mathematics

MDPI

Article

A Dual Fusion Pipeline to Discover Tactical Knowledge Guided by Implicit Graph Representation Learning

Xiaodong Wang [1], Pei He [1], Hongjing Yao [2], Xiangnan Shi [3], Jiwei Wang [1] and Yangming Guo [2,*]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; wangxiaodong@nwpu.edu.cn (X.W.); hepei002@mail.nwpu.edu.cn (P.H.); jiweiwang@mail.nwpu.edu.cn (J.W.)
[2] School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China; hjyao@mail.nwpu.edu.cn
[3] Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China; shixiangnan@mail.nwpu.edu.cn
* Correspondence: yangming_g@nwpu.edu.cn

Abstract: Discovering tactical knowledge aims to extract tactical data derived from battlefield signal data, which is vital in information warfare. The learning and reasoning from battlefield signal information can help commanders make effective decisions. However, traditional methods are limited in capturing sequential and global representation due to their reliance on prior knowledge or feature engineering. The current models based on deep learning focus on extracting implicit behavioral characteristics from combat process data, overlooking the embedded martial knowledge within the recognition of combat intentions. In this work, we fill the above challenge by proposing a dual fusion pipeline introducing graph representation learning into sequence learning to construct tactical behavior sequence graphs expressing implicit martial knowledge, named TBGCN. Specifically, the TBGCN utilizes graph representation learning to represent prior knowledge by building a graph to induce deep learning paradigms, and sequence learning finds the hidden representation from the target's serialized data. Then, we employ a fusion module to merge two such representations. The significance of integrating graphs with deep learning lies in using the artificial experience of implicit graph structure guiding adaptive learning, which can improve representation ability and model generalization. Extensive experimental results demonstrate that the proposed TBGCN can effectively discover tactical knowledge and significantly outperform the traditional and deep learning methods.

Keywords: graph representation learning; graph structure; discovering tactical knowledge; deep learning

MSC: 68T07

check for updates

Citation: Wang, X.; He, P.; Yao, H.; Shi, X.; Wang, J.; Guo, Y. A Dual Fusion Pipeline to Discover Tactical Knowledge Guided by Implicit Graph Representation Learning. *Mathematics* **2024**, *12*, 528. https://doi.org/10.3390/math12040528

Academic Editor: Ke-Lin Du

Received: 19 December 2023
Revised: 31 January 2024
Accepted: 31 January 2024
Published: 8 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Discovering tactical knowledge refers to extracting the representation from battlefield signal data to achieve accurate and efficient recognition of enemy intent and assist commanders in perceiving the tactical behavior. In information warfare, the amount of data is produced by the renewal of platforms and the development of sensors. How to intelligently and efficiently discover tactical patterns from these data is a non-trivial task in equipment construction.

Information warfare involves data with massive growth, complex types, and a high degree of coupling, which put the requirements for the generalization and robustness of the intelligent methods in the learning and concluding the pattern of combat [1]. Traditional methods are dominated by rule statistics and artificial neural networks. The former lacks generalization and can only identify pre-defined knowledge and patterns. The latter lacks robustness and cannot deal sufficiently with noise signals contained in

different modes [2]. With the rapid development of deep learning, methods of adaptive learning from massive data have promoted and expanded the generalization, making this efficient and convenient end-to-end learning into the mainstream of discovering tactical knowledge. Essentially, the process with the input signal and supervision signal adaptively adjusts parameters of the neural network [3] by gradient descent to fit the function [4]. Although the deep learning methods have advantages in feature extraction and data fitting [5], these methods can only learn the data distribution related to the training dataset, which only considers the data distribution without prior knowledge guiding adaptive learning. However, battlefield mission data are produced by the target's behavior sequence, which is an implicit graph structure knowledge. Further, both traditional methods and deep learning methods are single pipelines, whose ability for presentation is monotonous and have significant limitations with model generalization in this scenario.

In practice, tactical intentions stem from accomplishing tactical, operational tasks, defined as an ordered collection of interrelated combat actions undertaken by combat units to fulfill responsibilities or achieve specific operational objectives within specified battlefield conditions and spatiotemporal constraints. Our side lacks direct access to the adversary's overarching goals or knowledge of their respective sub-goals throughout combat. Our understanding is limited to continuous states detected from adversary entities, forming their behavioral sequences. Consequently, we engage in the reverse inference of the adversary's planning process based on this premise to discern their intentions. Therefore, the representation of combat intentions inherently embodies a tactical relationship. However, the current model predominantly focuses on extracting implicit behavioral features from combat process data, neglecting the embedded martial knowledge within the context of combat intention recognition. Consequently, both traditional methods and deep learning methods are focused on capturing only localized information and failing to portray the tactical martial knowledge implicit within combat intentions adequately, whose ability for presentation is monotonous and has significant limitations with model generalization in this scenario.

In this work, we fill the above gap by proposing a Tactical Behavior Graph Convolutional Network (TBGCN), which integrates graph representation learning and sequence learning to discover tactical knowledge. The TBGCN constructs a graph topology of the behavior sequence data as prior knowledge for guiding adaptive supervised learning and learns the serialized information with hidden representation. The significance of integrating graph topology structure as prior knowledge with deep learning lies in: (1) using artificial experiences to guide adaptive learning, which can improve the ability of representation; and (2) utilizing graph topology as prior knowledge, which can incorporate additional information to capture the rich relations among the behavior sequence. To the best of our knowledge, this is the first time analyzing graph representation learning to construct a graph topology as prior knowledge in guiding sequence learning in this task.

The contributions of this work can be summarized as follows:

- We build the graph topology as prior knowledge involving nodes and edges with features to represent the implicit graph structure of the target's behavior sequence from the battlefield signal data and guide supervised learning for accurate recognition of enemy intent.
- We propose a TBGCN, which utilizes graph representation learning to guide sequence learning. This method effectively combines artificial experience and the adaptive model, which can partially improve the representation ability and generalization of the model.
- We conduct comprehensive experiments on the tactical datasets and the results show the apparent superiority of our proposed pipeline.

The rest of the paper is organized as follows: Section 2 reviews the related works. Section 3 defines the problem. Section 4 presents the TBGCN model. Section 5 introduces the experimental datasets, evaluation metrics, implementation details, and presents the comparative and analysis experiments. Section 6 gives a conclusion.

2. Related Works

This section introduces the current methods of discovering tactical knowledge based on traditional ones and deep learning ones.

2.1. Traditional Methods

Traditional tactical knowledge discovery methods include Bayesian network (BN), artificial neural network (ANN), etc. Bayesian has reasoning and analytical abilities [6]. It combines graph and probability theory to construct a network and infer knowledge [7]. For example, Xu et al. [8] proposed a BN to evaluate and optimize the effectiveness of conventional missile anti-ship combat systems. Dahlbom et al. [9] compared DBN and fuzzy logic method for detecting hostile aircraft behaviors. Artificial neural networks can imitate the structure of the human brain, approximately mapping between the input signal and supervision signal through a backpropagation algorithm [10]. Variants of the method include a self-organizing map (SOM) [11], restricted Boltzmann machine (RBM) [12], deep trust network (DBN) [13], etc. Liu et al. [14] proposed a Similarity approach using Fuzzy Min–Max Neural Network (SAIRF) to intelligent classification tasks recognizing attack intentions. Although traditional methods can enlighten the reasoning process for combat rules or adaptively learning rule patterns, these methods need to extract the features that are highly relevant to the mission by feature engineering. That has significant limitations in solving our problem, i.e., massive original data.

2.2. Deep Learning-Based Methods

Deep learning evolves from the artificial neural networks and is successfully applied in many domains, including visual processing and analysis [15], natural language processing [16,17], time series analysis [18,19], etc. Deep learning models can adaptively extract features from the original data, which is utilized for target detection in the military field, finding the action track of the target, etc. For example, Yin et al. [20] proposed a deep learning method to detect the motion trajectory of sea ships. Xue et al. [21] presented a novel deep learning method named Panoramic Convolutional Long Short-Term Memory networks (PCLSTM) for recognizing the aerial target's intention. Zhang et al. [22] adopted deep learning to assess the threat degree of enemy targets. Zhang et al. [23] proposed a 1DCNN-BiLSTM hybrid neural network for recognizing air target combat intention. Zhou et al. [24] combined the advantages of the long short-term memory (LSTM) networks and decision tree developing an intention prediction method. Xia et al. [25] utilized the gated recurrent unit (GRU) network supplemented by the highest frequency method (HFM) to realize enemy intention recognition. However, deep learning only adaptively learns the expression from the original data, ignoring the guidance of prior knowledge to the training. Therefore, deep learning-based approaches in the military with a low fault tolerance rate have significant limitations.

