Video-based Person Re-identification with Accumulative Motion Context
Hao Liu, Zequn Jie, Karlekar Jayashree, Meibin Qi, Jianguo Jiang and Shuicheng Yan, Fellow, IEEE, Jiashi Feng

Abstract—Video based person re-identification plays a central role in realistic security and video surveillance. In this paper we propose a novel Accumulative Motion Context (AMOC) network for addressing this important problem, which effectively exploits the long-range motion context for robustly identifying the same person under challenging conditions. Given a video sequence of the same or different persons, the proposed AMOC network jointly learns appearance representation and motion context from a collection of adjacent frames using a two-stream convolutional architecture. Then AMOC accumulates clues from motion context by recurrent aggregation, allowing effective information flow among adjacent frames and capturing dynamic gist of the persons. The architecture of AMOC is end-to-end trainable and flow among adjacent frames and capturing dynamic gist of the persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapted to complement appearance persons. The architecture of AMOC is end-to-end trainable and thus motion context can be adapte...
this motion is called motion context. Therefore, we also develop the model that can well capture the spatial features and motion context from sequences of persons to improve the performance of re-identification. To achieve this goal, we design a novel two-stream spatial-temporal architecture, called the Accumulative Motion Context networks, which can end-to-end learn discriminative spatial representation and temporal motion context information from raw video frames. Especially for learning the temporal motion context information, inspired by a recent work FlowNet [18] which is able to end-to-end learn optical flow estimation from image pairs with convolutional networks, we design a new motion network in our AMOC architecture to perform the end-to-end motion context information learning task. Our approach is able to achieve better performance compared to other video person re-id methods, as clearly validated by extensive experimental results.

To sum up, we make the following contributions to video-based person re-identification:

- We propose a novel motion context accumulating network model which has two individual stream networks simultaneously learning spatial appearance features and temporal motion context information and accumulating them in a recurrent way.
- Our proposed model can directly learn both spatial features and motion context from raw person video frames in an end-to-end manner, instead of pre-extracting optical flow using the off-line algorithm.
- We quantitatively validate the good performance of our end-to-end two-stream recurrent convolutional network by comparing it with the state-of-the-arts on two benchmark datasets: iLIDS-VID [19] and PRID-2011 [20].

II. RELATED WORK

Person re-identification has been extensively studied in recent years. Existing works on person re-identification can be roughly divided into two types: person re-id for still images and person re-id for video sequences.

A. Person Re-id for Still Images

Previous works on person re-id for still images focus on the invariant feature representation and distance metric learning. Discriminative features that are invariant to environmental and view-point changes play a determining role in person re-id performance. [21] combines spatial and color information using an ensemble of discriminant localized features and classifiers selected by boosting to improve viewpoint invariance. Symmetry and asymmetry perceptual attributes are exploited in [1], based on the idea that features closer to the body axes of symmetry are more robust against scene clutter. [22] fits a body configuration composed of chest, head, thighs, and legs in pedestrian images and extracts per-part color information as well as color displacement within the whole body to handle pose variation. [23] turns local descriptors into the Fisher Vector to produce a global representation of an image. Kvitkovsky et al. [24] propose a novel illumination-invariant feature representation based on the logchromaticity (log) color space and demonstrate that color as a single cue shows relatively good performance in identifying persons under greatly varying imaging conditions.

After feature extraction, distance metric learning is used in person re-id to emphasize inter-person distance and de-emphasize intra-person distance. Large margin nearest neighbor metric (LMNN) [25] is proposed to improve the performance of the traditional kNN classification. Prosser et al. [26] formulate person re-identification as a ranking problem, and use ensembled RankSVMs to learn pairwise similarity. Zheng et al. [27] propose a soft discriminative scheme termed relative distance comparison (RDC) by large and small distances corresponding to wrong matches and right matches, respectively. [28] proposes a high dimensional representation of color and Scale Invariant Local Ternary Pattern (SILTP) histograms. It constructs a histogram of pixel features, and then takes its maximum values within horizontal strips to tackle viewpoint variations while maintaining local discrimination.

B. Person Re-id for Video Sequences

Recently, some works consider performing person re-id in video sequences. In this scenario, the person matching problem in videos is crucial in exploiting multiple frames in videos to boost the performance. [29] applies Dynamic Time Warping (DTW) to solve the sequence matching problem in video-based person re-id. [30] uses a conditional random field (CRF) to ensure similar frames in a video sequence to receive similar labels. [31] proposes to learn to map between the appearances in sequences by taking into account the differences between specific camera pairs. [19] introduces a pictorial video segmentation approach and deploys a fragment selecting and ranking model for person matching. [32] introduces a block sparse model to handle the video-based person re-id problem by the recovery problem in the embedding space. [13] proposes a spatio-temporal appearance representation method, and feature vectors that encode the spatially and temporally aligned appearance of the person in a walking cycle are extracted. [12] proposes a top-push distance learning (TDL) model incorporating a top-push constraint to quantify ambiguous video representation for video-based person re-id.

