Learning Dynamic Compact Memory Embedding for Deformable Visual Object Tracking

Hongtao Yu, Pengfei Zhu, Kaihua Zhang, Yu Wang, Shuai Zhao, Lei Wang, Tianzhu Zhang, Member, IEEE, and Qinghua Hu, Senior Member, IEEE

Abstract—Recently, template-based trackers have become the leading tracking algorithms with promising performance in terms of efficiency and accuracy. However, the correlation operation between query feature and the given template only achieves accurate target localization, but is prone to state estimation error, especially when the target suffers from severe deformation. To address this issue, segmentation-based trackers are proposed that use per-pixel matching to improve the tracking performance of deformable objects effectively. However, most of the existing trackers only match with the target features of the initial frame, thereby lacking the discrimination for handling a variety of challenging factors, e.g., similar distractors, background clutter, and appearance change. To this end, we propose a dynamic compact memory embedding technique to enhance the discrimination of the segmentation-based visual tracking method that can well tell the target from the background. Specifically, we initialize a memory embedding with the target features in the first frame. During the tracking process, the current target features that have certain correlation with the existing memory are updated to the memory embedding online. To further improve the tracking accuracy for deformable objects, we use a weighted point-to-global matching strategy to measure the correlation between the pixelwise query feature and the whole template, so as to capture more detailed deformation information. Extensive evaluations on six challenging tracking benchmarks including VOT2016, VOT2018, VOT2019, GOT-10K, TrackingNet, and LaSOT demonstrate the superiority of our method over recent remarkable trackers. Besides, our tracker outperforms the excellent segmentation-based trackers, i.e., D3S and SiamMask on the DAVIS2017 benchmark. The code is available at https://github.com/peace-love243/CMEDFL.

Index Terms—Compact memory, deformable feature, video object segmentation, visual object tracking.

I. INTRODUCTION

VISUAL object tracking (VOT) is a fundamental and challenging task in the computer vision community. It has numerous practical applications, such as traffic surveillance, human–computer interaction, autonomous robots, and autonomous driving [1], [2], [3], [4]. Although VOT achieves great improvement in terms of both accuracy and robustness, yet, there are some remaining challenges needed to be solved, e.g., similar distractors, background clutter, and deformation.

In recent years, the Siamese-network-based VOT methods have attracted widespread attention due to their high tracking speed and accuracy. SiamFC [5] and SINT [6] are the pioneers of the Siamese-network-based trackers. SiamFC applies the strategy of multiscale search to estimate the target scale state. Based on the original works, the developed anchor-based [7], [8], [9], [10], [11] and anchor-free [12], [13], [14], [15] methods adopt target scale regression strategy, which effectively improves the tracking performance for target scale variation. However, it is hard for a fixed template to adapt to changable scenarios and target appearance, and it may lead to mismatches between the template and the search region. Especially for the situations with obvious target appearance change and similar object distractors, the Siamese-network-based trackers usually fail. On the other line, the discriminative correlation filter (DCF)-based trackers [16], [17], [18], [19], [20], [21], [22], [23] use ridge regression to learn the filter, which is updated online. To a certain extent, the DCF-based trackers can alleviate the dilemma that fixed templates are hard to adapt to scene changes. Despite demonstrated success, both the Siamese-network-based and DCF-based tracking algorithms have a common drawback, that is, the limited template information and correlation-based matching make them unable to fully express the deformation of the target. To address this issue, a series of promising segmentation-based trackers are proposed [24], [25]. These trackers combine with the video object segmentation (VOS) model to exploit the insensitivity of the segmentation methods to nonrigid deformation.

It is worth noting that the encoding feature input to the segmentation network is generated from target similarity matching between the current query feature and the template from the initial frame. However, the template features cannot

Manuscript received 22 November 2021; revised 24 June 2022; accepted 18 September 2022. Date of publication 13 October 2022; date of current version 5 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2101904 and in part by the National Natural Science Foundation of China under Grant 61732011, Grant 61876127, Grant 61925602, Grant 62106174, Grant 62222608, and Grant 62276141. (Corresponding author: Pengfei Zhu.)

Hongtao Yu, Pengfei Zhu, Yu Wang, and Qinghua Hu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: yuhongtao@tju.edu.cn; zhupengfei@tju.edu.cn; wangyu_@tju.edu.cn; huqinghua@tju.edu.cn).

Kaihua Zhang is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: zhikhu@njust.edu.cn).

Tianzhu Zhang is with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China, and also with the Automotive Data of China (Tianjin) Company Ltd., Tianjin 300393, China (e-mail: zhaoshuai@catarc.ac.cn).

SiamFC applies the strategy of multiscale search to estimate the target scale state. Based on the original works, the developed anchor-based [7], [8], [9], [10], [11] and anchor-free [12], [13], [14], [15] methods adopt target scale regression strategy, which effectively improves the tracking performance for target scale variation. However, it is hard for a fixed template to adapt to changable scenarios and target appearance, and it may lead to mismatches between the template and the search region. Especially for the situations with obvious target appearance change and similar object distractors, the Siamese-network-based trackers usually fail. On the other line, the discriminative correlation filter (DCF)-based trackers [16], [17], [18], [19], [20], [21], [22], [23] use ridge regression to learn the filter, which is updated online. To a certain extent, the DCF-based trackers can alleviate the dilemma that fixed templates are hard to adapt to scene changes. Despite demonstrated success, both the Siamese-network-based and DCF-based tracking algorithms have a common drawback, that is, the limited template information and correlation-based matching make them unable to fully express the deformation of the target. To address this issue, a series of promising segmentation-based trackers are proposed [24], [25]. These trackers combine with the video object segmentation (VOS) model to exploit the insensitivity of the segmentation methods to nonrigid deformation.

It is worth noting that the encoding feature input to the segmentation network is generated from target similarity matching between the current query feature and the template from the initial frame. However, the template features cannot
well cover all the representations of the target to be tracked within the whole video sequence. In addition, the result of target similarity matching usually contains a certain amount of mismatching noises caused by similar distractors, background clutter, occlusion, etc. Therefore, the matching result cannot express accurate spatial position and scale state of the target. In summary, the template matching in VOT has the following two limitations. First, a single template cannot provide enough target information for target similarity matching. The initial template only contains limited and fixed target structure information, which cannot adapt to target appearance variations over time in the video sequence. Second, the existing matching methods [5], [7], [24], [25] conduct between the corresponding pixels merely, which is too rough to capture fine target deformation sufficiently.

To address the above issues, in this article, we propose a dynamic compact memory embedding (CME) mechanism and a deformable feature learning (DFL) module for visual tracking. Our tracker is dubbed CMEDFL. First, inspired by the Hash algorithm [26], we develop a dynamic memory embedding mechanism for target similarity matching. By retrieving the feature affinity matrix which is generated during target similarity matching, we obtain the correlation between the query feature and the existing memory. Then, we merge the high-correlation parts between the existing memory and the current target feature. In this way, we form a memory without repetition. Besides, we concatenate the parts with medium correlation to the memory and directly discard the irrelevant ones. Therefore, the diversity and compactness of the memory embedding can be ensured. High-quality memory embedding provides complete target information in historical frames, thus effectively dealing with target occlusion, similar distractors, and other variations. To further perceive the detailed target deformation, we propose a DFL module. Compared with the existing methods based on correlation operation [5], [7], [9] or target similarity matching [25], [27], the proposed DFL method adopts a pixel-to-global association strategy. By aggregating the weighted correlation between each query pixel and the entire reference features, the deformation information of the target can be effectively captured.

