Towards Efficient Object Re-Identification with a Novel Cloud-Edge Collaborative Framework

Chuanming Wang*, Yuxin Yang*, Mengshi Qi, Huanhuan Zhang, Huadong Ma†

The State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications

Abstract

Object re-identification (ReID) is committed to searching for objects of the same identity across cameras, and its real-world deployment is gradually increasing. Current ReID methods assume that the deployed system follows the centralized processing paradigm, i.e., all computations are conducted in the cloud server and edge devices are only used to capture images. As the number of videos experiences a rapid escalation, this paradigm has become impractical due to the finite computational resources in the cloud server. Therefore, the ReID system should be converted to fit in the cloud-edge collaborative processing paradigm, which is crucial to boost its scalability and practicality. However, current works lack relevant research on this important specific issue, making it difficult to adapt them into a cloud-edge framework effectively. In this paper, we propose a cloud-edge collaborative inference framework for ReID systems, aiming to expedite the return of the desired image captured by the camera to the cloud server by learning the spatial-temporal correlations among objects. In the system, a Distribution-aware Correlation Modeling network (DaCM) is particularly proposed to embed the spatial-temporal correlations of the camera network implicitly into a graph structure, and it can be applied 1) in the cloud to regulate the size of the upload window and 2) on the edge device to adjust the sequence of images, respectively. Notably, the proposed DaCM can be seamlessly combined with traditional ReID methods, enabling their application within our proposed edge-cloud collaborative framework. Extensive experiments demonstrate that our method obviously reduces transmission overhead and significantly improves performance.

1 Introduction

Object re-identification (ReID) (He et al. 2021, 2023; Li, Sun, and Li 2023; Luo et al. 2019; Huynh 2021; Ge et al. 2024; Chen et al. 2021; Fan et al. 2018; Fu et al. 2024; Qi et al. 2021; Qi, Wang, and Li 2017; Liu et al. 2018) aims to retrieve specific objects captured by non-overlapping cameras, which usually serves as a fundamental task in the field of multimedia processing. It can facilitate users in searching objects accurately across diverse scenes and views, significantly alleviating manual overhead in visual surveillance.

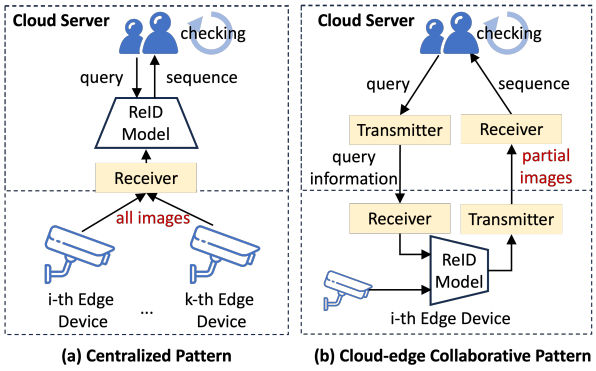

Figure 1: Illustration of the difference between centralized and cloud-edge collaborative patterns for ReID systems.

With the increasing demands, ReID systems have been widely deployed in various real-world scenarios for vehicle or person searching, playing an important role in traffic monitoring, safety management, etc. Therefore, an expanding number of innovative technologies have been introduced to promote the accuracy of ReID system continuously, including establishing elaborate feature extractors (Luo et al. 2019; Huynh 2021; He et al. 2021; Li, Sun, and Li 2023), developing data transmission schemes (Jain et al. 2020), and designing inference strategies (Zhong et al. 2017).

Due to its intrinsic cross-scene nature, a ReID system typically consists of a central cloud server, multiple edge devices (such as cameras), and a communication network for transmitting images and associated data. Previous ReID methods typically follow a centralized processing pattern, where, as illustrated on the left of Fig. 1, edge devices merely capture images and upload **all images** to the cloud server via the connected network. The cloud server then utilizes a deep neural network to extract features and compute similarities between the query and returned images. However, with the rapid proliferation of cameras, this processing pattern imposes excessive strain on the communication network's bandwidth and the cloud server's computing and storage capacities, leading to significant service delays and a compromised user experience. Consequently, in line with current technological trends and driven by the advancement

*These authors contributed equally.
†Corresponding author: Huadong Ma.

of device computing power (Angel et al. 2022), as depicted on the right of Fig. 1, the ReID system should seamlessly integrate into a cloud-based collaborative framework. Feature extraction should occur locally at the edge device, with **partial images** being uploaded to the cloud server based on the ReID model's outputs, thereby alleviating the burden on network communication and cloud computing.

Some previous methods (Zhuang et al. 2020; Zhuang, Wen, and Zhang 2021; Jiang et al. 2023) want to establish new-style cloud-edge collaborative frameworks, but they primarily concentrate on the training phase. For instance, FedReID (Zhuang et al. 2020) and FedUReID (Zhuang, Wen, and Zhang 2021) embed the federated learning into ReID system, delving into strategies to exploit distributed data to continuously optimize the feature extractor, thereby enhancing search accuracy. Besides, some works (Jiang et al. 2023) propose to deploy a segment of the deep neural network to the edge devices, mitigating the computing cost of the cloud server, although it still necessitates a substantial amount of data transmission. We argue that the inference phase also holds greater significance for a practical ReID system, and a meticulous scheme should be developed to fully leverage the advantages of both the cloud server and edge devices. Therefore, in contrast to previous methods, we introduce a pioneering cloud-edge collaborative ReID system that places a heightened emphasis on optimizing the efficiency and effectiveness of the inference process, a domain that has been under-explored in existing research.

