STAR: Spatial-Temporal Tracklet Matching for Multi-Object Tracking

Xuewei Bai[1] **Yongcai Wang**[1*] **Deying Li**[1†] **Haodi Ping**[2‡] **Chunxu Li**[1,3]
[1] School of Information, Renmin University of China
[2] School of Computer Science, Beijing University of Technology
[3] China Waterborne Transport Research Institute
{bai_xuewei,ycw,deyingli}@ruc.edu.cn
haodi.ping@bjut.edu.cn
lichunxu@wti.ac.cn

Abstract

Existing tracking-by-detection Multi-Object Tracking methods mainly rely on associating objects with tracklets using motion and appearance features. However, variations in viewpoint and occlusions can result in discrepancies between the features of current objects and those of historical tracklets. To tackle these challenges, this paper proposes a novel Spatial-Temporal Tracklet Graph Matching paradigm (STAR). The core idea of STAR is to achieve long-term, reliable object association through the association of Tracklet Clips (TCs). TCs are segments of confidently associated multi-object trajectories, which are linked through graph matching. Specifically, STAR initializes TCs using a Confident Initial Tracklet Generator (CITG) and constructs a TC graph via Tracklet Clip Graph Construction (TCGC). In TCGC, each object in a TC is treated as a vertex, with the appearance and local topology features encoded on the vertex. The vertices and edges of the TC graph are then updated through message propagation to capture higher-order features. Finally, a Tracklet Clip Graph Matching (TCGM) method is proposed to efficiently and accurately associate the TCs through graph matching. STAR is model-agnostic, allowing for seamless integration with existing methods to enhance their performance. Extensive experiments on diverse datasets, including MOTChallenge, DanceTrack, and VisDrone2021-MOT, demonstrate the robustness and versatility of STAR, significantly improving tracking performance under challenging conditions. The code is available at https://github.com/baixuewei430-dotcom/STAR.

1 Introduction

Multi-object tracking (MOT) is a longstanding task in computer vision [1, 2, 3, 4], generally divided into two paradigms: tracking-by-detection (TBD) and joint detection-and-tracking (JDT). Currently, TBD [5, 6, 7, 8, 9] generally outperforms JDT in terms of accuracy. The core task in TBD involves effectively extracting object features and designing accurate association strategies to assign stable IDs to the same object. However, existing methods still encounter challenges in feature extraction and data association in difficult scenarios, such as crowded environments or frequent occlusions.

*Dr. Wang is supported in part by the National Natural Science Foundation of China Grant No.61972404, Public Computing Cloud, Renmin University of China, and the Blockchain Lab. School of Information, Renmin University of China.
†Dr. Li is supported in part by the National Natural Science Foundation of China Grant No.12071478.
‡Dr. Ping is supported by the Beijing Natural Science Foundation under Grant 4244076.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Reliable object association in crowded or occluded scenarios poses challenges for both feature extraction and association. Existing methods utilize a variety of features, including object appearance [10, 11, 12, 13], motion [14, 15, 16, 17, 18], temporal [19, 20, 21], and neighborhood topology features [22, 23, 24, 25]. However, these features may not be sufficiently distinctive in such challenging environments. For association, methods mainly employ bipartite matching [26, 27, 28, 29, 30] and graph matching [31, 32]. While graph matching provides higher accuracy, its computational cost increases rapidly with the number of objects and is highly reliant on the distinctiveness of features.

To address the challenges of object association in crowded and occluded scenarios, we propose a novel **S**patial-**T**emporal tr**A**cklet g**R**aph matching paradigm (STAR). The key aspect of STAR is its ability to maintain reliable long-term object associations, even when objects are occasionally obstructed. The central idea is to enhance feature distinctiveness by utilizing Tracklet Clips instead of individual object instances. A Tracklet Clip (TC) refers to a segment of confidently associated trajectories of multiple objects, which captures the appearance, spatial, and temporal features of an object, making it more distinctive in the feature space. Additionally, we further enhance the features of TCs by integrating topology information and high-order features through graph neural networks.

More specifically, STAR consists of three components. (1) The Confident Initial Tracklet Generator (CITG) uses a dynamically adjusted IoU-based method to produce initial tracklet segments from the input video, ensuring consistency by adaptively separating tracklets of the same object during occlusion. (2) The Tracklet Clip Graph Construction component rearranges tracklet segments by timestamps, divides them into tracklet clips (TCs), and converts them into feature graphs. Each tracklet within a TC is treated as a vertex, while the relationships between these vertices are modeled as edges. Vertex features integrate appearance, topology, and temporal information, with message propagation used to capture higher-order features. (3) The Tracklet Clip Graph Matching (TCGM) module enables fast and accurate TC association through graph matching. The main contributions of this work are summarized as follows:

1. STAR innovatively reformulates the object association problem into the construction and matching of Tracklet Clip feature graphs, thus effectively addressing occlusions.

2. The proposed Tracklet Clip Graph Construction (TCGC) method enhances distinctiveness and robustness of each object's feature by constructing multi-object spatial-temporal feature graphs.

3. Our Tracklet Clip Graph Matching (TCGM) approach employs tracklet-level graph matching to not only enhance matching accuracy but also overcome the efficiency bottlenecks associated with traditional frame-level graph matching.

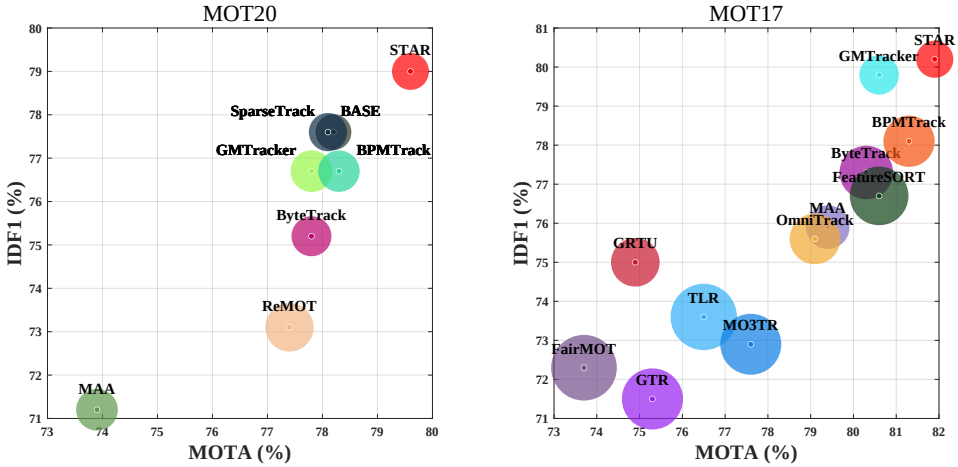

Figure 1: Comparison on MOT17 and MOT20. The x-axis represents MOTA, the y-axis represents IDF1, and the bubble size indicates the number of ID Switches (smaller is better).

2 Related Work

MOT has gained significant attention across industry and academia. We review relevant methods focusing on feature extraction and data association approaches.

2.1 Feature Extraction

Features fall into four categories: motion, appearance, temporal, and topology. We examine how existing methods address feature discontinuities caused by viewpoint changes.

Motion Feature. Basic motion features (position, velocity, bounding box dimensions) become unreliable with irregular camera movement. OCSORT [33] modifies the Kalman filter [34] to prioritize detection results. StrongSORT [35] and MAT [36] use regression to connect fragmented trajectories. While BoT-SORT [11] and MAA [15] compensate for camera motion through coordinate transformations, they remain computationally expensive and struggle in crowded scenes.