Deep learning methods have also been applied to video-based person re-id to simultaneously solve feature representation and metric learning problem. Usually DNNs are used to learn ranking functions based on pairs [3] or triplets [33] of images. Such methods typically rely on a deep network (e.g. Siamese network [34]) used for feature mapping from raw images to a feature space where images from the same person are close while images from different persons are widely separated. [10] uses a recurrent neural network to learn the interaction between multiple frames in a video and a Siamese network to learn the discriminative video-level features for person re-id. [9] uses the Long-Short Term Memory (LSTM) network to aggregate frame-wise person features in a recurrent manner. Unlike the existing deep learning based methods for person re-id in videos, our proposed Accumulative Motion Context (AMOC) networks introduces an end-to-end two-stream architecture that has specialized network streams for
learning spatial appearance and temporal feature representations individually. Spatial appearance information from the raw video frame input and temporal motion information from the optical flow predicted by the motion network are processed respectively and then fused at higher recurrent layers to form a discriminative video-level representation.

III. PROPOSED METHOD

We propose an end-to-end Accumulative Motion Context Network (AMOC) based architecture that addresses the video person re-identification problem through joint spatial appearance learning and motion context accumulating from raw video frames. We first introduce the overall architecture of our AMOC model (III-A), which is illustrated in Fig. 1. Then for each pair of frames from a person sequence, the details of motion networks and spatial networks and how they collaborate with each other are described. Besides, the recurrent fusion layers to integrate two-stream spatial-temporal information and fusion method are elaborated. Finally, implementation details of training and test are introduced for reproducing the results.

A. Architecture Overview

Fig. 1 illustrates the architecture of the proposed end-to-end Accumulative Motion Context network (AMOC). In our architecture each two consecutive frames is processed by a two-stream network to learn spatial appearance and temporal motion features representing the person’s appearance at a certain time instant. Then these two-stream features are fused in a recurrent way for learning discriminative accumulative motion contexts. After that, the temporal pooling layer integrates the two-stream features in time from an arbitrarily long video sequence into a single feature representation. Finally, the two-stream sub-networks for two sequences from two different cameras are constructed following the Siamese network architecture [34] in which the parameters of Camera A networks and Camera B networks are shared. To end-to-end train this network, we adopt multi-task loss functions including contrastive loss and classification loss. The contrastive loss decides whether two sequences describe the same person or not while the classification loss predicts the identity of the person in the sequence. In the following, we will give more detailed explanations to each component of our proposed network.

B. End-to-end Two-stream Networks

As aforementioned, each frame in a sequence of the person is processed by two convolutional network streams jointly. Specifically, one stream uses spatial networks (yellow boxes in Fig. 1) to learn spatial features from raw video frames while for the other stream spatial networks (green boxes in Fig. 1) are applied on the motion context features produced by motion networks (blue boxes in Fig. 1) to learn the temporal feature representations at each location between a pair of video frames. In this subsection, we introduce the details of the motion networks and spatial networks, and then describe how they work together.

1) The Motion Networks: As shown in Fig. 1 at each time-step a pair of consecutive video frames of a person is processed by the motion network within AMOC (corresponding to blue boxes in Fig. 1) to predict motion between the adjacent frames. Similar to the structure used in [18], the motion network consists of several convolutional and deconvolutional layers for up-sizing the high-level coarse feature maps learned by convolutional layers. In particular, it has 6 convolutional layers (corresponding to “Conv1”, “Conv1”, “Conv2”, “Conv2”, “Conv2_1”, “Conv3” and “Conv3_1”) with stride of 2 (the simplest form of pooling) in six of them and a tanh non-linearity after each layer. Taking the concatenated two person frames as inputs with size of \( h \times w \times 6 \) (where \( h \) is the height of the frame and \( w \) is the width), the network employs 6 convolutional layers to learn high-level abstract representations of the frame pair by producing the feature maps with reduced sizes (w.r.t. the raw input frames). However, this size shrinking process could result in low resolution and harm the performance of person re-id. So in order to provide dense per-pixel predictions we need to refine the coarse pooled representation. To perform the refinement, we apply the “deconvolution” on feature maps, and concatenate them with corresponding feature maps from the “contractive” part of the network and an upsampled coarser flow prediction (“Pred1”, “Pred2”, “Pred3”). In this way, the networks could preserve both the high-level information passed from coarser feature maps and refine local information provided in lower layer feature maps. Each step increases the resolution by a factor of 2. We repeat this process for twice, leading to the final predicted flow (“Pred3”) whose resolution is still as half as that of raw input. Note that our motion networks do not have any fully connected layers, which can take video frames of arbitrary size as input. This motion network can be end-to-end trained by using optical flow generated by several off-the-shelf algorithms as supervision, such as Lucas-Kanade [35] and EpicFlow [36] algorithm. The training details will be described in Sec. III-D1.

Fig. 3. The structure of spatial networks of our proposed accumulative motion context network at one time-step. It consists of 3 convolutional layers and 3 max-pooling layers with a tanh non-linearity layer interpolated after each convolutional layer. And there is a fully-connected layer at the top of last max-pooling layer. The purple cubes are convolutional kernels while the red ones are pooling kernels.

