

北京航空航天大学学报
Journal of Beijing University of Aeronautics and Astronautics
ISSN 1001-5965,CN 11-2625/V

《北京航空航天大学学报》网络首发论文

题目：基于图学习方法的无人机蜂群作战目标分配
作者：邓恒林，朱政，陈赛宇，宋申民，魏喜庆
DOI：10.13700/j.bh.1001-5965.2026.0400
收稿日期：2026-06-16
网络首发日期：2026-11-12
引用格式：邓恒林，朱政，陈赛宇，宋申民，魏喜庆．基于图学习方法的无人机蜂群作战目标分配[J/OL]．北京航空航天大学学报．
https://doi.org/10.13700/j.bh.1001-5965.2026.0400



网络首发：在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

出版确认：纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

基于图学习方法的无人机蜂群作战目标分配

邓恒林[1]，朱政[2]，陈赛宇[1]，宋申民[3]，魏喜庆[1,✉]

（1. 上海电机学院 电子信息学院，上海 201306；2. 中电科技扬州宝军电子有限公司，扬州 225003；
3. 哈尔滨工业大学 航天学院，哈尔滨 150001）
✉通信作者 E-mail: weixq@sdju.edu.cn

摘要 针对无人机蜂群目标分配中组合空间快速增长、传统图神经网络对分配关系中的边语义利用不足以及解质量与决策时效难以兼顾的问题，提出一种边感知图注意力网络（EGAT）。该方法将无人机与目标之间的摧毁概率作为边特征显式引入消息传递，利用边中心注意力、节点—边联合注意力及特征融合机制增强节点属性与分配关系的协同表征；进一步结合深度 Q 网络（DQN）构建 EGAT-DQN 顺序决策模型。在 5 架无人机、5 个目标的 1000 个测试实例上进行验证，结果显示：EGAT 平均最优性差距为 1.01%，最优解比例达到 85.20%，优于所比较的图学习方法；EGAT-DQN 的时延为 1.26 s，低于 EGAT 的 5.53 s，但解质量有所下降。结果表明，边语义建模能够有效提高目标分配解质量，并为不同决策需求下的快速求解与顺序决策提供相应方法。

关键词 无人机蜂群；目标分配；二分图；深度学习；强化学习；组合优化
中图分类号 V279；E911；TP183 文献标志码 A DOI: 10.13700/j.bh.1001-5965.2026.0400

Graph learning approach for target allocation in UAV swarm combat

Deng Henglin[1], Zhu Zheng[2], Chen Saiyu[1], Song Shenmin[3], Wei Xiqing[1,✉]

（1. School of Electronic Information, Shanghai Dianji University, Shanghai 201306, China；
2. China Electronics Technology Corporation Yangzhou Baojun Electronics Co., Ltd., Yangzhou 225003, China；
3. School of Astronautics, Harbin Institute of Technology, Harbin 150001, China）
✉E-mail: weixq@sdju.edu.cn

Abstract Target assignment for unmanned aerial vehicle (UAV) swarms faces rapidly expanding combinatorial spaces, insufficient exploitation of semantic edge information in assignment relations by conventional graph neural networks, and difficulty in balancing solution quality with decision efficiency. An edge-aware graph attention network (EGAT) was proposed to address these problems. The method explicitly incorporated the kill probabilities between UAVs and targets as edge features into message passing, and enhanced the joint representation of node attributes and assignment relations through edge-centered attention, node-edge joint attention, and feature fusion. EGAT was further combined with a deep Q-network (DQN) to construct EGAT-DQN for sequential decision making. The proposed methods were evaluated on 1,000 test instances with five UAVs and five targets. EGAT achieved a mean optimality gap of 1.01% and an optimal-solution rate of 85.20%, outperforming the compared graph-learning methods. EGAT-DQN achieved a latency of 1.26 s, lower than the 5.53 s of EGAT, but with lower solution quality. The results demonstrate that explicit modeling of edge semantics can effectively improve target-assignment quality and provide corresponding approaches for fast solving and sequential decision making under different decision requirements.

Key words UAV swarm; target assignment; bipartite graph; deep learning; reinforcement learning; combinatorial optimization

现代战争正迅速向高动态性和体系化作战模式演变。由于成本低廉且机动性强，无人机（UAV）已成为军事任务中的核心力量，人工智能技术的发展促使群集作战成为一种主流范式[1]。在群集作战

收稿日期：2026-06-16
基金项目：国家自然科学基金（62273277）
英文基金项目：National Natural Science Foundation of China (62273277)
网络首发时间：2026-11-12 10:03:55 网络首发地址：https://link.cnki.net/urlid/11.2625.v.20261111.1507.001

中，资源约束与任务复杂性并存，任务与无人机的精准匹配成为决定作战效能的关键。

武器-目标分配（WTA）问题是军事运筹学中一个经典的 NP 完全问题[2,3]。随着问题规模的增大，WTA 的计算复杂度呈指数级增长，这使得该问题在实际中极难求解。无人机群的任务分配是 WTA 的一个特例，也是智能作战系统的核心问题。其目标是使无人机资源与目标进行匹配，从而最大化整体作战效能。解决该问题的现有方法包括精确算法[4-6]、启发式算法[7-9]以及元启发式算法[10,11]。虽然精确算法能够提供最优解，但由于计算成本过高，在大型实时场景中往往难以实施。另一方面，启发式和元启发式方法虽具有更高的计算效率，但通常缺乏理论保证，容易陷入局部最优，且高度依赖人工设计的规则或领域特定知识，这限制了其对新场景的泛化能力。因此，在解决方案质量与计算效率之间取得平衡，仍是无人机群任务分配中的关键挑战。

近年来，研究人员开始转向深度学习[12,13]等数据驱动的方法，利用模型训练将耗时的在线求解任务转移给已训练好的模型，从而在部署时实现快速推理。图神经网络（GNNs）[14]凭借其强大的图结构数据处理能力，近年来得到快速发展，相关研究已从消息传递机制[15]、图卷积学习[16]、调度问题[17]、时空图卷积[18]以及图注意力机制[19]等方面不断提升图结构表示能力。在组合优化领域，GNN 已被应用于旅行推销员问题(TSP)[20]和图着色问题[21]等典型问题。GNN 能够捕捉图中的高阶关系，为建模和求解分配问题提供了新范式。然而，主流的 GNN——例如图卷积变体[16]和图注意力网络[19]——通常将边视为非语义链接，主要关注节点特征聚合，而未能充分利用丰富的边信息。在 WTA 中，边不仅连接节点，还承载着诸如摧毁概率和资源消耗等语义信息。若边语义建模不当，基于 GNN 的解决方案可能产生次优结果甚至完全失效。因此，设计一个能够敏锐感知并充分利用边特征的图学习框架，对于提升性能至关重要。

强化学习（RL）为动态决策和优化提供了强大的理论框架[22]。通过与环境的交互，RL 代理可以学习到最优策略。在无人机任务中，深度强化学习（DRL）已被广泛应用于路径规划[23]、目标跟踪[24]等任务。国内学者已从任务建模、预分配与重分配等方面，对无人机集群任务分配技术作了系统总结[25]，并针对不确定环境和航路约束分别提出多策略灰狼优化[26]以及任务分配与协同航路规划的耦合方法[27]。对于 WTA，RL 能够将该问题建模为马尔可夫决策过程（MDP），使策略网络能够直接输出分配决策，从而避免复杂的搜索过程[28]。尽管如此，独立的 RL 仍面临诸多挑战，例如随问题规模而快速扩大的状态空间、训练不稳定以及稀疏奖励等问题[13]。为增强强化学习对复杂任务状态的表征能力，近期研究将图神经网络（GNN）作为演员-评论家框架中的特征提取器，从而提升无人机对抗任务中的决策质量[29]，在本研究中，我们利用 GNN 高效地从无人机与目标的交互中提取特征，从而降低强化学习中有效状态空间的复杂度，并辅助在稀疏奖励下的策略学习。

我们首先提出了一种边感知图注意力网络（EGAT），该网络融合了节点和边信息，以增强针对 WTA 问题的图表示学习。在此框架基础上，我们进一步开发了 EGAT-DQN，该模型将 EGAT 与深度 Q 网络（DQN）相结合，以实现无人机蜂群目标分配中的高效策略学习和快速决策。