Appearance Feature. Deep learning models extract high-dimensional appearance vectors [37] that complement motion features. FairMOT [10] enhances feature extraction, while SimpleTrack [38] and BoT-SORT [11] optimize feature combinations. Xie et al. [39] develop region-based networks for fine-grained features. However, these methods perform poorly with UAV footage where object textures are less distinct and undergo perspective deformation.

Temporal Feature. These features model frame-to-frame dependencies. CenterTrack [19] and STTA [20] employ temporal attention, while McLaughlin et al. [40] use recurrent networks. Zhang et al. [21] propose orderless representations for better temporal modeling. These approaches, however, are sensitive to occlusions and often neglect multi-object relationships.

Topology Feature. Graph-based structures capture object relationships. GSM [22] builds directed graphs based on relative positions. GTAN [23] creates graphs between detections and trajectories but overlooks intra-frame relationships. [25] introduces topology features that remain stable under viewpoint changes but often ignore temporal information.

Existing methods underutilize tracklet-level features, limiting their effectiveness against occlusions and viewpoint changes.

2.2 Data Association

Data association connects current object features with previous tracklet features through bipartite or graph matching.

Bipartite Matching. This approach treats association as a linear assignment problem. ByteTrack [41] uses thresholding while retaining low-confidence detections. This two-stage association has become standard (used in [26, 27, 17, 28]). Some methods extend this approach: [29] uses three-stage association, while LG-Track [30] employs four stages. However, multi-stage approaches that don't consider all tracklet-detection pairs simultaneously can introduce identity switches.

Graph Matching. This approach formulates data association as a graph problem, where vertices represent objects or tracklets and edges capture their relationships. GM [31] was the first to apply this method in MOT, significantly improving association accuracy. GPM [9] pioneers the abstraction of the multi-object tracking problem into frame-level point set matching. SuperGlue [42] combines graph matching with deep learning but mainly focuses on keypoint relationships, neglecting important intra-frame connections. GMTracker [32] enhances convergence speed by replacing the Sinkhorn layer with a graph matching network. However, graph matching methods have notable limitations: computational costs increase rapidly with the number of objects, and accuracy is highly reliant on the distinctiveness of features, which restricts their scalability for large-scale MOT applications.

To address these challenges, we propose STAR, which utilizes spatial-temporal information to create distinctive and robust TC feature graphs, while efficiently reducing computational bottlenecks and maintaining accuracy through effective TC feature graph matching.

3 Methodology

3.1 Problem Definition and Overview

Given an input video sequence of L frames, the detected objects in frame t are represented as $\mathbb{I}^t = \{o_1, o_2, \ldots, o_{n_t}\}$, where n_t is the number of objects. Each detection o_i is a tuple $o_i = \{pos_i, \mathbf{f}_i, score_i, t_i\}$, with pos_i as the position, $\mathbf{f}_i$ as the appearance feature, $score_i$ as the confidence score, and t_i as the timestamp. The goal of multi-object tracking (MOT) is to generate trajectories for all objects by matching detections that refer to the same object, represented as $T^* = \{T_1, T_2, \ldots, T_n\}$. An overview of STAR is shown in Figure 2. The detection set is processed by the Confident Initial Tracklet Generator (CITG) to produce Confident Initial Tracklets (CITs), denoted as $G = \{g_1, g_2, \ldots, g_k\}$. It is important to ensure that CITs are generated with a strict object association strategy, so each CIT corresponds to the same object. Due to occlusions, a single object's trajectory may be split into multiple CITs, leading to a total number k of CITs that is typically greater than or equal to n. MOT is then reframed as matching these CITs to produce the final object trajectories, denoted as $G^* = \{g_1^*, g_2^*, \ldots, g_n^*\}$.

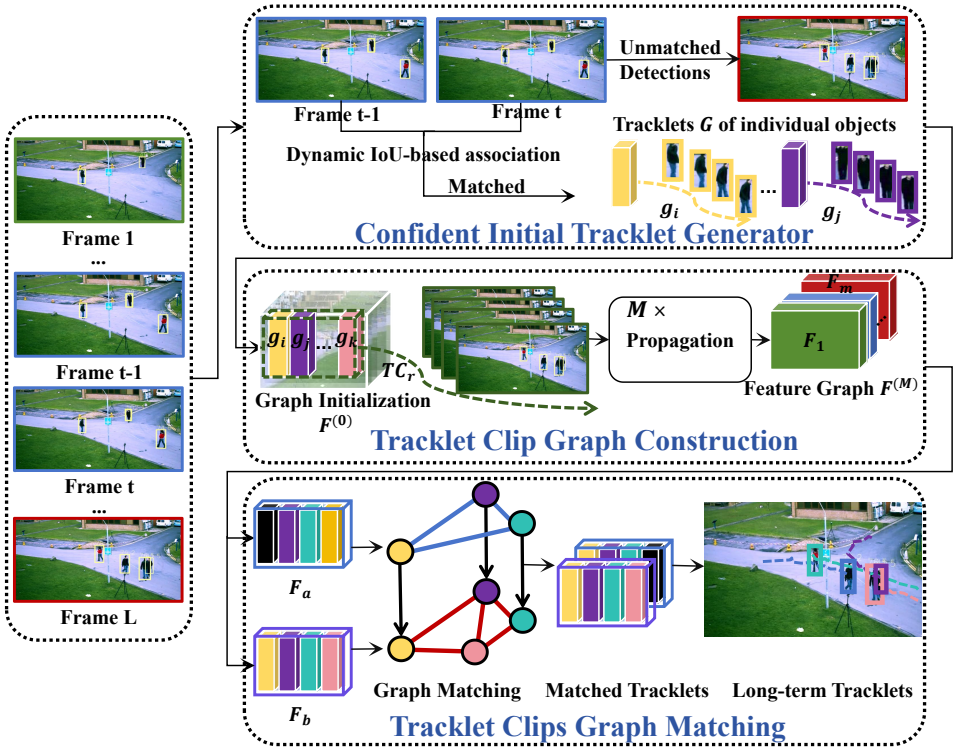

Figure 2: Overview of STAR. STAR consists of three essential components. CITG generates reliable initial tracklets G. The TCGC produces TCs and constructs robust spatial-temporal feature graphs $F^{(M)}$, where each object in a TC is represented as a vertex, with black cuboids serving as empty placeholders in F_a. TCGM facilitates efficient and accurate matching of object vertices across different TCs, ensuring robust associations.

3.2 Confident Initial Tracklet Generator (CITG)

A critical component of STAR is the CITG, which employs a strict association strategy. Tracklets are initialized based on object detections in the first frame and iteratively refined by calculating the IoU values between consecutive frames. If a detection matches multiple tracklets, none are selected. By prioritizing detections with higher IoU values, the generated tracklets correspond to the same object across frames. A tracklet is considered terminated if it remains unmatched for three consecutive frames, ensuring that only reliable and consistent tracklets are retained. To enhance generality while maintaining reliable associations, the IoU threshold is dynamically adjusted based on factors such as

object velocity, detection box size, and inter-frame time intervals. The reliably generated Confident Initial Tracklets (CITs) provide a solid foundation for the subsequent stages of TCGC and TCGM. Additional details can be found in Section A.2.

3.3 Tracklet Clip Graph Construction (TCGC)

Since CITs $G = \{g_1, g_2, \ldots, g_k\}$ have been obtained, two tasks will then be performed: generating Tracklet Clips (TCs) and constructing the TC graph, as shown in Figure 3.