The effectiveness of CMEDFL is verified on six visual tracking benchmarks: VOT2016 [28], VOT2018 [29], VOT2019 [30], GOT-10K [31], TrackingNet [32], and LaSOT [33]. Our CMEDFL obtains a new remarkable EAO score of 0.525 on VOT2018 [29]. Compared with the recent state-of-the-art trackers such as Ocean [15], D3S [25], and DiMP [23], our CMEDFL achieves competitive performance. Fig. 1 illustrates some sampled qualitative results of our method on four challenging video sequences. Our CMEDFL successfully overcomes the challenges of occlusion, deformation, background clutter, similar distractors, etc.

The main contributions of our work can be summarized as threefold.

1) We propose a CME mechanism for target similarity matching. The dynamic compact memory adjustment mechanism only stores high-quality historical information of the target, thus providing effective reference in complex situations such as similar distractors and background clutter.

2) We propose a DFL strategy to further improve the accuracy of tracking deformable target. The deformation of the target can be effectively obtained by establishing the global correlation between the per-pixel query feature and the entire reference template features.

3) Abundant comparison experiments conducted on six challenging tracking datasets show that our CMEDFL achieves remarkable performance compared with several state-of-the-art trackers. Moreover, CMEDFL outperforms the state-of-the-art segmentation-based trackers (i.e., D3S [25] and SiamMask [24]) and the general VOS methods (e.g., VM [27] and FAVOS [34]) on the DAIS2017 [35] benchmark.

II. RELATED WORK

A. Visual Object Tracking

1) Siamese-Network-Based Tracking: During tracking, the position of the maximum correlation between the search region and the fixed template is considered as the target localization. For better generalization, the Siamese-network-based trackers are usually trained by massive labeled data. SINT [6] and SiamFC [5] have a milestone impact on the development of visual tracking. They are the first attempt to train the Siamese networks for visual tracking. SiamRPN++ [9] and SiamDW [10] adjust the structure of ResNet [36] and successfully apply it to the Siamese-network-based tracking model, which significantly improves the tracking performance. SiamRPN [7] applies the region proposal network (RPN) to the Siamese network for tracking. The two-branch network has the classification head for foreground–background separation of anchors and the regression head for proposal refinement. Compared with the anchor-based methods [7], [8], [9], [10], the anchor-free tracking methods [12], [13], [14], [15] avoid abundant presets of anchor, thereby significantly reducing the model hyperparameters. These methods can achieve more

Fig. 1. Some sampled results generated by our tracker on four challenging video sequences, i.e., gymnastics1, fish4, basketball, and singer1. The red bounding boxes represent the ground-truth annotations given by the initial frames. Our tracker overcomes the existing challenges (e.g., object deformation, background clutter, occlusion, similar distractors, appearance change, to name a few) in the above sequences and achieves accurate tracking and segmentation results.
flexible regression of target bounding box and new state-of-the-art performance. Although it is simple and efficient, the fixed template is hard to faithfully express the target appearance and scale variations.

2) DCF-Based Tracking: MOSSE [16] is the first attempt to learn the filter coefficients by ridge regression in the Fourier domain. As online tracking methods, the DCF-based trackers present better adaptability and generalization to appearance and scale variations. Afterward, a variety of improved strategies further boost the performance of the DCF-based trackers, such as continuous convolution [18], dynamic updating of the training set [17], spatial regularization [21], and temporal smoothing regularization [19]. CFNet [37] attempts to insert the DCF on the template branch of the Siamese network. By treating DCF as a layer in the network, the model can be backpropagated so that the entire network can be trained end-to-end. The Siamese network enhances the representation of DCF. Meanwhile, DCF achieves the online updates of the template. ATOM [20] and DiMP [23] achieve new state-of-the-art tracking performance by combining the online update of DCF and the target localization refinement of the modified IOU-Net [38].

3) Segmentation-Based Tracking: CSR-DCF [39] constructs the target mask via the color histogram of the foreground and the background. Then, the boundary effect is suppressed well by adding the mask to the filters. SiamMask [24] extends the light segmentation network to the Siamese-network-based tracking model, significantly enhancing the target representation with the aid of segmentation loss. Compared with the general VOS methods [27], [34], [40], [41], SiamMask [24] achieves higher tracking speed owing to using the lightweight segmentation network. D3S [25] replaces the target regression branch in the tracking model with a segmentation network. By combining the accurate localization of DCF and the robustness of the segmentation model to target deformation, D3S achieves new state-of-the-art tracking performance. OceanPlus [42] uses the attention retrieval mechanism to obtain the coarse target segmentation mask. The coarse mask filters the background noise in the target feature maps, improving the segmentation accuracy. DMB [43] stores the historical appearance and spatial positioning information of the target, providing a rich reference for current target segmentation. Different from these methods, we develop CME and DFL to improve the discrimination and ability of details’ capturing for our CMEDFL tracker.

B. Memory Embedding for VOT and VOS

Some works [40], [41], [44] resort to learning effective memory embedding that can provide adequate reference for the VOT and VOS tasks. MemTrack [44] uses a dynamic memory bank to overcome tracking drift caused by the fixed template. By dynamically storing and reading the tracking results in historical frames and fusing with the initial target template, the target template can be updated more accurately. STM [40] stores dense features and masks of historical frames for current pixel-level spatio-temporal information matching. Dense reference information allows STM to handle appearance changes and occlusions well. To avoid memory redundancy and slow querying in excessive memory storage, AFB-URR [41] proposes an adaptive feature bank to organize the historical information of the target dynamically. It uses the weighted average method to merge similar memories and learns from the cache replacement strategy to eliminate the memory with the least query frequency. In this work, we propose a CME, which stores only the target features relevant to the existing memory during the tracking process to alleviate memory redundancy and false retrieval.

C. Deformable Features for Visual Analysis

The fixed pattern is hard to obtain changeable target representation, especially for nonrigid objects. Staple [45] introduces the color histogram for visual tracking to resist target deformation. SiamAttn [46] develops a deformable Siamese attention mechanism to generate the self-attention and cross-attention between the template and the search feature. Self-attention is used to extract contextual information. Cross-attention implicitly updates the target template by aggregating the contextual interdependencies between the template and the search region. SiamAttn [46] obtains outstanding tracking performance. Deformable DETR [47] applies multiscale deformable attention modules to replace the Transformer attention [48] for processing feature maps. The deformable attention acts as a filter for the key elements in all the feature map pixels, focusing only on a few sampling positions. With flexibility of the weights, this method achieves excellent performance in detection. To obtain the complete deformation of the target, the query feature performs a per-pixel search on the target template. Moreover, we apply the segmentation mask of the initial frame as the target posterior probability to enhance the extracted target information.

III. PROPOSED METHOD

A. Overall Pipeline

Fig. 2 outlines the architecture of our method. To resist the challenges during the tracking process, our model applies three key components, i.e., target similarity matching based on CME, DFL module, and DCF-based tracker. The current frame and the given initial frame are first encoded by the backbone network (i.e., ResNet50 [36]), yielding the searching and reference features. For computational efficiency, these features are reduced to 64 channels.

The target similarity matching module refers to the attention mechanism [48], which is presented in Section III-B. The CME module expands the target feature and segmentation masks to the compact memory, which effectively overcomes the occlusion, similar objects, and appearance variation during tracking. The DFL module described in Section III-C associates the search feature and the reference feature pixel by pixel and establishes the correspondence between similar parts of the target. Then the global target correspondence relationship can completely capture the deformation information. The DCF-based tracker is used to extract target localization. We refer to deep-correlation-filter-based tracker, i.e., ATOM [20]. The backbone feature is first reduced to
64 channels by a $1 \times 1$ convolutional layer. Then, the reduced feature is processed by a $4 \times 4$ convolution layer and a continuously differentiable activation function (PELU). The position at maximum response of the activation feature is considered as the target localization. This tracking block is trained by an efficient BackProp formulation online (refer to [20] for more details). Finally, the results of target similarity matching, the deformable feature, and spatial localization are concatenated together.