1 无人机蜂群分配问题描述

本研究探讨了无人机群在智能作战任务中的目标分配问题，采用以下假设：任务分配在单一决策周期内完成，目标价值和无人机对目标的摧毁概率已知且保持不变；训练和同分布测试均采用带权完全二部图，边权表示相应摧毁概率；不同无人机对同一目标的毁伤事件按原目标函数中的独立概率乘积计算；不显式考虑通信时延、航迹耦合、目标机动和复杂协同毁伤。在上述假设下，本文结果适用于信息完备、静态且规模接近训练设置的仿真目标分配。为此，我们将该问题形式化为二部图上的组合优化任务，并进一步融入了图结构学习和强化学习策略。

1.1 基本定义

考虑由多种类型无人机组成的无人机群。当同类型无人机的可用数量超过一架时，可将其视为属性完全相同的独立无人机。同样，敌方目标集也可划分为多种类型。设 U 表示无人机类型集，T 表示目标类型集。定义目标权重向量 $\boldsymbol{v}$，用于编码 T 中每个目标的相对重要性。设 x_{ij} 为分配变量，其中若

无人机 i 被分配至目标 j，则 $x_{ij}=1$，否则 $x_{ij}=0$。将 m 架无人机分配给 n 个目标将产生一个尺寸为 $m\times n$ 的决策矩阵，如式(1)所示：

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

式中：$\boldsymbol{X}$ 为分配矩阵；$x_{ij}\in\{0,1\}$ 为分配变量；$x_{ij}=1$ 表示将无人机 i 分配给目标 j；m 为无人机总数；n 为目标总数。

1.1.1 限制条件

该问题受以下约束条件限制：

（1）无人机可用性约束。对于每种无人机类型，分配的数量不得超过可用库存：

$$\sum_{j=1}^{n} x_{ij}\le c_i,\text{s.t.}\, i=1,2,\ldots,m,\ j=1,2,\ldots,n,\ c_i\in\mathbb{Z}^{+} \tag{2}$$

式中：x_{ij} 表示分配变量；c_i 表示类型为 i 的无人机可用数量。

（2）单分配约束。每个无人机最多只能被分配一次：

$$\sum_{j=1}^{n} x_{ij}\le 1,\text{s.t.}\, x_{ij}\in\{0,1\} \tag{3}$$

式中：x_{ij} 表示分配变量。

1.1.2 目标函数

假设无人机 i 击毁目标 j 的概率记为 $p_{ij}\in(0,1)$。若分配给目标 j 的无人机集合表示为 W_j，则目标 j 在攻击后的生存概率为 $\prod_{i\in W_j}(1-p_{ij})$。设目标 j 的价值为 v_j。因此，残值可表示为 $\sum_{j=1}^{n} v_j\prod_{i\in W_j}(1-p_{ij})$。据此，优化目标函数可表述如下，其中最优分配矩阵记为 $\boldsymbol{X}$：

$$\min\sum_{j\in J} v_j\prod_{i\in I} q_{ij}^{x_{ij}} \tag{4}$$

式中：v_j 为目标 j 的价值；q_{ij} 为无人机 i 对目标 j 的未摧毁概率；x_{ij} 为分配变量。

1.2 图建模

无人机群目标分配问题可以自然地建模为加权二部图。二部图 $G=(X,Y,E)$ 由两个互不相交的顶点集 X 和 Y 组成，E 为边集合，其中边仅存在于不同集合的节点之间。如果每条边都关联一个权重，则该图被称为加权二部图，记为 $G=(X,Y,E,W)$,在我们的建模中，每个实例对应一个完整的加权二部图。集合 X 表示无人机，集合 Y 表示敌方目标。边 $(i,j)\in E$ 编码了将无人机 i 分配给目标 j 的可行性，而边权重 w_{ij} 表示打击成功的概率 p_{ij}。在训练和编码阶段，我们采用带有边权重的完全二部图，其中边权重即为打击概率。分配决策在解码阶段生成，且不改变图的连通性。每个无人机节点 i 和目标节点 j 分别关联特征向量 $\boldsymbol{u}_i$ 和 $\boldsymbol{t}_j$，用于编码无人机资源约束和目标价值。

2　无人机蜂群分配算法

2.1 网络架构

在深入探讨详细的数学建模之前，我们首先概述所提出的边感知图注意力网络（EGAT）的整体框架和逻辑。EGAT 采用了一种分层设计，包含两个并行更新模块：边中心注意力模块和节点-边联合注意力模块。最后，融合路径进一步将边特征与邻近节点的特征进行整合，生成用于任务分配预测的

最终节点嵌入向量。

EGAT 模型以加权二部图作为输入，通过多层边-节点联合注意力机制，融合异构的节点和边特征，从而输出无人机与目标之间的匹配概率。我们首先将 WTA 问题建模为一个加权二部图 $G = (X, Y, E, W)$，其中无人机集 X 和目标集 Y 是两个不相交的顶点集，E 是它们之间的边集，W 表示边权重。

在训练阶段，我们对每个图的输入特征进行初始化。每个二部图包含固定数量的无人机节点 $|X|$ 和目标节点 $|Y|$，节点总数为 $N = |X| + |Y|$。初始节点特征维度设置为 $d_{\text{in}} = 2$，用于编码基本节点属性：使用 1 表示无人机节点，0 表示目标节点。无人机节点的另一标量特征编码了该无人机类型的可分配单位数量（本研究中每个具体单位均设为 1），而目标节点特征则编码目标值 V_j。边表示从每个无人机节点到每个目标节点的任务分配可行性，从而形成一个具有 $|X|\times|Y|$ 条边的完全二部图。边属性表示无人机 i 击落目标 j 的概率 p_{ij}，记为 $e_{ij} \in \mathbb{R}^1$.。因此，单个二部图实例的原始输入张量包括：节点特征矩阵 $H \in \mathbb{R}^{N\times d_{\text{in}}}$；表示二部图连通性的边索引（邻接）张量 A_{edge}；以及存储 p_{ij} 的边属性张量 E_{attr}。EGAT 由 l 个堆叠层组成，每个层包含 k 个注意力头。在实验中，我们设置 l = 2，k = 4。每个层的输出作为下一层的输入，最后通过一个融合层生成最终的匹配分数。网络输入包括节点特征矩阵 $\boldsymbol{H}$、边索引 A_{edge} 以及边特征 E_{attr}。在微批量处理下，多个图被拼接成一个大型图，通过批次索引区分属于不同实例的节点。我们初始化隐藏维度 d_{hidden}，并在实验中将其设为 128。无人机节点和目标节点特征首先通过具有权重 $\boldsymbol{W}_{\text{w1}} \in \mathbb{R}^{d_{\text{in}}\times d_{\text{hidden}}}$ 和 $\boldsymbol{W}_{\text{t1}} \in \mathbb{R}^{d_{\text{in}}\times d_{\text{hidden}}}$ 的线性层，随后进入具有权重 $\boldsymbol{W}_{\text{w2}} \in \mathbb{R}^{d_{\text{hidden}}\times d_{\text{out}}}$ 和 $\mathrm{W}_{\text{t2}} \in \mathbb{R}^{d_{\text{hidden}}\times d_{\text{out}}}$ 的第二个线性层，从而生成维度为 d_{out} 的特征。

我们采用一种以边为中心的注意力机制，该机制将所有指向同一目标的边建模；也就是说，对于给定的目标节点，与该目标共享的邻接边构成了邻接集 $\mathcal{N}_e$。为了保持对称性，引入了一个零辅助节点特征向量来连接当前边及其邻接边，从而形成注意力三元组，该过程如图 1 所示，图中展示了监督学习的整体工作流程。