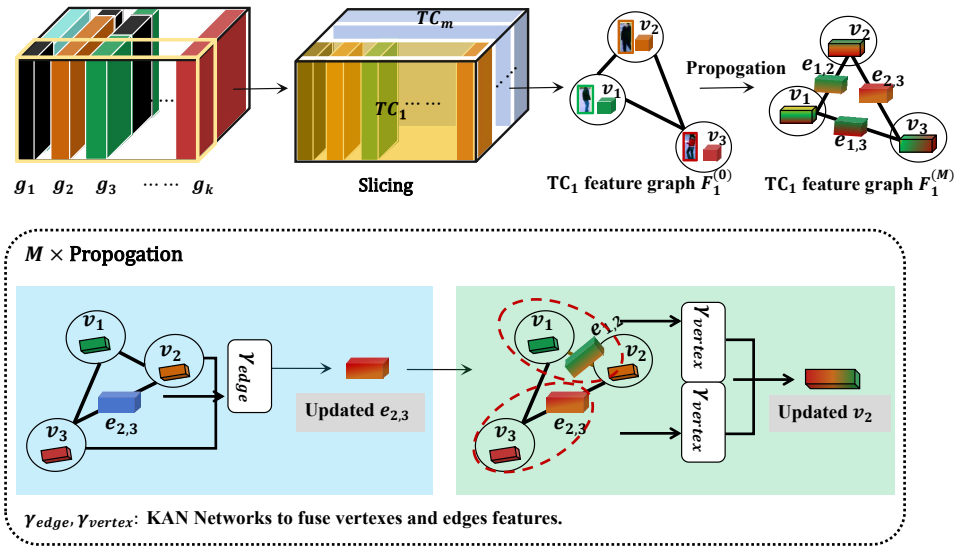

Figure 3: Overview of TCGC.

Generating Tracklet Clips. We arrange G by aligning in time and then cut them from time dimension, into fixed length Tracklet Clips (TC). Note that each TC contains trajectory segments of the objects detected within that short time interval. For object that haven't appeared in a frame in the TC, we use an empty placeholder which is represent by black cuboid in Figure 3. We denote the generated TCs by $TC = \{TC_1, TC_2, \ldots, TC_m\}$, and the interval length of each TC is N consecutive frames. A TC contains at most k objects. For each tracklet segment of object, we apply weighted pooling to the position and appearance features based on confidence scores, resulting in the aggregated feature $g_l^i = \{pos_l^i, \mathbf{f}_l^i\}$ for the lth object in the ith TC.

Constructing TC Graph. Each TC encodes the temporal features of the concurrently appeared objects during that interval. We then construct a graph for each TC, to further model the spatial, appearance, and topology features of the objects in TC. It involves two steps: **initialization** and **propagation**. Initialization builds a graph which treats the objects in the TC as vertices, and builds edges by distances and angles. Propagation refers to the message-passing process that extracts high-order topological features. The appearance feature is also considered, and the two types of features are concatenated. Since every TC is constructed using the same method, the superscript of TC is omitted. The constructed graph of each TC successfully encode the temporal, spatial and appearance features of the objects in the TC, making each object have more distinctive feature.

3.3.1 Initialization of a TC Graph

Appearance and topology features are utilized, which ensure stability under varying viewpoints. Due to dimensional differences between the topology and appearance features, we divide the TC feature graph $\mathbf{F} = (V, E)$ into two parts: $\mathbf{F}_{\text{app}}$ and $\mathbf{F}_{\text{topo}}$. Each object is treated as a vertex. Let $\mathbf{n}_k$ and $\tilde{\mathbf{n}}_k$ represent the topology and appearance features of the kth vertex, and $\mathbf{e}_{j,k}$ and $\tilde{\mathbf{e}}_{j,k}$ represent the topology and appearance features of the edge (j, k) respectively. Finally, we combine these two feature graphs as $\mathbf{F} = \mathbf{F}_{\text{app}} \cup \mathbf{F}_{\text{topo}}$, where $\mathbf{F}_{\text{topo}} = (\mathbf{n}, \mathbf{e})$ and $\mathbf{F}_{\text{app}} = (\tilde{\mathbf{n}}, \tilde{\mathbf{e}})$.

Vertex topology features are constructed using normalized lengths, angles, and positions relative to neighboring objects. We define the neighborhood set for vertex k as $\mathcal{N}_k = \{g_k \mid g_k \in$

TC and $\text{dist}(g_j, g_k) \leq r \times \min(h_0, w_0)\}$, where $r < 1$ is a constant, and h_0 and w_0 are the height and width of the input image, respectively. The function $\text{dist}(\cdot)$ represents the Euclidean distance between vertices. Then, the vertex topology features are as follows.

$$\begin{aligned} \mathbf{n}_j^{(0)} &= h_{\text{topo}}\left(\text{pos}_j \,\|\mathbf{l}_j\|\theta_j\right), \quad \text{where} \\ \text{pos}_j &= \left[\frac{x_j}{w_0}, \frac{y_j}{h_0}, \frac{w_j}{w_0}, \frac{h_j}{h_0}\right], \quad \mathbf{l}_j = \left[\frac{\text{dist}(g_j, g_k)}{\max(h_0, w_0)}\right] \quad g_k \in \mathcal{N}_j. \end{aligned} \tag{1}$$

$\|$ denotes concatenation, (x_j, y_j) are the bounding box center coordinates, and (w_j, h_j) are the bounding box dimensions. θ_j represents the angle between adjacent neighborhood objects. We concatenate position, distance, and angle information and pass it through h_{topo}, a Kolmogorov-Arnold Network (KAN) [43]. Edges are established between each vertex and its neighbors. The edge features are initialized based on positional and feature similarities.

$$\mathbf{e}_{j,k}^{(0)} = g_{\text{topo}}\left(\left[\frac{x_j - x_k}{w_0}, \frac{y_j - y_k}{h_0}, \text{IOU}\left(\text{pos}_j, \text{pos}_k\right), \log\left(\frac{w_j}{w_k}\right), \log\left(\frac{h_j}{h_k}\right), \cos(\mathbf{f}_j, \mathbf{f}_k)\right]\right). \tag{2}$$

The superscript (0) indicates the initial state. $\cos(\cdot)$ denotes cosine similarity, and the appearance features $\mathbf{f}_j \in \mathbb{R}^{512}$ are obtained from a Re-ID module [44]. The construction of $\mathbf{F}_{\text{app}}$ follows a similar approach of $\mathbf{F}_{\text{topo}}$, but using object's image appearance features. To balance the dimensions of topology and appearance features, we use a three-layer KAN to reduce the dimension for vertex features in $\mathbf{F}_{\text{app}}$ from 512 to 128.

