编号_____

南京航空航天大学

毕业论文

题　目　基于深度强化学习智能交通信号调度

学生姓名　陈建

学　　号　SX1916039

学　　院　计算机科学与技术学院

专　　业　计算机科学与技术

班　　级　1318001

指导教师　朱琨

二〇二一年十二月

南京航空航天大学
本科毕业论文诚信承诺书

本人郑重声明:所呈交的毕业论文(题目:基于深度强化学习智能交通信号调度)是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业论文中特别加以标注引用的内容外，本毕业论文不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名:　　年　月　日

(学号):

基于深度强化学习智能交通信号调度

摘 要

先进的学习技术的不断发展和更容易获得的实时交通数据使得动态调整交通信号成为可能。一些研究工作提出使用强化学习（RL）进行交通信号控制，与传统的方法相比，取得了卓越的性能。然而，现有的方法仍然有一些需要改进的地方。例如，大多数基于 RL 的单路口交通控制方法只关注于提高总效率，而忽略了公平问题。而对于多路口的交通信号控制，现有的方法在学习通信时试图将自己路口的所有信息传递给目标路口，这使得目标路口难以挖掘有效信息。

在本文中，我们对已有工作在这两种场景下的缺点进行了改进。对于单个路口的交通信号控制，我们提出了一个基于公平意识的 RL 模型，该模型受到比例公平调度的启发，可以在效率和公平之间提供一个良好的权衡。为了简化多路口交通信号控制中的协调工作，我们提出了一种新的建图方法，在汇总邻居信息时，可以消除与目标节点无关的信息。最后，我们进行综合实验来验证这两项工作的有效性。

关键词：交通信号控制，强化学习，公平性，协调，图神经网络

Traffic Signal Control Based on Deep Reinforcement Learning

Abstract

Increasing development in advanced learning techniques and easier access to real-time traffic data make it possible to adjust the traffic signal dynamically. Several studies have proposed to use reinforcement learning (RL) for traffic signal control and achieved superior performance compared with the traditional methods. However, existing methods still have something to be improved. For example, most of the RL-based single intersection traffic control methods only focus on improving the total efficiency but ignore the fairness problem. And for multi-intersection traffic signal control, Existing methods try to transfer all the information of their own intersection to the target intersection when learning to communicate, which makes it difficult for the target intersection to mine the effective information.

In this paper, we improve the existing work in these two scenarios. For single-intersection traffic signal control, we propose a fairness-aware RL-based model inspired by the proportional fair scheduling which can provide a good trade-off between efficiency and fairness. To simplify the coordination in multi-intersection traffic signal control, we propose a new making graph method which can eliminate information irrelevant to the target node when aggregating neighbor information. Finally, we conduct comprehensive experiments to verify the effectiveness of these two works.

Key Words: Traffic Signal Control, Reinforcement Learning, Fairness-Aware, Coor-

dination, Graph Neural Network

目录

第一章 研究背景

随着汽车制造业的快速发展以及城市化进程的推进，我国汽车保有量在不断的增加，交通拥堵情况也在不断恶化，极大的影响了人们的生活的城市的运作，同时这种拥堵现象也在向中小城市蔓延。为了缓解交通拥堵，很多城市也提出了不同的解决方法，有减少出行车辆数量的“限号”政策，也有通过加快城市道路建设来加大城市交通承载量的方法。其实，交通拥堵通常是由于不同的车流为了争夺同一个“行驶资源”而造成的。这一“行驶资源”通常就是车辆所处的交通路口，所以现代城市交通管理的装主要主要方法是在道路汇合的交叉路口安装信号灯并通过简单的策略来调度通过的车流，已到达减少交通拥塞的目的。但是随着车辆数量的不断增加，之前传统的交通信号控制策略已经难以应对现在更加复杂的交通模式。因此，如何制定出更加高效和智能的交通信号调度策略显得格外的重要。

随着车联网技术的发展，对于实时车辆数据的获取变得越来越容易，利用得到的车辆数据可以获得实时的交通状况，并且如何根据实时的交通状况来制定最优的策略一直是研究的热点。以往多数的研究是采用基于优化的方法，根据车流的状况计算出一个最优的信号灯的相位序列，但是这种方法要求车流的状况是比较简单的，例如服从均匀分布，与现实中的车流情况相比太过理想化，所以难以部署到实际场景中。伴随着人工智能技术的发展，一些研究者提出利用深度强化学习来控制信号灯，将整个交通信号灯控制建模成一个马尔可夫决策过程（Markov Decision Process）。对于每一次决策，输入当前的交通状况作为状态，输出一个作用在信号灯上的动作，例如变换到下一个相位（phase）。这种方法对于车流的情况没有限制，通过在大量不同的仿真车流下进行训练可以得到一个鲁棒的模型，能够应对不同的车流场景并做出最优的决策，并且这

种方法在通行效率上也比基于优化的方法和传统的规则控制方法更高。

1.1 研究意义

传统的交通控制方法在当下已经难以有效地减轻交通拥堵的情况，因为其更多的是通过一些预先设定的规则和一些根据历史数据总结出的经验来控制信号灯，没有考虑实时的交通状况，但是由于其简单以及易于部署的特点，绝大多数城市的信号灯都还在采用这种控制模式。

随着人工智能技术的飞速发展以及越来越多的城市交通数据，政府和企业正在积极寻求改善交通系统的智能交通信号控制解决方案。与传统交通信号控制不同的是，智能交通信号控制会根据实时的交通状况做出最优的决策并以此来控制信号灯的变化，已达到最大程度地减轻交通拥堵的目的。另一方面，随着最近强化学习技术的发展，我们看到学术界对使用强化学习来改善交通信号控制的热情越来越高涨，并且也提出了很多基于深度强化学习的智能交通信号控制方法，但是任然有很多问题需要研究。本文主要尝试解决已有工作中遗留的两个问题，即公平性问题和多路口场景下的协调通信问题。

公平性问题是指，不同车辆通过同一个路口所需的通行时间可能有很大的差别，因为信号灯可能为了提高整体通行效率而牺牲一些车辆，让这些车辆多等待一些时间，即便这些车辆可能是先进入路口的，这对这些车来说是不公平的。一个好的控制策略应该在提高通行效率的同时能够保证每辆车所需的通行时间大致相同，也就是说，车辆通行时间的方差应该越小越好。但是已有的工作都是使用车辆的平均通行时间来衡量通行效率，很自然的忽略了公平性问题。

在多路口交通信号控制问题中，路口之间通过通信来进行数据交互可以实现多个路口的协同控制，从而提高整个路网的通行效率。但是已有的工作在进行数据交互时，笼统地将所有的信息传递给目标路口，会导致目标路口难以提取出有效的信息，从而增加学习的难度，甚至学习出错误的策略。

第二章 概述

2.1 交通信号概述

交通信号控制是一个重要而具有挑战性的现实问题，其目的是通过协调车辆在道路交叉口的运动来最小化所有车辆的通行时间。目前广泛使用的交通信号控制系统仍然严重依赖过于简化的信息和基于规则的方法。车联网技术的发展、硬件性能的提升以及人工智能技术的进步使得我们现在有更丰富的数据、更多的计算能力和先进的方法来驱动智能交通的发展。交通信号控制的目的为了方便车辆在交叉路口的安全和高效移动。安全是通过信号灯指定不同车道的车通行来分离相互冲突的运动实现的。为了能够有效地优化通行效率，已有的工作提出了不同的指标来量化通行效率，主要有以下三个：

- 通行时间：在交通信号控制中，车辆的行驶时间被定义为一辆汽车进入系统的时间与离开系统的时间的差值。最常见的优化目标之一是就是减少进过路口的所有车辆的平均通行时间。
- 队列长度：队列长度是指路口等待车辆的数量，越大的队列长度意味着越多的等待车辆，路口的通行效率越低，反之通行效率越高。
- 路口吞吐量：吞吐量是指在一定期间内进过路口完成通行的车辆数量。越大的吞吐量代表着越高的通行效率，所以很多工作将最大化吞吐量作为优化的目标。

2.2 基本术语

- Approach: 指交叉路口的巷道。任何一个交叉路口都有两种 approach, 进入路口的 incoming approach 和离开路口的 outgoing approach。图 2.1(a) 描述了一个典型的有 8 个 approach（四个入口，四个出口）的交叉路口。

