Machine Learning
https://doi.org/10.1007/s10994-024-06592-1

LaMMOn: language model combined graph neural network for multi-target multi-camera tracking in online scenarios

Tuan T. Nguyen[1] · **Hoang H. Nguyen**[2] · **Mina Sartipi**[1] · **Marco Fisichella**[2]

Received: 5 December 2023 / Revised: 7 June 2024 / Accepted: 24 June 2024

Abstract
Multi-target multi-camera tracking is crucial to intelligent transportation systems. Numerous recent studies have been undertaken to address this issue. Nevertheless, using the approaches in real-world situations is challenging due to the scarcity of publicly available data and the laborious process of manually annotating the new dataset and creating a tailored rule-based matching system for each camera scenario. To address this issue, we present a novel solution termed LaMMOn, an end-to-end transformer and graph neural network-based multi-camera tracking model. LaMMOn consists of three main modules: (1) Language Model Detection (LMD) for object detection; (2) Language and Graph Model Association module (LGMA) for object tracking and trajectory clustering; (3) Text-to-embedding module (T2E) that overcome the problem of data limitation by synthesizing the object embedding from defined texts. LaMMOn can be run online in real-time scenarios and achieve a competitive result on many datasets, e.g., CityFlow (HOTA 76.46%), I24 (HOTA 25.7%), and TrackCUIP (HOTA 80.94%) with an acceptable FPS (from 12.20 to 13.37) for an online application.

Keywords Mtmct · Multi-camera tracking · Object tracking · Language model

Editor: Myra Spiliopoulou.

✉ Tuan T. Nguyen
xwz778@mocs.utc.edu

Hoang H. Nguyen
ehoang@l3s.de

Mina Sartipi
mina-sartipi@utc.edu

Marco Fisichella
mfisichella@l3s.de

[1] Center for Urban Informatics and Progress (CUIP), The University of Tennessee at Chattanooga, Chattanooga, TN, USA

[2] L3S Research Center, Leibniz University Hannover, Hannover, Lower Saxony, Germany

Published online: 15 July 2024

1 Introduction

Recent advancements in computer vision have greatly improved the efficiency of traffic control at the city level by enabling accurate prediction and analysis of high volumes of traffic. Including a vehicle tracking application is a crucial element in implementing intelligent traffic management systems. A vehicle tracking application combines the vehicle's spatial, temporal, and visual data to generate its trajectory. It can be utilized to monitor the path of cars within the urban area and ascertain their velocity and travel duration to enhance traffic efficiency. Multi-Target Multi-Camera Tracking (MTMCT) is a significant application in this domain. The objective of MTMCT is to generate a comprehensive global trajectory of a vehicle by extracting its trajectory from cameras positioned at various locations across a significant area. Tracking-by-detection is a fundamental paradigm in MTMCT, which consists of the following three components: *(i) object detection*, *(ii) multi-object tracking in single cameras (MOT)*, and *(iii) trajectory clustering*. Object detection involves identifying objects as bounding boxes within video frames. Multiple Object Tracking (MOT) subsequently monitors the object's motion in a single camera by matching bounding boxes of consecutive frames and generating tracklets. Ultimately, Trajectory clustering generates a global object activity map by combining tracklets from multiple cameras. In online MTMCT, the tracking task can function by linking objects using only past frames, while in offline MTMCT, it can operate by contemplating future and past frames as well. The tracking-by-detection paradigm has garnered considerable recognition and has demonstrated encouraging outcomes (Yao et al., 2022; Nguyen et al., 2023).

However, despite its fascinating design and impressive performance, the use of tracking-by-detection schema in real-world scenarios presents several challenges: (1) Non-generative: it is necessary to develop a novel matching rule (spatio-temporal) for every new camera scenario; (2) Data limitation: To the best of the authors' knowledge, the public MTMCT dataset is currently restricted, with only the Cityflow dataset (Tang et al., 2019) available; (3) High cost of the manual labeling: The fine-tuning procedure requires extensive time and effort to label the dataset manually. In response to the challenges above, we propose a generative end-to-end transformer-based MTMCT model called LaMMOn. Because it is an end-to-end model, LaMMOn is easily applied to different camera scenarios without the need to build new matching rules. Furthermore, LaMMOn addresses the issue of data limitation and the high cost of manual labeling by synthesizing object embeddings from text and utilizing these synthesized embeddings to fine-tune new datasets. The LaMMOn 's architecture is visualized in Fig. 1.

In general, LaMMOn contains three modules: (1) Language Model Detection module (LMD) is responsible for performing the *(i) object detection* task and generating object embedding; (2) Language and Graph Model Association module (LGMA) handles the tasks of *(ii) multi-object tracking in single camera (MOT)*, and *(iii) trajectory clustering* simultaneously; and (3) Text-to-embedding module (T2E) to synthesize the object embedding from texts which identify object feature such as: car type, car color and location. Fig. 1 presents the overview of LaMMOn architecture and Sect. 3 clarifies the methodology details.

The primary contributions of our paper are highlighted below:

- We propose a generative end-to-end MTMCT method called LaMMOn that effectively adapts to diverse traffic video datasets without the need for manual rule-based matching or manual labeling.

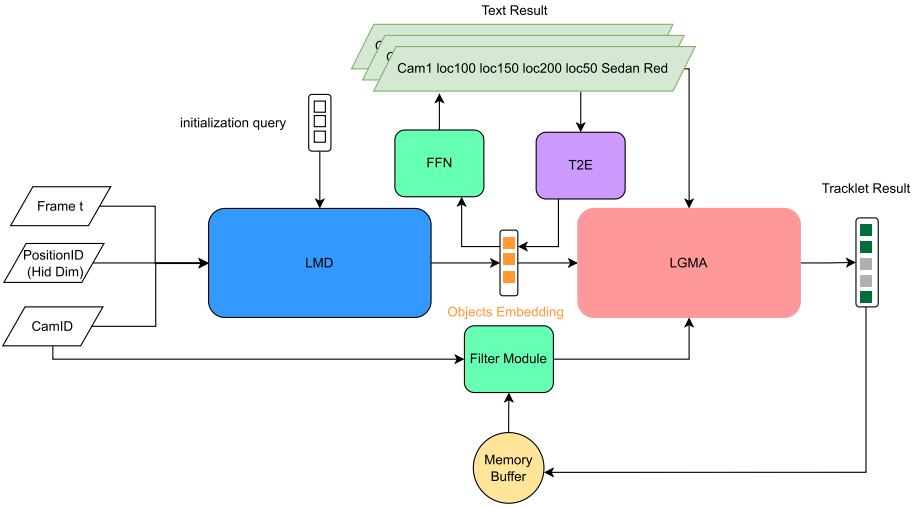

Fig. 1 Overview of the LAMMON architecture

- We propose a T2E module that can synthesize object embedding from text, solving the data limitation problem of the MTMCT problem.
- We propose the LGMA module to address the tasks of *(ii) multi-object tracking in single cameras (MOT)*, and *(iii) trajectory clustering* simultaneously. LGMA integrates both language and graph models to enhance the performance of the association. It introduces a novel perspective on the issue of object detection.
- Our online MTMCT application achieves a competitive result on many datasets: CityFlow (Tang et al., 2019) (HOTA 76.46%, IDF1 78.83%), I24 (Gloudemans et al., 2024) (HOTA 25.7%) and TrackCUIP (Harris et al., 2019) (HOTA 80.94%, IDF1 81.83%) with an acceptable FPS (from 12.20 to 13.37) for an online application.

The rest of the paper is structured as follows: related work is explained in Sects. 2, 3 presents our proposed architecture and methodology; Sect. 4 presents our datasets, evaluation metrics, experimental setups, results, and some ablation studies; Sect. 5 concludes with future work.

2 Related work

2.1 Multi-object tracking (MOT)

Multi-Object Tracking (MOT) is the process of associating objects observed in video frames from a single camera and creating tracklets for the detected objects. Multiple efficient techniques have been proposed for MOT, for example: Tracktor (Bergmann et al., 2019) offers proposals to a detector in the form of tracks and directly transfers the tracking ID. The object association strategy suggested in the CenterTrack algorithm (Zhou et al., 2020) relies on comparing the predicted positions of objects and their matching detections within established tracks. The TransCenter framework, as introduced in Xu et al. (2021), enhances the functionalities of CenterTrack by including the deformable DETR (Zhu et al.,

2020). The recent work from Hassan et al. (2023) combines the Siamese network and Deepsort to extract the features for the tracking.

2.2 Multi-target multi-camera tracking (MTMCT)

Multi-Target Multi-Camera Tracking (MTMCT) aims to build a global tracklet that tracks the object movement in multiple cameras. To our knowledge, all previous techniques have treated the MTMCT problem as a subsequent stage of the MOT problem. Typically, it is seen as an issue of clustering, where the input results from trajectories obtained from the MOT problem. In a prior study (Yao et al., 2022), the clustering step has incorporated spatial-temporal filtering and traffic laws. By imposing these constraints, the scope of the search is significantly narrowed, resulting in a substantial enhancement in the accuracy of vehicle re-identification. Using the identical camera distribution for both test data and training data, techniques (Ullah & Alaya Cheikh, 2018) acquire the transition time distribution for each pair of adjacent cameras without manual adjustment. The paper (Tesfaye et al., 2017) presents various methods for MTMCT with non-overlapping views.