3.3.2 Propagation

High-order features are obtained through iterative message passing. This process is executed for M iterations, with messages passed from vertices to edges and then from edges to vertices. The features of two connected vertices, $\mathbf{n}_j$ and $\mathbf{n}_k$, are first fused with the corresponding edge feature $\mathbf{e}_{j,k}$. Then, each vertex $\mathbf{n}_j$ aggregates messages from its neighboring edges and incident vertices.

$$\begin{aligned} (v \to e) : \mathbf{e}_{j,k}^{(m)} &= \mathbf{e}_{j,k}^{(m-1)} + \gamma_{\text{edge}}\left(\left(\mathbf{n}_j^{(m-1)} + \mathbf{n}_k^{(m-1)}\right) \| \,\text{KAN}\left(\mathbf{e}_{j,k}^{(m-1)}\right)\right) \\ (e \to v) : \mathbf{n}_j^{(m)} &= \eta^{(Q)}\left(\mathbf{n}_j^{(m-1)}, \oplus_{\mathbf{n}_k \in \mathcal{N}_j} \gamma_{\text{vertex}}\left(\mathbf{n}_j^{(m-1)}, \mathbf{e}_{j,k}^{(m)}\right)\right) \end{aligned} \tag{3}$$

Aggregating vertex features and neighboring edge features through three levels of fusion. (1) Fusion of Neighboring Vertex and Edge Features using the aggregation function γ_{vertex}, which employs a KAN and a ReLU activation function.(2) Aggregation of Neighboring Features using a permutation-invariant and differentiable aggregation function $\oplus$. (3) Final Vertex Feature Update using the function η^m.The update step $\eta(\cdot)$ is based on the message normalization proposed in [45].

$$\eta\left(\mathbf{n}_i, \mathbf{m}_i\right) = \text{KAN}\left(\mathbf{n}_i + s \cdot \|\mathbf{n}_i\|_2 \frac{\mathbf{m}_i}{\|\mathbf{m}_i\|_2}\right) \tag{4}$$

where s is a learnable factor, and $\|\cdot\|_2$ denotes the L2 norm. This step ensures that updated vertex features remain well-scaled and balanced. The same propagation process is applied to $\mathbf{F}_{\text{app}}$, resulting in aggregated high-order features $\mathbf{n}_i^{(M)}$ and $\tilde{\mathbf{n}}_i^{(M)}$. Finally, the vertex feature v_i of the graph $\mathbf{F}$ is obtained by concatenating these features. This process ensures that the final vertex features encode rich spatial-temporal information, enabling robust associations in subsequent tasks.

3.4 Tracklet Clip Graph Matching (TCGM)

After each TC graph has been constructed. The next step is to perform matching between different TCs. Our approach TCGM involves inputting the existing graph (including both vertex and edge features) into a differentiable graph matching layer [32], which produces the matching output. The core component of Tracklet Clip Graph Matching (TCGM) is the **differentiable graph matching layer**. This layer optimizes the matching process between detections and tracklets by solving a Quadratic Programming (QP) problem [32]. The result is an optimal matching score vector x, reshaped into a matrix form $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$ to generate the matching score map. To ensure effective training, the gradients of the graph matching layer are computed using the KKT conditions, aided by the implicit function theorem [32]. TCGM leverages spatial-temporal information to establish

robust associations between tracklets, significantly reducing identity switches caused by occlusions. By modeling interactions between tracklets within a learnable and differentiable framework, TCGM improves the accuracy of the association stage, resulting in precise and reliable trajectory generation for MOT tasks.

Complexity Analysis. Previous graph matching methods construct graphs for individual objects in each frame, resulting in $n_{\text{graph}} = n \times \mathcal{L}$ graphs, where n is the number of objects and $\mathcal{L}$ is the number of frames. The complexity of graph matching algorithms typically ranges from $O(n_{\text{graph}}^3)$ to $O(n_{\text{graph}}^2 \log n_{\text{graph}})$. By proposing TC graphs, we reduce the number of graphs to approximately $\frac{2}{N \times k}$ of the original count, remarkably improving the efficiency of graph matching.

3.5 Loss

To train the differentiable graph matching layer, we use a weighted binary cross-entropy loss function.

$$\mathcal{L}_{\text{track}} = \frac{-1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (N_2 - 1)\, y_{j,k} \log\left(\hat{y}_{j,k}\right) + \left(1 - y_{j,k}\right) \log\left(1 - \hat{y}_{j,k}\right), \tag{5}$$

where $\hat{y}_{j,k}$ is the predicted matching score between tracklet g_j and tracklet g_k, and $y_{j,k}$ is the ground truth indicating whether tracklet g_j belongs to tracklet g_k. $(N_2 - 1)$ is the weight used to balance the contributions of positive and negative samples to the loss. Due to the QP-based formulation of graph matching, the resulting score map $\mathbf{X}$ tends to have a relatively smooth distribution. To sharpen this distribution and improve the focus on high-confidence matches, we apply a softmax function with temperature τ.

$$\hat{y}_{j,k} = \text{Softmax}\left(x_{j,k}, \tau\right) = \frac{\exp\left(x_{j,k}/\tau\right)}{\sum_{j=1}^{N_2} \exp\left(x_{j,k}/\tau\right)}, \tag{6}$$

where $x_{j,k}$ is the original matching score from the score map $\mathbf{X}$, and τ is the inverse temperature parameter. A smaller τ sharpens the distribution, while a larger τ smooths it. Moreover, the method is applicable in both public and private detection settings. For public detection, the provided detections are directly used for tracking. For private detection, the training data is utilized to fine-tune the private detector. The total loss includes both detection and tracklet matching loss.

$$\mathcal{L} = \mathcal{L}_{\text{detect}} + \mathcal{L}_{\text{track}}, \tag{7}$$

where $\mathcal{L}_{\text{detect}}$ denotes the object detection loss, which depends on the private detector.

4 Experiments

4.1 Datasets and Metrics

Datasets. We select a variety of challenging benchmarks. The MOTChallenge [46, 47] datasets feature diverse scenes, viewpoints, and weather conditions. MOT17 includes 14 videos (7 for training) with three detection types: DPM [6], Faster R-CNN [7], and SDP [5]. MOT20 focuses on crowded scenes with 8 videos (4 for training and 4 for testing) that employ Faster R-CNN [7] detections. DanceTrack [48] contains 100 videos of various group dances, while VisDrone2021-MOT [49] comprises 96 sequences with around 40,000 frames across five object categories, posing challenges like occlusions and varying lighting conditions. These datasets present common issues in Multiple Object Tracking (MOT), such as frequent occlusions, irregular movements, and similar appearances, facilitating a comprehensive evaluation of STAR's robustness and generalization capabilities.

Metrics. The main metric for evaluating our method is the Higher Order Tracking Accuracy (HOTA) [50], which provides a balanced measure of both object detection accuracy (DetA) and association accuracy (AssA). Additionally, we also report MOTA [51], IDF1 [52], False Positives (FP), False Negatives (FN), Identity Switches (ID Sw.), and Frames Per Second (FPS) metrics.

4.2 Implementation Details

We employ several common data augmentation techniques, including random resize, crop, and color jitter. The input images are resized such that the shorter side is 800 pixels and the longer side is 1440

pixels. The proposed method is implemented using PyTorch. During training, $2N$ frames are sampled from each tracklet, resized to 256×128 pixels, and divided into two N-frame clips to enhance feature representation. The initial learning rate is set to 0.0003 and is reduced by a factor of 0.1 every 40 epochs. The model is trained for 150 epochs using the Adam optimizer with a mini-batch size of 32. Additional details and discussions can be found in Section A.4.

4.3 Comparison with State-of-the-Art Methods

We compare STAR with numerous previous methods on the MOTChallenge [46, 47], DanceTrack [48] and VisDrone2021-MOT [49] benchmarks, as shown in Table 1, Table 2, Table 3 and Table 4, respectively. YOLOX [8] is used as the detector to ensure a fair comparison. The best results are indicated in **bold**, and the second-best results are in underline.

MOTChallenge. We evaluate STAR in the private detection setting and compare its performance against several state-of-the-art algorithms. The results presented in Table 1 and Table 2 indicate that STAR consistently outperforms existing methods. Our tracklet-level paradigm effectively extracts and utilizes distinctive tracklet features, achieving improvements of 1.3% and 3.0% in HOTA on the MOT17 and MOT20 datasets, respectively, compared to SUSHI [53].

Table 1: Performance comparison with state-of-the-art methods on the MOT17 [46] test set.

