Holistic Correction with Object Prototype for Video Object Segmentation

Shengye Qiao[1 2], Changqun Xia[2*], Yanjie Liang[2], Gongjin Lan[3], Jia Li[1*]

[1]State Key Laboratory of Virtual Reality Technology and Systems, SCSE, Beihang University, Beijing, China
[2]Pengcheng Laboratory, Shenzhen, China
[3]Department of Computer Science and Technology, Southern University of Science and Technology, Shenzhen, China
{qiaoshy,jiali}@buaa.edu.cn, {xiachq,liangyj}@pcl.ac.cn, langj@sustech.edu.cn

Abstract

Recently, memory-based methods have achieved progress in semi-supervised video object segmentation. However, these methods still suffer from unstructured challenges, such as object transformations, occlusions and disappearance-reappearance. To this end, we propose a Holistic Correction Network (HCNet) to adaptively acquire concise object prototypes for holistic correction at semantic, spatial and temporal aspects. Specifically, an Adaptive Prototype Update module is firstly designed to construct multi-level core object representations by associating object variations in consecutive frames with segmentation quality assessment. Based on the updated object prototypes, Semantic, Spatial and Temporal Correction modules are respectively designed to enhance the object semantics in the entire frame, eliminate the incorrect semantic enhancement outside the object regions and calibrate the estimated object regions with temporal changes of objects. Through the holistic correction mechanism with effective object prototypes, our proposed HCNet can robustly and efficiently deal with diverse complex scenarios. Extensive and comprehensive experiments conducted on seven datasets demonstrate that our proposed HCNet can significantly improve the segmentation performance.

Introduction

Semi-supervised Video Object Segmentation (SVOS) is the task of accurately and efficiently segmenting the objects in the entire video sequence when the objects are specified with a precisely annotated mask in the first frame. This task can be widely applied in many fields such as video surveillance, embodied intelligence and autonomous driving (Liang et al. 2024; Li et al. 2023; He et al. 2024; Ma et al. 2023).

Existing SVOS methods are mainly matching-based and employ memory network to boost the segmentation performance, which continuously memorizes historical object-specific appearance cues to match and segment target objects in the current frame. The amelioration to the initial memory (Oh et al. 2019; Cheng, Tai, and Tang 2021), which stores all features of selected frames, can be mainly divided into two aspects: single-stage compressed memory (Li et al. 2022; Lin et al. 2022) and multi-stage memory (Cheng and Schwing 2022; Cheng et al. 2024). However, the compressed memory only contain frame-level semantic information, leading to unsatisfactory performance on object segmentation which requires adequate object-level semantic and spatial information. Meanwhile, the object memory proposed in the multi-stage memory is overly dependent on the predicted mask, easily leading to the accumulation of errors. Moreover, the update of object memory lacks temporal locality which is less effective to segment videos with larger temporal resolution. For example, the videos in emerging complex dataset VOST (Tokmakov, Li, and Gaidon 2023) are only labeled at 5 fps and have longer time-interval between consecutive frames than the common datasets. Therefore, a question arises: instead of unstable object memory, can we construct concise and robust object prototypes which simultaneously take into account high-level semantic information, low-level spatial information and temporal locality?

In addition to utilizing appearance cues with memory network, recent methods extract motion cues to localize objects. For example, BATMAN (Yu et al. 2022) acquire motion cues with optical flow to pay more attention to objects when matching with appearance cues. Despite the improvement in

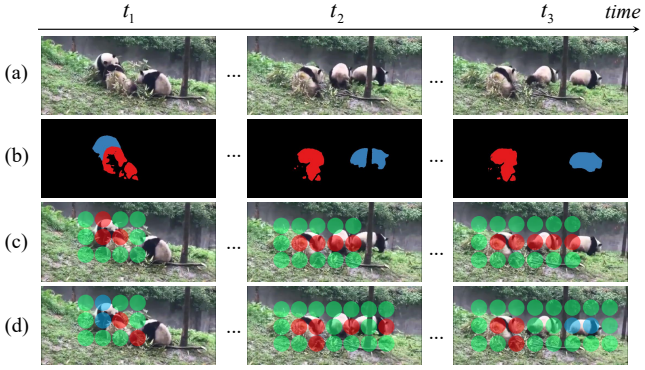

Figure 1: Comparison of segmentation results with visualization (we superimpose predicted masks on the input frame in the form of circles): (a) input frames, (b) groundtruth masks, (c) outputs of RAVOS (Miao et al. 2024), (d) outputs of Ours. Green, blue and red circles denote the background and two objects in the predicted mask. The regions covered with circles are the ones involved in the matching.

*Correspondence should be addressed to Changqun Xia (E-mail: xiachq@pcl.ac.cn) and Jia Li (E-mail: jiali@buaa.edu.cn).
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

accuracy, the efficiency is significantly reduced due to the computational complexity of optical flow itself. In contrast, RAVOS (Miao et al. 2024) constructs a motion estimator with a simple structure to track objects. Although it greatly reduces the computational burden introduced to obtain motion cues, it becomes fragile when facing complex scenarios as the estimated region based on the motion of objects cannot contain complete target objects causing loss of appearance cues. To exploit both appearance and motion cues for robust and efficient segmentation, we further delve into the errors occurred in segmentation which can be divided into semantic, spatial and temporal aspects as shown in Fig. 1. 1) incorrect semantic discrimination (the back of the panda corresponding to the red mask which is mixed with the cluttered background is not accurately identified at t_1, t_2, t_3 moments); 2) insufficient spatial localization (the distractor panda with similar appearance is regarded as one of target objects at t_2, t_3 moments); 3) incomplete temporal prediction (the prediction region cannot contain complete objects when encountered fast motion at t_2, t_3 moments). Therefore, there is an urgent need to holistically correct these errors with object prototypes at semantic, spatial and temporal aspects to achieve robust and efficient segmentation.