For a basic cloud-edge collaborative inference pipeline of the ReID system, when the user requests to search one certain object, one query image and its auxiliary information (denoted as *query* in Fig. 1) are initially dispatched to each edge device from the cloud server by a Transmitter. Then, the edge device extracts its feature via a local visual backbone and compares this feature with all local gallery images, and the resulting similarity is used to create the uploading sequence (denoted as *sequence* in Fig. 1). Due to transmission limitations, there is an upper bound to the number of data the cloud server can accept at one time, so the uploading sequence of images is uploaded in batches. The user checks the sequence and terminates this process when its desired image is returned. Therefore, to achieve an efficient and effective inference, the user's desired image should be returned to the cloud server swiftly. Therefore, two key points in our framework are: (i) the edge device with the desired image should have a higher chance of uploading the image to the cloud server, and (ii) the desired image should be positioned at the beginning of the upload sequence.

To handle the points above, we specifically introduce a distribution-aware correlation modeling network (DaCM), which is deployed in both the cloud server to adjust the bandwidths of edge devices and each edge device to re-rank the image indexes in the uploading sequence. The input of DaCM is spatial-temporal data, *i.e.* the timestamps and camera ID of images, which can be effortlessly obtained from the ReID system. Initially, it embeds spatial-temporal correlations into a graph structure by learning such a problem: what is the likelihood that an object will appear again in camera j after a time delay t, from where it was previously observed in camera i. After training, the topology of the camera network and the movement rules of the object in the current scene will be embedded into DaCM implicitly, so as to support the adjustment of the bandwidth allocated to the edge devices and the index of the image.

Furthermore, since we focus on a new ReID inference pattern, traditional evaluation protocols do not fully showcase the capabilities of proposed method. Thus, we propose several new evaluation protocols and their details will be described later. Finally, extensive experimental results demonstrate that our method improve the performance with a significant enhancement in accuracy and efficiency.

The contributions of our work can be summarised as:

- To handle the rapidly growing number of videos, we propose an inference framework for ReID systems, which can evolve current methods into a cloud-edge collaborative pattern, enhancing both efficiency and effectiveness.
- To boost the system's performance by increasing the return probability of the desired image, we design a Distribution-aware Correlation Modeling network that captures the spatial-temporal correlations of the scene.
- To demonstrate the superiority of our method, we introduce several evaluation protocols and conduct extensive experiments, with the results showcasing the significant enhancement achieved by our proposed framework.

2 Related Work

2.1 Object Re-identification

Earlier ReID methods are type-specific, relying on specific attributes of the object, and are applicable only to a particular type of objects, such as person ReID (Zheng et al. 2015; Ahmed, Jones, and Marks 2015; Cheng et al. 2016; Zheng, Zheng, and Yang 2017) and vehicle ReID (Liu et al. 2016a,b, 2017). As methods continue to advance, there is a growing trend towards developing generic ReID methods (Luo et al. 2019; Huynh 2021; He et al. 2021; Li, Sun, and Li 2023; Ye et al. 2022; Cheng et al. 2016; Sun et al. 2020; He et al. 2021) that are agnostic to the type of object being applied. All the above methods can be employed in our cloud-edge collaborative framework, partnering with DaCM for efficient and effective inference.

Since spatial-temporal information can be effortlessly obtained in a ReID system, some ReID methods (Huang et al. 2016; Cho et al. 2019; Wang et al. 2019) incorporate it to filter out unreasonable samples. Compared with them, our approach has several obvious differences : (i) Previous methods generate the spatial-temporal distribution through frequency statistics, whereas our approach employs a deep neural network to learn such correlations; (ii) Previous methods still adhere to centralized patterns, whereas our approach is developed within a cloud-edge collaborative framework; (iii) Previous methods only use such information to enhance performance, whereas our approach improves performance while achieving efficient inference. As a similar work, Jain et al. also interpolate such information in object searching, but the proposed Spatula directly filters out many candidate images that leads that (i) the desired image may not be found

even with the replay strategy, and (ii) it is hard to combined with neural networks for promising performance.

2.2 Cloud-Edge Collaborative Methods

Emerging cloud-edge collaboration approaches showcase their superiority in various systems and communication technologies. Noteworthy instances of these advanced methodologies are evident in seminal works, such as the collaborative occluded face recognition architecture (Zhang et al. 2023), the open-source framework SmartEye for real-time video analytic (Wang and Gao 2021), and the video service enhancement within an edge-cloud collaboration framework (Wu et al. 2021). The adaptation of cloud-device collaboration sensitive to changing environments (Gan et al. 2023), the real-time surveillance video analysis in Cloud-Edge architecture (Hou and Zhang 2021), and the Classification Driven Compression framework for reducing deep learning bandwidth consumption (Dong et al. 2020) further underscore the versatility and impact of these collaborative approaches. Unlike these methods, we focus on the ReID task and aim to achieve efficient and effective inference instead of model optimization.

3 Problem Definition

As shown in Fig. 2, given a query image $\mathcal{I}^q$ and auxiliary information $\{t^q, c^q, t^d\}$ (t^q and c^q denote the timestamp and camera ID of $\mathcal{I}^q$, respectively, while t^d represents the target time of the desired image), the ReID system sends these information to each edge device. Then, for the i-th edge device, it extracts the deep feature $\mathbf{f}^q$ from $\mathcal{I}^q$ via a local visual backbone and compute the similarity $\mathbf{s}^i \in \mathbb{R}^{N_i}$ between $\mathbf{f}^q$ and the features $\mathbf{G}^i \in \mathbb{R}^{N_i \times E}$ of all N_i gallery images on the i-th device (E is the dimension). The images in current edge device are ranked by $\mathbf{s}^i$ and sent to the cloud server in batches due to the limited bandwidth. Finally, users check the returned data, and the process can be terminated if the *desired images* are contained in current batch.

