Multi-Object Tracking in Satellite Videos With Graph-Based Multitask Modeling

Qibin He, *Graduate Student Member, IEEE*, Xian Sun, *Senior Member, IEEE*, Zhiyuan Yan, *Member, IEEE*, Beibei Li, and Kun Fu, *Senior Member, IEEE*

***Abstract*— Recently, satellite video has become an emerging means of earth observation, providing the possibility of tracking moving objects. However, the existing multi-object trackers are commonly designed for natural scenes without considering the characteristics of remotely sensed data. In addition, most trackers are composed of two independent stages of detection and reidentification (ReID), which means that they cannot be mutually promoted. To this end, we propose an end-to-end online framework, which is called TGraM, for multi-object tracking in satellite videos. It models multi-object tracking as a graph information reasoning procedure from the multitask learning perspective. Specifically, a graph-based spatiotemporal reasoning module is presented to mine the potential high-order correlations between video frames. Furthermore, considering the inconsistency of optimization objectives between detection and ReID, a multitask gradient adversarial learning strategy is designed to regularize each task-specific network. In addition, aiming at the data scarcity in this field, a large-scale and high-resolution Jilin-1 satellite video dataset for multi-object tracking (AIR-MOT) is built for the experiments. Compared with state-of-the-art multi-object trackers, TGraM achieves efficient collaborative learning between detection and ReID, improving the tracking accuracy by 1.2 multiple object tracking accuracy. The code and dataset will be available online (https://github.com/HeQibin/TGraM).**

***Index Terms*— Graph reasoning, multi-object tracking, multitask learning (MTL), satellite video.**

I. Introduction

MULTI-OBJECT tracking, aimed at detecting objects and associating them in time series, is essential in many fields, such as security monitoring, motion analysis, and traffic control [1], [2]. With the development of aerospace technology, spaceborne remote sensing has become an effective means of earth observation, especially the high-resolution satellite videos that can provide the possibility of tracking ground moving objects.

In recent years, several methods have been proposed for object tracking in satellite videos [4]–[11]. However, these methods mainly focus on a single object, which contradicts the multi-object characteristics of remotely sensed data, as shown in Fig. 1(a) and (b). High-resolution videos taken by satellites usually cover a wide area with multiple moving objects. Parallel tracking of multiple objects can help analyze video content and improve the efficiency of earth observation.

Although there are a few mature multi-object tracking methods in remote sensing, some cases in computer vision can provide a reference. Advanced online multi-object tracking methods can be roughly divided into two categories:

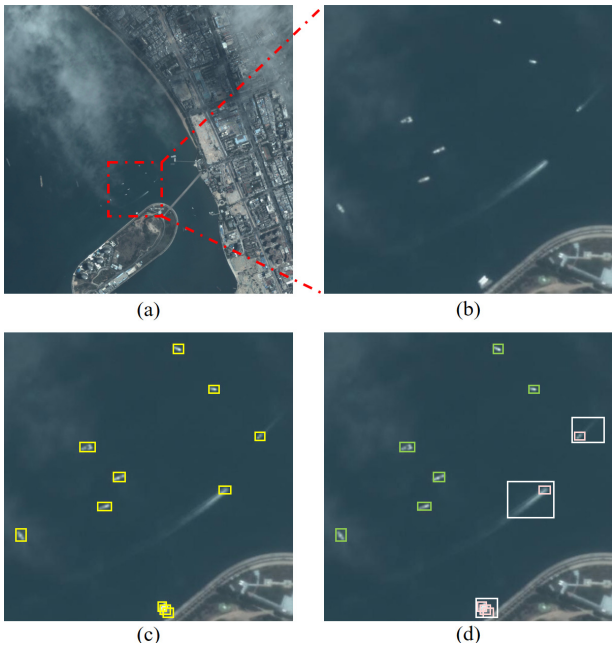

Fig. 1. Example of moving object detection and tracking in the AIR-MOT dataset. (a) Local area of Sanya in a frame of the satellite video. (b) Partially enlarged image of (a). Local area often contains multiple objects, but tracking methods in remote sensing mainly focus on single object. (c) Ground truth of moving object. (d) Result of [19], where the correct detections are in green, the missed detections are illustrated in pink and the false detections are in white (best viewed in color). Most previous multi-object trackers in computer vision only use spatial features and perform detection independently, making it difficult to detect objects with motion blur and dense distribution.

Manuscript received July 12, 2021; revised October 24, 2021, January 25, 2022, and February 9, 2022; accepted February 14, 2022. Date of publication February 16, 2022; date of current version March 31, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61725105 and in part by the National Major Project on High Resolution Earth Observation System of China under Grant GFZX0404120205. *(Corresponding author: Xian Sun.)*

Qibin He, Xian Sun, and Kun Fu are with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China, also with the Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China, also with University of Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: heqibin20@mails.ucas.ac.cn; sunxian@aircas.ac.cn; fukun@mail.ie.ac.cn).

Zhiyuan Yan is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China, and also with the Key Laboratory of Network Information System Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yanzy@aircas.ac.cn).

Beibei Li is with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Jilin 132000, China.

Digital Object Identifier 10.1109/TGRS.2022.3152250

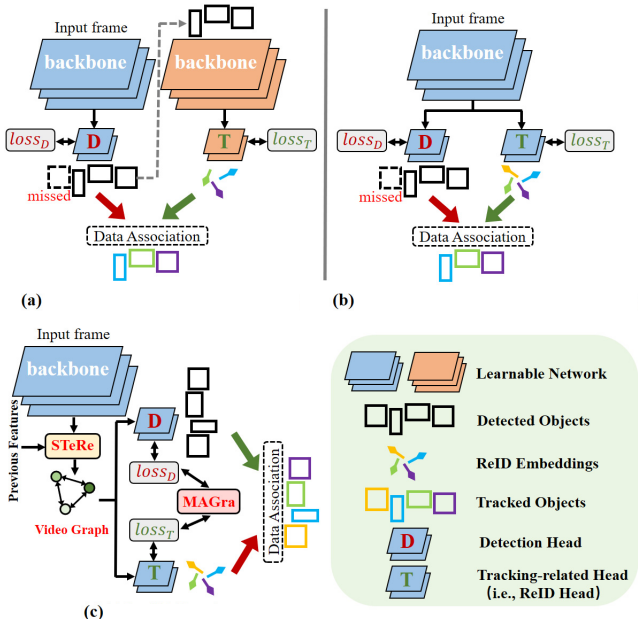

Fig. 2. Comparison of different online multi-object tracking pipelines. Our TGraM belongs to the JDT paradigm. Different from the traditional JDT methods, the proposed STeRe module in TGraM uses motion clues to improve detection, by modeling multitemporal features as an undirected graph. In addition, the designed MAGra strategy introduces adversarial learning to promote the similarity of the loss gradient distribution between detection and ReID tasks. (a) DBT. (b) JDT. (c) TGraM (ours).

detection-based-tracking (DBT) [12]–[15], [17], [25] and joint-detection-and-tracking (JDT) [18]–[26]. As shown in Fig. 2(a), the DBT paradigm treats detection and tracking as two independent tasks. It employs an off-the-shelf detector to generate detections and then applies another network to associate. This two-stage processing makes the DBT system inefficient and difficult to achieve real-time tracking. Since low latency is significant for spaceborne remote sensing, DBT is mostly not applicable. To settle this problem, the JDT methods are designed to perform detection and tracking simultaneously in a single-forward inference, as shown in Fig. 2(b).

However, directly applying the existing JDT methods to satellite videos will have some limitations: **1)** although most JDT methods share the backbone network, the tracking clues are not utilized during the detection stage, which may lead to poor performance in satellite videos. As shown in Fig. 1(c) and (d), the objects in satellite videos tend to be small, motion blur, and dense distribution due to the long-distance bird's view. Only relying on the spatial appearance features is insufficient for accurate detection. We believe that appropriate use of tracking clues (i.e., the predicted tracking offset of each object) will be a benefit to detection, and, in turn, reliable detection is the basis for a consistent and stable tracklet. **2)** The JDT method essentially models multi-object tracking as a multitask learning (MTL) problem, composed of detection and ReID tracking. But the common detection loss is not compatible with ReID loss in jointly training the shared network. The optimization objective of the training detection task is to expand interclass variance, while ReID pays more attention to intraclass variance [27]. Most existing JDT methods ignore such a phenomenon, leading to poor tracking performance to some extent. In the training phase, if the gradients of the two task losses are indistinguishable for the shared network, the problem can be effectively improved.

In this article, we propose a novel end-to-end online framework for accurate yet efficient multi-object tracking in satellite video, called **T**racking via **Gra**ph-based **M**ultitask Reasoning (TGraM). TGraM models multi-object tracking as a message reasoning-based graph information fusion process from an MTL perspective. By integrating tracking clues into the detection and designing a specially designed multitask training scheme, the above-mentioned two problems can be well-solved. Concretely, we propose a graph-based **s**patio**te**mporal **re**asoning (STeRe) module and a **m**ultitask **a**dversarial **gra**dient (MAGra) learning strategy. STeRe constructs a fully connected graph based on each video frame and performs graph reasoning on the stored explicit semantics to capture the matching similarity. Semantic similarity is treated as the motion clue, which assists the object feature propagation through message aggregation, thus forming spatiotemporal long-range dependencies. Then, MAGra conducts gradient adaptation for detection and ReID tasks in the training phase. After adaptation, the gradient tensor returned by each task loss function is indistinguishable, facilitating the collaborative learning of all shared layers.

