Cluster Head Detection for Hierarchical UAV Swarm With Graph Self-Supervised Learning

Zhiyu Mou ©, Feifei Gao ©, Fellow, IEEE, Jun Liu ©, Member, IEEE, Xiang Yun ©, and Qihui Wu ©

Abstract—In this paper, we study the cluster head detection problem of a hierarchical unmanned aerial vehicle (UAV) swarm network (USNET), consisting of multiple double-level clusters, from the perspective of the observers who are assumed not to know any internal information on the USNET. Specifically, we propose a multi-cluster graph attention self-supervised learning (MC-GASSL) framework for the observers. The MC-GASSL comprises two main algorithms, including the sequential metric learning (SML) algorithm for detecting the clusters in the USNET, and the graph attention self-supervised learning (GASSL) algorithm for detecting the cluster head in each cluster. The basic idea of the SML is to learn an embedding space where the UAVs can be easily clustered based on Euclidean distances, and the GASSL works by calculating the attention weights between UAVs and finding the attention center of the cluster. Extensive numerical results validate the effectiveness of the MC-GASSL. Moreover, the ablation study shows that the GASSL can detect the cluster head in single clusters with over 98% detection rate on average. The ablation study also shows that the clustering purity of the USNET with the SML exceeds that with traditional clustering algorithms by at least 10% on average.

Index Terms—Cluster head detection, graph attention network, self-supervised learning, hierarchical UAV swarm.

I. INTRODUCTION

UNMANNED aerial vehicle (UAV) swarm network (USNET) composed of several hierarchical UAV clusters usually has structural advantages over flat UAV swarms in many aspects, including collective management, communication efficiency, and labor divisions [1]. As a result, USNET has become a critical technology to a broad of UAV-aided scenarios, such as data collections [2], [3], area coverage [4] and securities [5]. Generating behaviors of UAVs is an important part in the USNET technique and has been investigated in many literatures. For example, the authors in [2], [4], [6], [7], [8] have studied the trajectory planning algorithms of USNET exhaustively in various scenes, while [9], [10], [11] and [12], [13] have proposed efficient methods on data transmissions and charging schedules for the USNET, respectively. Note that apart from the UAVs themselves, there usually exists another side of agents in practical applications of the USNET technique that respond to the behaviors of UAVs, such like ground users, clients or defenders [14]. They can all be referred to as observers. As high-level cluster head UAVs (HUAVs) often act as control and communication centers for low-level follower UAVs (FUAVs) in each UAV cluster [15], one of the prerequisites for observers is to detect the HUAVs of the USNET [16]. In fact, detecting the HUAVs is urgently needed in many practical scenes, especially in those involving offensive and defensive confrontations [17]. Many works have studied the cluster head selection problems [34], [35], [36], [37]. However, these problems mainly focus on selecting a cluster head from the perspective of the USNET itself to improve its own performance or abilities. Instead, the HUAV detection problem described above aims at let the observers know more about the inside structure of the USNET and help with their further decisions. Besides, the USNET usually know much information about each UAV, while the observers nearly do not know any prior information on the UAVs, especially in the confrontations scenarios. Actually, to the best of our knowledge, few literatures have deeply investigated the cluster head detection of USNET from the perspective of observers.

The cluster head detection of the USNET is a complex and even abstruse problem that faces many challenges [18]. The first challenge lies in the indistinguishability of the appearance of UAVs from different levels [19]. Specifically, the observers cannot tell the difference between HUAVs and FUAVs exclusively from exterior. Instead, they can only detect the HUAVs
by mining the UAVs’ flying patterns, such as patterns of positions, speeds, accelerations, etc., which makes the cluster head detection a non-trivial problem. Although there usually exists a function describing the flying rule of FUAVs with respect to HUAVs, named as the inherent following strategy (IFS), the observers cannot obtain the specific expression of such function in advance, especially in confrontation scenarios [17]. Besides, the communications links (CLs) between FUAVs and HUAVs that can help with the cluster head detection of the USNET are also unavailable to the observers. Hence, particular features of the IFS or the CLs cannot be utilized in designing the cluster head detection methods, which makes up the second challenge. Thirdly, the UAVs are constantly moving in three-dimensional (3D) spaces, making the USNET a dynamic graph instead of a static one. Therefore, the features of each UAV are composed of sequential position and speed vectors with various lengths, which increases the difficulties in finding the HUAVs in the USNET. Moreover, the number of UAV clusters in the USNET is usually not available to observers, which formulates another challenge in the cluster head detection problem.

In this paper, we study the cluster head detection problem of a hierarchical USNET consisting of multiple double-level clusters from the perspective of observers who nearly do not know any internal information on the USNET. To the best of our knowledge, we are the first to study such problem. The main contributions are summarized as follows.

- Firstly, we come up with a novel way to look at the HUA V in each cluster, i.e., viewing them as the attention center in each cluster. Based on this idea, we propose a graph attention self-supervised learning (GASSL) algorithm to detect the HUAVs in single clusters. Specifically, the GASSL calculates the attention weights of each UAV with respect to all other UAVs in the cluster and determines the HUA V as the UAV attaining the most attention weights. We also prove that the GASSL satisfies the necessary condition of detecting the HUA V within single clusters.

- Secondly, we develop a sequential metric learning (SML) algorithm to detect the clusters in the USNET. Specifically, the SML learns an embedding space for all UAVs, where the embeddings of UAVs from the same cluster stay close to each other while those of UAVs in distinct clusters remain distant, and clusters the USNET with clustering method based on Euclidean distances, such as K-means.

- Combining the GASSL algorithm and the SML algorithm, we propose the MC-GASSL algorithm to detect the HUAVs in the USNET with multiple clusters. Numerical results show that the MC-GASSL algorithm can efficiently detect all the HUAVs in the USNET under various IFSs and cluster numbers with low detection redundancies, which validates the effectiveness of it. We also analyze the computational complexity of the MC-GASSL algorithm, and the experiment results show that it can work in a nearly real-time manner.

- Moreover, the ablation studies on the GASSL show that the GASSL can detect the HUAVs in single clusters under various kinds of IFSs with over 98% detection rate on average. Moreover, the ablation studies on the SML show that the clustering purity of the USNET with MC-GASSL exceeds that with traditional clustering algorithms by at least 10% on average.

The rest parts of this paper are organized as follows. Section II presents the system models of the cluster head detection problem. Section III describes the proposed MC-GASSL framework, where Section III-A presents the details of the GASSL algorithm and Section III-B describes the details of the SML algorithm. Simulation results and analysis are provided in Section IV, and conclusions are made in Section V. The abbreviations are summarized in Table I, and the notations are summarized in Supplementary Material I. The preliminaries of the graph attention network and deep metric learning are presented in Supplementary Material II.

### II. SYSTEM MODEL

As shown in Fig. 1, we consider a two-level USNET, $\mathcal{U}_{\text{NET}}$, with $M \in \mathbb{N}_+$ clusters, where the $j$-th cluster is composed of one HUA V and $m_j \in \mathbb{N}_+$ FUAVs with exactly the same appearance, $j \in \mathcal{M} \triangleq \{1, 2, ..., M\}$. The total number of UAVs in $\mathcal{U}_{\text{NET}}$ can be calculated as $N \triangleq \sum_{j=1}^{M} (1 + m_j) = M + \sum_{j=1}^{M} m_j$. For convenience, each UAV is endowed with a fixed index $i \in \mathcal{N} \triangleq \{1, 2, ..., N\}$. Hereafter, we use UAV, FUA V and HUA V to represent a certain UAV, FUA V and HUA V with index $i$, respectively. In addition, we denote the index set of all UAVs in $\mathcal{U}_{\text{NET}}$ as $\mathcal{H} \subseteq \mathcal{N}$, and the index set of all UAVs within the $j$-th cluster as $\mathcal{N}_j \subseteq \mathcal{N}$, $\forall j$. Each FUA V has a communication link (CL) to the HUA V in the same cluster, but has no CLs to other UAVs. Establish an $X$-$Y$-$Z$ Cartesian coordinate for the USNET, and let the position of UAV $i$ at time step $t$ be $\mathbf{p}_{i,t} = [x_{i,t}, y_{i,t}, z_{i,t}]^T \in \mathbb{R}^3$, $t \in \mathbb{N}$, where $x_{i,t}$, $y_{i,t}$ and $z_{i,t}$ represent the $X$, $Y$ and $Z$ axis components, respectively. The speed of UAV $i$ at time step $t$ can then be defined as $\mathbf{v}_{i,t} = \mathbf{p}_{i,t+1} - \mathbf{p}_{i,t}$. Without loss of generality, we suppose that the HUAVs determine their speeds independently, while the FUAVs follow the HUAVs within the same clusters. To be specific, each FUA V in $\mathcal{U}_{\text{NET}}$ obeys an identical IFS. We suppose the speeds of FUAVs only depend on their historical

