DyGLIP: A Dynamic Graph Model with Link Prediction for Accurate Multi-Camera Multiple Object Tracking

Kha Gia Quach[1], Pha Nguyen[2], Huu Le[3], Thanh-Dat Truong[4], Chi Nhan Duong[1]
Minh-Triet Tran[5], Khoa Luu[4]
[1] Concordia University, CANADA [2] VinAI Research, VIETNAM
[3] Chalmers University of Technology, SWEDEN [4] University of Arkansas, USA
[5] University of Science, VNU-HCM, VIETNAM
[1]{dcnhan, kquach}@ieee.org, [2] v.phana@vinai.io, [3] huul@chalmers.se,
[4]{tt032, khoaluu}@uark.edu, [5]tmtriet@fit.hcmus.edu.vn

Abstract

Multi-Camera Multiple Object Tracking (MC-MOT) is a significant computer vision problem due to its emerging applicability in several real-world applications. Despite a large number of existing works, solving the data association problem in any MC-MOT pipeline is arguably one of the most challenging tasks. Developing a robust MC-MOT system, however, is still highly challenging due to many practical issues such as inconsistent lighting conditions, varying object movement patterns, or the trajectory occlusions of the objects between the cameras. To address these problems, this work, therefore, proposes a new Dynamic Graph Model with Link Prediction (DyGLIP) approach [1] to solve the data association task. Compared to existing methods, our new model offers several advantages, including better feature representations and the ability to recover from lost tracks during camera transitions. Moreover, our model works gracefully regardless of the overlapping ratios between the cameras. Experimental results show that we outperform existing MC-MOT algorithms by a large margin on several practical datasets. Notably, our model works favorably on online settings but can be extended to an incremental approach for large-scale datasets.

1. Introduction

Multi-Camera Multiple Object Tracking (MC-MOT) plays an essential role in computer vision due to its potential in many real-world applications such as self-driving cars, crowd behavior analysis, anomaly detection, etc. Although recent Multi-Camera Multiple Object Tracking (MC-MOT) methods have achieved promising results in several large-scale datasets, there are still many challenges that need to be addressed. Among them, data association, a crucial step in determining the performance of an MC-MOT pipeline, has attracted a considerable amount of attention lately. While

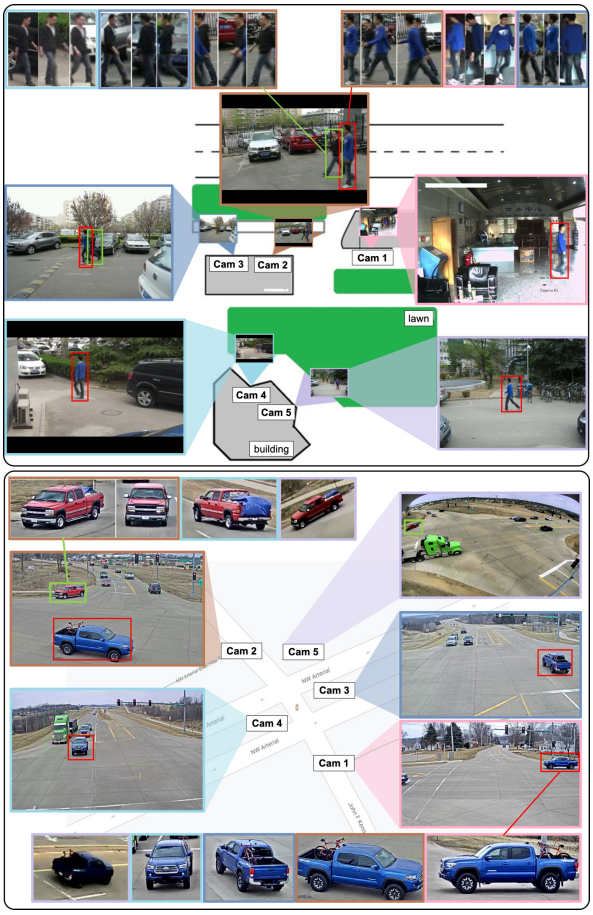

Figure 1. Top: Pedestrians in the MCT dataset [9], Bottom: Cars in the CityFlow dataset [37].

[1] Visit https://github.com/uark-cviu/DyGLIP for the implementation of DyGLIP.

the association itself is challenging in single-camera MOT applications, this task becomes even more difficult in MC-MOT settings due to possibly inconsistent lightning conditions or occlusion patterns between the cameras. For example, feature vectors, given by an off-the-shelf object detector and Re-Identification (Re-ID) models, associated with a particular object during its transition from an indoor to an outdoor environment can be totally different as shown in Fig. 1. This breaks the connection between the trajectory representation of an object and its previous feature vectors, leading to wrong identity (ID) assignments. Failures in data association at a particular frame will potentially lead to long-term detrimental effects for an online tracking system. In practice, many circumstances that are much more complex than simply generate more new IDs which could significantly deteriorate the tracking accuracy. Therefore, improving data association plays a crucial role in determining the performance of an MC-MOT algorithm.

To tackle the problem of data association mentioned above, our work introduces a totally new perspective to tackle the association task in MC-MOT. Most previous works rely on feature vectors obtained from an underlying object detector and use them to solve the assignment problem, e.g., using nearest neighbors [33], clustering [29, 50] or solving an instance of non-negative matrix factorization [16]. Instead, we propose to consider the problem of data association as *a link prediction on a graph*, where the graph vertices are associated with the tracks. To construct our predictor, we introduce a new dynamic graph formulation that can take into account the temporal information of an object over a period of time and its relation to other objects. As shown in the experiments in Section 4.3, this dynamic graph formulation allows leveraging the feature representations and moving patterns of each object to improve ID assignment, leading to better performance compared to the state-of-the-art (SOTA) methods.

Contributions The main contributions of our work can be summarized as follows. Firstly, a new MC-MOT framework is presented using the link prediction in conjunction with a dynamic graph formulation. We demonstrate that this new model significantly improves the association task in numerous MC-MOT datasets by a large margin. To the best of our knowledge, it is the first time link prediction and dynamic graph are used together in MC-MOT. Secondly, the proposed dynamic graph will be incorporated with *the attention mechanism*, allowing dynamically accumulating temporal and spatial information to result in a new graph embedding yielding highly accurate link prediction results.