In the STeRe module, the video frames are indicated as nodes, while the pairwise relations between two frames are expressed as the edge between the corresponding nodes. Graph convolution is applied to perform Laplacian smoothing on the graph. Such reasoning cannot only extract the joint semantics in the embedding space but also build spatiotemporal long-range dependencies. This guarantees that the input of detection and ReID incorporates temporal motion clues. Then, MAGra trains the gradient discriminator between the shared network and head network (i.e., detection and ReID); thus, the gradient distribution of each task tends to be similar, as shown in Fig. 2(c). The discriminator employs gradient backpropagation to pass the adversarial signal back to the main network to regularize its weights, similar to double backpropagation [28]. Therefore, the performance degradation caused by different learning objectives of detection and ReID task can be alleviated.

In summary, the main contributions are as follows.

1) We present a novel end-to-end online framework, called TGraM, for accurate yet efficient tracking in high-resolution satellite videos. Compared with other trackers, it achieves efficient collaborative learning between detection and ReID and realizes parallel real-time tracking for multiple objects.
2) The STeRe module is proposed to mine the potential high-order correlations through graph reasoning and construct spatiotemporal long-range dependencies. In this way, the tracking performance for tough scenarios (i.e., the objects with motion blur and dense distribution) can be improved effectively.
3) Considering the inconsistency of learning objectives between detection and ReID, the MAGra strategy is designed to eliminate discriminative information of gradient sources and regularize the task-specific network via adversarial signal.

4) To verify the effectiveness of the proposed framework, we build a large-scale and high-resolution satellite video dataset for multi-object tracking (i.e., AIR-MOT), including two types of objects: aircrafts and ships. AIR-MOT will be opened to the community, which is one of the earliest public datasets in this field.

The experimental results on AIR-MOT demonstrate that our TGraM achieves better tracking performance than previous methods, proving the superiority.

II. Preliminaries and Related Works

TGraM applies graph reasoning to construct spatiotemporal dependencies and models multi-object tracking from the perspective of MTL. In this section, we will review the relevant basic principles.

A. Graph Reasoning

Graph convolution is a reasoning operation similar to convolution that performs on graph structure data [32]. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its adjacency matrix $\boldsymbol{A}$, its degree matrix $\boldsymbol{D}$ is the diagonal of $\boldsymbol{A}$. Correspondingly, the normalized Laplacian matrix $\boldsymbol{L}$ of graph $\mathcal{G}$ can be defined as

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \tag{1}$$

where $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{L}$ is a positive semidefinite matrix, including a complete set of eigenvectors $\boldsymbol{V}$, determined by $\boldsymbol{L} = \boldsymbol{V}\Lambda\boldsymbol{V}^T$. Using $\boldsymbol{V}$ to perform the Fourier transform of the graph (i.e., $\hat{\boldsymbol{s}} = \boldsymbol{V}^T\boldsymbol{s}$), the graph signal $\boldsymbol{s} \in \mathbb{R}^n$ can be converted to the spectral domain.

Extending the convolution to the structured space of the graph, it can be understood as decomposing $\boldsymbol{s}$ in the spectral domain and then multiplying the corresponding frequencies by the spectral filter $g_{\boldsymbol{\theta}}$ (parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$), namely

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} = \boldsymbol{V} g_{\boldsymbol{\theta}} \boldsymbol{V}^T \boldsymbol{s}. \tag{2}$$

Equation (2) needs to explicitly calculate the Laplacian eigenvectors, causing a lot of computational burden. To improve this problem, Defferrard *et al.* [33] try to use Chebyshev polynomials to approximate $g_{\boldsymbol{\theta}}$ to the Kth order. Thus, the convolution of the graph signal $\boldsymbol{s}$ can be expressed as

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} \approx \sum_{k=0}^{K} \boldsymbol{\theta}_k \boldsymbol{T}_k \boldsymbol{s} \tag{3}$$

where $\{\boldsymbol{\theta}_k\}$ is the Chebyshev coefficient vector and $\{\boldsymbol{T}_k\}$ is the Chebyshev polynomial. Later, Kipf and Welling [32] further simplify (3) by restricting $K = 1$, and approximate the maximum eigenvalue of $\boldsymbol{L}$ to 2. Based on this, the graph convolution is transformed into

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} = \boldsymbol{\theta}\left(\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\right)\boldsymbol{s}. \tag{4}$$

Equation (4) is further normalized by the following equation:

$$\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \rightarrow \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tag{5}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\tilde{\boldsymbol{D}}_{ii} = \sum_j \tilde{\boldsymbol{D}}_{jj}$. Thus, the layerwise propagation rules of graph convolutional network (GCN) are as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{\theta}^{(l)}\right) \tag{6}$$

where $\boldsymbol{H}^{(l)}$ and $\boldsymbol{H}^{(l+1)}$ are the vertex features of the lth layer and $l+1$th layer, respectively, $\boldsymbol{\theta}^{(l)}$ is the weight matrix of layer l, and σ is nonlinearity activation function.

Equation (6) clarifies the details of the convolution of the graph structure data. Some works have applied it to multi-object tracking to improve feature representation, but they mainly focus on the postprocessing, i.e., formulating the data association as a graph matching problem [57], [63]. Differently, our STeRe module applies (6) to reason spatiotemporal long-range dependencies, and then assists detection and ReID instead of postprocessing. In addition, we design a data-driven approximate adjacency matrix A to facilitate learning contextual patterns.

B. Multitask Learning

MTL means that the model can infer the output of multiple tasks under a given input. Specifically, in the era of deep learning, MTL attempts to design a neural network that can learn shared representations from supervision signals of multiple tasks. Compared with the single-task situation, the MTL network benefits from inherent layer sharing, which can reduce memory usage and increase inference speed. In addition, the shared information of related tasks can also be adjusted to complement each other to improve performance. The current mainstream MTL network design technology can be divided into: soft parameter sharing [36]–[38] and hard parameter sharing [30], [39].

However, when the training optimization objectives of each task are inconsistent (e.g., detection and ReID in multi-object tracking), the joint learning of multiple different tasks is likely to negative transfer [40]. To address this problem, one idea is to explicitly construct complementary task-specific representations [41], [42], but the complexity raises with the number of tasks. Another idea is to eliminate factors that may hurt the performance of other tasks from the current task training [43], [44]. Bousmalis *et al.* [45] use similarity or orthogonality constraints to decouple network sharing or task-specific features. Ganin and Lempitsky [46] and Liu *et al.* [47] propose using adversarial training for domain adaptation in the feature space so that the discriminator cannot distinguish the source domain.

We understand that this training setting can promote network partition learning. Compared with previous work, the designed MAGra strategy hopes that the loss gradient of each task has a similar static distribution. It introduces a gradient discriminator into the training to add extra capacity to the network, instead of directly assigning weights. The discriminator is only active in backpropagation and will not affect the inference speed.

III. Methodology

Our goal is to integrate motion clues into detection to improve the tracking in tough scenes, and to alleviate the inconsistency of optimization objectives between detection and ReID based on MTL. To this end, we propose a graph-based STeRe module (Section III-B) to construct spatiotemporal dependencies and derive object motion, and an MAGra

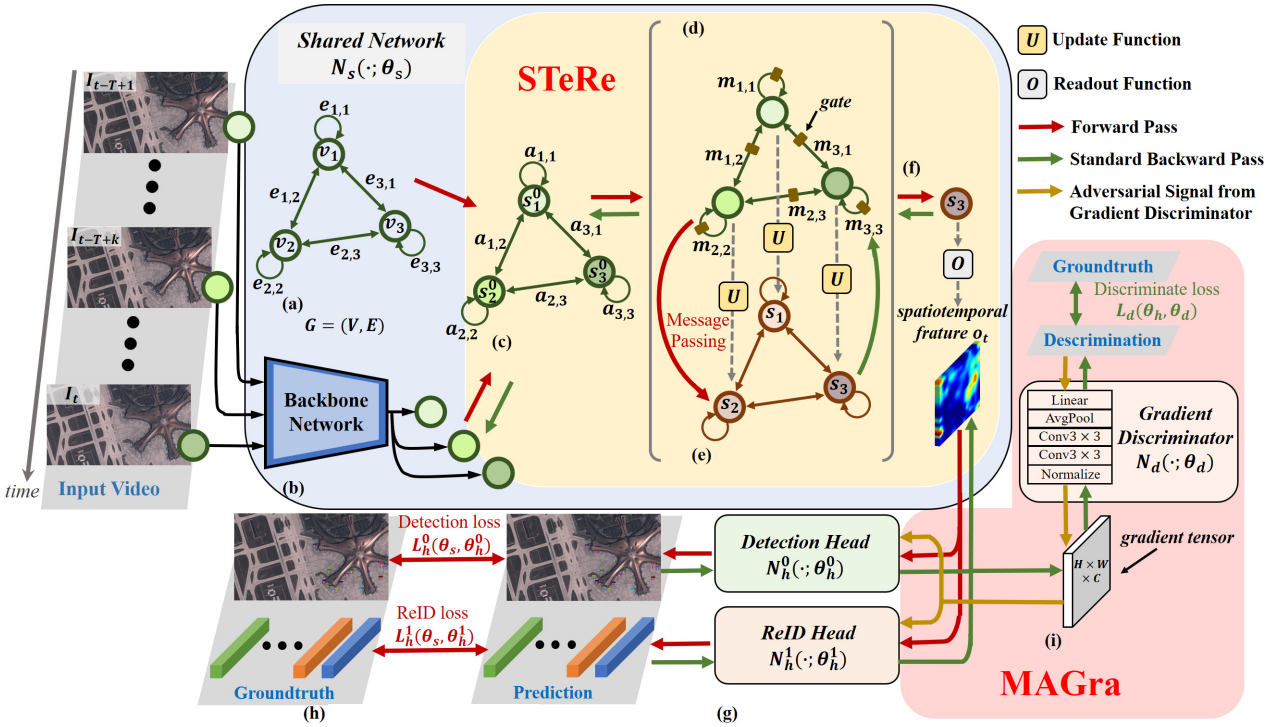

Fig. 3. Overview of the proposed TGraM framework. TGraM employs the STeRe module to build multitemporal (i.e., $T > 1$) content into an undirected graph, and then extracts the joint semantics in the embedding space to generate spatiotemporal feature $\boldsymbol{o}_t$. In the training phase, the MAGra strategy introduces the adversarial signal produced by the gradient discriminator to regularize the head network of each task, so that the gradient distribution tends to be similar. (a) Video graph. (b) Feature extraction. (c) Initial node and edge states. (d) Message reasoning and gated aggregation. (e) Node states update. (f) Readout. (g) Task-specific output. (h) Compute loss for each task. (i) Gradient adaptation.

learning strategy (Section III-C) for gradient adaptation and task-specific network regularization.

