IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 12, 2019, accepted January 17, 2020, date of publication February 6, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971964

Video Object Segmentation by Latent Outcome Regression

LIN ZHANG AND YAO LU

Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Yao Lu (vis_yl@bit.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61273273, and in part by the National Key Research and Development Plan of China under Grant 2017YFC0112001.

ABSTRACT This paper presents a novel algorithm for unsupervised video object segmentation (UVOS) in unconstrained scenarios. Although a large variety of methods have been proposed in the literature, segmenting generic objects is still challenging because different methods often perform well in different situations, and no single method can outperform the others in all cases. To address this, we propose to solve the problem of UVOS in a crowd-sourcing setting. We claim that one can achieve superior results by aggregating the predictions of multiple imperfect methods in a reasonable way. Specifically, we propose a latent regression algorithm for ensemble-based segmentation by jointly labelling pixels in a sequence and learning an adaptive weight for each single method in an ensemble. The pixel labellings offer the outcome (pseudo groundtruth) for regression and thus promote the procedure of weight learning, while the learnt weights could provide better shape priors for labelling, resulting in more accurate segmentation. Besides, Laplacian regularization is introduced into the regression to facilitate a stable learning of the weights. The most distinct feature of our algorithm is that it adaptively learns the contributions of different single methods for each test sequence, thus is capable of capturing the advantages of those methods while avoiding their weaknesses. In the experiments, our algorithm is built on 14 non-deep learning segmentation methods which are based on handcrafted features and require no training data. Experimental results on popular benchmarks show that our algorithm achieves compelling performance, even in comparison with deep learning-based methods. Furthermore, benefiting from the adaptive weight learning mechanism, our algorithm can achieve good flexibility and usability by choosing the most complementary single methods without losing too much performance.

INDEX TERMS Video object segmentation, latent regression, appearance modelling, unsupervised.

I. INTRODUCTION

Video object segmentation refers to the segmentation of primary objects across frames in a video clip. It is a fundamental research topic in the video analysis field and has a wide range of applications including video summarization [1], object tracking [2], video retrieval [3] and many more.

In the past decades, many methods have been proposed to address this task, which can be organized according to the level of human supervision they assume: 1) *Unsupervised* methods produce coherent space-time segments from bottom up, without human intervention. Since these methods have no prior assumption about the object to be segmented, some of them attempt to discover the primary object using motion, appearance or their combination [4]–[11], while some others firstly propose object-like regions [12] and then optimize their temporal connections in a locally [13]–[16] or fully connected graph [17] to generate space-time regions. 2) *Semi-supervised* methods take an object delineation manually in the initial frame and then propagate it to the remaining frames. Most prior methods formulate the task as an energy optimization problem over pixel-based [6], [18] or superpixel-based [19]–[21] graphs with frame-to-frame connection. To encourage long-range interactions, [22] introduces higher order supervoxel potential to guide the propagation towards broad spatio-temporal regions, while [23] builds a fully connected graph over object proposals extracted in all frames. 3) *Supervised*

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

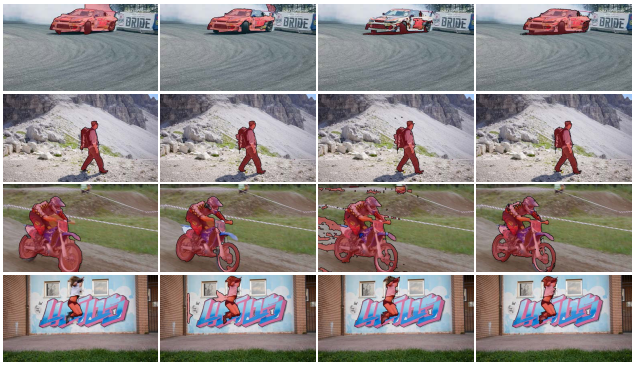

FIGURE 1. Segmentation results of representative frames in 4 sequences for four methods (From left to right: CVOS [9], JMP [25], NLC [17] and our method). We can see that the three compared methods may produce accurate segments in some sequences, but may perform poorly in others. By contrast, our method achieves superior results in all situations. (Best viewed in color).

methods are labor-intensive because they need a user in the loop to correct the segmentation results [24]–[26], [28], [29]. These methods are restricted to some specific applications, *e.g.* video post-production, due to such intensive supervision.

Despite remarkable progress in recent years, video object segmentation in unknown scenarios is still challenging due to the diversity of possible difficulties, such as object appearance variations, cluttered background, interrupting objects, motion blur and partial or complete occlusions. Many existing methods can deal with one or several factors, but few can consider them all. For example, NLC [17] is robust to many challenges, e.g. dynamic background, fast motion, motion blur and occlusions, however, its performance suffer a lot from large appearance changes of objects. By contrast, CVOS [9] can deal with appearance variations but it is not robust enough to inconspicuous motion. In Figure 1, we show some results of four methods over 4 sequences. We see that NLC and JMP often fail in cases that the objects undergo large changes of scale or appearance (Row #1 and Row #3). CVOS produces inaccurate segmentation under large displacement (Row #4). The imperfect segmentation results occur mainly because each method makes particular assumptions about the object to be segmented and thus will fail once the assumptions are violated. We see in Figure 1 that different methods master different challenges and thus can complement each other well. As a result, one can obtain more accurate segmentation by combining these methods in an appropriate way to exploit such complementary roles.

Motivated by these observations, we argue that an algorithm combining the strengths of different methods and avoiding their weaknesses could outperform any single method. One common strategy for the combination is to aggregate the solutions of multiple methods with different associated weights, in which a larger weight implies that the corresponding method is more powerful. The strategy has shown impressive performance on several vision tasks, *e.g.* saliency detection [30], [31], visual tracking [32] and video segmentation [33]. These works learn fixed weights from training data, and thus are inflexible for practical applications because if we want to add a new method, we have to re-train the model. Besides, aggregating with fixed weights is sub-optimal since different methods often perform on average different well among test data. Last, training data is expensive in some situations, *e.g.* video segmentation, making the data-driven approaches unfeasible.

In our previous work [33], we propose to solve the task of video object segmentation by data fusion. The method accepts a set of segment tracks as input, each of which is given by a segmentation method, and then fuse the segment tracks into a more accurate one. Although the method shows outstanding performance on the SegTrack dataset [34], it still has several limitations: 1) it considers that each single method contributes equally to the final result and combines them with equal weights, thus fails to exploit the complementary roles among the methods; 2) it estimates shape priors of objects by majority voting which may give poor estimation once most single methods fail to segment the objects.

In this work, we propose an unsupervised aggregation algorithm for video object segmentation, *i.e.*, that does not require any training data for learning the weights of single methods. Our algorithm is built on linear regression with a ℓ_2 loss that measures the discrepancy between the pseudo ground-truth, *i.e.* the outcome of the regressor, with the ensemble result of single methods. We propose to jointly learn segmentation labels and weights of single methods in the ensemble within a unified framework. More specifically, the framework consists of two modules, *i.e. segmentation inference* and *weight learning*. The segmentation inference module performs pixel-wise labeling within a Markov Random Field (MRF) incorporating appearance and shape priors. It provides the outcome for regression to learn the weights. In turn, the weight learning module provides shape priors for more accurate label inference. The two modules are repeated in an iterative way until convergence for each test video. Furthermore, we devise an approach to initialize the weights by evaluating for each segmentation its confidence belonging to foreground regions. We show that the approach yields good initialization for the weights, thereby accelerating the rate of convergence.