2) The Spatial Networks: As shown in Fig. 1 there are two spatial networks (yellow and green boxes) in both streams to learn spatial feature representations from raw video frames and temporal features at the spatial location between two consecutive frames. Here, both spatial networks lying in the two streams have the same structure, each of which contains
Fig. 1. The architecture of our proposed Accumulative Motion Context Network (AMOC). At each time-step, each pair of two consecutive frames is processed by a two-stream network, including spatial network (yellow and green boxes) and motion network (blue box), to learn spatial appearance and temporal motion feature representations. Then these two-stream features are fused in a recurrent way for learning discriminative accumulative motion contexts. The two-stream features are integrated by temporal pooling layer from an arbitrarily long video sequence into a single feature representation. Finally, the two-stream sub-networks for two sequences from two different cameras are constructed following the Siamese network architecture [34] in which the parameters of Camera A networks and Camera B networks are shared. The whole AMOC network is end-to-end trained by introducing multi-task losses (classification loss and contrastive loss) to satisfy both the contrastive objective and to predict the persons identity.

3 convolutional layers and 3 max-pooling layers with a non-linearity layer (in this paper we use \( \text{tanh} \)) interpolated after each convolutional layer. And there is a fully-connected layer at the top of last max-pooling layer. The details of spatial networks are shown in Fig. 3, where the purple cubes are convolutional kernels and the red ones are pooling kernels. All the strides in the convolutional layers and pooling layers are set as 2. Note that although two spatial networks have the same structure, they play different roles in two streams. The inputs of the network (yellow boxes in Fig. 1) are raw RGB images when serves as a spatial feature extractor. Otherwise, the inputs are the last predictions (“Pred3”) of motion networks (blue boxes in Fig. 1).

C. Spatial Fusion and Motion Context Accumulation

1) Spatial fusion: Here we consider different fusion methods (orange boxes in Fig. 1) to fuse the two stream networks. Our intention is to fuse the spatial features and motion context information at the spatial location such that channel responses at the same pixel position are put in correspondence. Because the structures of spatial networks in two streams are the same, the feature map produced by each layer in each stream has the exact location correlation. So the problem is how to fuse the outputs of corresponding layers of two streams. To motivate our fusion strategy, consider for example discriminating two walking persons. If legs move or arms swing periodically at some spatial location then the motion network can recognize that motion and obtain the motion context information from two consecutive frames, and the spatial network can recognize the location (legs or hands) and their combination so as to discriminate the persons.

This spatial correspondence can be easily obtained when the two networks have the same spatial resolution at the layers to be fused. The simplest way is overlaying (stacking) layers from one network on the other. However, there is also the issue of establishing the correct correspondence between one channel (or channels) in one network and the corresponding channel (or channels) of the other network. Suppose different channels in the spatial network learning spatial feature representation from one video frame are responsible for different body areas (head, arms, legs, etc.), and one channel in the spatial network following the motion network is responsible for contextual motion between two neighboring frames in the fields. Then, after the channels are stacked, the filters in the subsequent layers must learn the correspondence between these appropriate channels in order to best discriminate between these motions from different person samples. To make this more concrete, we investigate the following 3 ways of fusing layers between two stream networks. Suppose \( x^A \in \mathbb{R}^{H \times W \times D} \) and \( x^B \in \mathbb{R}^{H \times W \times D} \) are two feature maps from two layers and need to be fused, where \( W, H \) and \( D \) are the width, height and channel number of the respective feature maps. And \( y \) represents the fused feature maps. When applied to the feedforward spatial network architecture which is shown in Fig. 3 which consists of convolutional, max-pooling non-linearity and fully-connected
layers, the fusion can be performed at different points in the network to implement e.g. early-fusion or late-fusion.

- **Concatenation fusion** This fusion operation stacks the two feature maps at the same spatial locations $i, j$ across the feature channels $d$:

  \[
  y_{i,j,d}^{\text{cat}} = x_{i,j,d}^A, \quad y_{i,j,d}^{\text{cat}} = x_{i,j,d}^B, \quad 1 \leq i \leq H, 1 \leq j \leq W.
  \]  

- **Sum fusion** The sum fusion computes the sum of the two feature maps at the same spatial locations $i, j$ and channels $d$:

  \[
  y_{i,j,d}^{\text{sum}} = x_{i,j,d}^A + x_{i,j,d}^B,
  \]  

  where $x^A, x^B \in \mathbb{R}^{H \times W \times D}, y^{\text{cat}} \in \mathbb{R}^{H \times W \times 2D}$.

- **Max fusion** Similarly, max fusion takes the maximum of the two feature maps:

  \[
  y_{i,j,d}^{\text{max}} = \max \{ x_{i,j,d}^A, x_{i,j,d}^B \}.
  \]

Now, we briefly introduce where our fusion method should be applied to. Injecting fusion layers can have significant impact on the number of parameters and layers in a two-stream network, especially if only the network which is fused into is kept and the other network tower is truncated. For example, in Fig. 3 if the fusion operation is performed on the second max-pooling layers (“Max-pool2”) of two spatial networks in the two-stream network, then the previous parts (“Conv1, Max-pool1, Conv2” and non-linearity layers) are kept, and the layers (“Conv3, Max-pool3, Fully connected”) after fusion operation share one set of parameters. The illustration is shown in Fig. 4.

In the experimental section (Sec. IV-C), we evaluate and compare the performance of each of the three fusion methods in terms of their re-identification accuracy.