The details of the above works are summarized in Table 1. Inspired by these above efforts, our work solves these challenges by introducing the prior knowledge into the neural network and utilizing the variant of graph neural network to learn the implicit topology. Taking the topology as the prior knowledge for a better representation to enhance robustness and generalization of the model.

Table 1. Related works.

ID	Year	Method	Author	Description
1	2020	Traditional Method	Xu et al. [8]	Proposes a BN to evaluate and optimize the effectiveness of conventional missile anti-ship combat systems
2	2013		Zhou et al. [9]	Compared DBN and fuzzy logic method for detecting hostile aircraft behaviors
3	2018		Ahmed et al. [14]	Proposed a Similarity approach using Fuzzy Min–Max Neural Network (SAIRF) to intelligent classification tasks recognizing attack intention

Table 1. *Cont.*

ID	Year	Method	Author	Description
4	2020		Xu et al. [26]	Optimized the intention recognition by dynamic sequence Bayesian network
5	2021		Yin et al. [20]	Proposes a deep learning method to detect the motion trajectory of sea ships
6	2020		Xue et al. [21]	Presented Panoramic Convolutional Long Short-Term Memory networks (PCLSTM) for recognizing the aerial target's intention
7	2021		Zhang et al. [22]	Adopts deep learning to assess the threat degree of enemy targets
8	2023		Zhang et al. [23]	Proposed 1DCNN-BiLSTM hybrid neural network for recognizing air target combat intention
9	2020		Zhou et al. [24]	Combined the advantages of the long short-term memory (LSTM) networks and decision tree developing an intention prediction method
10	2021	Deep Learning	Teng et al. [27]	Designs a new deep learning method attention mechanism with temporal convolutional network and bidirectional gated recurrent unit (Attention-TCN-BiGRU) to improve the recognition of the combat intent of air targets
11	2023		Xia et al. [25]	Proposes a novel intention recognition method to realize enemy intention recognition in an uncertain and incomplete air combat information environment
12	2022		Wang et al. [28]	Proposes a recognition model to identify tactical intention of aerial target based on a multi-sense-scaled attention architecture
13	2022		Yang et al. [29]	Proposes an online hierarchical recognition method for target tactical intention in BVR air combat based on cascaded support vector machine (CSVM)
14	2022		Wang et al. [30]	Establishes a quick intention identification model based on hybrid neural network
15	2022		Wang et al. [31]	Proposes a target tactical intention recognition algorithm based on bi-directional Long Short-Term Memory (BiLSTM)

3. Problem Definition

The input for this task entails continuous time-series data obtained from sensors. These data comprises various features characterizing the combat entity's behavior over a specific duration, to elucidate the entity's combat intent within that temporal context. The neural network discovers the target's combat intent from the provided time-series data. This process involves leveraging sophisticated neural network architectures to effectively capture and distill meaningful patterns and representations from the dynamic and complex time-series information provided by the sensors.

Assume a given set of sequential data representing the input data for target_i, denoted as $X_i = [\vec{x}_1^i, \vec{x}_2^i, \ldots, \vec{x}_t^i] \in \mathbb{R}^{N \times W}$, where $\mathbb{R}$ represents the real number space, and N and W is the sequence length and feature dimension, respectively. The problem we address can be formalized as follows:

$$\begin{aligned} [\vec{r}_1^i, \vec{r}_2^i, \ldots, \vec{r}_t^i] &= f_1([\vec{x}_1^i, \vec{x}_2^i, \ldots, \vec{x}_t^i]), \\ y_i &= f_2([\vec{r}_1^i, \vec{r}_2^i, \ldots, \vec{r}_t^i]), \end{aligned} \tag{1}$$

where $[\vec{r}_1^i, \vec{r}_2^i, \ldots, \vec{r}_t^i]$ is the feature representation from the input sequential data, f_1 denotes the designed model, essentially a mapping function, f_2 is a classifier, and y_i represents the intent of the target i. This formulation encapsulates the process of feature aggregation and classification in the context of combat scenario data.

This study object is fighter jets, specifically choosing conventional missions related to fighter jet airstrikes as the operational intent space, as illustrated in Table 2, for experimental analysis. From numerous features characterizing the combat entities, we have selected, based on domain expertise, those features highly relevant to influencing combat intent. The specific features are detailed in Table 3. In this context, the term "operational intent space" refers to a conceptual framework that encapsulates the various tasks associated with fighter

jet missions. The selection of features is guided by the insights of domain experts, ensuring that the chosen characteristics are particularly pertinent to understanding and assessing combat intent.

Table 2. The intent space.

Label	**0**	**1**	**2**	**3**	**4**	**5**
Tactical	Patrol	Scout	Attack	Penetr	Evade	Cruise

Table 3. The specific features.

Features	**Description**
Velocity	the speed of the aircraft
Altitude	the height above a reference point
Latitude and Longitude	the geographical coordinates
Climb Angle	the angle of aircraft's ascent trajectory
Attack Angle	the angle at which it approaches a target
Radar Signals	the status of the radar

Analyzing relevant features of a combat entity allows for the determination of target intent. Taking the example of an attack intent, the relationships between velocity, altitude, radar status, and attack intent are illustrated.

Velocity and attack intent: The target's speed holds significant value in assessing the attacking intent of an airstrike target. Typically, fighter jets employ high speeds during aerial combat, often flying at around 70% to 80% of their maximum speed.

Altitude and attack intent: Altitude also plays a crucial role in determining the attacking intent of an airstrike target. Aerial combat typically occurs within a specific range of high altitudes (1000∼6000 m). Additionally, the type of missile carried can influence the altitude from which the attack is initiated. For example, if the attacking aircraft carries missiles designed for long-range engagement, it may approach our formation from medium altitude before launching an attack.

Radar status and attack intent: during an air-to-air attack by a fighter jet, the onboard radar is usually kept in an active state.

4. Methodology

This section introduces the proposed dual pipeline with the tactical behavior graph convolutional network, termed TBGCN, which adopts implicit graph topology as prior knowledge to guide learning tactical knowledge. The overview of the TBGCN is shown in Figure 1. Specifically, we started by constructing the behavior sequence topological graph based on the relationships between intent and behavior nodes, enabling a more comprehensive representation of global information. Following this, the graph convolutional model was applied to learn the representation of this topological graph. Then, sequence learning and graph representation learning were combined to extract the different features and utilize the attention mechanism as the fusion module, merging the two representations to obtain the global representation. Next, a readout function aggregated node features to obtain the overall feature representation, describing global characteristics. Finally, a softmax classification was employed to identify specific intentions.

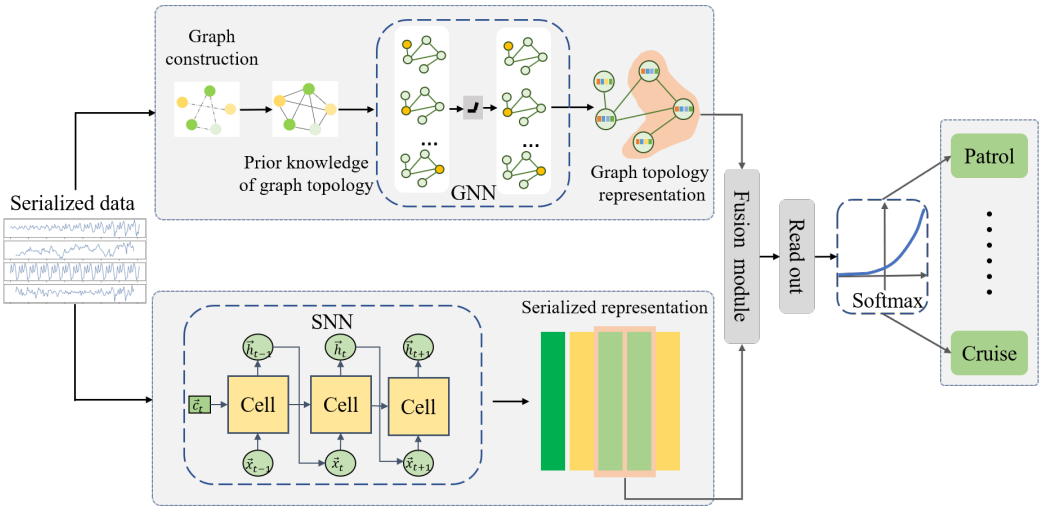


Figure 1. The architecture of dual pipeline with Tactical Behavior Graph Convolutional Network (TBGCN).

