Modeling and Simulation 建模与仿真, 2025, 14(3), 694-715
Published Online March 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.143257

Hans汉斯

基于多特征融合的异构车辆路径规划问题

谭云洁，倪　静*

上海理工大学管理学院，上海

收稿日期：2025年2月26日；录用日期：2025年3月19日；发布日期：2025年3月28日

摘　要

异构车辆路径规划(HCVRP)是当前物流优化领域的核心难题，由于车辆容量与速度的差异化约束、客户需求的动态分布以及多目标优化的内在冲突。针对现有方法存在的特征融合效率、计算复杂度与多目标协同能力上的不足，本文提出了一种基于深度强化学习的多特征融合框架(MFF-HVPP)。通过马尔可夫决策过程(MDP)建模动态路径规划问题，构建包含车辆状态与节点状态的复合状态空间，并设计双模态奖励函数以适配min-max与min-sum目标。同时，构建多特征融合编码器，通过位置嵌入提取节点局部特征与空间依赖，并设计了Transformer通道特征扩展模块(TransCFE)，通过通道维度的特征增强与残差连接，解决传统注意力机制中的梯度消失与过拟合问题。分层解码策略中，MFF-HVPP将路径规划解耦为车辆选择与节点选择的序列决策过程，通过结合概率化抽样实现了全局优化与局部搜索的平衡。实验表明，在120客户节点的min-max场景中，MFF-HVPP最大行程时间误差率(Gap)低至1.31%，计算效率较传统方法提升98%；在min-sum任务中，总行程时间优化误差率仅为1.07%，并且支持百节点级场景的实时响应。本文研究为复杂约束下的多目标路径规划提供了可扩展的理论框架，并为智能物流系统的动态调度奠定了技术基础。

关键词

车辆路径规划，深度强化学习，多特征融合，通道特征扩展，分层解码器

Heterogeneous Vehicle Routing Problem Based on Multi-Feature Fusion

Yunjie Tan, Jing Ni*

Business School, University of Shanghai for Science and Technology, Shanghai

Received: Feb. 26th, 2025; accepted: Mar. 19th, 2025; published: Mar. 28th, 2025

Abstract

The heterogeneous vehicle routing problem (HCVRP) is a core challenge in the field of logistics

*通讯作者。

文章引用: 谭云洁, 倪静. 基于多特征融合的异构车辆路径规划问题[J]. 建模与仿真, 2025, 14(3): 694-715.
DOI: 10.12677/mos.2025.143257

optimization, due to the differentiated constraints of vehicle capacity and speed, dynamic distribution of customer demand, and inherent conflicts in multi-objective optimization. In response to the limitations of existing methods in feature fusion efficiency, computational complexity, and multi-objective coordination ability, this paper proposes a multi-feature fusion framework based on deep reinforcement learning (MFF-HVPP). By modeling the dynamic routing problem using a Markov Decision Process (MDP), we construct a composite state space that includes vehicle and node states, and design a dual-modal reward function to accommodate the min-max and min-sum objectives. At the same time, a multi-feature fusion encoder is developed, which extracts local node features and spatial dependencies through position embeddings. We also propose a Transformer channel feature extension module (TransCFE), which enhances features along the channel dimension and uses residual connections to address the issues of gradient vanishing and overfitting found in traditional attention mechanisms. In the hierarchical decoding strategy, MFF-HVPP decouples the routing decision process into vehicle selection and node selection as a sequential decision process, achieving a balance between global optimization and local search through probabilistic sampling. Experiments show that in a min-max scenario with 120 customer nodes, the MFF-HVPP achieves a maximum travel time gap of just 1.31%, with computational efficiency improved by 98% compared to traditional methods. In the min-sum task, the total travel time optimization gap is only 1.07%, and it supports real-time responses in scenarios with up to 100 nodes. This research provides a scalable theoretical framework for multi-objective routing under complex constraints and lays the technical foundation for dynamic scheduling in intelligent logistics systems.

Keywords

Vehicle Routing, Deep Reinforcement Learning, Multi-Feature Fusion, Channel Feature Expansion, Hierarchical Decoder

1. 引言

车辆路径规划问题(Vehicle Routing Problem, VRP)作为物流优化领域的核心优化目标，其目标是通过合理设计车辆行驶路径，在满足客户需求与资源约束的前提下，最小化运输成本并提升服务效率。随着电子商务与即时配送需求的快速增长，传统同构车队(Homogeneous Fleet) [1]-[3]的路径规划模型已难以应对当前现实场景中车辆类型多样、客户分布复杂、动态需求多变等挑战。在此背景下，异构车队车辆路径规划问题(Heterogeneous Fleet Capacitated VRP, HCVRP) [4]-[6]因其对车辆容量、速度等差异化的建模能力，成为当前研究热点。然而，异构车队车辆路径规划问题也面临着多维约束显著增加后需要在规划过程中动态平衡车辆负载与任务分配的问题，以及传统优化方法在处理大规模问题时存在计算效率低、泛化性差的问题。

近年来，深度强化学习(DRL)为路径规划提供了新思路，但其在异构车队场景中的应用仍存在相应的瓶颈：现有模型对车辆状态、节点位置以及用户需求等特征的融合能力有限，且缺乏对复杂约束的适应性。针对这些问题，本文提出一种基于多特征融合的异构车辆路径规划框架。通过构建马尔可夫决策过程(MDP)，模型将车辆路径规划分解为动态的状态更新与序列决策问题，并设计多特征融合编码器(Multi-Feature Fusion Encoder)与 Transformer 通道特征扩展模块(TransCFE)，实现节点位置、客户需求及车辆容量等特征的深度交互与增强表达。此外，分层解码器结构(车辆选择解码器与节点选择解码器)通过概率化

抽样策略，兼顾求解效率与全局优化能力。

实验结果表明，MFF-HVPP 在两类典型 HCVRP 任务(min-max 与 min-sum)中均显著优于现有方法。在 120 客户节点的 min-max 场景中，MFF-Sample1280 方案将最大行程时间的相对误差率(Gap)降至 1.31%，较传统启发式算法(如 SISR、ACO)优化幅度超 10%，且计算效率提升两个数量级(8.66 s vs. 1378 s)。对于 min-sum 目标，在 V5 车队配置下，MFF 方法总行程时间误差率仅为 1.07%，同时保持 13.77 s 的实时响应能力，验证了其在大规模物流调度中的实用价值。同时，本文首次将通道扩展机制引入路径规划领域，通过 TransCFE 模块实现多模态特征的深度融合，为复杂约束下的多目标路径优化研究提供了新的技术路径。

2. 相关工作

随着物流配送、共享出行以及自动驾驶等领域的快速发展，车辆路径规划(Vehicle Routing Problem, VRP)的应用场景变得更加广泛和复杂。传统的解决方法主要依赖于数学建模和启发式算法，这些方法虽然在早期研究中取得了显著成果，但在面对真实场景中的动态环境、复杂约束和大规模问题时，存在一定的局限性。近年来，随着机器学习和深度学习技术的快速发展，车辆路径规划问题得到了全新的解决思路，不仅能够从历史数据中学习复杂的路径规划策略，还能在贴近真实动态环境中实时调整路径方案，极大地提升了路径规划的效率和适应性。在实际日常应用中，车辆路径规划问题往往涉及多种异构特征，包含不同类型的车辆车队、多样化的客户需求、动态的时间窗约束以及复杂的路网结构等。这些异构特征的存在使得传统的单一特征建模方法难以全面获取问题的复杂性，导致最终的路径规划方案较差。因此，如何有效融合多特征信息，构建更加贴近实际场景的路径规划模型，成为当前研究的重要方向。

传统的车辆路径规划方法通常可分为精确算法、启发式算法和元启发式算法三类，每类方法适用于不同规模和复杂度的 VRP 问题。这些方法在早期的研究中占据主导地位，并为后续的智能优化算法奠定了基础。精确算法[7]通过数学建模和优化技术求解车辆路径规划问题，能够找到全局最优解，但计算复杂度高，适用于小规模问题。Pecin [8]等人提出的基于分支定价的高效算法能解决大规模 VRP 问题。然而，精确算法在处理动态环境或复杂约束时，仍面临实时性挑战。Tilk [9]等人提出的基于分支定价的算法在保证解的全局最优性的同时，计算时间随问题规模指数增长，难以应对大规模问题。在处理异构车队问题(HFVRP)时，Repoussis [10]等人提出了一种基于动态规划的精确算法，能够处理不同类型车辆的路径规划问题，但其计算复杂度较高，难以应用于大规模问题。启发式算法通过经验规则快速求解，适用于大规模问题。经典的节约算法[11]通过合并路径减少总成本。近年来，改进的自适应大邻域搜索[12](ALNS)在动态 VRP 中表现优秀。Braekers [13]等人提出了一种基于 ALNS 的 VRPTW 求解方法，通过动态调整邻域结构改进解的质量实现高效处理时间窗约束，但在处理动态环境或复杂约束时，难以满足实时性要求。元启发式算法则通过模拟自然现象或生物行为求解 VRP，适用于复杂和大规模问题。Arnold [14]等人提出了一种基于遗传算法的混合方法，能够高效处理时间窗约束，但其计算时间较长且容易陷入局部最优。

随着机器学习技术的迅速发展，基于数据驱动的车辆路径规划方法逐渐成为研究热点。这些方法通过从历史数据中学习规律，从而生成相应的路径方案，并在动态环境和复杂约束下表现优异。当前，基于机器学习的方法可分为监督学习和无监督学习，适用于不同应用场景。监督学习方法利用历史数据训练模型预测路径，能够从大量标注数据中学习复杂的路径规划策略。Kool [15]等人提出了一种基于注意力机制的神经网络模型，能够捕捉路径规划中的长距离依赖关系，但是依赖于大量标注数据，且对动态环境的适应性较差。Shahbazian [16]等人则提出了一种基于监督学习的 HFVRP 求解方法，能够从历史数据中学习不同类型车辆的路径规划策略，但其训练需要大量的计算资源。无监督学习方法适用于无需标

注数据的场景，通过聚类算法划分区域，从而优化路径规划问题。Renaud [17]等人提出了一种结合 K-means 聚类和启发式算法的混合方法，从而无需标注数据，但是该方法解的质量通常不如监督学习方法，且对聚类算法的选择较为敏感。H Ewbank [18]等人提出了一种基于无监督学习的 HFVRP 求解方法，该方法无需标注数据，但其解的质量较差。

