Online Unsupervised Video Object Segmentation via Contrastive Motion Clustering

Lin Xi, *Student Member, IEEE,* Weihai Chen*, *Member, IEEE,* Xingming Wu, Zhong Liu, *Member, IEEE,* and Zhengguo Li, *Fellow, IEEE,*

***Abstract*—Online unsupervised video object segmentation (UVOS) uses the previous frames as its input to automatically separate the primary object(s) from a streaming video without using any further manual annotation. A major challenge is that the model has no access to the future and must rely solely on the history, *i.e.*, the segmentation mask is predicted from the current frame as soon as it is captured. In this work, a novel contrastive motion clustering algorithm with an optical flow as its input is proposed for the online UVOS by exploiting the common fate principle that visual elements tend to be perceived as a group if they possess the same motion pattern. We build a simple and effective auto-encoder to iteratively summarize non-learnable prototypical bases for the motion pattern, while the bases in turn help learn the representation of the embedding network. Further, a contrastive learning strategy based on a boundary prior is developed to improve foreground and background feature discrimination in the representation learning stage. The proposed algorithm can be optimized on arbitrarily-scale data (*i.e.*, frame, clip, dataset) and performed in an online fashion. Experiments on *DAVIS*$_{16}$, *FBMS*, and *SegTrackV2* datasets show that the accuracy of our method surpasses the previous state-of-the-art (SoTA) online UVOS method by a margin of 0.8%, 2.9%, and 1.1%, respectively. Furthermore, by using an online deep subspace clustering to tackle the motion grouping, our method is able to achieve higher accuracy at $3\times$ faster inference time compared to SoTA online UVOS method, and making a good trade-off between effectiveness and efficiency. Our code will be available upon the acceptance of our paper.**

***Index Terms*—Object segmentation, image motion analysis, unsupervised learning, self-supervised learning, optical flow, clustering methods.**

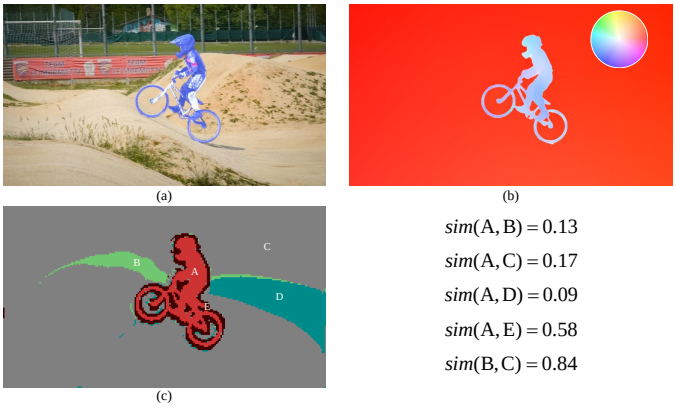


Fig. 1: Motion grouping. (a) The RGB image with ground truth; (b) the optical flow visualized by the inset color wheel; (c) the motion segmentation using our proposed prototypical subspace clustering framework (clusters k=5). According to the prototypical subspace bases, we clustered different motion patterns (*i.e.*, A-**red**, B-**green**, C-**grey**, D-**cyan**, and E-**dark red**). $sim(\cdot)$ is the similarity of different motion patterns and is normalized to $[0, 1]$.

I. Introduction

WHEN looking around in a dynamic scene, visual elements moving at the same speed and/or direction tend to attract human attention as part of a single stimulus. This principle is called common fate and is theorized by Gestalt psychology [1]. A common example is a BMX rider going through dirt jumps. If the rider and the BMX bike have the same trajectory, they are perceived as the same motion group. The background, which has a different trajectory than the BMX rider, does not appear to be part of the same group, as shown in Fig. 1.

According to the Gestalt principle of common fate, objects should share a common destination, moving together consistently throughout the scene. Therefore, the motion of objects can serve as an important cue for video object segmentation (VOS). In computer vision, the pixel-wise motion in the scene can be obtained from an optical flow estimation and used to determine, segment, and learn the objects. In the recent literature of VOS, many learning-based models [2]–[17] have been proposed to learn more discriminative objectness by leveraging motion information. While such models through supervised learning require massive pixel-wise annotations, they are limited to a small range of object categories predefined in the datasets. To reduce the cost of data labeling, numerous unsupervised approaches [18]–[24] have been proposed that use the motion cues. However, traditional physics-optimization-based approaches incur significant computational costs due to the optimization process over the entire video. Unsupervised methods [25]–[27] based on deep neural networks have gained significant advantage by learning a deep representation on a

This work was supported in part by the National Natural Science Foundation of China under grant 51975029 and U1909215, the Key Research and Development Program of Zhejiang Province under Grant 2021C03050, the Scientific Research Project of Agriculture and Social Development of Hangzhou under Grant No. 2020ZDSJ0881, and in part by the National Natural Science Foundation of China under grant 61620106012 and 61573048.

L. Xi, W. Chen, X. Wu, and Z. Liu are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: xilin1991@buaa.edu.cn; whchen@buaa.edu.cn; wxmbuaa@163.com; liuzhong@buaa.edu.cn).

Z. Li is with the SRO Department, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #21-01, Connexis South Tower, Singapore 138632, Republic of Singapore (e-mail: ezgli@i2r.a-star.edu.sg).

*Corresponding author: Weihai Chen.

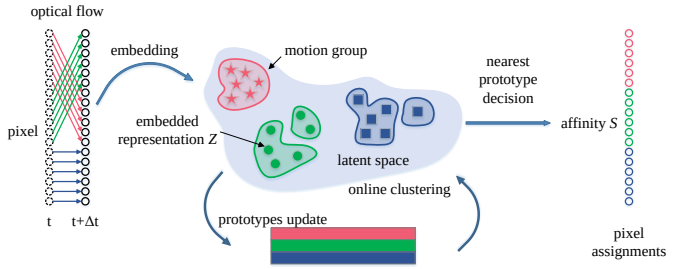


Fig. 2: Paradigm of the proposed clustering method. We iteratively summarize prototypical bases from the embedded representation $\boldsymbol{Z}$, and the $\boldsymbol{Z}$ are then combined with the prototypes to compute the affinity S.

video dataset. Although those methods achieve good performance, they cannot handle streaming video because they work offline and require the entire video to be processed before making predictions.

Unlike unsupervised video object segmentation (UVOS) in the offline setting, a major challenge for online UVOS is that the inference is performed solely on observations of the past, without utilizing the information from video frames in the future. While processing video online is beneficial for many applications, such as video compression [28]–[31], analysis [32]–[36], and editing [37]. Tokmakov *et al.* [38] proposed an end-to-end network to map the optical flow to motion segmentation, followed by an object proposal model [39] to extract the candidate objects. Similarly, Zhou *et al.* [40] proposed a method based on salient motion detection and object proposals which were directly predicted by a pre-trained model [41] without fine-tuning to obtain final CRF refined results. While these methods require training on a large dataset or object masks, they incur computational cost in the optimization (training) and inference process. In addition, some online UVOS methods suffered from another shortcoming: inappropriate use of motion cues. For example, Taylor *et al.* [42] used long-term temporal information to initialize the target object on a few frames for online unsupervised framework. However, uninformative history frames cause error accumulation during the propagation, which degrades the performance of online UVOS. Perazzi *et al.* [43] employed a salient detection method without considering the object information and motion cues of video content, which is a limitation in UVOS where the segmented object was defined as the main object in the scene with a distinctive motion.

We argue that an online UVOS model must predict frame-level segmentation in correlation to what happened hitherto, and as efficiently as possible for the online setting. Therefore, an online UVOS model should satisfy the following criteria: (i) frame-by-frame manner, (ii) relatively fast optimization and inference, and (iii) short-term temporal dependence.

In this paper, a novel online UVOS algorithm is proposed by using an efficient and accurate online deep subspace clustering for motion grouping, which directly factorizes the optical flow into k groups corresponding to k subspaces. The proposed algorithm processes one video frame at a time without any additional pre/post-processing (*i.e.*, [44], [45]), and the results for the frame is depend only on the previous frame. A simple video auto-encoder model is designed to summarize a set of subspace prototypes from the latent space, where the deep auto-encoder is optimized on each individual video sequence and does not require training on a large dataset. Specifically, the motion segmentation problem is addressed by training a generative model that is used to learn an embedded representation $\boldsymbol{Z}$ of the optical flow. The $\boldsymbol{Z}$ is then combined with the centers of each segment, *i.e.*, the prototypes, to construct the subspace affinity vector $\boldsymbol{S}$ for pixel assignments. Each pixel is assigned to the nearest prototype without relying on additional learnable parameters. The prototypes are formed by clustering nearby points in the embedding space. They represent motion groups following the common fate principle. The network structure of the proposed method is illustrated in Fig. 2. In order for the auto-encoder to effectively learn the discriminative features between the foreground and background, we further exploit an important scene prior [46], *i.e.*, the optical flow at the boundary of an image is significantly different from the motion direction of the object of interest, to design a pixel-level contrastive learning strategy.