4.1. Graph Representation Learning Pipeline

The graph representation learning pipeline is given in this section. This pipeline is composed of the graph neural networks (GNN), which build an implicit graph topology based on the decomposition behavior sequence of the tactical mission to guide the supervised learning by prior knowledge.

4.1.1. Graph Construction

This section introduces graph representation learning to incorporate implicit domain knowledge into guiding the supervised learning process with neural networks. Leveraging the decomposition of combat tasks, we construct a tactical behavior sequence topological graph to model the global feature. This graph structure embodies a heterogeneous nature, characterized by two distinct node types. As illustrated in Figure 2, it exemplifies reconnaissance, attack, and patrol. This graph comprises behavior and intent nodes, where edges of the same color as intent nodes represent intent–behavior relationships, while edges between behavior nodes are depicted in green. In constructing this heterogeneous graph, primary consideration was given to two fundamental elements within the datasets: the data samples and their intent labels. Each data sample is denoted as a data node, while each intent label corresponds to a label node. The pivotal concept underlying this process revolves around establishing interconnections between nodes to capture both intrinsic data sample structures and associations between labels.

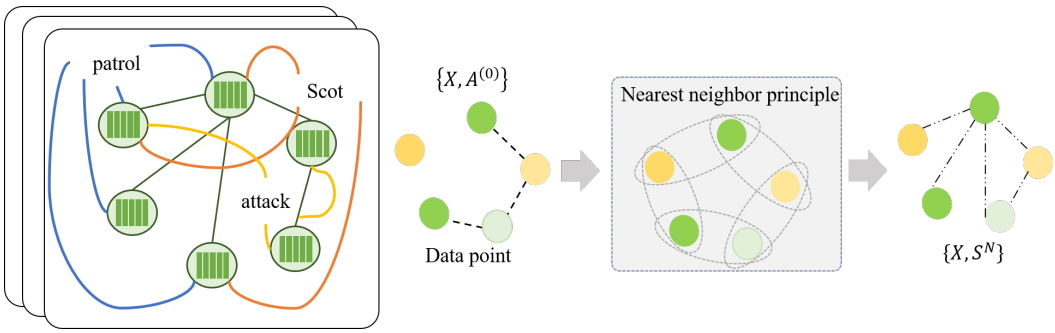


Figure 2. The tactical behavior sequence topological graph.

Specifically, we adopted two distinct methodologies for graph construction: the containment principle and the nearest neighbor principle.

The containment principle: We employ the containment principle, building the connection between each data element and its corresponding label nodes. This linkage

highlights the shared information between the data node and the label node, emphasizing the shared attributes and intention information. By connecting each data element directly to its label node, we accentuate their belongingness and category relationships, facilitating a more focused representation of class-based information. This structured linkage minimizes the complexity of the graph while reinforcing the understanding of how data elements align with specific categories or labels.

The nearest neighbor principle: We employ the nearest neighbor principle, building the connection between each data element and its adjacent elements to share the intrinsic relationships or similarities. To implement the nearest neighbor principle, we establish links to its nearest neighbors, indicating their similarity or proximity within the data space. By creating connections between data elements based on proximity or similarity, this method facilitates the representation of localized information, emphasizing the closeness or likeness of data points.

Formally, consider the topology as a collection of five elements:

$$\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \mathbf{A}_i, \mathbf{X}_i, \mathbf{E}_i\}, \tag{2}$$

where $\mathcal{G}_i$ denotes the graph topology set of target i; $\mathcal{V}_i$ is a set of vertices or node of $\mathcal{G}_i$; $\mathcal{E}_i$ indicates a set of edge; $\mathbf{A}_i \in \mathbb{R}^{N_i \times N_i}$ represents the adjacency matrix of $\mathcal{G}_i$; N_i is the number of nodes; $\mathbf{X}_i \in \mathbb{R}^{N_i \times D}$ represents feature matrix of node; D denotes the feature dimension; and $\mathbf{E}_i$ is the feature matrix of edges, if the feature of each edge is expressed by a vector $\vec{e}_t \in \mathbb{R}^E$, where E indicates the edge eigenvector dimension, then $\mathbf{E}_i \in \mathbb{R}^{N_i \times N_i \times E}$ is a three-dimensional tensor.

4.1.2. Graph Representation Learning

Graph representation learning learns implicit knowledge from graph topology, and the key is to extract contextual information from the graph node and achieve an informative representation. Figure 3 illustrates the overview of graph representation learning, where it utilizes a neighborhood aggregation scheme to learn latent node representation, following an AGGREGATION–COMBINE–PREDICTION pipeline.

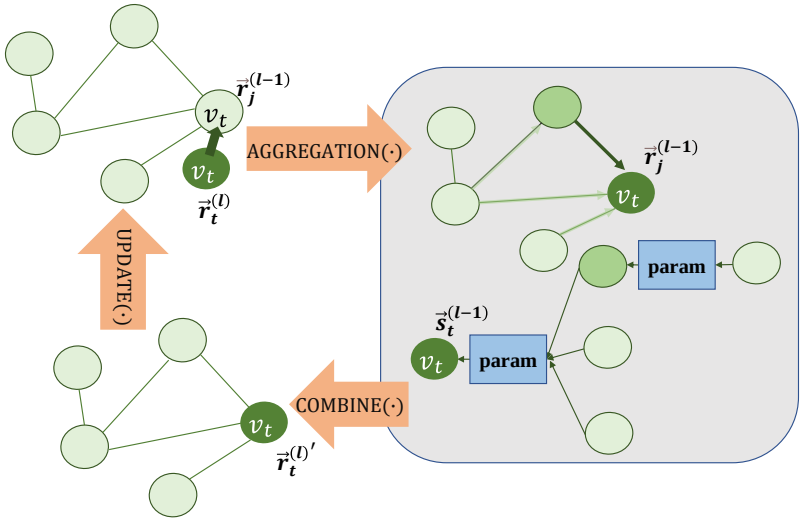


Figure 3. The graph representation learning.

The structure can be defined as follows:

$$\begin{aligned}
&\mathbf{R}_{\mathcal{G}} = \left[\vec{r}_1^{(l)} \cdots\cdots \vec{r}_t^{(l)}\right]^{\mathrm{T}},\\
&\vec{r}_t^{(l)} = \mathrm{UPDATE}\left(\vec{r}_t^{(l)'}\right),\\
&\vec{r}_t^{(l)'} = \mathrm{COMBINE}\left(\left\{\vec{s}_t^{(l-1)}, \vec{r}_t^{(l-1)}\right\}\right),\\
&\vec{s}_t^{(l-1)} = \mathrm{AGGREGATION}\left(\left\{\vec{r}_j^{(l-1)} : v_j \in \mathcal{N}_t\right\}\right),
\end{aligned} \tag{3}$$

where $\mathbf{R}_{\mathcal{G}} = [\vec{r}_1^{(l)} \cdots\cdots \vec{r}_t^{(l)}]^T$ of the last layer l represents the topological representation of the learned graph; UPDATE$(\cdot)$ denotes an update function that can learn parameters and update the representation $\vec{r}_t^{(l)'}$ of node v_t adaptively for getting a new representation $\vec{r}_t^{(l)}$; COMBINE$(\cdot)$ is a combination function to combine the aggregation representation $\vec{s}_t^{(l-1)}$ of v_t with its own representation $\vec{r}_t^{(l-1)}$ for obtaining the representation $\vec{r}_t^{(l)'}$ of node v_t in next layer l; AGGREGATION$(\cdot)$ represents an aggregation function to aggregate the hidden representation of $\mathcal{N}_t$ for obtaining the $\vec{s}_{t^{(l-1)}}$; $\mathcal{N}_t \subseteq \mathcal{V}$ indicates the neighborhood of $v_t \in \mathcal{V}$, v_j denotes the neighborhood of v_t; $\vec{r}_t^{(l-1)}$ is the hidden representation of v_t in the layer $l-1$, pay attention to the $\vec{r}_t^{(0)} = \vec{x}_t$.