Method	HOTA	MOTA	IDF1	ID Sw.
FairMOT [10]	59.3	73.7	72.3	3,303
GRTU [54]	62.0	74.9	75.0	1,812
TLR [55]	60.7	76.5	73.6	3,369
MAA [15]	62.0	79.4	75.9	1,452
GTR [56]	59.1	75.3	71.5	2,859
MO3TR [57]	60.3	77.6	72.9	2,847
ByteTrack [41]	63.1	80.3	77.3	2,196
GMTracker [32]	64.9	80.6	79.8	1,197
FeatureSORT [58]	64.2	80.6	76.7	2,637
SMILEtrack [17]	65.3	81.1	80.5	1,246
OmniTrack [13]	62.3	79.1	75.6	1,968
BPMTrack [59]	63.6	81.3	78.1	2,010
SparseTrack [60]	65.1	81.0	80.1	1,170
BoostTrack++[61]	66.6	80.7	82.2	1,062
OccluTrack+[62]	66.8	80.2	82.8	**951**
SUSHI[53]	66.5	81.1	**83.1**	1,149
STAR	**67.8**	**81.9**	80.2	1,057

Table 2: Performance comparison with state-of-the-art methods on the MOT20 [47] test set.

Method	HOTA	MOTA	IDF1	ID Sw.
FairMOT [10]	54.6	61.8	67.3	5,243
MAA [15]	57.3	73.9	71.2	1,331
ReMOT [63]	61.2	77.4	73.1	1,789
ByteTrack [41]	61.3	77.8	75.2	1,223
GMTracker [32]	62.9	77.8	76.7	1,331
BPMTrack [59]	62.3	78.3	76.7	1,314
BASE [16]	63.5	78.2	77.6	984
SparseTrack [60]	63.4	78.1	77.6	1,120
BoostTrack++[61]	66.4	77.7	82.0	762
OccluTrack+[62]	66.7	77.7	**82.7**	**429**
SUSHI[53]	64.3	74.3	79.8	706
STAR	**67.3**	**79.6**	79.0	1,047

DanceTrack. The complex scenarios characterized by frequent occlusions and irregular motion present significant challenges for tracking systems. Using the same pre-trained detector, STAR shows substantial improvements over SparseTrack [60], with improvement of 0.4% in HOTA, 0.5% in MOTA, 0.4% in AssA, and 0.2% in DetA as indicated in Table 3.

Table 3: Performance comparison with state-of-the-art methods on the DanceTrack [48] test set.

Method	HOTA	MOTA	IDF1	AssA	DetA
CenterTrack [19]	41.8	86.8	35.7	22.6	78.1
TraDes [12]	43.3	86.2	41.2	25.4	74.5
OCSORT [33]	55.1	**92.0**	54.6	38.3	**80.3**
FairMOT [10]	39.7	82.2	40.8	23.8	66.7
QDTrack [64]	54.2	87.7	50.4	36.8	80.1
GTR [56]	48.0	84.7	50.3	31.9	72.5
ByteTrack [41]	47.7	89.6	53.9	32.1	71.0
BoT-SORT [11]	54.7	91.3	56.0	37.8	79.6
SparseTrack [60]	55.5	91.3	**58.3**	39.1	78.9
STAR	**55.9**	91.8	57.9	**39.5**	79.1

These results highlight the considerable potential of our method in managing occlusion scenarios. Notably, even with a simple IoU distance association strategy, STAR achieves comparable or even superior performance relative to other methods.

VisDrone2021-MOT. This dataset presents even greater challenges due to frequent occlusions and varying lighting conditions. As shown in Table 4, STAR surpasses BoT-SORT by 2.9% in HOTA and 1.4% in IDF1, while outperforming OCSORT [33] by 2.4% in HOTA and 6.2% in IDF1. Furthermore, STAR demonstrates a 0.4% improvement in IDF1 over UGT while achieving a 1.6 FPS advantage, highlighting its efficiency in aerial tracking tasks.

Table 4: Performance comparison with state-of-the-art methods on the VisDrone2021-MOT [49] test set.

Method	HOTA	MOTA	IDF1	FN	FP	ID Sw.	FPS
DeepSORT [65]	36.9	34.4	46.7	110,989	21,077	1,784	18.5
ByteTrack [41]	40.7	39.5	50.4	105,518	16,257	1,581	**31.2**
BoT-SORT [11]	42.4	41.7	56.8	103,505	14,114	1,430	25.3
BiOU_Tracker [66]	40.2	40.7	48.8	103,188	15,794	2,029	28.5
MOTDT [67]	37.0	35.5	52.6	106,006	15,385	2,668	26.2
OCSORT [33]	42.9	41.6	52.0	132,279	22,019	2,859	15.4
StrongSORT [35]	36.6	33.3	42.5	185,503	12,214	1,980	12.1
UCMCTTrack [14]	37.1	28.1	38.4	150,590	9,244	7,231	17.8
QDTrack [64]	44.7	39.1	55.3	104,759	34,242	2,627	10.3
OUTrack [68]	34.0	35.0	44.5	115,570	25,276	3,335	17.4
FairMOT [10]	31.1	12.8	37.7	114,834	59,997	3,072	21.5
TrackFormer [69]	35.3	25.0	51.0	141,526	25,856	1,534	7.0
SGT [24]	43.6	39.2	54.8	110,652	**7,707**	951	27.9
UGT [25]	**45.5**	41.8	57.8	101,074	15,174	618	16.2
STAR	45.3	**41.9**	**58.2**	**100,832**	15,289	**602**	17.8

These results emphasize STAR's strong tracking stability and superior performance. Its effective modeling of distinctive spatial-temporal tracklet feature and the matching of TC graph enable STAR to achieve state-of-the-art results on MOTChallenge, DanceTrack, and VisDrone2021-MOT, establishing a new benchmark for multi-object tracking (MOT).

4.4 Ablation analysis

4.4.1 Effect of Tracklet Clip Graph Construction (TCGC)

Each TC consists of N consecutive frames. The analysis of the frame count (N) per TC is shown in Table 5 and Figure 4. It can be observed that $N = 6$ performs the best and $N = 4$ is the second-best. Although there was a slight improvement at $N = 6$, it was not significant. Therefore, we opted for $N = 4$ to ensure robust performance in challenging scenes.

For the sampling strategy with $N = 4$ and a stride length of L in TCGC, we experimented with various stride lengths to assess their impact on model performance. The detailed experimental results are presented in Table 6. The model achieved optimal performance at a stride length of $L = 1$. This smaller stride length allowed for more effective capture of continuity and changes over time, thus enhancing the model's temporal data processing capabilities. Furthermore, comparing stride lengths of $L = 2$ and $L = 1$ with a window size of $N = 4$, we found that although both settings yielded good performance, the $L = 2$ configuration provided high accuracy and improved efficiency.

Appearance similarity serves as a baseline for tracklet association. As shown in Table 7, distinctive and robust tracklet features significantly enhance tracking performance compared to models that rely solely on IoU distance for association. While Global Average Pooling (GAP) utilizes a simpler pooling strategy, our TCGC improves IDF1 from 61.9% to 69.2%. These results clearly highlight the importance of advanced tracklet feature extraction.

4.4.2 Effect of STAR

We demonstrate that integrating our proposed STAR method with existing tracking approaches enhances their performance, as summarized in Table 8. The evaluated baseline methods, DeepSORT [65], JDE [70], and CTracker [71], utilize IoU distance and frame-level features for tracklet

Table 5: Performance evaluation for various N values.