---

**Table I. The Summarization of Abbreviations**

| Abbreviations | Full Name |
|---------------|-----------|
| UAV           | unmanned aerial vehicle |
| USNET         | unmanned aerial vehicle swarm network |
| HUA V         | high-level cluster head UAV |
| FUA V         | low-level follower UAV |
| GASSL         | graph attention cluster head UAV |
| AGAT          | multi-cluster graph attention |
| MC-GASSL      | self-supervised learning |
| AGAT          | adaptive graph attention network |
| IFSN          | inherent follow strategy network |
| IFS           | inherent follow strategy |
| CL            | communication link |

---

**THE SUMMARIZATION OF ABBREVIATIONS**

| Abbreviations | Full Name |
|---------------|-----------|
| UAV           | unmanned aerial vehicle |
| USNET         | unmanned aerial vehicle swarm network |
| HUA V         | high-level cluster head UAV |
| FUA V         | low-level follower UAV |
| GASSL         | graph attention cluster head UAV |
| AGAT          | adaptive graph attention network |
| IFSN          | inherent follow strategy network |
| IFS           | inherent follow strategy |
| CL            | communication link |
positions in the past $T$ time steps, as well as their HUAVs’ historical speeds and positions in the past $T$ time steps, where $T_0 \in \mathbb{N}_+$ is a constant. The IFS can then be represented as a function $f(\cdot)$. Specifically, let UAV $i_F$ and UAV $i_j,H$ be an FUAV and an HUAV within the same cluster, respectively. The speed of FUAV $i_F$ at time step $t+1$ can be expressed as

$$v_{i_F,t+1} = f(P_{i_F,t+1}, v_{i_j,H,t}, P_{i_j,H,t-T_0+1}),$$

(1)

where $P_{i_F,t+1} \triangleq [P_{i_F,t+1}, \cdots, P_{i_F,t-T_0+1}]^T \in \mathbb{R}^{T_0+1 \times 3}$ and $V_{i_j,H,t} \triangleq [v_{i_j,H,t}, \cdots, v_{i_j,H,t-T_0+1}]^T \in \mathbb{R}^{T_0 \times 3}$. Note that (1) can encompass a wide range of functions. Examples of specific expressions of the IFS are shown in Table III. Note that as the HUAVs decide their speeds independently, UAVs from different clusters can be spatially mixed together. To avoid collisions, we assume that each UAV adopts the artificial potential field (APF) method [33] whose basic idea is described in the Supplementary Material VI.

A. Problem Settings

We study the cluster head detection problem of $U_{\NET}$ from the perspective of observers. Specifically, as the observers are not on the same side\(^2\) with $U_{\NET}$, they cannot obtain any information on the structures of $U_{\NET}$ in advance, including the number of clusters $M$, the number of FUAVs in each cluster $m_j$, and the CLs. Moreover, the observers are assumed to know the inputs to the IFS are $P_{i_F,t+1}$, $V_{i_j,H,t}$, $P_{i_j,H,t-T_0+1}$, but do not know the specific expressions of the IFS with these terms. In fact, the observers can only observe the positions and velocities of UAVs at each time step and detect the HUAVs by mining the relationship between these observations. All the perceptual information of the observers on $U_{\NET}$ is summarized in Table II. The goal of observers is to detect all the HUAVs in $U_{\NET}$ as accurately as possible.

\(^1\)See Supplementary Material III.A for further illustrations on (1).

\(^2\)For example, the observers are ground defenders, defending against $U_{\NET}$.

III. Multi-Cluster Graph Attention Self-Supervised Learning

To solve the cluster head detection problem of $U_{\NET}$ from the perspective of observers, we propose a multi-cluster graph attention self-supervised learning (MC-GASSL) framework. Specifically, the MC-GASSL framework consists of two main algorithms, including a sequential metric learning (SML) algorithm for clustering $U_{\NET}$, and a graph attention self-supervised learning (GASSL) algorithm for detecting HUAVs in each cluster. The MC-GASSL framework works in an iterative manner, alternating between clustering $U_{\NET}$ with the SML and detecting the HUAVs. The HUAVs in each cluster with the GASSL.

A. GASSL: Detecting the HUA in a Single Cluster

Suppose we have accurately clustered the USNET $U_{\NET}$. Without the loss of generality, let us consider the detection of the HUA in the $j$-th cluster, where the HUA takes the lead of $m_j$ FUAVs. For convenience, we here denote the index of the HUA in the $j$-th cluster as $i_j,H$, and denote the index set of all

| Information on $U_{\NET}$ | known/unknown to Observers |
|-----------------------------|-----------------------------|
| HUAVs $H$                   | unknown                     |
| cluster label of each UAV   | unknown                     |
| the number of clusters $M$  | unknown                     |
| FUAV number in each cluster, $m_j, V_j$ | unknown |
| CLs                         | unknown                     |
| specific expressions of the IFS $f(\cdot)$ | unknown |
| inputs to the IFS $f(\cdot)$ | known                      |
| positions $x_{j,t}$ and speeds $v_{j,t}$ of UAVs | known |

Fig. 1. Observers detect the cluster heads of a two-level USNET.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
the UAVs in the cluster as \( N_j \). As the IFS \( f(\cdot) \) inherently obeyed by each FUAV is only a function of positions and velocities of the HUA \( V_{ij,H} \) and the FUAV itself, each FUAV can be viewed as only focusing on the HUA \( V_{ij,H} \) in addition to itself and paying little attention to other FUAVs. In this way, the HUA \( V_{ij,H} \) acts as the attention center among all UAVs in the \( j \)-th cluster. Based on this point of view, we develop the GASSL algorithm to detect the HUA in each cluster. Its basic idea is to calculate the attention weights of each UAV \( i \) in the \( j \)-th cluster with respect to all other UAVs in the \( j \)-th cluster and determine the HUA \( V_{ij,H} \) as the UAV that attains the most attention weights.

To calculate the attention weights, we design a neural network with self-supervised learning schemes for each UAV \( i \) in the \( j \)-th cluster, as shown in Fig. 2. Specifically, the neural network is composed of two parts, including an adaptive graph attention network (AGAT) and an inherent follow strategy network (IFSN) \( \hat{f}(\cdot) \). The goal of the AGAT is to estimate the history position \( p_{ij,H,t-T_0+1} \) and speeds \( v_{ij,H,t} \) of the HUA \( V_{ij,H} \) based on the inputs of the history position and speeds of all UAVs in the \( j \)-th cluster, i.e.,

\[
\hat{V}_{i,t} \rightarrow V_{i,t}, \quad \hat{p}_{i,t-T_0+1} \rightarrow p_{i,t-T_0+1},
\]

where estimations \( \hat{V}_{i,t} \triangleq [\hat{v}_{i,t-T_0+1}, \hat{v}_{i,t-T_0+2}, ..., \hat{v}_{i,t}]^T \) and \( \hat{p}_{i,t-T_0+1} \) are the outputs of the AGAT. The goal of the IFSN \( \hat{f}(\cdot) \) is to fit IFS \( f(\cdot) \), i.e.,

\[
\hat{f}(\cdot) \rightarrow f(\cdot),
\]

where the input of the IFSN \( \hat{f}(\cdot) \) is the output of the AGAT, \( \hat{V}_{i,t} \) and \( \hat{p}_{i,t-T_0+1} \), as well as the historical positions of UAV \( i \) itself, \( \hat{V}_{i,t} \) and \( \hat{p}_{i,t-T_0+1} \), and the output of the IFSN \( \hat{f}(\cdot) \) is the prediction of the speed of UAV \( i \) at the next time step, \( \hat{v}_{i,t+1} \). Note that \( \hat{v}_{i,t+1} \) is also the output of the whole neural network.

The inference processes of the AGAT and IFSN, as well as the design of the loss function is described as follows.

1) AGAT: As shown in Fig. 2, the history speeds \( V_{i,t} \) and position \( p_{i,t-T_0+1} \) of UAV \( i \) are concatenated and pre-processed with a trainable multi-layer perceptron (MLP) \( L(\cdot) \) to obtain the query data. Note that the MLP \( L(\cdot) \) acts as the query matrix in a traditional graph attention network (GAT) [22]. As there are \( T_0 \) history speeds data and one history position data of each UAV \( i' \), we construct \( T_0+1 \) attention head modules in the AGAT. The \( l \)-th attention head is composed of a trainable MLP \( s_l(\cdot) \) acting as the similarity function, and a trainable parameter \( K_l \in \mathbb{R}^{K \times 3} \), where \( K \in \mathbb{N}_+ \) is a hyperparameter and \( l \in \{1, 2, ..., T_0+1\} \). The first \( T_0 \) attention heads deal with the history speeds of all UAVs, while the last attention head deals with the historical positions of all UAVs. Specifically, the \( l \)-th \( (l \neq T_0+1) \) attention head transforms the speeds of all UAVs at time step \( t-l+1 \) linearly by \( K_l \), obtaining the key data of the \( l \)-th attention head. Each key date is concatenated with query data and processed by similarity function \( s_l(\cdot) \) and softmax operation, as shown in (5), where \( \mathcal{Z} \) represents the concatenation operation, and \( \alpha_{i,i',l} \) is the attention value of UAV \( i \) with respect to the speed \( v_{i',t-l+1} \) of UAV \( i' \). The last attention head processes the positions of all UAVs at time step \( t-T_0+1 \) in the same way and outputs the attention values \( \alpha_{i,i',T_0+1}, v_{i',t} \) with respect to \( p_{i',t-T_0+1} \) as (6). The attention weight \( \alpha_{i,i'} \) of UAV \( i \) with respect to UAV \( i' \) is calculated as the average attention values in all \( T_0+1 \) attention heads, i.e.,