With the above consideration in mind, we propose a novel Holistic Correction Network (HCNet) to correct the above errors with concise object prototypes for robust and efficient video object segmentation. Specifically, we first construct an Adaptive Prototype Update module to retrieve object prototypical representations by evaluating the quality of current segmentation results before associating object variations in consecutive frames. Then we construct a Semantic Correction module to distinguish the semantic nuances between background and target objects. Meanwhile, we propose a Spatial Correction module to adaptively reduce the interference of similar objects and filter out incorrect semantic enhancements based on the corrected regions of objects. Furthermore, we design a Temporal Correction module to correct the estimated regions for containing all complete target objects, which can improve the robustness of segmentation and reduce the computational burden in the whole segmentation process. As shown in Fig. 2 (d), our proposed method can effectively improve the performance with the object prototypes and holistic correction mechanism. Extensive experiments comprehensively and sufficiently evaluated on seven SVOS datasets demonstrate our proposed HCNet can achieve significant improvements in accuracy and efficiency. In general, the contribution of our work can be summarized as follows:

- We propose a novel Holistic Correction Network (HCNet) for robust and efficient video object segmentation. The holistic correction mechanism with object prototypes can effectively address unstructured challenges such as object transformations, occlusions, similar distractors and fast motion.
- We construct an Adaptive Prototype Update module to generate object prototypes, and Semantic, Spatial, Temporal Correction modules are respectively constructed to holistically reduce redundant and erroneous matching.
- Extensive experiment results show that our proposed HCNet can achieve higher accuracy and efficiency when comprehensively evaluated on DAVIS, YouTube-VOS, MOSE, VOST and LVOS datasets.

Related Work

Memory-based Video Object Segmentation

Recently, memory-based methods (Oh et al. 2019; Cheng, Tai, and Tang 2021; Cheng and Schwing 2022; Cheng et al. 2024) employing memory network to draw on historical frames can greatly improve the accuracy. For example, STM (Oh et al. 2019) and STCN (Cheng, Tai, and Tang 2021) design general architectures that effectively combines with memory network, XMem (Cheng and Schwing 2022) construct more sophisticated multi-stage memory stores to make full use of memory network. ISVOS (Wang et al. 2023) and Cutie (Cheng et al. 2024) achieve progress by placing additional emphasis on object-level matching. Although these memory-based methods can achieve increasingly higher accuracy evaluated on benchmark datasets, they only use appearance cues and ignores motion cues and their segmentation efficiency is relatively low which cannot meet the requirements of practical applications.

Motion-aware Video Object Segmentation

There are motion-aware SVOS methods (Miao et al. 2024; Yu, Xia, and Li 2024) to exploit both appearance and motion cues. For example, RAVOS (Miao et al. 2024) first constructs a motion estimator with a simple structure to extract motion without affecting the overall efficiency, then it constructs regional memory to effectively utilize appearance cues for matching and segmentation. Although RAVOS can effectively improve the efficiency, this way of estimating coarse motion cues is very fragile and leads to severe performance degradation when faced with complex scenarios.

In this work, we deal with matching errors with the proposed holistic correction mechanism at semantic, spatial and temporal aspects for robust semi-supervised video object segmentation. Our proposed HCNet can achieve more stable and high-speed segmentation performance evaluated on common, complex and long-video datasets.

Methodology

Framework Overview

The overview of the proposed Holistic Correction Network is shown in Fig. 2. Given the t-th full video frame $I_t \in \mathbb{R}^{3\times H\times W}$, it is firstly cropped with the predicted bounding box $B_t = [x_{c,t}, y_{c,t}, h_t, w_t]$ to obtain the region input $R_t \in \mathbb{R}^{3\times h_t\times w_t}$, where $(x_{c,t}, y_{c,t}), h_t, w_t$ denote the center coordinate, height and width of predicted region respectively. Then the region R_t is encoded to generate features $\mathbf{F}_t^i \in \mathbb{R}^{C^i\times h_t^i\times w_t^i}$ where $i \in \{1,2,3,4\}$ denotes the i-th encoding stage, and the fourth-stage encoded feature $\mathbf{F}_t^4$ is sent into regional matching (Miao et al. 2024) with regional memory to obtain the initial object-specific features $\mathbf{F}_{t,reg}^4$. In parallel, the object prototypes $\mathbf{O}_t$ are updated with the proposed Adaptive Prototype Update module based

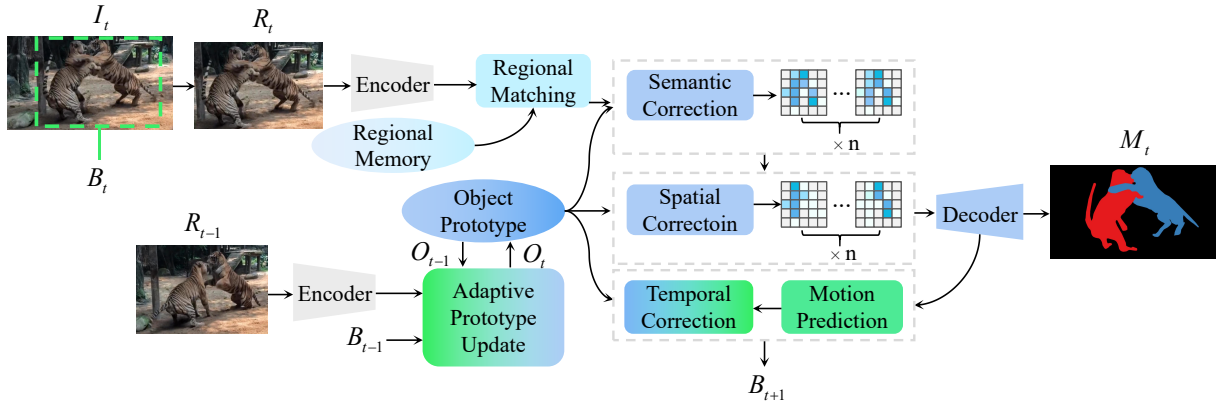

Figure 2: Overview of the proposed Holistic Correction Network. The frame I_t is firstly cropped based on the predicted bounding box B_t, encoded for regional matching, then sent into semantic and spatial correction modules, and decoded to generate the mask M_t. In parallel, the object prototypes $\mathbf{O}_t$ are updated with the Adaptive Prototype Update module. Finally, the predicted results are sent into the motion prediction and temporal correction modules to generate the final predicted bounding box B_{t+1}.

on the previous object prototypes $\mathbf{O}_{t-1}$, encoded features $\mathbf{F}^i_{t-1}$ of the last regional input R_{t-1} and predicted regions $B^j_{t-1}, j \in \{1, ..., n\}$ of n objects.