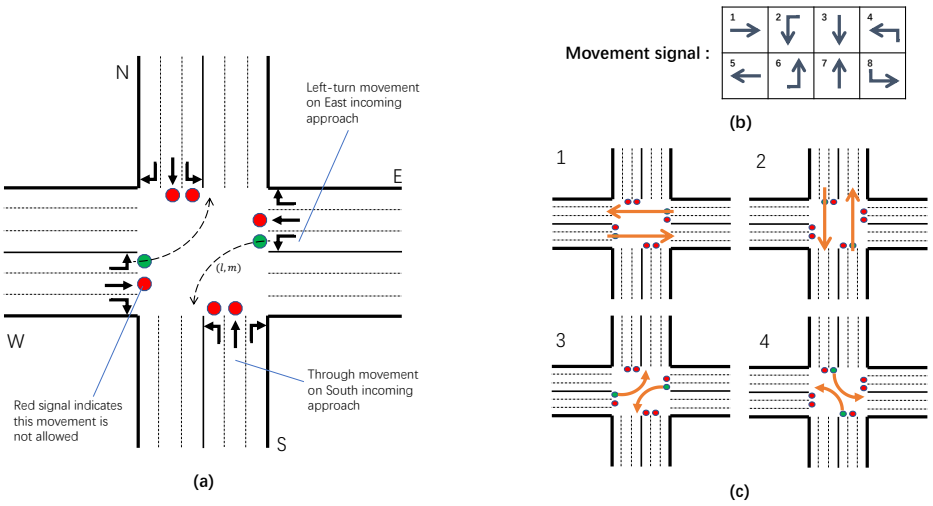


图 2.1 tsc

- Lane：一个 Approach 是由一组车道组成。与 Approach 的定义类似，车道也分为两种：转入车道（incoming lane）和转出车道 (outgoing lane)。
- Traffic movement：指的是车流从一个 incoming approach 运动到另一个 outgoing approach，表示为 (r_i, r_o)，其中 ri 和 ro 分别表示 incoming lane 和 outgoing lane。通常，traffic movement 可以分为左转、直行以及右转三种，在少数特殊的路口也支持 U-turn 的 traffic movement。
- Movement signal: 根据 traffic movement 定义的运动信号，绿色代表可以通行，红色代表禁止通行。根据大多数国家的交通规则，右转的 traffic movement 是可以不受信号约束的。
- Phase: 信号灯的一个 phase(相位) 是指非冲突运动信号的组合，这意味着这些信号可以同时设置为绿色，而不会引起安全冲突。图 2.1(c) 展示了最常用的四相位信号模式。
- Phase sequence: 相序，即一组相位的序列，它定义了一组相位及其变化顺序。
- Signal plan：信号计划，由一组相位序列及其相应的起始时间组成。通常

表示为 $(p_1,t_1)(p_2,t_2)\ldots(p_i,t_i)\ldots$ 其中 p_i 和 t_i 分别代表相位及其开始时间。

- Cycle-based signal plan：周期性信号计划，与普通的信号计划不同的是其中的相位序列是按循环顺序工作的，可以表示为 $(p_1,t_1^1)(p_2,t_2^1)\ldots(p_N,t_N^1)(p_1,t_1^2)(p_2,t_2^2)\ldots(p_N,t_N^2)\ldots$，其中 $p_1,p_2,\ldots,p_N$ 是重复出现的相位序列，t_i^j 是 j 周期中相位 p_i 的起始时间。具体地，$C^j = t_1^{j+1} - t_1^j$ 是第 j 周期的周期长度，$\left\{\frac{t_2^j-t_1^j}{C^j},\ldots,\frac{t_N^j-t_{N-1}^j}{C^j}\right\}$ 是第 j 周期中的相位分裂比（phase split ratio），表示每个相位持续时间占总周期长度的比重。现有的交通信号控制方法通常在一天中重复类似的相位序列。

2.3 传统交通控制方法

2.3.1 Fixed-Time

使用最为广泛的一种交通信号控制方法，按照事先固定的信号序列不断的循环，不依赖于任何类型的检测，例如行人按钮或车辆检测装置。所有道路和运动都以恒定的特定顺序提供服务。即使没有汽车或行人，信号也会改变，在交通较为稳定的情况下能够起到不错的效果，但是当交通变化很大时效率会很低。由于不需要安装任何检测器，所以这种方法的成本效益高。

2.3.2 Webster

对于单个交叉口，交通运输工程领域中的交通信号控制方法通常由三个部分组成：确定信号周期长度，确定信号相位序列以及相位分裂。Webster 是一种广泛使用的计算单个交叉路口的信号周期长度和相位分裂时间的方法。通过假设车流在一段时间内（例如，过去的五分钟或 10 分钟）是均匀到达的，可以计算出确切的最优周期和最佳相位分裂时间，从而最小化车量通行时间。

2.3.3 GreenWave

虽然使用 Webster 可以简单的控制单个交叉路口的交通信号，但是对于相邻的多个交叉路口，不能够简单地直接使用 Webster 来分别优化每一个路口，相邻路口信号灯的信号时间之间的偏移（即相邻路口信号周期起始时间的差

值）也需要进行优化，因为对于相距较近的路口来说，一个路口的控制策略可能会影响到其他路口。GreenWave 就是交通运输领域中最经典的协调相邻路口的信号控制方法，它通过优化相邻路口信号时间的偏移来减少车辆在某一方向行驶时的停留次数。其中路口之间的偏移量通过以下公式计算：

$$\Delta t_{i,j} = \frac{L_{i,j}}{v} \tag{2.1}$$

其中 $L_{i,j}$ 是路口 i 和路口 j 之间的道路长度，v 是道路上车辆的预期行驶速度。这种方法可以形成沿指定交通方向的绿色信号波，在该方向行驶的车辆可以受益于渐进的绿色信号级联，而不会在任何交叉口停留，如下图所示：

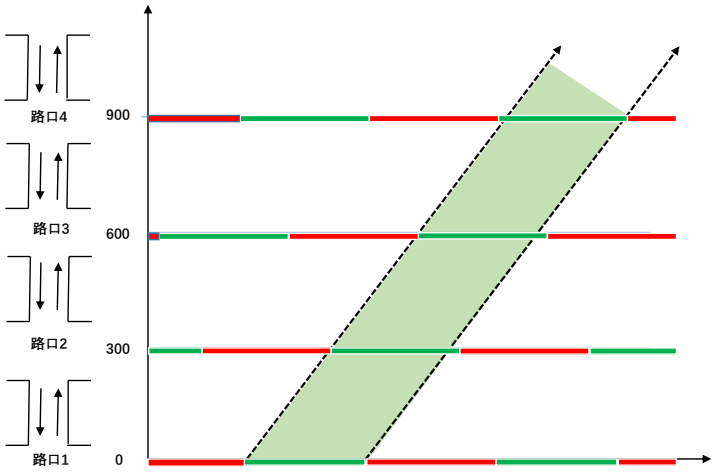


图 2.2　green-wave

2.3.4　Actuated Control

Actuated Control 根据当前信号相位和其他的竞争信号相位对绿色信号的请求来决定是否保持或者变化当前的相位。请求触发规则如下：

- 1. 当目前信号相位的持续时间未达到最小时间周期时，或在当前相位对应入车道上有车辆进入，并且在接近信号的距离内时，就会产生延长绿色信号时间的请求，以让车辆可以直接通过路口。
- 2. 当竞争信号相位的等待车辆数量大于一个阈值时，就会生成对绿色信号的请求。

根据规则的差异，Actuated Control 主要可以分为 Fully-Actuated Control 和 Semi-Actuated Control 两种。

2.3.5 SOTL

Self-Organizing Traffic Light Control(SOTL) 是一种具有附加需求响应规则的 Fully-Actuated Control 方法。它与 Fully-Actuated Control 的主要区别在于当前信号相位的绿色信号请求定义（虽然它们都需要最小的绿色相位持续时间），在 Fully-Actuated Control 中，当车辆接近信号灯时，就会产生延长绿色信号的请求，而在 SOTL 中，除非接近信号灯的车辆数量大于预先定义的一个阈值，否则就不会产生绿色信号请求。

2.3.6 Max-Pressure Control

Max-Pressure Control 的目的是通过最小化对应信号相位的压力（pressure）来平衡相邻路口之间的队列长度，从而降低过饱和的风险，其中压力的概念如图图 2.3所示：其中运动信号的压力是指转入车道上的车辆数减去相应的转出车道上的车辆数，而信号相位的压力定义为转入巷道和转出巷道上的总队列长度之间的差异。Varaiya[1] 等人证明了当将优化目标设为最小化单个路口的相位压力时，Max-Pressure Control 可以最大限度地提高真个路网的吞吐量。