\[
\alpha_{i,i'} = \frac{1}{T_0+1} \sum_{l=1}^{T_0+1} \alpha_{i,i',l}.
\]
Denote the parameters of the AGAT for UAV $i$ as $\Phi_i$. Note that the attention weight $\alpha_{i',i}$ may not equal to $\alpha_{i,i'}$ since $\alpha_{i',i}$ is derived from the AGAT for UAV $i'$ with different parameters $\Phi_i'$. The value data are directly made as the history speeds and positions themselves with no trainable transformation parameters. Then the element $\hat{V}_{i,t-t+1}$ in the output $\hat{V}_{i,t}$ is calculated as the weighted combination of the value data $v_{i',t-t+1}$ with the attention values of the $l$-th attention head $\alpha_{i,i'}$, i.e., $\hat{V}_{i,t-t+1} = \sum_{i'=1}^{1+m_1} \alpha_{i,i'} v_{i',t-t+1}$, where $l \neq T_0 + 1$. Similarly, the output $\hat{p}_{i,t-T_0+1}$ is derived from the weighted combination of the value data $p_{i',t-T_0+1}$ with the attention values of the last attention head, i.e.,

$$\hat{p}_{i,t-T_0+1} = \sum_{i'=1}^{1+m_1} \alpha_{i,i'} p_{i',t-T_0+1}.$$  

2) IFSN: The IFSN receives $\hat{V}_{i,t}$ and $\hat{p}_{i,t-T_0+1}$ from the AGAT, and concatenates them together with the positions $p_{i,t}$ of UAV $i$. The concatenations are standardized as zero-mean vectors with unit variance to make the speed and position data in the same order of magnitude. The standardized vectors are processed with a trainable MLP to obtain the estimation $\hat{v}_{i,t+1}$ of the speed of UAV $i$ at time step $t + 1$. Note that the parameters of IFSN $\hat{f}(\cdot)$ are shared among all UAVs. We denote the parameters of IFSN $\hat{f}(\cdot)$ as $\Gamma$. Then all the parameters of GASSL for UAV $i$ can be denoted as $\Theta_i = \{\Phi_i, \Gamma\}$.

3) Self-Supervised Loss Function: The loss function $\mathcal{L}(\Theta_i)$ of the whole neural network in the GASSL for UAV $i$ is designed as the square error between the real speed $v_{i,t+1}$ and the estimated speed $\hat{v}_{i,t+1}$, i.e.,

$$\mathcal{L}(\Theta_i) = (v_{i,t+1} - \hat{v}_{i,t+1})^2 = (v_{i,t+1} - \hat{f}(\hat{p}_{i,t}, \hat{V}_{i,t}, \hat{p}_{i,t-T_0+1}))^2.$$  

Note that $\mathcal{L}(\Theta_i) \geq 0$. As proved in Proposition 1, a sufficient condition for $\mathcal{L}(\Theta_i) = 0$ is that the IFSN $\hat{f}(\cdot)$ completely fits the IFS $f(\cdot)$ and UAV $i$ only focuses on the speeds and positions of HUA $i_j,H$. A sufficient condition for $\mathcal{L}(\Theta_i) = 0$ is that

- the IFSN $\hat{f}(\cdot)$ completely fits the IFS $f(\cdot)$, i.e., $\hat{f}(\cdot) = f(\cdot)$;
- UAV $i$ only focuses on the speeds and positions of HUA $i_j,H$, i.e., $\alpha_{i,i',t} = 1$ for $\forall l$ and $\alpha_{i,j',t} = 0$ for $\forall l' \neq i,j; \forall l'$, hold at the same time.

Proof: When $\alpha_{i,i,j',t} = 1, \forall l$ and $\alpha_{i,j',t} = 0, \forall l' \neq i,j,H; \forall l'$, we have

$$\hat{v}_{i,t-l+1} = \sum_{i'=1}^{1+m_1} \alpha_{i,i'} v_{i',t-l+1} = \sum_{i'=1}^{1+m_1} 1 \{i' = i,j,H\} v_{i',t-l+1} = v_{i,t-l+1},$$

and at the same time, we have

$$\hat{p}_{i,t-T_0+1} = \sum_{i'=1}^{1+m_1} \alpha_{i,i'} p_{i',t-T_0+1} = \sum_{i'=1}^{1+m_1} 1 \{i' = i,j,H\} p_{i',t-T_0+1} = p_{i,t-T_0+1}.$$  

Hence, we have $\hat{V}_{i,t} = V_{i,j,H,t}$ and $\hat{p}_{i,t-T_0+1} = p_{i,j,H,t-T_0+1}$. Since $\hat{f}(\cdot) = f(\cdot)$, we can derive

$$\mathcal{L}(\Theta_i) = (v_{i,t+1} - \hat{f}(\hat{p}_{i,t}, \hat{V}_{i,t}, \hat{p}_{i,t-T_0+1}))^2 = (v_{i,t+1} - f(\hat{p}_{i,t}, V_{i,j,H,t}, p_{i,j,H,t-T_0+1}))^2 = (v_{i,t+1} - v_{i,t+1})^2 = 0.$$  

So far, we have completed the proof.

Therefore, we can reduce the loss function $\mathcal{L}(\Theta_i)$ towards zero by training the neural networks in GASSL, and obtain the attention weights $\alpha_{i,j',t}$ of UAV $i$ with respect to all UAVs. Then, we choose the index of the UAV with maximum attention weights as the index of HUA for UAV $i$, i.e.,

$$i^{(i)}_{j,H} \leftarrow \arg\max_{i'} \alpha_{i,j',t}.$$  

Note that $i^{(i)}_{j,H}$ only represents the choice of HUA from the perspective of UAV $i$. We apply the GASSL to all $m+1$ UAVs in the $j$-th cluster and obtain the indexes of HUA from the view of all UAVs. Then the probability of UAV $i$ being the HUA $i_{j,H}$ can be calculated as

$$c_i = \frac{1}{m+1} \sum_{i'} 1 \{i^{(i')}_{j,H} = i\}.$$  

It is worth noting that as the IFS $f(\cdot)$ may only relates to part of the HUA’s history speeds or positions, attention heads dealing with irrelevant speed or position data of HUA can produce attention weights with random values after training. See Supplementary Material III.B for details.
The GASSL selects the UAV with the maximum probability as the HYAV, i.e.,
\[
\hat{i}_{j,H} \leftarrow \arg \max \limits_{i \in N_j} c_i,
\]
where \(\hat{i}_{j,H}\) denotes the index of the selected HYAV. The superiority of the GASSL compared to existing GNNs are presented in Supplementary Material V.A.

B. SML: Clustering the USNET \(\mathcal{U}_{\text{NET}}\)

Let us consider the way to cluster \(\mathcal{U}_{\text{NET}}\). As the UAVs from different clusters can be spatially mixed with each other, \(\mathcal{U}_{\text{NET}}\) cannot be directly clustered in the X-Y-Z Cartesian coordinate based on geometric distances. Inspired by the metric learning algorithms used in many other realms, such as computer vision, we develop the SML algorithm to cluster \(\mathcal{U}_{\text{NET}}\). Its basic idea is to first map each UAV based on its historical positions and velocities into a learned embedding space, where the embeddings of UAVs from the same cluster stay close to each other while the embeddings of UAVs belonging to distinct clusters keeps away from each other; and then cluster the UAVs directly based on the geometric distance in the embedding space by traditional clustering algorithms, such as K-means [24]. Note that as the history speeds and positions of each UAV are sequential data, we should leverage recurrent neural networks (RNNs) to extract the embeddings of it. There are many versions of RNNs, including the vanilla RNN, LSTM, GRU [23], SRU and the bi-direction version of them. Due to the advantages of good performance and fast convergence of the GRU, we here select it as the embedding extractor. See Supplementary Material V.B for detailed explanations on the reasons of selecting the GRU. As shown in Fig. 3, the embedding extractor is composed of \(T_0\) identical GRU units. Specifically, to extract the features of UAV \(i\), the embedding network receives the position \(p_{i,t-T_0+1}\) and the speed vectors in \(v_{i,t}\) of UAV \(i\) sequentially, and outputs the extracted feature \(e_{i,t} \in \mathbb{R}^3\) of it. The inference process of the embedding network and the way of training it is described as follows.