Denoting the user desired image as $\mathcal{I}^d$, we can see that the performance of cloud-edge collaborative ReID systems depends on when the $\mathcal{I}^d$ is returned to the server, which is influenced by: (1) the delay of the communication network; (2) the speed of feature extraction; (3) the rank of $\mathcal{I}^d$ in the uploading sequence, and (4) the amount of data that the camera (on which $\mathcal{I}^d$ is captured) can upload each time, i.e. the bandwidth allocated for the camera. The first two problems have been well studied by previous methods (Zhuang et al. 2020; Zhuang, Wen, and Zhang 2021; Jain et al. 2020; Kang et al. 2017; Zhang et al. 2020), but the last two problems still lack relevant methods. Therefore, in this paper, we focus on how to accelerate the system by advancing the position of $\mathcal{I}^d$ in the sequence and increasing the bandwidth utilization rate of the camera where $\mathcal{I}^d$ is located from the perspective of multimedia computing. The optimization objective to reduce the Transmission Number (TN) can be expressed as:

$$\arg\min_{\theta} \left(\Omega_i \left(\left\lceil \frac{\epsilon(\mathcal{U}^i(\mathbf{s}^i))}{b^i} \right\rceil \right) \right), \tag{1}$$

where $\mathcal{U}^i$ is the uploading image sequence of i-th edge device, which is determined by the score $\mathbf{s}$, b^i is the allocated

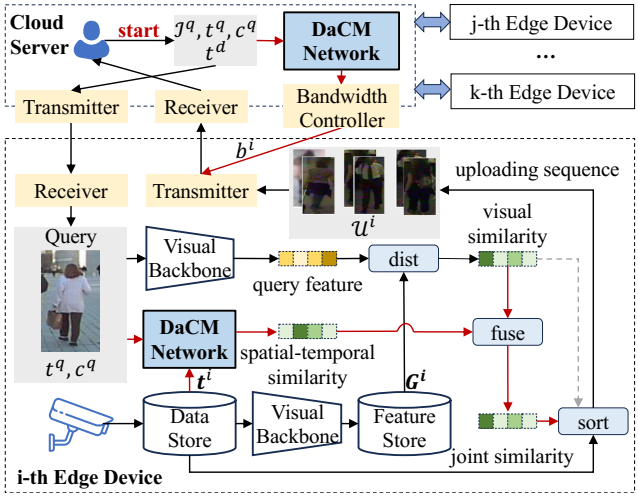

Figure 2: The overview of our proposed cloud-edge collaborative inference framework. The DaCM is deployed in both the cloud server and edge devices for adjusting uploading batch size b^i and image order in the uploading sequence. The red solid denotes the data flow enabled by the designed DaCM, and the gray dashed line denotes the previous data flow that can be removed by DaCM.

bandwidth for i-th edge device, and it means how many images can be uploaded each time, θ is the parameters should be optimized, which influence $\mathcal{U}^i$ and b^i. Function ϵ is used to return the index of $\mathcal{I}^d$ in $\mathcal{U}^i$, function Ω aggregate the results from all edge devices, and their implementations are contingent upon user requirements.

Most of previous methods pay much attention to learning a proper $\mathcal{U}^i$, i.e. forcing the images with same identity ID l^q have small $\mathbf{s}$ to make them at the front of the sequence, and they can not have an impact on b^i. In contrast, we propose the DaCM network, which can learn the spatial-temporal distribution of objects in scene and be used to boost the efficiency of the system by adjusting both $\mathcal{U}^i$ and b^i.

4 Proposed Approach

In this section, we first present the details of DaCM architecture and its training strategy, then describe how DaCM is used in the cloud-edge collaboration framework.

4.1 DaCM Architecture

The DaCM network performs an important role in the cloud-edge collaborative ReID system, and in this part, we describe its architecture. As shown in Fig. 3, DaCM mainly consists of three components, a spatial-temporal embedding module, multiple Correlation Modeling (CoMo) blocks, and a final classifier.

Spatial-temporal embedding. Inspired by the positional encoding manner used in (Vaswani et al. 2017), we first adopt the sinusoidal embedding to encode the temporal information. Denoted the timestamps of query image and target as t^q and t^d, we encode their difference to a feature vec-

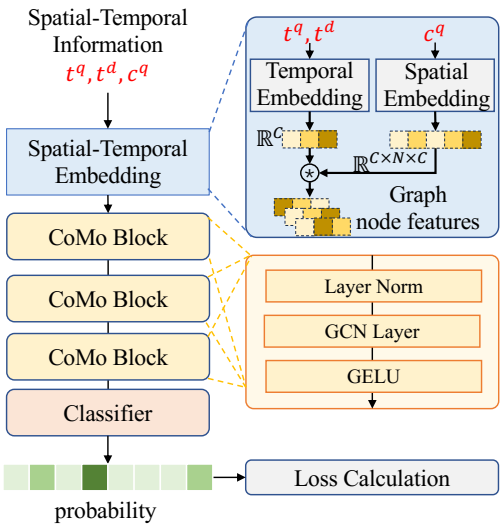

Figure 3: The architecture of DaCM network.

tor in the formulation of:

$$\mathbf{e}_{2i} = \sin\left(\frac{(t^d - t^q)}{\lambda^{\frac{2i}{D}}}\right), \ \mathbf{e}_{2i+1} = \cos\left(\frac{(t^d - t^q)}{\lambda^{\frac{2i}{D}}}\right), \quad (2)$$

where i is the dimension index, $\mathbf{e} \in \mathbb{R}^D$ is the results embedding, and D is the dimension of the embedding. λ denotes the max period of the sinusoidal function. The wavelengths form a geometric progression from 2π to $\lambda \cdot 2\pi$.

For the spatial information, considering the fixed topology of different cameras, we suggest employing learnable parameters $\mathbf{W} \in \mathbb{R}^{C\times D\times C\times D}$ to represent the spatial information. C is the number of cameras and the nodes in the graph. When c^q is provided as input, we choose $\mathbf{W}_{c^q}$ as the resulting spatial embedding, which is then combined with $\mathbf{e}$ to generate a graph structure:

$$\mathbf{A} = \frac{1}{\sum_j^D e_j} \sum_j^D \mathbf{W}_{c^q,j} * e_j + \mathbf{b} \in \mathbb{R}^{C\times D}, \quad (3)$$

where $\mathbf{b}$ is a learnable bias.