Afterwards, the multi-scale features $\{\mathbf{F}^2_t, \mathbf{F}^3_t, \mathbf{F}^4_{t,reg}\}$ with the updated object prototypes $\mathbf{O}_t$ are sent into semantic correction module to generate the semantic corrected features $\{\mathbf{F}^2_{t,sem}, \mathbf{F}^3_{t,sem}, \mathbf{F}^4_{t,sem}\}$. Then the semantic corrected features are used in spatial correction module with the updated object prototypes to identify the location of each object, producing the corresponding spatial corrected features $\{\mathbf{F}^2_{t,spa}, \mathbf{F}^3_{t,spa}, \mathbf{F}^4_{t,spa}\}$. After that, these spatial corrected features are sent into the decoder to generate the predicted mask M_t. Moreover, we perform motion prediction based on the locations of objects in the predicted mask to generate the initial predicted regions of each object. Then these initial predicted regions are sent into temporal correction module to generate each corrected region $B^j_{t+1} = [x^j_{c,t+1}, y^j_{c,t+1}, h^j_{t+1}, w^j_{t+1}], j \in \{1, ..., n\}$ of n objects. Finally we calculate the minimum union of each corrected region B^j_{t+1} to obtain B_{t+1} to crop the next frame I_{t+1}.

Adaptive Prototype Update

We generate the concise object prototypes with the proposed Adaptive Prototype Update module as shown in Fig. 3. Compared to existing object memory (Wang et al. 2023; Cheng et al. 2024), the object prototypes can be updated adaptively based on the complexity of video and the quality of segmentation, which is effective to adjust the semantics of prototypes according to the dynamic changes of target objects. Besides, benefit from the small size, the multi-scale object prototypes can be used to efficiently achieve multi-scale calculations in the holistic correction process.

We initialize the prototype of each object with the given frame I_0 and corresponding mask M_0 by first extracting each single-object features $\mathbf{F}^{i,j}_0 \in \mathbb{R}^{C^i \times h^{i,j}_0 \times w^{i,j}_0}$ from the encoded feature $\mathbf{F}^i_0 \in \mathbb{R}^{C^i \times h^i_0 \times w^i_0}$ based on the precise location of each object, where $i \in \{2, 3, 4\}$ denotes the i-th encoding stage and $j \in \{1, ..., n\}$ denotes the j-th object among n objects. Then we downsample the features $\mathbf{F}^{i,j}_0$ until their height and width are both smaller than r^i which is the prototype size, and we fill the empty position with feature interpolation to obtain the initial prototypes $\mathbf{O}^{i,j}_0 \in \mathbb{R}^{C^i \times r^i \times r^i}$. To update the prototypes, the encoded features are also extracted based on the predicted bounding box of each object to produce the features $\mathbf{F}^{i,j}_{t-1} \in \mathbb{R}^{C^i \times h^{i,j}_{t-1} \times w^{i,j}_{t-1}}$, which are then scaled and padded with zero vectors to obtain current object features $\mathbf{F}^{i,j}_{t-1,obj} \in \mathbb{R}^{C^i \times r^i \times r^i}$. Meanwhile, we record the position of padding to construct masks $\mathbf{M}^{i,j}_{t-1} \in \mathbb{R}^{r^i \times r^i}$ where padding positions are assigned the value of negative infinity and others are assigned the value of 0, thereby avoid these padding features from participating in subsequent calculations. As we perform the same operations for each object j of each encoding stage i, we omit the superscripts i, j for the subsequent clearer descriptions.

Then we estimate the aggregation weight $\hat{\sigma}_{t-1}$ by firstly performing cross-correlation (Bertinetto et al. 2016) between the previous prototypes $\mathbf{O}_{t-1}$ and the encoded features $\mathbf{F}_{t-1}$ extracted based on the predicted region as:

$$\mathbf{C}_{t-1} = \mathbf{O}_{t-1} \star \mathbf{F}_{t-1}, \tag{1}$$

where $\star$ denotes the cross-correlation operation and the output correlation map $\mathbf{C}_{t-1}$ only has a single channel, which is then used to calculate the mean and standard deviation as:

$$\mu_{t-1} = \frac{1}{m}\sum_{p=1}^{m} x_p, \sigma_{t-1} = \sqrt{\frac{1}{m}\sum_{p=1}^{m}(x_p - \mu_{t-1})^2}, \tag{2}$$

where m and x_p denote the total number of pixels and each value of the correlation map $\mathbf{C}_{t-1}$ respectively. And we further normalize the standard deviation σ_{t-1} as:

$$\hat{\sigma}_{t-1} = \arctan(\sigma_{t-1}) * \frac{2}{\pi}, \hat{\sigma}_{t-1} \in [0, 1) \tag{3}$$

The $\hat{\sigma}_{t-1}$ is close to 0 indicating that there are almost no related features of the target object in the predicted region,

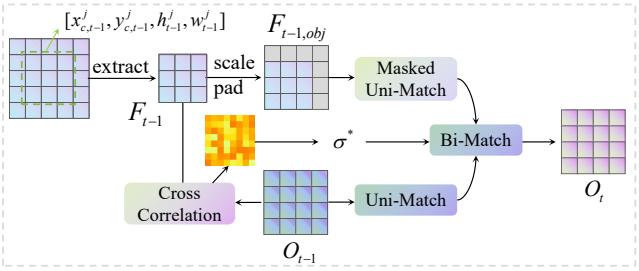

Figure 3: Illustration of the proposed Adaptive Prototype Update module. $[x^j_{c,t-1}, y^j_{c,t-1}, h^j_{t-1}, w^j_{t-1}]$ denotes the predicted bounding box of the j-th object.

which can be used to adaptively aggregate current features and previous prototypes.

Subsequently, we perform uni-matching on the previous prototypes $\mathbf{O}_{t-1}$ and masked uni-matching on the current features $\mathbf{F}_{t-1,obj}$ as:

$$\mathbf{O}^s_{t-1} = \text{Softmax}(\frac{\mathbf{O}_{t-1}\mathbf{O}^{tr}_{t-1}}{\sqrt{C}})\mathbf{O}_{t-1}, \tag{4}$$

$$\mathbf{F}^s_{t-1} = \text{Softmax}(\frac{\mathbf{F}_{t-1,obj}\mathbf{F}^{tr}_{t-1,obj}}{\sqrt{C}} + \mathbf{M}_{t-1})\mathbf{F}_{t-1,obj}, \tag{5}$$

where tr denotes the transpose of feature. Then we perform the bidirectional matching with the residual path on the outputs $\mathbf{O}^s_{t-1}$ and $\mathbf{F}^s_{t-1}$ as:

