Multi-Scene Generalized Trajectory Global Graph Solver with Composite Nodes for Multiple Object Tracking

Yan Gao¹, Haojun Xu¹, Jie Li¹*, Nannan Wang¹, Xinbo Gao¹,²*
¹Xidian University
²Chongqing University of Posts and Telecommunications
{yangao, haojunxu}@stu.xidian.edu.cn, nnwang@xidian.edu.cn, leejie@mail.xidian.edu.cn, gaoxb@cqupt.edu.cn

Abstract
The global multi-object tracking (MOT) system can consider interaction, occlusion, and other "visual blur" scenarios to ensure effective object tracking in long videos. Among them, graph-based tracking-by-detection paradigms achieve surprising performance. However, their fully-connected nature poses storage space requirements that challenge algorithm handling long videos. Currently, commonly used methods are still generated trajectories by building one-forward associations across frames. Such matches produced under the guidance of first-order similarity information may not be optimal from a longer-time perspective. Moreover, they often lack an end-to-end scheme for correcting mismatches. This paper proposes the Composite Node Message Passing Network (CoNo-Link), a multi-scene generalized framework for modeling ultra-long frames information for association. CoNo-Link’s solution is a low-storage overhead method for building constrained connected graphs. In addition to the previous method of treating objects as nodes, the network innovatively treats object trajectories as nodes for information interaction, improving the graph neural network’s feature representation capability. Specifically, we formulate the graph-building problem as a top-k selection task for some reliable objects or trajectories. Our model can learn better predictions on longer-time scales by adding composite nodes. As a result, our method outperforms the state-of-the-art in several commonly used datasets.

Introduction
Multi-Object Tracking (MOT) aims to generate trajectories for all moving objects in a video stream. It is a fundamental module for video content analysis in application areas such as autonomous driving and intelligent robotics. In recent years, tracking-by-detection as the dominant paradigm in the field divides the task into (i) frame-by-frame object detection and (ii) data association, i.e., linking potential targets to the correct object trajectory whenever possible. Data association is performed mainly on neighboring frames, greedily matching trajectories with detection proposals through simple cues such as position and motion prediction (Zhang et al. 2022; Tokmakov et al. 2021; Zhou, Koltun, and Krähenbühl 2020; Bergmann, Meinhardt, and Leal-Taixé 2019; Bewley et al. 2016) or appearance features (Pang et al. 2021; Zhang et al. 2021). The trajectories formed by such local trackers can satisfy the accuracy requirements. However, severe object occlusion and appearance changes in crowded scenes can pose challenges for long-term identity preservation. Many studies have executed offline associations over the entire video frame to obtain long-range trajectories. Such global trackers typically require the construction of global appearance and motion cues or pairwise association between tracklets on all frames based on graphical representations.

To maximize the applicability to each scenario, recent graph-based MOT studies (Brasó and Leal-Taixé 2020; Hornáková et al. 2020, 2021; Cetintas, Brasó, and Leal-Taixé 2023) have adopted a generic approach to global tracking by using a unified module to process videos in a single framework. While the research has shown promising results, there are still some issues with the generalized graph-based approach. A significant challenge is the need to consider the time cues of video clips more. Generic methods generally use a uniform, fully-connected graph-building system, and due to storage constraints, the node information encoded is limited to neighboring frames, forming trajectories that may not be optimal in the temporal dimension. In contrast, when humans perform dynamic tracking, they must maintain the temporal consistency of the perceived object
nodes. We then reason by the link block to predict the edge graph $G$ puts long clips and outputs a one-piece partially connected handle long clips. (ii) Distant nodes are reachable in neural network (GNN) link block that accomplishes reasoning about the entire domain. Specifically, NodeNet inputs long clips and outputs a one-piece partially connected graph $G_{\text{part}}$ containing potential connections and composite nodes. We then reason by the link block to predict the edge score for $G_{\text{part}}$ and output the final result.