Correlation modeling block. As for CoMo blocks, they are used to propagate the information among graph nodes, and as shown in Fig. 3, one CoMo block is comprised of a Layer Normalization, a GCN layer (Kipf and Welling 2017), and a GeLU activation, which can be formulated as:

$$\mathbf{A}^{l+1} = \mathrm{GELU}\left(\mathbf{E}^l \times \mathrm{LN}(\mathbf{A}^l) \times \mathbf{W}^l\right), \quad (4)$$

where $\mathbf{E}^l \in \mathbb{R}^{C\times C}$ is the adjacency matrix of the GCN layer, which can be learned during training, and $\mathbf{W}^l \in \mathbb{R}^{C\times C}$ denotes the transfer weight to update the information for each node feature.

Classifier. Finally, we apply an MLP for each node feature in the graph as the classifier, whose output dimension is set to 1. The outputs of all nodes are concatenated as the final output of the DaCM network:

$$\mathbf{y} = [\mathrm{MLP}(\mathbf{G}_0); \ldots; \mathrm{MLP}(\mathbf{G}_{C-1})] \in \mathbb{R}^C. \quad (5)$$

The MLP consists of the sequence of BatchNorm (Ioffe and Szegedy 2015), ReLU, and fully connected layers.

4.2 Objective Function

In this part, we describe how to train the DaCM network. For the training set $\mathcal{D}^{\text{train}}$, we randomly select two samples (q and d) with the same label ID but different camera IDs, and their spatial-temporal information is denoted $\{l^q, t^q, c^q\}$ and $\{l^d, t^d, c^d\}$, respectively. According to the time difference $t^d - t^q$ and camera id c^q, we obtain the output $\mathbf{y}$ via the proposed network, and then we optimize the network via the expectation:

$$\arg\max_\theta \mathbb{E}_{(q,d)\sim\mathcal{D}^{\text{train}}}[P(y = c^d | t^q, t^d, c^q; \theta)], \quad (6)$$

where θ is the network parameters. In implementation, this expectation can be easily converted to a classification problem, and we adopt the cross-entropy loss function to generate the gradients. The classification solves such a problem: determining the camera at which an object will appear again after a duration of $t^d - t^q$, starting from camera c^q. the network will acquire knowledge about the topology of devices deployed in the system and some characteristics of the target's movement, thereby facilitating efficient inference.

4.3 Inference

As discussed in Sec. 3, to achieve efficient inference, the system should promptly return the desired image to the cloud server by generating appropriate $\mathbf{s}$ and b. We elaborate on how to accomplish this goal by using the DaCM network.

Cloud-level inference. For cloud-level inference, its emphasis is on assigning a large b^i to the edge devices containing the desired image. Assuming the total bandwidth allocated for C edge devices is B, a simple strategy would be to distribute B equally among edge devices, but it lacks flexibility. We aim to decrease data transmission in the connected network and alleviate stress on the system by dynamically allocating bandwidth to the edge devices based on the spatial-temporal correlation learned in the DaCM network.

Given the query information of $\{\mathcal{I}^q, t^q, c^q, t^d\}$, we send the spatial-temporal information into the DaCM network, and it produces $\mathbf{y} \in \mathbb{R}^C$. The output representation is the chance of the target appearing under each camera at moment t^d. Intuitively, if $\mathbf{y}_i > \mathbf{y}_j$, the i-th edge device should be allocated with a larger bandwidth than the j-th edge device. Thus, we formulate this process as:

$$\hat{b}^i = \mathrm{softmax}(\mathbf{y}/\gamma_0)_i * B, \quad (7)$$

where γ_0 is used to smooth the probability to avoid extreme values. However, it overlooks a crucial factor—the uneven distribution of data among edge devices. The number of images on the edge devices can vary significantly, necessitating the consideration of this factor when allocating bandwidth. Therefore, we finally assign the bandwidth b^i for the i-th edge device as:

$$b^i = \frac{z^i * B}{\sum_j z^j}, \ z^i = \phi\left(\frac{\mathbf{y}}{\gamma_0}\right)_i * \left(\frac{\exp(|\mathcal{G}^i|)}{\gamma_1 \sum_j \exp(|\mathcal{G}^j|)}\right), \quad (8)$$

where ϕ is the $\mathrm{softmax}$ function. Note that for cloud-level inference, t^d should be provided by the user.

Edge-level inference. For edge-level inference, the focus is on re-ranking the index of gallery images in the uploading sequence by generating proper $\mathbf{s}^i$. Denoted the query image with its spatial-temporal information as $\{\mathcal{I}^q, c^q, t^q\}$, one gallery image at the c^d-th edge device as $\{\mathcal{I}^d, c^d, t^d\}$, we send $\{c^q, t^q, t^d\}$ into DaCM and obtain the output $\mathbf{y}$. If $\mathbf{y}_{c^q}$ is small, it implies that $\mathcal{I}^q$ and $\mathcal{I}^d$ do not match in spatial-temporal correlation, resulting in a minimal likelihood of having the same ID as the query image. Applying this operation to all gallery images on the i-th edge device, we obtain spatial-temporal similarity. Next, we delve into how to combine such spatial-temporal similarity with visual similarity.

Given a reliable visual similarity, it is difficult to build a reliable joint metric because the spatial-temporal similarity is unreliable and it is hard to assign appropriate weighting factors for these two types of metrics. Inspired by the joint metric proposed in (Wang et al. 2019), we adopt a smoothing operator to alleviate unreliable probability estimation. Denoted the spatial-temporal similarity as $\mathbf{o} \in \mathbb{R}^{N_i}$ and the visual similarity as $\mathbf{v} \in \mathbb{R}^{N_i}$ (assume it is produced by cosine distance function, and large value in $\mathbf{v}$ means the two features are similar), where N_i is the number of gallery images in the i-th device, the joint similarity is computed as:

$$s_k^i = -\frac{1}{1+\alpha \exp\left(\phi(-\frac{\mathbf{o}}{\beta})_k\right)} \frac{1}{1+\exp(\mathbf{v}_k - 1)}, \quad (9)$$

where α and β are hyper-parameters to balance these similarities, and gallery images are re-ranked according to $\mathbf{s}^i$.