2. Related Work

This section reviews some current methods on MC-MOT algorithms. There is a large amount of research on single-camera MOT tackling the matching/association problem [20, 27, 7, 23] or building an end-to-end framework that unified with the detector [36, 43, 40, 3, 11, 54, 22, 32, 19, 46, 47]. MC-MOT has recently received increasing attention intending to determine the global trajectory of all subjects in a multiple camera system simultaneously. Compared to single-camera MOT, MC-MOT is more challenging in the affinity stage. The difficulties may come from the significant changes in subject pose between cameras, a vast number of matching trajectories, or differences in object features.

Matching using spatio-temporal constraints: Kumar et al. [33] use pre-defined spatio-temporal camera connections, represented by an adjacent moving time matrix, to search for the targeted person when he/she disappeared in a field of view. Similarly, Jiang et al. [18] estimate the camera topology to significantly reduced the number of matching candidates. Styles et al. [35] propose several baseline methods that only solve a minor task that is classifying which camera will a disappeared target recur in. Jiang et al. [18] apply Gaussian Mixture Model to estimate the camera connectivity, therefore searching in probe tracklets set is less painful. Chen et al. [10] formulate the assignment task as a min-cost flow matching problem in a network.

3D matching on an overlapping field of view: You and Jiang [49] introduce a method that works specifically on overlapping fields of view by estimating people's ground heatmap. Chen et al. [8] reconstruct 3D geometric information to overcome 2D matching's limitations efficiently.

General matching approaches: He et al. generate local tracklets offline in all single views to construct a similarity matrix and then estimate global targets and their trajectory by performing the Matrix Factorization method. Ristani and Tomasi [29] adopt correlation clustering to solve the ID assignment task. Then two post-processing steps, i.e., interpolation and elimination, are also executed to fill gaps and filter indecision tracks. Yoon et al. [48] not only revisit the Multiple Hypothesis Tracking algorithm, which maintains a set of track hypotheses all the time but also reduce its expensiveness by introducing a gating mechanism for tree pruning. Zhang et al. [50] cluster IDs by using the Re-Ranking algorithm [52] on the global cost matrix.

3. The Proposed *DyGLIP* Method

In this section, we first briefly define the MC-MOT problem in Subsection 3.1. It is then re-formulated as a construction of a dynamic graph with an ID assignment problem in Subsection 3.2. A new attention mechanism is presented to enhance the robustness of the associated features for each node in the proposed graph in Subsection 3.3. Then ID assignment problem is solved via link prediction as in Subsection 3.4 to connect new nodes with existing nodes. Finally, we present the model learning in Section 3.5 and some anal-

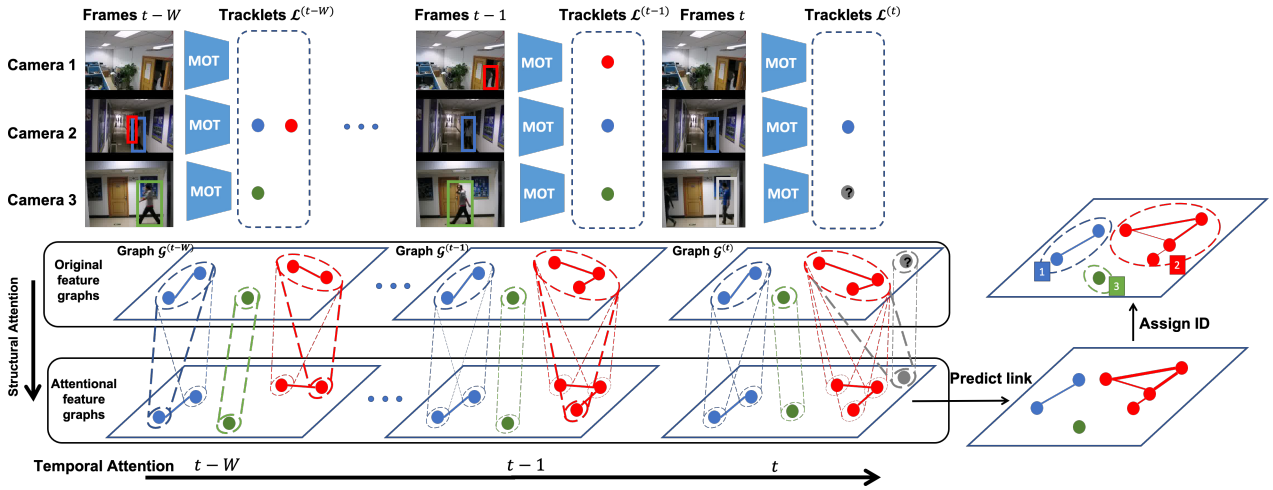

Figure 2. The proposed DyGLIP framework. Connected components, grouped by dash boundaries, represent known tracklets of unique objects in a multi-camera system. The gray node with a question mark is a new subject that has appeared in camera 3 at time step t. Such a node is successively added to our graph over time, its transformed features are computed by attending to existing nodes in the graph, and then its connections are predicted using structural and temporal attention feature embedding from graphs up to $t-W$ steps.

ysis on model complexity in Subsection 3.6.

3.1. Problem Formulation

In MC-MOT, it is assumed that the environment is continuously monitored by C cameras, denoted by the set $\mathcal{C} = \{c_1, \ldots, c_C\}$. In contrast to some prior methods [49, 8] that require certain overlapping ratios between the cameras, DyGLIP can gracefully handle both overlapping and non-overlapping scenarios. Similar to prior MC-MOT methods [16, 26], the task of tracking in each camera is assumed to be performed by an off-the-shelf single-camera MOT tracker. We choose DeepSORT [41] in this work, but it can be simply replaced by any other MOT trackers. At time step t, we obtain a set of local tracklets $\mathcal{L}_c^{(t)} = \{\mathbf{l}_j^{(t)}\}$ provided by each single-camera MOT tracker, where each $\mathbf{l}_j^{(t)}$ is a feature vector. Sometimes it is referred to as re-id features in the literature. Note that the set $\mathcal{L}_c^{(t)}$ may contain tracklets of new objects, and the global IDs of the tracklets are unknown. Moreover, one object could associate with multiple local tracklets across camera views. The aim of data association is, therefore, to assign the proper global IDs for the local tracklets in $\mathcal{L}_c^{(t)}$.