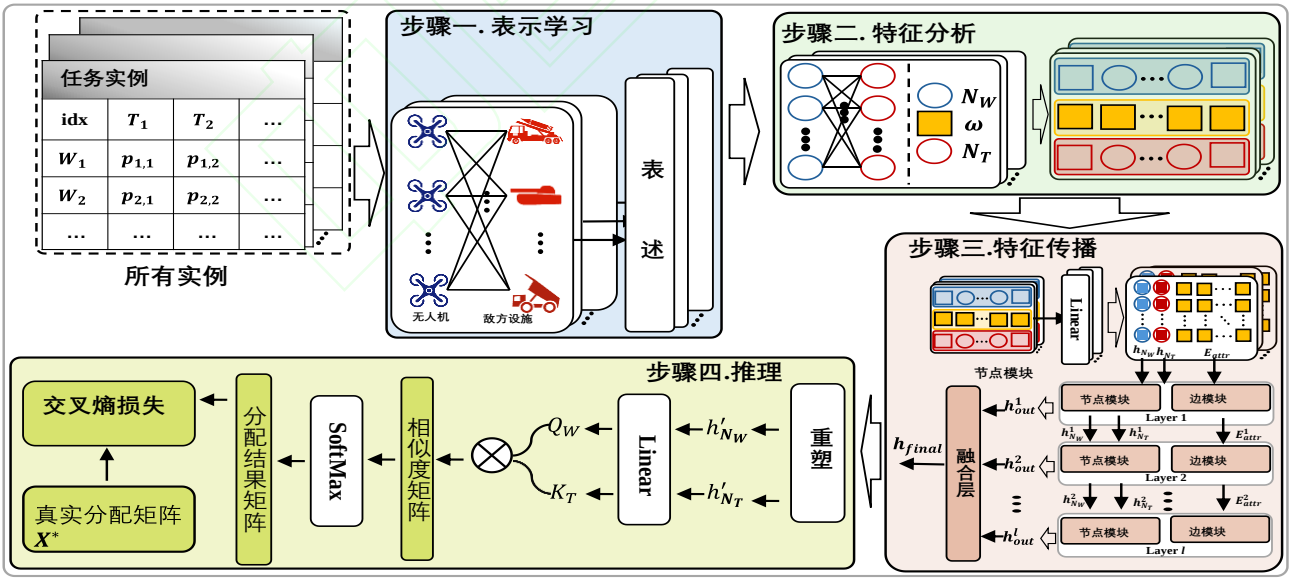

图 1 监督学习工作流程

Fig.1 Workflow of supervised learning

边缘特征的线性映射与多头变换：

$$\boldsymbol{e}_{pq}^{(l,k)} = \boldsymbol{W}_{e}^{(l,k)} \cdot \boldsymbol{e}_{pq}^{(l)} \tag{5}$$

式中：$\boldsymbol{e}_{pq}^{(l)}$ 为第 l 层中从节点 p 到节点 q 的边特征向量；$\boldsymbol{W}_e^{(l,k)}$ 为第 l 层第 k 个注意力头对应的可学习权重矩阵；$\boldsymbol{e}_{pq}^{(l,k)}$ 为变换后的边特征向量。

针对具有共同目标 j 的邻接边注意系数：

$$a_{pq}^{(l)}=\frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{(l,k)\mathrm{T}}[\boldsymbol{e}_{pq}^{(l,k)}\,\|\,\boldsymbol{e}_{q}^{(l,k)}\,\|\,\boldsymbol{h}_{j}^{(l)}]))}{\sum_{q'\in\mathcal{N}_p}\exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{(l,k)\mathrm{T}}[\boldsymbol{e}_{pq}^{(l,k)}\,\|\,\boldsymbol{e}_{q'}^{(l,k)}\,\|\,\boldsymbol{h}_{j}^{(l)}]))} \tag{6}$$

式中：$a_{pq}^{(l)}$ 为第 l 层中邻居边 q 对当前边 p 的注意力系数；$\boldsymbol{a}^{(l,k)}$ 为注意力权重向量；$\boldsymbol{e}_{pq}^{(l,k)}$ 为当前边特征向量；$\boldsymbol{e}_{q}^{(l,k)}$ 为邻居边特征向量；$\boldsymbol{h}_{j}^{(l)}$ 为目标节点 j 的特征向量；$\mathcal{N}_p$ 为与边 p 共享同一目标节点的邻居边集合；q' 为邻居边索引。

基于上述注意力系数，通过聚合邻接边来更新边特征：

$$\boldsymbol{e}_{pq}^{(l+1)}=\sigma\left(\sum_{q'\in\mathcal{N}_p}a_{pq'}^{(l)}\cdot\boldsymbol{e}_{pq'}^{(l)}\right) \tag{7}$$

式中：$\boldsymbol{e}_{pq}^{(l+1)}$ 为更新后的边特征向量；σ 为激活函数；$a_{pq'}^{(l)}$ 为注意力系数；$\boldsymbol{e}_{pq'}^{(l)}$ 为邻居边特征向量；$\mathcal{N}_p$ 为与边 p 共享同一目标节点的邻居边集合。

节点注意力模块与上述边中心注意力模块并行工作，基于原始图结构，在融合边-节点信息的同时更新节点特征。每个节点和边特征在进行注意力机制之前，都会经过一个共享的线性变换进行投影：

$$\boldsymbol{h}_{i}^{(l,k)}=\boldsymbol{W}_{h_N}^{(l,k)}\cdot\boldsymbol{h}_{N_i}^{(l)} \tag{8}$$

式中：$\boldsymbol{h}_{N_i}^{(l)}$ 为第 l 层节点 N_i 的特征向量；$\boldsymbol{W}_{h_N}^{(l,k)}$ 为可学习的投影矩阵；$\boldsymbol{h}_{i}^{(l,k)}$ 为投影后的特征向量。

对于每个头，我们将无人机节点特征、目标节点特征和边特征进行拼接，并计算注意力权重：

$$\beta_{ij}^{(l)}=\frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{b}^{(l,k)T}[\boldsymbol{h}_{i}^{(l)}\,\|\,\boldsymbol{h}_{j}^{(l)}\,\|\,\boldsymbol{e}_{ij}^{(l)}]))}{\sum_{k\in\mathcal{N}_i}\exp(\mathrm{LeakyReLU}(\boldsymbol{b}^{(l,k)T}[\boldsymbol{h}_{i}^{(l)}\,\|\,\boldsymbol{h}_{k}^{(l)}\,\|\,\boldsymbol{e}_{ik}^{(l)}]))} \tag{9}$$

式中：$\beta_{ij}^{(l)}$ 为节点 i 对邻居节点 j 的注意力系数；$\boldsymbol{b}^{(l,k)}$ 为注意力权重向量；$\boldsymbol{h}_{i}^{(l)},\boldsymbol{h}_{j}^{(l)}$ 分别为节点 i 和 j 的特征向量；$\boldsymbol{e}_{ij}^{(l)}$ 为边 (i,j) 的特征向量；$\mathcal{N}_i$ 为节点 i 的邻居节点集合；k 为邻居节点索引。随后，节点特征被更新并传递到下一层：

$$\boldsymbol{h}_{i}^{(l+1)}=\sigma\left(\sum_{\boldsymbol{h}_{j}^{(l)}\in\mathcal{N}_p}\beta_{ij}^{(l)}\cdot\boldsymbol{h}_{j}^{(l)}\right) \tag{10}$$

式中：$\boldsymbol{h}_{i}^{(l+1)}$ 为第 $l+1$ 层节点 i 的特征向量；$\beta_{ij}^{(l)}$ 为注意力系数；$\boldsymbol{h}_{j}^{(l)}$ 为邻居节点 j 的特征向量；$\mathcal{N}_i$ 为节点 i 的邻居节点集合。

在融合路径中，为了进一步增强边-节点协同建模，在每次节点更新后，我们引入了一条融合路径，该路径将入边特征与邻接节点特征进行联合聚合：

$$\boldsymbol{m}_{i}^{(l)}=\sigma\left(\sum_{j\in\mathcal{N}_i}a_{ij}^{(l)}\cdot\left[\boldsymbol{h}_{j}^{(l)}\,\|\,\boldsymbol{e}_{pq}^{(l)}\right]\right) \tag{11}$$

式中：$\boldsymbol{m}_{i}^{(l)}$ 为融合后的中间特征向量；$a_{ij}^{(l)}$ 为注意力系数；$\boldsymbol{h}_{j}^{(l)}$ 为邻居节点特征向量；$\boldsymbol{e}_{pq}^{(l)}$ 为边特征向量；$\mathcal{N}_i$ 为节点 i 的邻居节点集合。这里 $\boldsymbol{m}_{i}^{(l)}$ 不会被传入下一层；将 $\boldsymbol{m}_{i}^{(l,k)}$ 沿头部进行拼接，即可得

到最终输出 $\boldsymbol{h}_i^*$：

$$\boldsymbol{h}_i^* = \Big\|_{k=1}^{K}\Big(\Big\|_{l=1}^{L} m_i^{(l,k)}\Big) \tag{12}$$

2.2 监督推理

为了利用可靠的标签训练监督学习模型，我们采用分支定界法[6]离线求解小规模实例，从而获得最优分配作为训练目标。我们生成 S 个样本以构成训练集 $\mathcal{D}=\{(G_i,\boldsymbol{X}_i^*)\}_{i=1}^{S}$，其中 G_i 是二部图实例，$\boldsymbol{X}_i^*$ 是其精确分配。