A. Overview

The proposed TGraM employs a point-based approach to object tracking, regarding each point on the feature map as the object center or background. Given a video $\mathcal{I} = \{\boldsymbol{I}_t \in \mathbb{R}^{H_I \times W_I \times 3}\}_{t=1}^{N}$ with N frames, the tracker first uses the backbone network to generate corresponding feature maps $\mathcal{F} = \{\boldsymbol{F}_t \in \mathbb{R}^{H \times W \times C}\}_{t=1}^{N}$, where $H = (H_I/8)$, $W = (W_I/8)$, $C = 96$. As shown in Fig. 3, the STeRe module models $\{\boldsymbol{F}_{t-T+1}, \boldsymbol{F}_{t-T+2}, \ldots, \boldsymbol{F}_t\}$ as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which t and T, respectively, indicate the current time and the total selected times (i.e., the number of nodes). The node $\boldsymbol{v}_i \in \mathcal{V}$ indicates the feature map of the ith frame $\boldsymbol{F}_i$, and the edge $\boldsymbol{e}_{i,j} = (\boldsymbol{v}_i, \boldsymbol{v}_j) \in \mathcal{E}$ indicates the relation between $\boldsymbol{F}_i$ and $\boldsymbol{F}_j$. The adjacency matrix element $\boldsymbol{a}_{i,j} \in \boldsymbol{A}$ indicates the concrete weight of the edge $\boldsymbol{e}_{i,j}$. For each node $\boldsymbol{v}_i$, we apply graph convolution to reason the message $\boldsymbol{m}_i$ (i.e., the information summary from its neighbor $\boldsymbol{N}_i$) and update its state $\boldsymbol{s}_i$

$$\boldsymbol{m}_i = \sum_{\boldsymbol{v}_j \in \boldsymbol{N}_i} \boldsymbol{m}_{i,j} = \sum_{\boldsymbol{v}_j \in \boldsymbol{N}_i} M\left(\boldsymbol{s}_j, \boldsymbol{a}_{i,j}\right) \in \mathbb{R}^{H \times W \times C} \tag{7}$$

$$\boldsymbol{s}_i = U\left(\boldsymbol{s}_i^0, \boldsymbol{m}_i\right) \in \mathbb{R}^{H \times W \times C} \tag{8}$$

where the initial state $\boldsymbol{s}_i^0 = \boldsymbol{v}_i$. $M(\cdot)$ and $U(\cdot)$ are, respectively, message function and state update function. After updating, the spatiotemporal long-range dependency is constructed. Then the readout function $O(\cdot)$ is used to map the node representation at the current time to the node output

$$\boldsymbol{o}_t = O(\boldsymbol{s}_t) \in \mathbb{R}^{H \times W \times C}. \tag{9}$$

$\boldsymbol{o}_t$ is the output of the tracker's shared network $N_s(\cdot;\boldsymbol{\theta}_s)$ (parameterized by $\boldsymbol{\theta}_s$). On this basis, the detection and ReID head networks $\{N_h^i(\cdot;\boldsymbol{\theta}_h^i)\}_{i=0}^{1}$ (parameterized by $\boldsymbol{\theta}_h^i$) are built separately, and the corresponding task-specific output is generated.

Then, training the tracker derives the following optimization problem:

$$\min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_h^0, \boldsymbol{\theta}_h^1} \frac{1}{2} \sum_{i=0}^{1} L_h^i\left(\boldsymbol{\theta}_s, \boldsymbol{\theta}_h^i\right) + \lambda \cdot R\left(\boldsymbol{\theta}_h^0, \boldsymbol{\theta}_h^1\right) \tag{10}$$

where $\{L_h^i\}_{i=0}^{1}$ indicate the loss function of each task (similar to the setting in [48]), and R is an optional regularizer. Due to the inconsistency of optimization objectives between detection and ReID, the MAGra strategy is designed to promote the loss gradients of these two tasks $\boldsymbol{G}_d \in \mathbb{R}^n$ and $\boldsymbol{G}_r \in \mathbb{R}^n$ ($n = H \times W \times C$) to have similar statistical distribution. Based on [49], the empirical $\mathcal{H}$-divergence between $\boldsymbol{G}_d$ and $\boldsymbol{G}_r$ is used to estimate the distribution similarity

$$\hat{d}_{\mathcal{H}}(\boldsymbol{G}_d, \boldsymbol{G}_r) = 2\left(1 - \min_{\eta} \frac{1}{n}\left[\sum_{\boldsymbol{x} \in \boldsymbol{G}_d} I[\eta(\boldsymbol{x}) = 0] + \sum_{\boldsymbol{x} \in \boldsymbol{G}_r} I[\eta(\boldsymbol{x}) = 1]\right]\right) \tag{11}$$

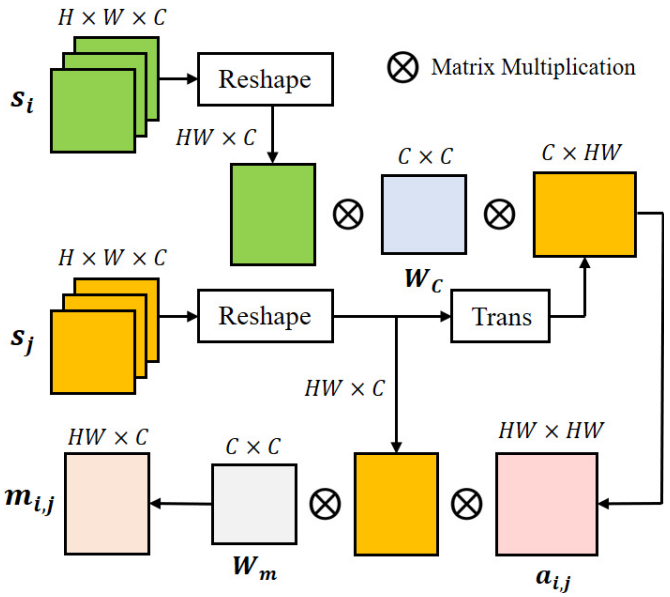

Fig. 4. Computation procedures of the similarity matrix $\boldsymbol{a}_{i,j}$ and message $\boldsymbol{m}_{i,j}$. $\boldsymbol{a}_{i,j}$ is calculated as an attention mechanism of node state $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, while $\boldsymbol{m}_{i,j}$ is reasoned through graph convolution.

where $\eta(\cdot)$ is the binary classifier, and $I[\cdot]$ is the indicator function. Specifically, MAGra estimates the "min" part of (11) through a gradient discriminator network $N_d(\cdot;\boldsymbol{\theta}_d)$ (parameterized by $\boldsymbol{\theta}_d$). The learning of $\boldsymbol{\theta}_d$ allows us to add a distribution adaptation term (i.e., regularizer R) to (10), and compete with $\{\boldsymbol{\theta}_h^i\}_{i=0}^1$ over the objective in an adversarial manner.

B. Graph-Based Spatiotemporal Reasoning

The key idea of the STeRe module is to perform message reasoning and passing on $\mathcal{G}$ to mine the rich high-order spatiotemporal relations within $\mathcal{I}$. This assists to capture video content from a global perspective, thereby improving dense and motion blurred objects in some frames. To fully model the basic relations between video frames, $\mathcal{G}$ is assumed to be fully connected, and each node includes self-connections. SteRe essentially extends the traditional connected GNNs, not only preserving spatial information but also extracting paired relations (edges) through a differentiable attention mechanism.

The edge $\boldsymbol{e}_{i,j} \in \mathcal{E}$ connects the nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, and the corresponding $\boldsymbol{a}_{i,j}$ is used to mine the semantic relation in the embedding space. For the case of $i = j$, $\boldsymbol{e}_{i,i}$ connects the node to itself. $\boldsymbol{a}_{i,i}$ is used to capture the internal correlation of the node state $\boldsymbol{s}_i$ (i.e., the internal representation of the frame). As shown in Fig. 4, $\boldsymbol{a}_{i,j}$ is calculated as an attention mechanism, convenient for modeling multilevel and long-range dependencies across regions

$$\boldsymbol{a}_{i,j} = F_{\text{att}}(\boldsymbol{s}_i, \boldsymbol{s}_j) = \boldsymbol{s}_i \boldsymbol{W}_c \boldsymbol{s}_j^T \in \mathbb{R}^{(HW)\times(HW)} \tag{12}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{C\times C}$ indicates the learnable weight, $\boldsymbol{s}_i \in \mathbb{R}^{(\text{HW})\times C}$ and $\boldsymbol{s}_j \in \mathbb{R}^{(\text{HW})\times C}$ are flattened into matrix forms. Since $\mathcal{G}$ is an undirected graph, A is a symmetric matrix (i.e., $\boldsymbol{a}_{i,j} = \boldsymbol{a}_{j,i}^T$). Each element of $\boldsymbol{a}_{i,j}$ reflects the similarity between each row in $\boldsymbol{s}_i$ and each column of $\boldsymbol{s}_j^T$. $\boldsymbol{a}_{i,j}$ explores the joint semantic representation by paying attention to each node pair $(\boldsymbol{v}_i, \boldsymbol{v}_j)$.