$G_{\text{part}}$ is the core of our CoNo-Link, inspired by the human visual system. Research (Hyvärinen et al. 2014) shows that infants can expand from tracking salient objects to arbitrary objects when they observe the world. That means observation and tracking is a continuous process in which there are priorities of interest. We summarize three tracking characteristics in the visual system and design two types of nodes and three types of connections accordingly. Among them, the detection node priority confirms the significantly moving objects, the trajectory node records the continuous motion of things, and the connections between composite nodes create pathways for the expression of relationships between arbitrary objects. We use a GNN to process the trajectory nodes and propose a complementary edge learning strategy for these nodes that allows potentially misclassified edges to resume learning.

Fig. 1 includes the node connectivity relations of different graph structures when generating a trajectory $T$. In the view of blue nodes, both the fully-connected frame-by-frame graph $G_{\text{fully}}$ and the hierarchical neighboring frames graph $G_{\text{localFully}}$ have to be connected to all the nodes of the next frame, while our $G_{\text{part}}$ only needs to be connected to meaningful object nodes. This scheme solves two limitations of the previous approaches: (i) Partially link significantly reduces memory consumption and can efficiently handle long clips. (ii) Distant nodes are reachable in $G_{\text{part}}$, with better feature learning potential. We evaluate the proposed CoNo-Link on several benchmarks: achieve a competitive 83.7 IDF1 and 82.7 MOTA on MOT17, 81.8 and 77.5 on MOT20, and 64.1 and 89.7 on DanceTrack.

In summary, we make the following main contributions.

- We propose a composite node messaging network that achieves globally optimal convergence by gradually aggregating meaningful trajectory nodes. It guarantees continuous solutions with significantly lower computational costs.
- We construct a partially connected graph based on composite nodes to build a perfect graph domain in which a generalized MOT graph solver can effectively utilize temporal information. It is a natural graph structure abstracted from human visual tracking.
- We achieve the state-of-the-art result in three public datasets and have multi-scene applicability.

Related Work

Local Short-Term Tracking

Many modern trackers run online using frame-by-frame or local frames (Bewley et al. 2016; Bergmann, Meinhardt, and Leal-Taixé 2019; Meinhardt et al. 2022; Pang et al. 2021; Tokmakov et al. 2021; Zeng et al. 2022; Zhang et al. 2022, 2021; Gao et al. 2022). Distance metrics for motion and spatial cues are often central to the design of trajectory formation for these methods. SORT (Bewley et al. 2016) and DeepSORT (Wojke, Bewley, and Paulus 2017; Wojke and Bewley 2018) have led many methods to use the Kalman filter as motion models (Bewley et al. 2016; Zhang et al. 2022, 2021; Dendorfer et al. 2022). CenterTrack (Zhou, Koltun, and Krähenbühl 2020) implements motion prediction and execution association through neural networks, which has led to the emergence of methods to improve motion prediction through improved model structures. Tracktor (Bergmann, Meinhardt, and Leal-Taixé 2019) inspired a frame-by-frame regression-based tracking framework (Li et al. 2022; Meinhardt et al. 2022; Zhou, Koltun, and Krähenbühl 2020; Bergman, Meinhardt, and Leal-Taixé 2019). Some trackers use appearance to identify the same object to increase robustness in low-quality video scenes (Pang et al. 2021; Zhang et al. 2021; Xu et al. 2019; Wojke, Bewley, and Paulus 2017), such as low frame rates or strong camera motion (Dendorfer et al. 2022). Although these pair-wise association-based trackers have good tracking stability, they do not focus on the long-term preservation of object identity. Here we do this by performing association on all objects throughout the temporal dimension.

Graph-Based Global Tracking

Graphs are a framework well suited for modeling data association, where each object trajectory can be considered a simple graph with entry and exit degrees of 1 for each node except for the start and end nodes. They use object detection as object nodes and represent edges as possible trajectory hypotheses. In contrast to trackers that use neighbor frame information, graph-based approaches define the cross-frame object association problem as a global combinatorial optimization problem (Koh et al. 2022; Dendorfer et al. 2020a; Brasó and Leal-Taixé 2020; Zeng et al. 2022). To this end, many studies have used different optimization strategies, including multi-cuts (Tang et al. 2017), minimal cliques (Zamir, Dehghan, and Shah 2012), network flow (Bercia et al. 2011; Butt and Collins 2013), and disjoint path approaches (Hornák et al. 2020, 2021). Following (Brasó and Leal-Taixé 2020), this paper relies on a simplified minimum cost flow formulation (Zhang, Li, and Nevatia 2008a) to ensure that graph-based network structures are manageable.