网络输出 $\boldsymbol{H}^* \in \mathbb{R}^{B\times N\times d_{\text{out}}}$（经批次 B 重塑）用于通过基于批量矩阵乘法的相似度计算进行分配预测。我们提取无人机节点和目标节点特征 $\boldsymbol{H}^*{}_{\text{w}} \in \mathbb{R}^{B\times|X|\times d_{\text{out}}}$ 以及 $\boldsymbol{H}^*{}_{\text{t}} \in \mathbb{R}^{B\times|Y|\times d_{\text{out}}}$，并通过共享线性投影层生成查询矩阵：

$$\boldsymbol{Q}_{\text{W}} = \phi\left(\boldsymbol{h}_{\text{W}}\right) \in \mathbb{R}^{\text{B}\times\text{m}\times d_{\text{out}}} \tag{13}$$

式中：$\boldsymbol{Q}_{\text{W}}$ 为查询矩阵，由无人机节点特征生成；ϕ 为投影函数；$\boldsymbol{h}_{\text{W}}$ 为无人机节点特征矩阵；B 为批次大小；m 为无人机数量；d_{out} 为输出特征维度。并将它们投影到一个共享的线性层上，随后应用 LayerNorm 和 ReLU 来调整维度：

$$\boldsymbol{K}_{\text{T}} = \phi\left(\boldsymbol{h}_{\text{T}}\right) \in \mathbb{R}^{\text{B}\times\text{n}\times d_{\text{out}}} \tag{14}$$

式中：$\boldsymbol{K}_{\text{T}}$ 为键矩阵，由目标节点特征生成；$\boldsymbol{h}_{\text{T}}$ 为目标节点特征矩阵；B 为批次大小；n 为无人机数量；d_{out} 为输出特征维度。然后，我们通过批量矩阵乘法计算相似度张量：

$$\boldsymbol{S} = \boldsymbol{Q}_{\text{W}} \cdot \boldsymbol{K}_{\text{T}}^{\top} \in \mathbb{R}^{\text{B}\times\text{m}\times\text{n}} \tag{15}$$

式中：$\boldsymbol{S}$ 为无人机与目标之间的匹配得分矩阵；$\boldsymbol{Q}_{\text{W}}$ 为查询矩阵，由无人机节点特征生成；$\boldsymbol{K}_{\text{T}}$ 为键矩阵，由目标节点特征生成。

对目标维度进行 Softmax 运算后，即可得到每个无人机分配给不同目标的概率。我们采用直接构造策略，即将每个无人机分配给概率最高的目标，从而得到最终的分配矩阵 $\hat{\boldsymbol{X}}$。

节点特征提取器、边融合模块、图注意力块以及线性投影中的所有可学习参数均通过反向传播进行联合优化。学习目标是交叉熵损失：

$$L = -\frac{1}{\text{B}}\left[\sum_{b=1}^{\text{B}}\sum_{i=1}^{\text{N}}\sum_{j=1}^{\text{M}} X_{i,j}^{*(b)} \log(P_{i,j}^{(b)})\right] \tag{16}$$

式中：L 为损失函数值；$X_b^*(i,j)\in\{0,1\}$ 为第 b 个样本的真实标签；$P_{i,j}^{(b)}\in(0,1)$ 为经过 Softmax 运算后的预测分配概率。为提高收敛速度，采用了指数衰减的学习率。

2.3 强化学习推理

我们将目标分配过程建模为一个基于图的马尔可夫决策过程（MDP）$M = (S, \mathcal{A}, P, R, \gamma)$：

- S：状态空间，描述当前的分配图。
- $\mathcal{A}$：动作空间，即可选的无人机-目标分配动作。
- P：转移核，给出执行某个动作后进入下一个状态的概率。
- R：奖励函数，衡量某个动作的效用。
- γ：贴现因子，用于平衡未来奖励与即时奖励。

定义 1（状态空间 S）。在决策步骤 t 处，状态 $s_t \in S$ 是一个具有动态节点和边特征的加权二部图 $G_t = (X, Y, E_t, \boldsymbol{W}_t)$，该图编码了目标分配的当前进展。

定义 2（状态依赖的动作空间 $\mathcal{A}(s_t)$）。动作空间并非固定不变，而是一个状态依赖函数。在状态 s_t 下，有效动作集 $\mathcal{A}(s_t)$ 定义为当前剩余无人机集 $X^t{}_{\text{avail}}$ 到目标集 Y 的所有可行映射的集合：

$\mathcal{A}(s_t)=\{a=(i,j)\mid i\in X^{t}_{\text{avail}}, j\in Y\}\subset\mathcal{A}_{\text{full}}$，其中 $\mathcal{A}_{\text{full}}=X\times Y$ 是所有可能的无人机-目标对的完整空间。

定义 3（终态 s_T）。当所有无人机的可用性标志均为零时，即 $\sum_{i\in X} a_i^T=0$，状态 $s_T\in S$ 被定义为终态。此时，剩余无人机的活动集变为 $X^{T}_{\text{avail}}=\varnothing$，边集变为 $E_T=\varnothing$，有效动作集变为 $\mathcal{A}(s_T)=\varnothing$，且所有分配动作均已耗尽。一旦环境进入终态 s_T，当前回合立即结束。在终态下，Q 函数的评估直接简化为最后的即时奖励，无需对后续状态进行递归计算。

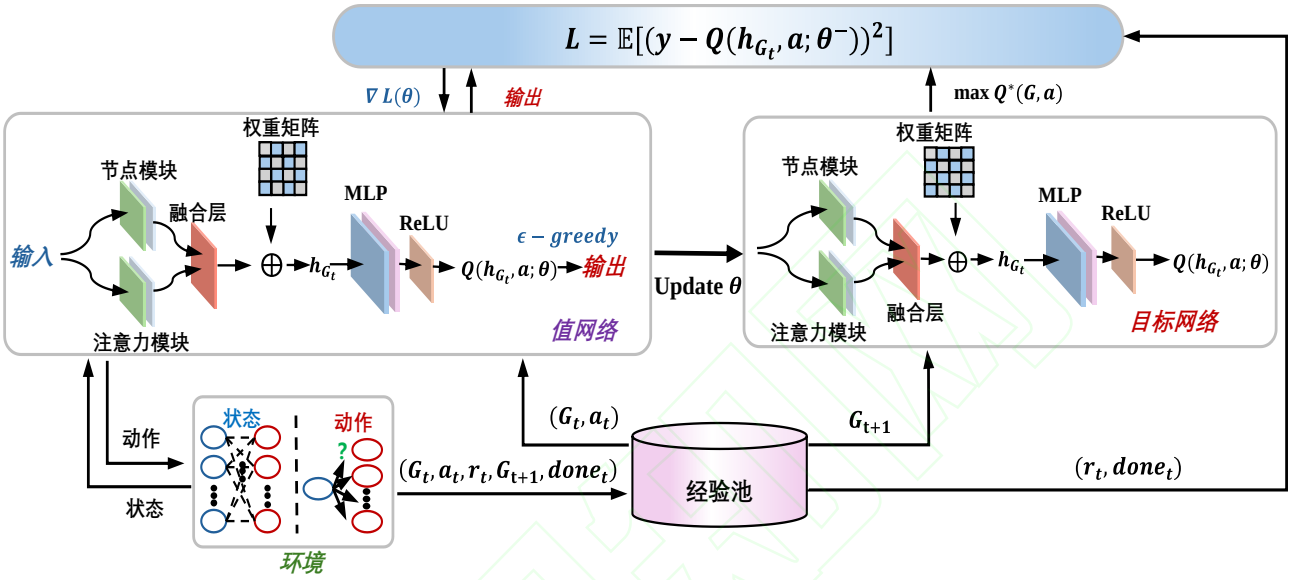

图 2　EGAT-DQN 的结构
Fig.2　EGAT-DQN architecture

图 2 展示了 EGAT-DQN 的架构。每个状态是一个编码当前任务分配情况的加权二分图；一个动作是选择一架无人机去攻击某个目标；而奖励则反映了击中该目标所获得的收益。通过学习最优策略，智能体能够最大化整个任务分配序列中的累计回报。