与传统的机器学习方法相比，深度学习能够从高维数据中学习复杂的路径规划策略，并在动态环境和复杂约束条件下表现出色。基于深度学习的方法主要分为图神经网络(GNN)和序列到序列模型(Seq2Seq)，分别适用于不同的应用场景。图神经网络(GNN)能够有效处理图结构数据，在 VRP 中得到了广泛应用。Khalil [19]等人提出了一种基于 GNN 的路径规划方法，能够处理大规模路网数据。W Kool [19]等人提出了一种基于图注意力网络的 VRPTW 求解方法，能够有效捕捉路网中的局部和全局信息并处理复杂的图结构数据，但该方法训练时间较长且对图结构的表示能力有限。E Bahovska [20]等人提出了一种基于 GNN 的 HFVRP 求解方法，能够从图结构数据中学习不同类型车辆的路径规划策略，但其性能依赖于图结构的质量，难以处理动态变化的图结构。序列到序列模型(Seq2Seq)通过生成路径序列求解 VRP，能够处理动态环境下的路径规划问题。Vinyals [21]等人提出的指针网络(Pointer Network)是 Seq2Seq 在路径规划中的经典应用，能够生成序列化的路径方案。近年来，Seq2Seq 模型在时间窗车辆路径规划问题(VRPTW)中得到了改进。Nazari [22]等人提出了一种基于 Seq2Seq 的 VRPTW 求解方法，能够处理动态环境下的时间窗约束，但难以处理大规模问题。强化学习通过智能体与环境的交互学习最优策略，在车辆路径规划中表现出强大的潜力，尤其适用于动态和复杂环境。与传统的优化方法相比，强化学习能够从交互数据中学习复杂的路径规划策略，并在动态环境中实时调整路径方案。基于强化学习的方法主要分为经典强化学习、深度强化学习和多智能体强化学习三大类，分别适用于不同的应用场景。经典强化学习算法被广泛用于求解小规模 VRP 问题。Watkins & Dayan [23]等人最先提出的 Q-learning 算法通过更新动作值函数来学习最优策略，B Ghaddar [24]等人提出了一种基于 Q-learning 的动态 VRPTW 求解方法，能够快速适应动态环境，但其在处理高维状态空间时存在局限性，难以处理复杂的约束条件。Li [25]等人提出了一种基于 Q-learning 的 HFVRP 求解方法，能够处理不同类型车辆的路径规划问题，但该方法解受限于状态空间的设计。深度强化学习结合了深度学习和强化学习的优势，能够处理高维状态空间和复杂的路径规划问题。Baty [26]等人提出了一种基于 DQN 的 VRPTW 求解方法，能够从高维数据中学习复杂的路径规划策略，但其训练时间较长且对超参数设置较为敏感。Kool [27]等人提出了一种结合深度学习和动态规划的车辆路径规划方法(DPDP)，能够显著提高大规模问题的求解效率，但在处理极其复杂的约束条件时仍存在一定的挑战。Chuanbo Hua [4]等人提出了一种基于注意力的编码器架构，实现了每个车辆的特征嵌入。Dong Ho Lee 等人[28]提出了一种数据驱动的训练策略，实现了自主快速构建高质量的轨迹。但上述方法中的特征表达能力受限于单一模态输入，仅考虑节点位置或车辆容量特征，从而对复杂约束包括容量动态更新、路径连续性等特征的建模能力不足。

综上所述，异构车辆路径规划问题在实际应用中面临诸多挑战，尤其是在多种车辆类型、复杂路况和动态需求下，如何实现高效的路径规划。现有的路径规划方法如基于经典调度算法的方案，虽然能够在一定程度上优化路径，但通常未能充分考虑不同车辆的特性与实时变化的环境因素，导致优化效果有限。机器学习和深度学习方法通过数据驱动的方式提高了路径规划的性能，但其忽视了车辆的载重、速度、能耗等多重特征，同时在面对不确定性因素时，常常缺乏应对能力。针对这一问题，本文提出了一种基于多特征融合的异构车辆路径规划算法。该算法通过结合不同类型车辆的特性，能够更有效地适应动态变化的物流需求，提高路径规划的精度和效率。实验结果表明，本文提出的路径规划算法在实际场景下，尤其是在涉及不同类型异构车辆的复杂配送任务中，展现了优秀的处理性能。该算法能够平衡不同车辆的特性，实现车辆调度的最优化，进一步提升了物流系统的运作效率，并有效降低了运输成本。

3. 研究方法

异构车队的车辆路径规划问题描述了一个过程，即所有满载车辆从仓库出发，依次访问客户的位置以满足他们的需求，限制每个客户只能访问一次，以及单次行程的车辆装载量永远不能超过它的能力。本节主要介绍具有 min-max 和 min-sum 目标的异构车队的车辆路径规划问题的数学模型。$N+1$ 个节点(客户和仓库)表示为 $X=\left(x^i\right)_{i=0}^{N}$，节点 x^0 表示仓库，客户可以表示 $X'=X\setminus\{0\}$。每个客户节点 $x^i\in R^3$ 定义为 $\left\{\left(c^i,d^i\right)\right\}$，其中前者包含节点 x^i 的二维位置坐标，后者指其需求。具体来说，仓库 $x^0=R^2$ 被定义为包含位置坐标的 $\left\{c^0\right\}$。考虑了具有不同容量的异构车辆，这尊重了现实世界的情况。因此，设 $V=\left\{v^i\right\}_{i=1}^{K}$ 表示车辆的异构车队，其中每个元素 v^i 定义为 $\left\{Q^i\right\}$，即车辆 v^i 的容量。

3.1. 变量定义

数学模型中使用的变量定义如下：

- y_{ij}^k：如果车辆 v^j 直接从节点 x^i 行驶到节点 x^j，则为 1，否则为 0；
- $D\left(x^i,x^j\right)$：节点 x^i 和节点 x^j 之间的欧几里得距离；
- l_i^k：到本次访问为止，访问节点 i 的车辆 v^k 的路线上的累积需求；
- f：车辆的速度。

为简化起见，假设所有车辆都具有相同的速度 f，可以很容易地将其扩展为采用不同的值。MM-HCVRP 目标公式可以定义如下：

$$\min\max_{v^k\in V}\left(\sum_{x^i\in X}\sum_{x^i\in X}\frac{D\left(x^i,x^j\right)}{f}y_{ij}^k\right) \tag{1}$$

模型受以下六个约束：

$$\sum_{x^i\in X}y_{ij}^k=\sum_{x^i\in X}y_{ji}^k,v^k\in V,x^j\in X \tag{2}$$

$$\sum_{v^k\in X}\sum_{x^i\in X}y_{ji}^k=1,x^j\in X \tag{3}$$

$$l_j^k\ge l_i^k+d_j-Q^k\left(1-y_{ji}^k\right),v^k\in V,x^i,x^j \tag{4}$$

$$d_j\le l_i^k\le Q^k,v^k\in V,x^i\in X' \tag{5}$$

$$y_{ij}^k\in\{0,1\}\ y_{ii}^k=0,v^k\in V,x^i,x^j\in X \tag{6}$$

$$d^i\ge 0,x^i\in X \tag{7}$$

目标是最小化所有车辆的最大行程时间。确保车辆进入节点的次数等于它们离开节点的次数，其中可以多次访问仓库节点。同时，确保每个客户只被访问一次，消除了次级行程，并保证任何车辆的累计需求不能超过其容量。

MS-HCVRP 与 MM-HCVRP 共享相同的约束，而目标如下：

$$\min\sum_{v^k\in X}\sum_{x^i\in X}\sum_{x^j\in X}\frac{D\left(x^i,x^j\right)}{f}y_{ij}^k \tag{8}$$

其中 f^k 代表车辆 v^k 的速度，它可能因车辆不同而不同。因此，它实际上是在最小化整个异构车队的总行程时间。

3.2. 问题建模

HCVRP 问题可以描述为所有车辆从仓库出发，逐一访问各客户，满足所有客户其需求，并确保每个客户仅且只能被访问一次，且每辆车在任何时刻的载货总量不得超过其容量上限的限制。

具体而言，HCVRP 问题中假设存在 $n+1$ 个节点，包括若干客户和一个仓库，可以用一个集合 $X=\left\{x^i\right\}_{i=0}^{n}$ 所表示。其中仓库节点定义为 x^0，客户集定义为 $X'=X\setminus\left\{x^0\right\}$。每个节点表述为 $x^i\in R^3$，可以表示为一个二元组 $\left\{\left(s^i,\ d^i\right)\right\}$，其中 s^i 是节点 x^i 的二维坐标，d^i 则表示为该节点的需求量，对于仓库节点 x^0，则该节点的需求量为 0。为了更加贴近实际应用，本文考虑了具有不同容量的异构车辆，本文定义车辆集为 $V=\left\{v^i\right\}_{i=1}^{m}$，其中 v^i 为每辆车的最大载重，将其定义为 $\left\{\left(Q^i\right)\right\}$。

在 $n+1$ 个节点中，节点 x_i 和 x_j 之间的欧氏距离表示为 $D\left(x_i,x_j\right)$。定义一个二值 y_{ij}^v，当车辆 v 从客户 x_i 直接前往客户 x_j 时，y_{ij}^v 取值为 1，反之 y_{ij}^v 则取值，同时变量 l_{ij}^v 则表示车辆 v 在从客户 x_i 前往客户 x_j 后剩余的载货量。为了简化模型，假设所有车辆的行驶速度为常数 f，但该假设可根据实际情况扩展为不同车辆具有不同的速度。基于此假设，MM-HCVRP 问题可以通过以下数学模型进行描述：

$$\min\max_{v\in V}\left(\sum_{i\in x}\sum_{j\in x}\frac{D\left(x_i,x_j\right)}{f}y_{ij}^v\right) \tag{9}$$

该目标函数是求解车辆路径规划中，最小化所有车辆的最大行驶距离，以避免某些车辆的过度负荷，从而均衡每辆车的行驶距离。其中，$D\left(x_i,x_j\right)$ 表示从节点 x_i 到节点 x_j 的欧氏距离，而 f 表示车辆的速度，通过距离除以速度以实现最小化每辆车的行驶距离，来找到每辆车的最优路径，最终达到最小化最大行驶距离的目的。

$$\sum_{i\in X}y_{ij}^v=\sum_{k\in X}y_{jk}^v,v\in V,j\in X' \tag{10}$$

这一约束确保每个客户的访问是连续的，即如果车辆 v 到达了节点 x_j，它必须从该节点离开去访问下一个节点 x_k。该约束确保每条路径是封闭的，当车辆访问一个客户后会继续访问下一个客户，防止出现车辆仅访问某个客户但不离开的问题。

$$\sum_{v\in V}\sum_{j\in X}y_{ij}^v=1,\quad i\in X' \tag{11}$$

这一约束条件要求每个客户 x_i 只能被一个车辆访问一次。无论有多少车辆，所有的客户只能由某辆车访问一次，这保证了客户的唯一访问性。

$$\sum_{v\in V}\sum_{i\in X}l_{ij}^v-\sum_{v\in V}\sum_{k\in X}l_{jk}^v=d^j,v\in V,j\in X' \tag{12}$$

这个公式确保每个车辆在访问客户 x_j 时，可以满足该客户的需求 d_j。它通过约束车辆的进出流量，保证每个客户的需求被精确满足。

$$d_jy_{ij}^v\le l_{ii}^v\le\left(Q^v-d^i\right)\cdot y_{ij}^v,v^k\in V,i\in X,j\in X \tag{13}$$

这条约束用于保证车辆在到达下一个客户 x_j 时，所携带的货物不会超过车辆的容量。Q^v 是车辆 v 的最大容量，d^i 是客户 x_i 的需求量。这个公式确保车辆在访问每个客户时，其剩余的载重满足车辆容量的限制。

$$y_{ij}^v\in\left\{0,1\right\},v\in V,i\in X,j\in X \tag{14}$$

该约束规定 y_{ij}^v 为二值变量，即当车辆 v 从节点 x_i 直接前往节点 x_j 时，变量值为 1，表示选择了这条

路径；否则为 0，表示未选择这条路径。

由于 HCVRP 的路径规划问题本质上也是一个序列决策问题，因此本文将其构建成为一个马尔可夫决策过程(MDP)。在此框架中，明确定义了状态空间、动作空间、状态转移函数以及奖励函数，以系统化解决车辆路径规划问题。