$$\mathbf{O}^c_{t-1} = \text{Softmax}(\frac{\mathbf{O}^s_{t-1}\mathbf{F}^{s,tr}_{t-1}}{\sqrt{C}} + \mathbf{M}_{t-1})\mathbf{F}^s_{t-1} + \mathbf{O}^s_{t-1}, \tag{6}$$

$$\mathbf{F}^c_{t-1} = \text{Softmax}(\frac{\mathbf{F}^s_{t-1}\mathbf{O}^{s,tr}_{t-1}}{\sqrt{C}})\mathbf{O}^s_{t-1} + \mathbf{F}^s_{t-1}, \tag{7}$$

then we modulate the outputs $\mathbf{F}^c_{t-1}$ and $\mathbf{O}^c_{t-1}$ with the aggregation factor $\hat{\sigma}_{t-1} \in [0, 1)$ generated in Eq. (3) to obtain the current prototypes $\mathbf{O}_t$ as:

$$\mathbf{O}_t = \hat{\sigma}_{t-1}\mathbf{F}^c_{t-1} + (1 - \hat{\sigma}_{t-1})\mathbf{O}^c_{t-1}, \tag{8}$$

the output prototypes $\mathbf{O}_t$ of target objects further involve in the following holistic correction process at semantic, spatial and temporal aspects.

Holistic Correction

After adaptively updating concise object prototypes which can simultaneously take into account high-level semantic information, low-level spatial information and temporal locality, we delicately design Semantic, Spatial and Temporal Correction modules for holistic correction with object prototypes by enhancing, filtering and calibrating appearance and motion cues.

Semantic Correction. After obtaining the feature $\mathbf{F}^4_{t,reg} \in \mathbb{R}^{C_v \times h^4_t \times w^4_t}$, we perform semantic correction to generate the semantic-corrected features $\{\mathbf{F}^2_{t,sem}, \mathbf{F}^3_{t,sem}, \mathbf{F}^4_{t,sem}\}$ as shown in Fig. 4 (a). As we deal with multiple objects independently in parallel as a batch following the same setting (Cheng and Schwing 2022; Cheng et al. 2024), we can effectively and efficiently correct each target object simultaneously. Specifically, for the j-th object, we first perform object matching between $\mathbf{F}^4_{t,reg}$ and $\mathbf{O}^{4,j}_t$, and concatenate the output and the feature $\mathbf{F}^4_{t,reg}$ on the channel dimension to generate the initial corrected features $\mathbf{F}^4_{t,temp} \in \mathbb{R}^{C^4 \times h^4_t \times w^4_t}$, then we perform channel-wise attention on the initial corrected features to generate the final semantic correction features $\mathbf{F}^4_{t,sem} \in \mathbb{R}^{C^4 \times h^4_t \times w^4_t}$, the overall process can be formulated as:

$$\mathbf{F}^4_{t,temp} = \text{Cat}(\text{Softmax}(\frac{\mathbf{O}^{4,j}_t \mathbf{F}^{4,tr}_{t,reg}}{\sqrt{C_v}})\mathbf{O}^4_t + \mathbf{F}^4_{t,reg}, \mathbf{F}^4_{t,reg}), \tag{9}$$

$$\mathbf{F}^4_{t,sem} = \text{Softmax}(\frac{\mathbf{F}^4_{t,temp}\mathbf{F}^{4,tr}_{t,temp}}{\sqrt{h^4_t w^4_t}})\mathbf{F}^4_{t,temp}. \tag{10}$$

Due to the small size r^i, for $i \in \{2, 3\}$ stages, we can still perform object matching between the original encoded features F^i_t and the updated object prototypes $\mathbf{O}^{i,j}_t$. Meanwhile, we use high-level semantics to guide these low-level features to correct the incorrect enhancements that are misled by similar low-level properties, such as textures and shapes. Through the object matching and semantic guidence, we can generate the final semantic-corrected features $\mathbf{F}^i_{t,sem} \in \mathbb{R}^{C^i \times h^i_t \times w^i_t}$, which can be formulated as:

$$\mathbf{F}^i_{t,temp} = \text{DC}(\text{Softmax}(\frac{\mathbf{O}^{i,j}_t \mathbf{F}^{i,tr}_t}{\sqrt{C^i}})\mathbf{O}^{i,j}_t + \mathbf{F}^i_t), \tag{11}$$

$$\mathbf{F}^i_{t,sem} = \text{Sig}(\text{CB}(\text{Up}(\mathbf{F}^{i+1}_{t,sem}))) \odot \mathbf{F}^i_{t,temp}, \tag{12}$$

where $i \in \{2, 3\}$, Sig denotes the sigmoid function, DC denotes 3×3 deformable convolutions, CB denotes 3×3 convolution and batchnorm layers, Up denotes the upsampling operation, $\odot$ denotes the hadamard product, and the output semantic-corrected features $\{\mathbf{F}^2_{t,sem}, \mathbf{F}^3_{t,sem}, \mathbf{F}^4_{t,sem}\}$ are then sent into the spatial correction process.

Spatial Correction. After obtaining the features $\{\mathbf{F}^2_{t,sem}, \mathbf{F}^3_{t,sem}, \mathbf{F}^4_{t,sem}\}$ from the semantic correction process, we further perform spatial correction on each feature $\mathbf{F}^i_{t,sem}$, the illustration of the spatial correction phase is shown in Fig. 4 (b). Specifically, we first extract the object-specific feature $\mathbf{F}^{i,j}_{t,sem} \in \mathbb{R}^{C^i \times h^{i,j}_t \times w^{i,j}_t}$ from $\mathbf{F}^i_{t,sem} \in \mathbb{R}^{C^i \times h^i_t \times w^i_t}$ based on the predicted object region $[x^j_{c,t}, y^j_{c,t}, h^j_t, w^j_t]$, and generate the local correlation map $\mathbf{C}^{i,j}_{t,loc} \in \mathbb{R}^{h^{i,j}_t \cdot w^{i,j}_t \times r^i \cdot r^i}$ between $\mathbf{F}^{i,j}_{t,sem} \in \mathbb{R}^{C^i \times h^{i,j}_t \cdot w^{i,j}_t}$ and the updated object prototypes $\mathbf{O}^{i,j}_t \in \mathbb{R}^{C^i \times r^i \cdot r^i}$, which can be formulated as:

$$\mathbf{C}^{i,j}_{t,loc} = \frac{\mathbf{F}^{i,j}_{t,sem}\mathbf{O}^{i,j}_t}{\sqrt{C^i}}, \tag{13}$$

where we omit the transpose superscript tr for clearer description. Subsequently, for each vector $\mathbf{c}[p], p \in \{1, ..., h^{i,j}_t \cdot w^{i,j}_t\}$ in the correlation map $\mathbf{C}^{i,j}_{t,loc}$, we select the most relevant value as:

$$c[p, q^*] = \max\{c[p, q], q \in \{1, ..., r^i \cdot r^i\}\}, \tag{14}$$

which are used to construct the local weight matrix $\mathbf{W}^{i,j}_t \in \mathbb{R}^{h^{i,j}_t \times w^{i,j}_t}$, and we further pad the local weight matrix with