在 DQN 框架下，我们采用所提出的 EGAT 作为状态编码器。在每个时间步，将当前状态图输入 EGAT，以获取无人机和目标节点特征矩阵以及节点注意力权重，随后通过加权平均池化将这些信息融合，从而生成图嵌入向量 $\boldsymbol{h}_G$。展平后，将 $\boldsymbol{h}_G$ 输入全连接 DQN 头，针对所有可行动作 $a\in\mathcal{A}$ 输出 $Q(\boldsymbol{h}_G,a;\theta)$。智能体遵循 $\epsilon-$greedy 策略与环境交互，接收即时奖励 r_t、下一状态图 G_{t+1} 以及终结标志 done。即时奖励由目标值 v_j 和无人机的摧毁概率 p_{ij} 共同决定：

$$r_t=v_j\cdot p_{ij} \tag{17}$$

式中：r_t 为时刻 t 的即时奖励；v_j 为目标 j 的价值；p_{ij} 为无人机 i 对目标 j 的摧毁概率。在此分配场景中，强化学习代理在每个步骤中仅执行一个动作，即向目标 j 分配武器 i。该动作会产生一个即时奖励 r_t。随后，通过贝尔曼递归方程对最优动作价值函数进行近似。奖励函数定义为该即时奖励 $r_t=v_j\cdot p_{ij}$。若直接将目标函数的累积残差值用作奖励，奖励信号将变得稀疏且延迟，导致 DQN 训练收敛缓慢，并增加策略更新的方差，因此，改采用单步即时奖励以提供密集的学习信号。当所有无人机均已分配完毕时，最终回报即为该回合内所有即时奖励之和：

$$R=\sum_{t=0}^{N-1}\gamma^{t}r_t \tag{18}$$

式中：R 为一个回合的总回报；γ 为折扣因子；r_t 为即时奖励。

目标是学习一个策略 $\pi(a \mid s)$，该策略通过选择动作 a_t 来最大化贴现累计回报，即近似最优动作-价值函数：

$$Q^*(G,a)=\max_{\pi}\mathbb{E}\left[\begin{matrix}r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots\\ \Big| G_t=G,a_t=a,\pi\end{matrix}\right] \tag{19}$$

式中：$Q^*(G,a)$ 为在状态图 G 下执行动作 a 的最优动作价值；$\mathbb{E}$ 为数学期望；r_t 为即时奖励；γ 为折扣因子；G_t 为时刻 t 的状态图；a_t 为时刻 t 的动作；π 为策略。

以上内容得出了贝尔曼最优方程：

$$Q^*(G,a)=\mathbb{E}_G\left[r+\gamma\max_{a'}Q^*(G',a')\,\middle|\,G,a\right] \tag{20}$$

式中：$Q^*(G,a)$ 为在状态图 G 下执行动作 a 的最优动作价值；$\mathbb{E}_G$ 为当前状态 G 的期望；r 为即时奖励；γ 为折扣因子；a' 为下一动作；$Q^*(G',a')$ 为在下一状态图 G' 下执行动作 a' 的最优动作价值；G,a 为当前状态和动作。

由于状态空间庞大，我们使用深度 Q 网络 $Q(s,a;\theta)\approx Q^*(s,a)$ 来近似 $Q^*(s,a)$，并最小化贝尔曼残差：

$$L=\mathbb{E}\left[\left(y-Q(\boldsymbol{h}_{G_t},a;\theta^-)\right)^2\right] \tag{21}$$

其中 $y=r_t+\gamma(1-\mathrm{done}_t)\max_{a'}Q(\boldsymbol{h}_{G'},a';\theta^-)$，式中：$L$ 为损失函数值；$\mathbb{E}$ 为数学期望；y 为目标 Q 值；$Q(\boldsymbol{h}_{G_t},a;\theta^-)$ 为预测 Q 值；done_t 为结束标志；$\boldsymbol{h}_{G'}$ 为下一状态图嵌入向量；a' 为下一动作；θ^- 为目标网络参数。

$Q(G,a;\theta)$ 表示预测值，θ^- 表示目标网络的参数。在训练过程中，每个状态转换元组 $(G_t, a_t, r_t, G_{t+1},\mathrm{done})$ 会被存储在重放缓冲区 $\mathcal{D}$ 中。我们从 $\mathcal{D}$ 中采样小批量数据，分别使用当前的 EGAT–DQN 和目标网络计算预测 Q 值与目标 Q 值，并端到端地更新 EGAT 和 DQN 的参数。目标网络会定期与在线网络进行同步。EGAT-DQN 的整体训练过程总结于表 1 中，该算法概述了与环境的交互、重放缓冲区的更新以及利用 EGAT 编码对 Q 网络进行的优化。

表 1 EGAT-DQN的算法工作流程
Table 1　Algorithmic workflow of EGAT-DQN

步骤	算法描述
1	初始化：初始化 $Q(\cdot;\theta)$ 和目标网络，令 $\theta^{-} \leftarrow \theta$；初始化经验回放缓冲区 $\mathcal{D}$，容量为 $\lvert\mathcal{D}\rvert$；设置 γ、批大小 B、目标网络更新周期 C 以及 ε 衰减计划。
2	当训练未完成执行
3	构建当前状态图 G_t；用EGAT计算 $\boldsymbol{h}_G$。
4	$\epsilon-\text{greedy}$：以概率 ε 随机选择动作 $\epsilon, a_t \sim \text{Unif}(\mathcal{A}(G_t))$；否则 $a_t \leftarrow \arg\max_{a\in\mathcal{A}(G_t)} Q(\boldsymbol{h}_G, a;\theta)$。
5	衰减 $\epsilon \rightarrow \max(\epsilon_{\min}, \text{Decay}(\epsilon))$。
6	执行动作 a_t；观测奖励 r_t、下一状态图 G_{t+1} 以及结束标志 done；将 $(G_t, a_t, r_t, G_{t+1}, \text{done})$ 存入 $\mathcal{D}$。
7	如果 $\lvert\mathcal{D}\rvert \geq \text{B}$ 则
8	从 $\mathcal{D}$ 中采样 B 个转移样本。
9	用EGAT对 G_t 和 G_{t+1} 编码，得到 $\boldsymbol{h}_G$ 和 $\boldsymbol{h}_{G'}$。
10	$\hat{q} \leftarrow Q(\boldsymbol{h}_G, a_t;\theta); y \leftarrow r_t + \gamma(1-\text{done})\max_{a'} Q(\boldsymbol{h}_{G'}, a';\theta^{-})$
11	计算损失 $\mathcal{L} \leftarrow \text{MSE}(\hat{q}, y)$；反向传播并用 Adam 优化器更新 θ。
12	结束如果
13	如果当前步数对 C 取模等于0则 $\theta^{-} \leftarrow \theta$
14	结束如果
15	如果 done $=1$ 则 重置环境
16	结束如果
17	结束当

3　实　验

实验在 PyTorch 2.4.1 环境下进行，同时使用 PyTorch Geometric 2.6.1 和 Python 3.8.18。实验环境为 Ubuntu 18.04，运行于 Intel Xeon Platinum 8352Y CPU @ 2.20GHz。采用 Adam 优化器，并采用指数学习率衰减方案。不同模型所用的超参数配置汇总于表 2 中。

表 2 不同方法的超参数设置
Table 2　Hyperparameter settings for different methods

设置	GAT	GCN	单头 GAT	EGAT	EGAT-DQN
隐藏层维度	128	128	128	128	128
注意力头数	4, 4, 4	n/a	1, 1, 1	4, 4, 4	4, 4, 4
初始学习率	10^{-3}	10^{-3}	10^{-3}	10^{-3}	5×10^{-5}
训练轮数	200	200	200	200	3000
批次大小	64	64	64	64	32

3.1 评估指标

参照文献[12]，我们采用最优性差距（OG）、最优解比例（%Optimal）以及最优性差距的上界百分位数（% upper bound, u.b.）。

• 最优性差距。对于一个测试实例，设 X 表示在网络生成的分配下所有目标的残差值之和，设 X^{*} 表示在最优分配下的相应和。

• 平均差距。测试集中所有实例的平均最优性差距用于评估模型的整体解决方案质量。

• 最优率。指网络达到最优解（即 $X = X^*$）的测试实例所占比例（以百分比表示），表明模型达到最优解的能力。

• 上界率(u.b.)。该指标表示在给定百分位数下，最优性差距不超过特定值的实例所占比例，反映了模型在不同“最坏情况”场景下的表现。较小的上界值表明模型在大多数情况下能保持相对较小的最优性差距。本文通过分析多个百分位数下的上界值来表征其尾部鲁棒性。