表 2.1列出了每种方法的要求和输出结果：

2.4 基于强化学习的交通信号控制

最近，人们提出了不同的人工智能技术来控制交通信号，例如遗传算法、群体智能以及强化学习。其中在这些技术中，强化学习在近年来更具趋势。

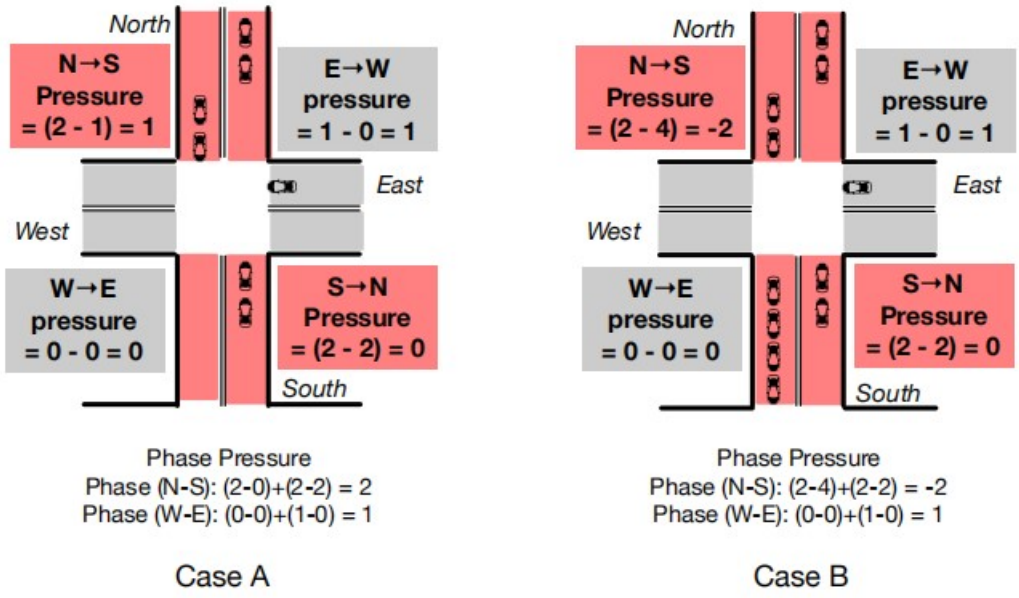


图 2.3　Max-Pressure 的压力图示

2.4.1　强化学习概述

通常单智能体强化学习问题被建模成马尔可夫决策过程（Markov Decision Process, MDP, 如图 2.4所示）$< \mathcal{S}, \mathcal{A}, P, R, \gamma >$，其中 $\mathcal{S}, \mathcal{A}, P, R, \gamma$ 分别表示状态集、动作集、概率状态转移函数、奖励函数和折扣因子。具体定义如下：

- $\mathcal{S}$：在时间步骤 t，智能体得到一个观测状态 $s^t \in \mathcal{S}$。
- $\mathcal{A}, P$：在时间步骤 t，智能体采取一个动作 $a^t \in \mathcal{A}$，然后环境根据状态

表 2.1　传统交通信号控制方法总结

方法	先验信息	输入	输出
Webster	相位序列	交通流量	基于周期的单个路口信号计划
GreenWave	信号计划	交通流量、速度限制、车道长度	基于周期的信号计划的偏移量
Actual Control，SOTL	相位序列	交通流量	是否变化到下一个相位
Max-Pressure Control	无	队列长度	所有交叉口的信号计划

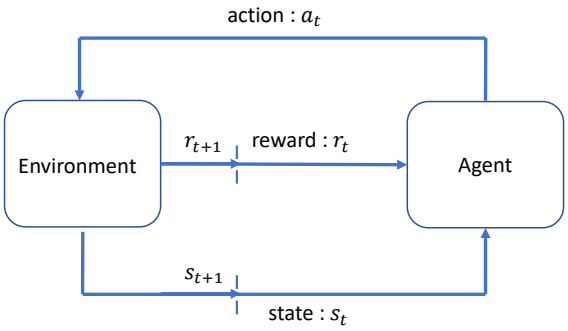


图 2.4　马尔可夫决策过程

转移函数转移到一个新的状态。

$$P\left(s^{t+1} \mid s^t, a^t\right) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \tag{2.2}$$

- R：在时间步骤 t，智能体通过奖励函数获得一个奖励 r^t。

$$R\left(s^t, a^t\right) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R} \tag{2.3}$$

- γ：智能体的目标是找到一种使预期收益最大化的策略，即累积（折扣）奖励之和。折扣因子决定了即时奖励与未来奖励的重要性。

$$G^t := \sum_{i=0}^{\infty} \gamma^i r^{t+i} \tag{2.4}$$

通常解决一个强化学习任务意味着要找到一个能够使预期收益最大化的最优策略 π^*，一般来说，我们难以直接找到这个最优策略，更多的是比较若干个不同的策略然后从中选出较好的那个作为局部最优解。而策略的筛选是通过比较其对应的价值函数来实现的，即通过寻找较优的价值函数来筛选出较优的策略。价值函数是对未来奖励的期望，根据输入的不同可以分为状态价值函数和动作价值函数。状态价值函数的定义如下：

$$V_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right] \tag{2.5}$$

其描述的是当在 t 时刻处于状态 s 的预期收益。动作价值函数的定义如下：

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] \tag{2.6}$$

其描述的是当在状态 s 下采取动作 a 的预期收益。

最优策略 π^* 对应的是最优状态价值函数和最优动作价值函数。最优状态价值函数定义为 $V_\pi^*(s) = \Updownarrow\neg\S_\pi V^\pi(s)$，它满足以下贝尔曼最优方程：

$$V^*\left(\mathrm{s}^t\right) = \max_{\mathrm{a}^t \in \mathcal{A}} \sum_{s^{t+1} \in \mathcal{S}} P\left(\mathrm{s}^{t+1} \mid \mathrm{s}^t, \mathrm{a}^t\right)\left[\mathrm{r} + \gamma v_*\left(\mathrm{s}^{t+1}\right)\right], \forall \mathrm{s}^t \in \mathcal{S} \tag{2.7}$$

最优动作价值函数定义为 $Q^*(s,a) = \Updownarrow\neg\S_\pi Q^\pi(s,a)$，其满足以下贝尔曼最优方程：

$$Q^*\left(s^t, a^t\right) = \sum_{s^{t+1} \in \mathcal{S}} P\left(s^{t+1} \mid s^t, a^t\right)\left[r^t + \gamma \max_{a^{t+1}} Q^*\left(s^{t+1}, a^{t+1}\right)\right], \forall s^t \in \mathcal{S}, a^t \in \mathcal{A} \tag{2.8}$$

2.4.1.1 分类

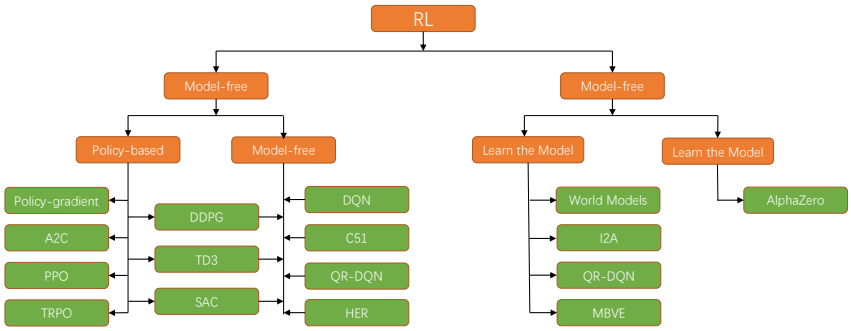
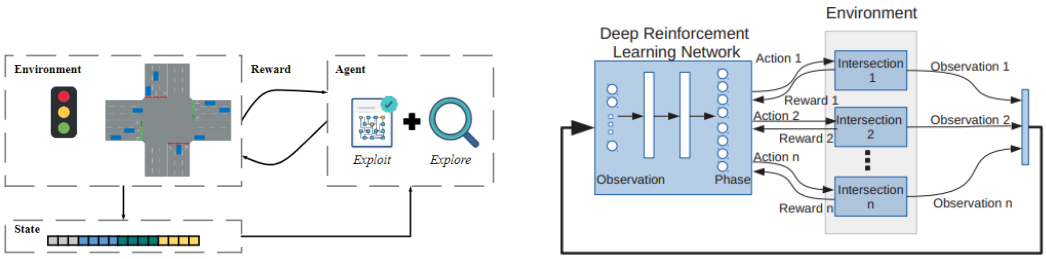