In summary, our approach has the following contributions and characteristics:

- We propose a novel framework for unsupervised video segmentation by solving a regularized regression problem with latent outcome.
- We present a robust approach to determine the initial weights which enables fast convergence of our algorithm.
- Extensive empirical results on two challenging benchmarks demonstrate that our approach outperforms both non-deep learning and deep learning based UVOS methods.

The paper is organized as follows. We briefly introduce the related work in Section II. The main framework of the

algorithm is described in Section III followed by the experimental result in Section IV. Finally, we draw conclusions in Section V.

II. RELATED WORK

We provide a brief overview of recent works in two relevant fields: video object segmentation and aggregation-based vision tasks.

A. VIDEO OBJECT SEGMENTATION

According to the level of human supervision, video object segmentation can be categorized into unsupervised, semi-supervised and supervised methods. For each category, we review both non-deep learning and deep learning based methods.

Unsupervised methods produce coherent space-time regions from the bottom up, without any user intervention. Since unsupervised methods make no prior assumption about objects to be segmented, they often discover primary objects using motion (*e.g.* optical flow for short term [7], trajectory for long term [8], [35]), appearance cues [5], [6], [12] or their combination [9], [10]. The motion based segmentation algorithms extract foreground objects on account of the noticeable moving of objects against the background. For instance, FST [7] generates optical flow boundaries to locate the objects. However, optical flow only captures short-term motion information which is not robust to large displacement or object occlusion. To address this, many methods [4], [8] replace optical flow with long-term point trajectory and formulate video segmentation as a spectral clustering problem. The appearance based methods generally consider saliency [10], [36], [37] and objectness [8], [13], [15] referring to dynamic human fixation and generic object-like regions [12], respectively. For example, Wang *et al.* [10] exploit spatio-temporal edges to estimate geodesic saliency and the most salient regions are considered as objects. Fukuchi *et al.* [37] utilize an attention model to calculate the visual attention density. Benefiting from the development of object proposal techniques [12], [38]–[40], a large number of methods [13]–[16], [41] build the spatio-temporal relationship among the proposals, and discover objects with graph-based algorithms.

Convolution neural networks (CNN) have been widely used in unsupervised video object segmentation [42]–[47]. Many deep learning based methods [45]–[47] learn object appearance and motion features separately in two CNN branches. In [45], optical flow and segmentation prediction are computed in a unified framework. Reference [47] designs a convolutional GRU module to fuse the appearance and the motion branch. In contrast, Song *et al.* [42] use convolutional LSTM module to combine appearance features of five consecutive images as motion module to segment motion salient foreground objects.

Semi-supervised methods take an object delineated manually in the first frame and then propagate it to the remaining frames. TSP [19] over-segments the annotated frame into superpixels and tracks the superpixels belonging to object regions across time. TSP is robust to objects with distinct colors, however, it may fail in large objects appearance variations and occlusions. Therefore, Wen *et al.* [21] integrate the multi-part tracking and segmentation into a unified energy optimization framework to refine segmentation. Tsai *et al.* [18] propose to compute the optical flow and segmentation mask simultaneously. They build a multi-scale appearance model to assist optical flow learning during segmentation propagation, producing more accurate segmentation results.

There are also many deep learning-based works using semi-supervised setting. MSK [48] propagates object masks by the output of previous frame as well as optical flow information. OSVOS [43] referring to one-shot video object segmentation, finetunes a pretrained model with the annotations of the first frame to adapt the network for target objects. Luiten *et al.* [49] further update the network online, making the network more robust to appearance variations of objects. Li *et al.* [50] propose a feature propagation module to adaptively fuse features over time via spatially variant convolution. Wug *et al.* [51] use a fast deep Siamese encode-decoder network to jointly perform mask propagation and object detection. Besides, Chen *et al.* [52] formulate the segmentation task as pixel-wise retrieval in a learnt embedding space. The annotated pixels are considered as reference, while pixels in target frames are classified into foreground or background using nearest neighbor search in the embedding space.

Supervised methods usually referred to as interactive segmentation methods [24], [26], [27], [52] that require users in the loop to correct the segmentation results. Since these algorithms require human labors, researches mainly restricted in some certain areas, such as video post-production, etc.

B. AGGREGATION-BASED VISION TASKS

Aggregation algorithms have been explored to solve many vision tasks, such as saliency detection [30], [31], [53] and visual tracking [32], [54]–[57]. However, our problem is largely different from these two tasks, and thus the techniques cannot be applied to solve our problem effectively. The saliency aggregation approaches [30], [32], [53] aim to acquire the weights of multiple saliency methods by learning from training data. However, it is more difficult to obtain enough training data for a video segmentation task. In [32], the authors propose different variants of fusion methods for object tracking based on the concept of attraction field, including a weighted combination of trackers and a trajectory optimization method. These methods are also not suitable for our problem because 1) the weights for different trackers are also supervised learnt from some training data; 2) it uses temporally coherent bounding boxes to represent the trajectory of an object over time, while our goal is to determine a space-time tube whose shape may deform as the object moves. Besides, several recent works [54]–[57] combine the advantages of generative-based and discriminative-based trackers for visual tracking. The main limitations lie in that these methods can fuse just several (less than five)

trackers, and moreover, the trackers need to be carefully designed or selected to achieve high performance.

Despite extensive research on related vision tasks, it is less explored in video segmentation except the work in [33]. It aggregates the predictions of multiple methods to obtain shape information of an object, which is subsequently incorporated with holistic appearance cues into an energy minimization framework for segmentation. The algorithm only uses 6 methods for combination and is evaluated on 5 videos. In contrast, our algorithm adaptively learns optimal weights for single methods in each video. This enables us to discover the superior methods as well as to suppress the inferior ones in any situations.

III. MAIN FRAMEWORK

Given a video sequence, we aim to label pixels of the primary object as foreground and others as background. We make no assumptions about the types of objects or scene context. Let L_i be a segment track given by the i-th single method. It consists of n binary masks, one for each frame. For clarity, we represent L_i with a column vector, *i.e.* $L_i \in \mathbb{R}^{(h\times w\times n)\times 1}$, where h and w are the height and width of the video, respectively. Given the segments $L = [L_1, \ldots, L_k]^T$ of k tracks, we wish to determine the labels of all pixels in the video $\hat{L} \in \mathbb{R}^{(h\times w\times n)\times 1}$. Our algorithm is built on the assumption that $\hat{L}$ can be approximately represented by a linear combination of these tracks,

$$\hat{L} = w^T L \tag{1}$$

where $w = [w_1, \ldots, w_k]$ is the weight distribution of the tracks. Note that in our formulation, w_i can be zero or negative. And in the case $w_i < 0$ the track is considered to be adversarial, while $w_i = 0$ means the i-th method is useless for predicting $\hat{L}$.

To estimate w, previous works [30], [31], [53] often learn from training data in a supervised way where ground-truth annotations are available for training set. However, in our unsupervised setting, it will be difficult to estimate the weights without training data. Considering the mutual dependency between $\hat{L}$ and w in Eq. 1, we propose to optimize them simultaneously within a latent outcome framework, which operates in an iterative way:

- *Latent outcome estimation $\hat{L}$:* It will be difficult to estimate $\hat{L}$ without w, hence, we propose a novel weight initialization scheme in Section III-C. Then, $\hat{L}$ is estimated by solving a spatio-temporal energy minimization problem (Section III-B).
- *Adaptive weight learning w:* the coarse $\hat{L}$ is considered as the pseudo ground-truth in the regression problem. We then solve w by solving a regularized regression problem (Section III-A).

The two modules are repeated until convergence or reach the predefined number of iterations. The final outputs of our algorithm are an optimal weight configuration w of single methods and the corresponding segmentation results $\hat{L}$. Our algorithm is summarized in Algorithm 1.