Unlike previous MTMCT approaches, LAMMON is an end-to-end model that simultaneously performs the tasks of detection and association for multiple cameras at the same time. The methodology is described in detail in Sect. 3.

2.3 Transformer in tracking

Transformer-based models are recently being extensively utilized in several domains that require image processing approaches (Sun et al., 2020; Ghaffar Nia et al., 2023; Chohan et al., 2023). In the MTMCT problem, many studies have been employing the transformer model to improve performance. Trackformer (Meinhardt et al., 2022) enhances the DETR model by incorporating extra object inquiries from existing tracks and propagating track IDs, similar to the approach used in Tracktor (Bergmann et al., 2019). TransTrack (Sun et al., 2020) employs past track information as queries and establishes associations between objects using updated bounding box locations. Furthermore, MO3TR (Zhu et al., 2022) incorporates a temporal attention module to modify the state of each track within a specific time frame. It then utilizes the updated track characteristics as queries in DETR. The underlying concept of these works involves utilizing the object query method in DETR (Carion et al., 2020) to progressively expand existing tracks on a frame-by-frame basis. Our utilization of transformers differs. The transformer-based detector, known as LMD, uses queries to identify many objects as bounding boxes simultaneously. Next, we employ an additional transformer-based module called LGMA to group the previously discovered boxes into global trajectories.

2.4 Graph neural network in tracking

Utilizing neural networks to handle data with a graph structure was the initial application of GNNs (Gori et al., 2005). The core idea is to design a graph with interconnected nodes and edges and to update node/edge properties based on these interconnections. In recent years, various GNNs (e.g., GraphConv, GCN, GAT, GGSNN) have been proposed, each with a distinct feature aggregation rule that has been demonstrated to be effective on a variety of transportation tasks (Weng et al., 2020; Kumarasamy et al., 2023; Li et al., 2020;

Khaleghian et al., 2023; Duan et al., 2019). Specifically, in GNN3DMOT (Weng et al., 2020), the authors design an unweighted graph in which each node represents an object feature at a particular frame, and each edge between two nodes at different frames represents the matching between detections. Graph-based methods in Duan et al. (2019) establish a global graph for multiple tracklets in different cameras and optimize for an MTMCT solution. Recently, the works in Nguyen et al. (2023, 2023a) built tracklet features in graph structures and used graph similarity to cluster the single-camera tracklet.

3 Methodology

LaMMOn can be partitioned into three modules: (1) The Language Model Detection module (LMD) performs the task of detecting objects and generating object embeddings; (2) The Language and Graph Model Association module (LGMA) handles multi-object tracking in single cameras and trajectory clustering at the same time; (3) The Text-to-embedding module (T2E) synthesizes object embeddings from texts, identifying object features such as car type, car color, and location. The general architecture is depicted in Fig. 1. To begin with, the video frame input is combined with Positional ID embedding and Camera ID embedding. Subsequently, the LMD accepts the concatenated embedding as input and generates the proposal object embedding for objects in each frame. The LGMA module then utilizes the object embedding and information from the memory buffer and filter module to produce the global tracklet. In addition, we use the synthesizer module to enhance the embedding of the synthesized proposal, addressing the challenge of limited data and ultimately improving the final outcome. In the subsequent part, we introduce the preliminaries and meticulously examine the intricacies of each module.

3.1 Preliminaries

In this section, we will formally define object detection, tracking, tracklet, and tracking schema.

Object detection. Consider a picture denoted by I. Object detection aims to accurately recognize and precisely determine the location of all interested objects in image I. An object detection module receives image I as input and generates a collection of objects $\{o_i\}$ with their respective locations $\{b_i\}, b_i \in \mathbb{R}^4$ as output. If the objects are of several classes, the detector will generate a classification score $s_i \in \mathbb{R}^C$ for a predetermined set of classes C. Our model focuses on only the object *car*, and then the classes C represent several automobile types, such as SUV, Sedan, or simply truck. In addition, our model generates a classification score for the color of the car and the camera ID. To summarize, our object detector produces the following outputs: $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$, where c represents the camera ID, $\{b_1, b_2, b_3, b_4\}$ represent the location of the automobile, and $\{s_1, s_2\}$ represent the kind and color of the car.

Tracking and Tracklet. Consider a sequence of images denoted by $I^1, I^2, ..., I^T$. An MTMCT model aims to identify and trace the tracklet $\tau_1, \tau_2, .., \tau_k$ of all objects within a certain period. Each tracklet, denoted as $\tau_k = \{\tau_k^1, \tau_k^1, ..., \tau_k^T\}$, represents a sequence of object locations and classification scores $\tau_k^T = \{c, b_1, b_2, b_3, b_4, s_1, s_2\}$ for a particular object over each frame.

Tracking schema. This study decomposes the tracking problem into per-frame object detection and multi-camera inter-frame object association. Specifically, LMD handles

per-frame object detection, and LGMA is responsible for multi-camera inter-frame object association. In per-frame object detection, LMD first finds N_t candidate objects $\{o_1^t, o_2^t, ...o_{N_t}^t\}$ as a set of location and classification scores $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$. Then, in multi-camera inter-frame object association, LGMA links current detected objects $\{o_1^t, o_2^t, ...o_{N_t}^t\}$ to existing tracklets $\tau_1, \tau_2, .., \tau_k$ and updates their status. Previous studies often established the association by considering pairwise matches between objects in consecutive frames (Bewley et al., 2016; Zhou et al., 2020) or by employing an optimization for global association (Frossard and Urtasun 2018; Brasó and Leal-Taixé 2020). Recently, GTR (Zhou et al., 2022) achieved single-pass joint detection and association in an end-to-end fashion. However, GTR only tracks the target in a single camera (video). Following this motivation, our model can carry out the end-to-end joint detection and association across several cameras in a synchronized manner. All video cameras are streamed and simultaneously perform object detection and object association in a single forward pass through the network.

3.2 Language model detection (LMD)

The architecture of LMD is depicted in Fig. 2. The LMD functions as a per-frame object detector, generating a collection of detections denoted as $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$. In this collection, the variable c represents the camera ID, while $\{b_1, b_2, b_3, b_4\}$ corresponds to the location of the automobile. Additionally, $\{s_1, s_2\}$ represent the type and color of the car, respectively. Following the concepts presented in pix2seq (Chen et al., 2021), we consider the representation of car location prediction as discrete tokens. Specifically, {x,y,w,h} in normal object detection are normalized to a number of bins (in this study, we set the number of bins as 1000) to produce $\{b_1, b_2, b_3, b_4\}$. The methodology employed in this study to generate these tokens is based on the architecture of Deformable DETR (Zhu et al., 2020) with encoder and decoder layers. Firstly, the video frame image is passed through a ResNet50 layer, after which it is integrated with both a Positional_ID encoder and a Camera_ID encoder. Subsequently, the aggregated feature map is sent toward the encoder and decoder layers of Deformable DETR (Zhu et al., 2020), generating the object embeddings. The object embeddings are fed into a feed-forward network (FFN) to generate the classification scores $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$.

Loss function. By the methodology proposed in the Deformable DETR (Zhu et al., 2020), our approach incorporates three distinct loss functions: cross-entropy loss, bounding

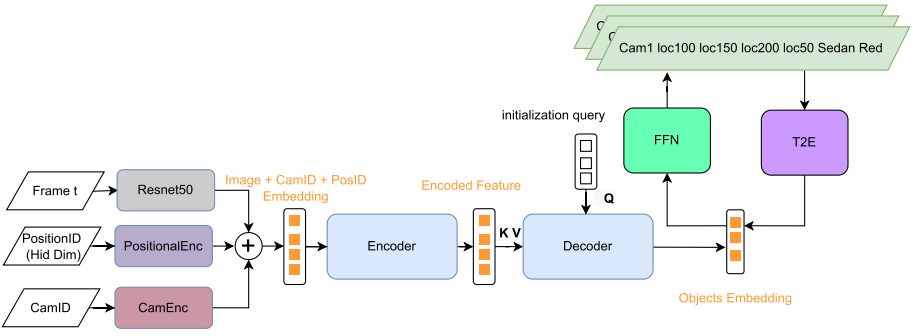

Fig. 2 LMD module architecture

box f1 loss, and bounding box IOU loss. It should be noted that the cross-entropy loss impacts both the bounding box F1 loss and the bounding box IOU loss. This is because the cross entropy loss incorporates the cross entropy loss of the four location predictions, which can also be used to calculate the bounding box F1 loss and the bounding box IOU loss. Specifically, the cross-entropy loss is computed as the sum of seven cross-entropy, each of which is weighted. In summary, the loss is calculated as follows:

$$loss = \alpha_1 * ce + \alpha_2 * bb\_f1 + \alpha_3 * bb\_IOU \tag{1}$$

where the set $\{ce, bb\_f1, bb\_IOU\}$ represents three types of loss functions, namely cross-entropy loss, bounding box f1 loss, and bounding box IOU loss. The weights of these losses are denoted as $\alpha_1, \alpha_2, \alpha_3$.