To establish the pixel-level association of the target between the key memory and query, we first generate the affinity matrix $A$. Specifically, for better matching, the key memory $M_k$ and query $F_t$ are processed by per-pixel $L_2$ normalization along each channel. For convenience, they are still written as $M_k$ and $F_t$. Moreover, $M_k$ and $F_t$ are reshaped into the size of $Thw \times c$ and $hw \times c$, respectively,

$$A = F_t \ast (M_k)^T$$  \hspace{1cm} (1)

where $\ast$ denotes the matrix multiplication; $(\cdot)^T$ stands for the matrix transpose; $A \in \mathbb{R}^{(Thw) \times (Thw)}$.

The affinity matrix $A$ measures the similarity of each pixel between the query map $F_t$ and the key memory $M_k$. It requires to further retrieve the value memory map to obtain the accurate matching target. Then, the foreground and background value memory maps $M_v$ and $M_b$ are reshaped into $Thw \times 1$ and are expressed as the vectors $m_v$ and $m_b$. For $i \in [1, Thw]$, the affinity vector $a^i \in \mathbb{R}^{Thw}$ retrieves the value memory vectors $m_v \in \mathbb{R}^{Thw \times 1}$ and $m_b$ via dot product calculation

$$\tilde{s}_f^i = a^i \cdot m_v$$  \hspace{1cm} (2)

$$\tilde{s}_b^i = a^i \cdot m_b$$  \hspace{1cm} (3)

where $\tilde{s}_f^i$ and $\tilde{s}_b^i \in \mathbb{R}^{Thw}$.

A matching score with high confidence ensures the accuracy of target matching. Thereby, we apply the top-$K$ averaging function to extract the target score in the retrieved
Besides, since the targets in adjacent frames tend to be similar, it is impossible to store all historical frame information into the memory, especially for long-term videos. During tracking, for example, the video frame $f_{t-1}$ contains target information will cause memory redundancy and unnecessary matching queries.

Inspired by Monga and Evans [26], we develop a compact memory adjustment mechanism for our model, forming a diverse and compact target memory. The keys stored in the Hash data structure are different from each other. Hash also uses keys for information retrieval, which has a higher efficiency of information search. Similarly, we can construct a diverse and less redundant memory for target similarity matching. Fig. 3 illustrates the structure of the dynamic CME. The affinity matrix $A$ measures the similarity between the current query feature and the existing memory. Based on $A$, we merge the high similarity (above the upper threshold) parts between query feature and existing memory. Query feature with moderate similarity to the existing memory will be expanded into memory storage. To avoid mismatching caused by low-quality memory, low-similarity target features are directly discarded.

Most of the VOS methods based on matching [27], [49] use the initial frame as a reference template, since the initial feature labeled with ground truth has an accurate and complete target description. Besides, in view of the calculation error of the model itself, therefore, we use the target information of the first frame to initialize the memory embedding and use it as the main part of the memory. Although the target in recent video frames is close to the target of the current frame, we complete target segmentation and obtains the target query feature $F_{t-1}$, as well as the foreground $Y^f_{t-1}$ and background $Y^b_{t-1}$.
To extract useful reference information, we first compare the four is 60.0, 58.8, 59.4, and 61.2. However, improper memory embeddings produce false similarity matching. (a) No memory. (b) All memory. (c) AFB [41]. (d) CME.

Fig. 4. Visual comparisons of different memory embeddings. Our CME achieves accurate tracking and segmentation. On DAVIS2017 [35], the $J_{FM}$ score of the four is 60.0, 58.8, 59.4, and 61.2. However, improper memory embeddings produce false similarity matching. (a) No memory. (b) All memory. (c) AFB [41]. (d) CME.

segmentation masks. To extract useful reference information, we first compare $F_{t-1}$ with the existing memory key $M_k$ to find the similar parts of the two. The affinity matrix $A \in \mathbb{R}^{(h \times w) \times (h \times w)}$ generated in the target similarity matching process measures the correlation between $F_{t-1}$ and $M_k$. Therefore, we directly use $A$ for dynamic management of the memory embedding. For each feature element $F_{t-1}(i) \in \mathbb{R}^c(i \in [1, h \times w])$ in $F_{t-1}$, we search $A$ to get the maximum similarity with memory element $M_k(j) \in \mathbb{R}^c(j \in [1, Thw])$

$$\text{Re}(F_{t-1}(i)) = \max_{j \in [1, Thw]} A(i, j)$$

When $A(i, j)$ takes the maximum value, we denote $j$ as $j'$. If the maximum correlation $A(i, j')$ between the two is greater than a certain upper threshold $\zeta$, we consider the two to be similar enough. In the Hash algorithm [26], the records in the Hash map with the identical key will be inserted into the same storage space. Therefore, we only save one of the multiple similar features into the memory. Considering the diversity of memory, we use weights to merge over-similar features and the corresponding memory. This avoids unnecessary retrievals and memory redundancy. According to the above analysis, the initial reference information is most accurate. Therefore, we use a small fusion weight $\beta$ to update the current features to the existing memory embedding to avoid the interference of model errors. The online update of the memory embedding is formulated as

$$M_k(j') = \beta F_{t-1}(i) + (1 - \beta) M_k(j')$$

$$M_{of}(j') = \beta Y^f_{t-1}(i) + (1 - \beta) M_{of}(j')$$

$$M_{ob}(j') = \beta Y^b_{t-1}(i) + (1 - \beta) M_{ob}(j').$$

For the maximum correlation $\text{Re}(F_{t-1}(i)) < \zeta$, we select features with the correlation value higher than the average value

$$\text{avg}(\text{Re}(F_{t-1}(i))) = \frac{1}{2e} \left( \frac{1}{thw} \sum_{i \in wh} \text{Re}(F_{t-1}(i)) \right)$$

where $e$ is Euler’s number. Then we expand them into the existing memory to ensure the diversity of memory. Meanwhile, the query operation with irrelevant memory is avoided, so as to achieve efficient and compact memory storing via the following operations:

$$\tilde{M}_k(j') = \text{Union}(M_k(j'), F_{t-1}(i))$$

$$\tilde{M}_{of}(j') = \text{Union}(M_{of}(j'), Y^f_{t-1}(i))$$

$$\tilde{M}_{ob}(j') = \text{Union}(M_{ob}(j'), Y^b_{t-1}(i))$$

where Union(·) denotes the taking union operation of the current feature and the corresponding memory.

Fig. 4 presents the comparisons of our compact memory embedding and two other related methods. As shown in the first row, storing all historical memories and adaptive feature bank (AFB [41]) improves the discrimination of target similarity matching to a certain extent. But for complex background clutter, redundant memories cause false target matching. Our memory embedding takes the diversity and compactness of the features and can achieve much better performance of target similarity matching.

C. Deformable Feature Learning

As illustrated in Fig. 5, with the assistance of CME, target similarity matching obtains better discrimination that can well solve tracking challenges, such as similar distractors and background clutter. It also has certain advantages in solving target deformation. But it is unable to solve the severe target deformation or spatial detail variations effectively. Inspired by the graph attention mechanism [50], we propose DFL to further alleviate the aforementioned dilemma. We construct the association between each pixel in the query $F$ and the whole key $R$ to capture complete target deformation information. Since the query and the key contain different representation of the target, for per pixel vector $f^i \in \mathbb{R}^c(i \in [1, w \times h])$ in the query, we apply the nonshared transformations and calculate the association of each pixel vector $r^j \in \mathbb{R}^c(j \in [1, w \times h])$ in the key to it, yielding the learnable pixelwise similarity function

$$z_{ij} = (W_F f^i)^T (W_R r^j)$$

where $W_F$ and $W_R$ indicate the learnable linear transformations which transform $f^i$ and $r^j$ into higher level representation.

