. Hu, L., Zhang, P., Zhang, B., Pan, P., Xu, Y., Jin, R.: Learning position and target consistency for memory-based video object segmentation. arXiv preprint arXiv:2104.04329 (2021) 2, 4, 10, 11
16. Hu, Y., Huang, J., Schwing, A.G.: Maskrnn: Instance level video object segmentation. In: NIPS. pp. 325–334 (2017) 3
17. Huang, X., Xu, J., Tai, Y., Tang, C.: Fast video object segmentation with temporal aggregation network and dynamic template matching. In: CVPR. pp. 8876–8886 (2020) 4
18. Jang, W., Kim, C.: Online video object segmentation via convolutional trident network. In: CVPR. pp. 7474–7483 (2017) 3
19. Katznelson, Y.: An introduction to harmonic analysis. Cambridge University Press (2004) 2, 7
20. Li, X., Wei, T., Chen, Y.P., Tai, Y., Tang, C.: FSS-1000: A 1000-class dataset for few-shot segmentation. In: CVPR. pp. 2866–2875 (2020) 9
21. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: ECCV. pp. 93–110 (2018) 3

22. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: ECCV. pp. 93–110 (2018) 4
23. Li, Y., Shen, Z., Shan, Y.: Fast video object segmentation using the global context module. In: ECCV. pp. 735–750 (2020) 2, 10, 11
24. Liang, S., Shen, X., Huang, J., Hua, X.S.: Video object segmentation with dynamic memory networks and adaptive object alignment. In: ICCV. pp. 8065–8074 (2021) 10, 11
25. Liang, Y., Li, X., Jafari, N.H., Chen, J.: Video object segmentation with adaptive feature bank and uncertain-region refinement. In: NIPS (2020) 2, 4, 11
26. Lin, S., Yang, L., Saleemi, I., Sengupta, S.: Robust high-resolution video matting with temporal guidance. In: WACV (2022) 6
27. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Gool, L.V.: Video object segmentation with episodic graph memory networks. In: ECCV. pp. 661–679 (2020) 4
28. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: ACCV. pp. 565–580 (2018) 4, 11
29. Maninis, K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Gool, L.V.: Video object segmentation without temporal information. TPAMI **41**(6), 1515–1530 (2019) 3
30. Mao, Y., Wang, N., Zhou, W., Li, H.: Joint inductive and transductive learning for video object segmentation. arXiv preprint arXiv:2108.03679 (2021) 10, 11
31. Milletari, F., Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016 (2016) 8
32. Ngan, K.N., Li, H.: Video segmentation and its applications. Springer Science & Business Media (2011) 1
33. Oh, S.W., Lee, J., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: CVPR. pp. 7376–7385 (2018) 9, 10
34. Oh, S.W., Lee, J., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV. pp. 9225–9234 (2019) 2, 4, 10, 11, 12
35. Park, H., Yoo, J., Jeong, S., Venkatesh, G., Kwak, N.: Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In: CVPR. pp. 8405–8414 (2021) 10, 11
36. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M.H., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. pp. 724–732 (2016) 1, 3, 10, 14
37. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelaez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 DAVIS challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) 1, 3, 11, 14
38. Qin, Z., Zhang, P., Wu, F., Li, X.: Fcanet: Frequency channel attention networks. arXiv preprint arXiv:2012.11879 (2020) 4
39. Seong, H., Hyun, J., Kim, E.: Kernelized memory network for video object segmentation. In: ECCV. pp. 629–645 (2020) 2, 4, 10, 11, 12
40. Seong, H., Oh, S.W., Lee, J., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. arXiv preprint arXiv:2109.11404 (2021) 2, 10, 11
41. Shen, X., Yang, J., Wei, C., Deng, B., Huang, J., Hua, X., Cheng, X., Liang, K.: Dct-mask: Discrete cosine transform mask representation for instance segmentation. In: CVPR (2021) 4
42. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended CSSD. TPAMI pp. 717–729 (2016) 9