2) **Motion Context Accumulation**: We now consider the techniques to combine fused features $f(t)$ (output of the fully-connected layer in Fig. 4) containing both spatial appearance and motion context information over time $t$. Because the length of a sequence is arbitrary, the motion context is also unfixed for each person. Therefore, we exploit Recurrent neural networks (RNN) which can process an arbitrarily long time-series using a neural network to address the problem of accumulating motion context information. Specifically, an RNN has feedback connections allowing it to remember information over time and produces an output based on both the current input and information from the previous time-steps at each time-step. As shown in Fig. 4, the recurrent connections of RNN are “unrolled” in time to create a very deep feed-forward network. Given the unrolled network, the lateral connections serve as “memory”, allowing information to flow between an arbitrary number of time-steps.

As video-based person re-identification involves recognizing a person from a sequence, accumulating the motion context information of each frame at each instant would be helpful to improve the performance of re-identification. This motion accumulation can be achieved by using the recurrent connections allowing information to be passed between time-steps. To be
more clear, we aim to better capture person spatial appearance features and motion context information present in the video sequence, and then to accumulate them along the time axis. Specifically, given the p-dimension output \( \mathbf{f}^{(t)} \in \mathbb{R}^{p \times 1} \) of the fused spatial networks, the RNN can be defined as follows:

\[
\mathbf{o}^{(t)} = M \mathbf{f}^{(t)} + N \mathbf{r}^{(t-1)}, \\
\mathbf{r}^{(t)} = \tanh(\mathbf{o}^{(t)}),
\]

(4) (5)

Here \( \mathbf{o}^{(t)} \in \mathbb{R}^{q \times 1} \) is the q-dimensional output of RNN at time-step \( t \), and \( \mathbf{r}^{(t-1)} \in \mathbb{R}^{q \times 1} \) contains the information on the RNN’s state at the previous time-step. The \( M \in \mathbb{R}^{q \times p} \) and \( N \in \mathbb{R}^{q \times q} \) represent the corresponding parameters for \( \mathbf{f}^{(t)} \) and \( \mathbf{r}^{(t-1)} \) respectively, where \( q \) is the dimension of the output of the last fully-connected layer in fusion part and \( p \) is the dimension of the feature embedding-space.

Although RNNs are able to accumulate the fused information, they still have some limitations. Specifically, some time-steps may be more dominant in the output of the RNN, which could reduce the RNNs effectiveness when used to accumulate the input information over a full sequence because discriminative frames may appear anywhere in the sequence. To overcome the drawback, similar to [10], we add a temporal pooling layer after RNN to allow for the aggregation of information across all time steps. The temporal pooling layer aims to capture long-term information present in the whole sequence, which combines with the motion context information accumulated through RNN. In this paper, we adopt mean-pooling over the temporal dimension to produce a single feature vector \( \mathbf{u} \) representing the person’s spatial appearance and motion information averaged over the whole input sequence. The pooling method is as follows:

\[
\mathbf{u} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{o}^{(t)},
\]

(6)

where \( T \) is the length of the sequence or time-steps.

3) Multi-task Loss: Similar to the method suggested in [10], we train the whole AMOC network to satisfy both the contrastive objective and to predict the persons identity. Given the sequence feature vector \( \mathbf{u} \) including accumulative spatial appearance feature and motion context information, output by the feature representation learning networks of our AMOC, we can predict the identity of the person in the sequence using the standard softmax function, which is defined as follows:

\[
I(\mathbf{u}) = P(z = c | \mathbf{u}) = \frac{\exp(S_c \mathbf{u})}{\sum_k \exp(S_k \mathbf{u})},
\]

(7)

where there are a total of \( K \) identities, \( z \) is the identity of the person, and \( S \) is the softmax weight matrix while \( S_c \) and \( S_k \) represent the \( c \)th and \( k \)th column of it, respectively. Then the corresponding softmax loss function (pink boxes in Fig. 1) is defined as follows:

\[
L_{\text{class}} = -\log(P(z = c | \mathbf{u})).
\]

(8)

Besides, given a pair of sequence feature vectors \( (\mathbf{u}_{(a)}, \mathbf{u}_{(b)}) \) output by the Siamese network, the contrastive loss (purple box in Fig. 1) function can be defined as

\[
L_{\text{con}} = \left\| \mathbf{u}_{(a)} - \mathbf{u}^+_{(b)} \right\|^2_2 + \max \left\{ 0, \alpha - \left\| \mathbf{u}_{(a)} - \mathbf{u}^-_{(b)} \right\|^2_2 \right\},
\]

(9)

where \( \mathbf{u}^+_{(b)} \) represents the positive pair of \( \mathbf{u}_{(a)} \) while \( \mathbf{u}^-_{(b)} \) represents the negative pair of \( \mathbf{u}_{(a)} \). The loss consists of two penalties: the first term penalizes a positive pair \( (\mathbf{u}_{(a)}, \mathbf{u}^+_{(b)}) \) that is too far apart, and the second penalizes a negative pair \( (\mathbf{u}_{(a)}, \mathbf{u}^-_{(b)}) \) that is closer than a margin \( \alpha \). If a negative pair is already separated by \( \alpha \), then there is no penalty for this pair and \( L_{\text{con}}(\mathbf{u}_{(a)}, \mathbf{u}^-_{(b)}) = 0 \).