Algorithm 1 Video Object Segmentation Aggregation

Input: A video $\mathcal{V}$, the predictions L of k single segmentation methods
Output: The optimal weight $\hat{w}$ and corresponding segmentation result $\hat{L}$

```
Initialize w with (15);
while not converged do
    Given w, estimate L by graph cut to minimize (8) based on
        - Object appearance (10)
        - Shape priors (10)
    Given L, learn w by optimizing (2) regularized by
        - Laplacian term (6)
        - Temporal term (7)
return L̂, ŵ;
```

A. REGRESSION WITH LATENT OUTCOME

Assuming $\hat{L}$ is given, we obtain w by solving a minimization problem with the following form

$$\min_w \mathcal{L}(w, \hat{L}, L) + \lambda_1 \Omega_l(w, L) + \lambda_2 \Omega_t(w, L) \tag{2}$$

where $\mathcal{L}$ is a loss function, Ω_l and Ω_t are two regularization terms. λ_1 and λ_2 are scalar weights reflecting the influence of the two regularizars.

The loss function $\mathcal{L}$ is established by introducing a penalty matrix Λ into the l_2 loss

$$\mathcal{L}(w, \hat{L}, L) = (w^T L - \hat{L})^T \Lambda (w^T L - \hat{L}) \tag{3}$$

where $\Lambda \in R^{(h\times w\times n)\times(h\times w\times n)}$ is a diagonal matrix with each diagonal element being the penalty for the corresponding pixel. Intuitively, for pixels with high confidence to be foreground (or background), we should heavily penalize w with which the regressor leads to inconsistent labels. In contrast, for the ambiguous pixels which we cannot determine their labels, we should penalize them moderately.

Denote $\mathcal{S}_f$ and $\mathcal{S}_b$ be the set of high confident foreground and background pixel seeds, respectively. They are estimated according to the probability of the pixels computed using the learnt weight w' in the previous iteration:

$$\mathcal{S}_f = \{i : L'_i > \tau\} \quad \mathcal{S}_b = \{i : L'_i = 0\} \tag{4}$$

where $L' = w^T L$ and L_i denotes the foreground probability of pixel i. Benefiting from the diversity of segments, we can accurately determine $\mathcal{S}_b$ in this way. However, the estimation of $\mathcal{S}_f$ will be sensitive to the threshold τ in some challenging situations. To address this problem, we apply the meanshift clustering method on all pixels in $\mathcal{S}_f$ using RGB color features and only keep the pixels in the largest cluster as our foreground seeds. Finally, we set each diagonal element $\Lambda_{i,i}$ as

$$\Lambda_{i,i} = \begin{cases} \gamma & \text{if} \quad i \in \mathcal{S}_f, \\ 1 & \text{if} \quad i \in \mathcal{S}_b, \\ 1-\gamma & \text{otherwise.} \end{cases}$$

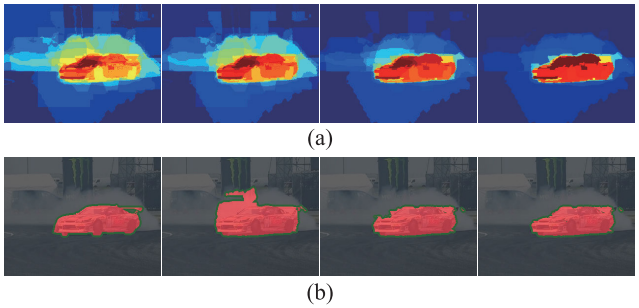

FIGURE 2. Examples showing shape priors (top row) and segmentation results (bottom row) for the car in the 29-th frame of *drift-chicane* sequence. From left to bottom: (top row) the shape prior obtained via average voting, weights initialization, the 1st and 2nd iteration; (bottom row) ground-truth, and three segmentation results corresponding to their shape priors right above them. We clearly see that the shape prior of the car becomes more and more precise after each iteration, which in turn leads to more accurate segmentation.

In our experiments, we set $\gamma = 0.8$ to penalize foreground seeds in case some seeds are inaccurately estimated.

The Laplacian term Ω_l encourages similar segment tracks to be similarly weighted. The similarities are captured by a graph $A \in \mathbb{R}^{k\times k}$ over the set of segment tracks. Each element A_{ij} in A denotes the affinity between the i-th and j-th tracks

$$A_{ij} = \mathcal{J}(L_i, L_j) \cdot \exp(-\frac{1}{\Gamma} D(L_i, L_j)) \tag{5}$$

where $\mathcal{J}$ denotes the Jaccard similarity that measures the average overlap ratio between the two tracks, and D represents the χ^2-distance between their color histograms in the Lab color space due to its perceptual accuracy. Γ is the mean of the χ^2-distance among all tracks. With this setup, we have

$$\Omega_l(w, L) = w^T \mathbb{L} w \tag{6}$$

where $\mathbb{L} = \text{Diag}(A1_k) - A$ is the unnormalized Laplacian matrix of the graph, with Diag(v) standing for a diagonal matrix with vector v in the diagonal. In this way, we can obtain similar predictive values w_i and w_j when the link (i, j) in the graph is strong.

The temporal term Ω_t aims to enforce the fusion in the temporal domain to be smooth. Directly optimization over pixels is computationally expensive. Hence, we break each frame I_t into n_t non-overlapping superpixels $\{s_{t,1}\}_{i=1}^{n_t}$ by multiple intersections [14] over all tracks. Let $\rho_{t,i}$ be the likelihood of superpixel $s_{t,i}$ that is computed by averaging the likelihood of foreground pixels in it. For each superpixel, we expect it to find foreground mappings in the previous and next frames by forward and backward optical flow, respectively and penalize those leaking to the background. To this end, we define Ω_t as

$$\begin{aligned}\Omega_t(w, L) = &\sum_{t=1}^{T-1}\sum_{i=1}^{n_t}(|s_{t,i}|\rho_{t,i} - \sum_{j\in\mathcal{N}_{t,i}^f} r_{i,j}^f |s_{t+1,j}|\rho_{t+1,j})^2 \\ &+ \sum_{t=2}^{T}\sum_{i=1}^{n_t}(|s_{t,i}|\rho_{t,i} - \sum_{j\in\mathcal{N}_{t,i}^b} r_{i,j}^b |s_{t-1,j}|\rho_{t-1,j})^2 \end{aligned} \tag{7}$$

where the two terms characterize the forward and backward temporal consistency, respectively. $\mathcal{N}_{t,i}^f(\mathcal{N}_{t,i}^b)$ indicates the superpixels that are mapped to $s_{t,i}$ by forward (backward) optical flow, $r_{i,j}^f(r_{i,j}^b)$ denotes the percentage of pixels in $s_{t,j}$ that is mapped to $s_{t+1,i}(s_{t-1,i})$, and $|s_{t,i}|$ is the number of pixels in $s_{t,i}$.

Problem (2) is smooth and convex with our definitions of the three terms. Therefore, we can effectively solve it with the L-BFGS algorithm. Given the estimation of w, we can easily calculate the probabilities of pixels according to Eq. 1. The foreground probabilities will be used in the next iteration for the selection of foreground and background seeds. Besides, it also serves as shape priors of objects to help predict the latent outcome $\hat{L}$ in the following.