Time-constrained ReID. For a ReID system, users sometimes wish to search for targets near a specific time, a task challenging to accomplish solely based on visual features. Sorting only by time may introduce a large number of unrelated images. Therefore, the key to achieving tcReID lies in how to effectively combine time information with visual information. We observe that Eq. (9) offers a natural way to fulfill such a task. However, Eq. (9) does not satisfy the requirement because it does not introduce t^d into the similarity calculation. Therefore, we propose a new formulation to meet the requirements of tcReID task.

Denoted the query data and target as $\{\mathcal{I}^q, t^q, c^q, t^d\}$, we construct a pattern bank by calculating the correlation between the query image and gallery images in the edge device. DaCM takes in $\{\mathcal{I}^q, t^q, c^q, t^{g_i}\}$ (t^{g_i} is the timestamp of g_i-th gallery image) and output $\mathbf{a}^{g_i} \in \mathbb{R}^C$. This process is applied to all gallery images and we collect them as a pattern bank $\mathbf{B} \in \mathbb{R}^{N_i \times C}$ of $\mathcal{I}^q$. Then we send the true target time and query data into DaCM and output $\mathbf{y}$. We calculate the similarity for constructing an uploading image sequence in the form of:

$$\hat{s}_k^i = \frac{s_k^i}{1+\exp\left(\cos(\mathbf{B}_k, \mathbf{y}) - 1\right)}, \quad (10)$$

where $\cos$ denotes the cosine distance function. Finally, we sort the gallery images according to $\hat{\mathbf{s}}^i$ and return them to the cloud server in batches. In addition, we find that the solely using the outputs of DaCM may lead to outlier problems. Therefore, in order to ensure the stability, we combine the output of DaCM with the spatial-temporal correlation obtained by the frequency statistics method (Wang et al. 2019).

5 Experiments

5.1 Experimental Settings

Datasets. We mainly evaluate our proposed framework and method on the DukeMTMC-reID (Zheng, Zheng, and Yang 2017) and Market-1501 (Zheng et al. 2015) datasets, since they are annotated with high-quality timestamp.

Compared methods. We compare our method with several inference strategies to show its performance, including:

- *Pattern-C* denotes the conventional centralized inference strategy, which collects all images captured by edge devices and conducts similarities calculations in the cloud server. We use it as the baseline to show the boosting effectiveness of different inference strategies.
- *Pattern-CE* denotes a simple cloud-edge collaborative inference strategy: the amount of transmission is evenly distributed to each edge device and each edge device assigns the upload sequence according to the distance between the query image and gallery images. By comparing with this strategy, we can see the improvements brought by the proposed DaCM network.

Besides, we design a DaCM network to boost the efficiency of the ReID system, and we also replace the DaCM in the system with *stReID* (Wang et al. 2019), which uses frequency statistics to model spatial-temporal associations, and it interpolates the statistical spatial-temporal distribution into the similarity calculation process for person ReID.

Hyper-parameters: To train the DaCM network, we employ Adam (Kingma and Ba 2015) as the optimizer. The initial learning rate is set to 0.01 and is reduced by 10 for every 30 epochs. γ_0 and γ_1 are both set to 0.01 as the default. α and β are bot set to 0.1. λ in Eq. (2) is set to 10,000 as the default. B is set to $3 * C$, i.e., each edge device can upload an average of three images at a time.

5.2 Evaluation Protocols

We propose several novel protocols to show the performance of the cloud-edge collaborative inference. Let us initially provide a definition of the *desired image*, as the proposed protocols hinge upon this conceptual foundation. A desired image is a particular sample among the gallery images, sharing the same identity as the query image and possessing a timestamp in proximity to a given target time.

- *mean Transmission Number(mTN)*: it is a protocol used to present the efficiency of the method. For each pair of one query image and one gallery image with same identity ID, there exists one corresponding TN as shown in Eq.(1), and we average the TN of all pairs as mTN.
- *precise Rank@K (PR-K)*: it is calculated by checking whether top-k gallery images contain the desired image that has the same ID with the query image and is closest to the target time, so pR-K is a stricter protocol than R-K.

Methods	R-1↑	mAP↑
BoW+kissme (Zheng et al. 2015)	25.1	12.2
LOMO+XQDA (Liao et al. 2015)	30.8	17.0
PAN (Zheng, Zheng, and Yang 2019)	71.6	51.5
SVDNet (Sun et al. 2017)	76.7	56.8
HA-CNN (Li, Zhu, and Gong 2018)	80.5	63.8
APR (Lin et al. 2019)	70.7	51.9
Human Parsing (Kalayeh et al. 2018)	84.4	71.0
PSE+ECN (Sarfraz et al. 2018)	85.2	79.8
CLIP-ReID (Li, Sun, and Li 2023)	90.0	80.7
PCB (Sun et al. 2018)	82.3	70.7
PCB + stReID (Wang et al. 2019)	94.3	84.0
PCB + InSTD (Ren et al. 2021)	92.7	86.1
PCB + Ours (OE)	**96.2**	**89.5**
SBS (He et al. 2023)	90.8	79.9
SBS + stReID	95.4	83.0
SBS + Ours (OE)	**96.7**	**89.8**
TranReID (He et al. 2021)	90.8	81.8
TranReID + stReID	96.2	88.6
TranReID + Ours (OE)	**96.8**	**91.0**

Table 1: Boosting Effect on DukeMTMC-reID dataset.

Methods	R-1↑	mAP↑
BoW+kissme (Zheng et al. 2015)	44.4	20.8
PAN (Zheng, Zheng, and Yang 2019)	82.8	63.4
SVDNet (Sun et al. 2017)	82.3	62.1
HA-CNN (Li, Zhu, and Gong 2018)	91.2	75.7
APR (Lin et al. 2019)	84.3	64.7
Human Parsing (Kalayeh et al. 2018)	93.9	-
PSE+ECN (Sarfraz et al. 2018)	90.3	84.0
CLIP-ReID (Li, Sun, and Li 2023)	95.7	89.8
SBS (He et al. 2023)	95.8	**89.0**
SBS + stReID	96.1	86.8
SBS + Ours(OE)	**96.4**	88.2
TranReID (He et al. 2021)	95.2	89.0
TranReID + stReID	96.6	**89.8**
TranReID + Ours(OE)	**96.9**	89.0

Table 2: Boosting Effect on Market-1501 dataset.