在构建的 MDP 模型中，状态 s_t 包含两个部分：车辆状态 V_t 和节点状态 X_t。其中，车辆状态 V_t 描述了每辆车的当前状态，包括其剩余容量 o_t、已累积的行驶时间 T_t 以及已行驶的路径 G_t。其中，车辆的容量反映其剩余的承载能力，行驶时间则记录车辆自出发以来的总耗时，行驶路径包含了车辆已经访问过的节点序列。节点状态 X_t 则反映每个客户节点的位置信息 s_i 以及其需求量 d_i。在路径规划中，车辆的任务是在考虑容量约束的情况下满足所有客户的需求。

车辆状态根据动作 $at=\left(v_i,x_j\right)$ 进行更新，更新具体流程为

1) 如果车辆 k 选择了节点 j，其容量将减少该客户的需求量 d_j，否则容量保持不变。剩余容量 o_{t+1}^k 更新公式：

$$o_{t+1}^k=\begin{cases}o_t^k-d_j,\text{if} \quad k=i\\ o_t^k, \qquad \text{otherwise}\end{cases} \tag{15}$$

2) $D\left(g_t^k,x_j\right)$ 表示车辆从最后访问的节点 g_t^k 到节点 x_j 的欧式距离，f 为车辆速度，记录了车辆累积的行驶时间，行驶时间 T_{t+1}^k 的更新为：

$$T_{t+1}^k=\begin{cases}T_t^k+\dfrac{D\left(g_t^k,x_j\right)}{f}, \text{ if} \quad k=i\\ T_t^k, \qquad\qquad \text{otherwise}\end{cases} \tag{16}$$

3) 当车辆 k 访问节点 j 时，节点 j 将被加入车辆 k 的访问路径中。行驶路径 G_{t+1}^k 的更新为：

$$G_{t+1}^k\begin{cases}G_t^k\cup\left\{x_j\right\}, \text{ if} \quad k=i\\ G_t^k, \qquad\qquad \text{otherwise}\end{cases} \tag{17}$$

同时，节点状态 X_t 的更新则涉及客户需求的变化。每个节点的需求将在被访问后更新为 0：

$$d_{t+1}^l\begin{cases}0, \quad \text{if} \quad k=i\\ d_t^l, \text{ otherwise}\end{cases} \tag{18}$$

奖励函数：

为了最小化所有车辆的最大行驶时间，奖励函数 R 被定义为该最大值的负值，即：

$$R=-\max_{v\in V}\left(\sum_{t=0}^{T}r_t\right) \tag{19}$$

其中，r_t 是每一步操作对应的增量行驶时间。

该图展示了本文的策略网络的结构。首先，将节点特征、车辆特征和位置特征输入到相应的嵌入模块中进行处理。节点特征通过卷积神经网络(CNN Embedding)进行嵌入，车辆特征直接输入，位置特征则通过位置嵌入(Positional Embedding)模块进行处理。然后，这些特征被整合到多特征融合编码器(Multi-Feature Fusion Encoder)中，形成统一的特征表示。

在路径构建过程中，策略网络首先通过车辆选择解码器(Vehicle Selection Decoder)选择一个合适的车辆 v_t^i，然后通过节点选择解码器(Node Selection Decoder)选择一个节点 x_t^j，即车辆在当前时间步 t 应访问的客户节点。每个时间步 t 的操作 a_t 由选择的车辆 v_t^i 和节点 x_t^j 组成，即 $a_t=\left(v_t^i,x_t^j\right)$。这一步骤完成后，

部分解(Partial Solution)和状态 s_t 将根据选择的操作进行更新。

在这个策略网络中，编码器负责对节点特征、车辆特征和位置特征进行一次性处理，而车辆选择解码器和节点选择解码器则在路径构建过程中多次执行，直至构建出完整的路径解决方案。每一步的选择都基于当前的部分解和全局的路径规划状态进行优化调整。

这种架构的设计目的是实现复杂路径规划问题的高效求解。通过独立的车辆选择解码器和节点选择解码器，网络能够灵活应对异构车辆和多样化客户需求的场景。车辆选择解码器根据车辆的状态(如剩余载重、行驶路径等)来优化车辆的调度，避免车辆过载或未充分利用的情况。同时，节点选择解码器专注于客户节点的选择，考虑因素包括节点的需求、位置等，确保每一步的路径选择能够最大化满足需求(图1)。

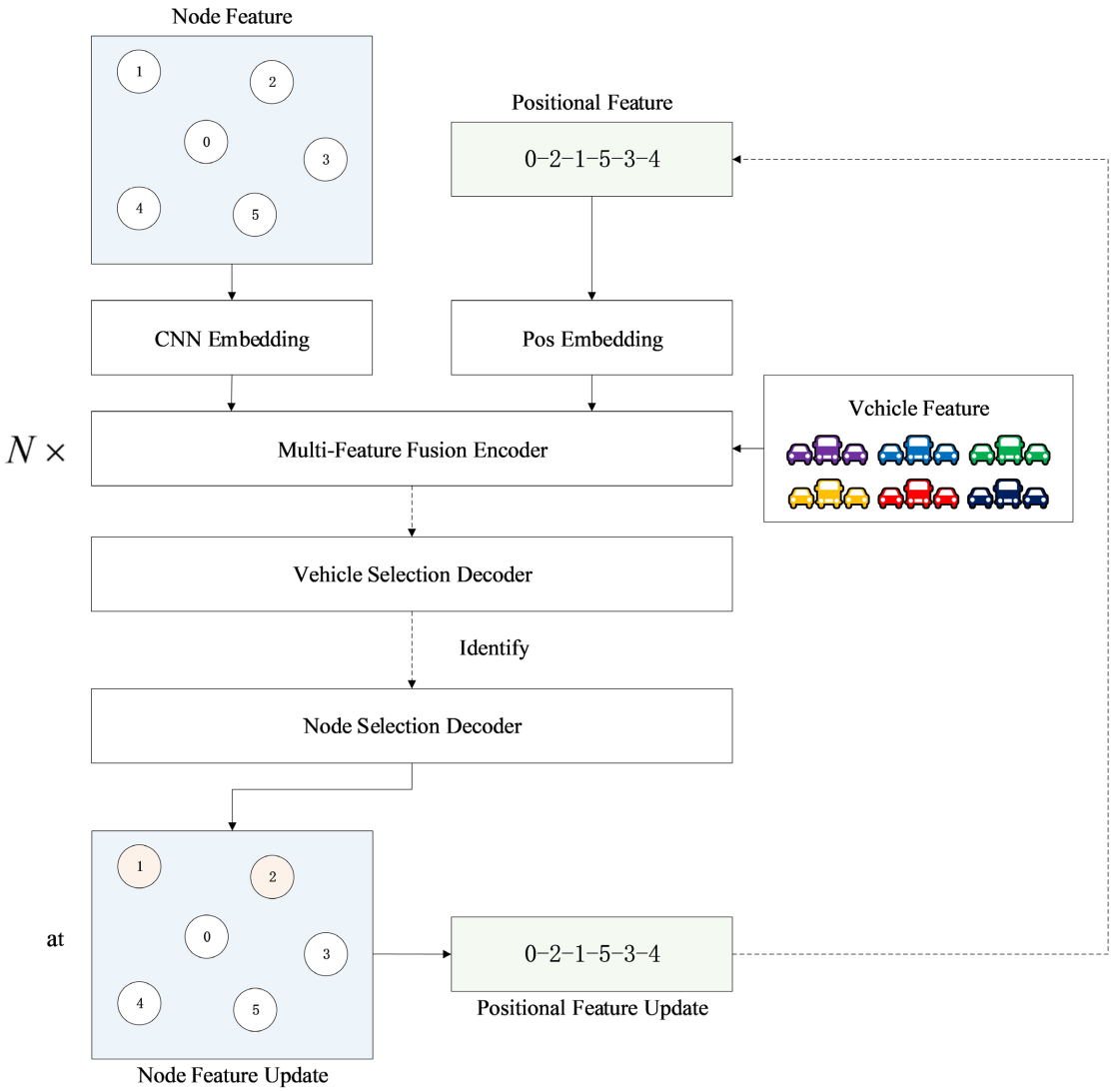

Figure 1. Network structure overview
图 1. 网络结构概图

此外，这种设计通过将路径构建问题分解为车辆选择和节点选择两个独立但相互关联的子问题，减少了决策复杂度，提升了策略网络的泛化能力。在这个框架中，编码器可以看作是全局信息的提取模块，

而解码器则根据不同的状态信息做出针对性的局部决策。这种分层决策机制使得该方法在处理复杂路径规划问题时能够保持较高泛化能力。

3.3. 多特征融合编码器(Multi-Feature Fusion Encoder)

编码器将问题实例的原始特征(即客户位置、客户需求和车辆容量)嵌入到更高维空间中，然后通过注意力层对其进行处理，以更好地进行特征提取(图 2)。

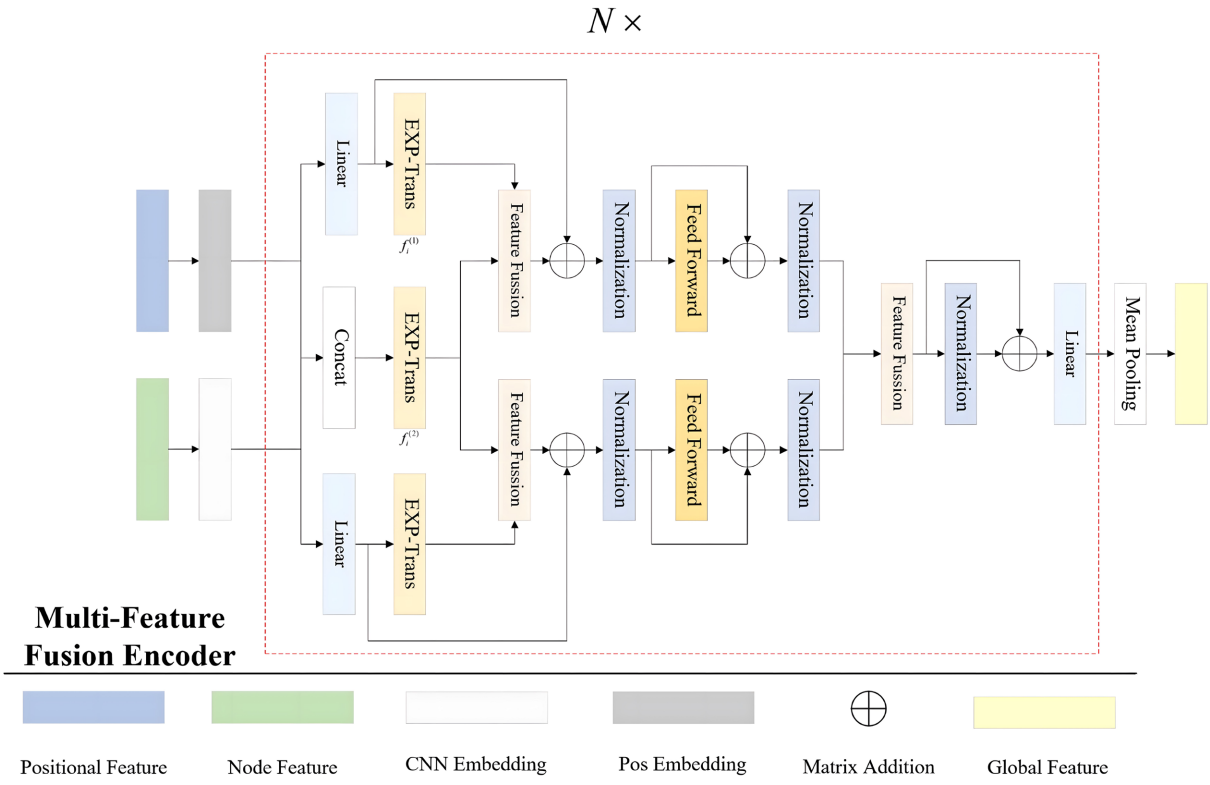

Figure 2. Multi feature fusion encoder
图 2. 多特征融合编码器

3.3.1. 特征处理

对于网络输入的位置特征(Positional Feature) $P_i \in R^2$ 通过位置编码器(Positional Embedding)进行处理，将位置特征映射到高维特征空间，生成每个节点的位置特征嵌入。如公式(20)所示，其中 $h_{\text{pos}_i} \in R^d$，d 为位置特征嵌入后的特征维度。

$$h_{\text{pos}_i} = \text{PE}\left(P_i\right) \tag{20}$$

然后，位置嵌入特征 h_{pos_i} 经过线性层 Linear_1，转换为适合后续处理的特征 h'_{pos_i}，具体表现为公式(21)，其中，线性变换的权重矩阵为 $W_{\text{pos}} \in R^{d\times d}$，偏置项设为 b_{pos}。

$$h'_{\text{pos}_i} = W_{\text{pos}} h_{\text{pos}_i} + b_{\text{pos}} \tag{21}$$