4.1.3. Tactical Behavior Graph Convolutional Network

In this section, the employed Graph Convolutional Network (GCN) operates on a specific graph $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \mathbf{A}_i, \mathbf{X}_i, \mathbf{E}_i\}$. Here, $\mathcal{V}_i$ represents the set of nodes, $\mathcal{E}_i$ indicates the set of edges, $\mathbf{A}_i$ is the adjacency matrix, $\mathbf{X}_i$ stands for node features, and $\mathbf{E}_i$ denotes edge features.

The GCN model is designed to understand relationships within this graph. For this specific $\mathcal{G}_i$, where the adjacency matrix $\mathbf{A}_i$ and node features $\mathbf{X}_i$ are considered as inputs, the GCN proceeds as follows.

Given an input feature sequence $\mathbf{X}_i$ for N nodes of dimension M, and assuming the adjacency matrix $\mathbf{A}_i$ alongside its degree matrix D_i, where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The node features $\mathbf{X}_i$ and adjacency matrix $\mathbf{A}_i$ are utilized as inputs to the GCN model. The model aggregates information from neighboring nodes and computes node representations $H^{(l)}$, following Equation (4):

$$H^{(l)} = \sigma(\hat{\mathbf{A}}_i \mathbf{X}_i \mathbf{W}_0), \tag{4}$$

where σ is a non-linear activation function, and $\hat{\mathbf{A}}_i = \hat{\mathbf{D}}_i^{-1/2} \mathbf{A}_i \hat{\mathbf{D}}_i^{-1/2}$ normalizes the adjacency matrix for the graph $\mathcal{G}_i$. Here, $\mathbf{W}_0$ represents the weight matrix.

Successively, another GCN layer is added to incorporate higher-order neighborhood information, as outlined in Equation (5):

$$H^{(l+1)} = \sigma(\hat{\mathbf{A}}_i H^{(l)} \mathbf{W}^{(l)}). \tag{5}$$

4.2. Sequence Learning Pipeline

The combat data received by sensors is continuous sequential data with a long-time dependence. Therefore, we propose a sequence learning pipeline composed of sequence neural networks (SNN), such as long short-term memory (LSTM), the variants of LSTM and transformer, for such dependence to learn a representation. The sequence learning pipeline can be defined as follows:

$$
\begin{aligned}
\mathbf{R}_{\mathcal{S}} &= \left[\vec{r}_1^{(L)} \cdots\cdots \vec{r}_t^{(L)}\right]^{\mathrm{T}}, \text{s.t. } t = 1, 2, \ldots, N, \\
\vec{r}_t &= \sigma\left(\mathbf{W}'\vec{h}_t\right), \\
\vec{h}_t &= \vec{z}^o \odot \tanh(\vec{c}_t), \\
\vec{c}_t &= \vec{z}^f \odot \vec{c}_{t-1} + \vec{z}^e \odot \vec{z}, \\
\vec{z} &= \tanh\left(\mathbf{W}\left(\vec{x}_t \| \vec{h}_{t-1}\right)\right), \\
\vec{z}^e &= \sigma\left(\mathbf{W}^e\left(\vec{x}_t \| \vec{h}_{t-1}\right)\right), \\
\vec{z}^f &= \sigma\left(\mathbf{W}^f\left(\vec{x}_t \| \vec{h}_{t-1}\right)\right), \\
\vec{z}^o &= \sigma\left(\mathbf{W}^o\left(\vec{x}_t \| \vec{h}_{t-1}\right)\right),
\end{aligned}
\tag{6}
$$

where $\mathbf{R}_{\mathcal{S}} \in \mathbb{R}^{N \times D}$ denotes the representation matrix; N represents the length of sequence data; $\vec{r}_t \in \mathbb{R}^D$ is the representation learned by Formula (6); $\sigma(\cdot)$ represents the sigmoid function; $\vec{h}_t \in \mathbb{R}^D$ is hidden transmission cell state; $\vec{z} \in \mathbb{R}^D$, $\vec{z}^e \in \mathbb{R}^D$, $\vec{z}^f \in \mathbb{R}^D$, $\vec{z}^o \in \mathbb{R}^D$ are gated states; $\mathbf{W} \in \mathbb{R}^{D \times 2D}$, $\mathbf{W}^e \in \mathbb{R}^{D \times 2D}$, $\mathbf{W}^f \in \mathbb{R}^{D \times 2D}$,$\mathbf{W}^o \in \mathbb{R}^{D \times 2D}$,$\mathbf{W}' \in \mathbb{R}^{D \times D}$ denote learning adaptive weight matrices; $\odot$ represents the Hadamard product; and $(\cdot \parallel \cdot)$ denotes the concatenation of two vectors.

Take LSTM network as an example, the schematic of LSTM neural network is shown in Figure 4. LSTM learns context by the state $\vec{h}_t$ and the cell transport state $\vec{c}_t$. The former changes rapidly, capturing short-term context dependencies. The latter changes slowly, capturing long-time context dependencies. In addition, there are three stages in the LSTM, i.e., the forget stage, the selective memory stage and the output stage. The forget stage controls the last state $\vec{z}^f$ through gate $\vec{c}_{t-1}$ to retain or forget. The selective memory stage selects memory for input $\vec{x}_t$, and the selected gating is controlled by $\vec{z}^e$. The output stage determines the outputs for the current state that is controlled through $\vec{z}^o$.

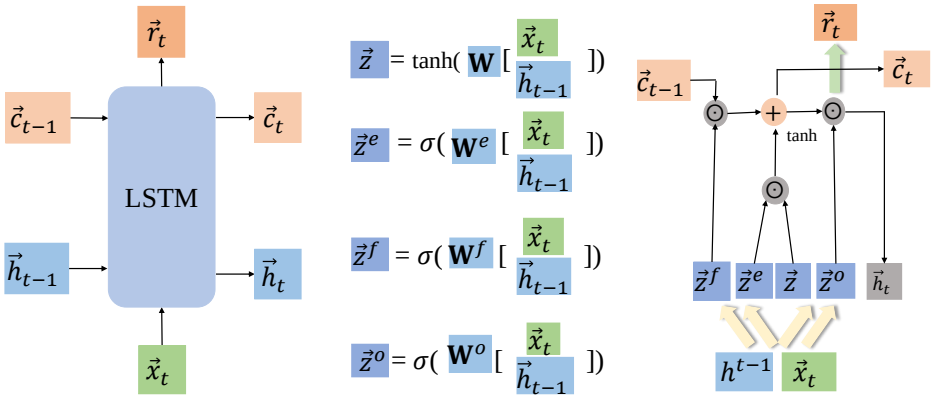


Figure 4. Schematic of LSTM neural network.

The sequence learning pipeline can learn the time dependency from data. However, battlefield data not only has time dependency, but also contains tactical relationships; that is, the behavior sequence of the tactical mission needs to be performed. Therefore, we construct a graph topology as the prior knowledge and utilize the graph representation learning pipeline to combine the sequence learning pipeline guiding the supervised learning.