图 2.5 强化学习分类及常见算法

强化学习主要可以分为两大类：Model-based（有模型）和 Model-free（无模型）。其中 Model-based 又可以分为基于策略函数（Policy-based）和值函数



(a) 单路口场景　　(b) 多路口场景

图 2.6　基于强化学习的交通信号控制框架

（Value-based）两大类，而 Model-free 可以分为模型学习（Learn the Model）和给定模型（Given the Model）两大类，具体如图 2.5所示。

Model-free 方法是指在不知道状态转移函数函数的情况下，通过采样大量的经验来学习策略函数（Policy Function）或者价值函数（Value 函数）。其中 Policy-based 的方法直接将拟学习的策略参数化，在以最大化奖励的目标下，直接对策略函数进行优化。而 Value-based 的方法则是学习一个最佳策略对应的动作价值函数的近似。当然，Policy-based 和 Value-based 并非无法兼容的，有一类方法（Actor-Critic）就是融合了两中方法的思想。这类方法同时使用策略和价值评估来做出决策，其中，智能体会根据策略做出动作，而价值函数会对做出的动作给出评分，这样可以在原有的基于策略的方法的基础上加速学习过程，取得更好的效果。代表算法如图 2.5中的 DDPG[2]、TD3[3] 以及 SAC[4] 等。

Model-based 方法是指在已知状态转移函数的情况下，学习用一个模型去模拟环境，然后用这个模拟的环境去预测接下来可能会发生的所有所有情况并从中选择最有利的情况。

2.4.2　基于强化学习的交通信号控制框架

根据路网规模的不同，基于强化学习的交通信号控制框架可以分为以下两类：

- 单路口交通信号控制：图 2.6a描述了应用强化学习框架到单路口交通信

号控制问题上的基本思路。环境是道路上的交通状况，智能体要做的是控制交通信号。在每个调度时刻 t，获取环境的状态描述 s^t（例如，当前的信号相位，车辆的等待时间，排队长度，车辆的位置），智能体将根据这个状态描述对下一步采取的行动作出预测，以使预期收益最大化，然后该动作将在环境中执行，并且产生一个奖励 r^t。通常，在决策过程中，智能体采取的策略结合了对所学策略的利用和对新策略的探索。

- 多路口交通信号控制：图 2.6b描述了应用强化学习框架到多路口交通信号控制问题上的基本思路。智能体被定义为环境中 N 个路口的信号控制者。目标是学习每个智能体的最优策略，以优化整个环境中所有路口的通行效率。在每个调度时刻 t，每个智能体 i 观察环境的一部分作为观察点 o_i^t。智能体将对接下来要采取的行动 α^t 做出预测。这些动作将在环境中执行，并产生奖励 r_i^t，其中奖励可以在环境中的单个路口或一组路口的层面上定义。

2.4.3 基本要素

使用强化学习来解决交通信号控制问题要先确定以下几个基本要素:

奖励设计：由于强化学习是以最大化累计奖励为目标来学习的，所以奖励的选择决定了学习的方向。在交通信号控制问题中，虽然最终目标是尽量减少所有车辆的通行时间，但由于几个原因，通行时间很难直接作为 RL 的有效奖励。首先，车辆的行驶时间不仅受交通信号灯的影响，还受车辆自由流动速度等其他因素的影响。其次，当交通信号控制器事先不知道车辆行驶目的地（在现实世界中往往是这样），优化道路上所有车辆的通行时间变得特别困难。在这种情况下，车辆的通行时间只能在多个动作完成后车辆完全离开路口后才能测量。已有工作的奖励设计通常是基于一些可以直接在一个动作后测量的指标的加权和。例如，等待车辆的队列长度、车辆等待时间、速度、累计延迟、路口的吞吐量、车辆平均停车次数、信号变化频率（信号在一定时间段内变化的次数，学习到的策略不应该太过频繁的改变信号）以及路口的压力（Max-pressure

中定义的 pressure）等。虽然将奖励定义为几个因素的加权线性组合是现有研究中的一种常见做法，并且取得了不错的效果，但是这种特别的设计存在两个问题。第一，无法保证最大化设计的奖励等价于最优的通行效率，因为它们在交通运输理论中没有直接联系。第二，调整每个奖励函数因子的权重是相当棘手的，在权重设置上的微小差异可能会导致最终的结果有显著的差别。

状态表示：状态表示是以一种数值化的形式来描述路口的交通状况，描述的越全面越有利于快速学习到最优策略，通常使用多个要素组合来描述交通状况，例如，队列长度、车辆等待时间、车辆数量（包含非等待车辆），车辆速度、车辆位置分布以及当前信号灯的相位等。最近，在基于 RL 的交通信号控制算法中出现了使用更复杂状态的趋势，希望能够更全面地描述交通状况。Mousavi、Van derPol 以及 Wei Hua 等人在他们的研究工作中提出使用位置图片来当作状态描述。但是，具有如此高维度的状态学习往往需要大量的训练样本，这意味着训练 RL 智能体需要很长时间。更重要的是，较长的学习进度不一定会导致显著的性能增益，因为智能体可能需要花费更多的时间从状态表示中提取有用的信息。因此，状态的表示应该简洁且能够充分地描述环境。

动作选择机制：动作选择机制决定了以何种方式来控制信号灯，不同的动作机制有不同的影响。主要可以总结为以下四种方式：

- 确定当前相位时长：在这中动作选择机制下，智能体学习通过从预定义的候选时间段（比如，10 秒、15 秒、20 秒等）中选择来设置当前相位的持续时间。
- 确定基于周期的相位比：这种方式定义的动作为下一个周期的相位分裂比（phase split ratio）通常，给出总周期长度，并预先定义一个包含一些相位比的候选集。
- 保持或改变当前相位：这种方式也是基于周期性的信号计划，通常一个二进制数来定义动作。例如，1 表示保持当前相位，0 表示变换到下一相位。

- 选择下一个相位：这种方式直接从待选相位序列中选择一个相位并变化到该相位，其中相位序列不是预定的。因此，这种信号控制方式更加的灵活，智能体学习在不同的而状态下选择最优的相位，而不假设信号会以循环的方式改变。

2.5 图神经网络

由于我们的工作中涉及到图神经网络的知识，这里给出一些有关图神经网络的简单描述。

2.5.1 图神经网络概述

深度网络的研究推进了模式识别和数据挖掘领域的发展。借助于计算资源的高速发展（如 GPU），深度学习在欧几里得数据（如图像、文本和视频）中取得巨大的成功。但是在一些应用场景下，数据（图）是由非欧几里得域生成的，任然需要有效分析。例如，在电子商务领域，一个基于图的学习系统能够利用用户和商品之间的交互以实现精准的推荐。在化学领域，分子被建模为图，新药研发需要测定其生物活性。在论文引用网络中，论文之间通过引用关系互相连接，需要将它们分成不同的类别。

图数据的复杂性对现有机器学习算法提出了巨大的挑战，因为图数据是不规则的。每张图大小不同、节点无序，一张图中的每个节点都有不同数目的邻近节点，使得一些在图像中容易计算的重要运算（如卷积）不能再直接应用于图。此外，现有机器学习算法的核心假设是实例彼此独立。然而，图数据中的每个实例都与周围的其它实例相关，含有一些复杂的连接信息，用于捕获数据之间的依赖关系，包括引用、朋友关系和相互作用。最近，越来越多的研究开始将深度学习方法应用到图数据领域。受到深度学习领域进展的驱动，研究人员在设计图神经网络的架构时借鉴了卷积网络、循环网络和深度自编码器的思想。

图神经网络的概念最早由 Gori[5] 等人提出，由 Scarselli[6] 等人进一步阐明。早期的初期是以迭代方式通过循环神经网络架构传播邻近信息来学习目标

节点的表示，直至达到稳定的状态。

图神经网络可以分为：图卷积网络（Convolutional graph neural networks），图注意力网络（Graph Attention Network），图自编码器（Graph Auto-encoder），图生成网络（Graph Generative Network）和图时空网络（Graph Spatial-Temporal Network）。