B. LATENT OUTCOME ESTIMATION

We formulate the latent outcome $\hat{L}$ estimation as a pixel labelling problem. Without loss of generality, we represent a labeling L as $L = \{l_i\}_i$, where $l_i \in \{0, 1\}$ is the label of the i-th pixel. We then estimate these labels as a minimum of an energy function defined by

$$E(L) = \sum_{i\in\mathcal{V}}(\phi_i^a(l_i) + \lambda_3\phi_i^{sh}(l_i)) + \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(l_i, l_j) \tag{8}$$

where $\mathcal{V}$ denotes all pixels in the video, and $\mathcal{E}$ consists of all pairs of spatially $\mathcal{E}_s$ or temporally $\mathcal{E}_t$ adjacent pixels, *i.e.* $\mathcal{E} = \{\mathcal{E}_s, \mathcal{E}_t\}$. λ_3 is a weighting factor.

The unary term ϕ_i^a accounts for the cost of assigning each node the object or background labels according to the appearance cues. It is defined as

$$\phi_i^a = \begin{cases} 1 & \text{if } l_i = 1 \text{ and } i \notin \mathcal{S}_b, \\ 0 & \text{if } l_i = 1 \text{ and } i \in \mathcal{S}_b, \\ 0 & \text{if } l_i = 0 \text{ and } i \in \mathcal{S}_f, \\ p_b(i)/p_f(i) & \text{if } l_i = 0 \text{ and } i \notin \mathcal{S}_f. \end{cases} \tag{9}$$

where $p_f(i)$ and $p_b(i)$ give the probabilities of the i-th pixel being foreground and background, respectively. To this end, we learn weighted Gaussian mixture models (GMM) over RGB values of pixel colors. At each frame t, we learn a foreground model from all pixels in the video, and the i-th pixel at frame t' is weighted by its shape prior L' as well as its distance to t in time

$$\exp(-\beta \cdot (t - t')^2) \cdot L_i' \tag{10}$$

where the first exponential factor discounts the weight of pixel i over time [7]. The background GMM is learned in a similar fashion but with the second factor replaced by $1 - L_i'$.

The unary term ϕ^{sh} penalizes the assignment of pixels that are inconsistent with our shape priors to the foreground and vice versa. It has the following form

$$\phi_i^{sh}(l_i) = -\log L_i'^{l_i}(1 - L_i')^{1-l_i} \tag{11}$$

The pairwise term $\phi_{ij}(l_i, l_j)$ accounts for the cost of assigning different labels to spatially and temporally adjacent pixels

$$\phi_{ij}(l_i, l_j) = \delta(l_i \neq l_j) \exp^{-d(i,j)} \tag{12}$$

where the function d evaluates the edge, color and motion distance between neighboring pixels for some $\lambda_4 \geq 0$

$$d(i,j) = \max(e_i, e_j) + \lambda_4(\|c_i - c_j\|^2 + \|m_i - m_j\|^2) \quad (13)$$

where e_i denotes the edge intensity of pixel i given by [58]. c_i and m_i indicate pixel values in RGB image space and the corresponding optical flow field, respectively.

Since the pairwise term is sub-modular, we solve the optimization problem exactly with graph cut to obtain the estimation of $\hat{L}$.

C. WEIGHT INITIALIZATION

A good initialization of weights can largely speed up the convergence rate of our algorithm. In this work, we propose a scheme to score each segment track and the scores are then used as the initial values of the weights. Specifically, we firstly compute the average map $\bar{L} \in R^{h \times w \times n}$ as

$$\bar{L} = \frac{1}{k} \sum_{i=1}^{k} L_i \quad (14)$$

The $\bar{L}$ can be considered as a score map that accounts for the confidence of each segment belonging to an object because if a region shares more overlapping segments, the corresponding pixels in the score map will have higher values. Based on the score map, we are able to determine a score, *i.e.* the weight w, for each track

$$w_i = \max_{\tau \geq \epsilon} \mathcal{J}(\bar{L}_\tau, L_i) \quad (15)$$

where $\bar{L}_\tau = \{\bar{L} > \tau\}$ with $\tau (0 \leq \tau \leq 1)$ a threshold, and $\epsilon = 0.2$ ensures the minimal overlapping rate. $\mathcal{J}$ denotes the Jaccard similarity. Figure 2(a) illustrates the average map and the shape prior with initial weights of the 29-th frame in *drift-chicane* sequence. We can see that more accurate shape estimation of the car is obtained with the weight initialization scheme.

IV. EXPERIMENTAL RESULTS

This section presents the performance evaluation and analysis of the proposed algorithm. Two groups of experiments are conducted. First, our algorithm is compared with state-of-the-art video object segmentation methods, including both deep learning-based and non-deep learning-based methods. Second, we deeply discuss some important issues about our algorithm, *e.g.* component analysis, attribute-based analysis, running time, etc.

A. EXPERIMENTAL SETUP

1) PARAMETER SETTINGS

The algorithm is implemented in MATLAB, and all experiments are carried on a machine with a 2.93GHz Intel i7 processor. Parameters of our algorithm are fixed for all sequences with the following default values: $\lambda_1 = 10$, $\lambda_2 = 10$, $\lambda_3 = 5$, $\lambda_4 = 0.5$, $\beta = 0.0001$. The maximal number of iteration is empirically set to 3 for high efficiency.

2) DATASETS

We evaluate our algorithm on two public video object segmentation benchmarks, *e.g.* DAVIS2016 [59] and SegTrack-V2 [14].

DAVIS2016 [59] is one of the most popular datasets for evaluating video object segmentation algorithms. It contains 50 high quality video sequences with 3455 frames annotated in total. These videos span a wide range of typical challenging factors in video object segmentation such as background clutter, motion blur, occlusions and appearance change, etc. The videos in DAVIS are divided into training set (30) and validation set (20). Since our algorithm requires no training data, we use all 50 videos for evaluation in the experiments.

SegTrack-V2 [14] is introduced at 2013 to evaluate tracking and video object segmentation algorithms. It contains 14 low-resolution video sequences with 24 moving instances annotated in all frames. Among them, 8 videos contain only one primary object and the other 6 videos contain multiple objects. For accurate comparison, we only evaluate our algorithm on those 8 single object videos in the experiments.

3) EVALUATION METRICS

We evaluate the effectiveness of our method with four evaluation metrics: *region similarity* $\mathcal{J}$, *contour accuracy* $\mathcal{F}$, *temporal stability* $\mathcal{T}$ and *success plot*.

Region Similarity $\mathcal{J}$. Suppose r be the output segmentation and r' be the corresponding ground-truth. The region similarity between them is measured by the Jaccard similarity $\mathcal{J}$ that is defined as the intersection-over-union (IoU) of r and r', *i.e.* $\mathcal{J} = \frac{|r \bigcap r'|}{|r \bigcup r'|}$.

Contour Accuracy $\mathcal{F}$. Different from $\mathcal{J}$, this metric evaluates each segmentation result from a contour-based perspective, computed by F-measure $\mathcal{F} = \frac{2PR}{P+R}$, where P and R denote the precision and recall between the contour points of r and r', respectively.

Temporal Stability $\mathcal{T}$. The metric helps to recognize methods that produce jittery, unstable segmentation. It is obtained using Dynamic Time Warping (DTW) to match the contour points that minimize the distances of Shape Context Descriptor between two segments at consecutive frames.

Success plot. Success plot [60] is used as a global metric for evaluation. For a overlap threshold (x-axis on the plot), the success rate (y-axis on the plot) indicates the ratio of the frames whose segmentation prediction has more IoU with the groundtruth than the threshold. The overall success score is defined as the success rate when $th = 0.7$.