4.3. Fusion Module Based on Attention Mechanism

We have learned the representation of $\mathbf{R}_\delta = \left[\vec{r}_{\delta_1} \cdots\cdots \vec{r}_{\delta_t}\right]^{\mathrm{T}} \in \mathbb{R}^{N \times D}$ and $\mathbf{R}_G = \left[\vec{r}_{G_1} \cdots\cdots \vec{r}_{G_t}\right]^{\mathrm{T}} \in \mathbb{R}^{N \times D^{(L)}}$ by sequence learning pipeline and graph representation learn-

ing pipeline, respectively. In this section, we propose a fusion module based on attention mechanism to merge the two types of representation, which can be defined as follows:

$$\begin{aligned} \mathbf{R} &= [\vec{r}_1 \cdots\cdots \vec{r}_t]^{\mathrm{T}}, \\ \vec{r}_t &= \sum_k \left(\alpha_{\delta_k} \vec{r}_{\delta_t} + \alpha_{\mathcal{G}_k} \vec{r}_{\mathcal{G}_t}\right), \\ \alpha_{\mathcal{G}_k} &= \vec{a}_{\mathcal{G}_k}^{\mathrm{T}} (\mathbf{\Theta}_{\mathcal{G}} \vec{r}_{\mathcal{G}_t} \| \mathbf{\Theta}_{\mathcal{S}} \, \vec{r}_{\mathcal{S}_t}), \\ \alpha_{\mathcal{S}_k} &= \vec{a}_{\delta_k}^{\mathrm{T}} (\mathbf{\Theta}_{\mathcal{S}} \vec{r}_{\mathcal{S}_t} \| \mathbf{\Theta}_{G} \vec{r}_{G_t}), \end{aligned} \tag{7}$$

where $\mathbf{R} \in \mathbb{R}^{N \times D^{(L)}}$ denotes the final representation; $\mathbf{\Theta}_{\mathcal{S}} \in \mathbb{R}^{D^{(L)} \times D}$ and $\mathbf{\Theta}_{\mathcal{G}} \in \mathbb{R}^{D^{(L)} \times D^{(L)}}$ represent learnable adaptive parameter matrices; $\vec{a}_{\mathcal{S}}^{\mathrm{T}} k$ and $\vec{a}_{\mathcal{G}_k}^{\mathrm{T}}$ are parameter vectors of attention mechanisms, which can learn adaptively. Compressing the representation $\mathbf{R}$ in matrix by READOUT$(\cdot)$ function:

$$\vec{r} = \mathrm{READOUT}(\mathbf{R}) = \frac{1}{N} \sum_t \vec{r}_t, \tag{8}$$

where $\vec{r} \in \mathbb{R}^{D^{(L)}}$ and $D^{(L)}$ consistent with the sample number of categories. Putting $\vec{r}$ into the structure with softmax as the classifier, we obtain the result y_i:

$$y_i = \mathrm{Softmax}(\vec{r}). \tag{9}$$

Then, the representation obtained by sequence learning pipeline (Formula (6)) and the graph representation learning pipeline (Formula (3)) was merged by the attention mechanism (Formula (7)) for final representation $\vec{r}$ (Formula (8)) and then, putting representation $\vec{r}$ into Softmax (Formula (9)), we obtain the result y_i. The whole process of learning and reasoning finds the knowledge $\vec{r}$ and predicts the behavior y_i, which provides a tactical reference for the commanders. The algorithm of the dual fusion pipeline is shown in Algorithm 1.

Algorithm 1 Dual fusion pipeline.

Input:
The combat data of target i: $\mathbf{X}_i \leftarrow [\vec{x}_1 \cdots\cdots \vec{x}_t]^T \in \mathbb{R}^{N_i \times D}$
The representation of sequence learning pipeline: $\mathbf{R}_{\mathcal{S}}$;
The representation of sequence learning pipeline: $\mathbf{R}_{\mathcal{G}}^{(L)}$.
Output:
Cross-entropy loss of the predicted and true behavioral states of target i: $\mathcal{L}_{\textbf{Cross-Entropy}}, y_i$.

for each $e \leftarrow 1 : epoch$ **do**
  $\left[\vec{r}_{\delta_1}, \cdots, \cdots, \vec{r}_{\delta_t}\right] \leftarrow \mathbf{R}_{\mathcal{S}}^{\mathrm{T}}$
  $\left[\vec{r}_{G_1}, \cdots, \cdots, \vec{r}_{G_t}\right] \leftarrow \mathbf{R}_{G}^{(L)^{\mathrm{T}}}$
  if $e = 1$ **then**
    Initialize $\vec{a}_{S_k}^{\mathrm{T}}, \vec{a}_{G_k}^{\mathrm{T}}, \mathbf{\Theta}_S, \mathbf{\Theta}_G$;
  end if
  $\alpha_{\delta_k} \leftarrow \vec{a}_{\mathcal{S}_k}^{\mathrm{T}} (\mathbf{\Theta}_{\mathcal{S}} \vec{r}_{\mathcal{S}_t} \| \mathbf{\Theta}_{\mathcal{G}} \vec{r}_{\mathcal{G}_t})$
  $\alpha_{G_k} \leftarrow \vec{a}_{\mathcal{G}_k}^{\mathrm{T}} (\mathbf{\Theta}_{\mathcal{G}} \vec{r}_{\mathcal{G}_t} \| \mathbf{\Theta}_{\mathcal{S}} \vec{r}_{\mathcal{S}_t})$
  $\vec{r}_t \leftarrow \sum_k \left(\alpha_{s_k} \vec{r}_{\mathcal{S}_t} + \alpha_{\mathcal{G}_k} \vec{r}_{\mathcal{G}_t}\right)$
  $\vec{r} \leftarrow \frac{1}{N} \sum_t \vec{r}_t$
  $y_i \leftarrow \mathrm{Softmax}(\vec{r})$
  return $\mathcal{L}_{\text{Cross-Entropy}} (y_i, y), y_i$;
end for

5. Experiments

In this section, we evaluate the effectiveness of the proposed TBGCN on the tactical dataset. We first present the experimental setup, including the datasets, metrics, and implementation for our experiments. After that, comparative experiments and analysis experiments are introduced.

5.1. Experimental Setup

5.1.1. Datasets

We introduce a tactical dataset in this work, which comprises multi-dimensional time-series data and is generated through simulation software. Specifically, for each labeled intent, we script different tactics, which are executed by the aircraft entity to generate corresponding data. Due to factors such as unstable sampling and transmission environments, the raw data may contain certain outliers and noise, leading to computational errors and affecting the convergence of the model. To address this issue, we employ quantitative standardization methods for preprocessing the raw data.

Data normalization: Combat features from aircraft models exhibit significant differences in the states of various features, necessitating normalization to eliminate dimensional influences. Specifically, dynamic attribute data corresponding to combat intent has substantial fluctuations, and some features have large data ranges.

For example, a fighter jet attacking at maximum speed may reach speeds of around 2000 km/h, causing considerable fluctuations compared to its cruising speed. Additionally, the altitudes corresponding to low-altitude and super-high-altitude reconnaissance are 100–1000 m and above 15,000 m, respectively. Faced with such data features, the Z-score normalization method is chosen to standardize the data to a distribution with a mean of 0 and a variance of 1, as described by Equation (10):

$$x_{\text{scale}} = \frac{x - \mu}{\sigma}, \tag{10}$$

where μ is the mean of all sample data, and σ is the standard deviation of all sample data.

Dataset partitioning: This dataset comprises six tactical missions, i.e., patrol, scout, attack, penetration, evade, cruise. Seventy percent of the data are allocated for the training set, and the remaining 30% for the test set. The intent corresponding to each numeric value is mapped to one-hot encoding, representing each intent with binary values.

5.1.2. Metrics

We utilize accuracy, precision, recall, and F_1-score to evaluate the performance of the proposed method in comparative and analysis experiments. Specifically, accuracy is the percentage of correctly predicted samples, which assess global accuracy. However, the accuracy cannot precisely measure the model performance when the samples are unbalanced. Therefore, we assist evaluation with the other metrics, which are defined as:

$$\text{Accuray} = \frac{TP + TN}{TP + FN + FP + TN}, \tag{11}$$

$$\text{F}_1\text{-score} = \frac{2 \times precision \times recalll}{precision \ + recall}, \tag{12}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{14}$$

where TP, TN, FP, FN represent True Positive, True Negative, False Positive, and True Negative, respectively. Precision refers to the probability of true positive among the samples predicted as positive in the results; Recall refers to the probability that the positive of the

original is finally correctly predicted as a positive. Additionally, we apply the confusion matrix to visualize the classification performance of the model.

5.1.3. Baselines

To evaluate our TBGCN, we compare it with a wide range of baselines, including the RNN-based models: LSTM [32], D-LSTM [33], and Bi-LSTM [34]; the CNN-based model, TextCNN [16]; and the attentive-based model, transformer [35]. In addition, we also compared it with some current works, including the work of Zhou et al. [36], Zhao et al. [37], and Wang et al. [38].