43. Suh, S., Park, Y., Ko, K., Yang, S., Ahn, J., Shin, J., Kim, S.: Weighted mask R-CNN for improving adjacent boundary segmentation. J. Sensors **2021**, 8872947:1–8872947:8 (2021) 6
44. Sun, M., Xiao, J., Lim, E.G., Zhang, B., Zhao, Y.: Fast template matching and update for video object tracking and segmentation. In: CVPR. pp. 10788–10796 (2020) 4
45. Tang, Y., Jiang, Z., Xie, Z., Cao, Y., Zhang, Z., Torr, P.H.S., Hu, H.: Breaking shortcut: Exploring fully convolutional cycle-consistency for video correspondence learning. arXiv preprint arXiv:2105.05838 (2021) 2
46. Tsai, Y., Yang, M., Black, M.J.: Video segmentation via object flow. In: CVPR. pp. 3899–3908 (2016) 3
47. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.: FEELVOS: fast end-to-end embedding learning for video object segmentation. In: CVPR. pp. 9481–9490 (2019) 4, 10
48. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017) 3, 11
49. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017) 3
50. Wang, H., Jiang, X., Ren, H., Hu, Y., Bai, S.: Swiftnet: Real-time video object segmentation. In: CVPR. pp. 1296–1305 (2021) 2, 4, 10, 11
51. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: CVPR (2020) 3, 4, 6, 12
52. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR. pp. 3796–3805 (2017) 9
53. Wang, Z., Xu, J., Liu, L., Zhu, F., Shao, L.: Ranet: Ranking attention network for fast video object segmentation. In: ICCV. pp. 3977–3986 (2019) 10
54. Wen, P., Yang, R., Xu, Q., Qian, C., Huang, Q., Cong, R., Si, J.: DMVOS: discriminative matching for real-time video object segmentation. In: ACMMM. pp. 2048–2056 (2020) 10
55. Woo, S., Park, J., Lee, J., Kweon, I.S.: CBAM: convolutional block attention module. In: ECCV (2018) 9
56. Xiao, H., Feng, J., Lin, G., Liu, Y., Zhang, M.: Monet: Deep motion exploitation for video object segmentation. In: CVPR. pp. 1140–1148 (2018) 3
57. Xiao, M., Zheng, S., Liu, C., Wang, Y., He, D., Ke, G., Bian, J., Lin, Z., Liu, T.: Invertible image rescaling. In: ECCV (2020) 4
58. Xie, H., Yao, H., Zhou, S., Zhang, S., Sun, W.: Efficient regional memory network for video object segmentation. arXiv preprint arXiv:2103.12934 (2021) 2, 4, 10, 11
59. Xie, S., Tu, Z.: Holistically-nested edge detection. Int. J. Comput. Vis. **125**(1-3), 3–18 (2017) 8
60. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.S.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) 1, 3, 11, 14
61. Xu, X., Wang, J., Li, X., Lu, Y.: Reliable propagation-correction modulation for video object segmentation. In: AAAI. pp. 2946–2954 (2022) 2, 10, 11
62. Xu, Y., Fu, T., Yang, H., Lee, C.: Dynamic video segmentation network. In: CVPR. pp. 6556–6565 (2018) 3
63. Xu, Z.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: ICONIP (2019) 3, 4, 6
64. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR. pp. 6499–6507 (2018) 4

65. Yang, Z., Tang, Y., Bertinetto, L., Zhao, H., Torr, P.H.S.: Hierarchical interaction network for video object segmentation from referring expressions. In: BMVC. p. 254 (2021) 1
66. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: ECCV. pp. 332–348 (2020) 4, 10
67. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. arXiv preprint arXiv:2106.02638 (2021) 4, 9, 10, 11
68. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by multi-scale foreground-background integration. IEEE TPAMI (2021) 4
69. Yin, D., Lopes, R.G., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. In: NIPS (2019) 3, 4, 6
70. Yu, Z., Feng, C., Liu, M., Ramalingam, S.: Casenet: Deep category-aware semantic edge detection. In: CVPR (2017) 8
71. Zeng, Y., Zhang, P., Lin, Z.L., Zhang, J., Lu, H.: Towards high-resolution salient object detection. In: ICCV. pp. 7233–7242 (2019) 9
72. Zhang, P., Hu, L., Zhang, B., Pan, P.: Spatial constrained memory network for semi-supervised video object segmentation. CVPR Workshops (2020) 4
73. Zhang, Z., Fidler, S., Urtasun, R.: Instance-level segmentation for autonomous driving with deep densely connected mrfs. In: CVPR (2016) 1
74. Zhao, K., Kang, J., Jung, J., Sohn, G.: Building extraction from satellite images using mask r-cnn with building boundary regularization. In: CVPR Workshops (2018) 6
75. Zhou, T., Li, J., Wang, S., Tao, R., Shen, J.: Matnet: Motion-attentive transition network for zero-shot video object segmentation. IEEE Trans. Image Process. pp. 8326–8338 (2020) 4