节点特征(Node Feature)则通过卷积神经网络(CNN Embedding)进行编码处理。对于每一个节点的特征 X_i，通过卷积网络提取节点特征的局部信息，特征提取后得到节点特征的高维表示 $h_{\text{node}_i} \in R^d$。

$$h_{\text{node}_i} = \text{CNN}\left(X_i\right) \tag{22}$$

然后，节点嵌入特征 h_{node_i} 经过线性层 Linear_2，转换为适合后续处理的特征 h'_{node_i}，具体实现细节见表征式(23)，其中，线性变换的权重矩阵为 $W_{\text{node}} \in R^{d\times d}$，偏置项设为 b_{node}。

$$h'_{\text{node}_i} = W_{\text{node}} h_{\text{node}} + b_{\text{node}} \tag{23}$$

位置特征和节点特征分别经过各自的编码器处理后，输出的位置嵌入特征 h_{pos_i} 和节点嵌入特征 h_{node_i} 通过 Concat_1 拼接操作整合在一起得到特征 h_{concat}，以便后续阶段的特征融合和路径决策。拼接后的特征表示为：

$$h_{\text{concat}} = \text{concat}_1\left(h_{\text{pos}_i} \middle| h_{\text{node}_i}\right) \tag{24}$$

位置嵌入特征 h_{pos_i} 和节点嵌入特征 h_{node_i} 通过线性层得到特征 h'_{pos_i} 和 h'_{node_i}，以确保特征能够在同一维度上进行交互和计算。传统的车辆路径规划方法在处理异构车队和复杂路径时，通常面临两个主要缺点：局部依赖性和顺序处理效率低下。这些方法难以捕捉节点之间的全局依赖关系，无法同时高效处理车辆和节点的多维度特征，尤其是在节点数量庞大或路径复杂时，容易导致计算效率低且结果不佳。因此本文基于 transformer 架构基础上提出了一种 Transformer 通道特征扩展模块(Transformer channel feature extension)，该模块采用偏移注意力机制，针对车辆路径规划过程中局部特征学习时可能出现的特征丢失、梯度消失及过拟合问题，通过设计通道特征扩展结构，在通道维度上实现特征增强，同时可以并行处理所有节点，捕捉全局的节点间关系，从而在异构车队路径规划问题中提供更灵活、更准确的决策。具体实现细节见表征式(24)。

3.3.2. 通道特征扩充(TransCFE)模块

为了进一步优化特征的提取与融合，设计并应用了 TransCFE 模块，用于在不同特征维度上进行通道扩充和优化。具体来说，TransCFE 模块分别作用于位置特征、节点特征以及连接特征，确保在各个特征层次上都能够充分利用通道维度的表达能力，具体公式如下(图 3)：

位置特征的通道特征扩充：

$$h_{\text{pos-Trans}} = \text{TransCFE}\left(h'_{\text{pos}_i}\right) \tag{25}$$

其中，h'_{pos_i} 表示第 i 个位置特征的初始编码。通过 TransCFE 模块对其进行通道维度的扩展和聚合操作，生成增强后的位置特征 $h_{\text{pos-Trans}}$。这一过程能够在位置特征层次上聚合更多的空间信息，有效提升模型对位置信息的表达能力。

节点特征的通道特征扩充：

$$h_{\text{node-Trans}} = \text{TransCFE}\left(h'_{\text{node}_i}\right) \tag{26}$$

对于路径规划任务中的每个节点特征 h'_{node_i}，本文通过 TransCFE 模块进行通道扩展和特征优化，从而生成优化后的节点特征 $h_{\text{node-Trans}}$。这一过程通过对节点特征的通道维度进行深度融合，提升了模型对不同节点之间复杂关系的理解能力，有助于提高路径规划的精度。

特征融合后的通道特征扩充：

$$h_{\text{concat-Trans}} = \text{TransCFE}\left(h_{\text{concat}}\right) \tag{27}$$

h_{concat} 是多种特征(如位置特征与节点特征)进行融合后的结果。通过将融合特征输入到 TransCFE 模块中，本文能够在通道维度上进一步提升特征的表达能力，得到优化后的融合特征 $h_{\text{concat-Trans}}$。此过程使得多源信息能够在统一的特征空间中进行有效融合与调整，进而提升模型的整体性能。

通过对位置特征、节点特征和融合特征分别进行 TransCFE 处理，本文确保了特征在不同层次上的表

达能力得以充分发挥。在路径规划过程中，TransCFE 模块的引入不仅改善了特征丢失问题，还显著提高了模型对复杂场景的适应能力。该模块应用于 MFF-HVPP 中的特征提取与融合阶段，分别集成到特征的聚合和更新过程中。其目的是通过通道维度的优化提升特征表达的多样性，从而提高模型的泛化能力和路径规划的精度。TransCFE 模块的详细实现过程如下：

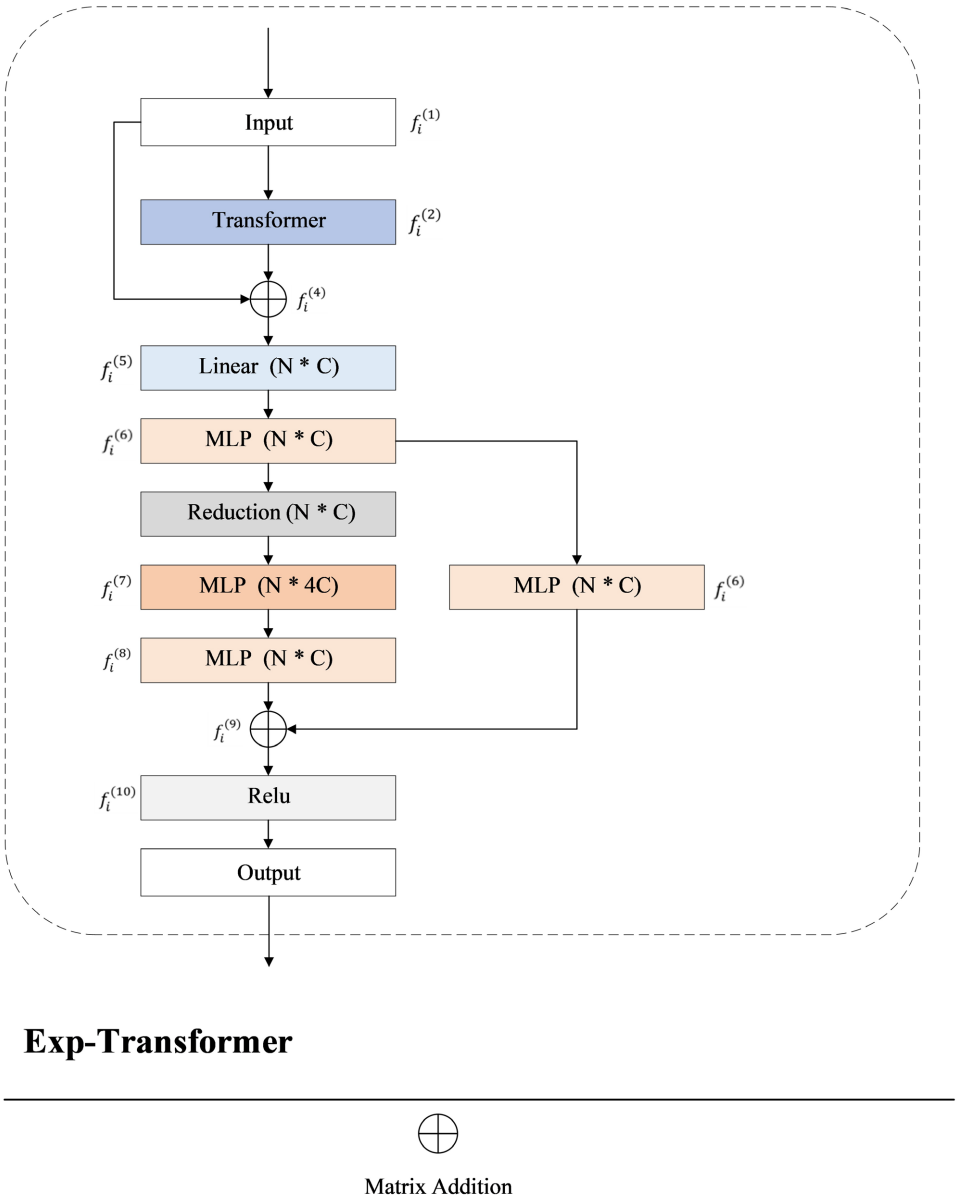

Figure 3. Transformer channel feature expansion module
图 3. Transformer 通道特征扩充模块

(1) 在车辆路径规划中，由于数据特征的多样性、复杂性，以及在不同道路和环境下的高度变化性，直接采用全局注意力机制会带来计算负担。为此，本文设计了 Transformer 通道注意力机制，旨在降低计算成本的同时提升特征聚合效果。具体而言，该机制通过在输入特征的通道维度上进行局部邻域特征聚合，获得增强的特征表示。其主要步骤包括：

对输入特征 $f_i^{(1)}$ 进行通道维度的聚合，应用线性层 Linear_1 和归一化层 Batchnorm，以逐点的方式进行特征变换和归一化处理，从而得到增强后的特征 $f_i^{(3)}$。相关计算公式见公式(28)和(29)：

$$f_i^{(2)} = \text{Transformer}\left(f_i^{(1)}\right) \tag{28}$$

$$f_i^{(3)} = \text{Batchnorm}\left(\text{Linear}_1\left(f_i^{(2)}\right)\right) \tag{29}$$