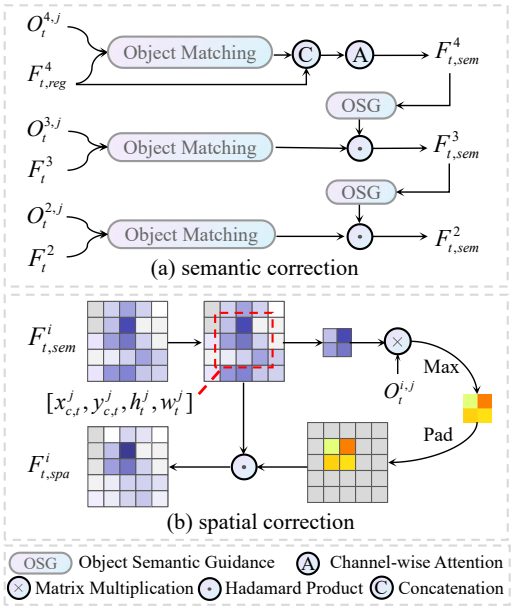

Figure 4: Illustration of (a) semantic correction and (b) spatial correction modules.

the smallest value of the local weight matrix to generate the global weight matrix $\mathbf{W}_t^i \in \mathbb{R}^{h_t^i \times w_t^i}$. Then we normalize the global weight matrix with the softmax function and expand it to have the same channel dimension as the feature $\mathbf{F}_{t,sem}^i$, and finally we perform element-wise multiplication between them to generate the spatial-corrected feature $\mathbf{F}_{t,spa}^i \in \mathbb{R}^{C^i \times h_t^i \times w_t^i}$, which can be formulated as:

$$\mathbf{F}_{t,spa}^i = \text{Expand}(\text{Softmax}(\mathbf{W}_t^i)) \odot \mathbf{F}_{t,sem}^i. \tag{15}$$

Then the output spatial self-corrected features $\{\mathbf{F}_{t,spa}^2, \mathbf{F}_{t,spa}^3, \mathbf{F}_{t,spa}^4\}$ are sent into the decoder which has the same architecture as (Cheng, Tai, and Tang 2021; Li et al. 2022) to generate the final predicted mask M_t.

Temporal Correction. After generating the predicted mask M_t of the frame I_t, we predict the locations of each object in the next frame I_{t+1} and perform the temporal correction on the initial predicted regions. We consider both semantic and spatial differences between the same object in consecutive frames, and design corresponding indicators which can evaluate the extent of object changes, thereby effectively correct the predicted location.

For the semantic difference, we evaluate the similarity between the current object features $\mathbf{F}_{t,obj}^{i,j} \in \mathbb{R}^{C^i \times r^i \times r^i}$ generated in the Adaptive Prototype Update module and the updated object prototypes $\mathbf{O}_t^{i,j} \in \mathbb{R}^{C^i \times r^i \times r^i}$. Specifically, we firstly perform the global average pooling (GAP) on these two object features and then calculate the cosine similarity between them, which can be formulated as:

$$s_{t,sem}^{i,j} = \frac{\text{GAP}(\mathbf{F}_{t,obj}^{i,j}) \cdot \text{GAP}(\mathbf{O}_t^{i,j})}{\|\text{GAP}(\mathbf{F}_{t,obj}^{i,j})\| \|\text{GAP}(\mathbf{O}_t^{i,j})\|}, s_{t,sem}^{i,j} \in [0,1], \tag{16}$$

where the similarity indicators $s_{t,sem}^{i,j}$ are further averaged to generate the final semantic difference indicator $s_{t,sem}^j$. $s_{t,sem}^j$ decreases when the j-th object itself is occluded, thus we have to expand the region range to deal with changes of objects under invisible conditions.

Meanwhile, we calculate the spatial difference between the location of each object in the predicted masks M_{t-1} and M_t of the frame I_{t-1} and I_t, denoted as P_{t-1}^j and P_t^j respectively. Formally, we calculate the IOU between these two predicted pixel-level position as:

$$s_{t,spa}^j = \frac{P_{t-1}^j \cap P_t^j}{P_{t-1}^j \cup P_t^j}, s_{t,spa}^j \in [0,1], \tag{17}$$

compared to calculating IOU based on the bounding box, the spatial difference indicator $s_{t,spa}^j$ can reflect the pixel-level changes in the position of each object, and $s_{t,spa}^j$ decreases when the motion of j-th object changes significantly which means that the range of predicted region has to be expanded to address fast motion.

Motion Prediction. We predict the regions of objects in the next frame based on the motion of objects, and correct the predicted regions with the indicators including semantic difference $s_{t,sem}^j$ and spatial difference $s_{t,spa}^j$.

Formally, we define the state space of j-th object motion as $(x_c^j, y_c^j, h^j, w^j, \dot{x}_c^j, \dot{y}_c^j, \dot{h}^j, \dot{w}^j)$ where $(x_c^j, y_c^j), h^j, w^j$ denote the center coordinate, height and width respectively, $\dot{x}_c^j, \dot{y}_c^j, \dot{h}^j, \dot{w}^j$ denote the respective velocities. The current regions of each object extracted from the predicted mask M_t is combined with the current motion state to predict the temporary regions $[x_{c,t+1}^j, y_{c,t+1}^j, \hat{h}_{t+1}^j, \hat{w}_{t+1}^j]$ of objects in the next frame with a standard Kalman Filter. Then we adjust the height and width of the predicted location as:

$$s_t^j = (1 - s_{t,sem}^j) \times (1 - s_{t,spa}^j), \tag{18}$$

$$\begin{cases} h_{t+1}^j = \min(H, \hat{h}_{t+1}^j + s_t^j \times H), \\ w_{t+1}^j = \min(W, \hat{w}_{t+1}^j + s_t^j \times W), \end{cases} \tag{19}$$

where H, W denote the original height and width of the input frame and h_{t+1}^j, w_{t+1}^j denote the corrected height and width. Then we can generate the final corrected bounding box $B_{t+1} = [x_{c,t+1}, y_{c,t+1}, h_{t+1}, w_{t+1}]$ by calculating the minimum union of each corrected region $[x_{c,t+1}^j, y_{c,t+1}^j, h_{t+1}^j, w_{t+1}^j]$ to crop the next frame I_{t+1}.