In practice, given an unassigned tracklet $\mathbf{l}_j^{(t)}$, most MC-MOT algorithms rely on the information in $\mathbf{l}_j^{(t)}$ to find its correlation to the set of known tracklets $\cup_{c=1}^{C} \cup_{k=1}^{t-1} \mathcal{L}_c^{(k)}$ obtained from previous time steps. In this work, we show that solely using information from the feature vectors $\mathbf{l}_j^{(t)}$ could lead to a wrong data association. Our proposed method, on the other hand, uses the set of obtained tracklets and embeds them into a dynamic graph model. The graph is then equipped with structural and temporal attention, offering us a richer tracklet representation.

3.2. Dynamic Graph Formulation

We first introduce our graph formulation (See Fig. 2). At a particular time step t, we construct a graph $\mathcal{G}^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)})$, where the vertex set $\mathcal{V}_t$ contains *all the tracklets* tracked up to time t. Unlike existing works that build a fixed graph and conduct relevant data association algorithms, we maintain a *dynamic graph* during our tracking process, which is a key novelty of our work. More specifically, at each time step t, new vertices are added into our graph, hence our vertices are growing over time, i.e., $\mathcal{V}^{(t)} = \mathcal{V}^{(t-1)} \cup \mathcal{N}^{(t)}$, where we use $\mathcal{N}^{(t)}$ to denote the set of new vertices, i.e., new tracklets, obtained from the MOT tracker in each camera. Before the data association step, the connection between these new vertices to the vertices of $\mathcal{V}^{(t-1)}$ is not determined.

We denote $f(v)$ as the feature vector associate with a node $v \in \mathcal{V}^{(t)}$. Given two nodes v_i and v_j, an edge exists that links the two vertices if these two tracklets represent the same object. From our experiments, we choose $f(v)$ to be the re-id feature of the tracklet associated with node v. In the following, we will discuss how the attention mechanism can be utilized to extract the embedding features for each node dynamically and how the robust connections can be established using link prediction.

3.3. Dynamic Graph with Self-Attention Module

We now introduce the self-attention module in our proposed dynamic graph and its building blocks, i.e., graph structural and temporal attention layers. Although using attention for dynamic graph has been considered in the literature [42, 39, 31], their methods cannot be straightforwardly applied to our problem. Inspired by these works, we propose a novel self-attention mechanism (see Fig. 2) for our

model. The goal of these attention layers is to attend inter-cameras and intra-cameras information, which allows our model to capture variations between tracklets, as well as attend the temporal information over multiple time steps.

3.3.1 Graph Structural Attention Layer

Our attention layer takes into account not only the provided embedding features but also the camera information. In other words, the structural attention layer (SAL) takes the concatenation of node embeddings or features, i.e., $f(v) \in \mathbb{R}^{D_F}$, and its camera positional encoding, i.e., $\mathbf{c}_v \in \mathbb{R}^{D_C}$, as the input, $\mathbf{e}_v = \{f(v)||\mathbf{c}_v\} \in \mathbb{R}^{D_E}$, where $D_E = D_F + D_C$.

Given a set of camera-aware node features from a graph $\mathcal{G}^{(t)}$, $\mathbf{e}^{(t)} = \{\mathbf{e}_1^t, \cdots, \mathbf{e}_N^t\}$, where $N = |\mathcal{V}^{(t)}|$, our structural attention layer provides a new set of node features $\mathbf{h}^{(t)} = \{\mathbf{h}_1^t, \cdots, \mathbf{h}_N^t\}, \mathbf{h}_v \in \mathbb{R}^{D_H}$, as the output. The learning of self-attention features can be stabilized with multi-head attention, where the input features are transformed by L independent transformation and their transformed features are concatenated as in Eqn. (1).

$$\mathbf{h}_{v_i}^t = \underset{l=1}{\overset{L}{\text{Concat}}} \left[\sigma \left(\sum_{v_j \in \mathcal{V}^{(t)}} \alpha_{ij}^l \text{conv}_{1\times1}^l \left(\mathbf{e}_{v_j}^t \right) \right) \right] \quad (1)$$

where $\text{conv}_{1\times1}^l$ is a 1D convolutional layer with kernel size 1×1, $\sigma(\cdot)$ is a non-linear activation function, i.e., LeakyRELU, and α_{ij}^l are the attention coefficients for the l-th attention head as in Eqn. (2).

$$\alpha_{ij}^l = \frac{\exp\left(\sigma\left(\mathbf{W}_{ij}^T \left[\text{conv}_{1\times1}^l(\mathbf{e}_{v_i}^t) || \text{conv}_{1\times1}^l(\mathbf{e}_{v_j}^t)\right]\right)\right)}{\sum_{v_k \in \mathcal{V}^{(t)}} \exp\left(\sigma\left(\mathbf{W}_{kj}^T \left[\text{conv}_{1\times1}^l(\mathbf{e}_{v_k}^t) || \text{conv}_{1\times1}^l(\mathbf{e}_{v_j}^t)\right]\right)\right)} \quad (2)$$

where $\|$ is the feature vector concatenation operation, $\mathbf{W}_{ij}^T \in \mathbb{R}^{D_E \times D_H}$ is the shared weight of the transformation applied to edge (v_i, v_j) in the graph at time step t. Those normalized (by softmax) attention coefficients α_{ij}^l indicate the impact of node v_j's features to node v_i. To incorporate graph structure, we employ sparse adjacent matrices to compute α_{ij}^l for nodes j within its neighborhood of node i while ignoring others. This layer can be stacked to create multiple structural attention layers and applied independently for each graph $\mathcal{G}^{(t)}$ to capture the local structure of a node at each time step. In the next section, we will introduce how to attend to the graph dynamic's evolution or development across multiple time steps by using the temporal attention layer, which takes the node representation output from the structural attention layers.