Overall, our main contributions are: 1) a novel online deep subspace clustering method for the online UVOS by exploiting the motion cue; 2) an effective optimization approach for the online clustering that can handle an arbitrary video independently without being trained on large datasets; and 3) a pixel-level contrastive learning strategy that significantly improves the foreground and background feature discrimination for the auto-encoder. To validate these contributions, the proposed method is evaluated on three public benchmarks (*i.e.*, $DAVIS_{16}$, *FBMS*, and *SegTrackV2*); the proposed algorithm outperforms state-of-the-art (SoTA) online UVOS models, while being faster to optimize and infer.

II. Related Works

Motion segmentation is to identify and segment independently moving objects in a video, that is, to solve the problem of motion grouping. Many approaches tackle the issue from a motion clustering point of view. Shi *et al*. considered motion segmentation as a spatio-temporal image clustering problem [47]. To increase robustness, some methods use motion cues, such as point trajectories [48]–[51] or optical flow [19], [52] accumulated over multiple frames, to segment moving objects. Luo *et al*. [24] proposed a complexity awareness framework that exploits local clips and their relationships for motion segmentation. Kumar *et al*. proposed an algorithm to obtain the initial estimate of the model by dividing the scene into rigidly moving components to solve a grouping problem to associate pixels into a number of motion clusters [53]. Brox *et al*. defined pairwise distances between point trajectories from adjacent frames for the motion clustering [54]. Ochs and Brox [55] adopted the spectral clustering on hypergraphs which is a similarity map obtained from a third motion vector, instead of pairs [54] to segment point trajectories.

It is worth noting that Xie *et al*. [56] also inserted motion clustering into their object segmentation problem pipeline. However, under the premise of a supervised setting, this

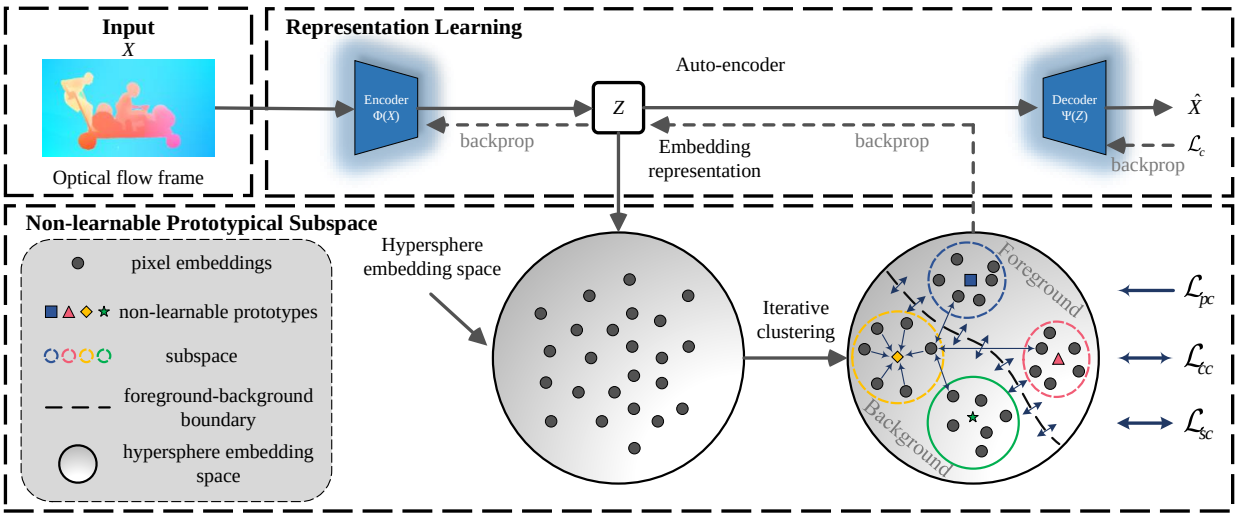


Fig. 3: The overview optimization diagram for our proposed method with the optical flow as our input. Given an optical flow $\boldsymbol{X}$, we utilizes auto-encoder to embed it into a p-dimensional embedding feature $\boldsymbol{Z}$ and outputs its corresponding reconstruction $\hat{\boldsymbol{X}}$. During the optimization phase, we iteratively summarize non-learnable prototypical bases for the motion pattern, while the bases are constrained by our proposed contrastive learning strategy to help shape the feature space. To obtain the final cluster labels, we use the proposed subspace clustering algorithm with a hard assignment to group each pixel to the prototypical bases.

method introduces a pixel-trajectory recurrent neural network that learns the trajectories of foreground pixels and clusters pixels over time. In contrast, the proposed algorithm learns motion patterns only using the optical flow without requiring any manual annotation. In addition, our resulting feature representations which are optimized on individual videos give us a global dependence over entire videos.

Unsupervised video object segmentation aims to automatically identify and segment the most visually prominent objects from the background in sequences, unlike semi-supervised [57]–[61] and referring [62]–[64] video object segmentation which involves human inspection. Recently, many approaches [2], [3], [38], [65], [66] have been proposed to tackle the offline UVOS. Although the term "unsupervised" is used here, in practice there are some differences from *fully unsupervised* settings. In general, many popular algorithms [4], [6]–[17] require supervised training on large-scale datasets to obtain the segmentation masks. Alternatively, a number of works [25]–[27] based on the offline setting employ a deep neural network to discover the objects of interest from the perspective of completely unsupervised concepts in the traditional methods. Lu *et al.* [67] proposed a unified framework for unsupervised learning, aimed at object segmentation through the exploitation of the inherent consistency across adjacent frames in unlabeled videos. Similar to our work, Yang *et al*. [26] used slot attention [68] to learn to segment objects in a self-supervised manner, and also took the optical flow as the input of the auto-encoder, which is a type of generative model (*e.g.*, VAEs [69] and GANs [70], [71]), but slot attention and binding graph neural networks (GNN) rely on large-scale datasets for training. DyStaB employs static and dynamic models to learn object saliency from motion in a video, which can then be applied at inference time to segment objects, even in static images [25]. Deformable Sprites (DeSprites) [27] are a type of video auto-encoder model that is optimized on each individual video. Our work also optimizes an auto-encoder on a specific sequence in an unsupervised manner. Unlike the SoTA offline UVOS method DeSprites, our goal is to cluster the points that share motion patterns in the embedding space, which significantly improves the effectiveness of the optimization and reduces the inference time for online UVOS.

Subspace clustering is to segment the original data space into its corresponding subspace. Classical subspace clustering methods used kernels to transform the original data into a high-dimensional latent feature space in which subspace clustering is performed [72]. Recently, there have been a few works that used deep learning techniques for feature extraction in subspace clustering. Ji *et al*. developed a convolutional auto-encoder network combined with a self-expression module [73], which showed significant improvement on several image datasets. Instead of constructing the affinity matrix for subspace clustering, Zhang *et al*. utilized deep neural networks to iteratively project data into a latent space and update the k-subspaces [74]. In [75], [76], a k-factorization subspace clustering was proposed for large-scale subspace clustering, which effectively reduces the complexity of clustering.

In this paper, we attempt to develop a joint optimization framework for the online UVOS that can simultaneously learn feature representation and subspace clustering. Inspired by the effectiveness of the k-FSC model [75], we combine a powerful CNN to define k non-learnable prototypes in the latent space as the k-subspace of clustering.

Contrastive learning is an attention-grabbing unsupervised representation learning method that maximizes the similarity of positive pairs while minimizing the similarity of negative pairs in a feature space [77], [78]. Li *et al*. proposed the contrastive clustering method, which performs dual contrastive learning at the instance and cluster level under a unified frame-

work [79]. By adopting the foreground-background saliency prior [46] for contrastive learning, we propose a novel pixel-level contrastive learning framework without the requirement of image-level supervision. Features from the foreground are pulled together and contrasted against those from the background, and vice versa.

III. Methods

In our "online unsupervised" setting, an optical flow is taken as our input and all pixels are assigned into different groups to predict a segment containing the moving object by an online deep subspace clustering on top of non-learnable prototypes. One video frame at a time is processed, and the results for the frame depend only on the previous frame. The overall framework of the proposed method is shown in Fig. 3.