To facilitate the comparison of the similarity between $f^i$ and different parts in $R$, we normalize $z_{ij}$ across all the pixels of
of the target completely. The DFL effectively solves this problem and realizes accurate segmentation and tracking of the target. (a) Baseline. (b) +CME. (c) +CME + DFL.

R via the softmax function, yielding the normalized similarity weight

\[ d_{ij} = \text{softmax}(z_{ij}) = \frac{\exp(z_{ij})}{\sum_{r \in R} \exp(z_{ij})} \]  

(16)

The normalized pixelwise similarity weight coefficients \( d_{ij} \) are used to aggregate the pixel-by-pixel features in R, thus generating the weighted deformation feature corresponding to \( f^t \) as

\[ v_i = \sum_{r \in R} d_{ij} \phi_v(r^t) \]  

(17)

where \( \phi_v(\cdot) \) denotes ReLU\((W_v * (\cdot))\) aiming to extract higher level representation of \( r^t \). Then we apply residual connection [36] to cascade the transformed query feature and the weighted deformation feature together, yielding the enhanced feature that contains the deformation information as

\[ \tilde{f} = \phi_v(\text{Concat}(v_i, \phi_v(f^t))) \]  

(18)

where \( \phi_v(\cdot) \) denotes ReLU\((W_v * (\cdot))\) aiming at reducing the dimensionality of features, and Concat(\(\cdot\)) denotes the cascade operation.

There are some unavoidable background mismatches in the matched target deformation features. Therefore, we use the target posterior probability (i.e., the target segmentation mask) \( P \) of the initial frame to enhance the confidence of the deformation feature. Specifically, the obtained target deformation \( \tilde{f} \) retrieves the foreground probability \( p^t_i \) in the initial frame to generate the final deformable feature by dot product

\[ \hat{f}^t = \tilde{f} \cdot p^t_i. \]  

(19)

D. Tracking With CME and DFL

This section outlines the application of the proposed CME and DFL to general visual object tracking.

1) Initialization: In the VOT task, a video sequence merely contains a given bounding box label. We first generate a pseudomask with the ground-truth box in the initial frame. Then, the pseudolabel is used to initialize our model and is converted into a more accurate target mask. The converted target mask is used to initialize the CME module and the template of the DFL module.

2) Tracking: During tracking, we extract an image patch four times the size of the target as the query area at the previous target localization. Then, the query region is processed by our CMEDFL model and obtains the target segmentation mask \( Y^t \). Moreover, the sum of each pixel in the background mask \( Y^b \) and the corresponding point in \( Y^t \) is 1. Meanwhile, the obtained query feature \( F_t \) and target segmentation mask are used to update the CME in an online manner. Finally, the output of our tracking model is the target segmentation mask. However, most visual object tracking benchmarks take the bounding box as the tracking result. Therefore, we first binarize the segmentation mask with a threshold of 0.5. Then the target region is extracted on the binary mask using the contour detection function in the OpenCV function library. We use the bounding box fitting method proposed in the VOT [28] benchmark to refine the extracted target region into the target bounding box as the final tracking result.

IV. EXPERIMENTS

A. Implementation Details

In this work, we use the first four stages of ResNet50 [36] pretrained on ImageNet as the backbone network to extract features. Object segmentation is a task of pixel-level classification which needs to use features with high confidence semantics. Therefore, we extract the fourth stage of the backbone network for target similarity matching and deformation feature extraction. Then, the backbone features are reduced to 64 channels via \( 1 \times 1 \) convolution layer followed by a \( 3 \times 3 \) convolution layer and ReLU activation. In the upsampling segmentation process, we use the first three stages of the backbone to supplement the target spatial detail information. We set the top-K as \( K = 3 \). The upper threshold \( \zeta \) of similar memory merging is set to 0.90. The fusion weight \( \beta \) is set to 0.001. The above settings are fixed in all the related experiments.

Network Training: Both the CME and DCF-based localization modules are updated online without being pretrained. But target similarity matching, deformable feature extracting, and upscaling segmentation network are pretrained on the Youtube-VOS [51] dataset. Similar to the sampling strategy use in the Siamese-network-based tracking model, a pair of
images with masks are sampled from the video sequence to construct the training samples. We minimize the cross-entropy loss via the Adam optimizer [52] with a learning rate of $8 \times 10^{-4}$ that has 0.2 decay every 15 epochs. The whole training process takes 22 h on an Nvidia Titan XP graphics card.

B. Evaluation on Tracking Datasets

To verify the effectiveness of the proposed model, we conduct extensive evaluations on six popular and challenging tracking benchmarks, including VOT2016 [28], VOT2018 [29], VOT2019 [30], GOT-10k [31], TrackingNet [32], and LaSOT [33]. In all the comparisons, the best three results are labeled in red, blue, and green fonts, respectively. Then, we will analyze in detail the experimental results obtained on each dataset.

The VOT datasets [28], [29], [30] are currently one of the most convincing and active visual tracking benchmarks. Each dataset contains 60 color video sequences with refined rotated bounding box annotations. VOT updates and supplements the data every year, which puts higher requirements on the performance of the trackers. The official VOT toolkit [28] provides three evaluation criteria, i.e., accuracy (average overlap), robustness (or reliability which is related to failure rate), and expected average overlap (EAO). EAO comprehensively considers the accuracy and robustness of the trackers. These benchmarks usually rank trackers by EAO score.

1) Quantitative Results on VOT2016: We compare our model with 18 state-of-the-art trackers on the VOT2016 [28] dataset, including the tracking with the segmentation methods (CSR-DCF [39], SiamMask [24], D3S [25]), DCF-based trackers (ASRCF [21], CCOT [18], Staple [45], ATOM [20]), and recent deep trackers (SPM [11], Update-Net [53], SiamRPN++ [9], SiamRPN [7], C-RPN [8], SiamAttn [46], ROAM++ [54], and MenTrack [44]). Besides, CCOT [18] is the best performing tracker on VOT2016 challenge [28]. Table I reports the comparison results. Our model achieves almost the best performance with the excellent EAO score of 0.563 and accuracy score of 0.671. Compared with the second and third best methods SiamAttn [46] and D3S [25], our CMEDFL improves EAO score by 2.6% and 7%, respectively. Besides, compared with recent trackers ATOM [20] and ROAM++ [54], CMEDFL obtains the EAO score gain of 13.3% and 12.2%. In terms of accuracy score, SiamAttn [46] achieves the best score with 0.680, which surpasses CMEDFL by 0.9%. CMEDFL outperforms the recent state-of-the-art trackers, such as D3S [25], ROAM++ [54], ATOM [20], SiamRPN++ [9], and SiamMask [24] by 1.1%, 7.2%, 6.1%, 3.1%, and 3.2%, respectively. Therefore, the above comparison results prove the effectiveness of the proposed compact memory embedding and deformation feature learning method.