3.2 结果与分析

同分布实验中，无人机数量与目标数量均设为 $M = N = 5$，并采用第 2 节中描述的相同建模约束。实验中目标值和摧毁概率分别从[25,100]和[0.6,0.9]范围内独立且均匀地采样。在此基础上，我们训练了监督式图神经网络模型 EGAT 和强化学习耦合模型 EGAT-DQN。在监督学习设置中，我们使用包含 100,000 个实例的训练集，并构建了规模分别为 1,000、5,000 和 10,000 的三个测试集用于评估。基线模型包括 GCN[16]、GAT[19]、单头 GAT[19]、传统 Quiz 问题启发式算法[10]，以及所提出的 EGAT 和 EGAT-DQN。如表 3 所示，在所有三种问题规模下，所提出的 EGAT 实现了最小的平均偏差（1%）

表 3 各方法求解三种规模数据集的性能比较

Table 3 Performance comparison of various methods for solving three-scale datasets

测试集规模	1000		5000		10000	
方法	时延/s	OG/% / %Optimal	时延/s	OG/% / %Optimal	时延/s	OG/% / %Optimal
GAT	2.12	2.66 / 62.5	4.28	2.74 / 62.26	5.90	2.89 / 62.23
单头 GAT	2.09	2.69 / 62.8	4.24	3.01 / 61.26	5.19	2.73 / 61.79
GCN	2.65	3.67 / 54	3.01	4.04 / 52.82	4.95	3.87 / 53.7
Quiz 启发式	0.21	30.5 / 24.7	1.22	27.4 / 29.28	2.32	29.79 / 31.07
EGAT	5.53	1.01 / 85.20	7.27	1.10 / 84.20	8.65	0.99 / 85.13
EGAT-DQN	1.26	9.59 / 0.7	3.57	9.70 / 0.6	4.58	9.65 / 0.8

和最高的最优率（85%），明显优于其他深度学习基线模型。EGAT-DQN 时延为 1.26 s，低于 EGAT 的 5.53 s，体现出较低时延优势；但其平均最优性差距为 9.59%，最优解比例为 0.70%，解质量明显低于 EGAT。由此可见，两种模型体现出不同的性能侧重：EGAT 更适合以解质量为主要目标的快速近似求解，EGAT-DQN 则通过顺序决策结构降低了推理开销，但伴随一定的解质量损失。

表 4 各方法在同一测试集规模的最优性差距上界百分位

Table 4 Upper bound percentile of the optimality gap among methods on the same test set scale

方法	50% u.b.	90% u.b.	95% u.b.	99% u.b.	100% u.b.
EGAT	0.00	1.12	6.73	21.82	56.62
GCN	0.00	9.92	14.86	26.10	46.38
GAT	0.00	11.17	16.75	29.32	53.50
单头 GAT	0.00	12.97	18.79	31.94	64.65
EGAT-DQN	9.48	15.63	17.43	20.67	25.82
Quiz 启发式	11.45	87.93	109.18	154.59	320.65

如表 4 所示，EGAT 在中位数和较高分位数处具有较低的最优性差距上界，表明其在大多数测试实例上能够保持较好的解质量。然而，在 99%和 100%分位数处，EGAT-DQN 的上界低于 EGAT，说明其在少量极端样本上的最大偏差受到一定限制。该现象可能与两种模型不同的训练范式有关。DQN 采用均方误差（MSE）优化时，对较大的时序差分（TD）误差给予更高惩罚，同时 ε-贪婪策略使智能体能够接触偏离当前最优轨迹的任务配置，从而有助于修正极端状态下的动作价值估计。相比之下，EGAT 主要基于最优分配标签进行监督学习，其整体解质量更高，但对训练数据中较少出现的异常分配状态覆盖相对有限。因此，两种方法在平均解质量与极端样本偏差方面表现出不同特点。

为了科学验证 EGAT 架构中每个创新组件的贡献，本节还介绍了系统性的消融实验。以 EGAT-

SL 作为基准模型，我们依次移除或替换其关键组件，并在标准测试集上评估了它们的性能。

表 5 消融实验：关键组件的有效性

Table 5　Ablation study: effectiveness of key components

变体	时延/s	OG/%	%Optimal
EGAT 基线	5.53	1.01	85.20
移除边特征	3.27	8.50	57.80
移除融合层	5.20	4.32	66.30
平均池化	4.56	5.10	60.43

结果如表 5 所示，EGAT 基线模型展现出最佳的整体性能，显著优于所有弱化版本。性能下降在移除边特征时最为明显，这表明边特征在建模决策相关性方面起着关键作用。移除节点-边融合层后，最优性 OG 值上升至 4.32%，这表明融合路径并非冗余组件，而是维持最终特征一致性并整合边信息的关键通道。将注意力机制替换为平均池化同样导致性能显著下降，这进一步证明差异化注意力权重机制比简单的均值聚合更适合处理异构的、对权重敏感的目标分配任务。

为了验证该算法在两个特定子场景中的有效性，构建了一个由 1,000 个特定实例组成的子场景测试集：(i) 高价值目标场景：目标值超过 80 的实例，共计 1000 个样本；(ii) 低摧毁概率场景：最小摧毁概率低于 0.65 的实例，共计 1000 个样本。我们针对这两个子场景评估了 EGAT 和基线模型 GAT 的性能，结果汇总于表 6。

表 6 特定子场景下的性能对比

Table 6　Performance comparison on special sub-scenarios

场景	模型	OG/%	%Optimal
完整测试集	EGAT	0.96	85.90
	GAT	2.68	63.66
高价值目标	EGAT	0.91	85.50
	GAT	2.58	64.12
低摧毁概率	EGAT	0.96	85.94
	GAT	2.69	63.65

如表 6 所示，在整个测试集及两个子场景中，EGAT 相对于最优解的性能差距保持在约 1%左右，且最优解率始终高于 85%。相比之下，GAT 在所有三个场景中均表现出超过 2.5%的性能差距，最优解率徘徊在 64%左右。结果证实，EGAT 在极端子场景中几乎没有性能下降，而 GAT 则始终表现不佳。

为确定 EGAT 的网络配置，对层数、隐藏维度和注意力头数进行敏感性分析。结果表明，层数由 1 增至 2、隐藏维度由 64 增至 128 以及注意力头数由 1 增至 4 时性能明显改善，继续增加模型规模则收益有限且计算开销增大。因此，后续实验采用 2 层 EGAT、128 维隐藏特征和 4 个注意力头。

为考察模型在训练设置之外的适用性，我们构建了跨规模测试实验、非对称 M≠N 场景和分布外测试实验。从问题规模、无人机与目标数量比例、图连接结构和目标价值分布 4 个方面考察方法在训练设置之外的适用性，并引入 Greedy 启发式[10]、遗传算法（GA）[7]和模拟退火算法（SA）[30]进行对比。

表 7 不同问题规模下各方法的求解性能

Table 7　Solution quality of different methods under different problem scales

方法	4×4	5×5	6×6	8×8
	OG/% / %Optimal	OG/% / %Optimal	OG/% / %Optimal	OG/% / %Optimal
EGAT	0.515 / 92.00	1.006 / 85.20	1.864 / 72.20	4.104 / 47.50
EGAT-DQN	8.41 / 3.40	9.42 / 0.90	10.57 / 0.10	11.85 / 0.00
贪婪算法	5.232 / 49.20	6.162 / 37.50	6.281 / 28.20	7.113 / 14.60
GA	2.039 / 72.24	3.917 / 49.58	5.040 / 33.88	6.722 / 15.64
SA	0.657 / 89.14	3.235 / 51.54	5.183 / 31.40	7.005 / 14.72

表 7 表明，在不重新训练的情况下，EGAT 能够直接处理 4×4～8×8 不同规模的二部图，并在各

测试规模下保持较好的相对解质量。随着测试规模逐渐偏离 5×5 训练条件，EGAT 的 OG 有所增大、%Optimal 有所下降，但在 8×8 场景下仍优于 Greedy、GA 和 SA。这说明基于参数共享的图消息传递结构赋予模型一定的跨规模适应能力，同时也表明该能力会随问题规模偏离训练分布而减弱。相比之下，EGAT-DQN 在规模增大时 OG 逐步上升，其主要价值仍体现在顺序决策建模，而非解质量优势。为考察节点数量比例变化对模型的影响，我们继续分析 M≠N 的非对称配置。

表 8 非对称无人机—目标数量配置下各方法的求解性能

Table 8 Solution quality of different methods under asymmetric UAV–target configurations