Problem Formulation. *Let* $\{\boldsymbol{X}^{(i)}_{t\to t+\Delta t} \in \mathbb{R}^{H\times W\times 2}\}_{i=1}^{N}$ *($N \in \mathbb{N}^*$) denote optical flow frames from the individual video, where $H \times W$ indicates the spatial resolution of images. Assume that a pixel point $\mathbf{x}^{(s)}_{t\to t+\Delta t}$ ($s \in \mathbb{R}^{H\times W}$) in an optical flow frame is drawn from a p-dimensional subspaces $\{\mathcal{S}_j\}_{j=1,\cdots,k}$ (i.e., $\mathbf{x}^{(s)}_{t\to t+\Delta t} \in \mathcal{S}_j$), where k clusters correspond to k different subspaces. For $j = 1, ..., k$, the pixel point $\mathbf{x}^{(s)}_{t\to t+\Delta t}$ can be formally represented as:*

$$\mathbf{x}^{(s)}_{t\to t+\Delta t} = g(\boldsymbol{U}_j\mathbf{v}) + \epsilon, \tag{1}$$

where one subspace can be expressed by a specific subspace base $\boldsymbol{U}_j \in \mathbb{R}^{p\times r_j}$ ($p < r_j$), $\mathbf{v} \in \mathbb{R}^{r_j}$ denotes a random variable, $\epsilon \in \mathbb{R}^2$ is random noise, and $g : \mathbb{R}^p \to \mathbb{R}^2$. Find k cluster patterns from the r-dimensional latent space $\mathcal{U}$ and $p < r$.

A. Network Formulation

The auto-encoder is a widely used self-supervised model and it can embed the raw data into a customizable latent space. A network architecture consisting of multiple convolutional layers is adopted to map the optical flow into the r-dimensional latent space $\mathcal{U}$, and then the p-dimensional embedding features $\boldsymbol{Z} \in \mathbb{R}^{\frac{H}{c}\times\frac{W}{c}\times p}$ is denoted by

$$\boldsymbol{Z} = F(\Phi(\boldsymbol{X})) + A(F(\Phi(\boldsymbol{X}))), \tag{2}$$

where c is a scale, and a feature multi-layer perceptron (MLP) which is stacked after the embedding features $\Phi(\boldsymbol{X})$ is denoted by $F : \mathbb{R}^r \to \mathbb{R}^p$ ($p < r$). A spatial attention module $A(\cdot)$, which is implemented by the sum of max and average pooling followed by an upsampling layer, is added to improve the spatial stability.

The embedding features $\boldsymbol{Z}$ are fed into a decoder $\Psi(\cdot)$ to reconstruct the optical flow, and the reconstruction loss $\mathcal{L}_c$ for the auto-encoder is formulated as:

$$\mathcal{L}_c = \frac{1}{S} \parallel \boldsymbol{X} - \hat{\boldsymbol{X}} \parallel_F^2 = \frac{1}{S}\sum_{s\in S} \parallel \mathbf{x}^{(s)} - \hat{\mathbf{x}}^{(s)} \parallel_2^2 \tag{3}$$

where $\hat{\boldsymbol{X}} = \Psi(\boldsymbol{Z})$ and S is the entire spatial grid. For simplicity, the subscript $t \to t + \Delta t$ is omitted for the optical flow ward in Eqs. 2 and 3. We only leverage the temporal information from the previous frames for the optical flow estimation, hence $\Delta t < 0$ in the online setting.

B. Non-learnable Prototypical Subspace Clustering

Suppose $\boldsymbol{P}^\top\boldsymbol{X} = \bar{\boldsymbol{X}}$, where $\bar{\boldsymbol{X}} = [\bar{\boldsymbol{X}}_1, \bar{\boldsymbol{X}}_2, \cdots, \bar{\boldsymbol{X}}_k]$ and $\boldsymbol{P}$ is an unknown permutation matrix. According to the assumption in the proposed **Problem Formulation**, once $\boldsymbol{U}_j$ is obtained from Eq. 1, the correct clusters are then identified. This implies that the $\boldsymbol{U}_j$ is not explicitly determined. Instead, a neural network is exploited to replace Eq. 1 by approximating $\mathbf{x}^{(s)}$, which yields the following formulation:

$$\hat{\mathbf{x}}^{(s)} = \Psi(\hat{\boldsymbol{U}}_j\hat{\mathbf{v}^{(s)}}), \tag{4}$$

where $\mathbf{z}^{(s)} := \hat{\boldsymbol{U}}_j\hat{\mathbf{v}^{(s)}}$ ($\mathbf{z}^{(s)} \in \mathbb{L}^{(s)}$), $\mathbf{z}^{(s)}$ refers to the embedding feature associated with pixel $\mathbf{x}^{(s)}$, and $\mathbb{L}^{(s)}$ indicates the true cluster to which $\mathbf{x}^{(s)}$ should be assigned. There exists a direct correspondence between the spatial positions of embedding features and pixels, establishing a one-to-one relationship between $\mathbf{z}^{(s)}$ and $\mathbf{x}^{(s)}$. In fact, it is difficult to determine $\mathbb{L}^{(s)}$ directly. Now the embedding feature $\mathbf{z}^{(s)}$ is ℓ_2 normalized so that it lies on the surface of a unit hypersphere, as shown in Fig. 4. A new variable $\mathcal{P} \in \mathbb{R}^{p\times k}$ is introduced as the subspace prototypes, where $\mathcal{P} = [\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_k]$ and $\parallel \mathcal{P}_j \parallel$=1, $j = 1, \cdots, k$, and p is the dimension of each prototype. The function of $\mathcal{P}_j$ is to summarize the subspace $\mathcal{S}_j$, $j = 1, \cdots, k$. Thus $\parallel \mathcal{P}_i^\top\mathcal{P}_j \parallel$ is assumed to be small enough for all $i \neq j$, *i.e.*,

$$\parallel \mathcal{P}_i^\top\mathcal{P}_j \parallel \leq \tau,\ i \neq j, \tag{5}$$

where τ is a small constant.

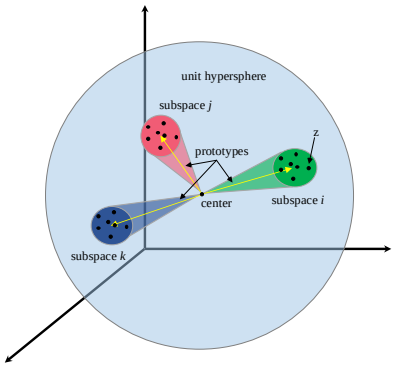


Fig. 4: The simplified diagram of $p-1$ dimensional unit hypersphere, where each subspace corresponds to the surface area of the unit hypersphere centered on different prototypes, denoted as $\mathcal{P}_j$. When $\parallel \mathcal{P}_i^\top\mathcal{P}_j \parallel$ is sufficiently small for all $i \neq j$, it means that each prototype $\mathcal{P}_j$ on the unit hypersphere is situated at a greater distance, enabling the identification of a suitable boundary for clustering.

The embedding feature $\mathbf{z}^{(s)}$ of a data sample is compared with $\mathcal{P}_j$ ($j = 1, \cdots, k$) to obtain the winning prototype as

$$\alpha^{(s)} = \arg\max_j \parallel {\mathbf{z}^{(s)}}^\top\mathcal{P}_j \parallel . \tag{6}$$

It is assumed that

$$s^{(s,\alpha)} = \parallel {\mathbf{z}^{(s)}}^\top\mathcal{P}_{\alpha^{(s)}} \parallel \gg \max_{j\neq\alpha^{(s)}} \parallel {\mathbf{z}^{(s)}}^\top\mathcal{P}_j \parallel,\ s = 1, \cdots, S, \tag{7}$$

where $s^{(s,\alpha)}$ denotes the affinity. In other words, maximizing the likelihood of Eq. 6 is assigning $\mathbf{z}^{(s)}$ to one of $\mathcal{P}$ with a probability distribution:

$$p(\alpha^{(s)}|\mathbf{z}^{(s)}) = \frac{\exp(s^{(s,\alpha)})}{\sum_{j=1}^{k}\exp(s^{(s,j)})}. \tag{8}$$

An online clustering strategy is adopted to update $\alpha^{(s)}$ so that the pixels with the same motion pattern are assigned to the prototype $\mathcal{P}_j$ belonging to that subspace $\mathcal{S}_j$ according to $s^{(s,j)}$. It can be known from the permutation matrix $\boldsymbol{P}$ that the mapping $\boldsymbol{T}$ that assigns the pixel $\mathbf{x}^{(s)}$ to the prototypes is related to it as

$$\boldsymbol{T}^\top \mathcal{P}^\top \boldsymbol{Z} \propto \boldsymbol{P}^\top \boldsymbol{X}, \tag{9}$$

where the column of $\boldsymbol{T} \in \mathbb{R}^{k\times p}$ is the one-hot assignment vector of pixel $\mathbf{x}^{(s)}$ over k prototypes. In other words, each pixel is assigned to a single prototype, and the sum of pixels matched by all the prototypes is equal to all pixels in the frame. Thus, the augmented assignment $\boldsymbol{T}$ now has the following constraints:

$$\boldsymbol{T}^\top \mathbf{1}_k = \mathbf{1}_S \text{ and } \boldsymbol{T}\mathbf{1}_S = \frac{S}{k}\mathbf{1}_k, \tag{10}$$

where $\mathbf{1}_k$ denotes the vector of all ones of k dimensions.