1) Inference Process of the Embedding Network: The \(\tau\)-th GRU unit takes both the output \(h_{i,\tau-1} \in \mathbb{R}^3\) of the previous GRU unit and \(v_{i,t-T_0+\tau}\) as input, and calculates \(h_{i,\tau} \in \mathbb{R}^3\) for the next GRU unit, where \(h_{i,\tau} \equiv p_{i,t-T_0+\tau}\) and \(h_{i,0} \equiv e_{i,t}\), \(\forall \tau \in \{1, 2, \ldots, T_0\}\). Inside the \(\tau\)-th GRU, \(h_{i,\tau-1}\) and \(v_{i,t-T_0+\tau}\) are concatenated and transformed to the reset gate \(g_r \in \mathbb{R}^3\) and update gate \(g_u \in \mathbb{R}^3\), i.e.,
\[
\begin{align*}
  g_r &= \sigma(W_r(h_{i,\tau-1} \downarrow v_{i,t-T_0+\tau})), \\
  g_u &= \sigma(W_u(h_{i,\tau-1} \downarrow v_{i,t-T_0+\tau})),
\end{align*}
\]
where \(W_r \in \mathbb{R}^{3 \times 6}\) and \(W_u \in \mathbb{R}^{3 \times 6}\) are two trainable parameters, and \(\sigma(\cdot)\) is the sigmoid function. The reset gate \(g_r\) resets \(h_{i,\tau-1}\) through Hadamard product, and the obtained reset vector is concatenated with \(v_{i,t-T_0+\tau}\) and transformed to vector \(h'_{i,\tau} \in \mathbb{R}^3\), i.e.,
\[
\begin{align*}
  h'_{i,\tau} &= \tanh \left( W_h(v_{i,t-T_0+\tau} \downarrow (g_r \odot h_{i,\tau-1})) \right),
\end{align*}
\]
where \(W_h \in \mathbb{R}^{3 \times 6}\) is a trainable parameter, and \(\tanh(\cdot)\) is the hyperbolic tangent function. Then the output \(h_{i,\tau}\) of the \(\tau\)-th GRU unit is calculated as
\[
\begin{align*}
  h_{i,\tau} &= (1 - g_u) \odot h_{i,\tau-1} + g_u \odot h'_{i,\tau}.
\end{align*}
\]
Note that each GRU unit shares the same trainable parameters, and we denote the set of all the trainable parameters in a GRU unit as \(W \equiv \{W_r, W_u, W_h\}\).

2) Metric Learning of the Embedding Network: To train the embedding network, we develop a deep metric learning method, as shown in Fig. 3. Specifically, we construct a Siamese network consisting of three identical embedding networks with shared trainable parameters \(W\). For convenience, we refer these embedding networks as anchor network, positive network and negative network. We also design a loss function \(L^s(W)\) in the form of the triplet loss function [27]. The embedding network is learned after training the Siamese network with \(L^s(W)\). The detailed training process is described as follows. Suppose we have a dataset \(D\) of the USNET \(\mathcal{U}_{\text{NET}}\), consisting of the cluster label of each UAV and its historical positions and velocities. During each episode of training, the anchor network and the positive network calculate the features of UAVs from the same cluster, while the negative network calculates the features of the UAV from a distinct cluster. Specifically, we first arbitrarily pick a UAV \(i_n\) from a random cluster \(j\), referred as the anchor UAV. Then we sample a UAV \(i_p\) from the same cluster with hard sampling method [26], referred as the positive UAV, i.e.,
\[
\begin{align*}
  i_p &= \arg \max \limits_{i \in N_j} \left\| \frac{p_{i,t-T_0+1}}{v_{i,t}} - \frac{p_{i_p,t-T_0+1}}{v_{i_p,t}} \right\|_2, \\
  i_n &= \arg \min \limits_{i \in N_j} \left\| \frac{p_{i,t-T_0+1}}{v_{i,t}} - \frac{p_{i_n,t-T_0+1}}{v_{i_n,t}} \right\|_2.
\end{align*}
\]
Afterwards, we input the historical positions and velocities of anchor UAV, positive UAV and negative UAV into the anchor
network, positive network and negative network, respectively, and obtain the corresponding extracted embeddings \(e_{i_a,t}, e_{i_p,t}\) and \(e_{i_n,t}\). We calculate the loss function \(L^s(W)\) as

\[
L^s(W) = \left[ \left\| e_{i_p,t} - e_{i_a,t} \right\|_2^2 - \left\| e_{i_n,t} - e_{i_a,t} \right\|_2 + \gamma \right]_+,
\]

(21)

where \(\gamma > 0\) is a constant, and \(L^s(W) \geq 0\) always holds. Note that \(L^s(W) = 0\) means that the distance between the extracted embeddings of the anchor UAV and the negative UAV, i.e., \(\left\| e_{i_a,t} - e_{i_n,t} \right\|_2\), is at least \(\gamma\) larger than the distance between the extracted embeddings of the anchor UAV and the positive UAV, i.e., \(\left\| e_{i_a,t} - e_{i_p,t} \right\|_2\). This is consistent with the aforementioned basic idea of the SML. In practice, we usually sample several anchor UAVs in various clusters at once and find the corresponding positive and negative UAVs to form a batch of training data. The loss function for the batch can be expressed as

\[
L^s_B(W) = \sum_{b=1}^{B} \left[ \left\| e_{i_p,t,b} - e_{i_a,t,b} \right\|_2^2 - \left\| e_{i_n,t,b} - e_{i_a,t,b} \right\|_2 + \gamma \right]_+,
\]

(22)

where \(B \in \mathbb{N}_+\) is the size of the batch, and \(e_{i_a,t,b}, e_{i_p,t,b}\) and \(e_{i_n,t,b}\) are the \(b\)-th anchor, positive, and negative UAV in the batch, respectively, \(\forall b \in \{1, 2, ..., B\}\). The parameter \(W\) can be updated by gradient descent, i.e.,

\[
W \leftarrow W - \beta' \nabla_W L^s_B(W),
\]

(23)

where \(\beta' > 0\) is the learning rate. We iteratively sample batches from \(D\) and apply (23) to the parameter \(W\) until convergence.

3) Clustering in the Embedding Space: After training the embedding network, it is able to learn a embedding space where the UAVs within the same cluster stay close to each other, while the UAVs in distinct clusters are separated away. Hence, we can leverage traditional clustering methods, such as K-means, spectral clustering, symmetric non-negative matrix factorization (SNMF), etc., to cluster the USNET \(U_{\text{USNET}}\). Recall that as the number of clusters \(M\) is unknown, we need to estimate the number of clusters before clustering the USNET. Hence, the SML utilizes the gap-statistic method [29] to produce the estimation of the number of clusters and leverages the K-means method to cluster the USNET \(U_{\text{USNET}}\) due to its advantages of simplicity and efficiency.

C. MC-GASSL Implementations in Practice

As the dataset \(D\) cannot be obtained by the observers in advance, the metric learning of the embedding network cannot be carried out before the online deployment of the MC-GASSL. This makes the SML unable to accurately cluster the USNET \(U_{\text{USNET}}\) at one time. Hence, in practice, the MC-GASSL works in an iterative manner, alternating between clustering the USNET \(U_{\text{USNET}}\) with the SML and detecting the HUA V in each cluster with the GASSL. The training of the embedding network with metric learning are carried out online during the iterations. However, as the embedding network is untrained before the online deployment of the MC-GASSL, the clustering precision in the first iteration can be low and the time complexity of the online training can be high. Hence, to deal with these two issues, we develop an offline meta learning for the embedding network. The details of the offline meta learning are shown below.

1) Offline Meta Learning for Embedding Network: As the meta learning can find promising initial parameters for neural networks working across multiple tasks [28], we leverage the meta learning to obtain the initial parameters for the embedding network in an offline manner. The construction of the meta learning dataset is described as follows. Specifically, we randomly construct \(F \in \mathbb{N}_+\) ISFs \(f_m\) in the form of (1), where \(m \in \{1, 2, \cdots, F\}\). These can form an IFS set \(F = \{f_m|\forall m\}\). We construct \(F\) USNETs with \(M_0 \in \mathbb{N}_+\) clusters each, where the FUA Vs in the \(m\)-th USNET obey the IFS \(f_m\) and the UAVs leverage the APF to avoid collisions. For each USNET, we simulate the flying of all the UAVs and record the speeds and the positions of each UAV in \(t_0\) time steps as well as its cluster index. The records of \(m\)-th USNET makes up the support dataset \(D^S_{m}\). We re-simulate the flying of all the USNET, and construct the query dataset \(D^Q_{m}\) in the same way. With the constructed datasets \(D^S_{m}\) and \(D^Q_{m}\), we can carry out the meta learning. Specifically, we train the embedding network \(M\) episodes with a two-step update each [28]. To begin with, we randomly initialize the parameter of the embedding network as \(W_0\). In the \(m\)-th episode, we take \(D^Q_{m}\) and \(D^S_{m}\) to update the parameter of the embedding network. Denote the parameter of the embedding network at the \(m\)-th episode as \(W_m\). Specifically, a temporary embedding network with parameter \(\Pi_m\) is endowed with \(W_{m-1}\), i.e., \(\Pi_m \leftarrow W_{m-1}\). The first step is to update the parameter \(\Pi_m\) towards the gradient \(\nabla_{\Pi_m} L^s_B(\Pi_m; D^S_m)\) by \(\beta_{\text{meta}} > 0\) step size, i.e.,

\[
\Pi^1_m = \Pi_m - \alpha_{\text{meta}} \nabla_{\Pi_m} L^s_B(\Pi_m; D^S_m) = W_{m-1} - \alpha_{\text{meta}} \nabla_{W_{m-1}} \left[ \sum_{b=1}^{B} \left[ \left\| e_{i_p,t,b} - e_{i_n,t,b} \right\|_2^2 - \left\| e_{i_n,t,b} - e_{i_a,t,b} \right\|_2 + \gamma \right]_+ \right],
\]