这种方式不仅保留了局部特征的完整性，同时避免了全局注意力的高计算成本。

(2) 特征变换。为了进一步降低梯度消失的风险，并提升特征学习的效果，采用了残差连接机制。具体做法是将学习到的特征 $f_i^{(3)}$ 与输入特征 $f_i^{(1)}$ 进行残差连接，以确保梯度的稳定性。然后，通过线性层 Linear2 对特征进行逐点变换，生成新的特征 $f_i^{(4)}$，具体实现细节见表征式(30)。

$$f_i^{(4)} = \text{Linear}_2\left(f_i^{(1)} + f_i^{(3)}\right) \tag{30}$$

在此基础上，使用层感知机 MLP 对特征进行进一步的几何和局部信息聚合，从而得到特征 $f_i^{(5)}$。MLP 的输入和输出维度均为 4C，具体实现细节见表征式(31)。

$$f_i^{(5)} = \text{MLP}\left(f_i^{(4)}\right) \tag{31}$$

(3) 通道维度特征扩充与调整。在车辆路径规划的特征聚合过程中，引入通道维度上的特征扩充。具体做法是通过 Reduction 模块对特征进行扩充，将特征 $f_i^{(5)}$ 复制 4 次，得到扩展后的特征 $f_i^{(6)}$。然后，通过 MLP 对这些扩展后的特征进行逐点变换，丰富特征的表达能力，具体实现细节见表征式(32)。此时，多层感知机 MLP 输入输出设置为 N*4C。

$$f_i^{(6)} = \text{MLP}\left(\text{Reduction}_1\left(f_i^{(5)}\right)\right) \tag{32}$$

特征 $f_i^{(6)}$ 再次通过多层感知机 MLP 聚合几何信息实现通道维度上的调整，得到特征 $f_i^{(7)}$，具体实现细节见表征式(33)。通过上述操作后，其输出被调整为 N*C。

$$f_i^{(7)} = \text{MLP}\left(f_i^{(6)}\right) \tag{33}$$

这一过程显著增强了逐点特征提取的能力，并改善了特征在不同采样阶段的传递效果。

(4) 残差连接与激活。为了缓解网络深度增加所带来的梯度消失问题，本文在输入和输出之间加入了残差连接，并通过 ReLU 激活函数对输出特征进行非线性变换，生成最终的特征 $f_i^{(8)}$。这种设计既能保持特征的原始信息，又能提升网络的非线性拟合能力具体实现细节见表征式(34)。

$$f_i^{(8)} = \text{Relu}\left(f_i^{(7)} + \text{MLP}\left(f_i^{(5)}\right)\right) \tag{34}$$

通过上述 TransCFE 模块的设计与应用，本文在通道维度上实现了特征的高效扩充，显著提升了异构车辆路径规划问题中特征提取与融合的效果。该模块不仅能缓解梯度消失问题，还能增强特征的表达能力，使得模型在大规模调度任务中表现更加出色。此外，通过通道特征的逐点变换与扩展操作，TransCFE 模块在优化模型的泛化能力与稳定性方面发挥了重要作用。

3.3.3. 特征融合

为了进一步增强空间位置信息在整体特征中的重要性，通过 FutrueFussion 模块融合位置特征和融合特征。初步的特征融合可以捕捉节点特征与位置信息的基本关联，但再次通过与位置特征融合可以增强空间信息的影响力，确保模型能够更准确地理解节点在不同位置上的表现，从而提高空间节点特征的学习效果。

$$h_{\text{fusion1}} = \text{FutrueFussion}\left(h_{\text{pos-Trans}} \middle| h_{\text{concat-Trans}}\right) \tag{35}$$

同时，为了进一步增强节点特征和位置两者之间的互补关系，通过 FutrueFussion 模块融合节点特征和融合特征。通过初步的特征融合，模型可以捕捉到节点的空间位置信息与其特性之间的关联性，而再次与节点特征融合可以使模型更好地整合这些信息，提升节点本身的表示能力。具体实现细节见表征式(36)

$$h_{\text{fusion2}} = \text{FutrueFussion}\left(h_{\text{node-Trans}} \middle| h_{\text{concat-Trans}}\right) \tag{36}$$

当网络更深时，为了缓解梯度消失的问题，在输入和输出之间添加一个残差 网络连接以弥补特征聚合过程中的损失以及防止过拟合。具体实现细节见表征式(37)。

$$h'_{\text{fusion1}} = h_{\text{fusion1}} + h_{\text{pos}_i},\ h'_{\text{fusion2}} = h_{\text{fusion2}} + h_{\text{node}_i} \tag{37}$$

为了减少不同模态特征之间数值尺度的差异，首先对融合后的特征进行 LayerNorm 归一化处理。特征 h'_{fusion1} 和 h'_{fusion2} 的特征归一化的具体实现细节见表征式(38)。归一化操作能够平衡特征的数值差异，使得后续的训练过程更加稳定。

$$h_{\text{norm1}} = \text{LayerNorm}\left(h'_{\text{fusion1}}\right),\ h_{\text{norm2}} = \text{LayerNorm}\left(h'_{\text{fusion2}}\right) \tag{38}$$

归一化后的特征经过前馈神经网络(Feed Forward Network, FFN)处理，以进一步提取深层次的特征。前馈神经网络通常包含 Linear 全连接层和 Relu 非线性激活函数。对于不同的特征 h_{norm1} 和 h_{norm2} 分别进行处理，具体实现细节见表征式(39)。这种网络结构能够对特征进行更深层次的特征提取，确保网络模型能够捕获到更高层次的特性信息。

$$h_{ff1} = \text{Linear}_3\left(\text{Relu}\left(h_{\text{norm1}}\right)\right),\ h_{ff2} = \text{Linear}_4\left(\text{Relu}\left(h_{\text{norm2}}\right)\right) \tag{39}$$

对于经过前馈神经网络处理后的特征通过加一个残差网络连接以弥补特征提取过程中的损失以及防止过拟合，同时通过 LayerNorm 归一化进行处理，平衡特征的数值差异，得到特征 h_{norm3} 和 h_{norm4}，具体实现细节见表征式(40)。

$$h_{\text{norm3}} = \text{LayerNorm}\left(h_{\text{norm1}} + h_{ff1}\right),\ h_{\text{norm4}} = \text{LayerNorm}\left(h_{\text{norm2}} + h_{ff2}\right) \tag{40}$$

节点特征和位置特征经过多层特征提取后通过 FutrueFussion 模块进行再融合，其目的是为了在各自的特征空间中充分提取出有用的高层次信息，并增强节点和位置信息的表达能力。同时，确保节点特征和位置特征在融合之前，两类特征能够保持各自的独立性，同时能够更好地捕捉更复杂的交互关系，

$$h_{\text{fusion3}} = \text{FutrueFussion}\left(h_{\text{norm1}} \middle| h_{\text{norm2}}\right) \tag{41}$$

同时，对于多特征融合处理的特征 h_{fusion3} 通过 LayerNorm 归一化进行处理，平衡特征的数值差异，同时通过一个残差网络连接以弥补特征提取过程中的损失，得到特征 h''_{fusion3}，具体实现细节见表征式(42)。

$$h'_{\text{fusion3}} = h_{\text{fusion3}} + \text{LayerNorm}\left(h_{\text{fusion3}}\right) \tag{42}$$

$$h''_{\text{fusion3}} = \text{Layer}\left(h'_{\text{fusion3}}\right) \tag{43}$$

3.3.4. 全局特征提取

在经过若干层的归一化和前馈神经网络处理后，所有节点的特征会进行池化操作。均值池化(Mean Pooling)通过对所有节点特征求平均，生成整个图的全局特征表示：

$$h_{\text{global}} = \frac{1}{N}\sum_{i=1}^{N} h''_{\text{fusion3}} \tag{44}$$

其中，N 是节点的数量，均值池化可以有效地整合整个图中节点的信息，整合为一个全局特征向量，生

成一个全局的特征 h_{global}。

最后，池化后的全局特征 h_{global} 经过线性变换，得到适用于下游任务的特征表示：

$$h'_{\text{global}} = w_{\text{global}} h_{\text{global}} + b_{\text{global}} \tag{45}$$

全局特征 h_{global} 可以用于后续的路径规划、节点分类等任务，作为整个编码器的输出。

3.4. 车辆选择解码器(Vehicle Selection Decoder)

不同的车辆在不同的状态、路径、任务目标下可能有不同的优先级，因此车辆选择解码器通过综合分析车辆和路径的特征，选择出最适合当前任务的车辆

3.4.1. 输入层分析

车辆特征嵌入(Vehicle Feature Embedding)

车辆特征通过嵌入层来表示每辆车的状态，输入的特征包括车辆的当前位置信息、累计行驶时间等。其中，第 i 辆车的状态特征为 $CV_i \in R^d_{\text{vehicle}}$，通过线性变换将其映射到新的特征空间，得到特征 h_{vehicle_i}，具体实现细节见表征式(46)，其中 W_{vehicle} 为权重矩阵，b_{vehicle} 为偏置项。

$$h_{\text{vehicle}_i} = W_{\text{vehicle}} CV_i + b_{\text{vehicle}} \tag{46}$$

路径特征嵌入(Route Feature Embedding)

路径特征通过嵌入层表示当前车辆路径的信息。其中，每辆车已经访问的路径特征为 CR_i，路径特征通过最大池化层 MaxPooling 进行操作提取主要特征最终得到特征 h_{route_i}，通过线性变换将其映射到新的特征空间，得到特征 h'_{route_i}，具体实现细节见表征式(47)和(48)。

$$h_{\text{route}_i} = \text{MaxPooling}\left(\text{CR}_i\right) \tag{47}$$

$$h'_{\text{route}_i} = W_{\text{route}} h_{\text{route}_i} + b_{\text{route}} \tag{48}$$

3.4.2. 前馈网络与特征融合

车辆与路径特征的前馈网络处理

处理完成后的车辆特征 h_{vehicle_i} 和路径特征 h'_{route_i} 分别经过独立的前馈神经网络(Feed Forward Network, FFN)进行处理，前馈网络包括线性变换和激活函数：

$$h'_{\text{vehicle}_i} = \text{Relu}\left(W_{ff} h_{\text{vehicle}_i} + b_{ff}\right),\ h''_{\text{route}_i} = \text{Relu}\left(W_{ff} h'_{\text{route}_i} + b_{ff}\right) \tag{49}$$

特征拼接(Concat)

经过前馈网络处理后的车辆特征和路径特征通过拼接操作整合在一起，以便后续的特征融合和决策。拼接后的特征表示为：

$$h_{\text{concat}_i} = \text{concat}\left(h'_{\text{vehicle}_i} \middle| h''_{\text{route}_i}\right) \tag{50}$$

该表示将车辆状态与路径状态结合，提供更丰富的信息用于车辆选择。

3.4.3. 线性层与 Softmax

线性层

拼接后的特征向量 h_{concat_i} 会通过线性层进行投影，生成一个表示该车辆是否适合被选择的分数：

$$h_{\text{linear}i} = \text{Linear}_4\left(h_{\text{concat}_i}\right) \tag{51}$$

该分数通过线性层计算后，反映了当前状态下选择该车辆的优先级。

最终，所有车辆的分数会通过 Softmax 操作转化为概率分布，表示每辆车被选择的概率：

$$p_{\text{vehicle}_i} = \frac{\exp\left(h_{\text{linear}i}\right)}{\sum_{j=1}^{m} \exp\left(h_{\text{linear}j}\right)} \tag{52}$$