The mapping $\boldsymbol{T}$ can be optimized by maximizing the probability distribution $p(\alpha^{(s)}|\mathbf{z}^{(s)})$ (Eq. 8) between the pixel embedding $\boldsymbol{Z}$ and the prototypes $\mathcal{P}$. The solution of the above optimization problem corresponds to the optimal transport [80]:

$$\begin{aligned} &\max_{\boldsymbol{T}\in\mathbb{R}_+^{S\times k}} \mathtt{Tr}(\boldsymbol{T}^\top \mathcal{P}^\top \boldsymbol{Z}) + \kappa h(\boldsymbol{T}), \\ &s.t.\ \ \boldsymbol{T}^\top \mathbf{1}_k = \mathbf{1}_S,\ \boldsymbol{T}\mathbf{1}_S = \tfrac{S}{k}\mathbf{1}_k, \end{aligned} \tag{11}$$

where $h(\boldsymbol{T})$ is an entropy, and $\kappa > 0$ is a parameter that controls the smoothness of the distribution. The efficient solver on GPU of Eq. 11 can be given as the Sinkhorn algorithm [81], [82]. Our online subspace clustering involves few matrix multiplications, so it is computed by few steps of iteration.

With Eq. 6, the prototype $\mathcal{P}_j$ is estimated from the pixel-wise feature embeddings that are with the highest confidence of clusters j ($j = 1, \cdots, k$). Specifically, for all pixels assigned to subspace j, the prototype $\mathcal{P}_j$ can be derived as the average of the pixel-wise embeddings, which is the center of the pixel embedding within segment j.

The proposed online subspace clustering method is performed as follows: pixels with the same motion pattern are first assigned to the prototype $\mathcal{P}_j$ belonging to that subspace $\mathcal{S}_j$, and then the prototypes are updated according to the assignments. It is natural to derive a training objective for pixel assignment from Eqs. 5 and 7 as

$$\mathcal{L}_{pc} = \frac{1}{S}\sum_{s=1}^{S}(1 - {\mathbf{z}^{(s)}}^\top \mathcal{P}_{\alpha^{(s)}})^2, \tag{12}$$

where $\mathcal{L}_{pc}$ indicates the prototypes' discrinimitiveness, and

$$\mathcal{L}_{cc} = -\frac{1}{S}\sum_{s=1}^{S}\log\frac{\exp({\mathbf{z}^{(s)}}^\top \mathcal{P}_{\alpha^{(s)}})}{\exp(\sum_{j=1}^{k}{\mathbf{z}^{(s)}}^\top \mathcal{P}_j)}, \tag{13}$$

$\mathcal{L}_{cc}$ is the cluster contrastive loss.

C. Contrastive Learning based on a Boundary Prior

Considering the fact that the motion of the foreground object is different from that of the background [46], a pixel-level contrastive learning strategy is introduced to improve the feature discrimination between the foreground and background. A core design philosophy for this strategy is that the average motion $\mathbf{m}$ of the boundary pixel is compared with all pixels $\mathbf{x}^{(s)}$ ($s \in \mathbb{R}^{H\times W}$) in the optical flow frame, so as to judge the similarity between the pixels and the boundary motion to determine whether it belongs to the background region. The cosine similarity between each pixel $\mathbf{x}^{(s)}$ and $\mathbf{m}$ is first computed as:

$$s^{(s)} = 1 - \frac{{\mathbf{x}^{(s)}}^\top \mathbf{m}}{\parallel \mathbf{x}^{(s)} \parallel \parallel \mathbf{m} \parallel}. \tag{14}$$

To determine the background region, a threshold δ is set to binarize the similarity map derived by Eq. 14. After then, the foreground and background sets are obtained by $\mathcal{K}^+ = \{\mathbf{x}^{(+)}|s^{(s)} < \delta\}$ and $\mathcal{K}^- = \{\mathbf{x}^{(-)}|s^{(s)} \geqslant \delta\}$, respectively. Thus, the foreground region $\mathcal{K}^+$ and the average motion $\mathbf{m}_f$ of the foreground pixels are treated as a pair. And the same is true for the background region $\mathcal{K}^-$ and the average motion $\mathbf{m}_b$ of the background pixels. Our contrastive learning framework aims to maximize the distance between the foreground and background representations. The final saliency contrastive loss is formulated as:

$$\begin{aligned} &\mathcal{L}_{sc} = \\ &-\tfrac{1}{\|\mathcal{K}^-\|}\textstyle\sum_{\mathcal{K}^-}\log\frac{\exp({\mathbf{x}^{(-)}}^\top \mathbf{m}_b)}{\exp({\mathbf{x}^{(-)}}^\top \mathbf{m}_b)+\exp({\mathbf{x}^{(-)}}^\top \mathbf{m}_f)} \\ &-\tfrac{1}{\|\mathcal{K}^+\|}\textstyle\sum_{\mathcal{K}^+}\log\frac{\exp({\mathbf{x}^{(+)}}^\top \mathbf{m}_f)}{\exp({\mathbf{x}^{(+)}}^\top \mathbf{m}_f)+\exp({\mathbf{x}^{(+)}}^\top \mathbf{m}_b)}. \end{aligned} \tag{15}$$

When the contrastive loss $\mathcal{L}_{sc}$ is applied to pull close and push apart the representations in positive and negative pairs, the motion pattern of the foreground object and the background in the optical flow are gradually separated.

D. Optimization

Stochastic gradient descent (SGD) is adopted to learn the parameters of the model, which consists of an auto-encoder, a feature MLP, and a spatial attention module.

Now we show how to solve the VOS using our proposed contrastive motion clustering algorithm. We initialize the parameters of the auto-encoder by Eq. 3. The prototypes $\mathcal{P}_j$ ($j = 1, \cdots, k$) are initialized by a Gaussian distribution and are normalized to have unit ℓ_2 norm.

At iteration t, we first obtain the embedding features $\boldsymbol{Z}$ using the auto-encoder and apply ℓ_2 normalization. Then, we assign the label α to each pixel with a posterior probability as in Eq. 8. In the cluster setting, we use the Sinkhorn algorithm [81] with hard assignment to group each pixel for the prototypes $\mathcal{P}_j$ ($j = 1, \cdots, k$), as proposed in Eq. 11. As the M-step of the Expectation-Maximization (EM) framework, the prototypes are then updated by accounting for the online clustering results. The non-learnable prototypes $\mathcal{P}_j$ are not learned by SGD, but are computed as the centers of the

Algorithm 1: Contrastive clustering of optical flow for online UVOS

Input: Optical flow; Number of clusters k; Embedding dimension r; Subspace dimension p; Hyperparameters κ, δ, η, λ_1, λ_2 and λ_3; Maximum iteration $T_{\max}$.

Output: Cluster labels of pixels.

```
Initialize auto-encoder by minimizing Eq. 3.
Initialize non-learnable prototypes from Gaussian distribution and apply ℓ2 normalization.
while t ⩽ T_max do
    Learn embedded representation Z.
    Compute cluster labels by solving Eqs. 8 and 11.
    Update non-learnable prototypes according to cluster labels as in Eq. 16.
    Compute and binarize the saliency map by Eq. 14 to obtain K+ and K− sets, respectively.
    Update the network parameters by minimizing the objective function in Eq. 17.
end
Use Eq. 6 to obtain the final updated cluster labels.
Assign pixel-wise foreground/background labels with the Hungarian algorithm based on boundary prior.
return Foreground/Background labels of pixels.
```

corresponding feature representations $\boldsymbol{Z}$. In particular, in each training iteration, each prototype is updated as:

$$\mathcal{P}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\alpha^{(s)}=j} \mathbf{z}^{(s)}, \tag{16}$$

where $|\mathcal{S}_j|$ is the number of pixels belonging to this subspace $\mathcal{S}_j$, and s denotes the spatial position. Meanwhile, we compute and binarize the saliency map by Eq. 14 to obtain $\mathcal{K}^+$ and $\mathcal{K}^-$ sets for boundary prior-based contrastive learning. The parameters of our model are directly optimized by minimizing the combinatorial loss over all training pixel samples from the each video:

$$\mathcal{L} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{pc} + \lambda_2 \mathcal{L}_{cc} + \lambda_3 \mathcal{L}_{sc}. \tag{17}$$

After performing each training iteration, the cluster labels are obtained from the maximum matching formula in Eq. 6. To guide the object segmentation, we use the boundary motion information as a prior. We found the optimal assignment between foreground and background with the Hungarian algorithm, using the cosine similarity between each prototype $\mathcal{P}_j$ and the background region $\mathcal{K}^-$ as a cost function with a threshold of η. Unlike the foreground region $\mathcal{K}^+$, the background region $\mathcal{K}^-$ tends to be more robust to noise than it.

For each frame, the online subspace clustering is then performed to achieve unsupervised motion segmentation. The whole optimization process is detailed in Algorithm 1. The proposed method achieves a joint optimization of subspace clustering and embedded representation learning.