For the message $\boldsymbol{m}_{i,j}$ received from the node $\boldsymbol{v}_j$, we apply graph convolution [i.e., (6)] to reason

$$\boldsymbol{m}_{i,j} = M(\boldsymbol{s}_j, \boldsymbol{a}_{i,j}) = \sigma(\boldsymbol{a}_{i,j}\boldsymbol{s}_j \boldsymbol{W}_m) \in \mathbb{R}^{(HW)\times C} \tag{13}$$

where σ is the logistic sigmoid function, and $\boldsymbol{W}_m \in \mathbb{R}^{C\times C}$ indicates the weight matrix. In previous studies [29], [34], the adjacency matrix is mostly data-independent parameters. To better preserve the internal spatial structure, our $\boldsymbol{a}_{i,j}$ is carefully designed to ensure that the learning long-range context pattern depends on the input features. The message function $M(\cdot)$ allocates the edge-weighted feature (i.e., message) of a node to its neighbors. The long-range dependencies between objects at different time (frames) are, thus, constructed. Then, $\boldsymbol{m}_{i,j}$ is reshaped into a 3-D tensor of $H \times W \times C$.

Besides, some nodes produce noise because of motion blur or dense distribution, so their messages may be useless or even deleterious. We present the learnable gate $G(\cdot)$ to evaluate the confidence of message $\boldsymbol{m}_{i,j}$

$$\boldsymbol{p}_{i,j} = G(\boldsymbol{m}_{i,j}) = \sigma\left(F_{\text{GAP}}(\boldsymbol{W}_g * \boldsymbol{m}_{i,j} + \boldsymbol{b}_g)\right) \in [0,1]^C \tag{14}$$

where "*" indicates convolution operation, $F_{\text{GAP}}(\cdot)$ indicates channel response generated by global average pool, and $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are learnable convolution filters and bias, respectively.

Following (7), node $\boldsymbol{v}_i$ uses gated summarization to collect messages from neighbors and itself

$$\boldsymbol{m}_i = \sum_{\boldsymbol{v}_j \in N_i} \boldsymbol{p}_{i,j} \odot \boldsymbol{m}_{i,j} \in \mathbb{R}^{H\times W\times C} \tag{15}$$

where "$\odot$" indicates the channelwise Hadamard product. Here, the gate mechanism is utilized to filter irrelevant information from node noise. After aggregating all the information from itself and neighbor nodes, $\boldsymbol{v}_i$ obtains a new state $\boldsymbol{s}_i$ based on the prior state $\boldsymbol{s}_i^0$ and the received message $\boldsymbol{m}_i$

$$\boldsymbol{s}_i = U(\boldsymbol{s}_i^0, \boldsymbol{m}_i) = \tanh\left(\boldsymbol{W}_u * F_{\text{CAT}}(\boldsymbol{s}_i^0, \boldsymbol{m}_i)\right) \in \mathbb{R}^{H\times W\times C} \tag{16}$$

where tanh is the hyperbolic tangent function, $F_{\text{CAT}}(\cdot)$ indicates channel concatenation, and $\boldsymbol{W}_u$ is the convolution filter.

After the message reasoning and passing, we obtain the node output at the current time t from $\boldsymbol{s}_t$ by readout function $O(\cdot)$. Slightly different from (9), we also feed the initial state $\boldsymbol{s}_t^0$ to $O(\cdot)$

$$\boldsymbol{o}_t = O_{\text{FCN}}([\boldsymbol{s}_t, \boldsymbol{s}_t^0]) \in \mathbb{R}^{H\times W\times C}. \tag{17}$$

To preserve the spatial information, the readout function is realized as a tiny FCN network, including two convolutional layers and a sigmoid function.

STeRe formulates the feature map of each frame as the node, where each pixel is regarded as a potential object. Compared with employing tracklets as nodes [23], our method is more robust, because the detection of previous frames does not affect the current frame. STeRe constructs the long-range dependencies of the objects at different times (frames) through message reasoning, and then propagates past features to update the current node. What is more, the above-mentioned functions are meticulously devised to avoid spatial information interference,

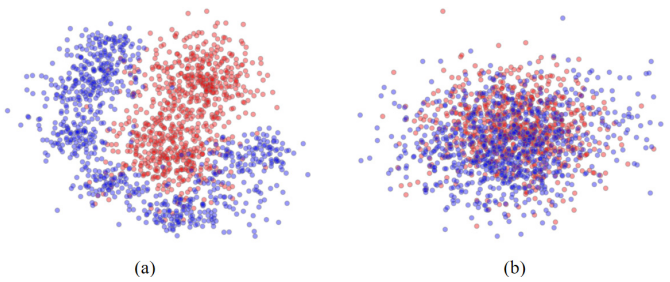

Fig. 5. Influence of MAGra on the distribution of the gradients on the AIR-MOT training set (best viewed in color). This illustrates t-SNE [52] visualization of the gradient tensor (a) in circumstance as no adaptation is performed and (b) in circumstance as our MAGra is incorporated into training. Red points represent the examples from the detection head, while blue ones correspond to the ReID head. The adaptation in our MAGra promotes the two distributions of gradients much closer.

and weights are shared among all nodes. This is essential for TGraM because it needs to complete real-time tracking of multiple objects.

C. Multitask Adversarial Gradient Learning

To ensure that no discriminative information about the gradient source (i.e., detection or ReID) is included in the shared network training, while keeping the risk low, the MAGra strategy is proposed. The gradient distributions from different tasks are expected to be sufficiently similar, i.e., the empirical $\mathcal{H}$-divergence [i.e., (11)] is sufficiently small, as shown in Fig. 5. Ben-David *et al.* [50] point out that when $\hat{d}_{\mathcal{H}}$ is difficult to accurately calculate, an algorithm that distinguishes the source of the distribution can be used for estimation. Inspired by this, MAGra employs the risk of training gradient discriminator $N_d(\cdot;\boldsymbol{\theta}_d) :\rightarrow \{0, 1\}$ to approximate the "min" part of (11). $N_d(\cdot;\boldsymbol{\theta}_d)$ receives the gradient tensor $\boldsymbol{g} \in \mathbb{R}^{H\times W\times C}$ from each task as input, and classifies the source. We define the loss as

$$L_d(N_d(\boldsymbol{g};\boldsymbol{\theta}_d), d) = d\log\frac{1}{N_d(\boldsymbol{g};\boldsymbol{\theta}_d)} + (1-d)\log\frac{1}{1-N_d(\boldsymbol{g};\boldsymbol{\theta}_d)} \tag{18}$$

where d is the binary label for distribution of $\boldsymbol{g}$, indicating that $\boldsymbol{g}$ is the loss gradient from detection ($\boldsymbol{g}\sim\boldsymbol{G}_d$ if $d=0$) or ReID ($\boldsymbol{g}\sim\boldsymbol{G}_r$ if $d=1$). This allows us to add the following regularizer to (10):

$$R\left(\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_h^1\right) = \max_{\theta_d}\left\{-\frac{1}{2}\left[\sum_{\boldsymbol{g}\in\boldsymbol{G}_d} L_d\left(\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_d\right) + \sum_{\boldsymbol{g}\in\boldsymbol{G}_r} L_d\left(\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_d\right)\right]\right\} \tag{19}$$

where $L_d(\boldsymbol{\theta}_h^i,\boldsymbol{\theta}_d) = L_d(N_d(\boldsymbol{g}(\boldsymbol{\theta}_h^i);\boldsymbol{\theta}_d), d)$, and $\boldsymbol{g}(\boldsymbol{\theta}_h^i)$ indicates the gradient of task i calculated by θ_h^i. The regularizer seeks to approximate the $\mathcal{H}$-divergence within (11), i.e., $2-R(\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_h^1)$ is a substitute for $\hat{d}_{\mathcal{H}}(\boldsymbol{G}_d,\boldsymbol{G}_r)$. The optimization problem given by (10) and (19) achieves a compromise between the minimization of distribution divergence and risk. The hyperparameter λ is used to adjust this tradeoff during the learning phase.

Algorithm 1 MAGra

Require: Inputs $\boldsymbol{x}$, labels for each task $\{\boldsymbol{y}_i\}_{i=0}^1$

1: Initialize tracker parameters $\{\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_h^1\}$, discriminator parameters $\boldsymbol{\theta}_d$, along with α and λ.
2: **for** $k=0$ to maxEpoch **do**
3: Sample a mini-batch.
4: Obtain the output $\{\boldsymbol{T}_i\}_{i=0}^1$ of each task via $N_s(\cdot;\boldsymbol{\theta}_s)$ and $\{N_h^i(\cdot;\boldsymbol{\theta}_h^i)\}_{i=0}^1$.
5: Calculate the gradient tensor $\boldsymbol{g}$ returned by each task based on $\{\boldsymbol{T}_i\}_{i=0}^1$.
6: Calculate the optimization objective on Eq. 20.
7: Update $\{\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_d\}$ via Eq. 23-25.
8: **end for**

Ensure: $N_s(\cdot;\boldsymbol{\theta}_s)$ and $\{N_h^i(\cdot;\boldsymbol{\theta}_h^i)\}_{i=0}^1$

For learning, we can rewrite the complete optimization objective of (10) as

$$E\left(\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_d\right) = \frac{1}{2}\sum_{i=0}^{1} L_h^i\left(\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^i\right) - \frac{\lambda}{2}\left[\sum_{\boldsymbol{g}\in\boldsymbol{G}_d} L_d\left(\boldsymbol{\theta}_h^0,\boldsymbol{\theta}_d\right) + \sum_{\boldsymbol{g}\in\boldsymbol{G}_r} L_d\left(\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_d\right)\right] \tag{20}$$

where the saddle point parameters $\hat{\boldsymbol{\theta}}_s$, $\hat{\boldsymbol{\theta}}_h^0$, $\hat{\boldsymbol{\theta}}_h^1$, and $\hat{\boldsymbol{\theta}}_d$ are determined by the following equations:

$$\left(\hat{\boldsymbol{\theta}}_s,\hat{\boldsymbol{\theta}}_h^0,\hat{\boldsymbol{\theta}}_h^1\right) = \arg\min_{\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_h^0} E\left(\boldsymbol{\theta}_s,\boldsymbol{\theta}_h^1,\boldsymbol{\theta}_h^1,\hat{\boldsymbol{\theta}}_d\right) \tag{21}$$

$$\hat{\boldsymbol{\theta}}_d = \arg\max_{\boldsymbol{\theta}_d} E\left(\hat{\boldsymbol{\theta}}_s,\hat{\boldsymbol{\theta}}_h^0,\hat{\boldsymbol{\theta}}_h^1,\boldsymbol{\theta}_d\right). \tag{22}$$

Therefore, the optimization problem involves the maximization of certain parameters and the minimization of other parameters.