N	mAP	Top-1	Top-5	Top-10
2	91.14	89.52	98.89	98.89
4	92.52	91.84	98.89	98.89
5	92.55	91.92	98.89	98.89
6	92.60	91.95	98.89	98.89
7	92.43	91.36	98.89	98.89
8	92.04	90.88	98.89	98.89

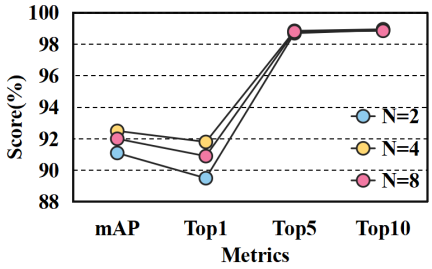

Figure 4: Effect of N in TCGC.

Table 6: Experimental results for various combinations of window size N and stride length L.

N	L	HOTA	MOTA	IDF1	ID Sw.
4	1	67.9	82.1	80.4	1,023
4	2	67.8	81.9	80.2	1,057
4	3	67.4	81.5	79.6	1,129
4	4	67.0	81.2	79.5	1,327

Table 7: Effect of Different Tracklet Description Methods.

Methods	IDF1	HOTA	MOTA	ID Sw.
Baseline	61.9	55.5	57.6	296
GAP	63.1	55.6	57.6	209
STTA [20]	68.6	58.7	57.7	133
TCGC	**69.2**	**59.0**	**57.9**	**125**

generation, with CTracker also incorporating topological information for data association. When comparing the baseline versions with their modified counterparts (DeepSORT*, JDE*, and CTracker*) that rely solely on IoU distance, we observe comparable MOTA scores but a significant drop in IDF1 (from 62.0% to 55.0%). This decline highlights IoU distance limitations in maintaining identity consistency across long trajectories and emphasizes the necessity for advanced feature extraction and data association techniques. Integrating STAR with these methods boosts performance, improving IDF1 by 3.2%, 3.5%, and 8.0% for DeepSORT, JDE, and CTracker, respectively. These results demonstrate STAR's effectiveness in enhancing identity preservation and association accuracy. In conclusion, STAR is model-agnostic and can be seamlessly integrated into various tracking frameworks.

Table 8: Performance of Adding STAR upon Existing Methods on the MOT16 Training Dataset.

Methods	MOTA	IDF1	HOTA	FP	FN	ID Sw.
Deepsort [65]	56.9	62.0	51.3	13,227	33,454	**932**
Deepsort*	56.6	55.0	48.3	**10,433**	35,627	1,883
Deepsort*+STAR	**58.8**	**65.2**	**54.3**	13,041	**31,105**	1,315
JDE [70]	73.1	68.9	55.1	6,593	21,788	1,312
JDE*	73.0	61.9	53.6	**6,185**	22,296	1,330
JDE*+STAR	**73.5**	**72.4**	**55.3**	6,871	**21,350**	**1,125**
CTracker [71]	76.2	68.6	61.1	**2,149**	23,188	912
CTracker*	76.2	68.6	61.1	**2,149**	23,188	912
CTracker*+STAR	**78.8**	**76.6**	**66.0**	3,981	**18,960**	**540**

5 Limitation and Conclusion

Unlike previous approaches that overlook historical information and temporal continuity, STAR extracts and effective leverages distinctive tracklet features through TC to address occlusion challenges. The framework consists of three core components. CITG efficiently generates reliable CITs, TCGC produces discriminative TC feature graphs by exploring spatial-temporal information within tracklets, and TCGM uses graph matching to enhance association accuracy and improve efficiency. Together, these components produce high-integrity trajectories and achieve state-of-the-art performance across three widely used benchmarks, demonstrating STAR's effectiveness and robustness. Despite its strong performance, the method faces efficiency limitations. Future work will focus on designing an end-to-end tracking framework to further enhance STAR's robustness and applicability in real-world multi-object tracking (MOT) scenarios.

References

[1] Lingyu Kong, Zhiyuan Yan, Yidan Zhang, Wenhui Diao, Zining Zhu, and Lei Wang. Cftracker: Multi-object tracking with cross-frame connections in satellite videos. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–14, 2023.

[2] Qibin He, Xian Sun, Zhiyuan Yan, Beibei Li, and Kun Fu. Multi-object tracking in satellite videos with graph-based multitask modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[3] Xiangkai Xu, Zhejun Feng, Changqing Cao, Chaoran Yu, Mengyuan Li, Zengyan Wu, Shubing Ye, and Yajie Shang. Stn-track: Multiobject tracking of unmanned aerial vehicles by swin transformer neck and new data association method. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:8734–8743, 2022.

[4] Qibin He, Xian Sun, Zhiyuan Yan, Bing Wang, Zicong Zhu, Wenhui Diao, and Michael Ying Yang. Ast: Adaptive self-supervised transformer for optical remote sensing representation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 200:41–54, 2023.

[5] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[6] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[8] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[9] Xuewei Bai, Yongcai Wang, Peng Wang, Chunxu Li, Shuo Wang, Xudong Cai, and Deying Li. A geometric and hypothesis-based method for low-overlap, sparse, and featureless point set matching. *ACM Trans. Sen. Netw.*, 21(5), September 2025.

[10] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International journal of computer vision*, 129:3069–3087, 2021.

[11] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

[12] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12352–12361, 2021.

[13] Junke Wang, Zuxuan Wu, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, and Yu-Gang Jiang. Omnitracker: Unifying visual object tracking by tracking-with-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[14] Kefu Yi, Kai Luo, Xiaolei Luo, Jiangui Huang, Hao Wu, Rongdong Hu, and Wei Hao. Ucmctrack: Multi-object tracking with uniform camera motion compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6702–6710, 2024.

[15] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 133–142, 2022.

[16] Martin Vonheim Larsen, Sigmund Johannes Ljosvoll Rolfsjord, Daniel Gusland, Jörgen Ahlberg, and Kim Mathiassen. Base: Probably a better approach to visual multi-object tracking. *Proceedings Copyright*, 110:121, 2024.

[17] Yu-Hsiang Wang, Jun-Wei Hsieh, Ping-Yang Chen, Ming-Ching Chang, Hung-Hin So, and Xin Li. Smiletrack: Similarity learning for occlusion-aware multiple object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5740–5748, 2024.

[18] Yuhao Guo, Yicheng Li, Shaohua Wang, Kecheng Sun, Mingchun Liu, and Zihan Wang. Pedestrian multi-object tracking combining appearance and spatial characteristics. *Expert Systems with Applications*, page 126772, 2025.

[19] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European conference on computer vision*, pages 474–490. Springer, 2020.

[20] Sisi You, Hantao Yao, Bing-Kun Bao, and Changsheng Xu. Multi-object tracking with spatial-temporal tracklet association. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(5):1–21, 2024.

[21] Le Zhang, Zenglin Shi, Joey Tianyi Zhou, Ming-Ming Cheng, Yun Liu, Jia-Wang Bian, Zeng Zeng, and Chunhua Shen. Ordered or orderless: A revisit for video based person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1460–1466, 2020.

[22] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In *IJCAI*, pages 530–536, 2020.

[23] Lv Jianfeng, Yu Zhongliang, Liu Yifan, and Sun Guanghui. Gtan: graph-based tracklet association network for multi-object tracking. *Neural Computing and Applications*, 36(8):3889–3902, 2024.

[24] Jeongseok Hyun, Myunggu Kang, Dongyoon Wee, and Dit-Yan Yeung. Detection recovery in online multi-object tracking with sparse graph tracker. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4850–4859, 2023.