Early graph-based methods often used methods based on handcrafted models (Takala and Pietikäinen 2007) or conditional random fields (Yang and Nevatia 2012) to obtain association cues. More recently, many approaches employ GNNs (Brasó, Cetintas, and Leal-Taixé 2022; Brasó and Leal-Taixé...
2020; Dai et al. 2021) or Transformers (Meinhardt et al. 2022; Zhou et al. 2022; Zeng et al. 2022) to learn features for the association. Li et al. (Hornaková et al. 2020) incorporates epistemic and pose features in optimizing disjoint path problems. MPNTrack (Brasó and Leal-Taixé 2020) proposes a neural solver to optimize a simplified graph. MOTR (Zeng et al. 2022) follows the DETR (Carion et al. 2020) structure to iteratively update the tracking queries in a propagated manner. The global association method GTR (Zhou et al. 2022) becomes popular. These methods first form short-range trajectories and then deals with lost trajectories. It is considerable.

Recently multi-stage tracking methods (Chen et al. 2020; Wu et al. 2021) become popular. These methods first form short-range trajectories and then deals with lost tracklets and occlusions over long periods. Different optimization techniques (Wu et al. 2021) and cues (Gupta, Dollár, and Girshick 2019) are designed for multi-stage merging when forming long-range trajectories. Many modern trackers incorporate such techniques to improve error correction (Berczal et al. 2021; Brasó and Leal-Taixé 2020; Meinhardt et al. 2022; Wojke and Bewley 2018). In this paper, we propose a learning strategy to gain the ability to reload missing edges during training.

Methodology

Preliminaries

Tracking-by-Detection (TBD) The TBD paradigm performs the task through frame-by-frame object detection and well-designed inter-target association. Let $I$ be a set of images in a video; the detector first identifies and locates targets in all elements of $I$, producing a set of objects $O$ with positions $\{p_i \mid p_i \in \mathbb{R}^4\}$. The object set $O$ used for the association is determined in some way, and it is common practice to keep the proposals above a set confidence threshold as candidate objects. Each candidate object $o_i$ contains the position $p_i$, the image patch $p_i$, and the time $t$. Then, the tracker associates each object in $O$ to obtain its trajectory over time $T = \{\tau_1, \ldots, \tau_K\}$, and $\tau_i = \{o_{t_1}, \ldots, o_{t_n}\}$ where $n_i$ is the trajectory length of the object $o_i$. In this paper, our algorithm also follows this paradigm.

Our tracker is built on modeling inter-frame object relations using an undirected graph $G = (V, E)$ where each node $V$ of $G$ corresponds to an object detection $o_i$. The edge $E \rightarrow \mathbb{R}^{V \times V}$ denotes the possible interactions between objects on different frames. The association assumption $E$ guarantees the connectivity of object pairs in different frames and facilitates trajectory error correction.

Graph-Based Tracking Baseline Based on $G$ as the underlying representation of inter-object relations, the proposed model is performed on a message-passing graph network tracker (Brasó and Leal-Taixé 2020) based on the classical MOT network flow formulation (Zhang, Li, and Neva-
Constructing a Partially Connected Tracking Graph

This study proposes matching using a composite node-based partial connectivity graph $G_{part}$ learning global information interactions over the time series of the entire video clip. Based on our experiments, we found that generating a fully connected graph and pruning it will always miss the edges. There are two types of reasons; one is that the same identity (ID) target is not in the video sampling window, and the other is that the connection is missing due to the feature learning problem of the model itself. Since missing links are unavoidable, we abandon the commonly used pruning method and instead construct the graph by identifying the “most promising” node relationships and backfilling the edges. Among them, the nodes can be the correct object detection or the shortest paths with the same ID that is easy to construct. The main idea is to follow a coarse-to-fine strategy: build an imperfect but relatively correct part of the node relationship graph and then determine new connections from the GNN to realize accurate tracking. Therefore, we propose NodeNet to establish the initial relationships between detections.