3.3.2 Graph Temporal Attention Layer

A temporal attention layer (TAL) is designed to capture the temporal evolution scheme in terms of links between nodes in a set of dynamic graphs. This layer takes a set of output representations or features from structural attention layers at different time steps and timestamps as inputs and provides the time-aware features for each node at each time step t. In particular, for each node $v \in \mathcal{V}^{(t)}$, we combine timestamp position encoding ($\mathbf{p}_v \in \mathbb{R}^{D_H}$) with the output from the SAL to obtain an order-aware sequence of input features for TAL as, $\mathbf{x}_v = \left[\mathbf{h}_v^1 + \mathbf{p}_v^1, \mathbf{h}_v^2 + \mathbf{p}_v^2, \cdots, \mathbf{h}_v^W + \mathbf{p}_v^W\right]$, where W is temporal window size, i.e., W = 3 in our experiments, for each node $v \in \mathcal{V}$. We stacked multiple time step together as matrices $\mathbf{X} \in \mathbb{R}^{W \times D_H}$. The l-th output of the multi-head TAL are defined using scaled dot-product attention as in Eq. (3)

$$\mathbf{z}_e^{(l)} = \text{attn}^{(l)}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_Z}} + \mathbf{M}\right)\mathbf{V} \quad (3)$$

where $l = 1, \cdots, L$, with L is the number of heads, $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_V$ are the "queries", "key" and "values" transformed features by linear projection matrices $\mathbf{W}_Q \in \mathbb{R}^{D_H \times D_Z}$, $\mathbf{W}_K \in \mathbb{R}^{D_H \times D_Z}$ and $\mathbf{W}_V \in \mathbb{R}^{D_H \times D_Z}$, respectively, as defined in [38], D_Z is the output feature dimension and $\mathbf{M} \in \mathbb{R}^{W \times W}$ is the mask matrix where each element $\mathbf{M} \in \{-\infty, 0\}$. We employ such masked attention to allow backward attention where each time step attends over all previous time steps. Thus, to set a zero attention weight, we assign $\mathbf{M}_{ij} = -\infty$, where $i > j$, otherwise $\mathbf{M}_{ij} = 0$. Similarly, we can create multiple stacked temporal attention layers by stacking temporal attention layers together. Then the multi-head output of the final layer will be concatenated and passed to a feed-foward neural network to capture non-linear interactions between the transformed features to provide the final set of node embeddings $\{\mathbf{e}'^{(1)}, \mathbf{e}'^{(2)}, \cdots, \mathbf{e}'^{(W)}\}$ as in [42].

3.4. Link Prediction

This section describes how we compute the probability of having a connection/link between two nodes and how we learn a link classifier jointly with the attention module to obtain the transformed features. In this way, we can guarantee that the self-attention module and the classifier provide the best link prediction accuracy. Given transformed features of a pair of nodes ($\mathbf{e}'^{(t)}_{v_i}$ and $\mathbf{e}'^{(t)}_{v_j}$), we compute the features or measurement that represent the similarity between those two nodes, and then it will be used as input for the classifier. The classifier provides a probability score $s \in [0, 1]$. The higher the score is, the more likely the two nodes are linked. We try with two different measurements and classifiers in Section 4.3, i.e., cosine distance by computing dot product of two feature vectors with Sigmoid as classifier and the Hadamard operator ($\mathbf{e}'^{(t)}_{v_i} \odot \mathbf{e}'^{(t)}_{v_j}$) with a fully connected layer and a softmax layer as classifier.

3.5. Model Learning

To train the proposed DyGLIP framework, our objective function is to learn representations capturing both structural and temporal information from dynamic graphs as well as to predict possible links between two arbitrary nodes in the graph using the learned representations. We first use a binary cross-entropy loss function to enforce nodes within a connected component to have similar feature embeddings. Then a classifier loss function to ensure classify two nodes based on measurement features as having a link or not.

$$\mathcal{L}(v_i) = \sum_{t}^{T} \Bigg(\sum_{v_j \in \mathcal{N}_b^{(t)}(v_i)} -\log\left(\sigma\left(< e'_{v_i}, e'_{v_j} >\right)\right) - w_g \sum_{v_k \in \mathcal{N}_g^{(t)}(v_i)} log\left(1 - \sigma\left(< e'_{v_i}, e'_{v_k} >\right)\right) + \sum_{v_j \in \mathcal{N}_a^{(t)}(v_i)} \mathcal{L}_c(v_i, v_j) \Bigg) \tag{4}$$

where $< \cdot >$ is the inner production between two vectors, σ is Sigmoid activation function, $\mathcal{N}_b^{(t)}(v_i)$ is the set of fixed-length random walk neighbor nodes of v_i at time step t, $\mathcal{N}_g^{(t)}(v_i)$ is a negative samples of v_i for time step t, $\mathcal{N}_a^{(t)}(v_i) = \mathcal{N}_b^{(t)}(v_i) \cup \mathcal{N}_g^{(t)}(v_i)$ and w_g, negative sampling ratio, is an adjustable hyper-parameter to balance the positive and negative samples. $\mathcal{L}_c$ is the loss for classifier.

3.6. Algorithmic Complexity

This section briefly discusses the computational complexity of our approach. We assume the network structure is fixed; hence the dimensions of the embedding feature vectors and layers are constant numbers. Therefore, the complexity of one network pass is constant, i.e., $O(1)$. Let us consider the graph $\mathcal{G}^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)})$ at the t-th time step. It can be observed that the overall complexity of our model depends on the structural and temporal attention modules.

Graph Structural Attention Module. From Eqn. (1), the time complexity to compute $\mathbf{h}_{v_i}^t$ depends on the number of attention coefficients α_{ij}^t. Due to the fixed structure assumption, the computation of α_{ij}^t in Eqn. (2) is $O(1)$. Thus, the overall time complexity of the Eqn. (2) is $O(|\mathcal{V}| + |\mathcal{E}|)$.