方法	5×6	5×8	6×5	8×5
	OG/% / %Optimal	OG/% / %Optimal	OG/% / %Optimal	OG/% / %Optimal
EGAT	15.612 / 28.40	17.987 / 8.00	6.450 / 22.10	9.413 / 4.80
EGAT-DQN	9.87 / 0.30	10.27 / 0.00	24.37 / 0.00	43.42 / 0.00
贪婪算法	3.500 / 38.40	2.039 / 37.60	6.111 / 36.10	9.143 / 26.70
GA	2.507 / 47.26	1.532 / 44.96	4.754 / 45.20	7.973 / 32.48
SA	2.270 / 48.52	1.533 / 43.64	4.939 / 42.92	8.780 / 27.66

如表 8 所示，与对称规模扩展相比，无人机与目标数量比例的变化对模型提出了更强的结构泛化要求。不同方法在两类非对称配置下表现出明显不同的敏感性。当无人机数量多于目标数量 M>N 时，EGAT 的性能相对稳定，其 OG 与 Greedy 较为接近，表明尽管测试图的节点比例偏离 5×5 训练条件，EGAT 仍能够保持一定的求解能力。相比之下，当目标数量多于无人机数量 M<N 时，EGAT 的 OG 上升较为明显，说明目标候选集合扩大后，节点比例变化对学习得到的匹配表示产生了更强影响。EGAT-DQN 对非对称结构表现出不同敏感性：在 5×6 和 5×8 配置下，其 OG 为 9.87%和 10.27%，低于 EGAT；而在 6×5 和 8×5 配置下升至 24.37%和 43.42%。总体上，EGAT 在 M>N 条件下更稳定，EGAT-DQN 在 M<N 条件下具有较小的平均最优性差距，差异可能与并行匹配和顺序决策机制有关。传统搜索方法在该类结构变化下仍保持较好的解质量，但需要针对每个实例执行在线搜索；EGAT 则无需针对测试实例重新优化即可直接生成分配结果。因此，表 8 进一步揭示了学习型快速推理方法与传统搜索方法在不同结构条件下的性能特点。为进一步考察另一类结构变化，下面分析无人机与目标之间可行连接减少的稀疏图场景。

表 9 稀疏连接条件下各方法的求解性能

Table 9 Solution quality of different methods under sparse connectivity

方法	连接密度 d=0.90	连接密度 d=0.75
	OG/% / %Optimal	OG/% / %Optimal
EGAT	9.997 / 72.60	23.937 / 53.50
贪婪算法	8.412 / 38.00	13.876 / 37.70
GA	3.651 / 53.18	3.004 / 64.58
SA	2.750 / 59.06	1.609 / 75.54

表 9 进一步考察完全二部图训练条件下模型对稀疏拓扑的适应性。在连接密度 d=0.90 时，EGAT 的最优解比例仍达到 72.60%，高于所比较的传统方法，表明模型对轻度拓扑扰动具有一定适应能力。随着连接密度继续降低，EGAT 的 OG 增大、%Optimal 下降，而 GA 和 SA 在强稀疏条件下表现更稳定。其原因在于稀疏连接同时改变节点度数、边邻域和注意力聚合结构，使测试图逐渐偏离完全二部图的训练分布；而传统搜索方法直接针对当前可行空间优化。由此可见，EGAT 能够适应一定程度的连接结构变化。

表 10 不同目标价值分布下各方法的求解性能

Table 10　Solution quality of different methods under different target-value distributions

方法	同分布区间	高价值区间	长尾价值分布
	OG/% / %Optimal	OG/% / %Optimal	OG/% / %Optimal
EGAT	1.006 / 85.20	1.134 / 82.20	5.774 / 52.80
贪婪算法	6.162 / 37.50	7.197 / 32.80	5.913 / 42.60
GA	3.917 / 49.58	4.324 / 46.64	3.908 / 56.16
SA	3.235 / 51.54	3.900 / 47.14	3.317 / 61.06

如表 10 所示，当目标价值由训练时的[25,100]扩展至高价值区间[100,200]时，OG 仅由 1.006%变化至 1.134%，%Optimal 仍达到 82.20%，且整体性能仍优于 Greedy、GA 和 SA，因此说明模型对保持分布形态不变的数值范围偏移具有较好的稳定性。在长尾价值分布下，由于目标函数按目标价值对剩余生存概率进行加权，长尾分布中的少量高价值目标会放大错误分配的影响导致模型性能有所下降，表明模型对分布形态变化更为敏感。

在相同的实验设置下，这两个模型均经过 200 个 epoch 的训练，并采用了早停机制以防止过拟合。结果显示，GAT 在约 50 个 epoch 时收敛，而 EGAT 则在约 80 个 epoch 时达到稳定状态。

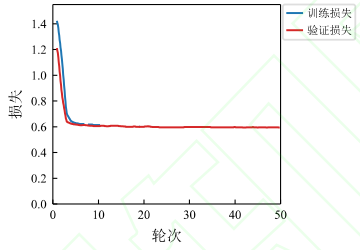

图 3　GAT 的损失曲线

Fig.3　Loss curves of GAT

图 3 展示了 GAT 的训练和验证损失。在前几个 epoch 中，损失值从 1.4 迅速降至 0.6，随后在第 10 个 epoch 之后趋于平稳，维持在 0.6 左右，这表明模型遇到了瓶颈，且在捕捉高阶图结构方面的能力有限。

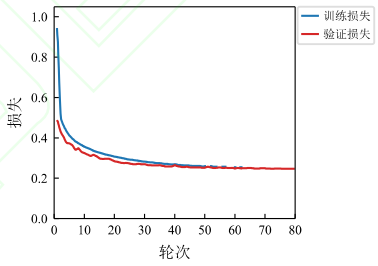

图 4　EGAT 的损失曲线

Fig.4　Loss curves of EGAT

图 4 展示了 EGAT 的训练和验证损失曲线。在相同条件下，EGAT 的初始损失约为 0.95，并在前 10 个 epoch 内迅速降至 0.4 以下。损失值持续平稳下降，并在大约 80 个 epoch 后收敛至 0.25 左右，这比 GAT 的最终损失低得多。此外，验证损失曲线始终与训练损失曲线保持高度一致，这不仅反映出更快的收敛速度，也体现了更强的泛化能力。综上所述，本次对比表明，EGAT 通过整合边特征，显著增强了图结构信息的表征能力。这带来了更快的收敛速度、更低的最终损失以及更高的稳定性，突显了边特征在提升模型性能方面所发挥的关键作用。

为了进一步评估所提方法的泛化能力，我们增加了七个不同规模的测试集。

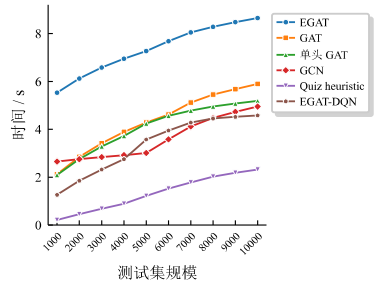

图 5　不同方法求解时间随测试集规模变化
Fig.5　Total running time of different methods versus test set size

如图 5 所示，所有方法的运行时间随测试集规模的增大而增加；在深度学习模型中，EGAT-DQN 始终呈现最低的增长曲线和最缓慢的增长速度，其运行时间低于 GAT、单头 GAT、GCN 和 EGAT。虽然启发式方法速度更快，但是它仅作为非学习型基线。

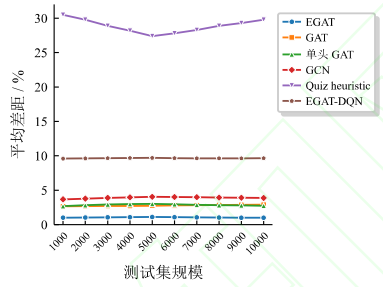

图 6　平均最优性差距随测试集规模变化
Fig.6　Mean optimality gap versus test set size

图 6 和图 7 共同凸显了 EGAT 在解的质量方面的优势。如图 6 所示，EGAT 在所有测试集规模下均能保持最小的平均偏差，这表明其具有强大的逼近最优解的能力。