Establish the initial relationships. We employ a query representation to capture objects’ temporal and interaction relations. Specifically, we give each query node an initially constructed coarse trajectory in the semantic feature space by track search but keep only the candidate links for each node along the temporal dimension. Let $O = \{o_{i1}^{1}, \cdots, o_{iN_t}^{1}\}$ be a set of detector output objects for image $I^1$ with total number $N = \sum_{t=1}^{T} N_t$. Let $F^t = \{f_{i1}^{t}, \cdots, f_{iN_t}^{t}\}$ be the set of $D$-dimensional features extracted from their corresponding bounding boxes and $F = F^1 \cup \cdots \cup F^T$ is the set of all features in time slice $T$. As shown in Fig. 4, all objects with $F \in R^{N \times D}$ and query $Q_k \in R^D$ are the inputs to our NodeNet. The association scores $a \in R^N$ that produce the coarse trajectories are the outputs. Formally, we use a softmax activation to model the likelihood of association between an object $i$ and each trajectory $k$ at time $t$ as $P_M(o_i|Q_k, F) = \frac{exp(a_i)}{\sum_{j \in \{0, \cdots, N_t\}} exp(a_j)}$. Then, the distribution of all objects at time $t$ corresponding to trajectory $k$ is $P^t(p|Q_k, F) = \sum_{i=1}^{N_t} I_{p \in p^t} P_M(o_i|Q_k, F)$, where $I_{\cdot}$ assigns a bounding box $p$ to each query. Ultimately, the distribution of trajectory $k$ over the entire period is $P^T(\tau|Q_k, F) = \prod_{t=1}^{T} P^t(\tau^t|Q_k, F)$.

During training, we maximize the log-likelihood of the ground-truth trajectory, and features that are not matched are used as background queries and supervised empty set frames. Let $\partial_k$ be matched objects for a ground-truth trajectory $\tau$, and $F^m_{\partial_k}$ be the features of $\partial_k$. For each trajectory $\tau$, we optimize the training objective of the query assignment as:

$$L_{NodeNet} = \log(F) + \sum_{\tau} L_{matched}(F, \tau),$$

$$L_{bg}(F) = -\sum_{m=1}^{T} \sum_{j \in \partial_k} \sum_{t=1}^{T} \log P_M(\alpha^t = \emptyset | F^m_{\partial_k}, F),$$

$$L_{matched}(F, \tau) = -\sum_{m \in \{1, \cdots, T\} | \partial_k^m \neq \emptyset} \sum_{t=1}^{T} \log P_M(\alpha^m_{\partial_k^m} | F^m_{\partial_k}, F).$$

(1)

We first obtain the similarity matrix $M$ in the inference process. The method is to let NodeNet process the video stream as a sliding window. The length of the video stream is $T_{clip} = 512$, the window size is $T = 32$, and the step size is 16. We use a backbone network to extract the detected frames within the window to obtain $F$ sequentially. Then, let $Q = K = V = F$ feed them into NodeNet, and output an association matrix of $N \times N$. Assign this matrix to the corresponding position in the large matrix $M$, and if there is already a value in the corresponding place, take the average value in the overlapping part.