To address this problem, MAGra updates the maximization parameters in the positive direction of the gradient through backpropagation, and vice versa. Thus, the saddle point defined by (21) and (22) can be transformed into the stationary point of the gradient update

$$\boldsymbol{\theta}_s \leftarrow \boldsymbol{\theta}_s - \frac{\alpha}{2}\sum_{i=0}^{1}\frac{\partial L_h^i}{\partial\boldsymbol{\theta}_s} \tag{23}$$

$$\left\{\boldsymbol{\theta}_h^i\right\}_{i=0}^1 \leftarrow \left\{\boldsymbol{\theta}_h^i - \alpha\left(\frac{\partial L_h^i}{\partial\boldsymbol{\theta}_h^i} - \lambda\frac{\partial L_d}{\partial\boldsymbol{\theta}_h^i}\right)\right\}_{i=0}^1 \tag{24}$$

$$\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d - \alpha\lambda\frac{\partial L_d}{\partial\boldsymbol{\theta}_d} \tag{25}$$

where α is the learning rate. Algorithm 1 provides the pseudocode of this learning process. In the training procedure, the tracker (parameterized by $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_h^0$, and $\boldsymbol{\theta}_h^1$) and the gradient discriminator (parameterized by $\boldsymbol{\theta}_d$) compete and optimize each other on the objective of (20) in an adversarial manner. MAGra effectively trains the head network of each task $\{N_h^i(\cdot;\boldsymbol{\theta}_h^i)\}_{i=0}^1$, which generates the corresponding gradient

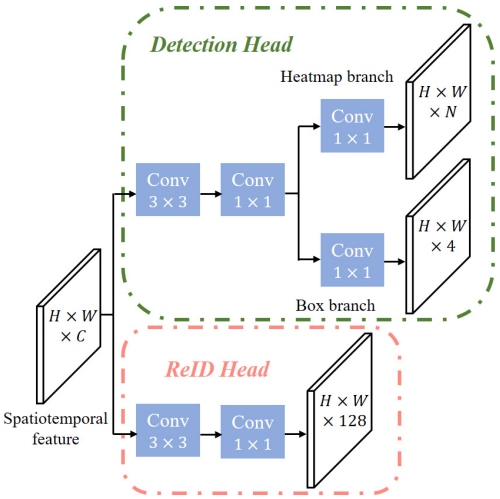

Fig. 6. Architecture of the detection head and ReID head. The detection head is composed of heat map branch and box size branch. The heat map branch generates a $H \times W \times N$ classwise confidence map ($H \times W \times 2$ by default), where the peaks are regarded as the object centers. The box branch outputs the center coordinates, height, and width of the corresponding bounding box. ReID head produces 128-D embedding for each object.

distribution, allowing equal and accurate representation learning for the shared network $N_s(\cdot;\boldsymbol{\theta}_s)$. While it weakens the ability of the gradient discriminator $N_d(\cdot;\boldsymbol{\theta}_d)$ to judge whether the gradient comes from the detection or ReID task.

D. Tracklet Generation

The holistic architecture of TGraM is demonstrated in Fig. 3. Based on the spatiotemporal feature $\boldsymbol{o}_t$, TGraM generates the bounding box of the object through the detection head. Then, the embedding output from the ReID head is used for data association, connecting the detection to the previous tracklet. As shown in Fig. 6, each head network is made up of lightweight convolution to ensure real-time requirements.

Concretely, the data association includes two steps, which will be explained by taking the detection dt with the center position at (i, j) as an example. Step **1)**: We first associate dt with the latest unmatched detection in the area centered at (i, j) at time $t-1$. The area radius is the quadratic mean of the height and width of the detection box. Step **2)**: If dt does not match any object in the first step, the embedding output of the ReID head $eb^t_{i,j}$ will be used to calculate the Euclidean distance with all historical or unmatched tracklets. dt will be assigned to the tracklet with the highest similarity and greater than a certain threshold. dt will start a new tracklet if it is still not matched. Step **3)** realizes long-term tracking via $eb^t_{i,j}$.

IV. Experimental Results

In this section, we will conduct a comprehensive evaluation of the presented TGraM on the AIR-MOT dataset. Specifically, the dataset and implementation details are briefly introduced first. Then, our ablation study on the key components of TGraM is carried out. At last, the holistic performance of the method is analyzed qualitatively and quantitatively.

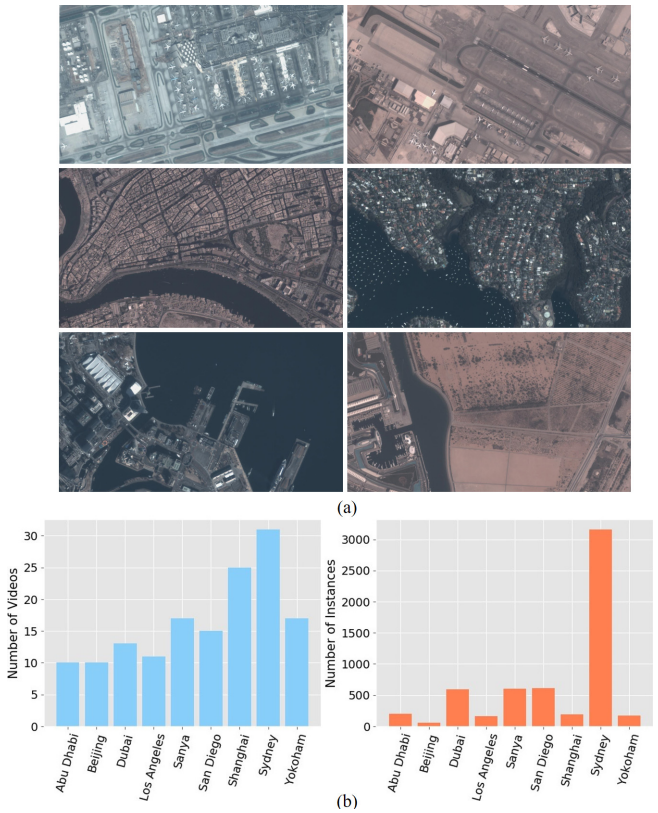

Fig. 7. Details of AIR-MOT dataset. (a) Some data with complex and diverse backgrounds. (b) Statistics of the dataset: the left histogram shows the number of videos per area; the right one shows the number of instances.

A. Experimental Data

To promote the research of multi-object tracking in satellite video, we have built a brand-new dataset, namely, AIR-MOT. Specifically, the samples in AIR-MOT are motion examples from different regions observed by satellites, so similar examples often have diverse and complex backgrounds. In addition, the samples in AIR-MOT cover multiple complete trajectories without sensor changes. These two characteristics ensure the applicability of the study of multi-object tracking because the potential interference is clearly avoided. AIR-MOT will be opened to the community in the near future, one of the earliest public datasets in this field. Next, we will first introduce the collection process, and then analyze the statistics of the dataset.

1) Dataset Collection: To obtain samples in AIR-MOT, we first collect multiple videos taken by the Jilin-1 satellite. Each collected video is divided into a group of shots by shot detection [55]. For the samples in the shot, we employ crowdsourcing services to conduct two rounds of annotation and inspection. In the first round, shots that are irrelevant and contain incomplete objects are filtered out. In the remaining shots, the object is annotated with location (i.e., detection label) and instance (i.e., ReID label) information. Before the second round, the experts are provided with explanatory descriptions and examples, to guide them in inspecting and correcting the labels generated in the previous round. The annotations of the final version are presented in the form of a

TABLE I
BASIC ATTRIBUTES OF AIR-MOT

Attribute	AIR-MOT
Object	Aircraft, Ship
Source	Jilin-1 Satellite
Sensor	MSS
Spatial Resolution	0.91 m – 1.27 m
Shooting Time	October 2017 October 2020
Number of Videos	149
Number of Instances	5736
Timestamp	70 – 326
Frame Rate	5 – 10
Size	1920×1080
Aera	Abu Dhabi, Beijing, Dubai, Los Angeles, Sanya, San Diego, Shanghai, Sydney, Yokoham

text file, where each line represents an object instance, similar to [14]. Each line contains nine values, such as

$$1, 1, 587, 141, 27, 34, 1, 1, 1$$
$$1, 2, 103, 801, 16, 14, 1, 2, 1$$
$$2, 4, 869, 684, 59, 76, 1, 2, 1.$$

The first value indicates which frame the object appears in, while the second value indicates the tracklet ID to which the object belongs. Each object can only be assigned to one unique tracklet. The next four values indicate the position of the object's bounding box in 2-D frame coordinates. The position is indicated by the upper left corner and the width and height of the bounding box. The next value indicates whether the object is considered (1) or ignored (0). The eighth value indicates the category of the object, i.e., aircraft (1) or ship (2). The last value indicates the visibility of the object, between 0 and 1. Some objects may not be visible because of the frame border cropping and occlusion.