Graph Temporal Attention Module The time complexity of this module is equivalent to the time complexity of Eqn. (3). We ignore the mask $\mathbf{M}$ in Eqn. (3), the time complexity of Eqn. (3) is dependent on the matrix multiplication between $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$, i.e., $\text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_Z}})\mathbf{V}$. Mathematically, the time complexity of operation $\text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_Z}})\mathbf{V}$ is $O(WD_Z^2 + W^2 D_Z)$ (W is the window time size). However, D_Z is a constant number; hence, the time complexity will be equivalent to $O(W^2)$. Consequently, the time complexity of our network is grown with respect to the size of graph $\mathcal{G}^{(t)}$ and the squared window time size W^2, i.e., $O(|\mathcal{V}| + |\mathcal{E}| + W^2)$. As W can be chosen to be fixed during the operation, our network's overall complexity grows with the number of tracks. Our theory aligns with the empirical results shown in Fig. 3, where the number of nodes can be processed per second (Hz) slightly drops when the number of nodes in the graph increases from 2K to 6K.

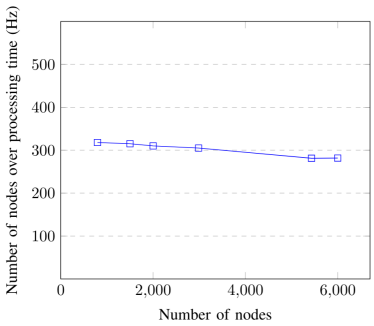

Figure 3. The performance of DyGLIP in terms of Hz w.r.t. number of nodes/tracks.

Datasets	#frames	#box	#ID	#Camera
HDA	75,207	64,028	85	13
SAIVT-Softbio	41,979	64,472	152	8
PRW	11,816	34,304	932	6
MARS	20,715	509,914	625	6
Total	**150K**	**672K**	**1.8K**	**33**

Table 1. Statistics of the joint training set.

4. Experimental Results

In this section, we conduct the experiments to demonstrate the benefits of attention modules, link regression and feature extraction for each nodes in terms of link prediction accuracy, i.e., Accuracy Under Curve (AUC) [14] and ID assignment accuracy, i.e., **1. CLEAR MOT metrics** [4] including MOTA, MOTP, ID Switch (IDS); **2. ID scores** [28] including F1 (IDF1), Precision (IDP), Recall (IDR); and **3. Multi-camera Tracking Accuracy** (MCTA) [9]. We also compare with other state-of-the-art approaches for both overlapping and non-overlapping FOVs on human and car multi-camera tracking datasets.

4.1. Datasets

Experiments conduct on small-scale datasets may provide biased results and may not valid when applying to large-scale datasets. In addition, Duke-MTMC [28], a large-scale datasets, was commonly used in MC-MOT research community is no longer publicly available. Thus, we introduce a large-scale dataset[2] by putting together eight publicly available datasets on Multi-camera settings, four for training and four for testing. Those datasets including **HDA Person** [24], **SAIVT-Softbio** [5], **PRW** [51], **MARS** [34], **PETS09** [12], **CAMPUS** [44], **EPFL** [13]

[2]This dataset will be publicly available.

and **MCT** [9]. Training subsets of HDA Person, SAIVT-Softbio, PRW, and MARS datasets are gathered to form the joint training set. Since our proposed DyGLIP does not require full video frames for training, we can use large-scale ReID datasets, e.g., MARS, and build dynamic graph and node's features based on the provided information for each bounding box, i.e., the ID of the pedestrian, camera, tracklet ID and frame number. We can also train on datasets without continuous frames, e.g., PRW, it still shows the evolution of dynamic graph as tracked subjects change over time. After combining the above datasets, we obtain a training set contains a total of 150K frames, with a total of 1.8K unique ID (see Table 1 for more details). A small subset of this joint training set is used as a validation set for our ablation studies in Subsection 4.3. We use four public benchmark datasets, including PETS09, CAMPUS, EPFL, and MCT, for performance evaluations. We also train and test on car datasets, i.e., CityFlow [37] from AI City Challenge 2020 [25].

4.2. Experimental Setup

We conduct MC-MOT experiments by learning dynamic graph representation with two time steps $\{\mathcal{G}^{(t-2)}, \mathcal{G}^{(t-1)}\}$ to predict/assign ID for new nodes in $\mathcal{G}^{(t)}$ at time step t. Thus, training data are split into mini-batch of a chunk size of 3 and employed a mini-batch gradient descent with the Adam optimizer to learn all the parameters of the attention module and the classifier. The attention module is implemented in Tensorflow [1]. We use two SALs with four heads, each computing 128 features (layer sizes of 512) and two TALs with 16 heads, each computing eight features (output sizes of 128), given the input raw (unattended) feature dimension of 512. The model is trained with a maximum of 100 epochs with a batch size of 512 chunks. Note that we apply some padding to combine those chunks (with a different number of nodes) into a batch for training. We choose the best performing model on the validation set for evaluation on four benchmark datasets.

4.3. Ablation Studies

This section presents several deeper studies to analyze our proposed model and justify our competitive performance presented in the following sections. More specifically, this section aims to demonstrate the following appealing properties of the proposed method: (1) ***Better feature representations***, even in severe changes in lighting conditions between the cameras; and (2) ***Recovery of correct representations for objects*** that lost tracks during camera transitions. In addition, we also conduct several other studies to evaluate the role of link regression and the influence of initial node embedding feature choices.

Better Representations It is well-known in most tracking applications that the data association task's accuracy is dependent mainly on the quality of the underlying feature representations. In other words, one expects that features representing the same object (from different cameras) during its trajectory must form a cluster in the feature space. While other methods use the features produced by MOT trackers, we will show in this section that our dynamic graph with the structural and temporal attention mechanism offers better representations for each node in the graph. We use a subset of frames from the same video for training and validating our method and use the ones in another video for testing. Note that our testing frames cover multiple objects, where each object has multiple transitions over different cameras. The features produced by our method for all the nodes are plotted (using t-SNE embedding [17]) in Fig. 4 (c), while the original node features obtained from MOT trackers are also plotted in Fig. 4 (b). In this figure, features produced by DyGLIP form better clusters than the original features. Furthermore, to quantitatively justify the role of attention, Tables 2 and 3 show the results with and without attention, where the attention mechanism significantly improves the performance. Fig. 5 illustrates how feature embedding of a subject/node change-over-time. Especially when the subject moves from one camera to another camera, the original features (OF) change with a large margin while the transformed features (AF) are quite stable. We measure the average between the cluster centroid and all the corresponding node features up to the time step t.