Finally, we jointly end-to-end train our architecture with both classification loss and contrastive loss. We can now define the overall multi-task training loss function \( L_{\text{multi}} \) for a single pair of person sequences, which jointly optimizes the classification cost and the contrastive cost as follows:

\[
L_{\text{multi}}(\mathbf{u}_{(a)}, \mathbf{u}_{(b)}) = L_{\text{con}}(\mathbf{u}_{(a)}, \mathbf{u}_{(b)}) + L_{\text{class}}(\mathbf{u}_{(a)}) + L_{\text{class}}(\mathbf{u}_{(b)}).
\]

(10)

Here, we give equal weights for the classification cost and contrastive cost terms. The above network can be trained end-to-end using back-propagation-through-time from raw video frames (details of our training parameters can be found in Sec. III-D). During the training phase, all recurrent connections are unrolled to create a deep feed-forward graph, where the weights of AMOC are shared between all time-steps. And in the test phase, we discard the multi-task loss functions and apply the AMOC on the raw video sequences as a feature extractor, where the feature vectors extracted by it can be directly compared using Euclidean distance.

D. Implementation Details

1) Pre-training of the Motion Networks: As aforementioned in the Sec. III-B2, we use the pre-extracted optical flow as the ground truth to pre-train the motion networks. Our aim is that our proposed architecture can directly learn motion context information from raw consecutive video frames. When the pre-training of motion networks is finished, we use the trained parameters to initialize the motion networks in the whole framework.

As shown in the Fig. 2 our motion networks produce three motion maps of optical flow maps at three levels of scale (“Pred1”, “Pred2”, “Pred3”) for a pair of person frames. Because the size of each prediction map is 1/8, 1/4 and 1/2 of the input frame size respectively, the pre-extracted optical flow maps are all downsampled to the corresponding sizes, then serve as the ground truth. In this paper, we use the smooth-L1 loss function [37] to compute the losses between the predictions \( \mathbf{e}^{(l)} \in \mathbb{R}^{h(l) \times w(l) \times 2} \) and ground truth optical flow maps \( \mathbf{g}^{(l)} \in \mathbb{R}^{h(l) \times w(l) \times 2} \), where \( l = 1, 2, 3 \). The loss function is defined as

\[
L_{\text{motion}}^{(l)}(\mathbf{e}^{(l)}, \mathbf{g}^{(l)}) = \sum_{i,j,k} \text{smooth}_L(e_{i,j,k}^{(l)} - g_{i,j,k}^{(l)}),
\]

(11)

\[
\text{smooth}_L(\theta) = \begin{cases} 
0.5 \theta^2 & \text{if } |\theta| < 1 \\
|\theta| - 0.5 & \text{otherwise}.
\end{cases}
\]

(12)
Then, the overall cost function can be written as

\[
L_{\text{motion\_all}} = \sum_{l=1}^{3} \omega_l L_{\text{motion}}^{(l)},
\]

where \(\omega_l\) represents the weight of each loss between different scale level prediction and ground truth map. In the pre-training phase of the motion network, we set them to \((0.01, 0.02, 0.08)\) respectively.

All the input pairs of video frames are resized to \(128 \times 64\) and Adam \([38]\) is chosen as the optimization method due to its faster convergence than standard stochastic gradient descent with momentum for our task. As recommended in \([38]\), we fix the parameters of Adam: \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\). Considering every pixel is a training sample, we use small mini-batches of 4 frame pairs. We start with learning rate \(\lambda = 1e^{-4}\) and then divide it by 2 every 10k iterations after the first 20k.

2) Training of the Overall Architecture: After pre-training the motion network, we use it to initialize the one in our AMOC to end-to-end train the whole network. In the end-to-end training process, we set margin \(\alpha\) to 2 in the Eqn. (9), and the embedding-space dimension is set to 128. Besides, the learning rate is set to \(1e^{-3}\). Note, as we mentioned in Sec. III-B1, the resolution of output final predicted flow map of the motion network (“Pred3”) is as half as that of input. Therefore, at each time-step, we resize the first frame of a pair to two scales of \(64 \times 32\) and \(128 \times 64\). For the second frame, we resize it to \(128 \times 64\). That is to say, at each time-step, the first-stream spatial network (yellow boxes in Fig.1) takes the first frame within a pair with size of \(64 \times 64\) as input to extract spatial appearance feature representations. And the second-stream networks, including spatial network and motion network (green and blue boxes in Fig.1) take the pair frames with sizes of \(128 \times 64\) as input to extract motion context information. This operation can be performed in an on-line way. Furthermore, it guarantees that the feature map generated by each layer of two spatial networks in two streams has the same resolution thus can be fused at an arbitrary layer.

3) Data Augmentation: To increase diversity of the training sequences, data augmentation including cropping and mirroring is applied. For a given sequence, the same augmentation is applied to all frames. During the testing phase, data augmentation is also applied, and the similarity scores between sequences are averaged over all the augmentation conditions.

IV. EXPERIMENTS

A. Datasets

In this paper, we use iLIDS-VID \([19]\) and PRID-2011 \([20]\), which are two public benchmarks available, to evaluate our proposed AMOC model.

iLIDS-VID: The iLIDS-VID dataset is created from the pedestrians captured in two non-overlapping camera views at an airport arrival hall under a multi-camera CCTV network. It is very challenging due to large clothing similarities among people, lighting and viewpoint variations across camera views, cluttered background and random occlusions. There are 600 image sequences of 300 distinct individuals in the dataset, with one pair of image sequences from two camera views for each person. The image sequences with an average number of 73 range in length from 23 to 192 image frames.