2) Dataset Statistics: AIR-MOT contains a total of 5736 instances, across 149 videos, ten full scenes collected from different regions of the world by Jilin-1 satellite from October 2017 to October 2020. As shown in Table I, the videos have more than 70 timestamps, with a frame rate of 5–10 FPS and a size of 1920 × 1080 pixels. In addition, since the videos are shot on a global scale, the background is complex and diverse even for similar objects, and the objects have multiscale characteristics. As shown in Fig. 7, while the number of instances in each video reflects the natural distribution of objects in the area, the changes in instances (including duration and position shifts) reflect the variety of object motions, helpful to verify the robustness of the tracker.

In the experiment, 70% of the original videos are randomly selected as the training set, and the remaining 30% are as the test set.

B. Implementation Details

1) Evaluation Metrics: To evaluate the performance of different methods in multi-object tracking in satellite video, we choose 12 common quantitative metrics, including multiple object tracking accuracy (MOTA), ID F1 score (IDF1), ID precision (IDP), ID recall (IDR), the number of false positives (FP), false negatives (FN), identity switches (IDs), the percentage of mostly tracked trajectories (MT), mostly lost trajectories (ML), overall parameters (Params), calculations (number of multi-adds), and frames per second (FPS) [14], [62]. Note that MOTA, IDF1, IDP, IDR, FP, FN, IDs, MT, and ML are used to evaluate accuracy; Params, Multi-adds, and FPS are used for efficiency.

TABLE II
EFFECTIVENESS OF EACH PROPOSED MODULE ON THE AIR-MOT TEST SET. THE PROPOSED STERE, MAGRA, AND OVERALL TGRAM ARE EVALUATED. "BASELINE + STERE + MAGRA" IS REPRESENTED BY "TGRAM." "↓" INDICATES LOWER IS BETTER. "↑" INDICATES HIGHER IS BETTER

Scheme	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓
CenterTrack [19]	63.6	65.1	215	4820	6028
Baseline	62.1	63.7	263	4237	6459
Baseline+STeRe	64.8	65.9	247	1836	4407
Baseline+MAGra	63.9	64.5	232	2806	5192
TGraM	**65.7**	**66.8**	**182**	**1482**	**3764**

2) Comparison Methods: The proposed method is compared with representative online trackers, including DeepSORT [12], RAN [56], HOGM [57], DAN [58], Tracktor + CTdet [20], CKDNet + SMTNet [11], TubeTK [59], CTracker [21], JDE [22], UMA [61], CenterTrack [19], GSDT [23], FairMOT [48], and TraDeS [24]. DeepSORT [12], RAN [56], HOGM [57], DAN [58], Tracktor + CTdet [20], and CKDNet + SMTNet [11] belong to the DBT paradigm, and the rest belong to JDT. We use faster-RCNN [54] to provide detections of each frame for DeepSORT [12], RAN [56], and HOGM [57]. To be fair, other details of each tracker are set according to the corresponding original article.

3) Experimental Setting: In the experiment, the compact model MobilenetV3-Small [53] is employed as the backbone to fully verify the effectiveness of the proposed framework. TGraM without STeRe and MAGra is set as the baseline tracker. Each frame of the original video is scaled to 1088 × 608 pixels, of which 32 are randomly selected as input batch. We use the Adam optimizer [60] to train our model for 70 epochs. Specifically, relevant hyperparameters are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The initial learning rate is set to $1.25e^{-4}$, and then it drops to e^{-5} at 60 epochs. Rotation, flipping, and color jittering are applied to augment the image. We set the number of nodes in the graph $\mathcal{G}$ as $T = 3$. Inference speed (i.e., FPS) is tested on an NVIDIA Titan RTX GPU. The experiment is executed five times and the average is used as the final result.

C. Ablation Studies

1) Effectiveness of TGraM: As shown in Table II, we compare the proposed STeRe, MAGra, and TGraM with our baseline tracker and CenterTrack [30]. Compared with the baseline, STeRe achieves better detection by reducing the FN

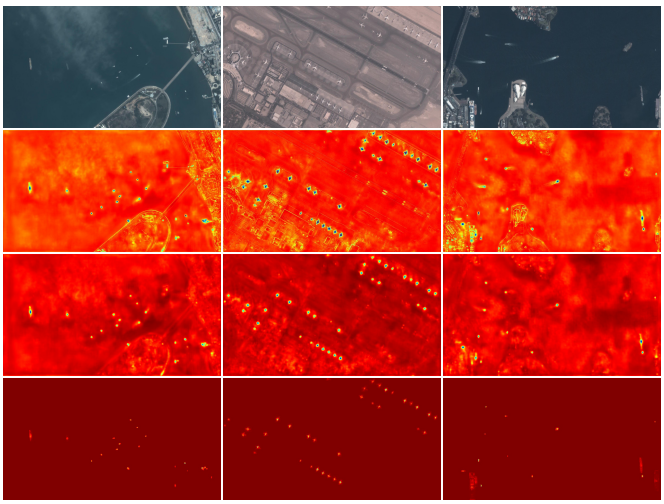

Fig. 8. Visualization of the attention maps for the STeRe module. The images in the first row illustrate the input images; the ones in the second row illustrate the attention maps generated by the input of STeRe; the ones in the third row illustrate the attention maps generated by the message reasoning via STeRe, i.e., (15); and the images in the last row illustrate the corresponding attention maps generated by the spatiotemporal feature, i.e., (17). Best viewed in color.

by 31.8% (i.e., recovering more missed objects), which verifies the effectiveness of our graph structure, tracking clues, and spatiotemporal dependencies. For MAGra, we directly add it to the baseline tracker. Since the spatiotemporal feature $\boldsymbol{o}_t$ is not available in the baseline, we only use the current node (frame) as the input of the head network. Compared with the baseline, MAGra implements better tracking by improving 0.8 IDF1 and reducing IDs by 11.9%. In addition, we observe that MAGra also reduces FN by 19.6%, confirming that stable tracking will assist detection. With the help of STeRe, TGraM reduces IDs from 263 to 182. What is more, in TGraM, the gradient adaptation from MAGra guides the passing of spatiotemporal messages in STeRe, which greatly reduces FN from 6459 to 3764. The recovery of missed objects and better IDs together improve the overall tracking performance, achieving 65.7 MOTA and 66.8 IDF1. The performance of TGraM is also better than the recent JDT method CenterTrack [19].

2) Effectiveness of STeRe: We visualize the information propagated in the STeRe module and the predicted spatiotemporal feature $\boldsymbol{o}_t$ in Fig. 8. After aggregating the reasoned message [(13)–(16)], the central feature of the object is significantly enhanced. In the case of a low frame rate, STeRe can accurately predict the tracking deviation of the object. In addition, $\boldsymbol{o}_t$ can even accurately track partially missed objects in intermediate frames. Even if the objects are blurred or dense in the intermediate frame, STeRe can successfully sample the central feature through the previous frames. These examples show that STeRe can predict the spatial information of objects in a wide range and provide robust motion clues. In addition, we also evaluate the influence of the number of nodes T on inference. The performance of different T values is shown in Table III. We find that with more input frames ($1 \rightarrow 3$), the performance improves accordingly. While considering even more frames ($3 \rightarrow 5$), the final performance does not change significantly, possibly due to redundant information in the video sequence.

TABLE III

PERFORMANCE IN COMPARISON ON THE AIR-MOT TEST SET WITH DIFFERENT NUMBER OF NODES. "T" REPRESENTS THE NUMBER OF NODES (I.E., FRAMES) EMPLOYED IN THE STERE MODULE

Scheme	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓	FPS↑
$T = 1$	63.1	63.8	317	2803	5027	**38.4**
$T = 2$	64.9	65.2	243	2175	4529	34.7
$T = 3$	**65.7**	**66.8**	182	**1482**	3764	31.2
$T = 4$	65.7	66.1	**177**	1536	4028	27.9
$T = 5$	65.4	66.3	189	1512	**3615**	24.3

TABLE IV

ABLATION FOR OUR MAGRA STRATEGY ON THE AIR-MOT TEST SET. "TGRAM WITHOUT MAGRA" REPRESENTS THE TRAINING PROCEDURE WITHOUT ADVERSARIAL SIGNALS (I.E., REMOVING THE REGULARIZER $R(\boldsymbol{\theta}_h^0, \boldsymbol{\theta}_h^1)$)

Scheme	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓
TGraM without MAGra	64.8	65.9	247	1836	4407
TGraM with MAGra	**65.7**	**66.8**	**182**	**1482**	**3764**

TABLE V

PERFORMANCE IN COMPARISON ON THE AIR-MOT TEST SET WITH GRADNORM [35] AND UNCERTAINTY WEIGHT [51]

Scheme	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓
GradNorm [35]	65.1	66.2	226	1948	4291
Uncertainty Weight [51]	65.5	65.6	219	1695	**3627**
MAGra	**65.7**	**66.8**	**182**	**1482**	3764

3) Effectiveness of MAGra: As shown in Fig. 5, after training with the MAGra strategy, the gradient distribution returned by the head network is basically similar. This indicates that discriminative information about the task source in the gradient is eliminated. As shown in Table IV, MAGra can reduce IDs and boost IDF1, helping to achieve long-term data association. MAGra not only learns effective ReID embedding but also improves detection performance, especially on MOTA. To fully verify the effectiveness of MAGra, we also use different MTL methods to train the tracker. As shown in Table V, the performance improvement brought by our MAGra is more obvious. We believe this is because the ReID loss in other methods still only focuses on the intraclass variance, inconsistent with the detection loss in joint training. However, our MAGra introduces $R(\boldsymbol{\theta}_h^0, \boldsymbol{\theta}_h^1)$ in (20) to generate adversarial signals to regularize each head network, to realize adaptive and equal learning between different tasks.