IV. Experiments

A. Experimental Setup

Datasets and evaluation metrics. To test the performance of our online subspace clustering, we carry out comprehensive experiments on the following three UVOS datasets:

DAVIS$_{16}$ [83] is currently the most popular VOS benchmark, consisting of 50 high-quality video sequences (30 videos for the `train` set and 20 for the `val` set). Each frame is densely annotated with pixel-wise ground truth for the foreground objects. We perform online clustering and evaluation on the validation set. For quantitative evaluation, we adopt standard metrics suggested by [83], namely region similarity $\mathcal{J}$, which is the intersection-over-union of the prediction and ground-truth, computing the mean over the `val` set.

FBMS [84] contains videos of multiple moving objects, providing test cases for multiple object segmentation. The *FBMS* has 59 sparsely annotated video sequences, with 30 sequences for validation.

SegTrackV2 [85] contains 14 densely annotated videos and 976 annotated frames. Each sequence contains 1-6 moving objects and presents challenges such as motion blur, appearance change, complex deformation, occlusion, slow motion, and interacting objects.

Following the evaluation protocol in [23], we combine multiple objects as a single foreground and use the region similarity $\mathcal{J}$ to measure the segmentation performance for the *FBMS* and *SegTrackV2*.

Implementation details. The optical flow is estimated by using the RAFT [86] and FlowFormer [87]. The flows are resized to the size of the original image [26], with each input frame having a size of 480×854 for the *DAVIS$_{16}$* and 480×640 for the *FBMS* and *SegTrackV2*. We convert the optical flow to 3-channel images with the standard visualization used for the optical flow and normalize it to [-1, 1], and use only the previous frames for the optical flow estimation in the online setting.

We construct our model with a CNN encoder of architecture [64, MP, 128, MP, 256] and a decoder with deconvolutional layers (or transposed convolution) [88], [89] that can be used for learnable guided upsampling of intermediate encoder representations. Here, MP denotes a max pooling layer with stride 2. The output dimension of the embedding network $\Phi(\cdot)$ is 256, *i.e.* r=256. In MLP, the number of hidden units is [256, 256] with ReLU as the activation function for the hidden layer. The output dimension p is 10. We first initialize our auto-encoder by pre-training 10 epochs on *DAVIS$_{16}$* `val`, which takes about 7 minutes for *DAVIS$_{16}$* with 480×854 resolution. The whole network is trained using the Adam [90] optimizer (β_1=0.9 and β_2=0.999) with a learning rate of 10^{-3}. The hyper-parameters are set empirically to: κ=0.05, δ=0.1, η=0.5, λ_1=λ_2=λ_3=0.01, and $T_{\max}$=100. We also discuss the impact of different values of the hyper-parameters in Section IV-B.

B. Ablation Studies

To demonstrate the influence of each component and hyper-parameters in our method, we perform an ablation study on the

TABLE I: Comparison of the three different optical flow methods as the input on the *DAVIS*$_{16}$ dataset, measured by the mean $\mathcal{J}$. In the inference step, we employ multi-scale and CRF to improve the final performance of MG [26].

Flow	Method	Mean $\mathcal{J}\uparrow$
PWC-Net [91]	MG [26]	63.7
	Ours	67.9 (+4.2)
RAFT [86]	MG [26]	68.3
	Ours	72.0 (+3.7)
FlowFormer [87]	MG [26]	70.3
	Ours	75.4 (+5.1)

TABLE II: Ablation study of the spatial attention module on the *DAVIS*$_{16}$ dataset, measured by the mean $\mathcal{J}$. We employ an auto-encoder, which is implemented by directly connecting the encoder $\Phi(\cdot)$ and the decoder $\Psi(\cdot)$ as the baseline for all experiments, denoted as AE.

Network Variant	Mean $\mathcal{J}\uparrow$
AE (baseline)	72.1
AE *w/.* max pooling	74.5 (+2.4)
AE *w/.* average pooling	74.3 (+2.2)
AE *w/.* $A(\cdot)$	75.4 (+3.3)

DAVIS$_{16}$ `val` set. The evaluation criterion is the mean region similarity ($\mathcal{J}$).

Choice of optical flow algorithm. Our model takes only the optical flow as the input to solve the motion grouping problem. Table I shows the effect of the quality of different inputs. With the same optical flow estimation methods (*i.e.*, PWC-Net [91], RAFT [86], and FlowFormer [87]), our proposed algorithm outperforms MG [26] by **4.2**%, **3.7**%, and **5.1**% points, respectively, in terms of mean $\mathcal{J}$ on the *DAVIS*$_{16}$ `val` set. The improved optical flow model (FlowFormer) further enlarges performance gains. Thus, the optical flow estimated by the FlowFormer is the input to our model.

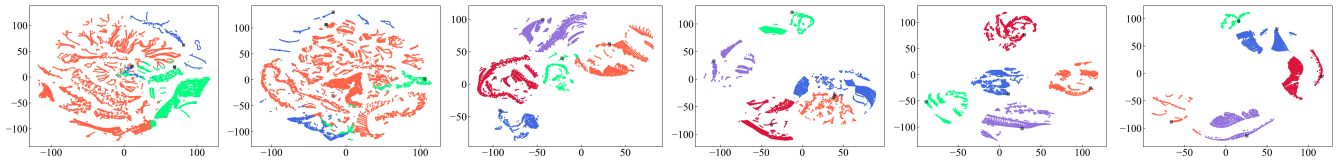
Effectiveness of spatial attention module. To verify the effect of the spatial attention module $A(\cdot)$ in Eq. 2, we gradually remove the spatial attention module $A(\cdot)$, the average pooling, and max pooling in our auto-encoder, denoted as AE, *w/.* max pooling, and *w/.* average pooling, respectively. This means that the embedding features $\boldsymbol{Z}$ are directly fed into the decoder $\Psi(\cdot)$, where we employ AE as the baseline for the ablation study. The results can be referred to in Table II. Compared to the baseline, the variants with max pooling and average pooling can independently boost the performance by **2.4**% and **2.2**%, respectively. Based on these high-performance variants, the spatial attention module $A(\cdot)$, which combines max and average pooling operations to enhance losing important information on the regions of object boundaries, further improves the performance by **3.3**% in terms of mean $\mathcal{J}$. This demonstrates the superiority of the spatial attention module.

Training objective. We investigate our overall training objective (Eq. 17). As shown in Table IIIa, the model with $\mathcal{L}_c$ alone achieves a mean $\mathcal{J}$ score of 73.9%. Adding $\mathcal{L}_{sc}$ brings a gain (*i.e.*, **0.5**%), which shows that it effectively improves the discriminability of foreground and background. After applying $\mathcal{L}_{pc}$ or $\mathcal{L}_{cc}$ individually, we observe that our model achieves improvements (*i.e.*, **0.4**%/**0.2**%), and their combinations further improve the performance by nearly **1.0**%. These facts not only demonstrate the effectiveness of $\mathcal{L}_{pc}$ and $\mathcal{L}_{cc}$ but also indicate that the contributions of the two constraints are almost orthogonal. Finally, combining all the losses together leads to the best performance, yielding a mean $\mathcal{J}$ score of 75.4%. This further confirms the effectiveness of our training objective.

Initialization of prototypes. We evaluate the different initialization strategies of the prototypes on the *DAVIS*$_{16}$ dataset to get a better impression of the performance. Table IIIb shows the results of prototypes initialized by the vector $\mathbf{0}$, the vector $\mathbf{1}$, the orthogonal vectors [93], the uniform distribution $\mathcal{U}(0,1)$, the standard normal distribution $\mathcal{N}(0,1)$, and the truncated normal distribution $\mathcal{N}(0,1)$. We notice that the prototypes filled with vectors $\mathbf{0}$ and $\mathbf{1}$ yield the worst performance compared to initializing the prototypes randomly. An improper initialization of the prototypes is problematic, and the constant initialization cannot guarantee the orthogonality of the prototypes and prevent all of them from collapsing onto a single point. It reveals that our method is sensitive to the initialization. We also compare different random initializations for the prototypes. We see that the random initialization outperforms the constant and the Gaussian initialization outperforms all the strategies. Fig. 5 presents the t-SNE [92] visualization of the learned embedded representation on the *bmx-trees* sequence from the *DAVIS*$_{16}$ dataset. In particular, without the random initialization of the prototypes, the representation learned from the auto-encoder failed to find a good clustering structure, leading to a somewhat inferior visualization. In contrast, the embedded representation leaned from the model with random initialization becomes significantly discriminative, and the proposed method can achieve promising performance when we have a good initialization of prototypes.