B. EVALUATION ON DAVIS DATASET

In this experiment, we directly use 14 methods evaluated in [59] for aggregation. As shown in [59], the performance of these 14 methods differ a lot. The methods with high performance, *e.g.* NLC [17], may facilitate the aggregation, while those with low performance, *e.g.* SF-LAB [61] may be unfavorable for the aggregation. It seems that one can only combine top methods to obtain good results, however,

TABLE 1. DAVIS TrainVal Set Results for overall performance of each algorithm. The metrics region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and temporal stability $\mathcal{T}$ are used for evaluation. Higher values are better for rows with ↑ and vice versa for rows with ↓. The best results for each metric are shown in bold font.

Dataset	Metric	Semi-supervised Method									
		Deep Learning Method				Non-Deep Learning Method					
		Lucid	MSK	CTN	VPN	OFL	BVS	FCP	JMP	HVS	SEA
DAVIS	Region Similarity $\mathcal{J}$ ↑	**0.866**	0.803	0.755	0.750	**0.711**	0.665	0.631	0.607	0.596	0.556
	Contour Accuracy $\mathcal{F}$ ↑	**0.848**	0.758	0.714	0.724	**0.679**	0.656	0.546	0.586	0.576	0.533
	Temporal Stability $\mathcal{T}$ ↓	**0.164**	0.189	0.198	0.300	0.224	0.317	0.294	**0.136**	0.305	0.141
Dataset	Metric	Unsupervised Method									
		Deep Learning Method		Non-Deep Learning Method							
		FSEG	LMP	ELM	ARP	TIS	MSG	FST	NLC	VOSA	**OURS**
DAVIS	Region Similarity $\mathcal{J}$ ↑	**0.716**	0.697	0.683	0.763	0.676	0.543	0.575	0.641	0.767	**0.795**
	Contour Accuracy $\mathcal{F}$ ↑	0.658	**0.663**	0.672	0.711	0.639	0.525	0.536	0.593	0.696	**0.749**
	Temporal Stability $\mathcal{T}$ ↓	**0.295**	0.688	0.258	0.359	0.320	0.263	0.293	0.366	0.248	**0.248**

we build our algorithm on all the 14 methods based on the observation that even the worst method can provide complementary cues for the best method.

The 14 single methods are given as below: SF-LAB [61], SF-MOT [61], NLC [17], CVOS [9], TRC [8], MSG [4], KEY [13], SAL [10], FST [7], TSP [19], SEA [62], HVS [20], JMP [25], FCP [23]. All the results are downloaded from the homepage of DAVIS.[1]

We compare our algorithm against top single methods as well as state-of-the-art deep learning based unsupervised methods: FSEG [46], LMP [47], non-deep learning unsupervised methods: ELM [63], ARP [64], TIS [65]. Besides, we report the results of deep learning based semi-supervised methods: Lucid [66], MSK [48], CTN [67], VPN [68] and non-deep learning semi-supervised methods: OFL [18] and BVS [69]. Furthermore, we also compare with our previous video segmentation algorithm VOSA [33]. We acquire their quantitative results from the DAVIS leaderboard.

1) QUANTITATIVE RESULTS

Table 1 summarizes the results of all top methods with respect to the metrics $\mathcal{J}$, $\mathcal{F}$ and $\mathcal{T}$. For each metric, we consider the *Mean* which is the error of each method over the whole dataset. This shows that our algorithm achieves the top result in terms of region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and temporal stability $\mathcal{T}$ in unsupervised methods.

The table demonstrates the effectiveness of our algorithm from several aspects:

1) Compared with non-deep learning segmentation algorithms (both semi-supervised and unsupervised), our algorithm achieves the highest average region similarity (**0.795**) over the 50 test video sequences. Our results also achieve a significant improvement over the second best algorithm ARP [64] (0.763) and the third best algorithm OFL [18] (0.711). Furthermore, our algorithm achieves the best contour accuracy (**0.749**) over the non-deep learning based algorithms and gains a competitive temporal stability score (0.248). This demonstrates that our algorithm produces segments well aligned with object boundaries and they are temporally consistent.

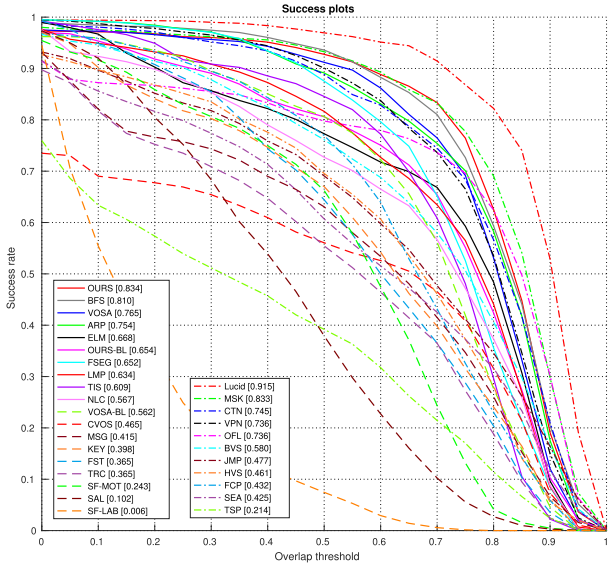

FIGURE 3. DAVIS TrainVal Results representing the success plots of all methods on the DAVIS TrainVal set. The methods are ranked at overlap threshold *th* = 0.7. (Best viewed in color).

2) Compared with deep learning based algorithms, the semi-supervised algorithms Lucid [66] and MSK [48] generate more accurate segmentation results than our algorithm because they use large scale training data and learn specific appearance models of target objects based on the annotations in the first frame. However, when we compare the results of unsupervised methods, our algorithm can outperform FSEG [46] and LMP [47] by a large margin. This result is inspiring, especially considering that we only use the results from non-deep learning methods and some of them show poor performance.

3) Compared with our previous aggregation method VOSA [33], we see that our method obtain +**2.8** improvement in terms of region similarity, and +**5.3** improvement in terms of contour accuracy. We attribute such large improvements to the adaptive weight learning mechanism which can find

[1] https://davischallenge.org/index.html

TABLE 2. SegTrack-V2 Results with the region similarity $\mathcal{J}$. The best results for each metric are shown in bold font. Results of the singe methods are computed using the author provided codes.Videos are single-object only.

Dataset	Sequence	Method										
		SF-LAB	FST	KEY	NLC	SAL	SF-MOT	TSP	MST	TIPS	TIPF	OURS
Segtrack-V2	Birdfall	0.00	0.01	0.14	0.43	0.12	0.69	0.10	0.04	0.03	0.47	0.56
	Frog	0.20	0.48	0.35	0.46	0.45	0.55	0.35	0.30	0.44	0.17	0.44
	Bird_of_Paradise	0.74	0.73	0.11	0.56	0.39	0.44	0.19	0.19	0.32	0.14	0.80
	Girl	0.12	0.42	0.54	0.69	0.63	0.61	0.17	0.59	0.63	0.27	0.64
	Monkey	0.41	0.59	0.59	0.69	0.76	0.61	0.11	0.22	0.56	0.40	0.60
	Parachute	0.00	0.76	0.90	0.88	0.74	0.87	0.53	0.62	0.57	0.70	0.88
	Soldier	0.13	0.52	0.09	0.76	0.33	0.12	0.35	0.47	0.44	0.31	0.76
	Worm	0.23	0.64	0.07	0.37	0.59	0.20	0.04	0.26	0.70	0.29	0.70
	Mean	0.23	0.52	0.35	0.61	0.50	0.51	0.23	0.34	0.46	0.34	**0.67**

the optimal combination of different methods for each test sequence. This enables our algorithm to fully capture the complementary roles of different methods.