PRID-2011: The PRID-2011 dataset totally contains 749 persons captured by two non-overlapping cameras. Among, there are 400 image sequences for 200 people from two camera views that are adjacent to each other, with sequences lengths of 5 to 675 frames. Compared with the iLIDS-VID dataset, it is less challenging due to being captured in non-crowded outdoor scenes with relatively simple and clean backgrounds and rare occlusions. Similar to the protocol used in \([10]\), we only use the first 200 persons appearing in both cameras for evaluation.

B. Experimental Settings and Evaluation Protocol

For all the experiments performed on both datasets, half of persons are extracted for training and the other half for testing. All experiments are conducted 10 times with different training/testing splits and the averaged results ensure their stability. Additionally, the training of networks in our architecture, including pre-training of motion networks and end-to-end training of the whole framework, are all implemented by using the Torch \([39]\) framework on NVIDIA GeForce GTX TITAN X GPU.

As we described in Sec. III-A our network is a Siamese-like network, so the positive and negative sequence pairs are randomly on-line selected during the training phase. However, positive and negative sequence pairs consist of two full sequences of an arbitrary length containing the same person or different persons under different cameras respectively. To guarantee the fairness of experiments, we follow the same sequence length setting in \([10]\). Considering the efficiency of training, a sub-sequence containing 16 consecutive frames is sampled from the full length sequence of a person during training. At each epoch, this random selection is performed once. During testing, the sequences under the first camera and second camera are regarded as the probe and the gallery respectively, as in \([10]\) and \([19]\). And the length of each person sequence is set to 128 for testing. If the length of a full sequence of a person is smaller than 128, we use the full-length sequence of this person to test.

In experiments, for each pedestrian, the matching of his or her probe sequence (captured by one camera) with the gallery sequence (captured by another camera) is ranked. To reflect the statistics of the ranks of true matches, the Cumulative Match Characteristic (CMC) curve is adopted as the evaluation metric. Specifically, in the testing phase, the Euclidean distances between probe sequence features and those of gallery sequences are computed firstly. Then, for each probe person, a rank order of all the persons in the gallery is sorted from the one with the smallest distance to the biggest distance. Finally, the percentage of true matches founded among the first \(m\) ranked persons is computed and denoted as \(\text{rank}(m)\).

C. Analysis of the Proposed AMOC Model

Before showing the comparison of our method with the state-of-the-arts, we conduct several analytic experiments on...
iLIDS-VID and PRID-2011 datasets to verify the effectiveness of our model for solving the video-based person re-identification problem. We analyze and investigate the effect of several factors upon the performance, which includes the generation of motion context information, the selection of spatial fusion method, and the location of performing spatial fusion. In this paper, we regard the method in [10] as our baseline, in which the spatial-temporal features are also employed in a recurrent way but without two-stream structure or spatial fusion method, and all the optical flow maps serving as motion information are pre-extracted in off-line way.

1) Effect of Different Motion Information: As described in Sec. III-B1, our motion network is able to end-to-end learn motion information from video sequence frames. Thus except for the Lucas-Kanade optical flow (LK-Flow) algorithm [35] used in [10], we also exploit the optical flow maps produced by EpicFlow [36] algorithm to investigate the effect of different motion information. The experimental results are shown in Tab. I. The AMOC networks in “AMOC + LK-Flow” and “AMOC + EpicFlow” methods are the non-end-to-end versions, the motion networks of which are replaced by the corresponding optical flow maps pre-extracted. Note that here we only study the effect of different optical flows carrying motion context information and verify the effectiveness of end-to-end learning. Therefore, all the shown results of our AMOC are achieved by using the “concatenation fusion” method introduced in the Sec. III-C1 and fusing two-stream networks at “Max-pool2” layer as illustrated in Fig. 4. In Tab. I “Baseline + LK-Flow” is the method from [10], and we observe that the performance is boosted when the LK-Flow is replaced by the EpicFlow to produce optical flow in both baseline methods or our AMOC method. As introduced in [36], EpicFlow assumes that contours often coincide with motion discontinuities and computes a dense correspondence field by performing a sparse-to-dense interpolation from an initial sparse set of matches, leveraging contour cues using an edge-aware distance. So it is more robust to motion boundaries, occlusions and large displacements than the LK-Flow algorithm [35], which is beneficial to the extraction of motion information between video frames. In addition, we can see that the performance of our “end-to-end AMOC” using different optical flow generation methods is both improved.

Moreover, we also visualize the optical flow maps produced by our motion networks in Fig. 5. The first and forth rows are the consecutive raw video frames from iLIDS-VID and PRID-2011 datasets while the optical flow maps computed using [36] for the two datasets are shown in the second and fifth rows. The third row and sixth row are the output flow maps of our motion networks. All the produced optical flow maps are encoded with the flow color coding [40] method which is also used in [36]. Different colors represent different directions of motions, and shapes indicate the speeds of motions. The faster the motion is, the darker its color will be. In this experiment, we only use the sequences of half persons of iLIDS-VID for training, and perform the motion networks on the left half persons’ sequences of iLIDS-VID and the whole PRID-2011 dataset. In other words, for the PRID-2011 dataset, there is no need to re-train the motion networks on it. From the results shown in Fig. 5, we can see that our motion networks can well approximate the optical flow maps produced by EpicFlow [36] and successfully capture the motion details of persons, such as the speed and amplitude of legs moving. Especially for the PRID-2011 dataset, our motion network can achieve the good optical flow estimation without using training data from PRID-2011, which means our motion network has a good generalization ability.