2) Quantitative Results on VOT2018: On VOT2018 [29], our method is compared with 23 state-of-the-art trackers, i.e., anchor-based and anchor-free Siamese-network-based methods (SiamRPN++ [9], C-RPN [8], SiamMask [24], SiamAttn [46], RDTrack [56], SiamBAN [14] and Ocean [15]), the meta-learning based method (Retina-MAML [55], ROAM++ [54], SiamRPN++ [9], and SiamMask [24]), and the DCF-based trackers (ATOM [20], DiMP [23], ECO [17], LADCF [22] which is the champion tracker of VOT2018). As shown in Table II, our method obtains the top-ranking performance with the best EAO of 0.525 and accuracy of 0.641, which exhibits obvious superiority over other methods. Both D3S [25] and Ocean-on [15] achieve remarkable performance with the EAO score of 0.489. Compared with the recent state-of-the-art trackers, such as RDTrack [56], SiamAttn [46], and ROAM++ [54], our CMEDFL achieves the EAO score gain of 5.5%, 5.5%, and 14.5%, respectively. Besides, we can observe that the segmentation-based trackers (D3S [25], SiamMask [24], and the DCF-based trackers (ATOM [20], DiMP [23], ECO [17], LADCF [22]) which is the champion tracker of VOT2018) perform better than our method in terms of accuracy score.

3) Quantitative Results on VOT2019: On VOT2019 [30], our method is compared with 13 state-of-the-art deep trackers related to the Siamese-network-based trackers [9], [10], [11],...
TABLE III

| Model                  | Source          | Accuracy  | Robustness | EAO  |
|------------------------|-----------------|-----------|------------|------|
| MemiTrack [44]         | ECCV2018        | 0.485     | 0.587      | 0.228|
| SPM [11]               | CVPR2019        | 0.577     | 0.507      | 0.275|
| SiamRPN++ [9]          | CVPR2019        | 0.580     | 0.446      | 0.292|
| SiamMask [24]          | CVPR2019        | 0.594     | 0.461      | 0.287|
| SiamD[10]              | CVPR2019        | 0.600     | 0.467      | 0.299|
| ATOM [20]              | ICCV2019        | 0.603     | 0.411      | 0.301|
| DiMP [23]              | ICCV2019        | 0.582     | 0.371      | 0.321|
| Ocean [15]             | CVPR2019        | 0.590     | 0.376      | 0.325|
| Ocean++ [15]           | CVPR2020        | 0.594     | 0.316      | 0.350|
| Retina-MAML [55]       | CVPR2020        | 0.570     | 0.366      | 0.313|
| SiamBAN [14]           | CVPR2020        | 0.602     | 0.396      | 0.327|
| RDTrack [56]           | CVPR2021        | 0.593     | 0.306      | 0.341|
| LightTrack-B [57]      | CVPR2021        | 0.552     | 0.310      | 0.357|
| CMEDFL                 | Ours            | 0.643     | 0.286      | 0.368|

TABLE IV

| Model                  | Source          | SR0.75  | SR0.5  | AO   |
|------------------------|-----------------|--------|--------|------|
| SiamFC [5]             | ECCVW2016       | 53.3   | 60.3   | 57.1 |
| Staple [45]            | CVPR2016        | 47.0   | 60.3   | 52.8 |
| ECO [17]               | CVPR2017        | 49.2   | 61.8   | 55.4 |
| CSR-DCF [19]           | CVPR2017        | 48.0   | 62.2   | 53.4 |
| CFNet [37]             | CVPR2017        | 53.3   | 65.4   | 57.8 |
| C-RPN [8]              | CVPR2019        | 61.9   | 74.5   | 66.9 |
| Update-Net [53]        | ICCV2019        | 62.5   | 75.1   | 67.7 |
| ATOM [20]              | ICCV2019        | 64.8   | 77.1   | 70.3 |
| SiamGraph [58]         | MM2020          | 63.8   | 77.1   | 70.9 |
| SiamFC++ [12]          | AAA2020         | 64.6   | 75.8   | 71.2 |
| ROAM++ [54]            | CVPR2020        | 62.3   | 75.4   | 67.0 |
| LightTrack-B [57]      | CVPR2021        | 70.8   | 78.9   | 73.3 |
| CMEDFL                 | Ours            | 65.1   | 76.1   | 71.3 |

TABLE V

| Model                  | Source          | Precision | N-Precision | Success  |
|------------------------|-----------------|-----------|-------------|----------|
| SiamFC [5]             | ECCVW2016       | 53.3      | 60.3        | 57.1     |
| Staple [45]            | CVPR2016        | 47.0      | 60.3        | 52.8     |
| ECO [17]               | CVPR2017        | 49.2      | 61.8        | 55.4     |
| CSR-DCF [19]           | CVPR2017        | 48.0      | 62.2        | 53.4     |
| CFNet [37]             | CVPR2017        | 53.3      | 65.4        | 57.8     |
| C-RPN [8]              | CVPR2019        | 61.9      | 74.5        | 66.9     |
| Update-Net [53]        | ICCV2019        | 62.5      | 75.1        | 67.7     |
| ATOM [20]              | ICCV2019        | 64.8      | 77.1        | 70.3     |
| SiamGraph [58]         | MM2020          | 63.8      | 77.1        | 70.9     |
| SiamFC++ [12]          | AAA2020         | 64.6      | 75.8        | 71.2     |
| ROAM++ [54]            | CVPR2020        | 62.3      | 75.4        | 67.0     |
| LightTrack-B [57]      | CVPR2021        | 70.8      | 78.9        | 73.3     |
| CMEDFL                 | Ours            | 65.1      | 76.1        | 71.3     |

4) Quantitative Results on GOT-10k: GOT-10k [31] is a large-scale tracking dataset consisting of over 10,000 sequences and 1.5 million annotations of the axis-aligned bounding box. The trackers are evaluated on the test set consisting of 180 videos via an online server. It uses the average overlap (AO) and success rate (SR) as evaluation criteria. We compare our method to 14 state-of-the-art trackers containing DCF-based trackers [17], [20], [37], [39], [45], Siamese-network-based trackers [5], [7], [9], [11], [12], [13], [24], meta-learning based method [54], and tracker with neural architecture search (NAS) [57]. The evaluation results are presented in Table V. LightTrack-B [57] achieves the best success and precision scores, which exceeds CMEDFL by 2% and 5.7%, respectively. Actually, the gap is owing to the fact that LightTrack-B [57] consumes more training data than CMEDFL. CMEDFL obtains the second best precision score with 65.1 and surpasses SiamFC++ [12], SiamGraph [58], ATOM [20], and ROAM++ [54] by 0.5%, 1.3%, 0.3%, and 2.8%, respectively. Besides, our method achieves the second best-performing success score of 71.3, which obtains 0.4%, 1%, and 4.3% success score gain than SiamGraph [58], ATOM [20], and ROAM++ [54], respectively. In our model, both deformable features and compact memory contribute to favorable performance.

5) Quantitative Results on LaSOT: LaSOT [33] is a large-scale visual tracking dataset. The testing set contains 280 video sequences. We conduct the comparisons between our method and 13 state-of-the-art trackers showing in the benchmark [33]. These methods consist of deep-neural-network-based trackers (such as SiamFC [5], VITAL [59], MDNet [60], SINT [6], CFNet [37], C-RPN [8], ROAM++ [54], and SiamDW [10]) and DCF-based tracker [17], [21], [39], [45]. Fig. 6 illustrates the comparison results in terms of precision and success. Our CMEDFL obtains 0.426 precision score and 0.441 success score. C-RPN [8] achieves the best success score and exceeds CMEDFL by 1.4%. Compared with 12 excellent trackers including the Siamese-network-based trackers [5], [8], [12], [53], [58], meta-learning based tracker [54], DCF-based trackers [17], [20], [37], [39], [45], and NAS-based method [57]. The evaluation results are presented in Table V. LightTrack-B [57] achieves the best success and precision scores, which exceeds CMEDFL by 2% and 5.7%, respectively. Actually, the gap is owing to the fact that LightTrack-B [57] consumes more training data than CMEDFL. CMEDFL obtains the second best precision score with 65.1 and surpasses SiamFC++ [12], SiamGraph [58], ATOM [20], and ROAM++ [54] by 0.5%, 1.3%, 0.3%, and 2.8%, respectively. Besides, our method achieves the second best-performing success score of 71.3, which obtains 0.4%, 1%, and 4.3% success score gain than SiamGraph [58], ATOM [20], and ROAM++ [54], respectively. In our model, both deformable features and compact memory contribute to favorable performance.