Methods	C	CE	OC	OC+OE
mTN↓	1561	9.56	5.39	**4.43**

Table 3: Performance of different inference strategies.

- *mean precise Rank (mpR)*: For the i-th query data, when pR-k_i is successful but pR-(k_i-1) is not successful, we record its precise Rank as k_i, and we average the k_i of all query data as mpR.

5.3 Experimental Results

Boosting effect of DaCM for ReID methods. Since the edge-level inference in our method can be seen as one kind of re-ranking technologies, we embedded it to several visual ReID methods (PCB (Sun et al. 2018), SBS (He et al. 2023), TransReID (He et al. 2021)) to show the boosting performance. Methodologies for comparison can be categorized into several different groups, including several classical methods such as LOMO+XQDA (Liao et al. 2015) and handcrafted approach BoW+kissme (Zheng et al. 2015), explicit deep learning methods including PAN (Zheng, Zheng, and Yang 2019), SVDNet (Sun et al. 2017) and HA-CNN (Li, Zhu, and Gong 2018), attribute-centric techniques including APR (Lin et al. 2019), mask-guided strategies including Human Parsing (Kalayeh et al. 2018), part-based approaches like PSE+ECN (Sarfraz et al. 2018), pose-oriented techniques like PCB (Sun et al. 2018), and a recent work CLIP-ReID (Li, Sun, and Li 2023).

The results evaluated on DukeMTMC-reID dataset for comparison are shown in Table 1. Without bells and whistles, our method outperforms all existing methods on the DukeMTMC-reID dataset. In addition, the robustness of our methodology is further highlighted when employing the same visual stream method. For instance, integrated with SBS (He et al. 2023), our approach outperforms stReID (Wang et al. 2019), elevating the rank-1 accuracy from 95.4% to 96.7%, and boosting mAP from 83.0% to 89.8%. Besides, the results evaluated on Market1501 dataset for comparison are shown in Table 2, and our method still gain obvious improvements than the baselines in term of the R-1.

Efficiency of the proposed ReID system. We compare the proposed approach with the strategies introduced in Sec. 5.1, and the results are shown in Table 3, where *C* and *CE* denotes the *Pattern-C* and *Pattern-CE*, respectively. *OC* and *OE* denotes using cloud-level inference and edge-level inference. Experiments are conducted on DukeMTMC-reID dataset. By analyzing the mTN values of different strategies, we can see that *Pattern-C* obtains a huge number of mTN since it requires uploading all images to the cloud server, and a simple cloud-edge collaborative framework (*Pattern-CE*) reduces mTN to 9.56, which saves much network traffic. Meanwhile, the results in the table also show that using DaCM network alone in the cloud server or in the edge devices can reduce the mTN to a certain extent, and the combination of them can lead to an optimal result. As for the protocols of pR-1 and mpR, most of the previous methods do not take it into consideration, and their methods only produce meaningless output. As shown in Table 4, if we use the visual similarity (*C* in table), it only achieves 0.94 pR-1. However, our method can achieve 44.74 pR-1, which is still low but it demonstrates that the proposed approach can be applied to the challenging task. *Lin* and *Sam* are two kinds of spatial-temporal embedding methods, and their details please refer to our extended version.

5.4 Ablation Study

α and β are two hyper-parameters used in Eq. 9, which will affect the image order in the uploading sequence. Thus, we conduct two sensitivity analysis experiments to investigate their impact on our ReID system. For the protocol of mTN, we only show the results generated by only using the DaCM network in the edge devices. As shown in Fig. 4, when α is in the range of 1.0$\sim$10.0, the system has much worse performance, and when α is set to 100, the system achieves a low

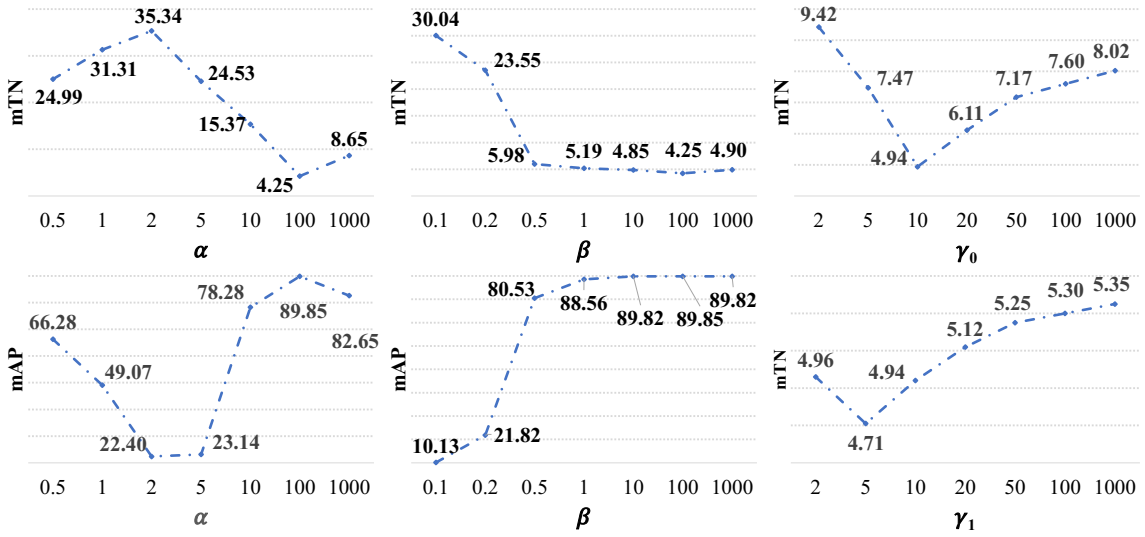

Figure 4: The effects of different values of α, β, γ_0, and γ_1.

protocol	C	Ours(Lin)	Ours(Sam)	Ours
pR-1↑	0.94	20.38	23.37	**44.74**
mpR↓	123.77	20.56	9.35	**1.48**

Table 4: Performance of different embedding strategies.

mTN value and a high mAP value. Besides, the performance improves as the value β increases.