其中 mmm 是车辆的数量，Softmax 确保所有车辆的选择概率总和为 1。

3.4.4. 车辆选择结果

根据 Softmax 输出的概率分布，通过最大化选择概率值的方法选出最优的车辆 V_{select}。

$$V_{\text{select}} = \arg\max\left(p_{\text{vehicle}_i}\right) \tag{53}$$

该车辆将被选中执行当前的任务，后续的节点选择解码器会进一步为该车辆选择要访问的节点。

3.5. 节点选择解码器

节点选择解码器的核心作用是基于车辆选择结果和全局特征，从一组候选节点中选出最优节点，作为下一步车辆将访问的目标节点。在路径规划和任务分配等复杂任务中，节点的选择对车辆的行驶路径和任务的完成起着至关重要的作用。因此，节点选择解码器的设计考虑了多种特征输入，通过多层特征处理和概率化输出，最终选择最合适的节点(图 4)。

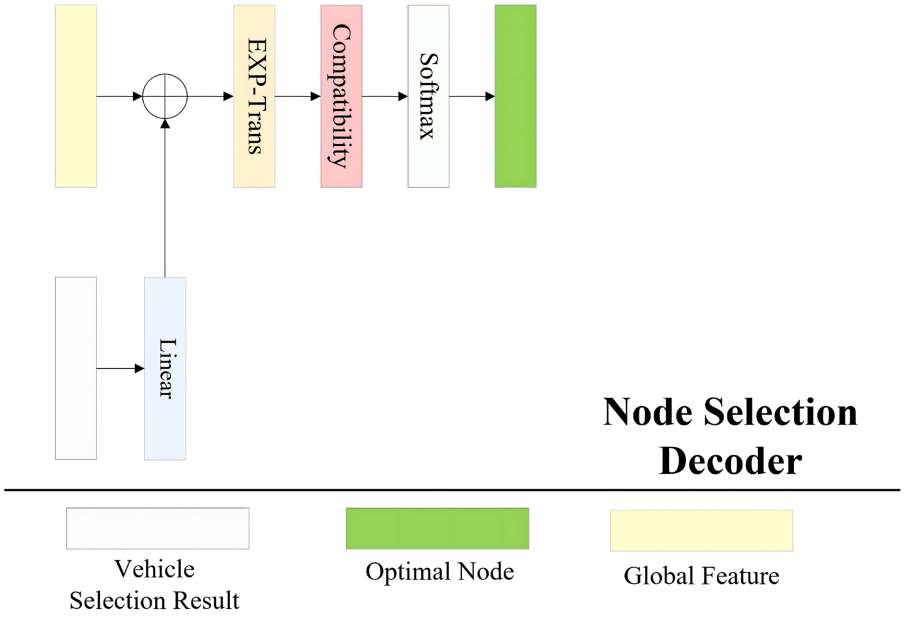

Figure 4. Node selection encoder
图 4. 节点选择编码器

3.5.1. 输入层分析

车辆选择结果(Vehicle Selection Result)

车辆选择结果来自于上一步的车辆选择解码器，表示当前被选中的车辆。车辆解码器中车辆选择结果表示为 V_{select}，通过线性变换 W_{vehicle} 将其映射到一个新的特征空间：

$$h_{\text{vehicle}} = W_{\text{vehicle}} V_{\text{select}} + b_{\text{vehicle}} \tag{54}$$

这里，W_{vehicle} 是线性变换的权重矩阵，b_{vehicle} 是偏置项。

全局特征(Global Feature)

全局特征 h_{global} 是从整个网络中提取出的综合特征，经过线性层处理后，与车辆选择结果进行融合

$$h'_{\text{global}} = W_{\text{global}} h_{\text{global}} + b_{\text{global}} \tag{55}$$

该操作将全局特征映射到与车辆特征相同的维度，便于后续的特征融合。

3.5.2. 特征融合与扩展转换(EXP Trans)

特征融合

车辆选择结果和全局特征经过各自的线性变换后，进行特征拼接操作，整合车辆选择和全局信息：

$$h_{\text{concatglobal}} = \text{concat}\left(h_{\text{vehicle}} \middle| h'_{\text{global}}\right) \tag{56}$$

融合后的特征包含了当前选择车辆的状态信息以及全局网络的特征。

扩展转换(EXP-Trans)

$$h_{\text{concat-Trans}} = \text{TransCFE}\left(h_{\text{concatglobal}}\right) \tag{57}$$

3.5.3. 兼容性计算(Compatibility Calculation)

兼容性评分

在经过扩展转换后，模型需要计算每个候选节点与当前车辆和全局状态的兼容性。假设有 N 个候选节点，扩展转换后的特征与每个节点的嵌入 h_{node_j} 进行兼容性计算。兼容性评分可以使用内积或其他相似性度量来计算：

$$c_j = \text{Compatibility}\left(h_{\text{concat-Trans}}, h_{\text{node}_j}\right) \tag{58}$$

兼容性评分 c_j 反映了当前特征与候选节点 j 之间的相似性或匹配程度。

3.5.4. Softmax 操作与节点选择

Softmax 归一化

对所有候选节点的兼容性评分 c_j 进行 Softmax 操作，转换为每个节点被选中的概率：

$$p_j = \frac{\exp\left(c_j\right)}{\sum_{k=1}^{N} \exp\left(c_k\right)} \tag{59}$$

Softmax 操作确保所有候选节点的选择概率总和为 1。

节点选择

根据 Softmax 输出的概率 p_j，可以通过最大化概率的方式选择最优节点 N_{opt}：

$$N_{\text{opt}} = \arg\max\left(p_j\right) \tag{60}$$

最优节点 N_{opt} 是与当前车辆最匹配的节点，表示模型选择的下一步行动目标。

3.6. 实验及结果分析

为了验证本文所提出的基于多特征融合的异构车辆路径规划算法的有效性，本文设计了一系列实验，以针对不同规模的车辆路径规划问题场景进行了不同的测试。这些实验的核心目标是评估算法在异构车辆路径规划问题(HCVRP)中的求解性能，包括在解决复杂问题时的计算效率、结果精度和求解时间。此外，本文还通过与其他经典算法进行对比，分析各类算法在不同场景下的优势以及不足。在本节的实验中，异构车辆具有不同的容量和速度，所有车辆从仓库出发，访问客户节点，完成路径规划任务。

3.6.1. 客户节点与车辆配置

为了模拟真实世界场景中的物流运输场景，本文对客户的节点和车辆的配置进行了详细设计。客户需求设置中，客户的需求值从集合$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$中随机抽取，代表了客户的不同配送需求，仓库的需求设置为 0。客户节点位置和仓库节点位置的设置中，所有车辆的起始点为仓库，客户节点和仓库节点均匀分布在单位平方为$[0,1]\times[0,1]$的二维平面内，采用了均匀分布随机抽样。同时，本文为实验设置了两种不同规模的车辆队伍，分为 V3 车队和 V5 车队。V3 车队由 3 辆异构车辆组成，车辆的容量分别设定为 20、25 和 30，客户节点数量为 40、60、80、100 和 120。V5 车队由 5 辆异构车辆组成，车辆的容量分别设定为 20、30、35、40 和 50，客户节点数量设为 80、100、120、140 和 160。通过这种设计，本文可以测试所提出的多特征融合的异构车辆路径规划算法在不同规模问题上的适应性，确保算法在面对不同数量的客户节点和车辆配置时，能够提供高效率和高精度的路径规划。

3.6.2. 车辆速度设置

车辆速度的不同会对路径规划产生显著影响，特别是在异构车队的路径规划问题当中，如何合理分配车辆任务成为算法的关键挑战之一。因此，本文在实验中针对不同的车队配置了不同的速度组合。Min-Max HCVRP 最小化最大行驶距离的异构车辆路径规划问题中将速度 f 简化为 1。对于 Min-Sum HCVRP 最小化总行驶距离的异构车辆路径规划问题，在 V3 车队中，每辆车的速度分别为$\frac{1}{4}$，$\frac{1}{5}$和$\frac{1}{6}$。在 V5 车队中，每辆车的速度分别为$\frac{1}{4}$，$\frac{1}{5}$，$\frac{1}{6}$，$\frac{1}{7}$和$\frac{1}{8}$。

这种速度配置可以模拟现实中不同车辆性能的差异，进一步验证算法的负载均衡性和调度灵活性。

3.6.3. 训练与测试策略

本文提出的车辆路径规划算法采用了深度强化学习(DRL)框架进行训练。为保证模型能够适应复杂场景，训练数据均为随机生成，涵盖了不同规模的客户节点和车辆配置。训练 Batch 批次为 2500，训练总规模达到 128 万个实例，以确保本文所提出的算法在足够多的样本下学习路径规划策略。同时，在训练过程中，模型通过策略网络不断优化，学习在不同场景中如何有效调度异构车辆，找到最优的行驶路径。本文还引入了学习率衰减机制，以确保模型在迭代过程中逐步收敛，避免过拟合问题的出现。每个场景的训练均在最大时间限制内完成，最终选择表现最优的模型进行测试和评估。

3.6.4. 评价指标

本文在实验中采用了三个核心指标来系统评估算法的性能，分别为：obj、gap 和 time。以下是这些指标的详细解释：

obj (目标函数值)：这是评估车辆路径规划问题的核心目标函数值，表示所有车辆完成配送任务的总行驶距离。该值越小，表示算法规划的路径越优，路径规划的效率越高。本文记录了各个场景下算法找到的最优 obj 值，并将其与已知的基准解进行对比。

$$\text{obj}=\sum_{k=1}^{M}\sum_{i=1}^{N}d\left(x_i,x_{i+1}\right)\cdot y_{i,k} \tag{61}$$

其中，M 是车辆的总数，N 代表客户节点的总数，$d\left(x_i,x_{i+1}\right)$表示车辆从节点 i 到节点 $i+1$ 的距离，$y_{i,k}$ 则表示第 k 辆车是否经过节点 i。

gap (相对误差率)：用于评估算法的求解精度，通过计算求解结果与已知最优解之间的相对误差。其公式定义为：

$$\text{gap} = \frac{|\text{obj} - \text{obj}'|}{\text{obj}'} \times 100\% \tag{62}$$

obj 为算法求解得到的目标函数值， obj′ 是已知的全局最优解。同时，求得的 gap 值越小，表示算法的求解精度越高，越接近全局最优解。在实验中，本文对不同规模的问题实例计算 gap，以验证算法在大规模问题中的性能表现。

time (计算时间)：该指标用于评估算法从开始运行到结束的总计算时间，time 越短说明算法在处理复杂场景时的计算效率越高。

3.6.5. 实验设备

本文所采用的所有实验均在一台配置为两张 NVIDIA RTX 3090 GPU、Intel Core I9 CPU 和 64 GB 内存的高性能工作站上进行。本文提出的多特征融合算法基于 PyTorch 框架，并在 CUDA 环境下进行并行加速计算。为了提高算法的收敛速度和稳定性，本文采用了 Adam 优化器，并在训练过程中逐步衰减学习率，以确保模型能够高效收敛。

3.6.6. 实验结果分析

在现代物流系统中，车辆路径优化问题(HCVRP)是提高物流运营效率的关键因素，尤其在面对大规模客户需求、多种运输工具和复杂的配送路线时。为了应对这一挑战，本研究通过两种典型的 HCVRP 问题(Min-Max HCVRP 和 Min-Sum HCVRP)进行实验，针对物流运输中的不同需求设置了两种车辆配置(V3 和 V5)，并生成了大量解方案(1280 和 12800)，评估了多种经典算法和本文提出方法在不同物流场景下的表现。

在实际物流系统中，目标是通过优化车辆行驶路线，降低运营成本，提高客户响应速度，并确保运输资源的合理配置。因此，实验不仅关注目标函数值(Obj)和相对误差率(Gap)，还特别注重计算时间(Time)，因为在物流系统中，快速求解高效路径对于实时调度至关重要。