Experiments

Experimental Settings

Implementation Details. For our proposed Holistic Correction Network HCNet and its lite version denoted as HCNet-Lite, we both adopt ResNet-50 as the encoder, the same decoder with iterative upsampling architecture (Cheng, Tai, and Tang 2021), and the same training loss as existing methods (Yang and Yang 2022). Compared with HCNet, HCNet-Lite turns the multi-scale corrections at semantic, spatial and temporal aspects into dealing with only the fourth-stage encoding features, and thus there are only the fourth-stage features updated and stored as object prototypes. Without multi-scale calculations, the advantage of our proposed

Method	DAVIS 2017 test			DAVIS 2017 val			YouTube-VOS 2019 val					FPS
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{G}\uparrow$	$\mathcal{J}_s\uparrow$	$\mathcal{F}_s\uparrow$	$\mathcal{J}_u\uparrow$	$\mathcal{F}_u\uparrow$	
CFBI (Yang, Wei, and Yang 2020)	75.0	71.4	78.7	81.9	79.3	84.5	81.6	80.2	84.6	77.2	84.5	5.9
RMNet (Xie et al. 2021)	75.0	71.9	78.1	83.5	81.0	86.0	-	-	-	-	-	12
CFBI+ (Yang, Wei, and Yang 2021b)	75.6	71.6	79.6	82.9	90.1	85.7	82.9	80.6	85.2	78.9	86.8	5.6
STCN (Cheng, Tai, and Tang 2021)	76.1	72.7	79.6	85.4	82.2	88.6	82.7	81.1	85.4	78,2	85.9	20.2
AOT-R50 (Yang, Wei, and Yang 2021a)	79.6	75.9	83.3	84.9	82.3	87.5	84.1	83.5	88.1	78.4	86.3	18
BATMAN (Yu et al. 2022)	82.2	78.4	86.1	86.2	83.2	89.3	85.0	84.5	89.3	79.0	87.2	-
RDE (Li et al. 2022)	77.4	73.6	81.2	84.2	80.8	87.5	81.9	81.1	85.5	76.2	84.8	27
XMem (Cheng and Schwing 2022)	81.0	77.4	84.5	86.2	82.9	89.5	85.5	84.3	88.6	80.3	88.6	22.6
DeAOT-R50 (Yang and Yang 2022)	80.7	76.9	84.5	85.2	82.2	88.2	85.9	84.6	89.4	80.8	88.9	22.4
ISVOS (Wang et al. 2023)	82.8	79.3	86.2	87.1	83.7	90.5	86.1	85.2	89.7	80.7	88.9	-
RAVOS (Miao et al. 2024)	80.8	77.1	84.5	86.1	82.9	89.3	82.8	81.9	86.6	77.5	85.4	42
Cutie (Cheng et al. 2024)	84.2	80.6	87.7	88.8	85.4	92.3	86.1	85.5	90.0	80.6	88.3	45*
HCNet-Lite	84.4	81.0	87.8	89.2	85.9	92.5	86.9	86.5	90.2	81.8	88.9	**67**
HCNet	**84.7**	**81.3**	**88.1**	**89.7**	**86.6**	**92.8**	**87.4**	**86.8**	**90.9**	**82.5**	**89.4**	44

Table 1: Quantitative comparison with state-of-the-art SVOS methods on common datasets including DAVIS 2017 test, DAVIS 2017 val and YouTube-VOS 2019 val. We record the inference speed (FPS) on the same 3090 GPU device and $*$ denotes that we retest the speed.

Method	DAVIS 2016 val			YouTube-VOS 2018 val				
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{G}\uparrow$	$\mathcal{J}_s\uparrow$	$\mathcal{F}_s\uparrow$	$\mathcal{J}_u\uparrow$	$\mathcal{F}_u\uparrow$
CFBI	89.4	88.3	90.5	81.8	81.9	86.3	75.6	83.4
RMNet	88.8	88.9	88.7	81.5	82.1	85.7	75.7	82.4
CFBI+	89.9	88.7	91.1	82.8	81.8	86.6	77.1	85.6
STCN	91.6	90.8	92.5	83.0	81.9	86.5	77.9	85.7
AOT-R50	91.1	90.1	92.1	84.1	83.7	88.5	78.1	86.1
BATMAN	92.5	90.7	94.2	85.3	84.7	89.8	79.2	87.4
RDE	91.1	89.7	92.5	-	-	-	-	-
XMem	91.5	90.4	92.7	85.7	84.6	89.3	80.2	88.7
DeAOT-R50	92.3	90.5	94.0	86.0	84.9	89.9	80.4	88.7
ISVOS	92.6	91.5	93.7	86.3	85.5	90.2	80.5	88.8
RAVOS	91.7	90.8	92.6	83.2	82.2	86.9	77.9	85.9
Cutie	-	-	-	86.1	85.8	90.5	80.0	88.0
HCNet-Lite	92.7	91.6	93.8	86.3	86.0	90.6	80.5	88.1
HCNet	**92.9**	**91.9**	**93.9**	**86.7**	**86.3**	**90.8**	**80.7**	**89.0**

Table 2: Quantitative comparison with state-of-the-art SVOS methods on common datasets including the validation sets of DAVIS 2016 and YouTube-VOS 2018.

holistic correction mechanism at efficiency for regional input is fully demonstrated.

Datasets and Evaluation Metrics. We comprehensively evaluate the performance of our method in face of various scenarios on seven datasets including common datasets DAVIS 2016 (Perazzi et al. 2016), DAVIS 2017 (Pont-Tuset et al. 2018), YouTube-VOS 2018 (Xu et al. 2018) and YouTube-VOS 2019 (Xu et al. 2018), complex datasets VOST (Tokmakov, Li, and Gaidon 2023) and MOSE (Ding et al. 2023) and long-video dataset LVOS (Hong et al. 2023).