Recover from Fragmented Tracks in single-camera. We demonstrate the ability to recover from assignment error in the previous steps using cluster of nodes, i.e., a set of nodes that form a connected component in the graph. Fig. 6 illustrates our proposed DyGLIP can recover single-camera

Features	ID F1 (%) ↑	IDP (%) ↑	IDR (%) ↑	IDS ↓
DyGLIP	**56.2**	**59.5**	**56.2**	**44**
− att	39	50.1	36.5	135

Table 2. MOT metrics comparison between DyGLIP (with attention modules) and − att (without attention modules).

Method	Val AUC	Test AUC
Link Regression with Attention	**97.79** %	**97.48** %
Link Regression w/o Attention	84.23 %	87.36 %

Table 3. Accuracy Under the Curve (AUC) comparison between using and without using attention modules.

Metrics	Classifier	Val AUC	Test AUC
Cosine distance	Sigmoid	81.9 %	68.1 %
Hadamard product	SM	**97.79** %	**97.48** %

Table 4. Area Under Curve (AUC) between various feature metrics and link classifiers, i.e., Sigmoid and Softmax (SM).

Features	Val AUC	Test AUC
ReID [53]	**97.79** %	**97.48** %
Detector [15]	73.95 %	66.75 %

Table 5. Area Under Curve (AUC) comparison between various initial features for each node in the proposed graph, i.e., ReID features or features from human detector.

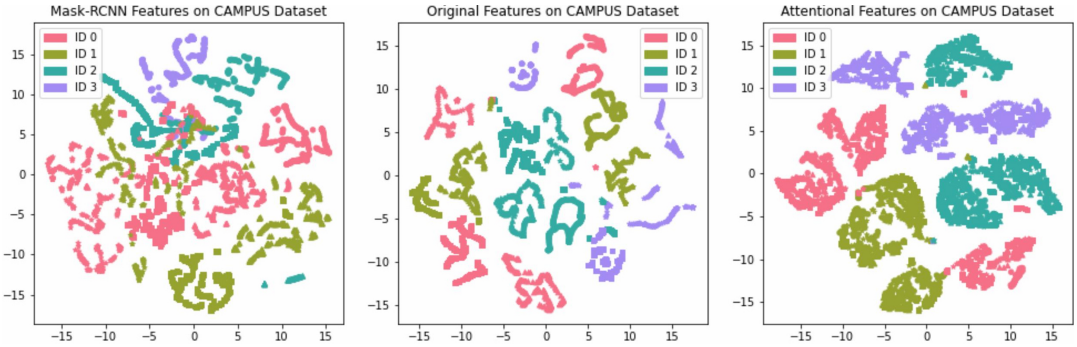

Figure 4. Node embedding in t-SNE space (a) Features from detector [15]. (b) Original features from ReID model [53]. (c) Transformed features with attention. Same color indicating same subject and same symbol indicating same camera. Each node corresponding to a time step in the video sequences. (**Best viewed in color**)

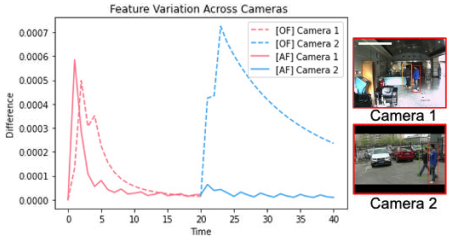

Figure 5. Feature variation across cameras. OF and AF denotes for original features and attentional features, respectively.

Sequence	Method	MOTA (%) ↑	MOTP (%) ↑
S2L1	KSP [2]	80	57
	B&P [30]	72	53
	HCT [44]	89	73
	TRACTA [16]	87.5	79.2
	DyGLIP	**93.5**	**94.7**

Table 6. Evaluation results on PETS09 dataset.

Sequence	Method	MOTA (%) ↑	MOTP (%) ↑
Passageway	KSP [2]	40	57
	HCT [44]	44	71
	TRACTA [16]	52.1	77.5
	DyGLIP	**70.4**	**97.2**
Basketball	KSP [2]	56	54
	HCT [44]	60	68
	TRACTA [16]	64.3	72.5
	DyGLIP	**66.3**	**89.5**

Table 7. Evaluation results on EPFL dataset.

MOT mistake on-the-fly.

Roles of Link Regression. We study the effects of different metrics and classifiers to predict the probability of having an edge connecting two nodes given their embedding features indicating they are in the same ID in Table 4. Our objective function optimizes the inner product that helps the model achieve much better AUC with Hadamard product metric and softmax classifier.

Influence of Initial Node Embedding Features. We study the effects of original embedding features for each tracker/node used as inputs to the attention module in the proposed DyGLIP in Table 5. We compare between using ReID features with the pre-trained model in [53] and features from detector, i.e., Mask-RCNN human detector [15] on human MC-MOT. Note that all other experiments for human MC-MOT (except vehicle MC-MOT), we use these ReID features as inputs to our attention module.

Seq	Method	MOTA ↑	MOTP ↑	MT ↑	ML ↓
Garden 1	HCT [44]	49%	71.9%	31.3%	6.3%
	STP [45]	57%	75%	–	–
	TRACTA [16]	58.5%	74.3%	30.6%	1.6%
	DyGLIP	**71.2%**	**91.6%**	**31.3%**	**0.0%**
Garden 2	HCT [44]	25.8%	71.6%	33.3%	11.1%
	STP [45]	30%	75%	–	–
	TRACTA [16]	35.5%	75.3%	16.9%	11.3%
	DyGLIP	**87.0%**	**98.4%**	**66.67%**	**0.0%**
Auditorium	HCT [44]	20.6%	69.2%	33.3%	11.1%
	STP [45]	24%	72%	–	–
	TRACTA [16]	33.7%	73.1%	37.3%	20.9%
	DyGLIP	**96.7%**	**99.5%**	**95.24%**	**0.0%**
Parkinglot	HCT [44]	24.1%	66.2%	6.7%	26.6%
	STP [45]	28%	68%	–	–
	TRACTA [16]	39.4%	74.9%	15.5%	10.3%
	DyGLIP	**72.8%**	**98.6%**	**26.67%**	**0.0%**

Table 8. Evaluation results on CAMPUS dataset.