Next, the variable *ce* is computed using the following formula:

$$ce = \beta_1 * c + \beta_2 * b_1 + \beta_3 * b_2 + \beta_4 * b_3 + \beta_5 * b_4 + \beta_6 * s_1 + \beta_7 * s_2 \tag{2}$$

where $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$ are cross entropy of the predictions of camera ID $\{c\}$, car location $\{b_1, b_2, b_3, b_4\}$, car's type and car's color $\{s_1, s_2\}$. And $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7\}$ are their weights.

Camera ID Encoder (CamEnc). Different from previous approaches that only consider camera IDs as learnable parameters or fixed one-hot vectors, our framework encodes camera features through the graph structure between adjacent cameras in a local area or route. Specifically, we build small subgraphs for each group of neighboring cameras, which are part of the global graph containing all the neighbor relationships of cameras in the datasets. We use state-of-the-art graph neural networks such as GCN (Welling and Kipf 2017), GIN (Xu et al. 2019), and GAT (Veličković et al. 2018) to extract node embeddings on the constructed global graph. The embeddings are used as camera features before aggregation with the *positional* encoding of the position ID and *Resnet50* encoding on each frame t of the LMD component. This approach allows camera embeddings to represent not only the camera IDs' information like previous methods but also capture the relationship structure between cameras in geographical space, which graph structure is visualized in Fig. 3. A comprehensive analysis is carried out in the Appendix to examine the performance of CamEnc in complete graphs (Scenarios 1, 2, 3, and 4-5) and a path graph (Scenario 6).

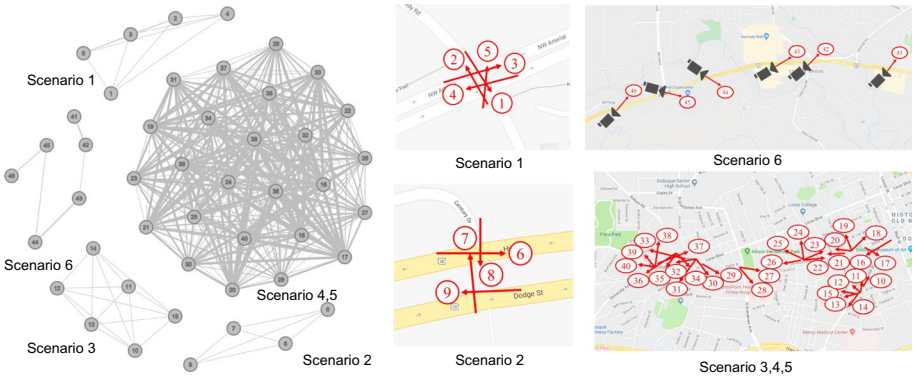

Fig. 3 Camera graph based on geographical location of different scenarios. The red arrows indicate the location and direction of the cameras.

3.3 Language and graph model association (LGMA)

The architecture of LGMA is shown in Fig. 4. LGMA conducts the multi-camera inter-frame object association task in an end-to-end fashion. The module takes as input objects embedding from LMD and existing tracklets $\tau_1, \tau_2, .., \tau_k$ from the Memory Buffer. It then links these embedded objects in related tracklets via Graph-Based Token Features to determine their updated status in the current frames. Finally, it updates the status of all the existing tracklets $\tau_1, \tau_2, .., \tau_k$ in the Memory Buffer.

Graph-Based Token Feature Construction. We create a graph based on token features to leverage the association information in token embeddings generated from LMD, with the nodes representing feature vectors (token embeddings) and the edges representing their Euclidean distance, which graph generating process presented in Fig. 5. We keep only the edges greater than the threshold value τ. The decision to set the distance threshold τ to 0.5 was determined by an empirical tuning proposed in the previous approaches (Fisichella, 2022; Nguyen et al., 2023). Similar to LMD, in LGMA architecture, graph neural networks such as GCN (Welling and Kipf 2017), GIN (Xu et al. 2019), and GAT (Veličković et al. 2018) are also used to extract node embedding features from these token feature graphs. These node embeddings are then aggregated with the object embeddings generated by LMD. This combination ensures that final embeddings not only contain token information representing tracklet features but also represent the correlation between tokens in the vector space through combining node embeddings of the token feature graphs. Furthermore, GNN functions as a tool to refine the acquired T2E weights. Due to the T2E being trained independently, its weight remains fixed throughout the training of the LaMMOn model. Subsequently, it is necessary to perform a fine-tuning using a GNN to augment the module's adaptability. Besides, since the generated graphs maintain small structures with less than 50 nodes and edges, the computational time to get node embeddings is almost negligible. Thus, this approach can improve the overall performance of the proposed architectures (see Sect. 4) but still guarantees the

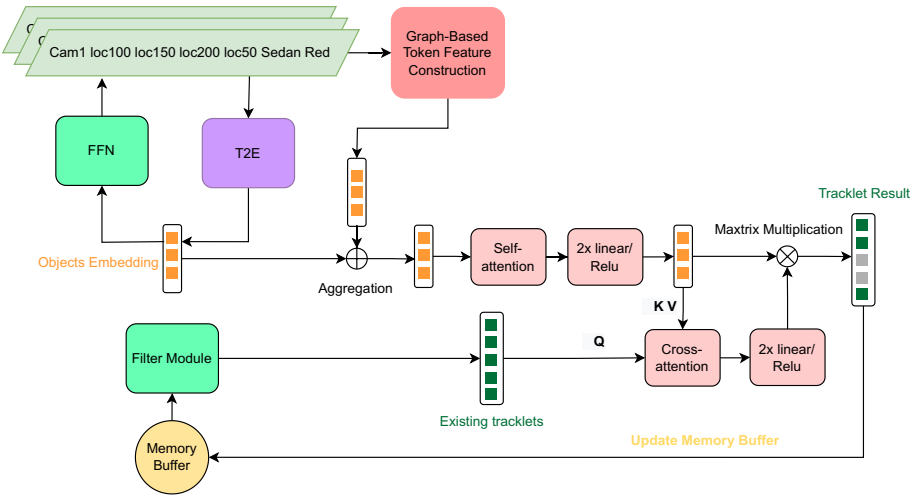

Fig. 4 LGMA module architecture. The input for LGMA is object embedding, and the output is the tracklet result. The component of LGMA is marked pink. Hence, the FFN, T2E, Filter modules do not belong to LGMA (Color figure online)

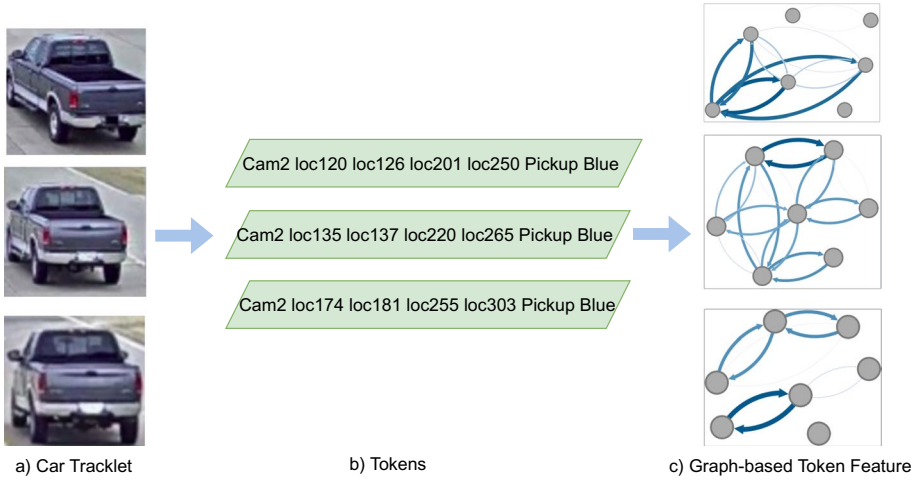

Fig. 5 Graph-based token feature visualization: **a** A tracklet consisting of three bounding boxes collected from various video frames; **b** Tokens generated by LMD from the input tracklet; **c** LGMA-constructed graphs in which nodes are token features (e.g., tokens of the first bbox are: Cam2, loc120, loc126, loc201, loc250, Pickup, and Blue) and edges represent the Euclidean distance between nodes, with the darker the edges, the greater the distance (Color figure online)

model's prediction and operability for online scenarios, which always require real-time processing.

Global Association. Adopting the design proposed by GTR (Zhou et al., 2022), LGMA carries out the task of associating multiple frames simultaneously. The object embedding is regarded as the input for the encoder, while the existing tracklets are regarded as the input for the decoder. More precisely, the object embedding is passed through self-attention layers and then through two linear/ReLU layers. Then, it proceeds to cross-attention layers as key and value (K, V).

On the other hand, the existing tracklet is used as the query (Q) input for the cross-attention layers. In the end, matrix multiplication is performed to compute the association score between the object embedding and the existing tracklets. The encoder-decoder design bears a resemblance to GTR (Zhou et al., 2022); however, based on our ablation study shown in Sect. 4, both the encoder and decoder consist of 2 layers (GTR set number of layers for both encoder and decoder as 1). The other parameter remains consistent with the original GTR model (Zhou et al., 2022). The process is depicted in Fig. 4.