γ_0 and γ_1 are used in Eq. (8) to adjust the bandwidth assigned for each edge devices. Thus, we adjust their different values to show their impact on system performance. The results are shown in Fig. 4. Since these two hyper-parameters only affect the traffic of the connected network, we only present their effect for the protocol of mTN.

5.5 Visualization

To help understand our approach, we provide some visualization examples to illustrate the impact of our proposed method, as depicted in Fig. 5. Instances marked with a red box signify inconsistency with the query image, while those marked with a green box indicate consistency.

These visualizations underscore the challenges of distinguishing certain images based solely on visual appearance. For instance, in the initial set of images (a), the individual in the third image is in a blue long-sleeved shirt and black pants, sharing a notable resemblance with the person in the query image. In the second row of the visualization results, the woman in the first two images of (d) wears a black hoodie, denim pants, and black boots, and holds white rolls of paper, exhibiting a noticeable similarity in appearance with the person in the query image. However, as can be seen from (f), the pedestrian in the correct images is facing away from the camera, and the coat does not exhibit any features of the white paper roll. In contrast, the person in the third image of (e) is attired in all black, and lacks a hat, but shares a dark hair color, bearing a strong resemblance to

Figure 5: Visualization of retrieval examples.

the pedestrians depicted in the two returned images. However, the stReID method fails to filter out this incorrect result based on the spatial-temporal statistic approach, and our method effectively filters out unreliable returned images.

6 Conclusion

The increasing volume of videos makes the traditional centralized ReID system impractical, and the current cloud-edge collaborative methods face challenges related to bandwidth constraints and search efficiency. To address these problems, we introduce a pioneering cloud-edge collaborative ReID framework. By leveraging a distribution-aware correlation modeling network, our approach enables efficient inference, ensuring the desired image returns to the cloud server as early as possible. Comparative experiments demonstrate our approach can reduce data transmission and improve the performance across various baselines, showing its superiority. We also acknowledge that it requires time stamps and is inappropriate for mobile devices, which motivate our future research on utilizing unstable spatial-temporal data to achieve high-quality correlation learning.

Acknowledgments

This work was supported in part by the Funds for the NSFC Project under Grant U24B20176, 62202063, 62406038, 62302058, and China Postdoctoral Science Foundation under Grant 2024M760280.

References

Ahmed, E.; Jones, M. J.; and Marks, T. K. 2015. An improved deep learning architecture for person re-identification. In *CVPR*, 3908–3916.

Angel, N. A.; Ravindran, D.; Vincent, P. M. D. R.; Srinivasan, K.; and Hu, Y. 2022. Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. *Sensors*, 22(1): 196.

Chen, H.; Wang, Y.; Lagadec, B.; Dantcheva, A.; and Brémond, F. 2021. Joint Generative and Contrastive Learning for Unsupervised Person Re-Identification. In *IEEE CVPR*, 2004–2013.

Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; and Zheng, N. 2016. Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function. In *CVPR*, 1335–1344.

Cho, Y.; Kim, S.; Park, J.; Lee, K.; and Yoon, K. 2019. Joint person re-identification and camera network topology inference in multiple cameras. *CVIU*, 180: 34–46.

Dong, Y.; Zhao, P.; Yu, H.; Zhao, C.; and Yang, S. 2020. CDC: Classification Driven Compression for Bandwidth Efficient Edge-Cloud Collaborative Deep Learning. In *IJCAI*, 3378–3384.

Fan, H.; Zheng, L.; Yan, C.; and Yang, Y. 2018. Unsupervised Person Re-identification: Clustering and Fine-tuning. *ACM TOMM*, 14(4): 83:1–83:18.

Fu, H.; Cui, K.; Wang, C.; Qi, M.; and Ma, H. 2024. Mutual Distillation Learning for Person Re-Identification. *IEEE TMM*, 26: 8981–8995.

Gan, Y.; Pan, M.; Zhang, R.; Ling, Z.; Zhao, L.; Liu, J.; and Zhang, S. 2023. Cloud-Device Collaborative Adaptation to Continual Changing Environments in the Real-World. In *CVPR*, 12157–12166.

Ge, Y.; Zhu, F.; Chen, D.; Zhao, R.; Wang, X.; and Li, H. 2024. Structured Domain Adaptation With Online Relation Regularization for Unsupervised Person Re-ID. *IEEE TNNLS*, 35(1): 258–271.

He, L.; Liao, X.; Liu, W.; Liu, X.; Cheng, P.; and Mei, T. 2023. FastReID: A Pytorch Toolbox for General Instance Re-identification. In *ACM MM*, 9664–9667.

He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; and Jiang, W. 2021. TransReID: Transformer-based Object Re-Identification. In *ICCV*, 14993–15002.

Hou, B.; and Zhang, J. 2021. Real-time Surveillance Video Salient Object Detection Using Collaborative Cloud-Edge Deep Reinforcement Learning. In *IJCNN*, 1–8.

Huang, W.; Hu, R.; Liang, C.; Yu, Y.; Wang, Z.; Zhong, X.; and Zhang, C. 2016. Camera Network Based Person Re-identification by Leveraging Spatial-Temporal Constraint and Multiple Cameras Relations. In *MMM*, volume 9516, 174–186.

Huynh, S. V. 2021. A Strong Baseline for Vehicle Re-Identification. In *CVPRW*, 4147–4154.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, volume 37, 448–456.