Figure 3 shows a comparison with these methods over success plot metric. In the plot, the left brackets list the unsupervised methods and our baseline methods (*i.e.* BFS, OURS-BL, VOSA-BL which are used for ablation studies), while the right brackets contains semi-supervised methods. As shown in the plot, our algorithm outperforms all unsupervised methods and most semi-supervised methods, indicating the effectiveness of our algorithm.

Furthermore, we observe that our algorithm even outperforms the reference method BFS, which is designed by always selecting the best algorithm for each sequence. Such good results are remarkable, given the fact that our algorithm is fully unsupervised as well as the fusion result is easily affected by the methods with poor performance, *e.g.* SF-LAB and SAL. Performing better than BFS emphasizes the strength of the proposed algorithm and reinforces that properly aggregation of single methods does enhance the segmentation accuracy. We owe this performance to the collaboration between *segmentation label inference* and *weights learning*, each of which benefits from the other to improve itself.

2) QUALITATIVE RESULTS

Qualitative video segmentation results for 8 representative sequences from the DAVIS dataset [59] are presented in Figure 4. We see that our algorithm can handle typical challenges in the task of video object segmentation, *e.g.* shape deformation (*horsejump-high*, *dog-agility*), scale changes (*soapbox*, *drift-chicane*), fast motion patterns (*breakdance-flare*), noticeable appearance changes (*motocross-bumps*), object with distinct colors (*roller-blade*) and occlusions (*libby*).

C. EVALUATION ON SegTrack-V2

In this experiment, we evaluate our method on the SegTrack-V2 benchmark. Different from DAVIS, this benchmark does not provide pre-computed segmentation results. Therefore, we run the codes of single methods to obtain the segments. Here, we only use the single methods whose codes are public available, *i.e.* SAL [10], SF-MOT [61], SF-LAB [61], TSP [19], KEY [13], FST [7], NLC [17], MSG [4], plus two new methods TIPF [70] and TIPS [36].

1) QUANTITATIVE RESULTS

Table 2 reports our performance on SegTrack-V2. The region similarity $\mathcal{J}$ is used here for evaluation. As can be seen, our algorithm outperforms other methods overall, achieving the best region similarity score (**0.67**), 6% higher than the second best NLC (0.61). It should be noted that our algorithm is worse than the best single method in the *Birdfall* sequence. The main reason lies in that most of the single methods, *e.g.* SF-LAB [61], FST [7], KEY [13], TSP [19] and TIPS [36] fail in this sequence. This brings too much noise into the aggregated segment set, which are hard for our algorithm to handle. This also indicates that there's still large room for improvement of current video object segmentation methods.

2) QUALITATIVE RESULTS

Representative pixel labeling results of four video sequences on SegTrack-V2 are shown in Figure 5. It can be observed that our method successfully segment out the foreground object in these low-resolution frames. Our method has the ability to process objects with shape deformation (*bird_of_paradise*, *worm*), appearance changes (*monkey*), scale variations (*solider*). The qualitative and quantitative results further demonstrate the effectiveness of our method.

D. IN-DEPTH VALIDATION EXPERIMENTS

In this section, we offer more detailed analysis for the proposed algorithm in several aspects with the DAVIS dataset. We perform attribute-based study, exploit the weight learning mechanism, evaluate the contributions of each single method and conduct runtime analysis.

1) *Attribute-Based Analysis:* The videos in DAVIS are characterized with 15 challenge attributes, background clutter (BC), deformation (DEF), motion blur (MB), fast motion (FM), low resolution (LR), occlusion (OCC), out-of-view (OV), scale-variation (SV), appearance changes (AC), edge ambiguity (EA), camera-shake (CS), heterogeneous object (HO), interacting objects (IO), dynamic background (DB) and shape complexity (SC). With these

FIGURE 4. Qualitative segmentation results on eight video sequences from **DAVIS** [59]. From top to bottom: ***horsejump-high***, ***soapbox***, ***drift-chicane***, ***dog-agility***, ***breakdance-flare***, ***motocross-bumps***, ***roller-blade***, ***libby***. It can be observed that our algorithm can deal with a large set of scenarios and is robust to scale variations, motion blur, occlusions and background clutter. (Best viewed in color).

attribute annotations, we conduct a more detailed evaluation to verify the proposed algorithm in dealing with various challenges. Table 3 presents the results of our algorithm together with the top 11 single methods according to the rankings in [59]. Besides, we also show the results of our previous method, *i.e.* VOSA.

The attribute-based results (Table 3) show that our algorithm performs outstandingly well in all situations. This can be explained as that for each challenge attribute, *e.g.* AC, our algorithm can automatically find the single methods that are robust to it and associate them with relatively larger weights. As a consequence, our algorithm can adapt to very complex scenarios.

2) Does the weight learning help? In the previous experiments, we show the remarkable performance of our approach compared with the state of the art. However, we see that VOSA also shows fairly good results even by simply combining single methods with equal weights. Hence, we next to verify whether the weight learning mechanism helps or not.

To verify this question, we build two baseline methods, OURS-BL and VOSA-BL, by using the 7 worst methods in Figure 3 for aggregation. What can be expected is that both of the baselines will experience a large decrease of performance in this case because only bad results are used. However, with this setting, we can measure the two methods in extreme situations. Figure 3 shows a comparison with these

FIGURE 5. Qualitative segmentation results on four video sequences from **SegTrack-V2** [14]. From top row to the bottom: ***monkey***, ***bird_of_paradise***, ***solider***, ***worm***. (Best viewed in color).

TABLE 3. **DAVIS** Results for Attribute-based performance. For each method, we report its average region similarity $\mathcal{J}$ over the sequences with a specific attribute (left), *e.g.* AC, as well as the performance gain (or loss) for the method for the remaining sequences without that attribute (right). The best results for each metric are shown in bold font.