(24)

where \(\Pi^0_m\) is the updated parameter of the temporary embedding network, \(e_{i_p,t,b}, e_{i_n,t,b}\) and \(e_{i_a,t,b}\) are the extracted features of the \(b\)-th anchor, positive, and negative UAV sampled in \(D^S_m\) using hard sampling method. Afterwards, the parameter of the embedding network is updated towards the gradient \(\nabla_{\Pi^1_m} L^s_B(\Pi^1_m; D^Q_m)\) by \(\alpha_{\text{meta}}\) step size, i.e.,

\[
W_m = W_{m-1} - \alpha_{\text{meta}} \nabla_{W_m} L^s_B(\Pi^1_m; D^Q_m) = W_{m-1} - \alpha_{\text{meta}} \nabla_{W_m} \left[ \sum_{b=1}^{B} \left[ \left\| e_{i_p,t,b} - e_{i_n,t,b} \right\|_2^2 - \left\| e_{i_n,t,b} - e_{i_a,t,b} \right\|_2 + \gamma \right]_+ \right],
\]

(25)

where \(e_{i_p,t,b}, e_{i_n,t,b}\) and \(e_{i_a,t,b}\) are the extracted features of the \(b\)-th anchor, positive, and negative UAV sampled in \(D^Q_m\) using the hard sampling method. This constitutes the second step update. After \(M_0\) episodes, the parameter \(W_{M_0}\) will be used as the meta parameter to initialize the embedding network.
2) **Online Execution of the MC-GASSL:** After initializing the embedding network with the meta parameter \( W_{M_0} \), the MC-GASSL starts to work online in an iterative manner. Specifically, in each iteration, the observers first record the positions and speeds of all UAVs in \( T_0 \) time steps. Note that all the FUAVs follow the IFS whose specific expression is unknown to the observers. The UAVs’ fly independently and all the UAVs adopt the APF to avoid collisions. Then the observers clusters the USNET \( U_{\text{NET}} \) using the SML algorithm with the records as the inputs. After clustering, the observers leverages the GASSL to detect the HUA Vs in each cluster. Here, the records of the UAVs in each cluster make up the online training data for the GASSL algorithm. Note that as the GASSL is a self-supervised learning algorithm, there is no need to provide the ground truth during the online training. Notably, to reduce the time complexity, we leverage the multi-process technique and train the neural networks in the GASSL for all UAVs in each cluster in parallel. Moreover, with the trained IFSN \( \hat{f}(\cdot) \), the MC-GASSL re-clusters the USNET \( U_{\text{NET}} \) to provide the dataset \( D \) for the online training of the embedding network. The re-cluster process is described as follows: each UAV \( i \) selects its HUA \( i_H \) among all other UAVs as the one that can minimize the distance between the predicted velocity calculated by the IFSN \( \hat{f}(\cdot) \) and the true velocity \( v_{i,t} \), i.e.,

\[
i_H \leftarrow \arg \min_{i \neq H} \left\| \hat{f}(p_{i,t}, v_{i,t}, p_{i_H,t}) - v_{i,t} \right\|^2_2.
\] (26)

The USNET \( U_{\text{NET}} \) can be grouped into clusters based on the UAVs that are regarded as the HUAVs of other UAVs. In this way, the dataset \( D \) consisting of the cluster label of each UAV and its historical positions and velocities can be constructed, based on which the embedding network can be trained with the proposed metric learning scheme. The overall algorithm of MC-GASSL is summarized in Algorithm 1.

3) **Time Complexity Analysis:** Recall that the neural network in the GASSL should be applied to each UAV in every cluster, taking the history speeds and positions of all other UAVs from the same cluster as inputs. Let the training episode of the neural network as \( T_G \), and the number of processes in the multi-process be \( N_p \). Then the time complexity of training the neural networks in GASSL for all \( N \) UAVs in the USNET \( U_{\text{NET}} \) is calculated as \( O(T_G N / N_p) \). As for the SML, the time complexity of the inference of the embedding network is proportional to \( T_0 \) and \( N \), and is calculated as \( O(T_0 N) \). The time complexity of training the embedding network is proportional to the training episode, denoted as \( T_S \), as well as \( T_0 \). Due to the hard sampling method, it also proportional to \( N \). Hence, the time complexity of training the embedding network is \( O(T_S T_0 N) \). Overall, the time complexity of the MC-GASSL can be calculated as \( O(T_E(T_G N / N_p + T_S T_0 N)) \), where \( T_E \) denotes the number of iterations. Hence, the MC-GASSL can be executed in an approximate real-time manner.

### IV. Experiments

We conduct simulated experiments to validate the effectiveness of the proposed algorithms. Specifically, we answer the following five questions in our experiments: (1) Can the MC-GASSL effectively detect the HUAVs from the perspective of observers? (2) What is the performance of the GASSL for detecting the HUA in a single cluster? (3) What is the performance of the SML for clustering the USNET \( U_{\text{NET}} \)? (4) Can the meta learning speed up the online training of the embedding network? (5) What is the time complexity of the GASSL and the SML? Can the MC-GASSL execute in a nearly real-time manner?

### A. Experiment Setups

1) **Setups on the USNET \( U_{\text{NET}} \):** In the simulation, the UAVs in the USNET are initially distributed in a 1,000m \( \times \) 1,000m \( \times \) 100m three-dimensional space. The numbers of FUAVs in distinct clusters may be different, but are all not less than 2, i.e., \( m_j \geq 2, \forall j \in M \). The speeds of distinct HUAVs are generated independently and randomly, while FUAVs in

---

### Algorithm 1 MC-GASSL Framework

**Inputs:** The inputs to the IFS.

**Hyper-parameters:** The maximum iteration number \( T_{\text{MAX}} \).

**Outputs:** The estimated HUAVs indexes.

**Offline Meta Learning:**

1: Construct the IFS set \( \mathcal{F} \).
2: Build the support dataset \( D^m_S \) and the query dataset \( D^m_Q \) of the USNET \( U_{\text{NET}} \), \( m \in \{1, 2, ..., M_0\} \).
3: Randomly initialize the parameter of the embedding network in the SML \( W_0 \).
4: for \( m = 1 \) to \( M_0 \) do
5: Sample a batch of anchor, positive, and negative UAVs by hard sampling method.
6: Train one step on parameter \( W_m \) with (24) based on \( D^m_S \). Update \( W_{m-1} \) to \( W_m \) with (25) based on \( D^m_Q \).
end for
7: Obtain the meta parameters \( W_{M_0} \) of the embedding network.

**Initializations:** The embedding network \( W_{M_0} \) of the SML is initialized with the meta parameter \( W_{M_0} \).

**Online Executions:**

1: for \( \tau = 1 \) to \( T_{\text{MAX}} \) do
2: Observe the positions and velocities of all UAVs for \( T_0 \) time steps.
3: **SML:** calculate the embeddings of all UAVs with the embedding network, estimate the number of clusters with gap-statistic method, and cluster the UAVs with K-Means.
4: **GASSL:** obtain the attention weights with self-supervised learning, determine the HUAVs in each cluster based on the attention weights.
5: if all the HUAVs have been detected then
6: Break.
7: end if
8: Construct the dataset \( D \) with (26). Further train the embedding network based on \( D \).
end for
different clusters obey the same IFS \( f(\cdot) \). Let \( T_0 = 4 \). As shown in Table III, we implement 7 different types of the IFS \( f(\cdot) \), including functions of the linear and quadratic combinations of the HUA V’s history speeds \( f_1 \) to \( f_5 \), functions of positions \( f_6 \), and even neural networks \( f_7 \). Note that we let the observers destroy the UAVs once they are detected as the HUA Vs by the observers. If there still remain HUAVs in the USNET that are not destroyed, then the UAVs will move towards the nearest HUAV to form new clusters. In the next detection iteration, the MC-GASSL detects the HUAVs based on the positions and speeds of UAVs in the new USNET.

2) Network Structures: In the neural networks of AGAT, the query matrix \( L(\cdot) \) and similarity function \( s_l(\cdot), \forall l \in \{1, 2, \ldots, T_0 + 1\} \) are MLPs of one and two hidden layers, respectively. The MLP in the IFSN \( \hat{f}(\cdot) \) is composed of three hidden layers, and there is no constraint layer in the IFSN \( \hat{f}(\cdot) \). The hyper-parameter fine-tuning of the neural networks is shown in Supplementary Material IV.B. Overall, the hyper-parameters hardly influence the performance of the proposed algorithms.

3) Performance Index: The performance indexes for the MC-GASSL include the average number of detection iterations, the number of successfully detected HUAVs and the number of detection errors. The average number of detection iterations refers to the number of iterations used for detecting all the HUAVs on average. The number of successfully detected HUAVs should equal to the number of clusters after all the HUAVs are detected. As the SML may not precisely detect the clusters during iterations, some FAUs may be falsely identified as HUAVs by the GASSL. Hence, the number of detection errors is another important performance index for the MC-GASSL. Overall, the smaller average number of detection iterations and the number of detection errors are, the better the performance of the MC-GASSL is.