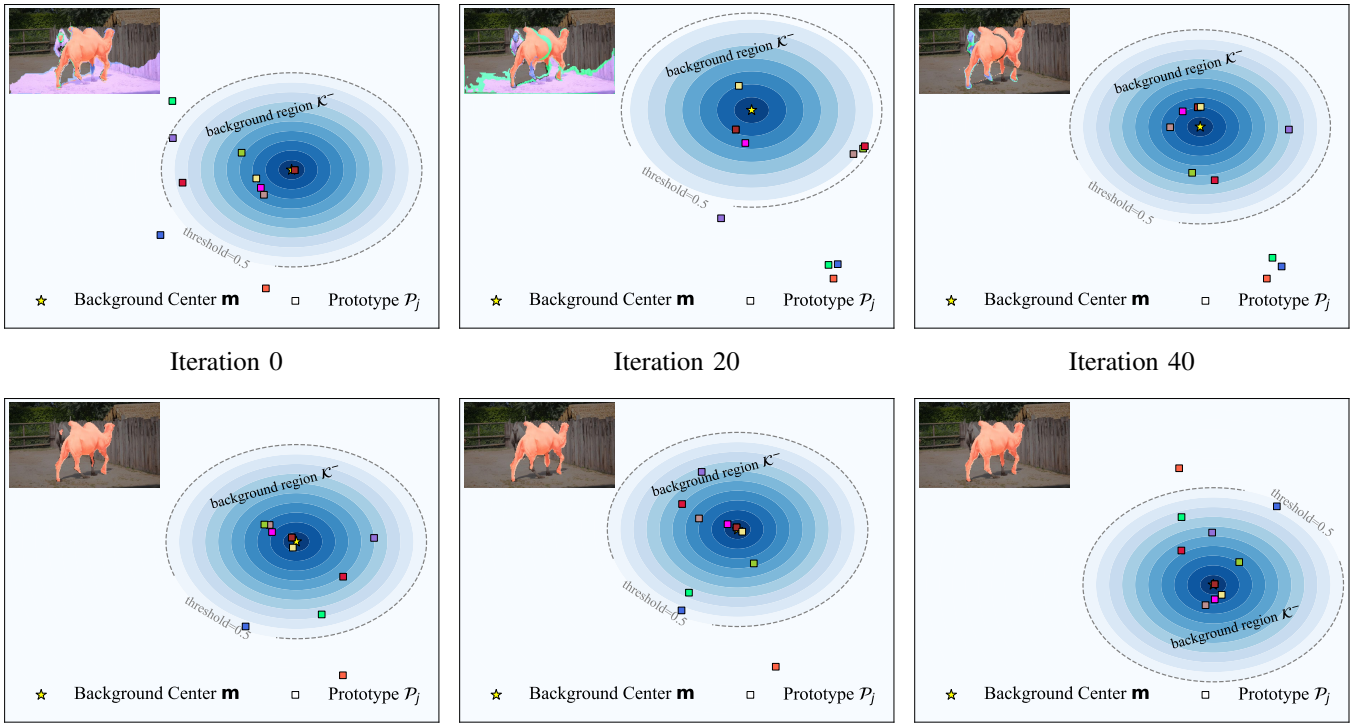
Number of clusters. Table IIIc reports the performance of our approach with regard to the number of clusters k. For k=2, we directly segment foreground-background into 2 groups. This baseline obtains a score of 65.6%. We can see that as k increases, the mean $\mathcal{J}$ first increases and then decreases. Furthermore, when we use more clusters (*i.e.*, k=5), we see a clear performance boost (65.6%→69.2%). The score improves further when k=20 or k=30 is allowed; however, increasing k beyond 30 gives marginal returns in performance. Therefore, we empirically set k=30 for a better trade-off between the accuracy and computational cost.

Background threshold. To discriminate the foreground from the background distractors, we introduce boundary motion information as prior knowledge and propose a contrastive loss based on a boundary prior to guide object segmentation. Fig. 6 shows the t-SNE [92] visualization of the embedded representation learned by the proposed method on the camel sequence from the *DAVIS*$_{16}$ dataset in different iterations, as it is important to understand how the representation evolves during training. In this scenario, when a similar object ditractor and texture background appears (*e.g.*, the small camel around the target object), our model fails to capture the primary target in early iterations. However, with the help of the contrastive loss based on a boundary prior, we see that the embedded representation of the foreground object becomes more and more discriminative as the training iterations increase. For



Vector **0** Vector **1** Orthogonal $\mathcal{U}(0,1)$ $\mathcal{N}(0,1)$ Truncated $\mathcal{N}(0,1)$

Fig. 5: Visualization of the embedded representations $\boldsymbol{Z}$ with t-SNE [92] on the *bmx-trees* sequence from the *DAVIS*$_{16}$ dataset. Note that the number of prototypes k is set to 5 for each initialization condition, and we optimize our model for 10 iterations on each frame. ■ represents the each prototype $\mathcal{P}_j$.



Iteration 0 Iteration 20 Iteration 40

Iteration 60 Iteration 80 Iteration 100

Fig. 6: Visualization of the distribution of the background region $\mathcal{K}^-$ (blue region) and the each prototype $\mathcal{P}_j$ (square) during the training iteration. Based on the cosine similarity between each prototype $\mathcal{P}_j$ and the background region $\mathcal{K}^-$, we draw a contour with a threshold of 0.5. A darker color indicates a higher similarity. Based on the observations, the distribution of prototypes is iteratively refined. The prototype of the foreground objects (■) far away from that of the background center (☆) and the background distractors (*e.g.*, ■ and ■) can be filtered by contrastive learning based on a boundary prior. The segmentation results of each iteration are shown in the upper left corner of each figure. t-SNE [92] is used to reduce the dimensionality of the features.

each video, the background region $\mathcal{K}^-$ explicitly maintains the consistency of the motion across the entire video. We empirically choose δ for the foreground saliency map decision to evaluate our model. The results are listed in Table IIId. We can see that when as δ increases, the mean $\mathcal{J}$ decreases. As a result, we empirically set δ=0.1 with the best performance.

C. Comparison with SoTAs

To widely discuss the speed-accuracy trade-offs in online methods, we show the detailed results in Table IV, with seven online UVOS methods, *e.g.*, FSEG [4], SAGE [21], SFM [43], and UOVOS [40], taken from the VOS benchmark.

***DAVIS*$_{16}$ `val`.** As shown in Table IV, our method achieves the best performance among all of the online algorithms in terms of mean $\mathcal{J}$. Compared to the second-best method UOVOS [40] which uses the pre-trained Mask R-CNN [41] to remove the moving background regions, our model achieves a gain of **0.8**% in mean $\mathcal{J}$. It is worth noting that our model relies on an auto-encoder without any other additional neural network structures to implement online subspace clustering for the UVOS. In terms of runtime efficiency, SFM [43] is the only faster online segmentation method implemented in C++. We achieve a much higher region similarity (**22.2**%) while being faster.

We also show qualitative results in Fig. 7. We choose some

TABLE III: Ablation studies of the proposed method on the *DAVIS*$_{16}$ dataset, measured by the mean $\mathcal{J}$.

(a) Training Objective $\mathcal{L}$

$\mathcal{L}_c$	$\mathcal{L}_{sc}$	$\mathcal{L}_{pc}$	$\mathcal{L}_{cc}$	Mean $\mathcal{J}\uparrow$
✓				73.9
✓	✓			74.4 (+0.5)
✓		✓		74.3 (+0.4)
✓			✓	74.1 (+0.2)
✓		✓	✓	74.7 (+0.8)
✓	✓	✓	✓	**75.4** (+1.5)

(b) Initialization of prototypes

Init. Method	Mean $\mathcal{J}\uparrow$
Vector $\mathbf{0}$	44.4
Vector $\mathbf{1}$	43.5
Orthogonal [93]	75.1
$\mathcal{U}(0,1)$	74.8
$\mathcal{N}(0,1)$	**75.4**
Truncated $\mathcal{N}(0,1)$	75.3

(c) Number of clusters k

Clusters k	Mean $\mathcal{J}\uparrow$
k=2	65.6
k=5	69.2
k=10	73.9
k=20	75.2
k=30	**75.4**
k=50	74.9

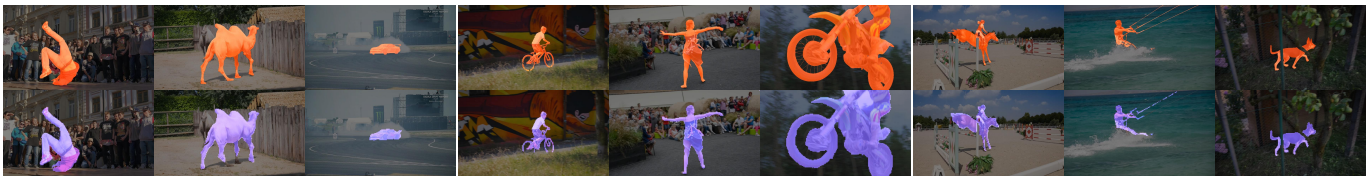
(d) Background threshold δ

Threshold δ	Mean $\mathcal{J}\uparrow$
δ=0.05	74.4
δ=0.1	**75.4**
δ=0.2	74.6
δ=0.5	73.7
δ=0.7	73.2
δ=0.9	72.9

TABLE IV: Quantitative results on the `val` set of video object segmentation benchmarks, using the region similarity $\mathcal{J}$. The best performance scores are highlighted in **bold**. The Extra Model in the fifth column denotes the required pre-training model. Runtime excludes the optical flow computation. OF and RGB represent the optical flow and RGB image, respectively.