5.1.4. Implementation Details

The hardware equipment, manufactured by Alienware in Kunshan, China, consists of an Intel 8-core i7-8700K CPU with a 3.70 GHz processor, 64 GB of memory, and a GeForce RTX 2060OC graphics card. 1T SSD. The deep learning framework is Tensorflow 2.0. In this work, a cross-entropy loss is utilized, and the optimizer Adam is introduced to update the learning rate.

5.2. Comparative Experiments

We conduct extensive experiments with baselines in Section 5.1 to comprehensively evaluate our proposed methods. The results are shown in Table 4. In addition, we also present an evaluation of the proposed method on the previous works [36–38].

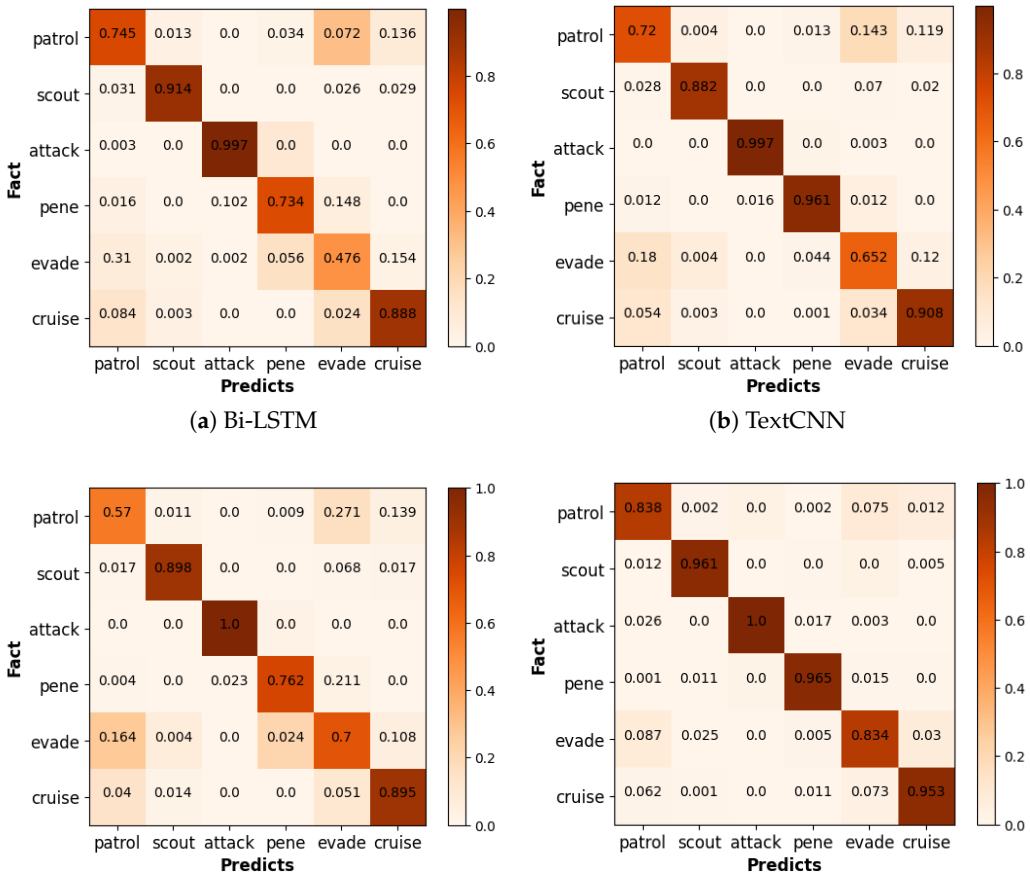
Table 4. The comparison of discovering tactical knowledge.

Methods	Models	Model Type	Accuracy	F_1 Score	Precision	Recall
Deep learning models	LSTM [32]	RNN	83.43	85.27	83.44	84.14
	D-LSTM [33]	RNN	82.94	84.61	82.94	83.57
	Bi-LSTM [34]	RNN	**84.24**	**85.87**	**84.24**	**84.65**
	TextCNN [16]	CNN	**86.58**	**86.86**	**86.58**	**87.21**
	Transformer [35]	Attentive	80.14	77.91	77.91	78.41
Previous works	Zhou et al. [36]	RNN	81.52	84.16	81.52	82.5
	Zhao et al. [37]	RNN	81.50	84.36	81.50	82.46
	Wang et al. [38]	RNN	**83.43**	**85.27**	**83.44**	**84.14**
Dual fusion pipeline	**TBGCN**	**pipeline**	**92.26**	**93.79**	**92.26**	**92.79**

As shown in Table 4, our proposed method achieves the best performance among all the baseline models. In deep learning methods, among the RNN-based models such as LSTM, D-LSTM, and Bi-LSTM, the Bi-LSTM model exhibits the highest accuracy and F1 score, reaching 84.24% and 85.87%, respectively. This model's bidirectional structure and capacity to capture both past and future contexts contribute to its superior performance. Moreover, the CNN-based model, TextCNN, performs better than RNN-based models, achieving an accuracy of 86.58% and an F1 score of 86.86%. Its ability to capture local patterns and relationships within the input sequences contributes to its enhanced performance. Additionally, the Transformer model, known for its attention mechanisms, performs reasonably well but falls short compared to the LSTM- and CNN-based models, achieving an accuracy of 80.14% and an F1 score of 77.91%. In contrast, previous works by Zhou et al. and Zhao et al., employing RNN-based approaches, yield 81.52% and 81.50% accuracy, respectively. Notably, Wang et al.'s work achieves comparable results to the LSTM-based methods, further validating the effectiveness of these recurrent models. Among these methods, the best performance model in each type is marked with bold underline. Notably, the Dual Fusion Pipeline model, TBGCN, showcases significantly superior performance

compared to the aforementioned methods, achieving an impressive accuracy of 92.26% and an F1 score of 93.79%. This model's dual fusion strategy, combining multiple sources of information, contributes to its exceptional performance, enabling a more comprehensive understanding of the data and resulting in superior classification outcomes.

As shown in Figure 5, we give the confusion matrix of each model to visualize the classification result; each column represents the prediction, and each row represents the fact. The correct predictions of each class are on the diagonal of the matrix. The darker the color, the better the performance. In addition, we normalize by row because the categories are unbalanced. Compared with the above baselines, our dual fusion pipeline not only captures the long-distance dependencies with sequence pipeline, but also contains implicit topology-preserving prior knowledge, yet achieves better performance.



(**a**) Bi-LSTM (**b**) TextCNN

(**c**) Wang et al. (**d**) TBGCN

Figure 5. The confusion matrix of best model in different methods.

5.3. Analysis Experiments

This section compares the effectiveness of different pipelines, and analyzes the factors that affect the fusion pipeline.

5.3.1. Ablation Study of Dual Fusion Pipeline

In the ablation study, we compare the performance of dual fusion pipeline with single pipelines, i.e., sequence pipeline and graph representation pipeline.

We evaluate each pipeline, respectively, to compare the performance. The result is presented in Table 5, where we observe that the dual fusion pipeline consistently outperforms the single. To graph representation learning pipeline, some efficient graph neural networks are introduced, including Graph Convolutional Neural Network (GCN), GraphSAmple and aggreGatE (GraphSAGE), Graph Isomorphism Network (GIN), and Graph

Attention Networks (GAT). These models leverage the graph structures to capture relational information among data instances. Notably, Graph Convolutional Network (GCN) demonstrates competitive performance with an accuracy of 84.8% and an F1 score of 86.9%. To the sequence learning pipeline, we introduce D-LSTM, Bi-LSTM, and Transformer for comparison. Among these, the Bi-LSTM model achieves the highest accuracy and F1 score, demonstrating competitive performance compared to other sequence-based models. The TBGCN model achieves an exceptional accuracy of 92.26% and an impressive F1 score of 93.79%. This superior performance can be attributed to the fusion of both sequence and graph information, which exploit the implicit knowledge by graph and utilize the sequential information allowing for a comprehensive understanding of data from different perspectives. The integration of these diverse information sources enables a more holistic representation of combat intention features, contributing to superior classification accuracy.

Table 5. The comparison of the single pipeline and fusion pipeline.