The performance index for the ablation study on the GASSL is the detection rate of HUAVs in single clusters. The detection rate is defined as the ratio between the number of experiment times \( T_{\text{succ}} \) when the HUAV in the cluster is successfully detected and the total number of experiment times \( T_{\text{exp}} \), i.e.,

\[
\text{detection rate} = \frac{T_{\text{succ}}}{T_{\text{exp}}} \times 100\%.
\]

Table IV lists the configurations on the main devices of the used personal computer.

### B. Main Results

To answer question (1): We construct a USNET consisting of 5 UAV clusters with a total of 82 UAVs. We use the MC-GASSL to detect the HUAVs when the constructed

| Type | Notations | IFS \( f(\cdot) \) | Descriptions |
|------|-----------|--------------------|-------------|
| 1    | \( f_1 \) | \( v_{i,t+1} = \text{norm}(\kappa_0 v_{i,t} + \kappa_n n) \) | proportional to the previous speed \( v_{i,t} \) |
| 2    | \( f_2 \) | \( v_{i,t+1} = \text{norm}\left(\sum_{r=0}^{1} \kappa_r v_{i,t-r} + \kappa_n n\right) \) | linear combination of \( v_{i,t}, v_{i,t-1} \) |
| 3    | \( f_3 \) | \( v_{i,t+1} = \text{norm}\left(\sum_{r=0}^{2} \kappa_r v_{i,t-r} + \kappa_n n\right) \) | linear combination of \( v_{i,t}, v_{i,t-1}, v_{i,t-2} \) |
| 4    | \( f_4 \) | \( v_{i,t+1} = \text{norm}\left(\sum_{r=0}^{1} \kappa_r v_{i,t-r} \otimes v_{i,t-r} + \kappa_n n\right) \) | quadratic combination of \( v_{i,t}, v_{i,t-1} \) |
| 5    | \( f_5 \) | \( v_{i,t+1} = \text{norm}\left(\sum_{r=0}^{2} \kappa_r v_{i,t-r} \otimes v_{i,t-r} + \kappa_n n\right) \) | quadratic combination of \( v_{i,t}, v_{i,t-1}, v_{i,t-2} \) |
| 6    | \( f_6 \) | \( v_{i,t+1} = \text{norm}\left(\kappa_0 v_{i,t} + \kappa_n n\right) \) | following speed \( v_{i,t} \), keep in range \( \kappa_r \) with HUAV |
| 7    | \( f_7 \) | \( v_{i,t+1} = \text{MLP}(P_{i,t}, v_{i,t}, p_{i,t-3}) \) | fully connected neural network with inputs \( P_{i,t}, v_{i,t}, p_{i,t-3} \) |

\( \kappa_0 = 1, \kappa_1 = 1, \kappa_2 = 1, \kappa_3 = 1, \kappa_5 = 1, \kappa_6 = 0.05, \kappa_n = 60, \text{norm}(\cdot) = \frac{\|\cdot\|_2}{\|\cdot\|_2} \),

\( n \in \mathbb{R}^3 \) represents noise, and \( n \sim \mathcal{N}(0, I) \).
TABLE IV
THE DETECTION RATE OF HUA Vs IN SINGLE CLUSTERS WITH THE GASSL

| Type | IFS | 2   | 3   | 5   | 10  | 15  | 20  | 25  | 30  | 40  | 50  |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1    | $f_1$ | 98.1% | 100.0% | 100.0% | 99.8% | 99.4% | 99.3% | 100.0% | 98.9% | 99.9% |
| 2    | $f_2$ | 99.8% | 100.0% | 100.0% | 100.0% | 98.3% | 99.2% | 99.1% | 100.0% | 100.0% |
| 3    | $f_3$ | 99.9% | 100.0% | 100.0% | 100.0% | 100.0% | 99.9% | 98.9% | 100.0% | 100.0% |
| 4    | $f_4$ | 92.0% | 93.4% | 99.2% | 91.4% | 98.3% | 98.5% | 99.1% | 99.5% | 94.2% | 97.5% |
| 5    | $f_5$ | 98.4% | 91.4% | 98.5% | 99.1% | 99.3% | 95.4% | 98.2% | 100.0% | 98.1% | 99.3% |
| 6    | $f_6$ | 98.2% | 100.0% | 99.9% | 98.5% | 99.1% | 99.2% | 99.9% | 100.0% | 98.9% | 99.5% |
| 7    | $f_7$ | 99.9% | 98.7% | 100.0% | 99.8% | 100.0% | 99.2% | 99.4% | 98.3% | 100.0% | 99.2% |

The detection rate is calculated as the ratio between the number of successfully detected HUA Vs and the number of experiment times.

TABLE V
HARDWARE RESOURCES USED TO CARRY OUT THE PROPOSED ALGORITHMS

| Devices | Configurations |
|---------|----------------|
| CPU     | model name: Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz |
|         | physical cores: 2 |
|         | cpu cores: 16 |
|         | logic cores: 64 |
|         | RAM: 376GiB |
| GPU     | model name: NVIDIA GeForce RTX 3090 |
|         | memory: 24576MiB |

USNET obeys IFS $f_1$ to $f_7$, and the detection results are shown in Fig. 5(a). We can see that the MC-GASSL can detect all 5 HUA Vs with an average of 3 to 4 detection iterations regardless of the types of the obeyed IFS. The number of successful detected HUA Vs is 5 in all seven cases, where the detection redundancy is about $\frac{5}{3} = 6.1\% \ll 1$. Hence, the MC-GASSL is effective in detecting cluster heads when the USNET obeys different IFSs. In addition, we construct USNETs with $M = 2, 3, 5, 8, 10$ clusters that obey the same IFS $f = f_1$ and has a total of $N = 82$ UAVs. We detect the cluster heads of the USNET with MC-GASSL, and the detection results are shown in Fig. 5(b). We can see that the number of average detection rounds as well as the number of detection redundancy increase with the number of UAV clusters. Nonetheless, the MC-GASSL can detect all the HUA Vs successfully within an average of $R = 7$ detection rounds, and the average detection redundancy is smaller than $\frac{15.2}{82} = 22.2\%$. Hence, the MC-GASSL is effective in detecting cluster heads of USNETs with various UAV clusters. Moreover, we construct USNETs with different number of clusters $M = 5$, obeying the same type of IFS $f = f_1$, but with different number of UAVs. We utilize the MC-GASSL to detect the cluster heads of USNETs, and the detection results are shown in Fig. 5(c). We can see that the MC-GASSL can detect all 5 HUA Vs after an average of 3 to 4 detection rounds regardless of the total number of UAVs. The number of successful detected HUA Vs is 5, while the number of redundant HUA Vs is about 5 in all cases. Hence, the MC-GASSL is effective in detecting the cluster heads of USNETs with various number of total UAVs. To visualize, we take the case where $f = f_1$ and $M = 8$ as an example. The total number of UAVs in the USNET is 185. Fig. 6 shows the cluster head detection process with MC-GASSL, where Fig. 6(a) represents the ground truth of the USNET, Fig. 6(b) and 6(c) shows the attention weights and detected HUA Vs during the first and second detection round, respectively, and Fig. 7 shows the detection rates during the whole detection process. We can see that the MC-GASSL in the first round detects 8 UAVs, of which 6 UAVs are correct HUA Vs and 2 UAVs are false detections. The MC-GASSL finds all the remaining HUA Vs in the second round. The total consumption of UAVs is about 5.40%, where the successful detection rate is 4.32% and redundant detection rate is 1.08%.

C. Ablation Study

1) Performance of the GASSL: To answer question (2): We simulate with all seven types of the IFS $f(\cdot)$ under different number of UAVs 1,000 times each\(^4\). The detection rate of the GASSL are shown in Table IV. We can see that the detection rate achieves 100.00\% in many cases, and the average detection rate is 98.2\%. Hence, GASSL can effectively detect the HUA Vs in single clusters. Take the case where the IFS is $f_1$ and the number of UAVs is 15 as an example\(^5\). Note that the real index of HUA Vs is $i_L = 7$. Fig. 4 shows the heat map of the attentions between UAVs when using the GASSL, where the $i$-th row in Fig. 4(a) displays the attention weights of UAV $i$ with respect to all UAVs, and the $i$-th row in Fig. 4(b)–4(f) displays the attention values of UAV $i$ to all UAVs in the attention head 1 to 5, respectively. We can see that UAVs are paying the most attention to UAV 7 in all the attention heads except head 5. From

\(^4\)The loss function of the GASSL is shown in Supplementary Material IVC.

\(^5\)More examples of the attention weights in different cases are represented in Supplementary Material IV.A.
Fig. 4. Heat map of attentions weights and attention values of each attention head. The real index of HUA\( V \) is \( i_L = 7 \), while the detected HUA\( \hat{V}_1 = \{7\} \).

Fig. 5. The detection results with MC-GASSL.