3.6.7. Min-Max HCVRP 结果分析

在物流调度中，Min-Max HCVRP 的优化目标是通过均衡各车辆的最大行驶距离，确保运输任务的均衡分配，避免车辆过度负载或闲置，从而提高整个物流系统的运营效率。在 Min-Max HCVRP 中，优化目标是最小化所有车辆中的最大行驶距离，以确保车辆负载的均衡性。通过 V3 和 V5 配置的不同车辆组合，本文分析了 SISR、VNS、ACO、FA 等经典算法与本文提出方法(MFF-Greedy 和 MFF-Sample1280)在各个场景中的性能。

目标函数值(Obj)：在 V3 配置下(如 V3-C100 和 V3-C120)，MFF-Sample1280 方法在优化最大行驶距离方面表现优异。在 V3-C120 场景中，MFF-Sample1280 的 Obj 值为 10.52，相比 SISR 的 10.81 和 ACO 的 11.79，优化幅度达到 10.8%。在 V5 配置下(如 V5-C100 和 V5-C120)，MFF 方法同样展现了显著的优势，尤其在复杂问题中，MFF-Sample1280 的 Obj 值为 15.66，显著优于传统算法的表现。

相对误差率(Gap)：在 Min-Max HCVRP 中，MFF 方法的 Gap 值显著低于其他算法，表明其在接近最优解方面的能力更强。例如，在 V3-C120 场景中，MFF-Sample1280 的 Gap 值为 1.34%，而 ACO 和 FA 的误差率分别为 4.13%和 9.47%。在 V5-C120 场景中，MFF-Sample1280 的 Gap 为 1.07%，相比于 VNS 的 2.96%和 ACO 的 8.85%，显示出更强的求解精度。

计算时间(Time)：在计算时间方面，MFF 方法通过概率抽样生成的多解策略(如 Sample1280 和 Sample12800)，显著减少了在高维度输入下的求解时间。例如，在 V3-C120 场景中，MFF-Sample1280 的求解时间为 8.66 s，相比于 SISR 的 13,785 s，大幅缩短了求解时间。这种时间优势使得 MFF 方法在需要实时响应的物流场景中具有较强的竞争力(表 1)。

Table 1. Analysis of Min-Max HCVRP results
表 1. Min-Max HCVRP 结果分析

	Method	V3-C40			V3-C60			V3-C80			V3-C100			V3-C120		
		Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
Min-max	SISR	4.00	0%	245 s	5.58	0%	468 s	7.27	0%	752 s	8.89	0%	1135 s	10.42	0%	1657 s
	VNS	4.17	4.25%	115 s	5.80	3.94%	294 s	7.57	4.13%	612 s	9.20	3.49%	927 s	10.81	3.74%	1378 s
	ACO	4.31	7.75%	209 s	6.18	10.75%	317 s	8.14	11.97%	601 s	10.05	13.05%	878 s	11.79	13.15%	1242 s
	FA	4.49	12.25%	168 s	6.30	12.90%	285 s	8.32	14.44%	397 s	10.11	13.72%	522 s	11.98	14.97%	667 s
	AM (Greedy)	4.85	21.25%	0.37 s	6.57	17.74%	0.54 s	8.32	14.44%	0.82 s	9.98	12.26%	1.07 s	11.63	11.61%	1.28 s
	AM (Sample1280)	4.36	9.00%	0.88 s	5.99	7.39%	1.19 s	7.73	6.33%	1.81 s	9.36	5.29%	2.51 s	10.94	4.99%	3.37 s
	AM (Sample12800)	4.31	7.75%	1.35 s	5.92	6.09%	2.46 s	7.66	5.36%	3.67 s	9.28	4.39%	5.17 s	10.85	4.13%	6.93 s
	MFF (Greedy)	4.45	11.03%	0.69 s	6.01	8.77%	0.80 s	7.81	7.51%	1.10 s	9.53	6.00%	1.42 s	10.92	5.22%	1.87 s
	MFF (Sample1280)	4.17	4.17%	1.19 s	5.57	3.36%	1.41 s	7.38	2.85%	2.26 s	9.05	2.11%	3.22 s	10.60	1.91%	4.48 s
	MFF (Sample12800)	4.14	3.34%	1.61 s	5.23	2.64%	2.92 s	7.44	2.32%	4.61 s	9.01	1.34%	6.70 s	10.52	1.31%	8.66 s
Min-sum	Exact-solver	55.43*	0%	71 s	78.47*	0%	214 s	102.42*	0%	793 s	124.61*	0%	2512 s	-	-	-
	SISR	55.79	0.65%	(254 s)	79.12	0.83%	−478	103.41	0.97%	763 s	126.19	1.27%	(1140 s)	149.1	0%	1667 s
	VNS	57.54	3.81%	109 s	81.44	3.78%	291 s	106.18	3.67%	547 s	129.32	3.78%	828 s	152.56	2.32%	1217 s
	ACO	60.11	8.44%	196 s	86.05	9.66%	302 s	113.75	11.06%	593 s	140.61	12.84%	859 s	166.5	11.67%	1189 s
	FA	59.94	8.14%	164 s	85.36	8.78%	272 s	112.81	10.14%	388 s	138.92	11.48%	518 s	164.53	10.35%	653 s
	AM (Greedy)	66.54	20.04%	0.49 s	91.19	16.21%	0.83 s	117.22	14.45%	1.01 s	141.14	13.27%	1.23 s	164.57	10.38%	1.41 s
	AM (Sample1280)	60.95	9.96%	0.92 s	85.74	9.26%	1.17 s	111.78	9.14%	1.79 s	135.61	8.83%	2.49 s	159.11	6.71%	3.30 s
	AM (Sample12800)	60.26	8.71%	1.35 s	84.96	8.27%	2.31 s	110.94	8.32%	3.61 s	134.72	8.11%	5.19 s	158.11	6.10%	6.86 s
	MFF (Greedy)	58.77	6.39%	0.59 s	82.98	5.77%	0.99 s	108.39	5.81%	1.03 s	131.72	5.73%	1.53 s	54.51	3.65%	1.84 s

Table 2. Analysis of Min-Sum HCVRP results
表 2. Min-Sum HCVRP 结果分析

	Method	V5-C40			V5-C60			V5-C80			V5-C100			V5-C120		
		Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
Min-max	SISR	3.90	0%	727 s	4.72	0%	1091 s	5.48	0%	1572 s	6.33	0%	1863 s	7.16	0%	2521 s
	VNS	4.15	6.41%	725 s	4.98	7.19%	1046 s	5.81	6.02%	1454 s	6.67	5.37%	2213 s	7.53	5.17%	3321 s
	ACO	4.50	15.38%	612 s	5.56	17.80%	890 s	6.47	18.07%	1285 s	7.52	18.80%	2081 s	8.51	18.85%	2898 s
	FA	4.61	18.21%	412 s	5.62	19.07%	541 s	6.58	20.07%	682 s	7.60	20.06%	822 s	8.64	20.67%	964 s
	AM (Greedy)	4.84	24.10%	1.08 s	5.70	20.76%	1.31 s	6.57	19.89%	1.74 s	7.49	18.33%	1.93 s	8.34	16.48%	2.15 s
	AM (Sample1280)	4.32	10.77%	1.88 s	5.18	8.75%	2.64 s	6.03	10.04%	3.38 s	6.93	9.48%	4.47 s	7.75	8.24%	5.73 s
	AM (Sample12800)	4.25	8.97%	3.71 s	5.11	8.26%	5.19 s	5.95	8.58%	6.94 s	6.86	8.37%	8.73 s	7.69	7.40%	10.69 s
	MFF (Greedy)	4.36	11.79%	1.29 s	5.20	10.17%	1.64 s	5.94	8.39%	2.38 s	6.78	7.11%	2.43 s	7.61	6.28%	3.02 s
	MFF (Sample1280)	4.08	4.62%	2.66 s	4.91	4.03%	3.66 s	5.66	3.28%	5.08 s	6.51	2.84%	6.48 s	7.34	2.51%	8.52 s
	MFF (Sample12800)	4.04	3.59%	5.06 s	4.87	3.18%	7.20 s	5.62	2.55%	9.65 s	6.47	2.21%	10.93 s	7.30	1.96%	13.76 s
Min-sum	Exact-solver	102.42*	0%	1787 s	124.63*	0%	6085 s	-	-	-	-	-	-	-	-	-
	SISR	103.49	1.04%	(735 s)	126.35	1.38%	(1107 s)	149.18	0%	(1580 s)	172.88	0%	(1881 s)	196.51	0%	(2539 s)
	VNS	109.91	7.31%	538 s	133.28	6.94%	811 s	156.37	4.82%	1386 s	180.08	4.16%	2080 s	203.95	3.79%	2896 s
	ACO	118.58	15.78%	608 s	146.51	17.56%	865 s	171.82	15.18%	1269 s	200.73	16.11%	1922 s	229.64	16.86%	2803 s
	FA	116.13	13.39%	401 s	142.39	14.25%	532 s	167.87	12.53%	677 s	196.48	13.65%	801 s	223.49	13.73%	955 s
	AM (Greedy)	128.31	25.28%	0.82 s	152.91	22.69%	1.28 s	177.39	18.91%	1.45 s	201.85	16.76%	1.69 s	227.10	15.57%	1.81 s
	AM (Sample1280)	119.41	16.59%	1.83 s	144.23	15.73%	2.66 s	168.95	13.25%	3.63 s	193.65	12.01%	4.68 s	218.67	11.28%	5.49 s
	AM (Sample12800)	118.04	15.25%	3.74 s	142.79	14.57%	5.20 s	167.45	12.25%	7.02 s	192.13	11.13%	8.93 s	217.14	10.50%	11.01 s
	MFF (Greedy)	108.23	5.76%	1.22 s	131.83	5.86%	1.66 s	154.67	3.67%	2.02 s	178.65	3.29%	3.01 s	202.76	3.21%	3.52 s
	MFF (Sample1280)	105.44	2.97%	2.67 s	128.54	3.33%	4.09 s	151.23	1.39%	5.25 s	175.12	1.21%	6.65 s	199.02	1.32%	8.64 s
	MFF (Sample12800)	104.49	2.32%	5.34 s	128.15	2.79%	7.57 s	150.75	1.24%	9.38 s	174.55	1.04%	11.21 s	198.54	1.07%	13.77 s

3.6.8. Min-Sum HCVRP 结果分析

在 Min-Sum HCVRP 中，目标是最小化所有车辆的总行驶距离，从而提升整体运输效率。实验结果表明，MFF 方法在 V3 和 V5 两种车辆配置下均展现出卓越的性能(表 2)。

目标函数值(Obj)：在 Min-Sum HCVRP 问题中，MFF-Sample1280 方法在优化总行驶距离上具有显著优势。在 V5-C120 场景中，MFF-Sample1280 的 Obj 值为 198.54，相比于 VNS 的 207.36 和 ACO 的 223.49，优化了约 11.1%。这种优势在大规模客户节点场景中尤为突出，表明 MFF 方法在复杂路径优化问题中，能够更高效地减少整体运输成本。

相对误差率(Gap)：MFF 方法的误差率在多个实验场景中显著低于其他方法，特别是在大规模问题中。在 V5-C120 场景下，MFF-Sample1280 的 Gap 为 1.96%，相比 ACO 的 8.85%和 FA 的 11.28%，展现出更高的求解精度。低误差率意味着 MFF 方法能够更准确地找到接近最优的路径规划方案，为物流调度提供更可靠的决策支持。