Attribute	Methods											
	NLC	CVOS	TRC	MSG	KEY	FST	SEA	HVS	JMP	FCP	VOSA	OURS
LR	0.67 *-0.04*	0.42 *+0.13*	0.47 *+0.04*	0.51 *+0.05*	0.54 *+0.04*	0.53 *+0.05*	0.47 *+0.11*	0.49 *+0.15*	0.51 *+0.13*	0.59 *+0.06*	0.74 *+0.04*	**0.76 *+0.05***
SV	0.54 *+0.15*	0.44 *+0.11*	0.44 *+0.10*	0.52 *+0.03*	0.50 *+0.10*	0.50 *+0.10*	0.49 *+0.09*	0.46 *+0.20*	0.58 *+0.04*	0.52 *+0.16*	0.73 *+0.05*	**0.75 *+0.06***
SC	0.61 *+0.06*	0.51 *+0.01*	0.47 *+0.05*	0.54 *+0.01*	0.51 *+0.11*	0.53 *+0.08*	0.51 *+0.09*	0.57 *+0.04*	0.53 *+0.14*	0.59 *+0.07*	0.71 *+0.10*	**0.73 *+0.10***
FM	0.64 *+0.00*	0.37 *+0.24*	0.41 *+0.16*	0.46 *+0.14*	0.50 *+0.12*	0.50 *+0.12*	0.40 *+0.28*	0.42 *+0.31*	0.50 *+0.18*	0.55 *+0.13*	0.72 *+0.07*	**0.76 *+0.06***
CS	0.61 *+0.05*	0.43 *+0.11*	0.56 *-0.08*	0.55 *-0.01*	0.52 *+0.06*	0.54 *+0.05*	0.42 *+0.18*	0.56 *+0.05*	0.61 *-0.00*	0.61 *+0.03*	0.75 *+0.03*	**0.76 *+0.04***
IO	0.63 *+0.03*	0.55 *-0.07*	0.48 *+0.05*	0.58 *-0.08*	0.54 *+0.06*	0.49 *+0.17*	0.54 *+0.03*	0.57 *+0.06*	0.59 *+0.04*	0.60 *+0.07*	0.75 *+0.03*	**0.77 *+0.05***
DB	0.53 *+0.15*	0.37 *+0.18*	0.39 *+0.15*	0.43 *+0.15*	0.52 *+0.07*	0.53 *+0.06*	0.58 *-0.03*	0.60 *-0.01*	0.60 *+0.01*	0.62 *+0.01*	0.74 *+0.03*	**0.77 *+0.03***
MB	0.61 *+0.04*	0.36 *+0.23*	0.32 *+0.27*	0.35 *+0.29*	0.51 *+0.08*	0.48 *+0.14*	0.39 *+0.24*	0.44 *+0.24*	0.51 *+0.15*	0.53 *+0.15*	0.69 *+0.12*	**0.72 *+0.12***
DEF	0.68 *-0.10*	0.52 *-0.00*	0.49 *+0.02*	0.52 *+0.06*	0.57 *-0.01*	0.57 *+0.01*	0.50 *+0.14*	0.59 *+0.01*	0.59 *+0.03*	0.61 *+0.05*	0.76 *+0.02*	**0.79 *+0.01***
OCC	0.70 *-0.09*	0.43 *+0.13*	0.44 *+0.10*	0.48 *+0.10*	0.52 *+0.08*	0.53 *+0.07*	0.47 *+0.13*	0.53 *+0.11*	0.47 *+0.21*	0.59 *+0.07*	0.72 *+0.07*	**0.75 *+0.07***
HO	0.65 *-0.04*	0.50 *+0.04*	0.47 *+0.14*	0.54 *+0.01*	0.54 *+0.12*	0.55 *+0.10*	0.49 *+0.24*	0.54 *+0.21*	0.56 *+0.17*	0.60 *+0.13*	0.75 *+0.06*	**0.78 *+0.05***
EA	0.51 *+0.24*	0.42 *+0.18*	0.47 *+0.06*	0.46 *+0.15*	0.49 *+0.15*	0.52 *+0.10*	0.52 *+0.07*	0.55 *+0.09*	0.56 *+0.09*	0.58 *+0.10*	0.70 *+0.12*	**0.73 *+0.13***
OV	0.52 *+0.15*	0.37 *+0.19*	0.34 *+0.20*	0.42 *+0.16*	0.43 *+0.18*	0.50 *+0.10*	0.44 *+0.14*	0.42 *+0.23*	0.61 *-0.01*	0.53 *+0.13*	0.71 *+0.08*	**0.73 *+0.08***
AC	0.54 *+0.13*	0.42 *+0.12*	0.37 *+0.17*	0.48 *+0.08*	0.42 *+0.19*	0.55 *+0.04*	0.46 *+0.12*	0.42 *+0.23*	0.58 *+0.03*	0.51 *+0.16*	0.74 *+0.04*	**0.77 *+0.03***
BC	0.46 *+0.23*	0.46 *+0.06*	0.51 *-0.01*	0.55 *-0.00*	0.53 *+0.05*	0.57 *+0.00*	0.59 *-0.04*	0.62 *-0.03*	0.61 *-0.00*	0.59 *+0.05*	0.74 *+0.04*	**0.76 *+0.04***

methods using success plots. Their success plots show that the performance of VOSA-BL degrades more quickly than OURS-BL, suggesting that with equal weights, VOSA cannot work well in this extreme case. Besides, it is inspiring that OURS-BL even outperforms the best single method, *i.e.* NLC in terms of region similarity $\mathcal{J}$. The comparison shows that our algorithm benefits a lot from the weight learning module, and also demonstrates the advantages of our algorithm over VOSA.

3) What can we learn from the weights we obtained? Figure 6 shows a heatmap representing the optimal weights of single methods we learned on 5 attribute-based video subsets. We see that for some methods, their weights are always high (*e.g.* NLC) or low (*e.g.* SF-LAB) in each subset due to their global good or poor performance. For other methods, the weights vary a lot among different subsets. Take JMP as an example, its weight in AC subset is much larger than that in OCC subset. Also in Table 3, we have demonstrated that

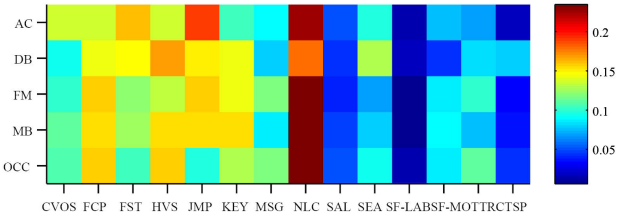

FIGURE 6. A heatmap showing average weights of single methods over sequences with the 5 attributes.

JMP is more robust to appearance changes than occlusions. Thus, we can learn that the weights can faithfully account for the performance of single methods, and thus provide us the possibilities to analyze the strengths and weaknesses of the methods in unsupervised ways.

4) How the aggregation of different methods affect the segmentation accuracy? To investigate which method affects

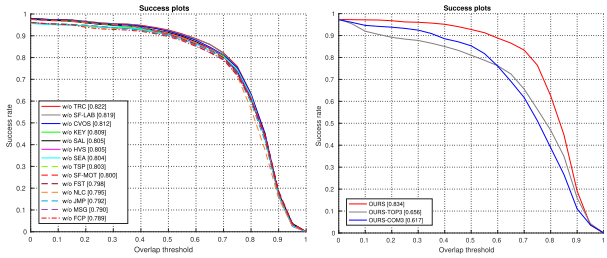

FIGURE 7. (a) The aggregation results when one method is removed at each time. Here *w/o* means *without*. (b) Comparison of the aggregation results using all single methods (OURS), the top 3 single methods (OURS-TOP3), and the selected 3 methods with high complementaries (OURS-COM3).

TABLE 4. Running time in seconds for each component on the DAVIS dataset with 480p resolution.

Component	Time (Seconds/Frame)
Optical Flow	1.2
Weight Initialization	0.4
Weight Learning (1 iteration)	0.5
Segmentation Inference (1 iteration)	0.2
Total (3 iterations)	3.7

the most to our aggregation strategy, we remove one single method at each time and aggregate the remaining methods. As illustrated in Figure 7(a), every single method contributes to our aggregation to a certain extent. Surprisingly, the performance degrades by 4.5% when FCP is removed in comparison with 3.9% by removing the best single method, *i.e.* NLC. This means that the more complementary methods are more effective to boost our performance rather than the best single method.

Furthermore, we aim to investigate the performance by only aggregating the best three single methods (*i.e.* NLC, JMP, CVOS) or the most complementary three methods (*i.e.* KEY, MSG, JMP). In Figure 7(b), the curves OURS-TOP3 and OURS-COM3 indicate the success plots of the two settings, respectively. We can see that when the overal threshold $th \in [0.3, 0.6]$, OURS-COM3, benefiting from the complementary property, shows better performance than OURS-TOP3. When the threshold $th > 0.6$, OURS-TOP3 outperforms OURS-COM3 because the former actually combines more accurate segments. Furthermore, it is encouraging that despite combining only three methods, both OURS-TOP3 and OURS-COM3 clearly outperforms the best single method, *i.e.* NLC.