Fig. 6. The detection process of the USNET with MC-GASSL.
the attention weights, we can determine the detected HUA V as UAV 7, which is consistent with the real index of HUA V $i_L = 7$. Fig. 8 indicates the attention relationship between UAVs, where the thickness of the line reflects the magnitude of the attention weight between two UAVs, and UAV 7 has the most great attentions from all UAVs in the USNET. Fig. 9 shows the hierarchy structure of the cluster, where UAV 7 acts as the HUA V.

2) Performance of the SML: To answer question (3): We construct USNETs obeying IFSs $f_1$ to $f_7$, respectively, and compare the clustering performance between the SML and other traditional clustering algorithms, including K-Means, symmetric non-negative matrix factorization (SNMF) [31] and spectral clustering [32]. Fig. 10(a) shows the clustering purities of these four algorithms. We can see that the average clustering purities of SML exceed those of other algorithms in all seven cases, which indicates the effectiveness of the SML in clustering the USNET. To visualize the learned embedding space, Fig. 10 shows an example of the UAVs’ feature changing during the metric learning process in the SML, where the USNET has $M = 5$ clusters with a total of $N = 237$ UAVs, obeying the IFS $f = f_1$. As shown in Fig. 10(b), the initial embeddings of UAVs in different clusters are mixed together and can hardly be clustered with traditional cluster methods, such as K-Means. Nonetheless, as the metric learning processes, embeddings of UAVs in distinct clusters move away from each other, while the embeddings of UAVs in the same clusters gradually converge together, as displayed in Fig. 10(c). Hence, the SML algorithm can map the historical positions and velocities of UAVs to a proper feature space, where features of UAVs from the same cluster stay close to each other and features of UAVs from distinct clusters stay away from each other.

D. Time Complexity Results

To answer question (4): Fig. 11 shows the loss function during the offline meta learning process. We can see that the loss function decreases with the learning episodes and converge to the value about 0.125. This indicates that the meta learning makes the GRU network cluster the USNET with various type of IFS within small metric learning loss. Fig. 12 shows the loss function of the online metric learning process using the GRU network initialized with meta parameters, pre-trained parameters [30] and random parameters, respectively. We can see that the starting point of the loss function with meta parameters is much smaller than that with pre-trained parameters and random parameters. The loss functions of all three kinds of parameters drops as the metric learning processes. Nonetheless, the loss function of meta parameters is always smaller than the other two loss functions and converges to a lower value. Hence, the offline meta learning can help the GRU network find better initialized parameters and improve the performance of online metric learning.

To answer question (5): We provide the real execution time of the proposed GASSL and SML algorithms, as well as the whole MC-GASSL framework in the response and the revised manuscript. Specifically, we provide the online training time of the neural network in the GASSL applied to each UAV in a cluster, as shown in Fig. 13 in the response. We can see that the average online training time is only about 2 seconds when the number of UAVs in the cluster is 3 and no more than 7 seconds when the number of UAVs in the cluster is 36. In addition, we let the number of process in the multi-process technique be $N_p = 12$ and provide the execution times of the GASSL under different number of UAVs, as shown in Fig. 14 in the response. We can see that the execution time is proportional to the number of UAVs, which is consistent with the time complexity expression of the GASSL, $O(T_G N_p/N_p)$. We can also see that the execution times of the GASSL using multi-process technique are much smaller than that with the single-process approach, especially as the number of UAVs increases.
Fig. 10. SML algorithm simulation results.
Fig. 11. Loss function during meta learning.
Fig. 12. Loss function of online metric learning.
Fig. 13. Online training time of the GASSL.
Fig. 14. Execution time of the GASSL.
Fig. 15. Inference time of the embedding network in the SML algorithm.
Fig. 16. Average time of stages in an iteration of the MC-GASSL.
As for the SML, we provide the inference times of the embedding network under different $T_0$, as shown in Fig. 15 in the response. We can see that the inference time of the embedding network for one UAV is only in the order of milliseconds ($10^{-3}$ seconds). Moreover, we conduct the training of the embedding network with different datasets 100 times, and the average time of training the embedding network is about 80 seconds. We test the MC-GASSL with a USNET of $M=5$ clusters and $N=82$ UAVs and summarize the average execution time of each algorithm during one iteration of the MC-GASSL in Fig. 16 in the response. We can see that the total time needed for each iteration is 182.39 seconds (about 3.04 minutes) on average. Note that the training process of the embedding network in the last iteration is not needed. The total execution time of the MC-GASSL also relies on the total number of iterations. The average of the total execution time of the MC-GASSL can be calculated as:

$$\text{average total execution time} = \text{average time per iteration} \times \text{number of iterations}.$$  

where the reason of subtracting the time of the last two stages, i.e., the “re-clustering & build the dataset” stage (7.87s) and the “SML: training the embedding network” stage (42.51s), is that we do not need to execute these two stages in the last iteration of the MC-GASSL since all the HUA Vs are detected in the “GASSL” stage in the last iteration. We also note that the number of iterations is usually not large. We conduct the experiment of testing the MC-GASSL algorithm ten times, and the average number of iterations is 3. The results are shown in Table VI. As an example, we next show the real total execution time of the MC-GASSL applied to the USNET in Fig. 6(a), where the MC-GASSL detects all the HUA Vs within two iterations. As shown in Fig. 17, the real execution time is 289.26 seconds. This is acceptable to the observers, e.g., the observers in the confrontation scenario, since a confrontation can often last for tens of minutes or even hours. Hence, the MC-GASSL executions in an approximate real-time manner.

V. CONCLUSION

In this paper, we study the cluster head detection problem of a two-level USNET with multiple clusters from the perspective of the observers, who are not aware of any internal information on the USNET, including the IFS. We propose the MC-GASSL framework for the observers. The MC-GASSL works in an iterative manner, alternating between detecting the clusters with the SML and detecting the HUA Vs within each single cluster with the GASSL. Numerical results show that the MC-GASSL can efficiently detect all the HUA Vs in the USNET under various IFSs and cluster numbers with low detection redundancies, which validates the effectiveness of it. In addition, the ablation studies on the GASSL show that the GASSL can detect the HUA Vs in single clusters under various kinds of IFSs with over 98% accuracy on average. Moreover, the ablation studies on the SML show that the clustering purity of the USNET with MC-GASSL exceeds that with traditional clustering algorithms by at least 10% on average.

TABLE VI
THE NUMBER OF ITERATIONS OF THE MC-GASSL IN TEN EXPERIMENTS

| Number of Iterations | 1  | 2  | 3  | 4  | 5  | 6  |
|---------------------|----|----|----|----|----|----|
| Times               | 0  | 5  | 3  | 0  | 1  | 1  |

REFERENCES

[1] H. Wang, H. Zhao, J. Zhang, D. Ma, J. Li and J. Wei, “Survey on unmanned aerial vehicle networks: A cyber physical system perspective,” IEEE Commun. Surv. Tut., vol. 22, no. 2, pp. 1027–1070, 2nd Quart. 2019.
[2] Y. Zhang, Z. Mou, F. Gao, L. Xing, J. Jiang and Z. Han, “Hierarchical deep reinforcement learning for backscattering data collection with multiple UAVs,” IEEE Internet Things J., vol. 8, no. 5, pp. 3786–3800, Mar. 2021.
[3] X. Xu, H. Zhao, H. Yao and S. Wang, “A blockchain-enabled energy-efficient data collection system for UAV-assisted IoT,” IEEE Internet Things J., vol. 8, no. 4, pp. 2431–2443, Feb. 2021.
[4] Z. Mou, Y. Zhang, F. Gao, H. Wang, T. Zhang and Z. Han, “Deep reinforcement learning based three-dimensional area coverage with UAV swarm,” IEEE J. Sel. Areas Commun., vol. 39, no. 10, pp. 3160–3176, Oct. 2021.
H. Li, B. Zhang, S. Qin, and J. Peng, "UA V-Clustering: Cluster head selection," IEEE Open J. Commun. Soc., vol. 2, pp. 1298–1306, 2021.

A. Madrano, A. Al-Kaff, and D. Martin, "3D trajectory planning method for UAVs in building emergencies," Sensors, vol. 20, no. 3, pp. 1–20, Jan. 2020, Art. no. 642.

H. Teng, I. Ahmad, A. Samn, and K. Chang, "3D optimal surveillance trajectory planning for multiple UAVs by using particle swarm optimization with surveillance area priority," IEEE Access, vol. 8, pp. 86516–86527, 2020.

Z. Mou, F. Gao, J. Liu and Q. Wu, "Resilient UAV swarm communications with graph convolutional neural network," IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 393–411, Jan. 2022.

F. Xiong, A. Li, H. Wang, and L. Tang, "An SDN-MQTT based communication system for battlefield UAV swarms," IEEE Commun. Mag., vol. 57, no. 8, pp. 41–47, Aug. 2019.

Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," IEEE Trans. Veh. Technol., vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

C. Zhan, Y. Zeng and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," IEEE Wireless Commun. Lett., vol. 7, no. 3, pp. 518–521, June 2018.