2.5.2 图卷积网络

图卷积网络是将卷积运算从传统数据（如图片、视频）推广到了图数据上的模型，如图 2.7所示。其主要思想是通过聚合节点 v 自身的特征和邻居节点的特征来生成节点 v 的表示。图卷积网络在构建许多其他的图神经网络模型方面发挥了重要作用。图卷积网络按照卷积的方式可以分为两类：基于谱

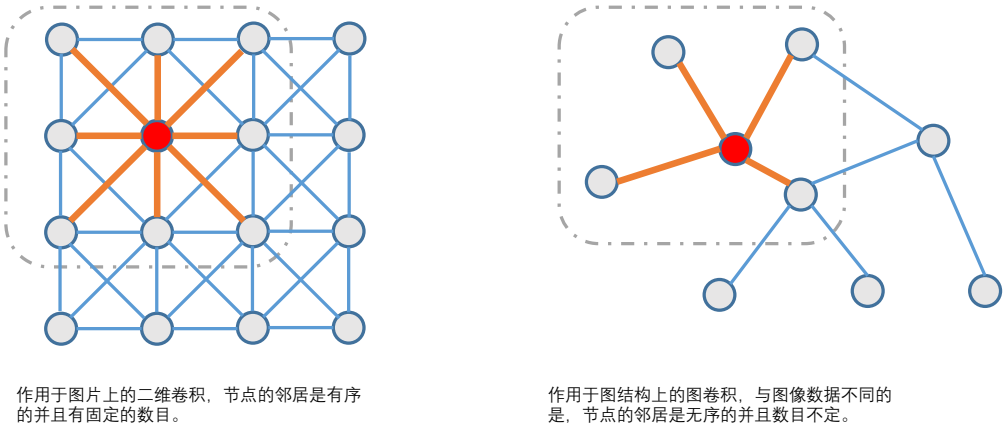


图 2.7 图卷积示意图

（spectral-based）和基于空间（spatial-based）的方法。基于谱的方法从信号处理的角度引入滤波器来定义图卷积 [82]，其中图卷积操作被解释为从图信号中去

除噪声。基于空间的方法则是通过信息传播来定义图卷积。

2.5.3 图注意力网络

图注意力网络是将注意力机制引入到基于空间域的图神经网络。图神经网络不需要使用拉普拉斯等矩阵进行复杂的计算，仅通过邻居节点的表征来更新目标节点的特征。由于能够放大数据中最重要部分的影响，注意力机制已经广泛应用到很多基于序列的任务中，图神经网络也受益于此，在聚合过程中使用注意力整合多个模型的输出。主要方法包括：

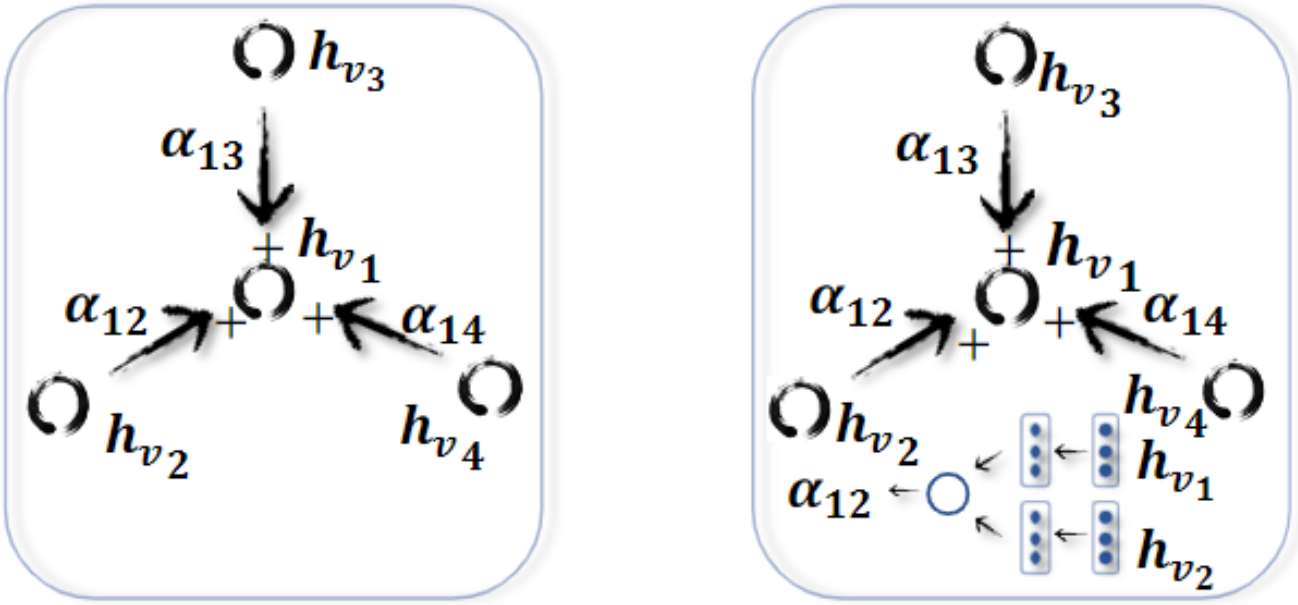


图 2.8 GCN 与 GAT 聚合信息的区别

- Graph Attention Network(GAT)[7]：本质上 GAT 是一种基于空间的图卷积网络，它与 GCN 的主要区别在于对邻居节点信息的聚合方式不同（图 2.8）。GCN 在在聚合过程中显式地为节点 v_i 的邻居 v_j 赋予一个非参数静态权重 $a_{ij} = \frac{1}{\sqrt{\deg(v_i)\deg(v_j)}}$。而 GAT 则是通过使用一个端到端的神经网络架构隐式地捕捉权重 a_{ij}，以便更重要的节点获得更大的权重，

具体操作如下：

$$\mathbf{h}_i^t = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha\left(\mathbf{h}_i^{t-1},\mathbf{h}_j^{t-1}\right)\mathbf{W}^{t-1}\mathbf{h}_j^{t-1}\right), \tag{2.9}$$

其中 $\alpha(\cdot)$ 是一个注意力函数，它可以动态地调整邻居节点 j 对目标节点 i 的贡献。通常为了学习不同子空间中的注意力权重，GAT 会使用多个注意力函数（即多头注意立机制，Multi-head Attention）：

$$\mathbf{h}_i^t = \|_{k=1}^K\sigma\left(\sum_{j\in\mathcal{N}_t}\alpha_k\left(\mathbf{h}_i^{t-1},\mathbf{h}_j^{t-1}\right)W_k^{t-1}\mathbf{h}_j^{t-1}\right), \tag{2.10}$$

- Gated Attention Network(GAAN)[8][9]：GAAN 除了采用多头注意力机制（self-attention mechanism）外，还引入了自注意机制来更新节点的隐藏状态。自注意机制可以为每个注意力头计算出一个额外的注意分数：

$$\mathbf{h}_i^t = \phi_o\left(\mathbf{x}_i\oplus\|_{k=1}^K g_i^k\sum_{j\in\mathcal{N}_t}\alpha_k\left(\mathbf{h}_i^{t-1},\mathbf{h}_j^{t-1}\right)\phi_v\left(\mathbf{h}_j^{t-1}\right)\right), \tag{2.11}$$

其中 ϕ_o和ϕ_v 是反馈神经网络，而 g_i^k 是第 k 个注意力头的权重。

- Graph Attention Model(GAM)：GAM 是一种用来解决图形分类问题的循环神经网络模型。它可以通过自适应地访问某个重要节点的序列来对图的信息进行处理，其模型定义如下：

$$\mathbf{h}_t = \mathbf{f}_h\left(\mathbf{f}_s\left(\mathbf{r}_{t-1},\mathbf{v}_{t-1},g;\theta_s\right),\mathbf{h}_{t-1};\theta_h\right), \tag{2.12}$$

其中 $\mathbf{f}_h(\cdot)$ 是一个 LSTM 网络，f_s 是一个 step network，它会优先访问当前节点 v_{t-1} 优先级高的邻居并将它们的信息进行聚合。