For DAVIS, MOSE, LVOS datasets, we adopt Jaccard index $\mathcal{J}$, contour accuracy F and their average $\mathcal{J}\&\mathcal{F}$ following (Cheng et al. 2024). For YouTube-VOS datasets, we report $\mathcal{J}_s$,$\mathcal{F}_s$ and $\mathcal{J}_u$, $\mathcal{F}_u$ for both seen and unseen categories, and the averaged overall score $\mathcal{G}$ following (Cheng and Schwing 2022). For VOST dataset, we report the stan-

Method	VOST val		VOST test		MOSE val		
	$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
CFBI	45.0	32.0	43.9	32.1	45.7	41.2	50.1
CFBI+	46.0	32.6	46.7	31.6	47.8	42.9	52.7
AOT-R50	48.7	36.4	49.9	37.1	57.2	53.1	61.3
RDE	41.3	30.6	40.7	30.2	48.8	44.6	52.9
XMem	44.1	33.8	44.0	32.0	59.6	55.4	63.7
DeAOT-R50	50.4	36.1	50.3	36.2	64.1	59.5	68.7
Cutie	-	-	-	-	68.3	64.2	72.3
RMem	51.8	40.4	-	-	-	-	-
HCNet-Lite	52.6	41.8	52.1	40.6	70.1	66.4	73.8
HCNet	**54.1**	**42.4**	**53.8**	**41.2**	**71.4**	**67.9**	**74.9**

Table 3: Quantitative comparison with state-of-the-art SVOS methods on complex datasets including the validation and test sets of VOST, the validation set of MOSE.

Method	MB	LVOS val			LVOS test			Mem
		$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	
CFBI	F+P	51.5	46.2	56.7	44.8	40.2	49.4	3.82
STCN	A	48.9	43.9	54.0	48.3	44.0	52.5	0.92
RDE	C	53.7	48.3	59.2	50.2	45.7	54.6	1.0
XMem	LT	52.9	48.1	57.7	50.9	46.5	55.3	3.34
DDMemory	C	61.9	56.3	67.4	55.7	50.3	61.2	0.88
Cutie	LT	60.1	55.9	64.2	56.2	51.8	60.5	2.36
HCNet-Lite	FIFO	60.6	56.2	65.0	56.4	52.1	60.7	1.48
HCNet	FIFO	**62.3**	**56.8**	**67.8**	**56.9**	**52.7**	**61.1**	2.52

Table 4: Quantitative comparison with state-of-the-art SVOS methods on long-video dataset LVOS including validation and test sets. MB denotes the kind of memory bank used in these memory-based methods. F+P, A, C, LT and FIFO denote first and previous frame, all frames, compressed memory, long-term memory and first-in-first-out memory respectively. Mem(G): maximum GPU memory usage.

#	SeC	SpC	TC	VOST val		MOSE val		
				$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
1				42.3	31.2	48.6	43.9	53.3
2	✓			43.5	32.3	50.3	45.6	55.0
3		✓		42.6	32.1	49.1	44.5	53.7
4	✓	✓		45.2	34.2	53.1	49.7	56.5
5			✓	48.3	36.1	57.8	53.7	61.9
6	✓		✓	52.8	41.5	70.1	66.0	74.2
7		✓	✓	52.1	40.9	67.6	63.4	71.8
8	✓	✓	✓	**54.1**	**42.4**	**71.4**	**67.9**	**74.9**

Table 5: Ablation studies on different correction modules including semantic correction (SeC), spatial correction (SpC) and temporal correction (TC).

Setting	VOST val		MOSE val		
	$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
only first w/o aggregation	49.2	37.6	59.7	55.3	64.1
only latest w/o aggregation	52.8	40.2	68.1	64.0	72.2
aggregation w/o modulation	53.6	41.7	70.8	67.1	74.5
aggregation w modulation	**54.1**	**42.4**	**71.4**	**67.9**	**74.9**

Table 6: Ablation studies on different aggregation and modulation settings in Adaptive Prototype Update module.

dard metric $\mathcal{J}$ and its special version $\mathcal{J}_{tr}$ representing the average Jaccard over the harder last 25% frames following (Tokmakov, Li, and Gaidon 2023).

Comparison with State-of-the-art

Results on Common Datasets. As shown in Tab. 1 and Tab. 2, our proposed HCNet-Lite and HCNet can both achieve higher accuracy than existing SOTA methods on common datasets including DAVIS 2016, DAVIS 2017, YouTube-VOS 2018 and YouTube-VOS 2019. Especially HCNet-Lite can not only improve the accuracy, but also achieve the highest segmentation speed of 67 FPS, which is improved by almost 50% compared to the newest SOTA method Cutie (Cheng et al. 2024).

Results on Complex Datasets. As shown in Tab. 3, our proposed HCNet-Lite and HCNet can achieve remarkable performance improvement on the complex datasets VOST and MOSE. Compared with the newest SOTA methods Cutie and RMem, HCNet can significantly improve the accuracy, achieving $2.3\%, 2.0\%$ and 3.1% gains in terms of $\mathcal{J}, \mathcal{J}_{tr}, \mathcal{J}\&\mathcal{F}$ on the validation sets of VOST and MOSE under the same training settings.

Results on Long-video Datasets. As shown in Tab. 4, our proposed HCNet can achieve the highest accuracy with competitive memory usage when evaluated on long-video dataset, meanwhile our proposed HCNet-Lite can achieve competitive performance with less GPU memory usage.

Ablation Studies

Correction Modules. As shown in Tab. 5, we verify the impact of semantic, spatial and temporal correction modules by setting different combinations. Compared to settings $1, 2, 3, 4$, temporal correction in setting 5 significantly improve the accuracy which means that regions containing complete objects are necessary and the proposed temporal correction can effectively improve the robustness of estimated regions. With the addition of semantic and spatial correction modules (settings $6, 7, 8$), the accuracy can be effectively improved.

Prototype Size	VOST val		MOSE val			FPS
	$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	
$r^4 = 6$	53.2	40.5	70.1	66.8	73.4	**51**
$r^4 = 7$	53.8	41.8	70.9	67.5	74.3	47
$r^4 = 8$	**54.1**	**42.4**	**71.4**	**67.9**	**74.9**	44
$r^4 = 9$	54.0	42.2	71.2	67.6	74.8	38

Table 7: Ablation studies on prototype size.