V. Hassan, V. Chamola, D. N. G. Krishna, and M. Guizani, "A distributed framework for energy trading between UAVs and charging stations for critical applications," IEEE Trans. Veh. Technol., vol. 69, no. 5, pp. 5391–5402, May 2020.

K. Wang, X. Zhang, L. Duan and J. Tie, "Multi-UAV cooperative trajectory for servicing dynamic demands and charging battery," IEEE Trans. Mob. Comput., vol. 22, no. 3, pp. 1599–1614, Mar. 2023.

B. Liu and J. Sun, "Stackelberg game under asymmetric information in unmanned aerial vehicle swarm active defense defense: From a multi-layer network perspective," in Proc. Int. Conf. Big Data Intell. Decis. Making, Guilin, China, Jul. 2021, pp. 75–79.

H. Li, B. Zhang, S. Qin, and J. Peng, "UAV-Clustering: Cluster head selection and update for UAV swarms searching with unknown target location," in Proc. IEEE 25th Int. Symp. World Wireless., Mobile Multim. Netw. Services Res. (WoWMoM), Belfast, UK. 2022, pp. 483–488.

C. Xu, K. Zhang, Y. Jiang, S. Niu, T. Yang, and H. Song, "Communications aware UAV swarm surveillance based on hierarchical architecture," Drones, vol. 5, no. 2, pp. 1–26, Apr. 2021.

D. Xing, Z. Zhen, H. Gong, "Offense-defense confrontation decision making for dynamic UAV swarm versus UAV swarm," Proc. Inst. Mech. Eng. Part G: J. Aerosp. Eng., vol. 233, no. 5, pp. 5689–5702, Jun. 2019.

F. Liu et al., “Deep learning for community detection: progress, challenges and opportunities,” in Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI), Yokohama, Japan, Jan. 2021, pp. 4981–4987.

A. Tahir, J. Böling, M. H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of unmanned aerial vehicles: A survey,” J. Ind. Inf. Integr., vol. 16, no. 100106, pp. 1–7, Dec. 2019.

J. Chen et al., “Joint task assignment and spectrum allocation in heterogeneous UAV communication networks: A coalition formation game-theoretic approach,” IEEE Trans. Wireless Commun., vol. 20, no.1, pp. 440–452, Jan. 2021.

C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, Anchorange, AK, USA, Jul. 2019, pp. 793–803.

P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in Proc. 6th Int. Conf. Learn. Represent. (ICLR), Vancouver Convention Center, Vancouver, QC, Canada, Feb. 2018, pp. 1–12.

K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in Proc. Conf. Empiricial Methods Nat. Lang. Process. (EMNLP), Doha, Qatar, Oct. 2014, pp. 1724–1734.

K. Krishna and M. Narasimha Murty, “Genetic K-means algorithm,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 29, no. 3, pp. 433–439, Jun. 1999.

J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, “Deep metric learning with angular loss,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, Hawaii, USA, Jul. 2017, pp. 2593–2601.

F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 815–823.

W. Ge, “Deep metric learning with hierarchical triplet loss,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 269–285.

C. Finn, R. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Proc. 34th Int. Conf. Mach. Learn. (ICML), International Convention Centre, Sydney, NSW, Australia, Jul. 2017, pp. 1126–1135.

R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” J. Roy. Statist. Soc., vol. 63, no. 2, pp. 411–423, Jan. 2002.

A. Raghu, J. Lorraine, S. Kornblith, M. McDermott, and D. K. Duvenaud, “Meta-learning to improve pre-training,” in Proc. 35th Conf. Neural Inf. Process. Syst., virtual only, May 2021, pp. 1–31.

Y. Pei, N. Chakrabort, and K. Sycara, “Nonnegative matrix tri-factorization with graph regularization for community detection in social networks,” in Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI), Buenos Aires, Argentina, Jun. 2015, pp. 2083–2089.

U. Von Luxburg, “A tutorial on spectral clustering,” Statist. Comput., vol. 17, no. 4, pp. 395–416, Aug. 2007.

J. Sun, J. Tang, and S. Lao, “Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm,” IEEE Access, vol. 5, pp. 18382–18390, 2017.

I. Chkrebtii, D. Riorian, and S. Sampalli, “Cluster-head election using fuzzy logic for wireless sensor networks,” in Proc. 3rd Ann. Commun. Netw. Services Res. Conf., Piscataway, NJ, USA: IEEE Press, May 2005, pp. 255–260.

R. Gupta, “Cluster head election in wireless sensor network: A comprehensive study and future directions,” Int. J. Comput. Netw. Appl., vol. 7, no. 6, pp. 178–192, Dec. 2020.

S. Soro, and W. B. Heinzelman, “Cluster head election techniques for coverage preservation in wireless sensor networks,” Ad Hoc Netw., vol. 7, no. 5, pp. 955–972, 2009.

E. T. Tsipourou, S. T. Paruchuri and J. S. Baras, “Interest, energy and physical-aware coalition formation and resource allocation in smart IoT applications,” in Proc. 51th Ann. Conf. Inform. Sci. Syst., Baltimore, MD, USA, Mar. 2017, pp. 1–6.

Zhiyu Mou received the B.Eng. degree in automation from Beijing Institute of Technology, China, in 2020, and the M.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2023. His research interests include swarm intelligent communications, deep reinforcement learning, optimizations, and graph neural networks.

Feifei Gao (Fellow, IEEE) received the B.Eng. degree from Xi’an Jiaotong University, Xi’an, China in 2002, the M.Sc. degree from McMaster University, Hamilton, ON, Canada, in 2004, and the Ph.D. degree from the National University of Singapore, Singapore, in 2007. Since 2011, he joined the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. His research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence assisted communications. He has authored/co-authored more than 150 refereed IEEE journal papers and more than 150 IEEE conference proceeding papers that are cited more than 11,000 times in Google Scholar. He has served as an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (Lead Guest Editor), IEEE TRANSACTIONS ON COMMUNICATIONS AND NETWORKING, IEEE SIGNAL PROCESSING LETTERS (Senior Editor), IEEE COMMUNICATIONS LETTERS (Senior Editor), IEEE WIRELESS COMMUNICATIONS LETTERS, and CHINA COMMUNICATIONS. He has also served as the symposium co-chair for 2019 IEEE Conference on Communications (ICC), 2018 IEEE Vehicular Technology Conference Spring (VTC), 2015 IEEE Conference on Communications (ICC), 2014 IEEE Global Communications Conference (GLOBECOM), 2014 IEEE Vehicular Technology Conference Fall (VTC), as well as Technical Committee Members for more than 50 IEEE conferences.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Jun Liu (Member, IEEE) received the Ph.D. degree from the Northeastern University, Shenyang, China, in 2011. She worked as a Lecturer with the College of Information Science and Engineering from 2011 to 2016 with the Northeastern University. She worked as a Postdoctoral Research Fellow from 2016 to 2018 with the CECA Department, Peking University. She is currently an Assistant Professor with the Institute of Network Sciences and Cyberspace, Tsinghua University, and a Researcher with Beijing National Research Center for Information Science and Technology, Beijing, China. Her research interests include wireless networks and space-air-terrestrial networks.

Xiang Yun received the B.Eng. and M.Eng. degrees from Beijing University of Posts and Telecommunications, in 2006 and 2009, respectively. Since 2015, he joined with Baicells Technologies Company, Ltd., and currently he is the Technical and Innovation Director and Head of Standard. His research interests include end to end system architecture of B5G and 6G system especially on NTN (Non-Terrestrial Network) and artificial intelligence embedded communications. He has served as the 3GPP work item rapporteur and MulteFire Alliance SON Work Group Chairman. He has published more than 20 IEEE conference papers and granted more than 80 patents globally.

Qihui Wu received the B.S. degree in communications engineering, and the M.S. and Ph.D. degrees in communications and information systems from the Institute of Communications Engineering, Nanjing, China, in 1994, 1997, and 2000, respectively. After graduation, he worked with the PLA University of Science and Technology, Nanjing. Since 2016, he has been with Nanjing University of Aeronautics and Astronautics (NUAA) and appointed as Changjiang Distinguished Professorship. Currently, he is the Vice-Principal of NUAA. His academic contributions of cognitive radio have been demonstrated in over 200 publications with more than 4000 citations, where over 10 papers are honored as the ESI highly cited paper. Furthermore, he has been invited to present keynotes and has been awarded a number of distinctions, such as the IEEE Signal Processing Society’s 2015 Young Author Best Paper Award, the 14th IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award, the First Prize of the Natural Science Award by China Institute of Electronics/Communications in 2020 and 2017, respectively, etc. Moreover, the broad impacts of his research have been widely publicized with more than 30+ invited talks in various international conferences. He has acted as a TPC and a General Chair of the international conferences, such as IEEE VTC, IEEE ICC, IEEE WCSP, and took part in a wide range of IEEE activities. He is currently directing the Key Lab of Ministry of Industry and Information Technology of China, working on more than 10 major research projects in the field of cognitive radio network, intelligent space control of electromagnetic spectrum and massive UAV cluster, sponsored by such as NFSC. His innovations have been authorized by more than 20 national and international patents and applied in such as Beidou Satellite and lunar exploration programs. He is an IET Fellow.