Memory Buffer and Filter Module. The memory buffer stores the object embeddings of all existing tracklets. The stored embedding is subsequently utilized to construct the tracklet representation, which serves as the query input for the decoder in the global association. Various methods can be employed to propose a presentation for a tracklet, including the most recent embedding, mean, logarithmic mean, or attention-based approaches (Cai et al., 2022). An ablation study is conducted in Sect. 4. To ensure a straightforward and efficient model, we utilize the average of the five most recent embeddings as the representation for the tracklet. However, employing the representation of every tracklet that currently exists may be imprudent and less efficient. The filter module is then designed to receive the Camera ID c of the current inputs and selectively choose just the representation of the tracklet in the adjacent camera as the query input for the decoder in the global association process.

3.4 Text to encoder (T2E)

T2E is specifically developed to address the problem of data limitations and the exorbitant expenses associated with human labeling. The primary concept is that instead of synthesizing video, we instruct an encoder (T2E) to generate the synthesized object embeddings using defined text tokens representing the object features $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$. Specifically, the Sentencepiece encoder (Kudo & Richardson, 2018) is employed as the main architecture, inspired by the Unified-IO approach (Lu et al., 2022). The T2E training is performed once, independently from LaMMOn. The input consists of text that identifies the object features $\{c, b_1, b_2, b_3, b_4, s_1, s_2\}$, whereas the training objective/ output is the object embedding obtained by LMD. Subsequently, the parameter of T2E remains unchanged when utilized to generate synthesized object embeddings. The synthetic object embeddings serve the same function as the real object embeddings obtained from the LMD result using real data (videos). More precisely, these synthetic representations are then employed to train the LGMA module (with frozen LMD parameters) The efficacy of T2E is showcased in Sect. 4 where we must assess LaMMOn 's performance in the test set (Scenario 6) of the CityFlow dataset while no label of this test set is given. So, we employ the T2E module to generate object embeddings for Scenario 6 and utilize them to fine-tune the LGMA module. We conduct an additional experiment in the appendix to demonstrate the efficacy of T2E in managing novel scenarios derived from other datasets. In addition, we provide a comprehensive and systematic explanation of how to use the T2E model in a novel scenario in the appendix. Furthermore, we provide an extensive evaluation to assess the similarity between real tokens and synthetic tokens in the appendix.

4 Experiment

We publicize the datasets and our LaMMOn model at https://github.com/elituan/lammon.

4.1 Dataset, implementation details and evaluation metric

We assess our technique by conducting experiments on three MTMCT tracking datasets: CityFlow (Tang et al., 2019), I24 (Gloudemans et al., 2024), and TrackCUIP (Harris et al., 2019).

The CityFlow dataset covers different types of streets, including intersections, highways, and road extensions. It comprises 3.25 h of traffic videos captured from 40 cameras at 10 intersections. The CityFlow test set consists of 20 min of street video from six cameras situated at six intersections. We extract the car's kind and color for CityFlow using the label from CityFlow-NL dataset (Feng et al., 2021).

The I24 dataset comprises 234 h of video recordings captured simultaneously from 234 overlapping HD cameras along a 4.2-mile section of an 8-10 lane interstate highway close to Nashville, TN, US. We utilize the ImageNet pre-trained Res2Net (Gao et al., 2019) model to extract the vehicle color for the I24 dataset.

The TrackCUIP is a private dataset carried out under the TestBed CUIP environment (Harris et al., 2019). The dataset comprises one-hour videos of traffic recordings

captured by four cameras positioned at four different crossroads. The TrackCUIP dataset utilizes 30 min of videos for training, with 10 min allocated for validation and 20 min assigned as the test set.

We train our models for a total of 65 epochs. The details of the parameters are discussed in Sect. 4. The training experiments use four Nvidia Tesla V100s with 32GB of memory each, whereas the inference experiments use one Nvidia Tesla V100 with 32GB of memory.

To assess our model's performance, we employ the **IDF1** (Ristani et al., 2016) and **HOTA** (Higher Order Tracking Accuracy) (Luiten et al. 2021) metrics. The detailed formula and explanation of these two metrics are provided in the Appendix.

4.2 Baselines

Due to the limited availability of public approaches for online MTMCT, we have chosen three baseline methods, namely **TADAM** (Guo et al. 2021), **BLSTM-MTP** (Kim et al. 2021), and **GraphBased Tracklet** (Nguyen et al. 2023), to showcase the effectiveness of our LAMMON model. **TADAM** is a model that combines position prediction and embedding association synergistically. To be more precise, the prediction process involves utilizing attentional modules to allocate greater focus towards targets and minimize attention towards distractors. These dependable embeddings can enhance the experience of identity awareness by aggregating memories. For **BLSTM-MTP**, their primary objective is to address the issue of efficiently considering all tracks in memory updating while minimizing spatial overhead. They achieve this by implementing a unique multi-track pooling module. Regarding **GraphBased Tracklet**, it is constructed by representing the tracklet feature as a graph structure and employing graph similarity scores to match tracklets captured by multiple cameras.

In addition, for the I24 dataset, we reuse the baselines' results from the original dataset paper (Gloudemans et al. 2024) including SORT, IOU, KIOU, ByteTrack (L2), ByteTrack (IOU).

4.3 Experimental results

In this section, we assess our methodology using three datasets: CityFlow (Tang et al. 2019), I24 (Gloudemans et al. 2024), and TrackCUIP (Harris et al. 2019), which were described in Sect. 4.1.

4.3.1 CityFlow dataset

We conduct multiple ablation studies on the CityFlow Dataset to optimize the parameters for LAMMON. Due to the page limit, we only present an ablation study for tuning parameters of increasing FPS and the T2E module; more details are given in the Appendix. The best IDF1 after tuning parameters of LMD and LGMA is 77.32%. However, the FPS is only 6.3, which is insufficient for an online MTMCT application. To increase the FPS, we conducted an ablation study with three parameters:

- **num_lay_LMD**: The number of layers is used in the encoder and decoder of the LMD module. The current value is 6.
- **GNN_dim**: It is the dimension of all GNN layers, with a value of 256.

Table 1 Tuning parameters of increasing FPS and the T2E module on the CityFlow Dataset

Tuning	num_lay_LMD	hid_dim	GNN_dim	T2E_min	IDF1	FPS
num_lay_LMD	6	256	256	No	77.32%	6.3
	4	256	256	No	**76.25%**	**7.5**
	3	256	256	No	73.38%	9
	2	256	256	No	68.43%	11.1
hid_dim	4	256	256	No	76.25%	7.5
	4	**128**	256	No	**75.88%**	**10.7**
	4	64	256	No	70.21%	11.2
GNN_dim	4	128	256	No	75.88%	10.7
	4	128	**128**	No	**74.12%**	**12.2**
	4	128	64	No	70.26%	12.7
T2E	4	128	128	No	74.12%	12.2
	4	128	128	2	74.36%	12.2
	4	128	128	4	75.23%	12.2
	4	128	128	8	76.94%	12.2
	4	128	128	**16**	**78.83%**	**12.2**
	4	128	128	32	78.51%	12.2

The boldfaced are the selected options

Table 2 Tracking result of LaMMOn and other online MTMCT methods on CityFlow dataset

	Method	FPS	IDF1	HOTA
Baselines	TADAM Guo et al. (2021)	13.30	53.47	51.36
	BLSTM-MTP Kim et al. (2021)	8.55	61.67	58.21
	GraphBased Tracklet Nguyen et al. (2023)	**14.03**	75.21	73.38
Ours	LaMMOn	12.2	**78.83**	**76.46**

Bold values indicate the models or parameter settings that achieve the best performance

- **hid_dim**: It is the LaMMOn hidden dimension, with a value of 256.

We carefully analyze the trade-off between inference speed frames per second (FPS) and IDF1 scores, then choose the most favorable alternatives. The outcome is illustrated in Table 1. We have achieved an FPS of 12.2 and an IDF1 score of 74.12%. This indicates a trade-off between a 6 FPS increase and a 3.2% decrease in IDF1. Furthermore, we assess the efficacy of the T2E module using the T2E_min parameter, which denotes the duration of the video that produces the same amount of object embeddings as synthetic object embeddings. It is assumed that there are 15 cars every minute, with each car appearing on camera for 45 s. These synthetic object embeddings are subsequently used for fine-tuning LGMA in 15 epochs. The data shown in Table 1 demonstrates a substantial increase in the IDF1 score as the length of the synthetic video increases. More precisely, the IDF1 metric shows a 4.7% improvement when 16 min of synthetic video are used.