[25] Chenwei Deng, Jiapeng Wu, Yuqi Han, Wenzheng Wang, and Jocelyn Chanussot. Learning a robust topological relationship for online multi-object tracking in uav scenarios. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[26] Kai Ren, Chuanping Hu, and Hao Xi. Rlm-tracking: online multi-pedestrian tracking supported by relative location mapping. *International Journal of Machine Learning and Cybernetics*, 15(7):2881–2897, 2024.

[27] Yi Li, Youyu Liu, Chuanen Zhou, Dezhang Xu, and Wanbao Tao. A lightweight scheme of deep appearance extraction for robust online multi-object tracking. *The Visual Computer*, 40(3):2049–2065, 2024.

[28] Duy Cuong Bui, Ngan Linh Nguyen, Anh Hiep Hoang, and Myungsik Yoo. Camtrack: a combined appearance-motion method for multiple-object tracking. *Machine Vision and Applications*, 35(4):62, 2024.

[29] Ye Li, Lei Wu, Yiping Chen, Xinzhong Wang, Guangqiang Yin, and Zhiguo Wang. Motion estimation and multi-stage association for tracking-by-detection. *Complex & Intelligent Systems*, 10(2):2445–2458, 2024.

[30] Jinlong Yang, Yandeng Ban, and Jianjun Liu. Local many-to-many matching via roi feature decomposition for multi-object tracking. *Signal, Image and Video Processing*, 18(10):6573–6589, 2024.

[31] Weiming Hu, Xinchu Shi, Zongwei Zhou, Junliang Xing, Haibin Ling, and Stephen Maybank. Dual l 1-normalized context aware tensor power iteration and its applications to multi-object tracking and multi-graph matching. *International Journal of Computer Vision*, 128:360–392, 2020.

[32] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: A practical paradigm for data association. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[33] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, Shuyuan Zhu, and Weiming Hu. Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, 31:3182–3196, 2022.

[34] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[35] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 25:8725–8737, 2023.

[36] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. Mat: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022.

[37] Yanbing Chen, Ke Wang, Hairong Ye, Lingbing Tao, and Zhixin Tie. Person re-identification in special scenes based on deep learning: A comprehensive survey. *Mathematics*, 12(16):2495, 2024.

[38] Jiaxin Li, Yan Ding, Hua-Liang Wei, Yutong Zhang, and Wenxiang Lin. Simpletrack: Rethinking and improving the jde approach for multi-object tracking. *Sensors*, 22(15):5863, 2022.

[39] Qiyang Xie, Daiying Zhou, Rui Tang, and Hao Feng. A deep cnn-based detection method for multi-scale fine-grained objects in remote sensing images. *IEEE Access*, 12:15622–15630, 2024.

[40] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. Recurrent convolutional network for video-based person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2016.

[41] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022.

[42] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[43] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[44] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3702–3712, 2019.

[45] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

[46] Anton Milan. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[47] P Dendorfer. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[48] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20993–21002, 2022.

[49] Guanlin Chen, Wenguan Wang, Zhijian He, Lujia Wang, Yixuan Yuan, Dingwen Zhang, Jinglin Zhang, Pengfei Zhu, Luc Van Gool, Junwei Han, et al. Visdrone-mot2021: The vision meets drone multiple object tracking challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2839–2846, 2021.

[50] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021.

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

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

[53] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22877–22887, 2023.

[54] Shuai Wang, Hao Sheng, Yang Zhang, Yubin Wu, and Zhang Xiong. A general recurrent tracking framework without real data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13219–13228, 2021.

[55] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3876–3886, 2021.

[56] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8771–8780, 2022.

[57] Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, Ian Reid, and Hamid Rezatofighi. 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, 2022.

[58] Hamidreza Hashempoor, Rosemary Koikara, and Yu Dong Hwang. Featuresort: essential features for effective tracking. *arXiv preprint arXiv:2407.04249*, 2024.

[59] Yan Gao, Haojun Xu, Jie Li, and Xinbo Gao. Bpmtrack: multi-object tracking with detection box application pattern mining. *IEEE Transactions on Image Processing*, 33:1508–1521, 2024.

[60] Zelin Liu, Xinggang Wang, Cheng Wang, Wenyu Liu, and Xiang Bai. Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[61] Vukašin Stanojević and Branimir Todorović. Boosttrack++: using tracklet information to detect more objects in multiple object tracking. *arXiv preprint arXiv:2408.13003*, 2024.

[62] Jianjun Gao, Yi Wang, Kim-Hui Yap, Kratika Garg, and Boon Siew Han. Occlutrack: Rethinking awareness of occlusion for enhancing multiple pedestrian tracking. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[63] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. Remot: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 106:104091, 2021.

[64] Tobias Fischer, Thomas E Huang, Jiangmiao Pang, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu. Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15380–15393, 2023.

[65] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[66] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4799–4808, 2023.

[67] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2018.

[68] Qiankun Liu, Dongdong Chen, Qi Chu, Lu Yuan, Bin Liu, Lei Zhang, and Nenghai Yu. Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing*, 483:333–347, 2022.

[69] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022.

[70] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European conference on computer vision*, pages 107–122. Springer, 2020.

[71] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 145–161. Springer, 2020.

[72] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018.

[73] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 941–951, 2019.

[74] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International conference on machine learning*, pages 4364–4375. PMLR, 2020.

[75] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6247–6257, 2020.

[76] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2452, 2021.

[77] Daniel Stadler and Jurgen Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10958–10967, 2021.

A Appendix

A.1 Overview

In the supplementary material, we primarily:

1. Provide more details of CITG in Appendix A.2.
2. Provide a overview of TCGM in Appendix A.3.
3. State more experimental details in Appendix A.4.
4. Provide additional experimental results in Appendix A.5.

A.2 Construction of CITG

A short-term trajectory continues to be associated with the detection in the next frame until it no longer matches any detection. The Confident Initial Tracklet Generator utilizes the Intersection over Union (IoU) metric to associate objects by calculating the overlap between detection boxes. Below is the detailed process.

Trajectory Initialization: In the first frame of the video, each detection box is initialized as an independent short-term trajectory.

IoU Calculation: For each trajectory in the current frame, the IoU with all detection boxes in the subsequent frame is calculated. The IoU value, ranging from 0 to 1, indicates the degree of overlap, with higher values denoting greater overlap.

Trajectory Matching: Trajectories are matched to detection boxes based on IoU values. A trajectory and a detection box from the next frame are considered to belong to the same object if their IoU exceeds a specific threshold. If a detection box matches multiple trajectories, none are selected. Unmatched detection boxes are initialized as new trajectories.

Dynamic Adjustment of IoU Threshold: To accommodate varying motion characteristics of targets, a dynamic adjustment method for the IoU threshold is employed. This method considers multiple factors, including the target's motion speed (v), detection box size (A), and time interval between frames (Δt). These factors dynamically influence the IoU threshold.

Trajectory Update and Termination: Matched detection boxes are added to corresponding trajectories, updating their state (such as location and timestamp). Trajectories that remain unmatched for more than three consecutive frames are terminated. Unmatched detection boxes initiate new trajectories.

Output Short-Term Trajectories: The aforementioned steps generate a set of short-term trajectories that capture the preliminary motion trajectories of all targets in the video. These trajectories form the basis for further trajectory association and long-term trajectory generation.

A.3 Overview of TCGM

A.4 More Implementation Details

We train our model on 24 NVIDIA RTX 2080Ti GPUs. During testing, the entire tracklet is used as input, with every N=4 frames treated as a clip.