计算时间(Time)：在计算时间方面，MFF 方法利用概率抽样生成的多解策略显著提高了求解效率。例如，在 V5-C120 场景中，MFF-Sample1280 的求解时间为 13.75 s，相比 ACO 的 28,985 s 和 FA 的 955 s，时间效率提升了多个数量级。即使在最复杂的场景中，MFF 方法依然能够在较短时间内完成求解，体现了其在大规模物流问题中的应用价值。

3.6.9. 综合对比与讨论

经典启发式算法的表现：SISR、VNS 和 ACO 等经典启发式算法在小规模场景中求解质量较高，但随着客户节点数量的增加，求解时间和误差率显著增加，表现出较大的计算瓶颈。相比之下，MFF 方法通过深度学习与概率抽样相结合的策略，能够更高效地应对大规模复杂输入，求解精度显著提高。

基于概率抽样的优势：通过生成 1280 或 12,800 个解方案，MFF 方法能够在高维特征空间中更广泛地探索最优解。与 AM-Greedy 等强化学习方法相比，MFF 的 Sample1280 策略在求解精度上更具优势，同时在求解速度上也保持了较高的效率。

V3 与 V5 配置的影响：V3 和 V5 两种车辆配置分别适用于不同规模的客户节点场景。在 V5 配置下，由于车辆数量和容量的增加，使得总运输任务能够更好地分配到各辆车上，MFF 方法在这种场景中的优化效果尤为显著。通过调整抽样数量(如 1280 和 12,800)，进一步提升了对复杂调度任务的适应能力。

3.6.10. 实验结果总结

实验结果表明，本文提出的多特征融合方法在处理 Min-Max HCVRP 和 Min-Sum HCVRP 时，均展现出了卓越的性能优势。特别是通过概率抽样策略，MFF 方法能够在大规模复杂场景下有效平衡计算时间和求解精度，显著提升了负载均衡性和整体运输效率。结合 V3 和 V5 的不同车辆配置，无论是均衡各车辆的负载还是最小化总运输成本，MFF 方法都能在较短时间内提供接近最优的解决方案，具有重要的实际应用价值。这一研究为智能物流系统的路径规划提供了新的思路，未来在实际物流环境中的应用前景广阔。

4. 总结

本文针对异构车队车辆路径规划(HCVRP)中多目标协同优化与动态约束的难题，提出了一种基于深度强化学习的多特征融合框架(MFF-HVPP)，在理论与实验层面取得显著提升。在理论建模方面，通过马尔可夫决策过程(MDP)对 HCVRP 进行动态建模，构建了包含车辆状态与节点状态的复合状态空间，并结合双模态动态奖励函数，分别实现了最小化最大行程时间(min-max)与总行程时间(min-sum)的优化目标，为解决多目标冲突提供了可扩展的数学框架。

在算法设计层面，本文提出多特征融合编码器，结合卷积神经网络(CNN)与位置嵌入技术，分别提取节点需求的局部特征与空间拓扑的全局依赖关系；通过引入 Transformer 通道特征扩展模块(TransCFE)，实现通道维度的特征聚合与残差连接，有效缓解传统注意力机制中的梯度消失与过拟合问题，显著提升了模型对异构车辆与动态需求的建模能力。同时，为了降低高维解空间的搜索复杂度，网络采用了分层解码架构，将路径规划解耦为车辆调度与节点选择的序列决策过程：其中车辆选择解码器基于实时车辆状态与全局特征动态分配任务，节点选择解码器则通过兼容性计算筛选最优访问节点，并结合概率化抽样策略实现平衡全局优化与局部搜索效率。

实验验证表明，MFF-HVPP 在两类典型 HCVRP 任务中均展现出显著优势。在 120 客户节点的 min-max 场景中，模型最大行程时间误差率(Gap)低至 1.31%，较传统 SISR 算法优化幅度超 10%，计算效率提升 98%；在 min-sum 任务中，总行程时间优化误差率仅为 1.07%，并且支持 160 个节点级场景实现 13.77 秒的实时响应。通过 V3 与 V5 异构车队配置的对比实验，进一步验证了模型在负载均衡与成本控制上的鲁棒性，尤其在复杂约束下仍能保持高效性与稳定性。

本研究不仅为智能物流系统的动态路径规划提供了高效、可扩展的解决方案，其方法论和模型框架还可迁移至无人机配送、多机器人协作等复杂优化问题，具有重要的工程应用价值。未来工作将聚焦于模型的实时动态需求与实时道路场景变化的适应性优化，推动智能调度技术在实际物流场景中的广泛落地，为物流行业提供技术支撑。

参考文献

[1] Escobar, J.W., Duque, J.L.R. and García-Cáceres, R. (2022) A Granular Tabu Search for the Refrigerated Vehicle Routing Problem with Homogeneous Fleet. *International Journal of Industrial Engineering Computations*, **13**, 135-150. https://doi.org/10.5267/j.ijiec.2021.6.001

[2] Rjeb, A., Gayon, J. and Norre, S. (2021) Sizing of a Homogeneous Fleet of Robots in a Logistics Warehouse: Transport Operation between Reception Area and Storage Area. *IFAC-PapersOnLine*, **54**, 552-557. https://doi.org/10.1016/j.ifacol.2021.08.169

[3] Chaikovskaia, M., Gayon, J., Chebab, Z.E. and Fauroux, J. (2021) Sizing of a Fleet of Cooperative Robots for the Transport of Homogeneous Loads. 2021 *IEEE 17th International Conference on Automation Science and Engineering* (*CASE*), Lyon, 23-27 August 2021, 1654-1659. https://doi.org/10.1109/case49439.2021.9551509

[4] Hua, C., Berto, F., Son, J., *et al*. (2025) CAMP: Collaborative Attention Model with Profiles for Vehicle Routing Problems. arXiv: 2501.02977.

[5] Liu, Q., Liu, C., Niu, S., *et al*. (2024) 2D-Ptr: 2D Array Pointer Network for Solving the Heterogeneous Capacitated Vehicle Routing Problem. *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, Auckland, 6-10 May 2024, 1238-1246.

[6] Deineko, E. and Kehrt, C. (2024) Learn to Solve Vehicle Routing Problems ASAP: A Neural Optimization Approach for Time-Constrained Vehicle Routing Problems with Finite Vehicle Fleet. arXiv: 2411.04777.

[7] Dantzig, G.B. and Ramser, J.H. (1959) The Truck Dispatching Problem. *Management Science*, **6**, 80-91. https://doi.org/10.1287/mnsc.6.1.80

[8] Pecin, D., Pessoa, A., Poggi, M. and Uchoa, E. (2016) Improved Branch-Cut-and-Price for Capacitated Vehicle Routing. *Mathematical Programming Computation*, **9**, 61-100. https://doi.org/10.1007/s12532-016-0108-8

[9] Munari, P. and Morabito, R. (2018) A Branch-Price-and-Cut Algorithm for the Vehicle Routing Problem with Time Windows and Multiple Deliverymen. *TOP*, **26**, 437-464. https://doi.org/10.1007/s11750-018-0481-8

[10] Repoussis, P.P. and Tarantilis, C.D. (2010) Solving the Fleet Size and Mix Vehicle Routing Problem with Time Windows via Adaptive Memory Programming. *Transportation Research Part C*: *Emerging Technologies*, **18**, 695-712. https://doi.org/10.1016/j.trc.2009.08.004

[11] Clarke, G. and Wright, J.W. (1964) Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, **12**, 568-581. https://doi.org/10.1287/opre.12.4.568

[12] Ropke, S. and Pisinger, D. (2006) An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, **40**, 455-472. https://doi.org/10.1287/trsc.1050.0135

[13] Braekers, K., Ramaekers, K. and Van Nieuwenhuyse, I. (2016) The Vehicle Routing Problem: State of the Art Classification and Review. *Computers & Industrial Engineering*, **99**, 300-313. https://doi.org/10.1016/j.cie.2015.12.007

[14] Arnold, F. and Sörensen, K. (2019) What Makes a VRP Solution Good? The Generation of Problem-Specific Knowledge for Heuristics. *Computers & Operations Research*, **106**, 280-288. https://doi.org/10.1016/j.cor.2018.02.007

[15] Kool, W., Van Hoof, H. and Welling, M. (2018) Attention, Learn to Solve Routing Problems! arXiv: 1803.08475.

[16] Shahbazian, R., Pugliese, L.D.P., Guerriero, F. and Macrina, G. (2024) Integrating Machine Learning into Vehicle Routing Problem: Methods and Applications. *IEEE Access*, **12**, 93087-93115. https://doi.org/10.1109/access.2024.3422479

[17] Renaud, J., Boctor, F.F. and Laporte, G. (1996) An Improved Petal Heuristic for the Vehicle Routeing Problem. *Journal of the Operational Research Society*, **47**, 329-336. https://doi.org/10.1057/palgrave.jors.0470211

[18] Ewbank, H., Wanke, P. and Hadi-Vencheh, A. (2015) An Unsupervised Fuzzy Clustering Approach to the Capacitated Vehicle Routing Problem. *Neural Computing and Applications*, **27**, 857-867. https://doi.org/10.1007/s00521-015-1901-4

[19] Dai, H., Khalil, E.B., Zhang, Y., *et al*. (2017) Learning Combinatorial Optimization Algorithms over Graphs. arXiv: 1704.01665.

[20] Bahovska, E. (2023) Graph Neural Networks in Neighborhood Selection for a Vehicle Routing Problem Solver. Master Thesis, Utrecht University.

[21] Vinyals, O., Fortunato, M. and Jaitly, N. (2015) Pointer Networks. arXiv: 1506.03134.

[22] Nazari, M., Oroojlooy, A., Snyder, L., *et al*. (2018) Reinforcement Learning for Solving the Vehicle Routing Problem. arXiv: 1802.04240.

[23] Watkins, C.J.C.H. and Dayan, P. (1992) Technical Note: Q-Learning. *Machine Learning*, **8**, 279-292. https://doi.org/10.1023/a:1022676722315

[24] Lin, B., Ghaddar, B. and Nathwani, J. (2022) Deep Reinforcement Learning for the Electric Vehicle Routing Problem with Time Windows. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 11528-11538. https://doi.org/10.1109/tits.2021.3105232

[25] Joe, W. and Lau, H.C. (2020) Deep Reinforcement Learning Approach to Solve Dynamic Vehicle Routing Problem with Stochastic Customers. *Proceedings of the International Conference on Automated Planning and Scheduling*, **30**, 394-402. https://doi.org/10.1609/icaps.v30i1.6685

[26] Baty, L., Jungel, K., Klein, P.S., Parmentier, A. and Schiffer, M. (2024) Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows. *Transportation Science*, **58**, 708-725. https://doi.org/10.1287/trsc.2023.0107

[27] Kool, W., van Hoof, H., Gromicho, J. and Welling, M. (2022) Deep Policy Dynamic Programming for Vehicle Routing Problems. In: Schaus, P., Ed., *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer, 190-213. https://doi.org/10.1007/978-3-031-08011-1_14

[28] Lee, D.H. and Ahn, J. (2023) A Deep Reinforcement Learning Approach to Solve the Vehicle Routing Problem with Resource Constraints. *AIAA SCITECH* 2023 *Forum*, National Harbor, 23-27 January 2023, 2262. https://doi.org/10.2514/6.2023-2662