2.5.4 图自编码器

图自编码器是一类图嵌入方法，其目的是利用神经网络将图的顶点表示为低维向量。典型的解决方案是利用多层感知机作为编码器来获取节点嵌入，其中解码器重建节点的邻域统计信息，如 positive pointwise mutual information（PPMI）或一阶和二阶近似值。主要包括基于GCN的自编码器，如Graph Autoencoder（GAE）[10] 和 Adversarially Regularized Graph Autoencoder (ARGA)[11]，以及

Network Representations with Adversarially Regularized Autoencoders (NetRA)[12]、Deep Neural Networks for Graph Representations (DNGR)[13]、Structural Deep Network Embedding (SDNE)[14] 和 Deep Recursive Network Embedding (DRNE)[15]。DNGR 和 SDNE 学习仅给出拓扑结构的节点嵌入，而 GAE、ARGA、NetRA、DRNE 用于学习当拓扑信息和节点内容特征都存在时的节点嵌入。图自动编码器的一个挑战是邻接矩阵 A 的稀疏性，这使得解码器的正条目数远远小于负条目数。为了解决这个问题，DNGR 重构了一个更密集的矩阵，即 PPMI 矩阵，SDNE 对邻接矩阵的零项进行惩罚，GAE 对邻接矩阵中的项进行重加权，NetRA 将图线性化为序列。

2.5.5 图生成网络

图生成网络的目标是在给定一组观察到的图的情况下生成新的图。图生成网络的许多方法都是特定于领域的。例如，在分子图生成中，一些工作模拟了称为 SMILES 的分子图的字符串表示。在自然语言处理中，生成语义图或知识图通常以给定的句子为条件。这种方法通常使用 GCN 或者其他框架作为基础构建模块，其中使用 GCN 构建的方法有：

- Molecular Generative Adversarial Networks (MolGAN)[16]：MolGAN 整合了图卷积网络、图注意力网络和强化学习以生成具有预期属性的图。MolGAN 由生成器和判别器组成，相互竞争以提高生成器的真实性。在 MolGAN 中，生成器尝试提出一个假图及其特征矩阵，而鉴别器旨在将假样本与经验数据区分开来。此外，还引入了一个奖励网络来促使生成器能够按照外部的评估生成具有特定属性的图。
- Deep Generative Models of Graphs (DGMG)[17]：利用基于空间的图卷积网络来获得现有图的隐藏表示。生成节点和边的决策过程是以整个图的表示为基础的。简而言之，DGMG 递归地在一个图中产生一个节点，直到达到某个停止条件。在添加新节点后的每一步，DGMG 都会反复决定是否向添加的节点添加边，直到决策的判定结果变为假。如果决策为真，

则评估将新添加节点连接到所有现有节点的概率分布，并从概率分布中抽取一个节点。将新节点及其边添加到现有图形后，DGMG 将更新图的表示。

使用其他架构作为基础模块的图生成网络有：

- GraphRNN[18]：通过两个层次的循环神经网络的深度图生成模型。图层次的 RNN 每次向节点序列添加一个新节点，而边层次 RNN 生成一个二进制序列，指示新添加的节点与序列中以前生成的节点之间的连接。为了将一个图线性化为一系列节点来训练图层次的 RNN，GraphRNN 采用了广度优先搜索（BFS）策略。为了建立训练边层次的 RNN 的二元序列模型，GraphRNN 假定序列服从多元伯努利分布或条件伯努利分布。
- NetGAN[19]：Netgan 将 LSTM 与 Wasserstein-GAN 结合在一起，使用基于随机行走的方法生成图形。GAN 框架由两个模块组成，一个生成器和一个鉴别器。生成器尽最大努力在 LSTM 网络中生成合理的随机行走序列，而鉴别器则试图区分伪造的随机行走序列和真实的随机行走序列。训练完成后，对一组随机行走中节点的共现矩阵进行正则化，我们可以得到一个新的图。

2.5.6 图时空网络

许多现实世界的应用中的图在图结构和图输入方面都是动态的。图时空神经网络在捕捉图的动态性方面占据了重要地位。这类方法旨在为动态节点输入建模，同时假设连接节点之间的相互依存关系。例如，一个交通网络由放置在道路上的速度传感器组成，边缘权重由传感器对之间的距离决定。由于一条道路的交通状况可能取决于其相邻道路的状况，在进行交通速度预测时有必要考虑空间依赖性。作为一种解决方案，STGNNs 同时捕捉图的空间和时间依赖性。图时空网络可以分为两个方向，一种是基于 RNN 的方法，另一种是基于 CNN 的方法。

大多数基于 RNN 的方法通过过滤输入和使用图卷积传递给循环单元的隐

藏状态来捕获时空依赖性。但是基于 RNN 的方法存在耗时的迭代传播和梯度爆炸或者消失的问题。作为替代解决方案，基于 CNN 的方法以非递归的方式处理空间-时间图，具有并行计算、稳定梯度和低内存需求的优势。基于 CNN 的方法将一维卷积层和图卷积层交织在一起，分别学习时间和空间的依赖关系。

2.5.7 任务分类

以图结构和节点特征信息作为输入，根据输出的类别，可以将 GNN 的分析任务分为以下几类：

- 节点级别：节点级的输出与节点回归（Node Regression）和节点分类（Node Classification）任务相关。如图卷积网络可以通过信息传播和图卷积操作提取出节点的潜在表示。使用多感知器或 softmax 层作为输出层，GNN 能够以端到端的方式执行节点级任务。
- 边级别：边级别的输出与边分类（edge classification）和链接预测（link prediction）任务相关。以 GNN 的两个节点的潜在表示作为输入，可以利用相似性函数或神经网络来预测一个边的标签或者连接强度。
- 图级别：图级别的输出与图分类任务相关。通常 GNN 会与池化（pooling）和读出（read-out）操作相结合，以获得在图级别上的紧凑表示。

第三章 单路口场景智能交通信号调度

3.1 相关工作

最近，强化学习算法在交通信号控制领域表现出了优异的性能，这些算法将当前道路上的交通状况当作状态，并通过与环境交互学习操控信号的策略。现有的基于强化学习的交通信号控制方法之间的差异主要体现在以下这三个方面：状态表示、奖励设计以及学习算法。

状态表示是对当前环境的定量描述。一些常见的状态特征包括：

- 队列长度（Queue length）：队列长度是车道上处于“等待”状态的车辆的数量。对于车辆“等待”状态，有不同的定义。在 [20] 中，速度小于 0.1米/s 的车辆被认为处于等待状态；在 [21,22] 中，等待车辆是指没有移动位置的车辆。
- 等待时间 (Waiting time)：车辆的等待时间定义为车辆处于“等待”状态的时间段。等待期的开始时间的定义可能有所不同，在 [20,23]，他们认为等待时间是从车辆移动的最后一个时间戳开始，而 [24,25] 认为等待时间是从车辆进入路网开始的。
- 交通流量（Traffic volume）：交通流量定义为车道上的车辆数量，等于该车道上处于等待状态的车辆和行驶车辆的总和。
- 相位（Phase）：将相位信息作为状态特征，首先要将其进行量化，[20,23] 用当前相位在预先定义的信号相位组中的索引值来确定。

通常情况下，状态描述会整合多个特征，以获得对交通状况更全面的描述。

在交通信号控制问题中，尽管最终的目标是使所有车辆的行驶时间最小化，但由于一些原因，行驶时间很难直接作为 RL 的有效奖励。首先，车辆的行驶时间不单单受交通信号的影响，还与其他的因素有关，例如车辆的速度。

其次，当交通信号控制器事先不知道车辆的目的地时（现实世界中经常出现这种情况），优化网络中所有车辆的行驶时间变得尤为困难。在这种情况下，只有当网络中的多个交叉路口采取了多项行动时，才能在车辆完成行程后测量车辆的行驶时间。因此，奖励功能通常被定义为下列因素的加权和，这些因素在智能体采取动作后可以被即刻测量出。

- 总队列长度：这里队列长度是所有 incoming lanes 的队列长度之和。[26] 证明了最小化队列长度相当于最小化所有车辆的行驶时间。
- 吞吐量：吞吐量定义为在最后一个动作后的特定时间间隔内通过交叉路口或离开网络的车辆总数。
- 速度：一个典型的奖励是取道路网中所有车辆的平均速度，平均速度越高意味着车辆行驶到目的地的速度越快，时间也就越短。
- 信号变化频率：信号改变的频率被定义为在某一时间段内信号改变的次数。直观地说，学到的政策不应该导致闪烁，即频繁地改变交通信号，因为车辆通过交叉口的有效绿色时间可能会减少。