6) Quantitative Results on TrackingNet: The Tracking-Net [32] is a large-scale dataset aimed at object tracking in the Wild. The testing set consists of 511 video sequences. The trackers are evaluated via an online server. It uses three metrics to evaluate the trackers, i.e., precision, normalized precision (N-precision), and success. We conduct the comparisons with 12 excellent trackers including the Siamese-network-based trackers [5], [8], [12], [53], [58], meta-learning based tracker [54], DCF-based trackers [17], [20], [37], [39], [45], and NAS-based method [57]. The evaluation results are presented in Table V. LightTrack-B [57] achieves the best success and precision scores, which exceeds CMEDFL by 2% and 5.7%, respectively. Actually, the gap is owing to the fact that LightTrack-B [57] consumes more training data than CMEDFL. CMEDFL obtains the second best precision score with 65.1 and surpasses SiamFC++ [12], SiamGraph [58], ATOM [20], and ROAM++ [54] by 0.5%, 1.3%, 0.3%, and 2.8%, respectively. Besides, our method achieves the second best-performing success score of 71.3, which obtains 0.4%, 1%, and 4.3% success score gain than SiamGraph [58], ATOM [20], and ROAM++ [54], respectively. In our model, both deformable features and compact memory contribute to favorable performance.
with SiamDW [10], ECO [17], and other models, CMEDFL achieves a certain improvement of success score. ROAM++ [54] outperforms CMEDFL by 0.3 in terms of precision score. However, compared with other trackers, CMEDFL achieves a better precision score.

7) Qualitative Results: We conduct visualization experiments on the VOT2016 [28] and GOT-10k [31] datasets. Fig. 7 illustrates the visualization results of our CMEDFL. These video sequences contain object deformation, similar distractors, background clutter, and other tracking challenges. The first two lines are the deformation case. Although these targets undergo drastic structural changes, our CMEDFL can still segment and track them well. Thus, the effectiveness of the proposed DFL method is verified. The similar distractor case is illustrated in the middle two lines. Similar distractor is one of the inherent challenges in tracking, especially when the target and similar objects block each other, and it is easy to cause tracking drift. With the help of CME, our CMEDFL tracker overcomes the obstruction of similar targets successfully. The last two lines illustrate the background clutter case. Although the boundary between complex background and target is fuzzy, our method can still segment and track the target well. CMEDFL achieves superior segmentation and tracking performance, which verifies that our tracker has certain adaptability and generalization for changes in tracking scene.

C. Evaluation on VOS Task

1) Quantitative Results on DAVIS2017: To further verify the segmentation performance of our model, we conduct evaluations on a popular VOS dataset, i.e., DAVIS2017 [35]. DAVIS2017 [35] takes the mean Jaccard index ($\mathcal{J}_M$) and mean F-measure ($\mathcal{F}_M$) as measures to evaluate the performance of each model. We compare our method with numerous VOS methods [27], [34], [40], [49], [61], [62], [63], [64], [65] and segmentation-based trackers [24], [25]. Table VI reports the comparison results. Compared with D3S [25] and SiamMask [24], our tracker achieves the $\mathcal{J}_M$ score gain of 1.9% and 5.4%, respectively. Although D3S [25] surpasses our method by 1.1% in terms of $\mathcal{F}_M$ score, our tracker obtains the best $\mathcal{J}_M$ score among three segmentation-based trackers. In terms of $\mathcal{F}_M$ score, our CMEDFL outperforms D3S and SiamMask by 0.4% and 4.8%, respectively. Even compared with some VOS models, i.e., FAVOS [34], VM [27], and OSMN [49], our method still has superior performance. Moreover, new state-of-the-art VOS methods, such as RSTCN [61] and R50-AOT-L [62], achieve excellent performance by more elaborate pretraining and main training. In terms of inference speed, SiamMask [24] directly uses the cross-correlated map as the segmentation input, so it realizes a high processing speed with 55 fps. Since our CMEDFL adopts the memory matching method which contains large calculation, and the tracking speed is 21 fps. The segmentation-based trackers only apply the light segmentation network. While achieving superior segmentation accuracy, their inference speed is significantly higher than the VOS methods [61], [62].

2) Qualitative Results: Fig. 8 illustrates the qualitative results of our tracker on DAVIS2017. Even if the object undergoes occlusion, deformation, and similar distractors, our method still achieves accurate target segmentation.

D. Ablation Study

1) Componentwise Comparison: To verify the efficacy of the proposed method, we conduct an ablation study of the proposed modules on VOT2016 [28], VOT2018 [29], and VOT2019 [30]. The experimental results are listed in Table VII. The baseline model contains a backbone network (as introduced in Section III-A) and target similarity matching module without memory embedding (as introduced in Section III-B). DFL$_N$ represents the target deformation feature without using the target mask of the initial frame.
TABLE VII
COMPONENTWISE COMPARISONS OF THE PROPOSED METHOD ON VOT2016, VOT2018, AND VOT2019 BENCHMARKS

| Methods         | VOT2016 | VOT2018 | VOT2019 |
|-----------------|---------|---------|---------|
|                 | Accuracy | Robustness | EAO     | Accuracy | Robustness | EAO     | Accuracy | Robustness | EAO     |
| Baseline        | 0.642   | 0.135   | 0.480   | 0.639   | 0.159   | 0.473   | 0.632   | 0.321   | 0.301   |
| +DFL_N          | 0.655   | 0.154   | 0.498   | 0.633   | 0.159   | 0.482   | 0.631   | 0.321   | 0.309   |
| +SelfA          | 0.650   | 0.131   | 0.496   | 0.629   | 0.164   | 0.482   | 0.628   | 0.326   | 0.300   |
| +NonL           | 0.643   | 0.121   | 0.500   | 0.628   | 0.140   | 0.489   | 0.620   | 0.312   | 0.302   |
| +DFL            | 0.657   | 0.112   | 0.520   | 0.641   | 0.150   | 0.496   | 0.640   | 0.336   | 0.313   |
| +AME            | 0.656   | 0.163   | 0.517   | 0.619   | 0.187   | 0.488   | 0.619   | 0.341   | 0.313   |
| +AFB[41]        | 0.637   | 0.154   | 0.516   | 0.612   | 0.183   | 0.498   | 0.606   | 0.351   | 0.318   |
| +CME            | 0.666   | 0.149   | 0.525   | 0.638   | 0.187   | 0.504   | 0.637   | 0.356   | 0.317   |
| +AME + DFL      | 0.667   | 0.138   | 0.524   | 0.634   | 0.183   | 0.505   | 0.629   | 0.316   | 0.344   |
| +AFB + DFL      | 0.661   | 0.154   | 0.523   | 0.619   | 0.187   | 0.500   | 0.607   | 0.296   | 0.360   |
| +CME + SelfA    | 0.663   | 0.140   | 0.534   | 0.620   | 0.169   | 0.500   | 0.623   | 0.336   | 0.333   |
| +CME + NonL     | 0.645   | 0.163   | 0.511   | 0.608   | 0.183   | 0.506   | 0.585   | 0.341   | 0.306   |
| +CME + DFL_N    | 0.669   | 0.135   | 0.541   | 0.632   | 0.150   | 0.506   | 0.636   | 0.301   | 0.333   |
| +CME + DFL     | 0.671   | 0.135   | 0.563   | 0.641   | 0.169   | 0.525   | 0.643   | 0.286   | 0.368   |

Fig. 8. Visual evaluations of our method on DAVIS2017. The sequences contain various challenges (occlusion, deformation, etc.). The target objects are marked in red, green, yellow, and others. “Ours” indicates the inference result of CMEDFL. “GT” denotes ground truth.

for postprocessing. DFL is the target deformation feature processed by the target mask of the initial frame. “SelfA” indicates the self-attention module [48]. We transform query features F to query, and transform reference feature to key and value. “Non” is the nonlocal attention module [66]. The settings of query, key, and value are similar to SelfA. AME indicates that we store the target information in all historical frames as memory. AFB is the adaptive memory embedding [41]. CME represents the proposed dynamic compact memory embedding.