4.4. Comparison with State-of-the-arts MC-MOT

To evaluate the proposed method, we first compare with other state-of-the-arts using three datasets PETS09 [12], CAMPUS [44], and EPFL [13] that contain videos with overlapping FOVs. Then, we also compare with other methods that do not require overlapping FOVs among different cameras on MCT [9] dataset. Finally, we perform evaluation on the validation set of CityFlow dataset [37] to compare with the winner of the 4th AI City Challenge 2020 [25]. Although our method focuses on the online association task, our results are comparable with offline approaches. Besides, our DyGLIP method works well on both overlapping and non-overlapping fields of view by treating two types equally and solving the assignment task in a general way.

Overlapping FOVs dataset on Human Tracking We compare with other MC-MOT methods, including K-Shortest Path (KSP) [2], Hierarchical Composition of Tracklet (HCT) [44], Brand-and-Price (B&P) [30] and Spatio-Temporal Parsing (STP) [45] that require overlapping FOVs among different camera views. We also compare with TRACTA [16] on overlapping FOVs videos. Tables 6,

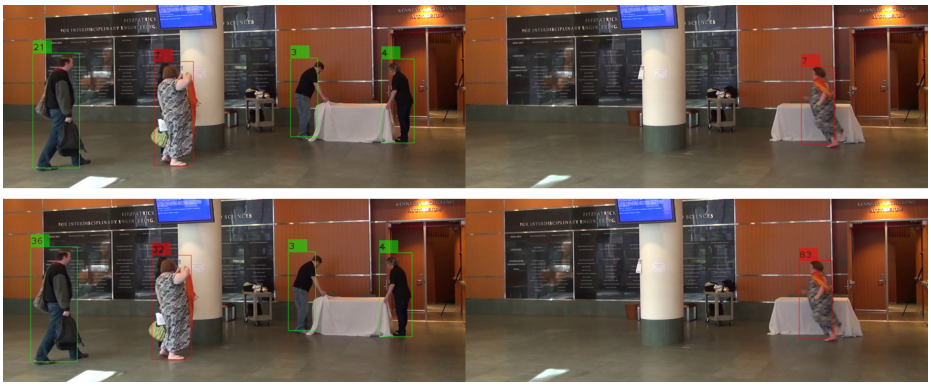

Figure 6. Our proposed method (up) corrects a negative matched case caused by a short-memorized MOT system (down). Determined ID is highlighted by the red bounding box.

Subset	Method	MCTA (%) ↑	MOTA (%) ↑	MOTP (%) ↑	Precision (%) ↑	Recall (%) ↑	IDS ↓
Dataset1	EGM [9]	41.2	59.4	68.0	79.7	59.2	1888
	RAC [6]	59.5	92.6	64.6	69.2	60.6	154
	ICLM [21]	61.2	87.3	68.1	77.2	60.9	112
	TRACTA [16]	70.8	**94.9**	85.2	92.7	**92.6**	71
	DyGLIP	**76.2**	86.7	**97.0**	**93.4**	86.8	**37**
Dataset2	EGM [9]	47.9	67.2	70.6	79.8	63.3	1985
	RAC [6]	62.6	86.8	73.7	69.5	78.4	171
	ICLM [21]	67.7	88.3	76.6	83.3	70.9	123
	TRACTA [16]	83.7	93.4	85.9	95.5	95.4	**60**
	DyGLIP	**91.9**	**95.7**	**96.8**	**97.7**	**95.9**	101
Dataset3	EGM [9]	18.6	27.0	64.7	82.1	53.5	525
	RAC [6]	5.6	9.2	55.3	47.5	66.2	666
	ICLM [21]	37.2	53.2	69.1	66.0	72.6	228
	TRACTA [16]	53.8	58.5	75.4	75.2	91.3	144
	DyGLIP	**89.4**	**92.7**	**96.5**	**98.2**	**93.7**	**122**
Dataset4	EGM [9]	28.4	35.8	71.1	83.6	61.9	3111
	RAC [6]	34.0	53.9	63.0	52.2	79.4	329
	ICLM [21]	54.3	62.5	86.8	87.6	86.0	189
	TRACTA [16]	71.5	79.6	90.0	86.3	**96.0**	**70**
	DyGLIP	**84.7**	**92.5**	**96.6**	**91.3**	92.9	100

Table 9. Evaluation results on MCT dataset.

Subset	Method	MOTA (%) ↑	ID F1 (%) ↑
S02	ELECTRICITY [26]	53.7	53.8
	DyGLIP	**90.9**	**64.9**
S05	ELECTRICITY [26]	74.1	36.4
	DyGLIP	**84.6**	**39.90**

Table 10. Results on CityFlow [37] validation set

7 and 8 show the results on PETS09, CAMPUS, and EPFL datasets, respectively. DyGLIP outperforms all the other methods with up to 63 % on certain metrics.

Non-overlapping FOVs dataset on Human Tracking This experiment compares DyGLIP with other MC-MOT methods, including EGM [9], RAC [6], ICLM [21] and TRACTA [16]. These methods do not require overlapping FOVs between different camera views. DyGLIP significantly outperforms all the methods with a large margin (up to 35%) in most metrics on MCT dataset as in Table 9.

Non-overlapping FOVs dataset on Car/Vehicles Tracking Table 10 shows the results on the AI City challenge validation set. We use the same ReID features as in ELECTRICITY [26]. Indeed, DyGLIP obtains much better results, higher on MOTA and ID F1 in S02 (37.2 % and 11.1%, respectively), in S05 (10.5 % and 3.5%, respectively), thanks to the dynamic graph formulation and the attention module.