Input	Match	VOST val		MOSE val			FPS
		$\mathcal{J}\uparrow$	$\mathcal{J}_{tr}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	
F	F	52.7	40.1	69.9	66.5	73.3	32
F	R	53.3	41.0	70.7	67.2	74.2	35
R	R	**54.1**	**42.4**	**71.4**	**67.9**	**74.9**	**44**

Table 8: Ablation studies on different frame size settings during input and the first matching stage. F, R denote Full and Regional respectively.

Prototype Update Settings. As shown in Tab. 6, we conduct experiments with different settings in Adaptive Prototype Update module. When only preserving object prototypes of the first given frame, the accuracy is much lower than these settings which update object prototypes with the latest features. And aggregation between previous prototypes and current features with the modulation factor can effectively further improve the accuracy.

Prototype Size. As the feature size of each encoding stage is half that of the previous stage, we only need to discuss the size of object prototypes in the fourth stage r^4. As shown in Tab. 7, when r^4 is set to 8, HCNet achieves the highest accuracy and the balanced efficiency, and the corresponding r^3 and r^2 is set as 16 and 32.

Regional Setting. As shown in Tab. 8, we compare the full and regional frame involving in the encoding and the first matching stage. When both are full size, the accuracy and efficiency are lower than the setting with both regional size. Therefore, we finally adopt the regional input and regional matching in the proposed HCNet and HCNet-Lite.

Conclusion

In this work, we propose a novel Holistic Correction Network (HCNet) for semi-supervised video object segmentation. Based on the adaptive update of object prototypes, we can perform semantic, spatial and temporal correction to correct erroneous matching. We conduct comprehensive experiments on existing common, complex and long-video datasets, and the evaluation results show that our proposed HCNet and HCNet-Lite can effectively improve the accuracy and efficiency of segmentation on various scenarios.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. 62132002, 62102206, 62202249) and Major Key Project of PCL (Grant No. PCL2024A04-4).

References

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *Eur. Conf. Comput. Vis.*, 850–865.

Cheng, H. K.; Oh, S. W.; Price, B.; Lee, J.-Y.; and Schwing, A. 2024. Putting the object back into video object segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 3151–3161.

Cheng, H. K.; and Schwing, A. G. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Eur. Conf. Comput. Vis.*, 640–658.

Cheng, H. K.; Tai, Y.-W.; and Tang, C.-K. 2021. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Adv. Neural Inform. Process. Syst.*, 11781–11794.

Ding, H.; Liu, C.; He, S.; Jiang, X.; Torr, P. H.; and Bai, S. 2023. MOSE: A New Dataset for Video Object Segmentation in Complex Scenes. In *IEEE/CVF Int. Conf. Comput. Vis.*, 20224–20234.

He, Z.; Xia, C.; Qiao, S.; and Li, J. 2024. Text-prompt Camouflaged Instance Segmentation with Graduated Camouflage Learning. In *ACM Int. Conf. Multimedia*, 5584–5593.

Hong, L.; Chen, W.; Liu, Z.; Zhang, W.; Guo, P.; Chen, Z.; and Zhang, W. 2023. Lvos: A benchmark for long-term video object segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 13480–13492.

Li, J.; Qiao, S.; Zhao, Z.; Xie, C.; Chen, X.; and Xia, C. 2023. Rethinking lightweight salient object detection via network depth-width tradeoff. *IEEE Trans. Image Process.*, 32: 5664–5677.

Li, M.; Hu, L.; Xiong, Z.; Zhang, B.; Pan, P.; and Liu, D. 2022. Recurrent dynamic embedding for video object segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 1332–1341.

Liang, Y.; Chen, H.; Wu, Q.; Xia, C.; and Li, J. 2024. Joint spatio-temporal similarity and discrimination learning for visual tracking. *IEEE Trans. Cir. and Sys. for Video Tech.*, 34: 7284–7300.

Lin, Z.; Yang, T.; Li, M.; Wang, Z.; Yuan, C.; Jiang, W.; and Liu, W. 2022. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 1362–1372.

Ma, M.; Xia, C.; Xie, C.; Chen, X.; and Li, J. 2023. Boosting broader receptive fields for salient object detection. *IEEE Trans. Image Process.*, 32: 1026–1038.

Miao, B.; Bennamoun, M.; Gao, Y.; and Mian, A. 2024. Region aware video object segmentation with deep motion modeling. *IEEE Trans. Image Process.*, 33: 2639–2651.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *IEEE/CVF Int. Conf. Comput. Vis.*, 9226–9235.

Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 724–732.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Gool, L. V. 2018. The 2017 DAVIS Challenge on Video Object Segmentation. arXiv:1704.00675.

Tokmakov, P.; Li, J.; and Gaidon, A. 2023. Breaking the" Object" in Video Object Segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 22836–22845.

Wang, J.; Chen, D.; Wu, Z.; Luo, C.; Tang, C.; Dai, X.; Zhao, Y.; Xie, Y.; Yuan, L.; and Jiang, Y.-G. 2023. Look before you match: Instance understanding matters in video object segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2268–2278.

Xie, H.; Yao, H.; Zhou, S.; Zhang, S.; and Sun, W. 2021. Efficient regional memory network for video object segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 1286–1295.

Xu, N.; Yang, L.; Fan, Y.; Yue, D.; Liang, Y.; Yang, J.; and Huang, T. 2018. YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark. arXiv:1809.03327.

Yang, Z.; Wei, Y.; and Yang, Y. 2020. Collaborative video object segmentation by foreground-background integration. In *Eur. Conf. Comput. Vis.*, 332–348.

Yang, Z.; Wei, Y.; and Yang, Y. 2021a. Associating objects with transformers for video object segmentation. *Adv. Neural Inform. Process. Syst.*, 34: 2491–2502.

Yang, Z.; Wei, Y.; and Yang, Y. 2021b. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9): 4701–4712.

Yang, Z.; and Yang, Y. 2022. Decoupling features in hierarchical propagation for video object segmentation. *Adv. Neural Inform. Process. Syst.*, 35: 36324–36336.

Yu, T.; Xia, C.; and Li, J. 2024. Towards imbalanced motion: part-decoupling network for video portrait segmentation. *Science China Information Sciences*, 67(7): 172104.

Yu, Y.; Yuan, J.; Mittal, G.; Fuxin, L.; and Chen, M. 2022. Batman: Bilateral attention transformer in motion-appearance neighboring space for video object segmentation. In *Eur. Conf. Comput. Vis.*, 612–629.