2) Effect of Spatial Fusion Method and Location: In this part, we investigate how and where to fuse two-stream networks in our end-to-end AMOC network. For these experiments, we use the same spatial network architecture introduced in Sec. III-B2. The fusion layer can be injected at any location, such as after “Max-pool 2”, i.e. its input is the output of “Max-pool 2” from the two streams. After the fusion layer a single processing stream is used.

We compare different fusion strategies in Tab. II where we report the rank1, rank5, rank10 and rank 20 recognition rate on both iLIDS-VID and PRID-2011 datasets. From the results, we see that “Concatenation” fusion method performs considerably higher than “Sum” and “Max” fusion methods. Compared to the fusion methods, our end-to-end AMOC network shows more sensitiveness to the location of spatial fusion. Specifically, for all three fusion methods, our method can achieve the best performance when the spatial fusion is performed after the “Max-pool2” layer. And fusion at FC (Fully-Connected) layers results in an evident drop in the performance. The reason for FC performing worse may be that at this layer spatial correspondences between spatial appearance and motion context information would be collapsed.

| Dataset       | iLIDS-VID       | PRID-2011       |
|---------------|----------------|-----------------|
| Methods       | Rank1 | Rank5 | Rank10 | Rank20 | Rank1 | Rank5 | Rank10 | Rank20 |
| Baseline + LK-Flow [10] | 58.0  | 84.0  | 91.0   | 96.0   | 70.0  | 90.0  | 95.0   | 97.0   |
| Baseline + EpicFlow [36] | 59.3  | 87.2  | 92.7   | 98.2   | 76.2  | 97.5  | 98.2   | 99.0   |
| AMOC + LK-Flow | 63.3  | 85.3  | 95.1   | 96.4   | 76.0  | 96.5  | 97.4   | 99.6   |
| AMOC + EpicFlow | 65.5  | 93.1  | 97.2   | 98.7   | 82.0  | 97.3  | 99.3   | 99.4   |
| end-to-end AMOC + LK-Flow | 65.3  | 87.3  | 96.1   | 98.4   | 78.0  | 97.2  | 99.1   | 99.7   |
| end-to-end AMOC + EpicFlow | 68.7  | 94.3  | 98.3   | 99.3   | 83.7  | 98.3  | 99.4   | 100.0  |
D. Comparison with State-of-the-Art Methods

We further evaluate the performance of end-to-end AMOC, by comparing it with the state-of-the-art matching methods for video-based person re-identification, including STA [13], DVR [11], ST2DL [14], PaMM [41], mvRMLLC+ST-Alignment [42], TAPR [15], SRID [32], AFDA [43], DVDL [44], and RFA-Net [9]. Comparing the CMC results shown in Tab. III, we can see that the non-end-to-end version of our AMOC can achieve higher performance than all the compared methods for both datasets. When the end-to-end AMOC is applied, the performance is further boosted, especially for the Rank1 protocol. The improvements are 3.2% and 1.7% for iLIDS-VID and PRID-2011 datasets respectively. Moreover, to our best knowledge, we are the first to introduce a two-stream network structure and end-to-end learning motion information from raw frame pairs to solve the video-based person re-identification problem. Compared to those methods also using spatial-temporal features, such as STA [13], RFA-Net [9] and TAPR [15], our AMOC can achieve performance improvement by a large margin for both datasets, due to the two-stream structure of our AMOC which separately deals with the spatial appearance and motion information from context and then performs feature integration through spatial fusion. From the results, we notice that the second best method, “mvRMLLC+ST-Alignment [42]”, can also achieve good performance on iLIDS-VID. However, this method needs the complex pre-processing of person video frames while our end-to-end AMOC can directly learn representations from raw video frames. Note, in Tab. III, “end-to-end AMOC + EpicFlow” means our end-to-end AMOC uses the motion networks pre-trained by using EpicFlow optical flow maps as the supervision.

V. CONCLUSION

In this work, we propose an end-to-end Accumulative Motion Context Network (AMOC) based method addressing video person re-identification problem through joint spatial appearance learning and motion context accumulating from raw video frames. We conducted extensive experiments on two public available video-based person re-identification datasets to validate our method. Experimental results demonstrated that our model outperforms other state-of-the-art methods in most cases, and verified that our accumulative motion context model is beneficial for the recognition accuracy in person matching.