Method	Training & Optimization	OF	RGB	Input Resolution	Extra Model	Mean $\mathcal{J}\uparrow$ *DAVIS*$_{16}$	*FBMS*	*SegTrackV2*	Runtime $\downarrow$ (s/frames)
FSEG [4]	•	✓	✓	480×854	-	71.6	-	61.4	7.2
LMP [38]	•	✓		480×854	-	64.5	-	-	18.3
ELM [18]	★	✓	✓	-	-	61.8	61.6	-	-
SAGE [21]	★	✓		-	-	42.6	61.2	57.6	0.9
CVOS [42]	★	✓		480×854	-	51.4	-	-	-
SFM [43]	★		✓	480×854	-	53.2	-	-	0.15
UOVOS [40]	★	✓	✓	480×854	Mask R-CNN [41]	74.6	63.9	61.5	0.36
Ours	★	✓		480×854	-	**75.4**	**66.8**	**62.6**	**0.12**

★ is fully unsupervised and the model is optimized in a single input frame. • is pre-trained on a video dataset.



Dynamic background Motion blur Occlusion

Fig. 7: Qualitative results of the proposed method on challenging scenarios from the *DAVIS*$_{16}$. From left to right: dynamic background (*breakdance*, *camel*, and *drift-chicane*), motion blur (*bmx-trees*, *dance-twirl*, and *motocross-jump*), and occlusion (*horsejump-high*, *kite-surf*, and *libby*). The ground truth is shown in the top row, and our results are shown in the bottom row.

videos from the *DAVIS*$_{16}$ dataset with the cases of dynamic background, motion blur, and occlusion. It can be seen that our model can handle different challenges. For example, our method can segment foreground objects when they are occluded by the background, as shown in the occlusion case in Fig. 7. When a similar object distractor appears (*e.g.*, the crowd in *breakdance*, or the small camel in *camel*), our method is able to discriminate a foreground target from background distractors.

***FBMS* `val`.** As shown in Table IV, our method significantly outperforms all previous published works on the *FBMS* `val` set compared to online UVOS methods. For instance, on the mean $\mathcal{J}$ metric, our method surpasses UOVOS [40] by **2.9**% and SAGE [21] by **5.2**%. In comparison to the performance on *DAVIS*$_{16}$, our method has a certain gap (*i.e.*, **8.6**%) on the *FBMS* dataset. This is because our method relies only on the optical flow, and some sequences of the *FBMS* dataset contain multiple objects in a single video. In these challenging videos, only a subset of objects are moving, so it is difficult to determine all the objects by optical flow without considering other cues.

***SegTrackV2* `val`.** We also report the performance of the low-resolution dataset in Table IV. Compared to the high-resolution *DAVIS*$_{16}$ dataset, it is more difficult to train an accurate optical flow model on the *SegTrackV2* dataset. Our algorithm outperforms the online methods, *i.e.*, SAGE [21], FSEG [4], and UOVOS [40], by **5.0**%, **1.2**%, and **1.1**%, respectively, in terms of mean $\mathcal{J}$, respectively. However, compared to the high-resolution datasets (*i.e.*, *DAVIS*$_{16}$ and *FBMS*), our method performs worse on the *SegTrackV2* dataset for the following reasons: 1) we have only grouped the pixels that possess the same motion pattern based on the optical flow, which significantly limits the model in segmenting objects when the flow is incomplete; and 2) some of the low-resolution videos from the *SegTrackV2* dataset affect the performance of our model, which only uses the optical flow as its input. This effect is also seen in FSEG [4], SAGE [21], and UOVOS [40]. In particular, since UOVOS [40] detects the foreground object based on a salient motion map, using Mask R-CNN on incomplete optical flow provides little improvement.

D. Runtime Comparison

To further investigate the computational efficiency of our proposed method, we report the inference time comparisons on the $DAVIS_{16}$ datasets at 480p resolution. We compare our model with the SoTA online methods that share their codes or include the corresponding experimental results, including the SFM [43], and UOVOS [40]. For the inference time comparison, we run the public code of other methods and our code under the same conditions on the NVIDIA TITAN RTX GPU. The analysis results are summarized in the last column of Table IV.

As shown in Table IV, our algorithm shows a faster speed than other competitors. For online UVOS settings, model efficiency is an important metric. Our model achieves a more favorable accuracy-efficiency trade-off than the existing best online method UOVOS [40], while achieving higher accuracy. The main computational cost of UOVOS [40] lies in the object proposal component, which is based on Mask R-CNN [41]. However, our model relies on an auto-encoder and online clustering strategy for the UVOS without any other additional neural network structures. Compared to the faster online method SFM [43], our model achieves a **22.2**% higher mean $\mathcal{J}$.

V. Conclusion Remarks and Discussions

In this paper, an efficient contrastive subspace motion clustering is proposed for online unsupervised video object segmentation (UVOS) by exploring an online clustering strategy for motion grouping. Specifically, non-learnable prototypical bases are iteratively summarized from the feature space for different motion patterns, and these bases help to optimize the feature representation in return. Experimental results demonstrated that our method outperforms state-of-the-art (SoTA) online UVOS algorithms.

In real-world scenarios, the performance of our online UVOS system may be violated due to the presence of low-resolution input, making it inaccurate for small objects. Therefore, we can see that our method performs worse on the low-resolution dataset, *i.e.*, *SegTrackV2*, compared to the high quality video data. The saturation of moving objects, such as white vehicles under sunshine and black vehicles in dim lighting conditions, will also affect the proposed UVOS method. The problem will be investigated by utilizing the results in [71] and [44], [94], respectively. Furthermore, considering the significance of real-time aspects in online UVOS, we will incorporate a light-weighted design [95], [96] in our future research.

References

[1] M. Wertheimer, "Untersuchungen zur lehre von der gestalt. ii," *Psychologische forschung*, vol. 4, no. 1, pp. 301–350, 1923.

[2] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 4491–4500.

[3] P. Tokmakov, C. Schmid, and K. Alahari, "Learning to segment moving objects," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 282–301, 2019.

[4] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2126.

[5] M. Siam, C. Jiang, S. Lu, L. Petrich, M. Gamal, M. Elhoseiny, and M. Jagersand, "Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 50–56.

[6] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen, "MATNet: Motion-attentive transition network for zero-shot video object segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 8326–8338, 2020.

[7] M. Faisal, I. Akhter, M. Ali, and R. Hartley, "EpO-Net: Exploiting geometric constraints on dense trajectories for motion saliency," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1873–1882.

[8] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. H. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3059–3069.

[9] W. Wang, J. Shen, X. Lu, S. C. H. Hoi, and H. Ling, "Paying attention to video object pattern understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2413–2428, 2021.

[10] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3618–3627.

[11] X. Lu, W. Wang, J. Shen, D. Crandall, and J. Luo, "Zero-shot video object segmentation with co-attention siamese networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2228–2242, 2022.

[12] W. Wang, X. Lu, J. Shen, D. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9235–9244.

[13] X. Lu, W. Wang, J. Shen, D. J. Crandall, and L. Van Gool, "Segmenting objects from relational visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7885–7897, 2022.

[14] Y. Zhou, X. Xu, F. Shen, X. Zhu, and H. T. Shen, "Flow-edge guided unsupervised video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8116–8127, 2022.

[15] L. Xi, W. Chen, X. Wu, Z. Liu, and Z. Li, "Implicit motion-compensated network for unsupervised video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6279–6292, 2022.

[16] S. Ren, W. Liu, Y. Liu, H. Chen, G. Han, and S. He, "Reciprocal transformations for unsupervised video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 455–15 464.

[17] K. Zhang, Z. Zhao, D. Liu, Q. Liu, and B. Liu, "Deep transport network for unsupervised video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 8781–8790.

[18] D. Lao and G. Sundaramoorthi, "Extending layered models to 3d motion," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 441–457.

[19] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013, pp. 1777–1784.

[20] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 2, 2014.

[21] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3395–3402.

[22] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 786–802.

[23] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised moving object detection via contextual information separation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 879–888.

[24] B. Luo, H. Li, F. Meng, Q. Wu, and K. N. Ngan, "An unsupervised method to extract video object via complexity awareness and object local parts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 7, pp. 1580–1594, 2018.

[25] Y. Yang, B. Lai, and S. Soatto, "DyStaB: Unsupervised object segmentation via dynamic-static bootstrapping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2826–2836.

[26] C. Yang, H. Lamdouar, E. Lu, A. Zisserman, and W. Xie, "Self-supervised video object segmentation by motion grouping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 7177–7188.

[27] V. Ye, Z. Li, R. Tucker, A. Kanazawa, and N. Snavely, "Deformable sprites for unsupervised video decomposition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2647–2656.

[28] Y. Liu, Z. Li, and Y. Soh, "Region-of-interest based resource allocation for conversational video communication of h.264/avc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.

[29] Y. Liu, Z. G. Li, and Y. C. Soh, "A novel rate control scheme for low delay video communication of h.264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 1, pp. 68–78, 2007.