Fig. 6 Visualization of tracking result in CityFlow dataset

Table 3 Tracking result of LaMMOn and other methods on I24 Dataset

	HOTA	Recall
SORT	9.5	73.6
IOU	1.1	20.4
KIOU	8.5	73.9
ByteTrack (L2)	9.5	73.6
ByteTrack (IOU)	8.5	75.9
TADAM	12.7	73.6
BLSTM-MTP	14.3	73.5
GraphBased Tracklet	20.2	76.9
LaMMOn - Ours	**25.7**	**79.4**

Bold values indicate the models or parameter settings that achieve the best performance

Finally, the outcome of LaMMOn is displayed in Table 2, alongside other online MTMCT baselines for comparison. LaMMOn surpasses existing methods and gets the highest outcome, with an IDF1 score of 78.83%, a HOTA score of 76.46%, and an FPS of 12.2. Our model also has the capability to attain higher accuracy to compete with the offline-scenario-only existing MTMCT models (Shim et al., 2021; Hou et al., 2019; Specker et al., 2021). Nevertheless, a compromise exists between the IDF1 score and the inference speed FPS, implying that the most effective models may be excessively sluggish for an online MTMCT application. In addition, a visualization of tracking results in the CityFlow dataset is shown in Fig. 6.

4.3.2 I24 Dataset

We use the identical parameters to train the I24 dataset as we did for the CityFlow dataset. The baseline results in Table 3 are reused from the original I24 dataset paper (Gloudemans

et al. 2024), excluding the TADAM (Guo et al. 2021), BLSTM-MTP (Kim et al. 2021) and GraphBased Tracklet Nguyen et al. (2023) models. In this experiment, we also used the same train/validation/test sets as described in the original papers (Gloudemans et al., 2024, 2023) to ensure the consistency of the reported results. Table 3 shows that our model outperforms the baselines from the I24 dataset in both HOTA and Recall metrics. Besides, the baseline models, as mentioned in the original paper, use a given detection component, while our proposed approach is an end-to-end model that combines detection and association in one complete framework. With a significant increase of 5.5% and 2.5% in HOTA and Recall, respectively, the end-to-end architecture of our proposed model has shown more potential than the state-of-the-art approaches in the real-time MTMCT tasks. It is important to understand that the I24 dataset is extremely large, consisting of 234 h of video, which is 72 times larger than the CityFlow dataset. Due to its size, the ground truth labels have not been completely assigned manually. More precisely, a portion of the data is labeled manually, and the GPS data from 270 vehicles is utilized to establish a matching rule for the manually labeled data. Subsequently, the ground truth label is generated by using the matching rule given above and making manual corrections. As stated in the original paper, the maximum theoretical performance is HOTA 53.1%.

4.3.3 TrackCUIP dataset

For the TrackCUIP dataset, we conduct the training using the same sets of parameters for the CityFlow dataset. Table 4 presents the performance of LaMMOn and other baselines on the TrackCUIP dataset. The results show that LaMMOn outperforms all other baselines, with an increase of 4.42% and 2.82% in IDF1 and HOTA, respectively, while keeping the FPS at an acceptable rate for an online algorithm.

5 Conclusion and future work

We present an innovative solution for MTMCT application with an end-to-end multi-camera tracking model based on transformers and graph neural networks, called LaMMOn. Our model overcomes the limitations of the tracking-by-detection paradigm by introducing a generative approach that enables adaptation to new traffic videos by reducing the need for manual labeling. The synthesis of object embeddings from text descriptions, as demonstrated by our Language Model Detection (LMD) and Text-to-embedding (T2E) modules, significantly reduces the data labeling effort and improves the model's applicability in different scenarios. In addition, our trajectory clustering method incorporating the Language

Table 4 Tracking result of LaMMOn and other methods on TrackCUIP dataset

	FPS	IDF1	HOTA
TADAM	13.51	56.55	58.24
BLSTM-MTP	9.72	65.36	64.43
GraphBased Tracklet	**14.53**	77.41	78.12
LaMMOn - Ours	13.37	**81.83**	**80.94**

Bold values indicate the models or parameter settings that achieve the best performance

and Graph Model Association (LGMA) demonstrates the efficiency of using synthetic embeddings for tracklet generation. This approach overcomes the data limitations of multi-camera tracking and ensures adaptability to different traffic scenarios. Finally, LAMMON demonstrates real-time online capabilities and achieves competitive performance on many datasets, such as CityFlow (HOTA 76.46%, IDF1 78.83%), I24 (HOTA 25.7%) and Track-CUIP (HOTA 80.94%, IDF1 81.83%).

In the future, we aim to improve the robustness of the model further by exploring additional language-based graph features and extending its applicability to different datasets. One possible direction is to delve deeper into optimizing the building of graph structures in extracting camera ID encodings. The success of our model is a significant step towards overcoming the challenges in real-world MTMCT applications and promises improved efficiency and scalability in intelligent transportation systems.

Appendix A evaluation metrics

The proposed MTMCT method is evaluated using two popular metrics for the MTMCT problem: **IDF1** (Ristani et al. 2016) and **HOTA** (Higher Order Tracking Accuracy) (Luiten et al. 2021). Furthermore, when working with I24 datasets, we employ the Recall metric to evaluate the performance of the proposed method in comparison to other baseline methods mentioned in the original dataset study (Gloudemans et al., 2024). In addition, for the detection task in the LMD module, we use mean average precision (mAP) as a metric to evaluate the result.

A.1 IDF1

IDF1 is defined as the ratio of the number of correctly identified objects to the number of ground truth and average objects. The IDF1 formula is presented below:

$$IDF1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{A1}$$

where TP is true positive, FP is false positive, and FN is false negative matching.

A.2 HOTA

HOTA evaluates the alignment of matching detections' trajectories and calculates an average score while simultaneously penalizing unmatched detections. HOTA effectively balances the detection and association errors by giving equal weight to both detection and association accuracy in its formulation. Below is the HOTA formula:

$$HOTA = \sqrt{\frac{\sum_{c \in \{TP\}} A(c)}{TP + FN + FP}} \tag{A2}$$

$$A(c) = \frac{TPA(c)}{TPA(c) + FNA(c) + FPA(c)} \tag{A3}$$

where TPAs (True Positive Associations), FPAs (False Positive Associations) and FNAs (False Negative Associations) are designated for each TP (True Positive). Given a specific TP, c, The set of TPAs consists of TPs that have both the same gtID and prID as c.

$$TPA(c) = \{k\}, k \in \{TP \| prID(k) = prID(c) \wedge gtID(k) = gtID(c)\} \tag{A4}$$

Similarly, given a specific TP, c, The collection of FNAs consists of the gtDets that share the same gtID as c, but have been assigned a different prID than c or no prID if they were overlooked:

$$FNA(c) = \{k\}, k \in \begin{cases} \{TP \| prID(k) \neq prID(c) \wedge gtID(k) = gtID(c)\} \\ \cup \{FN \| gtID(k) = gtID(c)\} \end{cases} \tag{A5}$$

Finally, given a specific TP, c, The collection of FPAs consists of prDets that share the same prID as c, but have been allocated a different gtID than c or no gtID if they do not relate to an object:

$$FPA(c) = \{k\}, k \in \begin{cases} \{TP \| prID(k) = prID(c) \wedge gtID(k) \neq gtID(c)\} \\ \cup \{FP \| prID(k) = prID(c)\} \end{cases} \tag{A6}$$

Appendix B ablation studies on cityFlow dataset

This section shows the additional ablation studies learning the parameters for LMD and LGMMA modules.

B.1 LMD ablation study

For the LMD module, we examine eight parameters:

- **loss_wei**: $\{\alpha_1, \alpha_2, \alpha_3\}$ from the loss equation 1. The default value is {5,2,2}.
- **ce_wei**: $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7\}$ from the ce loss equation 2. The default values are {1,1,1,1,1,1,1}.
- γ: the γ from focal loss (Lin et al., 2017). The concept involves shifting the model's focus towards samples that are difficult to classify. It is crucial as it aids in addressing the challenge posed by a large number of classes. The default value is 2.
- **lr**: The learning rate. The default value is $1.00E-04$.
- **num_bins**: The number of bins for for nomalizing the coordinate $\{b_1, b_2, b_3, b_4\}$. The default value is 500.
- **accu_bat**: The Accumulate gradient batch (Smith et al., 2017). The goal is to enhance the stability of the training process by augmenting the batch size. The default value is 1.
- **dropout**: The drop out in neuron network. The default value is 0.1.
- **num_lay_LMD**: The number of layers in the encoder and decoder of deformable DETR. The default value is 6.