Baseline obtains the EAO scores of 0.480, 0.473, and 0.301 on VOT2016, VOT2018, and VOT2019, respectively. Besides, the achieved accuracy scores are 0.642, 0.639, and 0.632. With the help of the DFL module, the EAO score of baseline increases to 0.520, 0.496, and 0.313, achieving obvious gain of 4%, 2.3%, and 1.2%, respectively. Although “+SelfA” and “+NonL” improve the EAO scores of baseline both on VOT2016 and VOT2018, there is a certain performance gap compared with “+DFL.” This verifies that our proposed DFL can capture finer spatial details of the target than general attention methods [48], [66]. DFL enables to promote more robust tracking performance. In addition, “+DFL_N” improves the tracking performance of baseline to a certain extent. Specifically, on VOT2016, VOT2018, and VOT2019, an increase of 1.9%, 0.9%, and 0.8% is achieved, respectively. Yet, compared with “+DFL,” there is a gap of 2.2%, 1.4%, and 0.4% EAO scores on VOT2016, VOT2018, and VOT2019, respectively.

Compared with a single target template, memory can provide richer information reference for target similarity matching. Therefore, the above three types of memory embedding, “+AME,” “+AFB,” and “+CME,” achieve certain performance improvement compared with baseline. For example, on the VOT2016 benchmark, the above three memory embeddings obtain 3.7%, 3.6%, and 4.5% of EAO scores improvement, respectively, compared with baseline. On VOT2018, the EAO score gains of 1.5%, 2.5%, and 3.1% is achieved, respectively. The compact memory removes redundant historical reference information, ensuring the credibility of target similarity matching. “+CME” significantly improves the EAO score of the baseline on VOT2016, VOT2018, and VOT2019 by 2.2%, 3.1%, and 1.6%, respectively. In terms of EAO and accuracy scores, “+CME” shows certain performance advantages compared with “+AME” and “+AFB.” For example, on VOT2018 [29], “+CME” exceeds “+AME” and “+AFB” 1.6% and 0.6% EAO scores, and achieves 1.9% and 2.6% accuracy scores’ improvement, respectively.

Finally, we combine the well-performing +CME with two deformation feature learning modules (DFL and DFL_N), as well as two general attention methods (SelfA and NonL). With the help of compact memory embedding CME, the EAO scores of “+DFL” and “+DFL_N” are improved by 4.3% on VOT2016, and by 2.4% and 5.5% on VOT2019, respectively. The tracking performance of “+SelfA” and “+NonL” is also improved to a certain extent. However, their accuracy and
EAO scores are lower than “+CME + DFL.” Therefore, the effectiveness of CME is further verified. Besides, we evaluate the combinations of the DFL module and the above three memory embeddings. It can be found that the deformation feature learning module “+DFL” can promote the tracking performance of the above three memory embedding models. For example, on the VOT2019 dataset, “+AME + DFL,” “+AFB + DFL,” and “+CME + DFL” achieve 3.1%, 4.2%, and 5.1% EAO scores gains compared with “+AME,” “+AFB,” and “+CME,” respectively. Moreover, on the VOT2018 dataset, the EAO score gains of 1.7%, 0.2%, and 2.1% are also achieved, respectively. Therefore, the effectiveness of the deformation feature learning module DFL proposed in the article is also verified.

2) Attribute Evaluation: To further verify the effectiveness of the proposed CME and DFL for addressing challenges, i.e., appearance change, deformation, and others. We conduct video attribute evaluation on VOT2016 [28]. VOT2016 is annotated with five attributes, including OCC (occlusion), MC (motion-change), SC (size-change), IC (illumination-change), and CM (camera-motion). OA indicates the overall video frames. UA denotes those frames without labeling with above five attributes. Table VIII reports the EAO score of each method. Compared with baseline and “+DFL,” “+CME” shows stronger discrimination in complex scenarios of OCC, MC, and IC. Especially in IC, the target undergoes appearance changes. Besides, the tracker is prone to failure when similar objects occlude each other in OCC. They all bring great challenges to the matching-based tracking methods. “+CME” surpasses baseline and “+DFL” by 4.9% and 2.2% in OCC, and by 3.3% and 2.4% in IC, respectively. Therefore, the effectiveness of the proposed CME for overcoming appearance variations and similar distractors is validated. In SC, the object usually undergoes deformation. “+DFL” achieves more robust performance than baseline and “+CME.” Specifically, “+DFL” surpasses baseline and “+CME” by 6.9% and 3.9%. With the combination of CME and DFL, the tracking performance of CMEDFL is further improved. Therefore, CME and DFL are capable of overcoming the challenges of OCC, IC, and SC, respectively.

3) Influence of Parameters: In Section III-B1, when the similarity value between the current target feature and the existing memory is higher than the upper threshold $\zeta$, we think the two are similar enough to be the same. Therefore, the parameter $\zeta$ measures the degree of similarity between the current target feature and the existing memory. To reduce the redundancy of the memory, we use a fusion weight $\beta$ to merge the highly similar parts between the current target feature and the existing memory. We fix $\beta$ and set it as 0.001. Then, we explore the effect of $\zeta$ on our CMEDFL on the VOT2016 [28] dataset. As shown in Fig. 9, the accuracy score of CMEDFL is relatively stable for the setting of $\zeta$. But for EAO score, the tracking performance of CMEDFL fluctuates slightly, and when $\zeta$ is set to 0.90, the best EAO score is achieved. Besides, we fix $\zeta$ and set it as 0.90. Fig. 10 illustrates the effects of $\beta$ on CMEDFL on the VOT2016 [28] dataset. As $\beta$ increases, the accuracy and EAO score of CMEDFL decrease. When $\beta$ is set to 0.001, CMEDFL achieves the best EAO score.

4) Failing and Recovering Cases: The overlapping occlusion of adjacent similar targets in a short time can easily lead to tracking failure. Especially for the tracking model updated online, it is easy to cause unreliable model update and even model corruption. This challenge has high requirements for the discrimination of the trackers. Fig. 11 illustrates the tracking performance of our CMEDFL under this case. We can find that when the target is occluded, similar object close to the target will cause great interference to the tracking model, which makes the model make wrong judgment. We introduce dynamic compact memory into the model. Moreover,
the memory stores the feature information of the target in historical frames, so it can provide rich information reference for target tracking to avoid tracking drift. Although the target undergoes tracking error in a short time when it is blocked, our model can still track correctly once the target is restored to the field of view.

5) Comparisons of Memory Size: Table IX reports the comparison results of graphics card storage usage among four memory methods. We record the graphics card usage on each frame of the above four methods. The minimum, maximum, and average values are selected as the evaluation criteria. “+DFL” does not use memory, so its storage usage remains at 1.217 GB. The average size of CMEDFL is about twice of +DFL. However, compared with “+AFB +DFL” and “+AME +DFL,” the average and maximum size of “+CME +DFL” are only 50% and 33% of theirs approximately. “+CME +DFL” achieves both compact memory and robust tracking performance simultaneously.