After obtaining the similarity matrix $M$, we will perform a frame-by-frame association. For the first frame, we initialize all detections to trajectories. For the $N_t$ bounding boxes in the subsequent frame $I^t$, the similarity sub-matrix $M$ of the bounding boxes within the window of size $min(t, T)$ is taken out in $M$ with this frame as the end of the window. The similarity between the query detections (trajectories) and the object detections within the window that have the same ID is summed to obtain the similarity matrix of each ID to the object of the current frame $M \in R^{ID \times N_t}$. In addition, we compute the IoU matrix $\hat{M} \in R^{ID \times N_t}$ between the last occurrence of each ID box and the current frame box. Our final cost matrix is $C = -\max(\hat{M}, M)$. Further, we use the Hungarian algorithm (Kuhn 2010) to get the optimal association for each ID. We start a new trajectory if the average association score with any previous ID is below a threshold $\theta$. Otherwise, we attach the generated trajectory’s underlying current detection (query) to the matching existing trajectory. In the similarity matrix $M$, we connect the optimal associations for each ID and the Top-$k$ bounding boxes with the highest similarity. These detections connections (Det-Det links) are the potential links we need.

Predict several short paths. We feed $G_{part}$ containing $T_{clip}$ information into the GNN for edge prediction. Then, we aggregate some natural short trajectories generated by the GNN into trajectory nodes $N^{track}$. Precisely, we fill in
the predicted scores of the GNN into \( M_{GNN} \) as the similarity between the nodes and determine the trajectory IDs using the same association mode for generating Det-Det links. For each path (containing Det-Traj links), the detection features within the trajectory are averaged as the new trajectory feature representation. Finally, we build connections between \( N^{traj} \) (Traj-Traj links) and do not establish the relationship if temporal IoU exists between \( N^{det} \) and \( N^{traj} \).

\( G^{part} \) is now the entire form that the complete set of candidate links established between pre-generated composite nodes. It contains Det-Det links, Det-Traj links, and Traj-Traj links, and we illustrate its construction in Fig. 3. These three links are significant because the Det-Det link generates new \( N^{traj} \). Det-Traj link is responsible for building the information interaction between \( N^{det} \) and \( N^{traj} \). The Traj-Traj link is responsible for fusing the sub-trajectories. In each iteration, the composite nodes undergo a directed transformation, and we transform the Traj-Traj links that satisfy the requirements into new \( N^{traj} \), which are then merged with the existing \( N^{traj} \) to generate the final result.

The summary of \( G^{part} \) vs. \( G^{full} \). The pre-generated composite nodes retain meaningful object nodes and reduce invalid edges, alleviating the labeling imbalance. This strategy reduces storage occupancy, so the graph contains all the information in clip time, guaranteeing spatial information interaction over long distances. At the same time, establishing multiple rather than a single kind of edges makes some nodes unavailable to reachable, making mega-graph long-distance matching plausible. Through upper-bound analysis experiments, we determined that the restricted connected graph based on composite node learning performs better than the hierarchical neighboring frame fully connected graph learning. Also, proof that our method satisfies globally optimal matching is provided in the supplementary.

Learn an Effective Link Tracker. We use a message-passing GNN (Brasó and Leal-Taixé 2020) to process a partially connected graph with composite nodes. Our main contribution is a learning strategy for GNN training. Specifically, the Traj-Traj links of our \( G^{part} \) are fully-connected edges constructed for \( N^{traj} \). Trajectory nodes are relatively few at this time, from 10k down to around the 0.1k level (see Table 2), making it possible to build fully-connected graphs between them, and can minimize the problem of missing links when creating graphs. We obtain the final trajectory by performing edge classification to decide whether to keep the established connections. We show the learning process in algorithm 1. The key to making the GNN learn efficiently is maintaining the proportion of positive sample edges, which is why our method works. In the following, we detail a GNN link block that makes long-time node connections.

Let \( G^I = (V^I, E^I) \) be a graph of our iteration \( I \). \( h_n^{(0)} \) is the node embedding for each \( n \in V^I \) and \( h_n^{(0)} \) is the edge embedding for each \( (u, v) \in E^I \). Following the time-aware framework of (Brasó and Leal-Taixé 2020), the GNN aims to learn a function that encodes the higher-order semantic context contained in the node and edge feature vectors via information propagation. It feeds the edge embedding to a multi-layer perceptron (MLP) and outputs a score \( y_{pred}^{(u,v)} = MLP(h^{(s)}_{u,v}) \), where \( (s) \) denotes the number of message passes. We set this score to a similarity with a minimum threshold \( \epsilon \) limitation for association. As mentioned,
we compute similarity scores based on association cues between nodes recorded in $M_{GNN}$. The cues contain the similarity of the appearance embedding and the estimated score based on the closest temporal distance. Additional details are provided in the supplementary material.