MOTChallenge. In the MOT16 dataset [46], only objects with a visible ratio greater than 0.3 are selected, resulting in 517 training identities, 438 gallery identities, and 429 query identities. The total number of training videos is 2,065, and there are 2,173 testing videos, with each ground truth trajectory divided into four variable-length tracklets.

VisDrone2021-MOT. The VisDrone2021-MOT-train set, which consists of 56 sequences, is used for training, while the VisDrone2021-MOT-test-dev set, containing 17 sequences, is used for testing. During evaluation, a single tracklet per identity serves as the query, with the remaining tracklets in the gallery.

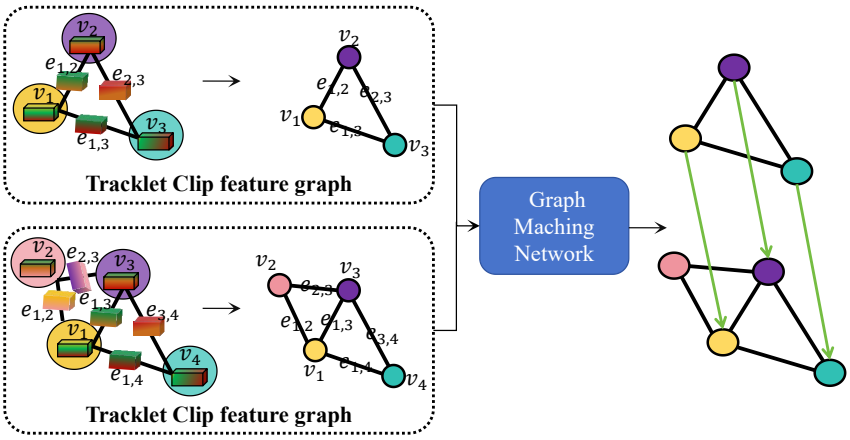

Figure 5: Overview of TCGM.

UAVDT. STAR is also evaluated on the UAVDT dataset [72], which provides diverse scenarios and challenges. The UAVDT dataset contains 50 sequences with over 80,000 frames, capturing various situations that present frequent occlusions and significant camera motion.

A.5 More Comparisons with State-of-the-Art Methods

We compare the proposed STAR method with traditional **TBD** approaches using Tracktor to refine detections in the public detection setting. As shown in Table 9, STAR outperforms existing methods on most evaluation metrics, achieving significant improvements in MOTA, IDF1, and HOTA across the MOT15, MOT16, MOT17, and MOT20 datasets. The tracklet-level approach of STAR integrates tracklet information during feature extraction and data association, which facilitates more robust tracking.

Table 9: Performance comparison of Public Detection with state-of-the-art methods on the MOTChallenge test set.The best results are indicated in **bold**, and the second-best results are in underline.

Datasets	Methods	HOTA	MOTA	IDF1	FP	FN	ID Sw.
MOT15	Tracktor [73]	37.6	46.6	47.6	**4,624**	26,896	1,280
	LFP [74]	43.8	47.2	57.6	7,635	24,277	554
	MPNTrack [75]	45.0	51.5	58.6	7,620	21,780	**375**
	STAR	**45.8**	**55.8**	**59.9**	5,983	**20,700**	475
MOT16	Tracktor [73]	44.6	56.2	54.9	2,394	76,844	617
	GSM [22]	45.9	57.0	58.2	4,332	73,543	475
	LFP [74]	49.6	57.5	64.1	4,249	72,868	**335**
	MPNTrack [75]	48.9	58.6	61.7	4,949	70,252	354
	LPC [76]	**51.7**	58.8	**67.6**	6,167	68,432	435
	MO3TR [57]	50.3	**64.2**	60.6	7,620	**56,761**	929
	TMOH [77]	50.7	63.2	63.5	3,122	63,376	635
	GMTracker [32]	48.9	55.9	63.9	**2,371**	77,545	531
	STAR	51.2	64.1	66.5	2,427	62,377	511
MOT17	Tracktor [73]	44.8	56.3	55.1	8,866	235,449	1,987
	GSM [22]	45.7	56.4	57.8	14,379	230,174	1,485
	LFP [74]	50.7	58.2	65.2	16,850	217,944	**1,022**
	MPNTrack [75]	49.0	58.8	61.7	17,413	213,594	1,185
	LPC [76]	51.5	59.0	**66.8**	23,102	206,948	1,122
	MO3TR [57]	49.6	63.2	60.2	21,966	**182,860**	2,841
	TMOH [77]	50.4	62.1	62.8	10,951	201,135	1,897
	GMTracker [32]	49.1	56.2	63.8	**8,719**	236,541	1,778
	STAR	**52.5**	**64.2**	66.5	8,971	190,636	1,994
MOT20	Tracktor [73]	42.1	52.6	52.7	**6,930**	236,680	**1,648**
	TMOH [77]	48.9	60.1	61.2	8,043	165,899	2,342
	STAR	**52.9**	**64.1**	**66.5**	39,357	**143,583**	2,825

For the UAVDT dataset, 40 sequences are randomly selected for training, while 10 sequences are designated for testing. As summarized in Table 10, STAR demonstrates superior performance on the UAVDT dataset, achieving HOTA, MOTA, and IDF1 scores of 63.4%, 71.4%, and 80.3%, respectively, surpassing all other methods. STAR outperforms BoT-SORT [11] by 2.0% in HOTA and 3.7% in MOTA, highlighting its effective modeling of topological relationships. Additionally, STAR exceeds UGT [25], the best-performing method on UAVDT, by 1.4 FPS in terms of computational efficiency.

Table 10: Performance comparison with state-of-the-art methods on the UAVDT [72] test set. The best results are indicated in **bold**, and the second-best results are in underline.

Method	HOTA	MOTA	IDF1	FN	FP	ID Sw.	FPS
DeepSORT [65]	62.0	68.5	78.6	20,035	4,008	**61**	20.3
ByteTrack [41]	62.2	68.8	78.8	20,010	3,796	102	32.1
BoT-SORT [11]	61.4	67.7	78.5	20,296	4,323	64	25.0
BiOU_Tracker [66]	62.9	70.3	79.6	17,405	5,224	79	**38.1**
MOTDT [67]	61.8	66.5	77.8	17,760	5,824	76	22.4
OCSORT [33]	59.8	69.5	74.5	18,246	6,480	249	13.3
StrongSORT [35]	58.3	49.3	72.6	**10,606**	28,032	134	10.6
UCMCTTrack [14]	54.1	61.0	65.9	25,888	**2,482**	984	15.4
QDTrack [64]	61.2	70.3	76.1	17,304	6,420	92	13.4
OUTrack [68]	58.6	64.3	66.5	18,193	7,826	240	18.2
FairMOT [10]	49.1	51.2	66.5	33,102	4,136	110	25.5
TrackFormer [69]	43.2	37.9	53.3	45,197	5,582	680	7.94
SGT [24]	58.2	57.8	77.0	11,658	20,598	102	30.0
UGT [25]	**63.6**	**71.6**	80.0	17,285	4,357	67	18.3
STAR	63.4	71.4	**80.3**	17,462	4,389	74	19.7

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction match theoretical and experimental results, and reflect the results may be generalized to other settings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation of this work is discuss in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All the equation in the paper have been numbered and cross-referenced.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper disclose the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This paper provide instructions to reproduce the main experimental results in Section 4.2 and Section A.4.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper provide instructions to train and test details necessary to understand the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification:This paper reports statistical results, which is staticed by MOTMetrics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provide information on the computer resources in Section A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper may have positive societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are well documented and it is provided alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.