Table 5 Tuning the first four parameters of the LMD module

Tuning	loss_wei	ce_wei	γ	lr	mAP (%)
loss_wei	5,2,2	1,1,1,1,1,1,1	2	1.00E-04	17.05
	2,2,2	1,1,1,1,1,1,1	2	1.00E-04	15.63
	2,5,2	1,1,1,1,1,1,1	2	1.00E-04	**20.14**
	2,2,5	1,1,1,1,1,1,1	2	1.00E-04	16.42
	5,0,0	1,1,1,1,1,1,1	2	1.00E-04	11.26
ce_wei	2,5,2	1,1,1,1,1,1,1	2	1.00E-04	20.14
	2,5,2	1,1,1,1,1,2,2	2	1.00E-04	21.27
	2,5,2	1,1,1,1,1,4,4	2	1.00E-04	23.73
	2,5,2	**1,1,1,1,1,0.5,0.5**	2	1.00E-04	**28.12**
	2,5,2	1,1,1,1,1,0.2,0.2	2	1.00E-04	23.18
	2,5,2	1,1,1,1,1,0.1,0.1	2	1.00E-04	17.26
gamma	2,5,2	1,1,1,1,1,0.5,0.5	2	1.00E-04	28.12
	2,5,2	1,1,1,1,1,0.5,0.5	1	1.00E-04	25.45
	2,5,2	1,1,1,1,1,0.5,0.5	**4**	1.00E-04	**29.04**
	2,5,2	1,1,1,1,1,0.5,0.5	8	1.00E-04	28.43
lr	2,5,2	1,1,1,1,1,0.5,0.5	4	1.00E-04	29.04
	2,5,2	1,1,1,1,1,0.5,0.5	4	**2.00E-04**	**31.68**
	2,5,2	1,1,1,1,1,0.5,0.5	4	4.00E-04	30.36
	2,5,2	1,1,1,1,1,0.5,0.5	4	5.00E-05	27.54

The boldfaced are the selected options

We systematically adjusted each parameter individually by training the model at 65 epochs. The outcomes of the initial four parameters are shown in Table 5. The table demonstrates a significant improvement in the results by proper parameter tuning. The result shows that the parameters β_6 and β_7 are sensitive and have a big impact on LMD results. The reason for that is the quantity of car type and color labels is relatively small in comparison to the quantity of location token labels. More precisely, as stated in Cityflow-NL dataset (Feng et al., 2021), the color of the vehicle has eleven labels while the type of the car has only seven. Conversely, the location token has 1000 labels.

Table 6 shows the results of the last four parameters. Although LMD yields the highest mAP of 35.81 at the num_lay_LMD parameter value of 8, we decide to select a num_lay_LMD parameter value of 6 which delivers a mAP of 35.67. The rationale behind this is the trade-off between the speed of the inference stage and the marginal improvement in mAP.

B.2 CamEnc Ablation Study On Various Graph Structures

The illustration in Fig. 3 shows that the generated graph is either a complete graph (Scenarios 1, 2, 3, and 4-5) or a path graph (Scenario 6). Considering that the adapted GNNs (GIN, GCN, and GAT) are constrained by structural features, it is worth examining if these GNNs can converge and contribute to prediction accuracy. Hence, ablation studies of CamEnc on different graph structures are conducted in this section.

Firstly, we carry out an experiment to evaluate the efficiency of LaMMOn with and without CamEnc using GNN approaches (GIN, GCN, GAT) on the Scenario 2 dataset (complete graph). In this experiment, we commence the training of LaMMOn from the

Table 6 Tuning the last four parameters of the LMD module

Tuning	num_bin	accu_bat	dropout	num_lay_LMD	mAP
num_bin	500	1	0.1	6	31.68
	250	1	0.1	6	28.32
	1000	1	0.1	6	**32.74**
	2000	1	0.1	6	32.68
accu_bat	1000	1	0.1	**6**	32.74
	1000	2	0.1	6	33.21
	1000	**4**	0.1	6	**34.57**
	1000	8	0.1	6	32.42
dropout	1000	**4**	0.1	6	34.57
	1000	4	0.05	6	30.19
	1000	4	**0.2**	6	**35.67**
	1000	4	0.4	6	34.12
num_lay_LMD	1000	4	0.2	**6**	**35.67**
	1000	4	0.2	4	34.53
	1000	4	0.2	8	35.81

The boldfaced are the selected options

Table 7 LaMMOn performance on scenario 2 of CityFlow dataset with and without CamEnc

GNN for Cam_Enc	IDF1	HOTA
No graph embedding	72.41	71.55
GIN	**72.53**	**71.74**
GCN	72.06	71.41
GAT	71.98	71.03

Bold values indicate the models or parameter settings that achieve the best performance

beginning, utilizing the data obtained from Scenarios 1, 3, and 4-5. Next, we execute the inference on the dataset of Scenario 2. The result in Table 7 demonstrates that the suggested CamEnc approach has a restricted effect on scenarios when camera graphs are complete graphs.

Secondly, it is necessary to do another ablation study on a graph that is distinct from the complete graph in order to fully assess the effectiveness of CamEnc. Therefore, we proceed to conduct an identical experiment on Scenario 6, which is a path graph and not a complete one. In this experiment, we utilize data from Scenarios 1, 2, 3, 4, and 5 to train the LaMMOn model and subsequently do inference on Scenario 6. The result in Table 8 shows that the utilization of CamEnc significantly enhances the IDF1 score of LaMMOn by 4.3% in situations where camera graphs are not complete graphs.

The two experimental results above demonstrate that, while the complete graph structures (Scenarios 1, 2, 3, and 4-5) of the Camera ID Encoder (CamEnc) show only a small improvement, the path graph structure in Scenario 6 shows a significant improvement in the final embeddings of the CamEnc component. Therefore, using the graph topology on the CamEnc component is necessary for the LMD module.

Table 8 LAMMON performance on scenario 2 of CityFlow dataset with and without CamEnc

GNN for CamEnc	IDF1	HOTA
No graph embedding	74.53	74.36
GIN	**78.83**	**76.48**
GCN	77.95	76.07
GAT	78.16	75.84

Bold values indicate the models or parameter settings that achieve the best performance

B.3 LGMA ablation study

After freezing the LMD parameters as in the previous section, we conduct an ablation study on the parameters of LGMA (without using T2E) as follows:

- **num_fra**: The number of frames in the global association is handled concurrently. The larger the value of num_fra, the faster the model will run. Nevertheless, an excessively large value for num_fra can have a detrimental impact on the model's performance. The default value is 4.
- **mem_buff**: The technique of generating the representation of tracklets is in a memory buffer. The default option is to choose the most recent embedding. The *attention* option uses the method described in the meMOT (Cai et al., 2022) with the default parameters.
- **num_lay**: The number of layers is in the encoder and decoder of LGMA. The default value is 1.
- **graph_tok**: The GNN layer type generates graph-based token feature construction. The default value is *no*, indicating that we did not use Graph-Based Feature Construction and only used the object embedding.
- τ: The distance threshold used to create a graph for Graph-Based Token Feature Construction in the LGMA module. More precisely, we retain only the edges that are larger than the threshold value τ. Put simply, reducing this value will result in a higher quantity of edges in the graph that is produced, and conversely, increasing the value will lead to a lower number of edges.

We methodically fine-tuned each of these parameters separately by training the model for 65 epochs. The results are presented in Table 9. The table illustrates a significant improvement in the results achieved by optimizing the parameters appropriately. Especially by implementing graph-based token feature construction, the IDF1 metric has increased by 3.78%.

Table 9 Tuning parameters of the LGMA module. The boldfaced are the selected options

Tuning	num_fra	mem_buff	num_lay	graph_tok	τ	IDF1
num_fra	4	most_resent	1	no	0.5	66.66%
	8	most_resent	1	no	0.5	**68.11%**
	16	most_resent	1	no	0.5	67.73%
	32	most_resent	1	no	0.5	64.21%
	64	most_resent	1	no	0.5	61.49%
mem_buff	8	most_resent	1	no	0.5	68.11%
	8	mean_3	1	no	0.5	69.05%
	8	**mean_5**	1	no	0.5	**71.81%**
	8	mean_7	1	no	0.5	69.66%
	8	log_mean_3	**1**	no	0.5	**69.10%**
	8	log_mean_5	1	no	0.5	69.42%
	8	log_mean_7	1	no	0.5	68.63%
	8	attention	1	no	0.5	69.49%
num_lay	8	mean_5	1	no	0.5	71.81%
	8	mean_5	**2**	no	0.5	**73.54%**
	8	mean_5	4	no	0.5	73.03%
	8	mean_5	6	no	0.5	72.84%
graph_tok	8	mean_5	2	no	0.5	73.54%
	8	mean_5	2	GAT	0.5	74.23%
	8	mean_5	2	GCN	0.5	73.59%
	8	mean_5	2	GGNN	0.5	74.41%
	8	mean_5	2	Gin	0.5	76.14%
	8	mean_5	2	**GraphSage**	0.5	**77.32%**
τ	8	mean_5	2	GraphSage	0.3	75.81%
	8	mean_5	2	GraphSage	0.4	77.05%
	8	mean_5	2	GraphSage	**0.5**	**77.32%**
	8	mean_5	2	GraphSage	0.6	76.88%
	8	mean_5	2	GraphSage	0.7	76.23%

Bold values indicate the models or parameter settings that achieve the best performance

Appendix C comprehensive analysis and application guide of the T2E module in novel scenarios

C.1 detailed application Guide for the T2E module in novel scenarios

In general, the T2E module is trained and employed by utilizing prior spatial and temporal information in a new scenario to create synthetic object embeddings. These object embeddings are then used to train the LGMA module. To provide additional clarification, let's consider an example of utilizing the T2E module to generate the synthetic object embedding on the CityFlow dataset's test set, specifically scenario 6. Firstly, we obtained the camera locations as shown in Fig. 7.

The cars are required to travel through the cameras in the sequence 41→42→43→44→45→46 (or vice versa 46→45→44→43→42→41). Figure 8 displays the arrows that represent the potential paths of cars traveling from camera 42 toward camera 43.