3.2 已有工作中的不足

虽然目前已经有不少基于强化学习的智能交通信号控制的研究，但是已有的方法更多的只注重提高通行效率，例如最小化队列长度或者最大化吞吐量，而忽略了公平性问题。事实上，这样的目标会导致学习到一个有偏见的策略。例如可能会出现如图 3.1所示的"last-vehicle" 情况：N-S 方向的道路上有源源不断的车辆到达，而 E-W 方向上是该车流的最后一辆车（Last-Vehicle）。显然，为了最大化通行效率，会优先放行 N-S 方向上的车流，而 E-W 车道上的车要等所有的所有的车流通过才能够得到响应，这对 Last-Vehicle 来说是极其不公平的。

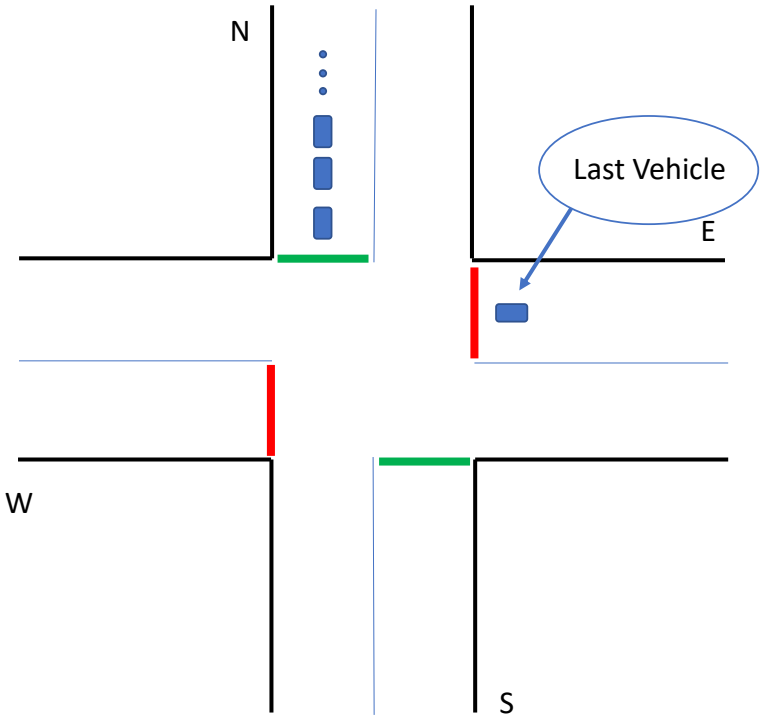


图 3.1 last vehicle

3.3 改进

3.3.1 目标

本工作的目的是在提高通行效率（最小化平均通行时间）的同时，希望每条车道能够有尽可能相同的服务延迟（得到放行所需的时间）。

3.3.2 智能体设计

状态表示

在 t 时刻的状态 $S(t)$ 由以下几个部分组成：

1. 交通流量：$\boldsymbol{V}(t) = \{V_1(t), V_2(t), \cdots, V_M(t)\}$。其中 $V_i(t)$ 表示第 i 条进近车道上车的数量。值得注意的是，由于右转不受限于信号灯的特殊性，这里我们不考虑右车道的交通流量。
2. 平均吞吐量：$\overline{\boldsymbol{L}}(t) = \{\overline{L}_1(t), \overline{L}_2(t), \cdots, \overline{L}_M(t)\}$。其中 $\overline{L}_i(t)$ 表示第 i 条进

近车道的平均吞吐量。同上，不考虑右车道的平均吞吐量。

3. 信号相位：$\boldsymbol{P}(t)$ 是当前信号相位的数字化表示，1 表示绿色，可以通行；0 表示红色，禁止通行。

所以 $S(t) = \{\boldsymbol{V}(t)||\overline{\boldsymbol{L}}(t)||\boldsymbol{P}(t)\}$

动作选择

在本文中，动作选择机制是每次选择即将转换的信号相位。之后，交通信号灯将转换到这一新的相位并持续 Δt 的时间。为了安全起见，我们在两个不同的信号相位之间插入了 3 秒的黄色信号和 2 秒的红色信号。如果新选择的相位和当前相位相同，则不插入黄色和红色信号，以确保交通流畅。

奖励函数

受 PFS 分配原则的启发，我们设计了一个可以在效率和公平之间提供良好的平衡的奖励函数，如下所示：

$$r = -\sum_{i=1}^{M} \frac{Q_i(t)}{\overline{L}_i(t) + \delta}, \tag{3.1}$$

其中 $Q_i(t)$ 和 $\overline{L}_i(t)$ 分别是第 i 条进近车道的队列长度和平均吞吐量。在每一次调度后（这里，我们将一次动作选择视作一次调度），$\overline{L}_i(t)$ 按照以下方式进行更新：

$$\overline{L}_i(t) = (1 - \frac{1}{W})\overline{L}_i(t-1) + \frac{1}{W}L_i(t), \tag{3.2}$$

其中 $L_i(t)$ 是此次调度中车道 i 上得到放行的车的数量，W 是一个平衡通行效率和公平性的参数。另外，为了避免公式3.1的分母为 0，我们加上了一个可以忽略不计的正数 δ。

模型框架

如图 3.2所示，这里我们用 DQN 作为学习算法，并且采用经验回放[27]（experience replay）方法定期提取样本来更新模型，具体算法过程如算法 1所示。

Algorithm 1 基于 DQN 的交通信号控制训练流程

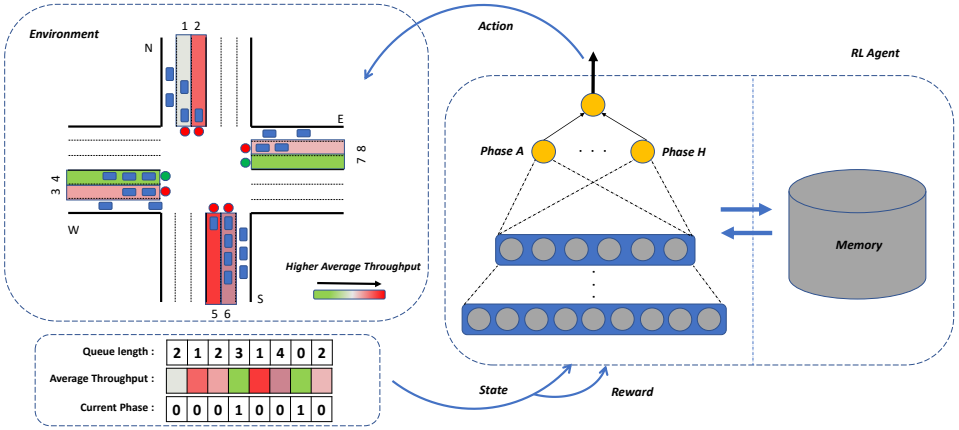


图 3.2　系统模型

输入： E：学习片段数

T：每个学习片段的步数

b：学习经验数

ϵ：随机选择动作概率

γ：折扣因子

Δt：信号维持时间

for $episode = 1, E$ **do**

　初始化环境。

　生成车辆。

　for $t = 1, T$ **do**

　　从环境中获取状态观测 s_t。

　　生成一个 0 到 1 之间的随机数 $rand$。

　　if $rand < \epsilon$ **then**

　　　从动作空间中随机采样一个动作 a_t。

　　else

　　　使用 DQN 模型选择动作：$a_t = \arg\max_a Q(s_t, a; \theta)$。

end if

将当前信号更改为 a_t 并维持 δt 秒时间。

更新每条车道的平均吞吐量。

环境转移到新的状态 s_{t+1} 并返回一个奖励 r_{t+1}。

将经验 $(s_t, a_t, r_{t+1}, s_{t+1})$ 存储到经验回放池 M 中。

if $|M| > b$ **then**

从经验回放池 M 中随机采样 b 条经验数据。

计算损失函数 $\mathcal{L}_j$: $\mathcal{L}_Q = [r_{t+1} + \gamma \arg\max_{a'} Q(s_{t+1}, a'; \theta) - Q(s_t, a_t; \theta)]^2$;

更新 DQN 参数。

end if

end for

end for

3.4 实验

实验在 SUMO（simulation of Urban MObility）[1]仿真平台上进行，利用该模拟器可以方便地实时获取车辆状态，并通过改变交通信号来控制交通运行。我们实现了一个四路交叉口作为我们的实验场景，交叉口与四个 150 米长的路段相连，每条道路有三条引入车道和三条引出车道。