6) Inference Speed: On the DAISY2017 [35] dataset, our trackers achieve the inference speed of 21 fps. On the GOT-10k [31], TrackingNet [32], and VOT [28], [29], [30] benchmarks, the tracking speed of the CMEDFL tracker is 19 fps.

V. CONCLUSION

In this article, we propose to learn dynamic CME and deformable feature for deformable visual object tracking. The CME takes the initial target feature as the basis. We maintain the compactness and diversification of memory by consulting the idea of Hash map. It absorbs the features related to the existing memory in the tracking process, thus avoiding the target mismatch caused by uncorrelated memory and improving the discrimination of the tracking model effectively. The DFL module establishes a weighted correlation between the pixelwise query feature and the entire template. The captured target deformation complements detailed information for target segmentation. The favorable performance achieved on seven challenging benchmarks including the VOT and VOS tasks all verify the effectiveness of our model.

REFERENCES

[1] J. Xing, H. Ai, and S. Lao, “Multiple human tracking based on multi-view upper-body detection and discriminative learning,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 1698–1701.

[2] L. Liu, J. Xing, H. Ai, and X. Ruan, “Hand posture recognition using finger geometric feature,” in Proc. IEEE Int. Conf. Pattern Recognit. (ICPR), Nov. 2012, pp. 565–568.

[3] K.-H. Lee and J.-N. Hwang, “On-road pedestrian tracking across multiple driving recorders,” IEEE Trans. Multimedia, vol. 17, no. 9, pp. 1429–1438, Sep. 2015.

[4] S. Tang, M. Andriluka, B. Andres, and B. Schiele, “Multiple people tracking by lifted multicut and person re-identification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3701–3710.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional Siamese networks for object tracking,” in Proc. Comput. Vis. ECCV Workshops, vol. 9914, 2016, pp. 850–865.

[6] R. Tao, E. Gavves, and A. W. M. Smeulders, “Siamese instance search for tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1420–1429.

[7] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with Siamese region proposal network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8971–8980.

[8] H. Fan and H. Ling, “Siamese cascaded region proposal networks for real-time visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7952–7961.

[9] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “SiamRPN++: Evolution of Siamese visual tracking with very deep networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4282–4291.

[10] Z. Zhang and H. Peng, “Deeper and wider Siamese networks for real-time visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4591–4600.

[11] G. Wang, C. Luo, Z. Xiong, and W. Zeng, “SPM-tracker: Series-parallel matching for real-time visual object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3643–3652.

[12] Y. Xu, Z. Wang, Z. Li, Y. Ye, and G. Yu, “SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines,” in Proc. Conf. Artif. Intell. (AAAI), 2020, pp. 12549–12556.

[13] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, “SiamCAR: Siamese fully convolutional classification and regression for visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6268–6276.

[14] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, “Siamese box adaptive network for visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6667–6676.

[15] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, “Ocean: Object-aware anchor-free tracking,” in Proc. Eur. Conf. Comput. Vis., vol. 12366, 2020, pp. 771–787.

[16] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 2544–2550.

[17] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ECO: Efficient convolution operators for tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6931–6939.

[18] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in Proc. Eur. Conf. Comput. Vis., vol. 9909, 2016, pp. 472–488.

[19] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, “Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking,” IEEE Trans. Image Process., vol. 28, no. 11, pp. 5596–5609, Nov. 2019.

[20] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ATOM: Accurate tracking by overlap maximization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4660–4669.

[21] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, “Visual tracking via adaptive spatially-regularized correlation filters,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4670–4679.

[22] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, “Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking,” IEEE Trans. Image Process., vol. 28, no. 11, pp. 5596–5609, Nov. 2019.

[23] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, “Learning discriminative model prediction for tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6181–6190.

[24] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast online object tracking and segmentation: A unifying approach,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1328–1338.

[25] A. Lukezic, J. Matas, and M. Kristan, “D3S—A discriminative single shot segmentation tracker,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7131–7140.

[26] V. Monga and B. L. Evans, “Perceptual image hashing via feature points: Performance evaluation and tradeoffs,” IEEE Trans. Image Process., vol. 15, no. 11, pp. 3452–3465, Nov. 2006.

[27] Y. Hu, J. Huang, and A. G. Schwing, “VideoMatch: Matching based video object segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 11212, Sep. 2018, pp. 56–73.
Kaihua Zhang received the B.S. degree in technology and science of electronic information from the Ocean University of China (OUC), Qingdao, China, in 2006, the M.S. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, China, in 2009, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2013. From August 2009 to August 2010, he worked as a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include image segmentation, level sets, and visual tracking.

Yu Wang received the B.S. degree in communication engineering, the M.S. degree in software engineering, and the Ph.D. degree in computer applications and techniques from Tianjin University, Tianjin, China, in 2013, 2016, and 2020, respectively. He is currently an Assistant Professor with Tianjin University. He has published more than 20 peer-reviewed articles including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His research interests are data mining and machine learning, especially multigranularity learning in open and dynamic environment for computer vision and industrial applications.

Shuai Zhao is currently the Director of the Intelligent and Connected Vehicles (ICV) Data Department and China Automotive Data Company Ltd., the Coordination Expert of International Standards and Regulations of China Automotive Standards Committee, and a member of Intelligent Driving Special Committee of China Artificial Intelligence Society and China-Association for Standardization of Automation and Measuring System (C-ASAM) Steering Committee. He is mainly engaged in the related work of intelligent networked vehicle scenario database and simulation, participated in the compilation of development and application of automatic driving test scenario technology and other works, and participated in the formulation of standards and policies for intelligent vehicle scene simulation at home and abroad.

Lei Wang is currently an Algorithm Engineer of the China Automotive Technology and Research Center, Tianjin, China. He is mainly engaged in the related work of image recognition, detection, and tracking algorithm regarding intelligent networked vehicle area, deeply participating in the work of radar’s detection and denoising technology based on the machine learning algorithm, simultaneously specialized in the area of LiDAR state-of-the-art deep learning recognition algorithm.

Tianzhu Zhang (Member, IEEE) is currently a Professor with the University of Science and Technology of China (USTC), Hefei, China. He has authored or coauthored over 70 journal and conference papers in these areas, including over 40 IEEE/ACM Transactions articles [TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI)/TRANSACTIONS ON IMAGE PROCESSING (TIP)/INTERNATIONAL JOURNAL OF COMPUTER VISION (IJCV)] and over 30 top-tier computer vision conference papers [IEEE International Conference on Computer Vision (ICCV)/IEEE Conference on Computer Vision and Pattern Recognition (CVPR)/European Conference on Computer Vision (ECCV)]. His research interests include pattern recognition, computer vision, multimedia analysis, and machine learning.

Prof. Zhang has served as the Area Chair for ECCV 2020, the Association for Computing Machinery (ACM) Multimedia 2020, CVPR 2020, ICCV 2019, and ACM Multimedia 2019, an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) and Neurocomputing, and the Publicity Chair for the International Conference on Internet Multimedia Computing and Service (ICIMCS) 2015 and ACM Multimedia Asia 2019.

Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He joined the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, as a Post-Doctoral Fellow. He became a Full Professor with Tianjin University, Tianjin, China, in 2012. He is currently a Chair Professor and the Deputy Dean of the College of Intelligence and Computing. He has published more than 300 peer-reviewed articles in the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His research interests are focused on uncertainty modeling, multimodality learning, incremental learning, and continual learning these years, funded by National Natural Science Foundation of China and The National Key Research and Development Program of China. Prof. Hu was a recipient of the Best Paper Award from the International Conference on Machine Learning and Computing (ICMLC) in 2015 and the IEEE International Conference on Multimedia and Expo (ICME) in 2021. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, ACTA AUTOMATICA SINICA, and ACTA ELECTRONICA SINICA.