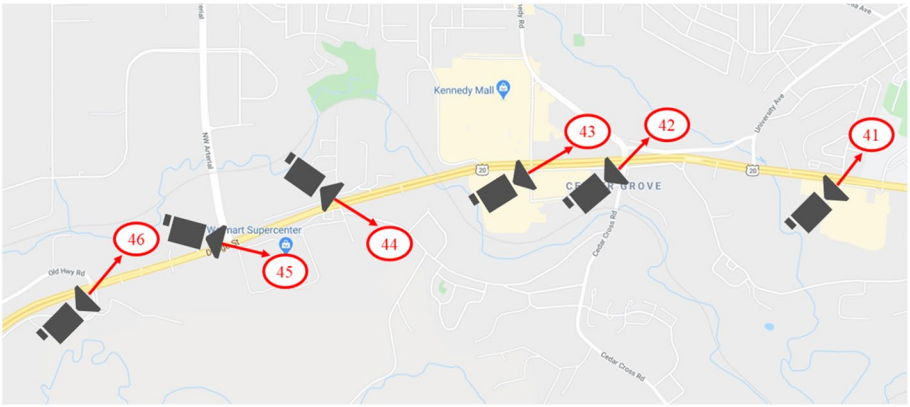

Fig. 7 Camera location in the test set CityFlow (scenario 6)

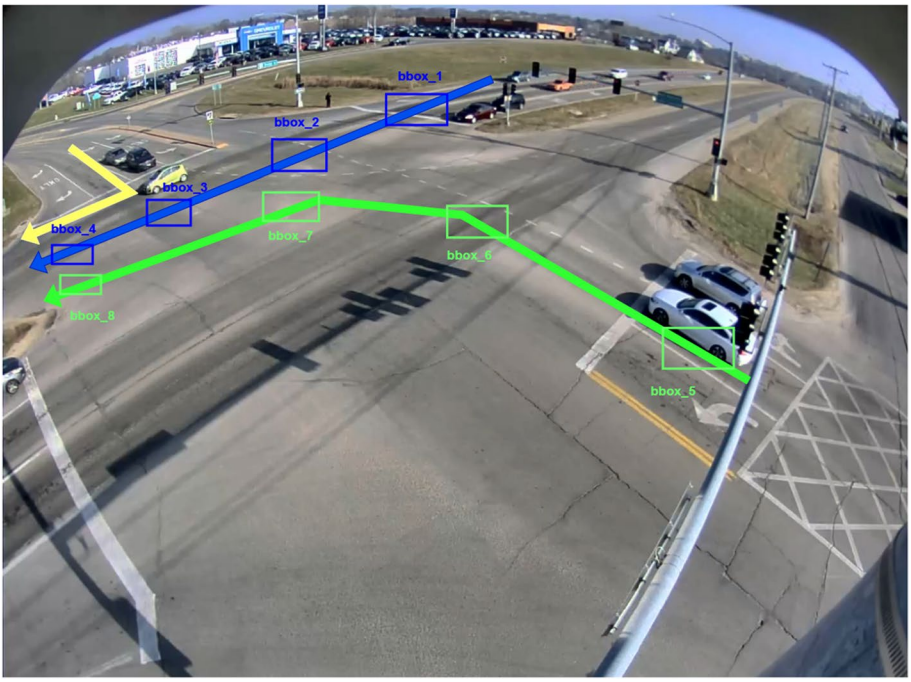

Fig. 8 Generated Trajectory in Camera 42

Next, we proceed methodically, adhering to each individual step.

- *Step 1:* We construct four bounding boxes for each trajectory in a linear manner (see picture 2 above). It is important to note that this example has only four bounding boxes for each trajectory. However, in reality, this number is far more than 4 and is contingent upon the duration of the trajectory and our underlying assumption. In section 4.4.1, we specify our assumption as follows: "It is assumed that there are 15 cars every minute,

Table 10 Textual representation for generated trajectories of camera 42 in Fig. 8

Frame ID	Label	Defined text
1	bbox_1	Cam42 loc547 loc141 loc84 loc40 Sedan blue
1	bbox_5	Cam42 loc920 loc460 loc96 loc54 SUV green
2	bbox_2	Cam42 loc393 loc203 loc74 loc37 Sedan blue
2	bbox_6	Cam42 loc626 loc289 loc80 loc41 SUV green
3	bbox_3	Cam42 loc217 loc276 loc57 loc31 Sedan blue
3	bbox_7	Cam42 loc380 loc270 loc76 loc37 SUV green
4	bbox_4	Cam42 loc91 loc51 loc54 loc25 Sedan blue
4	bbox_8	Cam42 loc98 loc374 loc52 loc26 SUV green

Fig. 9 Generated Trajectory in Camera 43

with each car appearing on camera for 45 s." Hence, at a frame rate of 10 frames per second, there are 6750 bounding boxes each minute, and the length of a minute is indicated by the `T2E_min` parameter.

- *Step 2:* We simply produce a textual representation of the bounding boxes of trajectories in Fig. 8. The texts that have been specified are shown in Table 10.
- *Step 3:* Similarly, it is easy to create textual representations for the bounding boxes of trajectories in camera 43 (as well as other adjacent cameras). The trajectories observed in Camera 43 and their corresponding labels are shown in Fig. 9 and Table 11.

Table 11 Textual representation for generated trajectories of camera 43 in Fig. 9

Frame ID	Label	Defined text
5	bbox_9	Cam43 loc894 loc97 loc46 loc25 Sedan blue
5	bbox_13	Cam43 loc977 loc114 loc54 loc30 SUV green
6	bbox_10	Cam43 loc576 loc129 loc72 loc43 Sedan blue
6	bbox_14	Cam43 loc663 loc164 loc78 loc44 SUV green
7	bbox_11	Cam43 loc301 loc184 loc66 loc33 Sedan blue
7	bbox_15	Cam43 loc390 loc208 loc68 loc40 SUV green
8	bbox_12	Cam43 loc100 loc230 loc56 loc24 Sedan blue
8	bbox_16	Cam43 loc159 loc254 loc53 loc28 SUV green

- *Step 4:* We input the defined text mentioned above into the T2E module, frame by frame, in the sequential order of frame ID. This process generates object embeddings, which are then utilized to train the LGMA module.
- *Step 5:* We utilize a trained LAMMON model to make inference on the test set.

C.2 accuracy evaluation of real vs. synthetic tokens generated by the T2E module

In order to assess the accuracy of the generated tokens, we carry out an experiment to compare the actual tokens with the synthetic tokens that are generated from T2E. The following is a sequential procedure:

- *Step 1:* We generate real tokens for 5000 bounding boxes in each data scenario (the total number of bounding boxes for each scenario is shown in the table below). To clarify, there is a combined total of 25,000 tokens across five different data scenarios.
- *Step 2:* We employ T2E to produce 25000 synthetic tokens from the corresponding textual representation.
- *Step 3:* We assess the difference between tokens and synthetic tokens using the cosine similarity metric.
- *Step 4:* We compute the mean of the cosine similarity score for each class, which is determined by the type and color of the car. Next, we represent the aforementioned outcome in the heatmap Figs. 10

The findings indicate that the majority of the cosine similarity scores range from 0.8 to 0.92, hence showcasing the accuracy of the T2E module. Nevertheless, the performance of the “wagon”category is notably inferior compared to other categories, with an average score of 0.82. Additionally, several classes within the“wagon”category, such as wagon_blue, wagon_gray, wagon_brown, and wagon_silver, have scores below 0.8. This issue may arise because of the constraint of“wagon”in the training data and the resemblance between wagon and SUV in terms of their shape.

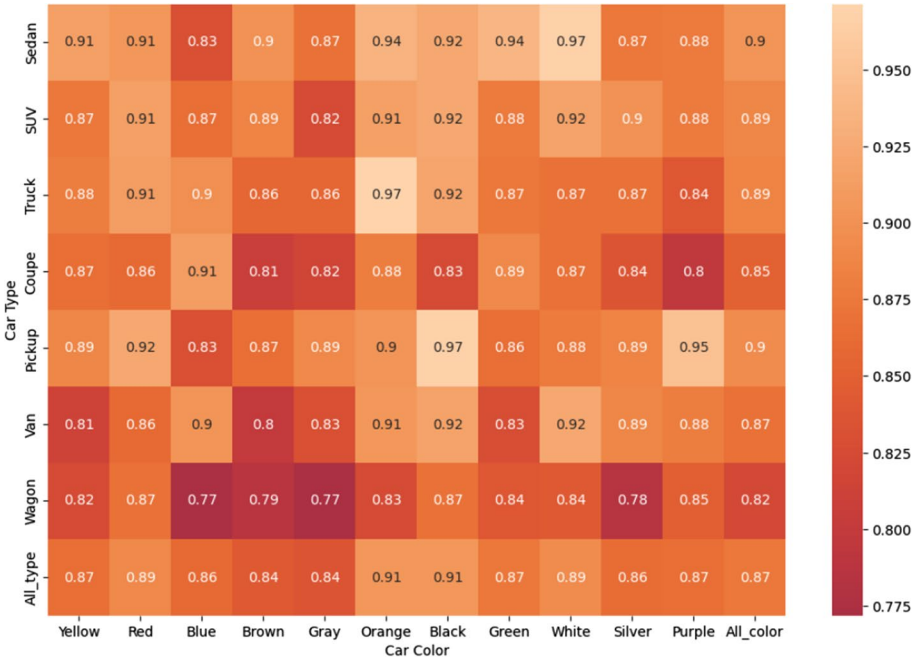

Fig. 10 Comparison of the real tokens and synthetic tokens using Cosine Similarity

C.3 T2E Module experimental study on novel scenario

In order to demonstrate the effectiveness of the T2E module, we conducted experimental research. We utilized LaMMOn, which used pre-trained parameters from the CityFlow dataset, to perform inference on the TrackCUIP test set. We compared two scenarios: one using the T2E module to synthesize data for TrackCUIP (with a T2E_min parameter of 32) and one without using the T2E module. The results from Table 12 indicate that the T2E module significantly enhances the HOTA by 6.15%, hence demonstrating the effectiveness of the T2E module in novel scenarios.