5. Conclusion

This paper has re-formulated the MC-MOT problem with a dynamic graph model and treated the global tracklet ID association between multi-camera as link assignment in the proposed graph. The robustness of the proposed dynamic graph is further improved with attention modules that capture structural and temporal variations across multi-camera and multiple time steps. The experiments show significant performance improvements in both human and vehicle tracking datasets in multi-camera with overlapping and non-overlapping FOVs settings.

Acknowledgement

We would like to thank the management team and all other members at VinAI Research for their tremendous support, especially Huy Bui, an AI engineer at VinAI Research, for helping with implementation and experiments.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE TPAMI*, 33(9):1806–1819, 2011.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019.

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[5] Alina Bialkowski, Simon Denman, Sridha Sridharan, Clinton Fookes, and Patrick Lucey. A database for person re-identification in multi-camera surveillance networks. In T Tan and A S Mian, editors, *Proceedings of the 2012 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. Institute of Electrical and Electronic Engineers (IEEE), United States, 2012.

[6] Yinghao Cai and Gerard Medioni. Exploring context information for inter-camera multiple target tracking. In *IEEE Winter Conference on Applications of Computer Vision*, pages 761–768. IEEE, 2014.

[7] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time 'actor-critic' tracking. In *ECCV*, September 2018.

[8] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *CVPR*, June 2020.

[9] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. An equalized global graph model-based approach for multicamera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2016.

[10] W. Chen, L. Cao, X. Chen, and K. Huang. An equalized global graph model-based approach for multicamera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2017.

[11] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*, pages 6172–6181, 2019.

[12] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6, 2009.

[13] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE TPAMI*, 30(2):267–282, 2008.

[14] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[16] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE TIP*, 29:5191–5205, 2020.

[17] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *NeurIPS*, 15:857–864, 2002.

[18] Na Jiang, SiChen Bai, Yue Xu, Chang Xing, Zhong Zhou, and Wei Wu. Online inter-camera trajectory association exploiting person re-identification and camera topology. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1457–1465, New York, NY, USA, 2018. Association for Computing Machinery.

[19] Peng Jinlong, Wang Changan, Wan Fangbin, Wu Yang, Wang Yabiao, Tai Ying, Wang Chengjie, Li Jilin, Huang Feiyue, and Fu Yanwei. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. *ECCV*, 2020.

[20] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *ICCV*, pages 4696–4704, 2015.

[21] Young-Gun Lee, Zheng Tang, and Jenq-Neng Hwang. Online-learning-based human tracking across non-overlapping cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2870–2883, 2017.

[22] Jiahe Li, Xu Gao, and Tingting Jiang. Graph networks for multiple object tracking. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[23] Andrii Maksai and Pascal Fua. Eliminating exposure bias and metric mismatch in multiple object tracking. In *CVPR*, June 2019.

[24] Athira Nambiar, Matteo Taiana, Dario Figueira, Jacinto C Nascimento, and Alexandre Bernardino. A multi-camera video dataset for research on high-definition surveillance. *International Journal of Machine Intelligence and Sensory Signal Processing*, 1(3):267–286, 2014.

[25] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *CVPRW*, page 2665–2674, June 2020.

[26] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *CVPRW*, June 2020.

[27] Liangliang Ren, Jiwen Lu, Zifeng Wang, Qi Tian, and Jie Zhou. Collaborative deep reinforcement learning for multi-object tracking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 605–621, Cham, 2018. Springer International Publishing.

[28] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016.

[29] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, pages 6036–6046, 2018.

[30] B Rosenhahn, G Pons-Moll, and Laura Leal-Taixe. Branch-and-price global optimization for multi-view multi-target tracking. In *CVPR*, pages 1987–1994. IEEE Computer Society, 2012.

[31] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 519–527, 2020.

[32] Sun ShiJie, Akhtar Naveed, Song XiangYu, Song Huan-Sheng, Mian Ajmal, and Shah Mubarak. Simultaneous detection and tracking with motion modelling for multiple object tracking. *ECCV*, 2020.

[33] K. A. Shiva Kumar, K. R. Ramakrishnan, and G. N. Rathna. Distributed person of interest tracking in camera networks. In *Proceedings of the 11th International Conference on Distributed Smart Cameras*, ICDSC 2017, page 131–137, New York, NY, USA, 2017. Association for Computing Machinery.

[34] Springer. *MARS: A Video Benchmark for Large-Scale Person Re-identification*, 2016.

[35] O. Styles, T. Guha, V. Sanchez, and A. Kot. Multi-camera trajectory forecasting: Pedestrian trajectory prediction in a network of cameras. In *CVPRW*, pages 4379–4382, 2020.

[36] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE TPAMI*, 2019.

[37] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *CVPR*, pages 8797–8806, 2019.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[40] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *ECCV*, 2020.

[41] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017.

[42] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

[43] Yihong Xu, Yutong Ban, Xavier Alameda-Pineda, and Radu Horaud. Deepmot: A differentiable framework for training multiple object trackers. *arXiv preprint arXiv:1906.06618*, 2019.

[44] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *CVPR*, pages 4256–4265, 2016.

[45] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *AAAI*, pages 4299–4305, 2017.

[46] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *CVPR*, pages 6787–6796, 2020.

[47] Junbo Yin, Wenguan Wang, Qinghao Meng, Ruigang Yang, and Jianbing Shen. A unified object motion and affinity model for online multi-object tracking. In *CVPR*, 2020.

[48] K. Yoon, Y. Song, and M. Jeon. Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views. *IET Image Processing*, 12(7):1175–1184, 2018.

[49] Quanzeng You and Hao Jiang. Real-time 3d deep multi-camera tracking, 2020.

[50] Zhimeng Zhang, J. Wu, Xuan Zhang, and C. Zhang. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *ArXiv*, abs/1712.09531, 2017.

[51] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, and Qi Tian. Person re-identification in the wild. *arXiv preprint arXiv:1604.02531*, 2016.

[52] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, pages 3652–3661, 2017.

[53] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *ICCV*, 2019.

[54] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020.