**Training** CoNo-Link contains two GNNs (Brasó and Leal-Taixé 2020) which are trained jointly at all levels. For this purpose, we first train the first network and add the second network to train at a certain level. This ensures stability during the training process. Specifically, we unfreeze the second network after 5000 training iterations. We trained them using a focal loss (Lin et al. 2020) with a $\gamma = 1$ and summed all the losses as our final loss.

### Experiments

#### Datasets and Metrics

We conducted experiments on three public benchmarks: MOT17 (Leal-Taixé et al. 2015; Milan et al. 2016), MOT20 (Dendorfer et al. 2020b), and DanceTrack (Sun et al. 2022). The MOT series is a dense pedestrian tracking dataset evaluated under public and private detection protocols. DanceTrack is a dataset of dance videos with similar appearance and complex motion patterns. We evaluate the performance using several widely used metrics: IDF1 (Ristani et al. 2016), MOTA (Kasturi et al. 2009), and HOTA (Luiten et al. 2021). IDF1 focuses on identity maintenance quality, while the official metric MOTA focuses on detection quality and tracking stability (IDS). HOTA combines two aspects to unify detection localization and association performance.

#### Implementation Details

Training parameters: we used the pre-trained ResNet50-IBN (Dai et al. 2021) as our ReID network and froze it during training. The GNNs were co-trained with a learning rate of $3 \times 10^{-4}$, weight decay of $10^{-4}$, and a batch of 2 clips with 200 epochs. The optimizer was Adam (Kingma and Ba 2015). Inference: our model can handle sequences of arbitrary length, and for training and memory efficiency, we have $T_{clip}$ of 512 frames. During inference, we fill the trajectory gaps by linear interpolation. The runtime on MOT17 is 19 FPS with given detections. Object detection: for DanceTrack and the private settings of MOT17 and MOT20, we follow the detections obtained in the YOLOX (Ge et al. 2021) trained in (Zhang et al. 2022). Our tracker was implemented on a single RTX 3090 GPU using Python 3.10 and Pytorch 2.0.0 (Paszke et al. 2019).

#### Ablation Study

**Experimental Settings** In this section, we use the MOT17 dataset for all experiments. To evaluate our model, we employ four video sequences (04, 05, 09, and 11) for training and the most challenging three sequences (02, 10, and 13) for validation. All ablation experiments were performed on the validation set.

**Candidate Link Generation Structure** We consider NodeNet’s three patterns, IoU, ReID, and Trans, for generating potential connections. IoU and ReID denote Hugarian matching (Kuhn 2010) based on IoU and ReID cues, and Trans. denotes query-based Transformer matching (i.e., for similarity associations considering global information). Based on these three models, we start the search for the relation-building range Top-$k$ using MOTA, IDF1, and the ground truth (GT) links coverage (Cover.) as evaluation metrics. To obtain as high edge coverage as possible without being affected by environmental factors, we use GT boxes as inputs to NodeNet based on the purity assumption. Table 1 shows the results. We use query matching to establish relationships between the top 5 matches. Although $k = 10$ performs best, the gain is small, and the number of edges becomes large. We also show in Fig. 5 the effect of the window size for establishing potential connections on these metrics. There is the highest GT envelope at $T = 32$ and does not has memory overflow.