Methods	Pipeline Type	Models	Accuracy	F_1 Score	Precision	Recall
Single pipeline	Sequence pipeline	LSTM [32]	83.43	85.27	83.44	84.14
		D-LSTM [33]	82.94	84.61	82.94	83.57
		Bi-LSTM [34]	84.24	85.87	84.24	84.65
		Transformer [35]	80.14	77.91	77.91	78.41
	Graph representation pipeline	GAT [39]	83.28	86.59	83.28	84.46
		GraphSAGE [40]	84.7	86.52	84.7	85.4
		GIN [41]	84.48	87.15	84.48	85.44
		GCN [42]	84.8	86.9	84.8	85.54
Dual pipeline	**Our work**	**TBGCN**	**92.26**	**93.79**	**92.26**	**92.79**

5.3.2. Analysis of the Effectiveness of Different Pipelines

We perform an analysis study on our TBGCN to investigate the contributions of the specific single pipeline. The results of the analysis study are shown in Figure 6. The following experiments are designed: (1) Retain the pipeline sequence and change the graph representation the learning pipeline to analyze the impact of the graph learning pipeline on the fusion pipeline. (2) Maintain the graph representation learning pipeline and change the sequence learning pipeline to analyze the impact of the sequence learning pipeline in the fusion pipeline. D-LSTM, Bi-LSTM, and Transformer are presented in the sequence learning pipeline. We utilize GAT, GraphSAGE, and GCN for the graph representation learning pipeline.

We can see from Figure 6 that for the GAT pipeline, the Bi-LSTM model outperforms both D-LSTM and Transformer models, achieving an accuracy of 91.88% with a narrow deviation of $\pm$0.4%, showcasing its robustness. Under the GraphSAGE pipeline, similar trends emerge, with the Bi-LSTM model exhibiting the highest accuracy of 91.98% with a deviation of $\pm$0.8%, outperforming D-LSTM and Transformer. In the GCN pipeline, all models demonstrate higher accuracy than their performance in other pipelines. The GCN pipeline, known for its ability to capture graph structure information, showcases the best performance across models. The Bi-LSTM model achieves the highest accuracy of 92.26% with a deviation of $\pm$0.3%, outperforming both D-LSTM and Transformer models.

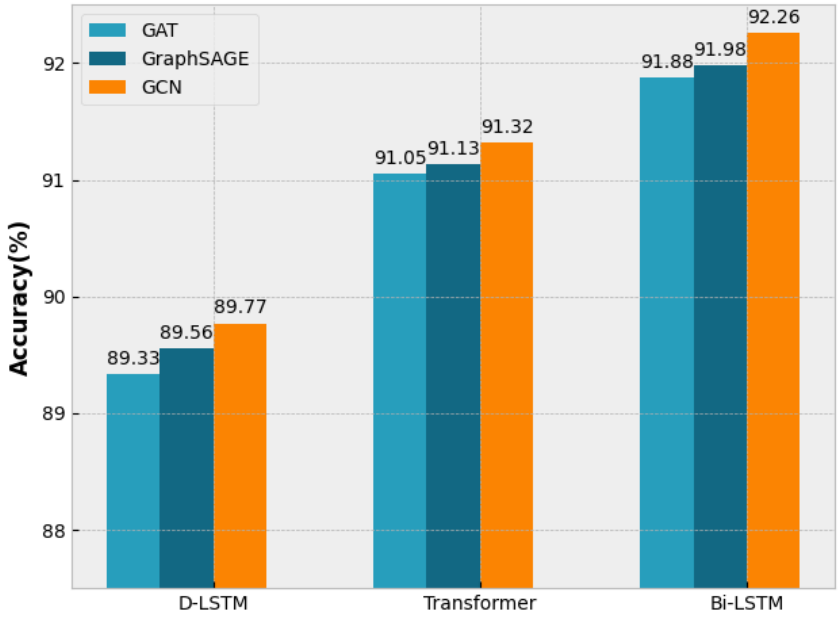


Figure 6. The comparison of different single pipelines in the fusion pipeline.

Moreover, the results show that changing the graph representation learning pipeline has no obvious impact on the accuracy, but changing the sequence learning pipeline can significantly affect the accuracy of the fusion pipeline. Our results successfully demonstrate when the sample size is sufficient, there is little difference in the implicit information obtained by different graph representation learning pipelines. Therefore, changing the graph representation learning pipeline cannot effectively improve the accuracy of the fusion pipeline. For example, when maintaining D-LSTM, the accuracy difference between GAT and GraphSAGE is only 0.23%. For sufficient samples, sequence information is significant. How to fully discover the sequence information in the data is crucial for performance. So, the performance will significantly be impacted due to the difference in the ability to capture the serialized information. For example, compared with D-LSTM, the Transformer performs better due to the attention mechanism, which can capture global sequence representation under the same graph representation learning pipeline. In conclusion, when the sample size is sufficient, the effect of the sequence learning pipeline on the fusion pipeline is more obvious.

5.3.3. Analysis of Different Fusion Modules

In this section, an analysis study is performed to demonstrate the superiority of our fusion module. To do this, the traditional fusion modules, i.e., mean, sum, and max are introduced to compare with our module based on attention. The mean operation computes the average of features, the sum totals their values, and the max extracts the maximum value among the features. The results of different fusion modules are shown in Figure 7.

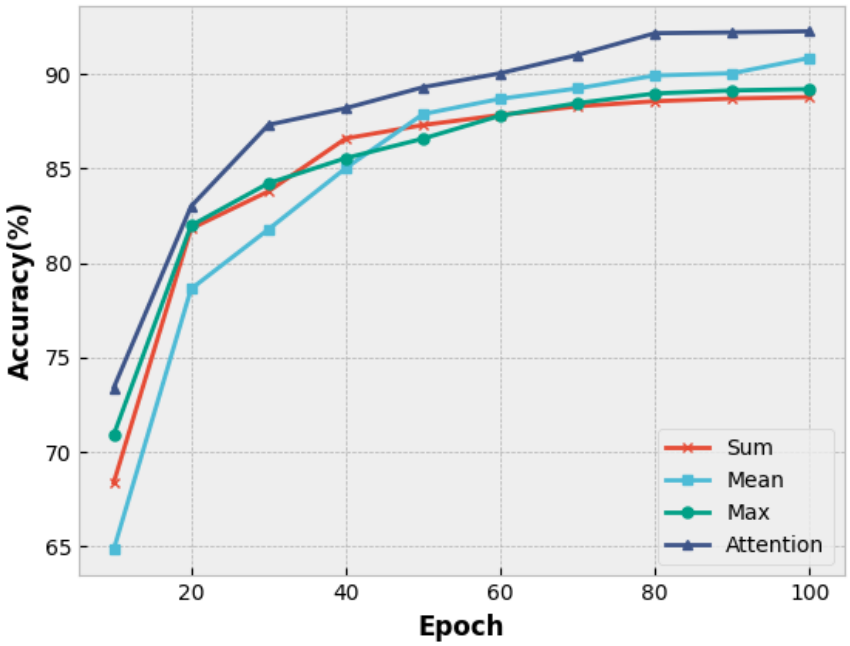


Figure 7. The comparison of different fusion modules.

It is observed that the proposed attention mechanism outperforms other fusion modules significantly, which shows the effectiveness of attention capturing the importance of features, and the simple sum, max, or mean module may lose the varying importance of features. The attention-based fusion module emphasizes its capability to effectively discern feature importance, contributing to improved performance compared to the basic aggregation methods. This result underscores the importance of attention mechanisms in capturing nuanced feature relationships.

6. Conclusions

In this study, a dual fusion pipeline is presented for discovering tactical knowledge, aiming to address the challenge of uncovering the target combat intent from the provided time-series data. The dual fusion pipeline constructs the implicit graph topology from the tactical serialized data, which is utilized as prior knowledge to guide the learning process. The sequence learning pipeline and the graph representation learning pipeline combine to learn the representations through tactical serialized data. Then, the learned representations are fused through the attention mechanism to obtain the final tactical knowledge, which provides an intelligent reference for commanders. To verify the effectiveness of our dual fusion pipeline, we conduct exhaustive experiments on existing baselines. The experiment result shows our approach achieving significant improvement in most evaluation criteria and proves that the proposed method can provide a reference for the commander to efficient recognition of enemy intent.