5) Runtime Analysis We conduct runtime analysis for our algorithm from two aspects: running time of all single methods and our aggregation algorithm. On one hand, since our algorithm requires the results from single methods, it is actually slower than each single one. However, in practice, we could largely enhance the speed of the procedure by data-sharing (*e.g.* optical flow) and parallelization. Besides, we could use less methods (*e.g.* 3) for aggregation, in which case our algorithm has been demonstrated in Figure 7(b) to achieve relatively accurate segmentation. On the other hand, in Table 4 we report the time consumption of each component used in our algorithm on the DAVIS dataset. To be fair, we estimate optical flow via LDOF [71] as other methods.

V. CONCLUSION

In this paper, we present an unsupervised aggregation algorithm to segment primary objects in unconstrained videos. The algorithm is based on the idea that different segmentation methods often complement each other well. Our algorithm explores the complementary roles among single methods via a regularized regression model with latent outcome, which is solved in an iterative manner. That is, the *weight learning* and *segmentation inference* modules collaborate and improve the quality of the solution in each iteration. We demonstrate that the proposed algorithm significantly improves performance on the DAVIS and SegTrack-V2 dataset. Ablation studies with OURS-BL prove that our good performance lies in our learning scheme rather than the single methods.

REFERENCES

[1] Y. Jae Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1346–1353.
[2] J. Son, I. Jung, K. Park, and B. Han, "Tracking-by-segmentation with online gradient boosting decision tree," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3056–3064.
[3] V. Mezaris, I. Kompatsiaris, N. Boulgouris, and M. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 606–621, May 2004.
[4] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 282–295.
[5] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato, "Superpixel-based video object segmentation using perceptual organization and location prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4814–4822.
[6] W.-D. Jang, C. Lee, and C.-S. Kim, "Primary object segmentation in videos via alternate convex optimization of foreground and background distributions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 696–704.
[7] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1777–1784.
[8] K. Fragkiadaki, G. Zhang, and J. Shi, "Video segmentation by tracing discontinuities in a trajectory embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1846–1853.
[9] B. Taylor, V. Karasev, and S. Soattoc, "Causal video object segmentation from persistence of occlusions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4268–4276.
[10] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3395–3402.
[11] Z. Qiu, T. Yao, and T. Mei, "Learning deep spatio-temporal dependence for semantic video segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 939–949, Apr. 2018.
[12] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 328–335.
[13] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1995–2002.
[14] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2192–2199.
[15] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 670–677.
[16] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 628–635.

[17] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proc. BMVC*, vol. 2, 2014, p. 6.
[18] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3899–3908.
[19] J. Chang, D. Wei, and J. W. F. Iii, "A video representation using temporal superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2051–2058.
[20] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2141–2148.
[21] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang, "JOTS: Joint online tracking and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2226–2234.
[22] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 656–671.
[23] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung, "Fully connected object proposals for video segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3227–3234.
[24] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video SnapCut: Robust video object cutout using localized classifiers," *ACM Trans. Graph.*, vol. 28, p. 70, Jul. 2009.
[25] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen, "JumpCut: Non-successive mask transfer and interpolation for video cutout," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–10, Oct. 2015.
[26] B. L. Price, B. S. Morse, and S. Cohen, "LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 779–786.
[27] F. Zhong, X. Qin, Q. Peng, and X. Meng, "Discontinuity-aware video object cutout," *ACM Trans. Graph.*, vol. 31, no. 6, p. 175, Nov. 2012.
[28] B. Luo, H. Li, F. Meng, Q. Wu, and C. Huang, "Video object segmentation via global consistency aware query strategy," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1482–1493, Jul. 2017.
[29] O. Sener, K. Ugur, and A. A. Alatan, "Efficient MRF energy propagation for video segmentation via bilateral filters," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1292–1302, Aug. 2014.
[30] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1131–1138.
[31] J. Wang, A. Borji, C.-C. J. Kuo, and L. Itti, "Learning a combined model of visual saliency for fixation prediction," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1566–1579, Apr. 2016.
[32] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 170–185.
[33] T. Zhou, Y. Lu, H. Di, and J. Zhang, "Video object segmentation aggregation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2016, pp. 1–6.
[34] D. Tsai, M. Flagg, and J. M. Rehg, "Motion coherent tracking using multi-label MRF optimization," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 190–202, Nov. 2012.
[35] L. Chen, J. Shen, W. Wang, and B. Ni, "Video object segmentation via dense trajectories," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2225–2234, Dec. 2015.
[36] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.
[37] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato, "Saliency-based video segmentation with graph cuts and sequentially updated priors," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2009, pp. 638–641.
[38] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
[39] K. E. A. Van De Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1879–1886.
[40] A. Humayun, F. Li, and J. M. Rehg, "RIGOR: Reusing inference in graph cuts for generating object regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 336–343.
[41] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia, "Semantic object segmentation via detection in weakly labeled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3641–3649.
[42] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 715–731.
[43] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. V. Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.
[44] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao, "Motion-attentive transition for zero-shot video object segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 476–483.
[45] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 686–695.
[46] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2126.
[47] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4491–4500.
[48] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2017, pp. 2663–2672.
[49] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," 2018, *arXiv:1807.09190*. [Online]. Available: https://arxiv.org/abs/1807.09190
[50] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5997–6005.
[51] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7376–7385.
[52] Y. Chen, J. Pont-Tuset, A. Montes, and L. V. Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1189–1198.
[53] O. Le Meur and Z. Liu, "Saliency aggregation: Does unity make strength?" in *Proc. Asian Conf. Comput. Vis. (ECCV)*, 2014, pp. 18–32.
[54] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.
[55] N. Wang and D.-Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Proc. ICML*, 2014, pp. 1107–1115.
[56] T. Zhou, Y. Lu, and H. Di, "Locality-constrained collaborative model for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 313–325, Feb. 2017, doi: 10.1109/tcsvt.2015.2493498.
[57] X. Zhang, W. Li, M. Fan, D. Wang, and X. Ye, "Multi-modality tracker aggregation: From generative to discriminative," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1937–1943.
[58] P. Dollar and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1841–1848.
[59] F. Perazzi, J. Pont-Tuset, B. Mcwilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.
[60] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
[61] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 733–740.
[62] S. A. Ramakanth and R. V. Babu, "SeamSeg: Video object segmentation using patch seams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 376–383.
[63] D. Lao and G. Sundaramoorthi, "Extending layered models to 3D motion," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 435–451.
[64] Y. J. Koh and C.-S. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, pp. 7417–7425.
[65] B. Griffin and J. Corso, "Tukey-inspired video object segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1723–1733.
[66] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for video object segmentation," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1175–1197, Sep. 2019.

[67] W.-D. Jang and C.-S. Kim, "Online video object segmentation via convolutional trident network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5849–5858.
[68] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 451–461.
[69] N. Marki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, "Bilateral space video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 743–751.
[70] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai, and C.-W. Lin, "A video saliency detection model in compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 27–38, Jan. 2014.
[71] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

LIN ZHANG received the B.S. and M.S. degrees in control science and engineering from the North China University of Technology, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree in computer science with the Beijing Institute of Technology, China. Her main research interests include video saliency and video segmentation.

YAO LU received the B.S. degree in electronics from Northeast University, Shenyang, China, in 1982, and the Ph.D. degree in computer science from Gunma University, Gunma, Japan, in 2003.

He was a Lecturer and an Associate Professor with Hebei University, China, from 1986 to 1998, and a Foreign Researcher with Gunma University, in 1999. In 2003, he was an invited Professor with the Engineering Faculty, Gunma University, and a Visiting Fellow with The University of Sydney, Australia. He is currently a Professor with the Department of Computer Science, Beijing Institute of Technology, Beijing, China. He has published more than 100 articles in international conferences and journals. His research interests include neural networks, image processing and video analysis, and pattern recognition.