D. Comparison With the State-of-the-Arts

1) Quantitative Analysis: As shown in Table VI, we compare the proposed TGraM with the state-of-the-art trackers on the test set of AIR-MOT. The two best results in Table VI are highlighted in red and blue, respectively. The performance of

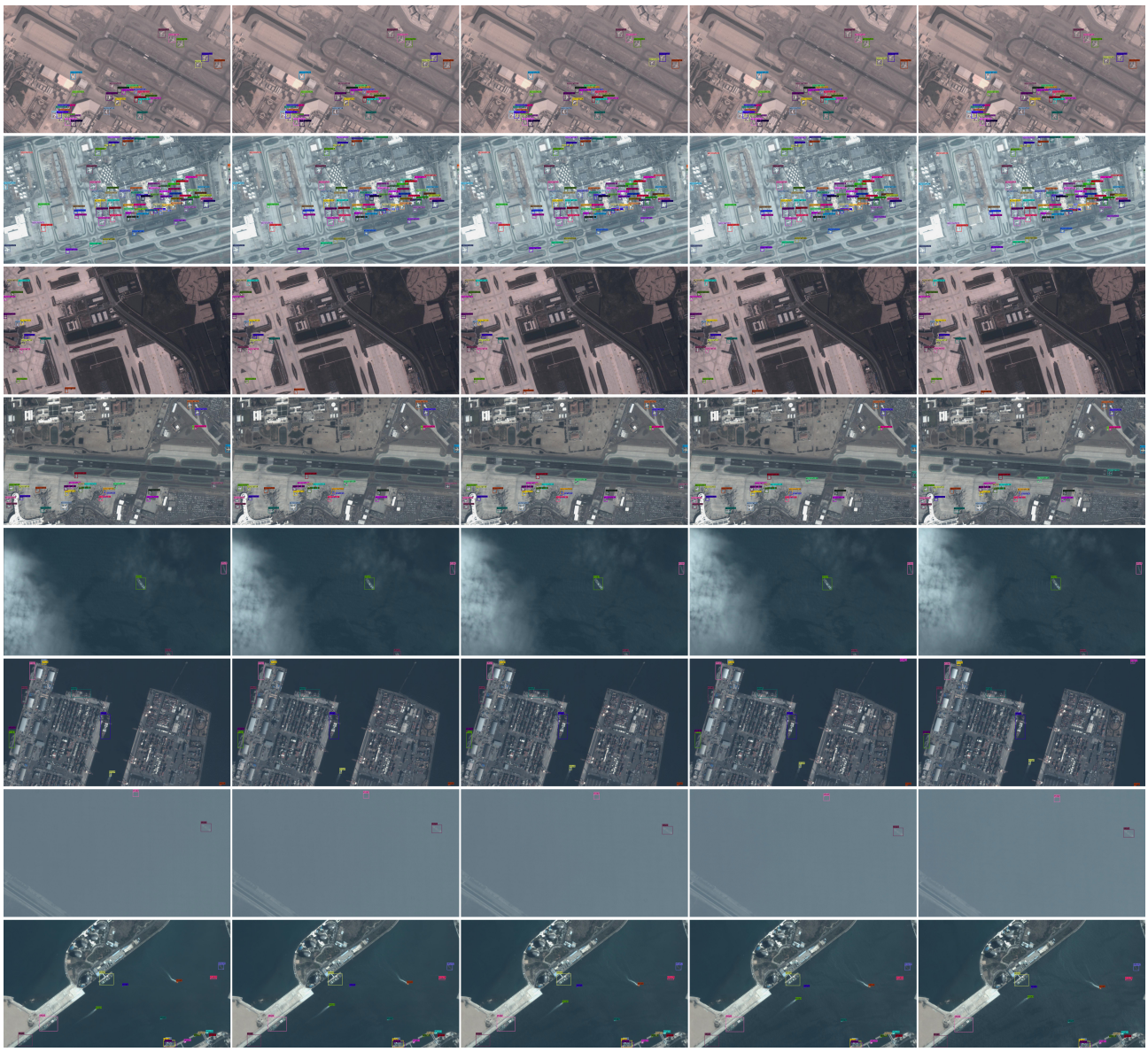

Fig. 9. Visualization that TGraM tracks objects on the AIR-MOT test set. Each row represents an independent sequence of satellite video frames. The results show that TGraM has excellent capability for tracking multiscale objects with complex backgrounds.

our TGraM tracker on AIR-MOT is 1.2 MOTA higher than the second-best tracker. Compared with the trackers of the JDT paradigm (i.e., TubeTK [59], CTracker [21], JDE [22], UMA [61], CenterTrack [19], GSDT [23], FairMOT [48], and TraDeS [24]), we have achieved the best results on MOTA, IDF1, FN, MT, and other metrics. The superiority of our TGraM is not only reflected in accuracy, but also efficiency, as shown in Table VII. Because the trackers of the DBT paradigm (i.e., DeepSORT [12], RAN [56], HOGM [57], DAN [58], Tracktor + CTdet [20], and CKDNet + SMT-Net [11]) run detection and tracking separately, their processing speed obviously cannot meet the real-time requirements. However, the inference speed of our TGraM is greater than the frame rate of Jilin-1 satellite video, suitable for satellite on-orbit processing. What is more, TGraM also has significant advantages in terms of calculation and parameters, which reduces the resource requirements for the spaceborne computing platform.

2) Qualitative Analysis: Fig. 9 demonstrates part of the visualization results of our TGraM on AIR-MOT. We find that TGraM can accurately track multiscale objects in diverse and complex backgrounds. Even TGraM can detect difficult samples in some intermediate frames (e.g., dense or blurred objects). We believe this is due to the graph structure constructed by STeRe, which propagates motion clues from previous frames to generate spatiotemporal features. In addition, the proposed MAGra strategy ensures that the tracker learns the detection and ReID fairly and accurately and realizes the mutual improvement between the two tasks.

E. Discussion

1) Hyperparameter Analysis: Hyperparameter λ should be carefully tuned to achieve an excellent performance of TGraM. λ adjusts the tradeoff between the minimization of distribution divergence and risk in the optimization problem given by (20). To analyze the influence of λ value on the performance of

TABLE VI

ACCURACY COMPARISON ON THE AIR-MOT TEST SET. THE TOP TWO RESULTS ARE HIGHLIGHTED IN **RED** AND **BLUE**, RESPECTIVELY

Method	Publication	Year	DBT	JDT	MOTA↑	IDF1↑	IDP↑	IDR↑	MT↑	ML↓	FP↓	FN↓	IDs↓
DeepSORT [12]	ICIP	2017	✓		56.3	53.6	56.0	51.4	28.3%	29.5%	8364	12,057	1427
RAN [56]	WACV	2018	✓		57.9	55.0	58.3	52.1	28.7%	29.1%	7682	11,362	1249
HOGM [57]	ICPR	2018	✓		58.1	55.1	57.9	52.6	28.9%	29.3%	7041	11,208	1315
DAN [58]	TPAMI	2019	✓		58.4	55.4	57.8	53.2	29.6%	28.7%	6922	10,964	962
Tracktor+CTdet [20]	ICCV	2019	✓		59.3	56.8	59.1	54.7	28.8%	27.2%	7183	10,527	883
CKDNet+SMTNet [11]	ISPRS	2021	✓		64.2	64.9	65.7	64.3	36.1%	23.4%	2816	4057	304
TubeTK [59]	CVPR	2020		✓	60.7	58.1	60.6	55.8	31.9%	26.4%	5207	7364	639
CTracker [21]	ECCV	2020		✓	61.5	61.8	64.2	59.6	32.9%	25.2%	4592	3952	374
JDE [22]	ECCV	2020		✓	62.9	64.5	65.9	63.2	31.4%	22.9%	3749	4870	329
UMA [61]	CVPR	2020		✓	63.4	64.3	66.3	62.6	31.8%	23.1%	3049	4253	296
CenterTrack [19]	ECCV	2020		✓	63.6	65.1	67.5	62.9	34.7%	23.8%	4820	6028	215
GSDT [23]	ICRA	2021		✓	63.9	65.9	67.1	**64.9**	35.9%	**22.4%**	2894	4071	283
FairMOT [48]	Arxiv	2020		✓	64.1	65.4	66.8	64.1	**36.5%**	24.1%	2518	**3485**	273
TraDeS [24]	CVPR	2021		✓	**64.5**	**66.2**	**67.9**	64.6	**36.9%**	**22.1%**	**1759**	3928	**157**
TGraM (Ours)	-	-		✓	**65.7**	**66.8**	**68.6**	**65.2**	36.2%	22.5%	**1484**	**3764**	**182**

TABLE VII

EFFICIENCY COMPARISON ON THE AIR-MOT TEST SET. MULTI-ADDS AND FPS ARE ESTIMATED ON AN NVIDIA TITAN RTX GPU

Method	Params(M)↓	Multi-adds(G)↓	FPS↑
TubeTK [59]	78.3	214.9	3.1
JDE [22]	67.2	67.7	19.2
UMA [61]	58.1	173.6	5.8
GSDT [23]	48.4	126.2	18.5
CTracker [21]	33.9	28.5	**34.5**
TraDeS [24]	21.7	44.9	17.6
FairMOT [48]	19.8	72.2	21.3
CenterTrack [19]	18.6	37.1	24.8
TGraM (Ours)	**12.3**	**22.3**	31.7

[a] 1 M = 1×10^6
[b] 1 G = 1×10^9

TABLE VIII

PERFORMANCE IN COMPARISON ON THE AIR-MOT TEST SET WITH DIFFERENT VALUE OF THE HYPERPARAMETER λ

Scheme	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓
$\lambda = 0.05$	64.9	65.7	253	1873	4427
$\lambda = 0.1$	65.0	65.9	231	1795	3982
$\lambda = 0.5$	65.2	66.4	209	1633	**3459**
$\lambda = 1$	**65.7**	**66.8**	182	**1482**	3764
$\lambda = 5$	65.6	66.8	**169**	1506	3898
$\lambda = 10$	65.4	66.5	178	1491	3795

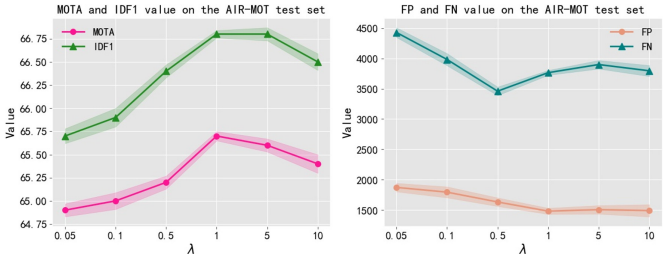

Fig. 10. Sensitivity analysis of hyperparameter λ for TGraM on the AIR-MOT test set, when λ = 0.05, 0.1, 0.5, 1, 5, 10.