In future work, we will extend our dual fusion pipeline with the parallel pipeline learning of a multi-modal model to capture the rich structure and content information in battlefield data.

Author Contributions: Conceptualization, X.W. and P.H.; methodology, X.W. and H.Y.; software, P.H. and X.S.; validation, X.S., J.W. and Y.G.; formal analysis, Y.G.; investigation, J.W.; resources, Y.G.; data curation, X.S.; writing—original draft preparation, X.W..; writing—review and editing, P.H.; visualization, H.Y.; supervision, X.S.; project administration, J.W.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Basic scientific research grant number: JCKY2021205A004. The APC was partial funded by Basic scientific research.

Data Availability Statement: Data available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nikitha, M.A.; Sai Swetha, B.S.; Mantripragada, K.H.; Jayapandian, N. The future warfare with multidomain applications of artificial intelligence: Research perspective. In Proceedings of the Second International Conference on Sustainable Expert Systems: ICSES 2021, Singapore, 17–18 September 2021; pp. 2329–2341.
2. Kajaree, D.; Rabi, B.N. A survey on machine learning: concept, algorithms and applications. *Int. J. Innov. Res. Comput. Commun. Eng.* **2017**, *5*, 1301–1309.
3. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3981–3989.
4. Olivier, C.; Bernhard, S.; Alexander, Z. Semi-Supervised Learning. *IEEE Trans. Neural Netw.* **2009**, *20*, 542.
5. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Allen, G.I. Handbook of Graphical Models. *J. Am. Stat. Assoc.* **2020**, *115*, 1556–1557.
7. Judea, P. Fusion, propagation, and structuring in belief networks. *Artif. Intell.* **1986**, *29*, 366–413.
8. Xu, L.; Qiao, D.; Liang, Y.; Xu, L. A Novel DBN-based Intention Inference Algorithm for Warship Air Combat. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; IEEE: New York, NY, USA; pp. 916–921.
9. Dahlbom, A. A comparison of two approaches for situation detection in an air-to-air combat scenario. In Proceedings of the Modeling Decisions for Artificial Intelligence: 10th International Conference, Barcelona, Spain, 20–22 November 2013; pp. 70–81.
10. Perkins, D.N.; Johnson, P.N.; Wason, P.C. *Thinking: Readings in Cognitive Science*; Cambridge University Press: Cambridge, UK, 1977.
11. Teuvo, K. The self-organizing map. *Neurocomputing* **1998**, *21*, 1–6.
12. Welling, M.; Rosenzvi, M.; Hinton, G. Exponential Family Harmoniums with an Application to Information Retrieval. *Zvi* **2005**, *17*, 123–128.
13. Hinton, G.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *131*, 504–507. [CrossRef]
14. Ahmed, A.A.; Mohammed, M.F. SAIRF: A similarity approach for attack intention recognition using fuzzy min-max neural network. *J. Comput. Sci.* **2018**, *25*, 467–473. [CrossRef]
15. Cheng, Q.; Tong, M.S.; Yu, X.H.; Zhuang, S.L. CNN-based visual processing approach for biological sample microinjection systems. *Neurocomputing* **2021**, *459*, 70–80. [CrossRef]
16. Yoon, K. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Empirical Methods in NLP Process (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
17. Zhu, P.; Wang, B.; Tang, K.; Zhang, H.; Cui, X.; Wang, Z. A knowledge-guided graph attention network for emotion-cause pair extraction. *Knowl. Based Syst.* **2024**, *286*, 111342. [CrossRef]
18. Cheng, L.; Zhu, P.; Sun, W.; Han, Z.; Tang, K.; Cui, X. Time series classification by Euclidean distance-based visibility graph. *Phys. Stat. Mech. Appl.* **2023**, *625*, 129010. [CrossRef]
19. Li, H. Multivariate time series clustering based on common principal component analysis. *Neurocomputing* **2019**, *349*, 239–247. [CrossRef]
20. Cheng, Y.; Zhang, W.D. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2018**, *272*, 63–73. [CrossRef]
21. Xue, J.; Zhu, J.; Xiao, J.; Tong, S.; Huang, L. Panoramic convolutional long short-term memory networks for combat intension recognition of aerial targets. *IEEE Access* **2020**, *8*, 183312–183323. [CrossRef]
22. Li, Y.; Li, K.; Zheng, Y.; Morys, B.; Pan, S.; Wang, J. Threat assessment techniques in intelligent vehicles: A comparative survey. *IEEE Intell. Transp. Syst. Mag.* **2020**, *13*, 71–91. [CrossRef]
23. Zhang, C.; Zhou, Y.; Li, H.; Liang, F.; Song, Z.; Yuan, K. Combat Intention Recognition of Air Targets Based on 1DCNN-BiLSTM. *IEEE Access* **2023**, *11*, 134504–134516. [CrossRef]
24. Zhou, T.; Chen, M.; Wang, Y.; He, J.; Yang, C. Information entropy-based intention prediction of aerial targets under uncertain and incomplete information. *Entropy* **2020**, *22*, 279. [CrossRef]
25. Xia, J.; Chen, M.; Fang, W. Air Combat Intention Recognition with Incomplete Information Based on Decision Tree and GRU Network. *Entropy* **2023**, *25*, 671. [CrossRef] [PubMed]
26. Xu, Y.; Cheng, S.; Zhang, H.; Chen, Z. Air target combat intention identification based on IE-DSBN. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 12–14 June 2020; IEEE: New York, NY, USA; 2020; pp. 36–40.
27. Teng, F.; Song, Y.; Guo, X. Attention-TCN-BiGRU: An Air Target Combat Intention Recognition Model. *Mathematics* **2021**, *9*, 2412. [CrossRef]
28. He, W.; Wang, Y.; Cui, L.; Su, R.; Wei, L. Learning Embedding Features Based on Multisense-Scaled Attention Architecture to Improve the Predictive Performance of Air Combat Intention Recognition. *IEEE Access* **2022**, *10*, 104923–104933.

29. Yang, Z.; Sun, Z.X.; Piao, H.Y.; Huang, J.C.; Zhou, D.Y.; Ren, Z. Online hierarchical recognition method for target tactical intention in beyond-visual-range air combat. *Def. Technol.* **2022**, *18*, 1349–1361. [CrossRef]
30. Wang, Y.; Wang, J.; Fan, S.; Wang, Y. Quick intention identification of an enemy aerial target through information classification processing. *Aerosp. Sci. Technol.* **2023**, *132*, 108005. [CrossRef]
31. Wang, X.; Yang, Z.; Zhan, G.; Huang, J.; Chai, S.; Zhou, D. Tactical Intention Recognition Method of Air Combat Target Based on BiLSTM network. In Proceedings of the 2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China, 28–30 October 2022; pp. 63–67.
32. Zhang, X.; Zhao, J.; Le, Y.C. Character-level convolutional networks for text classification. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NIPS, Montreal, QC, Canada, 7–12 Deceember 2015; pp. 649–657.
33. Yogatama, D.; Dyer, C.; Ling, W.; Blunsom, P. Generative and discriminative text classification with recurrent neural networks. In Proceedings of the thirty-fourth International Conference on Machine Learning (ICML 2017), Sydney, Australia, 6–11 August 2017.
34. Liu, P.F.; Qiu, X.P.; Huang, X.J. Recurrent neural network for text classification with multi-task learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'16), New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4 December 2017; pp. 5998–6008.
36. Zhou, W.; Yao, P.; Zhang, J.; Wang, X.; Wei, S. Combat intention recognition for aerial targets based on deep neural network. *Acta Aeronaut. Astronaut. Sin.* **2018**, *39*, 200–208.
37. Zhao, J.; Yang, H.; Qiu, L. Recognition of aerial target types based on recurrent neural network. *Aerosp. Control.* **2020**, *38*, 47–53. (In Chinese)
38. Wang, Z.G.; Wang, Y.G.; Yang, N.; Mi, Y.F.; Qu, L.X. Construction method of flight data mining model based on lstm. *Acta Aeronaut. Astronaut. Sin.* **2021**, *42*, 47–53. (In Chinese)
39. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *Stat* **2017**, *1050*, 10–48550.
40. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 12 March 2017; pp. 1024–1034.
41. Xu, K.; Hu, W.H.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
42. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.