[30] Z. Li, C. Zhu, N. Ling, X. Yang, G. Feng, S. Wu, and F. Pan, "A unified architecture for real-time video-coding systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 472–487, 2003.

[31] Z. Li, F. Pan, G. Feng, K. P. Lim, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for jvt," in *JVT 7th Meeting, Pattaya, Mar2003*, 2003, JVT-G012.

[32] X. Song and G. Fan, "Joint key-frame extraction and object segmentation for content-based video analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 904–914, 2006.

[33] J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, and J. Feng, "Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing," in *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2018, p. 792–800.

[34] J. Li, J. Zhao, C. Lang, Y. Li, Y. Wei, G. Guo, T. Sim, S. Yan, and J. Feng, "Multi-human parsing with a graph-based generative adversarial model," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 1, pp. 29:1–29:21, 2021.

[35] J. Zhao, J. Li, X. Nie, F. Zhao, Y. Chen, Z. Wang, J. Feng, and S. Yan, "Self-supervised neural aggregation networks for human parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1595–1603.

[36] J. Li, J. Zhao, Y. Chen, S. Roy, S. Yan, J. Feng, and T. Sim, "Multi-human parsing machines," in *Proceedings of the ACM International Conference on Multimedia (ACM MM)s*, 2018, p. 45–53.

[37] A. Criminisi, T. Sharp, C. Rother, and P. P'erez, "Geodesic image and video editing," *ACM Transactions on Graphics*, vol. 29, no. 5, 2010.

[38] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 531–539.

[39] P. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 9905, 2016, pp. 75–91.

[40] T. Zhuo, Z. Cheng, P. Zhang, Y. Wong, and M. Kankanhalli, "Unsupervised online video object segmentation with motion property understanding," *IEEE Transactions on Image Processing*, pp. 237–249, 2020.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[42] B. Taylor, V. Karasev, and S. Soattoc, "Causal video object segmentation from persistence of occlusions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4268–4276.

[43] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 733–740.

[44] Y. Xu, Z. Liu, X. Wu, W. Chen, C. Wen, and Z. Li, "Deep joint demosaicing and high dynamic range imaging within a single shot," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4255–4270, 2022.

[45] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2011, pp. 109–117.

[46] Y. Hu, J. Huang, and A. G. Schwing, "Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 11205, 2018, pp. 813–830.

[47] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1998, pp. 1154–1160.

[48] M. Keuper, "Higher-order minimum cost lifted multicuts for motion segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, pp. 4252–4260.

[49] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015, pp. 3271–3279.

[50] P. Ochs and T. Brox, "Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011, pp. 1583–1590.

[51] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

[52] Y. Weiss, "Smoothness in layers: Motion segmentation using nonparametric mixture estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 520–526.

[53] M. Pawan Kumar, P. H. Torr, and A. Zisserman, "Learning layered motion segmentations of video," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 301–319, 2008.

[54] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 282–295.

[55] P. Ochs and T. Brox, "Higher order motion models and spectral clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 614–621.

[56] C. Xie, Y. Xiang, Z. Harchaoui, and D. Fox, "Object discovery in videos as foreground motion clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9986–9995.

[57] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5320–5329.

[58] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3491–3500.

[59] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9225–9234.

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

[61] H. K. Cheng and A. G. Schwing, "XMem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 640–658.

[62] D. Wu, X. Dong, L. Shao, and J. Shen, "Multi-level representation learning with semantic alignment for referring video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4986–4995.

[63] H. Ding, C. Liu, S. Wang, and X. Jiang, "Vision-language transformer and query generation for referring segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 16 301–16 310.

[64] S. Seo, J.-Y. Lee, and B. Han, "Urvos: Unified referring video object segmentation network with a large-scale benchmark," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 208–223.

[65] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, pp. 686–695.

[66] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proceedings of*

the European Conference on Computer Vision (ECCV), September 2018, pp. 744–760.
[67] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. H. Hoi, "Learning video object segmentation from unlabeled videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8957–8967.
[68] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 11 525–11 538, 2020.
[69] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.
[70] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.
[71] Z. Liu, Z. Li, X. Wu, Z. Liu, and W. Chen, "Dsrgan: Detail prior-assisted perceptual single image super-resolution via generative adversarial networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7418–7431, 2022.
[72] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, and Z. Lin, "Self-supervised convolutional subspace clustering network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5468–5477.
[73] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, p. 23–32.
[74] T. Zhang, P. Ji, M. Harandi, R. Hartley, and I. Reid, "Scalable deep k-subspace clustering," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2019, pp. 466–481.
[75] J. Fan, "Large-scale subspace clustering via k-factorization," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, p. 342–352.
[76] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang, "Efficient deep embedded subspace clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 21–30.
[77] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 119, 2022, pp. 1597–1607.
[78] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 119, 2020, pp. 9929–9939.
[79] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 8547–8555.
[80] G. Peyré and M. Cuturi, "Computational optimal transport," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.
[81] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 2292–2300.
[82] P. Knopp and R. Sinkhorn, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343 – 348, 1967.
[83] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 724–732.
[84] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.
[85] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2192–2199.
[86] Z. Teed and J. Deng, "RAFT: recurrent all-pairs field transforms for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 12347, 2020, pp. 402–419.
[87] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 13677, 2022, pp. 668–685.
[88] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2018–2025.
[89] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1520–1528.
[90] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
[91] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.
[92] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
[93] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.
[94] C. Zheng, Z. Li, Y. Yang, and S. Wu, "Single image brightening via multi-scale exposure fusion with hybrid learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1425–1435, 2021.
[95] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled siamese networks for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8896–8909, 2022.
[96] Z. Zhao, S. Zhao, and J. Shen, "Real-time and light-weighted unsupervised video object segmentation network," *Pattern Recognition*, vol. 120, p. 108120, 2021.

Lin Xi received my B.Eng. degree in Measurement and Control Technology and Instrumentation from Beijing University of Chemical Technology (BUCT), Beijing, China, in 2014 and M.E. degree in Mechanical Engineering from BUCT, Beijing, China, in 2017. He is a Ph.D. candidate in the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, under the supervision of Prof. Weihai Chen. His research interests include video object segmentation and non-contact physiological signal monitoring.

Weihai Chen (M'00) received the B.Eng. degree from Zhejiang University, China, in 1982, and the M.Eng. and Ph.D. degrees from Beihang University, China, in 1988 and 1996, respectively. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191. He has published more than 200 technical papers in referred journals and conference proceedings and filed more than 20 patents. His research interests include computer vision, image processing, automation, bio-inspired robotics, and precision mechanism.

Prof. Chen is the recipient of the 2017 Best Paper Award for the IEEE Transactions on Industrial Electronics and the 2018 IET Premium Award for the IET Intelligent Transport Systems. He is a member of the Technical Committee on Manufacturing Automation of the IEEE Robotics and Automation Society.

Xingming Wu received the B.Eng. and M.Eng. degrees from Zhejiang University, China, in 1981 and 1985, respectively. He is currently a Professor with the School of Automation Science and Electronic Engineering, Beihang University, China. His main research directions are image processing, scene understanding, visual mapping, autonomous mobile robots, and embedded systems.

Zhong Liu received the M.S. degree and the Ph.D. degree from Beihang University, Beijing, China, in 2006 and 2018, respectively. He is currently an Associate Professor with the School of Automation Science and Electronic Engineering, Beihang University, China. His current research interests include computer vision, saliency detection and computer control.

Zhengguo Li (F'23) received the B.Sci (Applied Mathematics). and M.Eng. (Automatic Control) from Northeastern University, Shenyang, China, in 1992 and 1995, respectively, and the Ph.D. degree (Automatic Control) from Nanyang Technological University, Singapore, in 2001.

His research interests include video processing and delivery, computational photography, switched and impulsive control, sensor fusion and physics-driven deep learning. He has co-authored one monograph, more than 200 journal/conference papers including more than 60 IEEE Transactions, and eleven granted USA patents, including normative technologies on scalable extension of H.264/AVC and HEVC. He has been actively involved in the development of H.264/AVC and HEVC since 2002. He had three informative proposals adopted by the H.264/AVC and three normative proposals adopted by the HEVC. Currently, he is with the Agency for Science, Technology and Research, Singapore.

Zhengguo served as a sponsorship/exhibition chair of IEEE ISCAS 2024, TPC Chair of IEEE IECON 2023 and 2020, a special session chair of IEEE ICASSP 2022, a Technical Brief Co-Founder of SIGGRAPH Asia, and leading Workshop Chair of IEEE ICME in 2013. He was an Associate Editor of IEEE Signal Processing Letters from 2014-2016, IEEE Transactions on Image Processing from 2016-2020, IEEE Transactions on Circuits and Systems for Video Technology from 2020-2023, APSIPA Transactions on Signal and Information Processing from 2022-2025, and a Senior Area Editor of IEEE Transactions on Image Processing from 2020-2024.