Fig. 11. Failure cases of point-like objects on the AIR-MOT test set. The first and third rows show failure tracking results by TGraM. The second and fourth rows label ground-truth objects.

TGraM, we perform multi-object tracking with λ in the range of {0.05, 0.1, 0.5, 1, 5, 10} on AIR-MOT. The experimental results are shown in Table VIII and Fig. 10. As λ increases (0.05 $\rightarrow$ 1), the tracking performance is improved due to the greater penalty of distribution divergence. In addition, it can be seen from the curves of FN and FP that the detection has also been slightly improved. While λ continues to increase (1 $\rightarrow$ 10), the tracking performance does not change significantly and even slightly decreases. We believe that this is due to the neglect of minimizing the training risk of the tracker and the excessive focus on the gradient classifier. Therefore, to implement a balanced performance between the minimization of distribution divergence and source risk,

λ cannot be too small or too large. In our experiments, we set λ to 1 based on experience and obtain good performance.

2) Limitations: TGraM can achieve good multi-object tracking performance in harsh environments, but there are still some errors in point-like object tracking. Point-like objects are usually recognizable visually but occupy very few pixels. Most point-like objects are very similar to the background in space, easily leading to omissions in detection. Especially for point-like objects with short trajectories, this problem is even more serious due to the limited temporal information. Fig. 11 shows two typical cases of point-like object tracking errors, including several missed detections. In the future, we aim to improve the model to further improve the robustness of point-like object detection and tracking.

V. Conclusion

This article presents a novel online JDT framework TGraM for multi-object tracking in satellite video. Compared with previous methods, TGraM realizes real-time parallel tracking of multiple objects in complex environments. Concretely, the STeRe module employs graph structure to capture the relations between video frames and explore potential high-order spatiotemporal correlations. The MAGra strategy introduces the adversarial signal to regularize task-specific networks and eliminates discriminative information of gradient sources. In future work, we will pay more attention to the tracking of point-like objects in wide-area scenarios and further improve our TGraM to make it more fault-tolerant. In addition, we will continue to expand the dataset to build AIR-MOT 2.0, including adding more videos and more labeled objects (e.g., moving vehicles).

References

[1] H. Liu, Y. Gu, T. Wang, and S. Li, "Satellite video super-resolution based on adaptively spatiotemporal neighbors and nonlocal similarity regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8372–8383, Dec. 2020.

[2] R. Almar, E. W. J. Bergsma, P. Maisongrande, and L. P. M. de Almeida, "Wave-derived coastal bathymetry from satellite video imagery: A showcase with Pleiades persistent mode," *Remote Sens. Environ.*, vol. 231, Sep. 2019, Art. no. 111263.

[3] Z. He, J. Li, L. Liu, D. He, and M. Xiao, "Multiframe video satellite image super-resolution via attention-based residual learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[4] B. Du, Y. Sun, S. Cai, C. Wu, and Q. Du, "Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 168–172, Feb. 2018.

[5] Y. Guo, D. Yang, and Z. Chen, "Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by Kalman filter," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3538–3551, Sep. 2019.

[6] J. Shao, B. Du, C. Wu, and L. Zhang, "Tracking objects from satellite videos: A velocity feature based correlation filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7860–7871, Oct. 2019.

[7] J. Shao, B. Du, C. Wu, and L. Zhang, "Can we track targets from space? A hybrid kernel correlation filter tracker for satellite video," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8719–8731, Nov. 2019.

[8] Z. Hu, D. Yang, K. Zhang, and Z. Chen, "Object tracking in satellite videos based on convolutional regression network with appearance and motion features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 783–793, 2020.

[9] S. Xuan, S. Li, M. Han, X. Wan, and G. S. Xia, "Object tracking in satellite videos by improved correlation filters with motion estimations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1074–1086, Feb. 2020.

[10] Y. Wang, T. Wang, G. Zhang, Q. Cheng, and J.-Q. Wu, "Small target tracking in satellite videos using background compensation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7010–7021, Oct. 2020.

[11] J. Feng *et al.*, "Cross-frame keypoint-based and spatial motion information-guided networks for moving vehicle detection and tracking in satellite videos," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 116–130, Jul. 2021.

[12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[13] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3701–3710.

[14] A. Milan *et al.*, "Mot16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[15] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2018, pp. 3987–3997.

[16] L. Porzi, M. Hofinger, I. Ruiz, J. Serrat, S. R. Bulò, and P. Kontschieder, "Learning multi-object tracking and segmentation from automatic annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 6845–6854.

[17] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10359–10366.

[18] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," 2021, *arXiv:2101.02702*.

[19] X. Zhou *et al.*, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[20] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[21] J. Peng *et al.*, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[22] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[23] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 13708–13715.

[24] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 12352–12361.

[25] Z. Lu, V. Rathod, R. Votel, and J. Huang, "RetinaTrack: Online single stage joint detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14656–14666.

[26] M. Wang, J. Tighe, and D. Modolo, "Combining detection and tracking for human pose estimation in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 11085–11093.

[27] D. Chen, S. Zhang, W. Ouyang, J. Yang, and Y. Tai, "Person search via a mask-guided two-stream CNN model," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 734–750.

[28] H. Drucker and Y. Le Cun, "Double backpropagation increasing generalization performance," in *Proc. Seattle Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, 1991, pp. 145–150.

[29] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9245–9255.

[30] M. Teichmann, M. Weber, M. Zollner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time joint semantic reasoning for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1013–1020.

[31] Q. He, X. Sun, Z. Yan, and K. Fu, "DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[33] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[34] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1858–1868.

[35] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," 2017, *arXiv:1711.02257*.

[36] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-net: Multi-tasks guided Prediction-and-Distillation network for simultaneous depth estimation and scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 675–684.

[37] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4106–4115.

[38] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint taskrecursive learning for semantic segmentation and depth estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 235–251.

[39] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multitask learning?" 2019, *arXiv:1905.07553*.

[40] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov. 2005.

[41] A. A. Rusu *et al.*, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[42] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, Mar. 2020.

[43] K. James *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[44] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017.

[45] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016.

[46] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015.

[47] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2017.

[48] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," 2020, *arXiv:2004.01888*.

[49] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.

[50] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 137–144.

[51] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[52] L. V. D. Maaten, "Barnes-hut-SNE," 2013, *arXiv:1301.3342*.

[53] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[55] E. Apostolidis and V. Mezaris, "Fast shot segmentation combining global and local visual descriptors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6583–6587.

[56] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.

[57] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2018, pp. 1809–1814.

[58] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, Jan. 2021.

[59] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6307–6317.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[61] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 6767–6776.

[62] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008, doi: 10.1155/2008/246309.

[63] J. He, Z. Huang, N. Wang, and Z. Zhang, "Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 5299–5309.

Qibin He (Graduate Student Member, IEEE) received the B.Sc. degree from Beijing Institute of Technology, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, and the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing.

His research interests include computer vision and remote sensing image interpretation.

Xian Sun (Senior Member, IEEE) received the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences (CAS), China, in 2009.

He was a Visiting Scholar with the Karlsruher Institut für Technologie, Karlsruhe, Germany, in 2013. He is currently a Professor with the Aerospace Information Research Institute, CAS. His research interests include computer vision, geospatial data mining, and remote sensing image understanding.

Dr. Sun was a recipient of the Outstanding Science and Technology Achievement Prize at CAS in 2016 and the First Prize for the State Scientific and Technological Progress of China in 2019. He serves as an Associate Editor for IEEE ACCESS and a Guest Editor for a special issue of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and other journals.

Zhiyuan Yan (Member, IEEE) received the B.Sc. degree from Xiamen University, Xiamen, China, in 2016, and the M.Sc. degree from the Aerospace Information Research Institute, Chinese Academy of Sciences (CAS), Beijing, China, in 2019.

She is currently an Assistant Engineer with the Aerospace Information Research Institute, CAS. Her research interests include computer vision and remote sensing image analysis.

Beibei Li received the B.Sc. degree in satellite photography from Wuhan University, Wuhan, China, in 2014.

He is currently an Assistant Research Fellow with Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Beijing, China, and Chang Guang Satellite Technology Company, Ltd., Jilin, China. His research interests include the industrialization of remote sensing applications and satellite data processing.

Kun Fu (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1995, 1999, and 2002, respectively.

He is currently a Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences (CAS), Beijing, China. His research interests include computer vision, remote sensing image understanding, and geospatial data mining and visualization.

Dr. Fu was a recipient of the First Prize for the State Scientific and Technological Progress of China in 2015 and 2019, the Outstanding Science and Technology Achievement Prize at the CAS in 2016, the Scientific and Technological Innovation Leading Talent by the National High-Level Talents Special Support Plan in 2017, and the Distinguished Young Scholars from the National Natural Science Foundation of China in 2017.