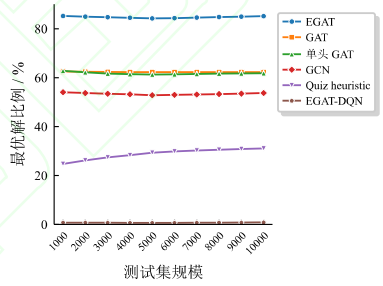

图 7　最优解比例随测试集规模变化
Fig.7　Percentage of optimal solutions versus test set size

相应地，图 7 显示 EGAT 获得了最高的最优解比例，明显超越了所有其他深度学习模型。此外，所有基于学习的方法都比启发式基线方法具有显著优势，后者既存在较大偏差，最优率也较低。这些结果突显了深度学习方法在提升解的质量方面的有效性，其中 EGAT 表现出了最佳的整体性能。

4　结论

本文针对无人机蜂群作战中的目标分配问题，提出边感知图注意力网络（EGAT）以及 EGAT-DQN 算法。

1）提出了一种边感知图注意力网络（EGAT），通过将边特征融入消息传递过程，EGAT 实现了最小的平均最优性差距和最高的最优率，表现优于传统启发式算法和标准 GNN 基线。

2）在此基础上，将 EGAT 与深度 Q 网络结合，进一步提出了 EGAT–DQN，以编码加权二分图状态并支持快速在线决策，在仿真实验中，EGAT-DQN 在第 99–100 百分位数处获得了最低的上界指

标，其 OG 和最优解比例明显弱于 EGAT，说明 EGAT-DQN 的主要作用在于顺序决策建模，而非提高解质量。

3）消融实验表明，移除边特征后 EGAT 的最优性差距从 1.01%急剧上升至 8.50%，最优解比例从 85.20%降至 57.80%；移除融合层或替换注意力机制为平均池化均导致性能显著下降。这说明边特征、融合路径和差异化注意力权重是 EGAT 性能提升的关键因素。

4）扩展实验表明，EGAT 在 4×4 至 8×8 的对称规模扩展中保持较好的相对解质量，在目标价值范围变化下性能变化较小。

尽管具备这些优势，本文框架仍存在一定局限：假设环境为静态，EGAT-DQN 虽然具有较低时延，但其解质量仍明显低于 EGAT，顺序决策模型的求解质量仍有进一步提升空间。未来将放宽静态环境这一假设以处理动态场景，整合跨模态线索，并将感知信息与先验知识融合，从而提升其在复杂作战环境中的适用性和整体适应能力。

参考文献（References）

[1] Alqudsi Y,Makaraci M.UAV swarms:research,challenges,and future directions[J].Journal of Engineering and Applied Science,2025,72(1):12.

[2] Manne A S.A target-assignment problem[J].Operations Research,1958,6(3):346-351.

[3] Lloyd S P,Witsenhausen H S.Weapons allocation is NP-complete[C]//Proceedings of the 1986 Summer Computer Simulation Conference.San Diego,CA:Society for Computer Simulation,1986:1054-1058.

[4] Kline A,Ahner D,Hill R.The weapon-target assignment problem[J].Computers & Operations Research,2019,105:226-236.

[5] Lee D,Shin M K,Choi H L.Weapon target assignment problem with interference constraints[C]//Proceedings of the AIAA Scitech 2020 Forum.Reston,VA:American Institute of Aeronautics and Astronautics,2020:AIAA 2020-0388.

[6] Andersen A C,Pavlikov K,Toffolo T A.Weapon-target assignment problem:exact and approximate solution algorithms[J].Annals of Operations Research,2022,312(2):581-606.

[7] Li X,Zhou D,Pan Q.Weapon-target assignment problem by multi-objective evolutionary algorithm based on decomposition[J].Complexity,2018,2018:8623051.

[8] He S,Yue S,Wang G,et al.Target assignment algorithm for joint air-defense operation based on spatial crowdsourcing mode[J].Electronics,2022,11(11):1779.

[9] Bai Z,Zhou H,Shi J,et al.A hybrid multi-objective evolutionary algorithm with high solving efficiency for UAV defense programming[J].Swarm and Evolutionary Computation,2024,87:101572.

[10] Ahuja R K,Kumar A,Jha K C,et al.Exact and heuristic algorithms for the weapon-target assignment problem[J].Operations Research,2007,55(6):1136-1146.

[11] Mekaway H I,Abd El-Wahab M S,Hashem M.Novel goal-based weapon target assignment doctrine[J].Journal of Aerospace Computing,Information,and Communication,2009,6(1):2-29.

[12] Gibbons D,Lim C C,Shi P.Deep learning for bipartite assignment problems[C]//Proceedings of the 2019 IEEE International Conference on Systems,Man,and Cybernetics.New York,NY:Institute of Electrical and Electronics Engineers,2019:2318-2325.

[13] Shokoohi M,Afsharchi M,Shah-Hoseini H.Dynamic distributed constraint optimization using multi-agent reinforcement learning[J].Soft Computing,2022,26(8):3601-3629.

[14] Scarselli F,Gori M,Tsoi A C,et al.The graph neural network model[J].IEEE Transactions on Neural Networks,2008,20(1):61-80.

[15] Gilmer J,Schoenholz S S,Riley P F,et al.Neural message passing for quantum chemistry[C]//Proceedings of the 34th International Conference on Machine Learning.Cambridge,MA:JMLR.org,2017:1263-1272.

[16] Jiang B,Zhang Z,Lin D,et al.Semi-supervised learning with graph learning-convolutional networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.Los Alamitos,CA:IEEE Computer Society,2019:11313-11320.

[17] Ying R,He R,Chen K,et al.Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.New York,NY:Association for Computing Machinery,2018:974-983.

[18] Yu B,Yin H,Zhu Z.Spatio-temporal graph convolutional networks:a deep learning framework for traffic forecasting[C]//Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.Stockholm,Sweden:AAAI Press,2018:3634-3640.

[19] Veličković P,Cucurull G,Casanova A,et al.Graph attention networks[C]//International Conference on Learning Representations.Vancouver,BC,Canada:OpenReview.net,2018.

[20] Boffa M,Houidi Z B,Krolikowski J,et al.Neural combinatorial optimization beyond the TSP:Existing architectures under-represent graph structure[C]//AAAI'22 GCLR 2022 workshop on Graphs and more Complex structures for Learning and Reasoning.Vancouver,BC,Canada:AAAI Press,2022.

[21] Langedal K,Manne F.Graph neural networks as ordering heuristics for parallel graph coloring[C]//Proceedings of the 2025 SIAM Symposium on Algorithm Engineering and Experiments.Philadelphia,PA:Society for Industrial and Applied Mathematics,2025:56-67.

[22] Hady M A,Hu S,Pratama M,et al.Multi-agent reinforcement learning for resources allocation optimization:a survey[J].Artificial Intelligence Review,2025,58(11):354.

[23] Tang J,Liang Y,Li K.Dynamic scene path planning of UAVs based on deep reinforcement learning[J].Drones,2024,8(2):60.

[24] Zhao J,Liu H,Sun J,et al.Deep reinforcement learning-based end-to-end control for UAV dynamic target tracking[J].Biomimetics,2022,7(4):197.

[25] 毕文豪，张梦琦，高飞，等．无人机集群任务分配技术研究综述[J]．系统工程与电子技术，2024，46（3）：922-934．
Bi W H,Zhang M Q,Gao F,et al.Review on UAV swarm task allocation technology[J].Systems Engineering and Electronics,2024,46(3):922-934(in Chinese).

[26] 张安，杨咪，毕文豪，等．基于多策略 GWO 算法的不确定环境下异构多无人机任务分配[J]．航空学报，2023，44（8）：327115．
Zhang A,Yang M,Bi W H,et al.Task allocation of heterogeneous multi-UAVs in uncertain environment based on multi-strategy integrated GWO[J].Acta Aeronautica et Astronautica Sinica,2023,44(8):327115(in Chinese).

[27] 王然然，魏文领，杨铭超，等．考虑协同航路规划的多无人机任务分配[J]．航空学报，2020，41（S2）：724234.
Wang R R,Wei W L,Yang M C,et al.Task allocation of multiple UAVs considering cooperative route planning[J].Acta Aeronautica et Astronautica Sinica,2020,41(S2):724234(in Chinese).
[28] Li T,Wang G,Fu Q,et al.An intelligent algorithm for solving weapon-target assignment problem:DDPG-DNPE algorithm[J].Computers,Materials & Continua,2023,76(3):3499-3522.
[29] Hu K,Pan H,Han C,et al.Graph neural network-enhanced multi-agent reinforcement learning for intelligent UAV confrontation[J].Aerospace,2025,12(8):687.
[30] Kirkpatrick S,Gelatt C D Jr,Vecchi M P.Optimization by simulated annealing[J].Science,1983,220(4598):671-680.