#### Temporally Connected Graph Domains

We investigate the upper bounds on the performance of our proposed partially connected graph and the recent hierarchical fully connected graph. The experiment still uses GT detection as input for graph construction. We observe higher upper bounds for almost all metrics of $G_{part}$ in Table 2, including a reduction of IDS to 18. We calculate the edge and node numbers in Table 4 for both graphs under the same upper bound. $G_{local, fully}$ has an average of 256,095 edges and 7,482 nodes, whereas our $G_{part}$ has only 143,603 edges and 151 nodes. The results show that the proposed $G_{part}$ achieves 99.9% GT edge coverage before rounding than $G_{local, fully}$ when using fewer edges to build the graph. As expected, our $G_{part}$ is compelling in graph domain construction because it preserves temporal connectivity in addition to considering neighboring information. It allows the GNN to “see” nodes over time and make more credible decisions, thus generating significantly less IDS.

#### CoNo-Link’s Component Ablation Experiments

We show the performance of each part on the MOT17 validation set in Table 3. The first row shows the results after feed-

---

![Figure 5: The window size experiment of NodeNet.](image-url)
slightly decreases MOTA, the other strategies improve in all metrics and that our approach has a better balance. In scenarios with multiple complex motion patterns, we have good improvements on all metrics and that our approach has a better balance.

Table 2: Experiments on the upper bound of graph structures.

<table>
<thead>
<tr>
<th># Method</th>
<th>MOTA</th>
<th>IDF1</th>
<th>IDS</th>
<th>HOTA</th>
<th>MT</th>
<th>Cover.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G^{\text{global}}_\text{full}$</td>
<td>61.7</td>
<td>76.2</td>
<td>38</td>
<td>61.4</td>
<td>315</td>
<td>99.9</td>
</tr>
<tr>
<td>$G^{\text{part}}$</td>
<td>61.8</td>
<td>76.3</td>
<td>18</td>
<td>61.5</td>
<td>316</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of CoNo-Link.

<table>
<thead>
<tr>
<th># Method</th>
<th>GPLOPs</th>
<th>FPS</th>
<th>VRAM (MiB)</th>
<th>$D_{\text{avg}}$</th>
<th>$N_{\text{avg}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G^{\text{global}}_\text{full}$ + GNN</td>
<td>46.7</td>
<td>17.1</td>
<td>3,689</td>
<td>256,095</td>
<td>7,482</td>
</tr>
<tr>
<td>$G^{\text{part}}$</td>
<td>36.3</td>
<td>18.9</td>
<td>3,523</td>
<td>143,603</td>
<td>151</td>
</tr>
</tbody>
</table>

Table 4: Computational complexity experiments.

Benchmark Results

**MOT** We report the results of our model in Table 5 for MOT17 and MOT20 under the private detection protocol. Our approach achieved the desired results on both challenges. On MOT17, we outperform SUSHI based on hierarchical fully-connected graphs on all metrics. In the highly crowded scene of MOT20, it slightly underperforms SUSHI in identity switches. This result may be due to the large number of relational interactions in a short period that impact our approach. Still, again, the global information ensures that there are enough accurate tracklets in the results. Compared to ByteTrack, our model improves this by 6.4 IDF1, 4.0 HOTA, and 2.4 MOTA, reducing IDS by 50.3%. Our performance demonstrates the positive significance of the time-domain connectivity partially connected graph.

**DanceTrack** Table 6 demonstrates that in scenarios with multiple complex motion patterns, we have good improvements on all metrics and that our approach has a better balance between IDF1 and MOTA performance without excessive loss of accuracy. Results of this dataset show that an overall evaluation of trajectories in the temporal dimension can provide some correct clues for association.

Conclusion

This paper presents CoNo-Link, a feasible method for oversized graph tracking. Through ablation experiments, we demonstrate that partially connected graphs can reduce the video memory footprint and efficiently handle long video clips; the trajectory learning strategy can improve the upper bound of $G^{\text{part}}$ performance. In addition, the proposed method outperforms the SOTA approaches on three benchmarks. In the future, we would like to design an approach with CLIP (Radford et al. 2021) to enhance the performance in tracking through multimodal tasks, e.g., image caption.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 62176195, 62036007, U22A2096, U21A20514, and 62221005; in part by the Technology Innovation Leading Program of Shaanxi under Grant 2022QFY01-15; in part by Open Research Projects of Zhejiang Lab under Grant 2021KGOA0B01.
References