REFERENCES

[1] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person re-identification by symmetry-driven accumulation of local features,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2360–2367.
[2] W. Li, R. Zhao, T. Xiao, and X. Wang, “Deepreid: Deep filter pairing neural network for person re-identification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 152–159.
### TABLE II

**Rank1, Rank5, Rank10 and Rank20 Recognition Rate (in %) of Various Fusion Methods on iLIDS-VID and PRID-2011 Datasets.**

| Fusion Method | Dataset | iLIDS-VID | PRID-2011 |
|---------------|----------|-----------|-----------|
|               |          | Rank1 | Rank5 | Rank10 | Rank20 | Rank1 | Rank5 | Rank10 | Rank20 |
| Sum           |          | Tanh1  | 60.8  | 90.1  | 91.7   | 96.5  | 71.5  | 93.4   | 97.3   | 98.1   |
|               |          | Max-pool1 | 61.2  | 89.8  | 90.3   | 95.1  | 72.6  | 92.8   | 96.2   | 96.9   |
|               |          | Tanh2   | 65.5  | 91.8  | 95.6   | 96.5  | 79.9  | 95.8   | 97.6   | 97.8   |
|               |          | Max-pool2 | 67.8  | 93.4  | 96.5   | 98.3  | 80.6  | 96.6   | 97.8   | 99.2   |
|               |          | Tanh3   | 63.9  | 91.8  | 96.7   | 97.7  | 78.3  | 94.5   | 97.8   | 98.2   |
|               |          | Max-pool3 | 65.0  | 92.8  | 97.9   | 98.4  | 78.8  | 94.9   | 98.0   | 99.1   |
|               |          | FC      | 60.7  | 88.1  | 93.2   | 94.3  | 72.0  | 91.2   | 93.8   | 94.9   |
| Max           |          | Tanh1  | 60.6  | 90.0  | 91.9   | 97.2  | 73.1  | 94.9   | 97.2   | 99.5   |
|               |          | Max-pool1 | 61.0  | 91.2  | 93.1   | 97.8  | 75.1  | 94.9   | 99.0   | 99.5   |
|               |          | Tanh2   | 66.9  | 94.2  | 97.1   | 98.9  | 81.2  | 97.3   | 98.5   | 99.3   |
|               |          | Max-pool2 | 68.2  | 95.4  | 97.5   | 98.9  | 81.6  | 98.6   | 98.8   | 99.3   |
|               |          | Tanh3   | 64.0  | 91.5  | 96.4   | 97.3  | 78.1  | 94.1   | 97.3   | 98.2   |
|               |          | Max-pool3 | 64.3  | 93.1  | 98.2   | 98.3  | 78.4  | 94.5   | 97.8   | 99.0   |
|               |          | FC      | 61.2  | 89.1  | 93.6   | 94.9  | 73.5  | 90.2   | 92.9   | 94.7   |
| Concatenation |          | Max-pool2 | 68.7  | 94.3  | 98.3   | 99.3  | 83.7  | 98.3   | 99.4   | 100    |
|               |          | Tanh3   | 65.2  | 92.3  | 97.1   | 98.4  | 80.0  | 96.3   | 99.4   | 99.6   |
|               |          | Max-pool3 | 66.1  | 92.8  | 97.9   | 98.4  | 80.0  | 96.9   | 99.8   | 99.8   |
|               |          | FC      | 62.3  | 88.3  | 93.9   | 95.6  | 73.2  | 91.3   | 94.4   | 96.7   |

### TABLE III

**Comparison of Our End-to-End AMOC’s Performance on iLIDS-VID and PRID-2011 Datasets to the State-of-the-Arts.**

| Dataset | iLIDS-VID | PRID-2011 |
|---------|-----------|-----------|
| Methods | Rank1 | Rank5 | Rank10 | Rank20 | Rank1 | Rank5 | Rank10 | Rank20 |
| Baseline [10] | 58.0 | 84.0 | 91.0 | 96.0 | 70.0 | 90.0 | 95.0 | 97.0 |
| STA [13] | 44.3 | 71.7 | 83.7 | 91.7 | 64.1 | 87.3 | 89.9 | 92.0 |
| DVR [11] | 39.5 | 61.1 | 71.7 | 81.8 | 40.0 | 71.7 | 84.5 | 92.2 |
| TDL [12] | 56.3 | 87.6 | 95.6 | 98.3 | 56.7 | 80.0 | 87.6 | 93.6 |
| SI²DL [14] | 48.7 | 81.1 | 89.2 | 97.3 | 76.7 | 95.6 | 96.7 | 98.9 |
| PaMM [41] | 30.3 | 56.3 | 70.3 | 82.7 | 56.5 | 85.7 | 96.3 | 97.0 |
| mvRMLLC+ST-Alignment [42] | **69.1** | 89.9 | 96.4 | 98.5 | 66.8 | 91.3 | 96.2 | 98.8 |
| TAPR [15] | 55.0 | 87.5 | 93.8 | 97.2 | 73.9 | 94.6 | 94.7 | 98.9 |
| SRID [32] | 24.9 | 44.5 | 55.6 | 66.2 | 35.1 | 59.4 | 69.8 | 79.7 |
| AFDA [43] | 37.5 | 62.7 | 73.0 | 81.8 | 43.0 | 72.7 | 84.6 | 91.9 |
| DVDL [44] | 25.9 | 48.2 | 57.3 | 68.9 | 40.6 | 69.7 | 77.8 | 85.6 |
| RFA-Net [9] | 49.3 | 76.8 | 85.3 | 90.0 | 58.2 | 85.8 | 93.4 | 97.9 |
| AMOC + EpicFlow | 65.5 | 93.1 | 97.2 | 98.7 | 82.0 | 97.3 | 99.3 | 99.4 |
| end-to-end AMOC + EpicFlow | **68.7** | **94.3** | **98.3** | **99.3** | **83.7** | **98.3** | **99.4** | **100** |