Jain, S.; Zhang, X.; Zhou, Y.; Ananthanarayanan, G.; Jiang, J.; Shu, Y.; Bahl, P.; and Gonzalez, J. 2020. Spatula: Efficient cross-camera video analytics on large camera networks. In *IEEE/ACM SEC*, 110–124.

Jiang, P.; Xin, K.; Li, C.; and Zhou, Y. 2023. High-efficiency Device-Cloud Collaborative Transformer Model. In *CVPR*, 2204–2210.

Kalayeh, M. M.; Basaran, E.; Gökmen, M.; Kamasak, M. E.; and Shah, M. 2018. Human Semantic Parsing for Person Re-Identification. In *CVPR*, 1062–1071.

Kang, D.; Emmons, J.; Abuzaid, F.; Bailis, P.; and Zaharia, M. 2017. NoScope: optimizing neural network queries over video at scale. *Proc. VLDB Endow.*, 10(11): 1586–1597.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Li, S.; Sun, L.; and Li, Q. 2023. CLIP-ReID: Exploiting Vision-Language Model for Image Re-identification without Concrete Text Labels. In *AAAI*, 1405–1413.

Li, W.; Zhu, X.; and Gong, S. 2018. Harmonious Attention Network for Person Re-Identification. In *CVPR*, 2285–2294.

Liao, S.; Hu, Y.; Zhu, X.; and Li, S. Z. 2015. Person re-identification by Local Maximal Occurrence representation and metric learning. In *CVPR*, 2197–2206.

Lin, Y.; Zheng, L.; Zheng, Z.; Wu, Y.; Hu, Z.; Yan, C.; and Yang, Y. 2019. Improving person re-identification by attribute and identity learning. *Pattern Recognit.*, 95: 151–161.

Liu, W.; Liu, X.; Ma, H.; and Cheng, P. 2017. Beyond Human-level License Plate Super-resolution with Progressive Vehicle Search and Domain Priori GAN. In *ACM MM*, 1618–1626.

Liu, X.; Liu, W.; Ma, H.; and Fu, H. 2016a. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, 1–6.

Liu, X.; Liu, W.; Mei, T.; and Ma, H. 2016b. A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. In *ECCV*, volume 9906, 869–884.

Liu, X.; Liu, W.; Mei, T.; and Ma, H. 2018. PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance. *IEEE TMM*, 20(3): 645–658.

Luo, H.; Gu, Y.; Liao, X.; Lai, S.; and Jiang, W. 2019. Bag of Tricks and a Strong Baseline for Deep Person Re-Identification. In *CVPRW*, 1487–1495.

Qi, M.; Qin, J.; Yang, Y.; Wang, Y.; and Luo, J. 2021. Semantics-Aware Spatial-Temporal Binaries for Cross-Modal Video Retrieval. *IEEE TIP*, 30: 2989–3004.

Qi, M.; Wang, Y.; and Li, A. 2017. Online Cross-Modal Scene Retrieval by Binary Representation and Semantic Graph. In *ACM MM*, 744–752.

Ren, M.; He, L.; Liao, X.; Liu, W.; Wang, Y.; and Tan, T. 2021. Learning Instance-level Spatial-Temporal Patterns for Person Re-identification. In *ICCV*, 14910–14919.

Sarfraz, M. S.; Schumann, A.; Eberle, A.; and Stiefelhagen, R. 2018. A Pose-Sensitive Embedding for Person Re-Identification With Expanded Cross Neighborhood Re-Ranking. In *CVPR*, 420–429.

Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In *CVPR*, 6397–6406.

Sun, Y.; Zheng, L.; Deng, W.; and Wang, S. 2017. SVDNet for Pedestrian Retrieval. In *ICCV*, 3820–3828.

Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; and Wang, S. 2018. Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline). In *ECCV*, volume 11208, 501–518.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*, 5998–6008.

Wang, G.; Lai, J.; Huang, P.; and Xie, X. 2019. Spatial-Temporal Person Re-Identification. In *AAAI*, 8933–8940.

Wang, X.; and Gao, G. 2021. SmartEye: An Open Source Framework for Real-Time Video Analytics with Edge-Cloud Collaboration. In *ACM MM*, 3767–3770.

Wu, D.; Bao, R.; Li, Z.; Wang, H.; Zhang, H.; and Wang, R. 2021. Edge-Cloud Collaboration Enabled Video Service Enhancement: A Hybrid Human-Artificial Intelligence Scheme. *IEEE TMM*, 23: 2208–2221.

Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; and Hoi, S. C. H. 2022. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE TPAMI*, 44(6): 2872–2893.

Zhang, H.; Zhou, A.; Lu, J.; Ma, R.; Hu, Y.; Li, C.; Zhang, X.; Ma, H.; and Chen, X. 2020. OnRL: improving mobile video telephony via online reinforcement learning. In *MobiCom*, 29:1–29:14.

Zhang, P.; Huang, F.; Wu, D.; Yang, B.; Yang, Z.; and Tan, L. 2023. Device-Edge-Cloud Collaborative Acceleration Method Towards Occluded Face Recognition in High-Traffic Areas. *IEEE TMM*, 25: 1513–1520.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable Person Re-identification: A Benchmark. In *ICCV*, 1116–1124.

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro. In *ICCV*, 3774–3782.

Zheng, Z.; Zheng, L.; and Yang, Y. 2019. Pedestrian Alignment Network for Large-scale Person Re-Identification. *IEEE TCSVT*, 29(10): 3037–3045.

Zhong, Z.; Zheng, L.; Cao, D.; and Li, S. 2017. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 1318–1327.

Zhuang, W.; Wen, Y.; and Zhang, S. 2021. Joint Optimization in Edge-Cloud Continuum for Federated Unsupervised Person Re-identification. In *ACM MM*, 433–441.

Zhuang, W.; Wen, Y.; Zhang, X.; Gan, X.; Yin, D.; Zhou, D.; Zhang, S.; and Yi, S. 2020. Performance Optimization for Federated Person Re-identification via Benchmark Analysis. *CoRR*, abs/2008.11560.