Table 12 Tracking result of LaMMOn and other methods on TrackCUIP dataset

	IDF1	HOTA
LaMMOn- Without T2E	72.57	71.46
LaMMOn- With T2E	**77.95**	**77.61**

Author contributions T.T.N. conceived the original idea and experimental settings. T.T.N. and H.H.N. developed the framework. T.T.N. performed the experiments for the evaluation. M.F. and M.S. verified the analytical methods and directed the project.

Funding This material is based upon work partially supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Award Number DE-EE0009208. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Data availability All the datasets, excluding TrackCUIP, used in this study are publicly available at https://github.com/elituan/lammon, https://www.aicitychallenge.org/, and https://i24motion.org/.

Code availability https://github.com/elituan/lammon

Declarations

Conflict of interest The domain of each institution (semicolon separated) that the authors have a Conflict of interest with includes https://www.smu.edu.sg/; https://hcmut.edu.vn/; https://www.utc.edu/; https://www.l3s.de/; https://www.uni-hannover.de/de/.

Ethical approval This article does not require permission for ethics approval or consent to participation as this work is based on all the publicly available datasets.

Consent for publication All authors of this manuscript consent to its publication.

References

Bergmann, P., Meinhardt, T., & Leal-Taixe, L.(2019). Tracking without bells and whistles. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 941–951)

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B.(2016). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, (pp. 3464–3468). IEEE

Brasó, G., & Leal-Taixé, L.(2020). Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 6247–6257)

Cai, J., Xu, M., Li, W., Xiong, Y., Xia, W., Tu, Z., & Soatto, S.(2022). Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 8090–8100)

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S.(2020). End-to-end object detection with transformers. In: *European conference on computer vision*, (pp. 213–229). Springer

Chen, T., Saxena, S., Li, L., Fleet, D.J., & Hinton, G.(2021). Pix2seq: A language modeling framework for object detection. arXiv preprint arXiv:2109.10852

Chohan, S. R., Hu, G., Khan, A. U., Pasha, A. T., Saleem, F., & Sheikh, M. A. (2023). Iot as societal transformer: improving citizens' continuous usage intention in digital society through perceived public value. *Library Hi Tech, 41*(4), 1214–1237.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., &Tian, Q.(2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 6569–6578)

Feng, Q., Ablavsky, V., & Sclaroff, S.(2021). Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions. arXiv preprint arXiv:2101.04741

Fisichella, M. (2022). Siamese coding network and pair similarity prediction for near-duplicate image detection. *International Journal of Multimedia Information Retrieval, 11*(2), 159–170.

Frossard, D., & Urtasun, R.(2018). End-to-end learning of multi-sensor 3d tracking by detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, (pp. 635–642). IEEE

Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., & Torr, P. (2019). Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(2), 652–662.

Ghaffar Nia, N., Kaplanoglu, E., & Nasab, A. (2023). Evaluation of artificial intelligence techniques in disease diagnosis and prediction. *Discover Artificial Intelligence, 3*(1), 5.

Gloudemans, D., Zachár, G., Wang, Y., Ji, J., Nice, M., Bunting, M., Barbour, W.W., Sprinkle, J., Piccoli, B., & Monache, M.L.D., et al. (2024). So you think you can track?. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 4528–4538)

Gloudemans, D., Wang, Y., Gumm, G., Barbour, W., & Work, D.B.(2023). The interstate-24 3d dataset: A new benchmark for 3d multi-camera vehicle tracking. arXiv preprint arXiv:2308.14833

Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings 2005 IEEE international joint conference on neural networks*, (vol. 2, pp. 729–734)

Guo, S., Wang, J., Wang, X., &Tao, D.(2021). Online multiple object tracking with cross-task synergy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8136–8145)

Harris, A., Stovall, J., & Sartipi, M.(2019). Mlk smart corridor: An urban testbed for smart city applications. In *2019 IEEE international conference on big data (Big Data)* (pp. 3506–3511). IEEE

Hassan, Y., Zhao, J., Harris, A., & Sartipi, M.(2023). Deep learning-based framework for traffic estimation for the mlk smart corridor in downtown chattanooga, tn. In *2023 IEEE 26th international conference on intelligent transportation systems (ITSC)* (pp. 4564–4570). IEEE

Hou, Y., Zheng, L., Wang, Z., & Wang, S.(2019). Locality aware appearance metric for multi-target multi-camera tracking. arXiv preprint arXiv:1911.12037

Khaleghian, S., Neema, H., Sartipi, M., Tran, T., Sen, R., & Dubey, A.(2023). Calibrating real-world city traffic simulation model using vehicle speed data. In *2023 IEEE international conference on smart computing (SMARTCOMP)*, (pp. 303–308) . IEEE

Kim, C., Fuxin, L., Alotaibi, M., & Rehg, J.M.(2021). Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9553–9562)

Kudo, T., & Richardson, J.(2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226

Kumarasamy, V.K., Saroj, A.J., Liang, Y., Wu, D., Hunter, M.P., Guin, A., & Sartipi, M.(2023). Traffic signal optimization by integrating reinforcement learning and digital twins. In *2023 IEEE smart world congress (SWC)* (pp. 1–8). IEEE

Li, J., Gao, X., & Jiang, T.(2020). Graph networks for multiple object tracking. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, (pp. 719–728)

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P.(2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988)

Lu, J., Clark, C., Zellers, R., Mottaghi, R., & Kembhavi, A.(2022). Unified-io: A unified model for vision, language, and multi-modal tasks. arXiv preprint arXiv:2206.08916

Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision, 129*, 548–578.

Meinhardt, T., Kirillov, A., Leal-Taixe, L., &Feichtenhofer, C.(2022). Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8844–8854)

Nguyen, T. T., Nguyen, H. H., Sartipi, M., & Fisichella, M. (2023). Multi-vehicle multi-camera tracking with graph-based tracklet features. *IEEE Transactions on Multimedia, 26*, 972–983.

Nguyen, T. T., Nguyen, H. H., Sartipi, M., & Fisichella, M. (2023). Real-time multi-vehicle multi-camera tracking with graph-based tracklet features. *Transportation Research Record, 2678*(1), 296–308.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., & Tomasi, C.(2016). Performance measures and a data set for multi-target, multi-camera tracking. In: *European conference on computer vision* (pp. 17–35). Springer

Shim, K., Yoon, S., Ko, K., & Kim, C.(2021). Multi-target multi-camera vehicle tracking for city-scale traffic management. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4193–4200)

Smith, S.L., Kindermans, P.-J., Ying, C., & Le, Q.V.(2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489

Specker, A., Stadler, D., Florin, L., & Beyerer, J.(2021). An occlusion-aware multi-target multi-camera tracking system. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 4173–4182)

Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., & Luo, P.(2020). Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460

Tang, Z., Naphade, M., Liu, M.-Y., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D., &Hwang, J.-N.(2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8797–8806)

Tesfaye, Y.T., Zemene, E., Prati, A., Pelillo, M., &Shah, M.(2017). Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. arXiv preprint arXiv:1706.06196

Ullah, M., & Alaya Cheikh, F.(2018). A directed sparse graphical model for multi-target tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1816–1823)

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y.(2018). Graph attention networks. In *International conference on learning representations*

Welling, M., & Kipf, T.N.(2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the international conference on learning representations (ICLR)*

Weng, X., Wang, Y., Man, Y., & Kitani, K.M.(2020). Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 6499–6508)

Xu, K., Hu, W., Leskovec, J., Jegelka, S.(2019). How powerful are graph neural networks? In *International conference on learning representations*. https://openreview.net/forum?id=ryGs6iA5Km

Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., & Alameda-Pineda, X.(2021). Transcenter: Transformers with dense queries for multiple-object tracking

Yao, H., Duan, Z., Xie, Z., Chen, J., Wu, X., Xu, D., & Gao, Y.(2022). City-scale multi-camera vehicle tracking based on space-time-appearance features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3310–3318).

Zhou, X., Yin, T., Koltun, V., & Krähenbühl, P.(2022). Global tracking transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 8771–8780)

Zhou, X., Koltun, V., &Krähenbühl, P.(2020). Tracking objects as points. In *European conference on computer vision*, (pp. 474–490). Springer

Zhu, T., Hiller, M., Ehsanpour, M., Ma, R., Drummond, T., Reid, I., & Rezatofighi, H. (2022). Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(11), 12783–12797.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., &Dai, J.(2020). Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159