我们将 N-S 方向的道路设置为主干道，车辆到达量更多，将 W-E 方向的道路设置为次干道，车辆到达量较少。车辆到达服从泊松分布，这里我们设置 N-S 方向道路的交通流量比率为 ρ，W-E 方向道路的交通流量比率为 $1-\rho$，ρ 值越高，交通流量不平衡的状况越严重。为了对我们的方法进行综合评价，我们在不同的 ρ 值下进行了实验。注意，为了简化环境，这里我们不考虑行人交通的影响。

3.4.1 评价指标

我们使用以下指标来评估不同方法的效率和公平性表现：

[1]http://sumo.dlr.de/index.html

- 行驶时间：车辆行驶时间是指车辆进出路口的时间差。现有的大部分工作都集中在最小化所有车辆通过交叉路口的平均行驶时间。
- 延误时间：车辆延误时间是车辆通过交叉路口的实际时间与预期时间（以最高限速通过交叉路口所需的时间）之间的差值。
- 驾驶体验得分：此外，我们提出了一种新的评价指标，称为驾驶体验得分（Driving Experience Score，DES），来量化驾驶员的满意度，具体评分标准见下表：事实上，可能有更多的因素需要考虑（如燃油消耗），但是这里的目的是为了缓解车辆的过度延误情况，因此我们这里用延误时间作为评价标准。

表 3.1 驾驶体验得分标准

延误时间（s）	DES
$d \leq 40$	5
$40 < d \leq 80$	4
$80 < d \leq 120$	3
$120 < d \leq 160$	2
$d > 160$	1

3.4.2 比较方法

- FT(Fixed-Time Control[28])：这种方法以预先设定的方式循环改变信号。
- SOTL(Self-Organizing Traffic Light Control[29])：这是一种根据预先设定的阈值来改变信号的自适应方法。如果等待的车辆数量超过了这个阈值，则切换到下一个信号相位。
- LIT[26]：这是一种基于学习的方法，比大多数现有的致力于提高通行效率的方法效果更好。
- FIT(**F**airness-aware **I**ntelligent **T**raffic Light Control)：我们的方法。

3.4.3 性能表现

首先，我们通过实验评估了不同方法的通信效率的表现，为了得到一个综合的结果，我们在不同的 ρ 值下进行了实验，实验结果如图 3.3所示。可以观察到，我们的方法（FIT）的车辆通过路口的平均行驶时间远低于传统方法（行驶时间越短意味着效率越高），并且仅略低于只注重效率的 LIT 方法。

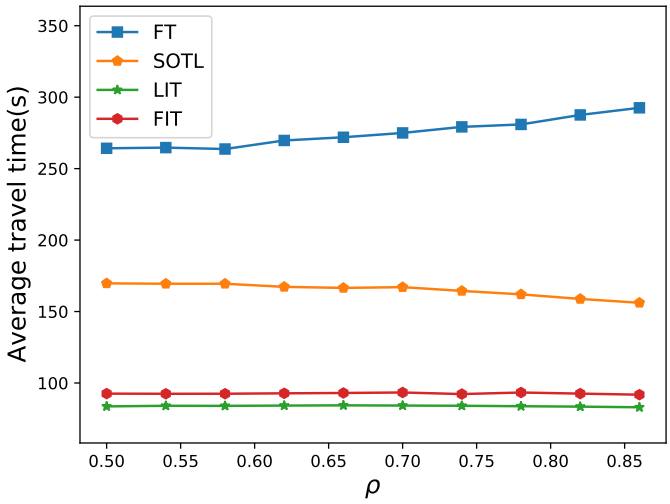


图 3.3 效率

其次，由于我们的主要研究目标是公平性，我们首先分析了在使用不同方法下每条车道的延误情况。这里我们用 Jain Fairness Index(JFI) 来量化公平性指标，JFI 的计算方式如下：

$$\mathcal{J} = \frac{\left(\sum_{i=1}^{M} \bar{D}_i\right)^2}{M \sum_{i=1}^{M} \bar{D}_i^2}, \tag{3.3}$$

其中 $\bar{D}_i$ 是车道 i 的平均延误时间。当每个车道具有相同的平均延误时间时，JFI 的值达到最大值，即 1。图 3.4展示了四种方法在不同 ρ 值下的平均延迟的 JFI 表现。从中我们可以看出 FT 和 LIT 的 JFI 值随着交通不平衡情况的加剧（即 ρ 值越大）而减小，而 FIT 任然能偶保持较高的值，并且高于同样能够保持稳定 JFI 值的 SOTL 方法。

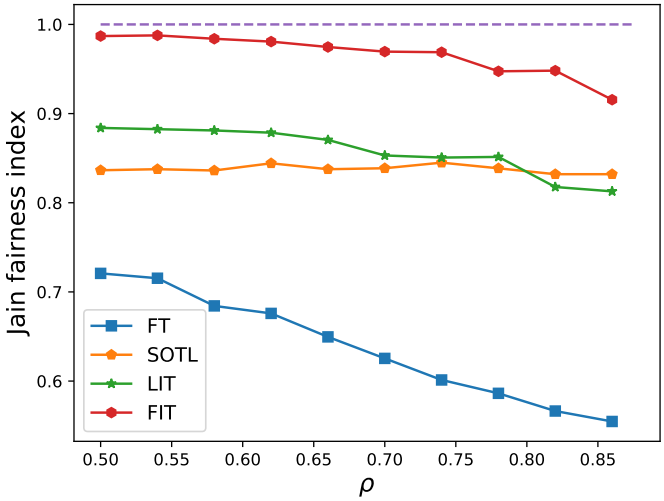


图 3.4　公平

然后，我们更加详细地研究了不同方法的延误情况。下面我们具体分析在主干道和支干道上四种方法在不同的 ρ 值情况下车辆延误时间分布情况。从图 3.5中我们可以看出，在主干道上，基于学习的方法（LIT 和 FIT）比传统方法（FT 和 SOTL）具有更低的延误时间，虽然 SOTL 方法整体上延迟也比较低，但是会有很多极端值，最高的延误时间甚至超过 800s。

从图 3.6中我们可以看出，在支干道上，随着 ρ 值的增加（即交通不平衡情况的加重），原先在主干道上表现优异的 LIT 方法性能开始恶化 ()，但是 FIT 依然能够保持一个相对低的延迟。

我们研究了不同方法的驾驶体验得分情况，图 3.7展示了在 $\rho = 0.75$ 的情况下不同方法的驾驶体验得分分布情况。从中我们可以看出，FT 方法超过半数的驾驶体验的分都是 1 分，由此可以看出该方法的不灵活。对于 SOTL 方法而言，虽然他的 5 分的比例最高，但是其得分分布的方差也是最高的。FIT 的得分分布与 LIT 相似，但 FIT 的方差低于 LIT，在以牺牲少量效率为代价的前提下。

最后，我们研究了 W 参数的影响，因为 W 是用来平衡效率和公平性的，

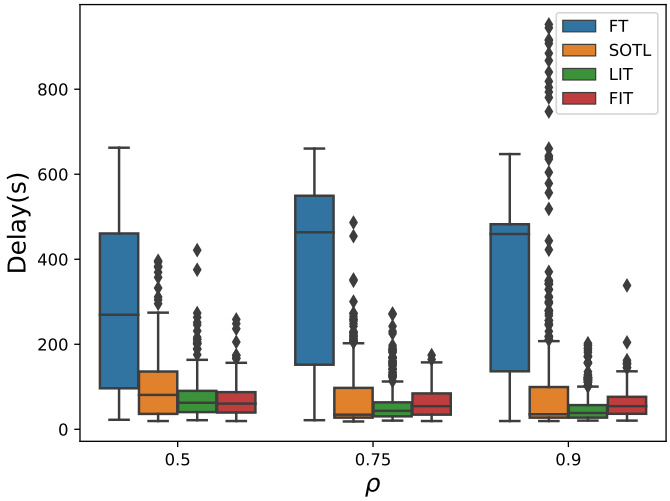


图 3.5　主干道 (N-S 方向) 的车辆延误时间分布

不同的值会导致学习到不同的策略，从图 3.8中我们可以看出 W 值越高，越有利于系统的公平性。相反，W 值越小，系统效率就越高。具体来说，当 $W = 1$ 时，我们方法 FIT 的整体性能接近于 LIT

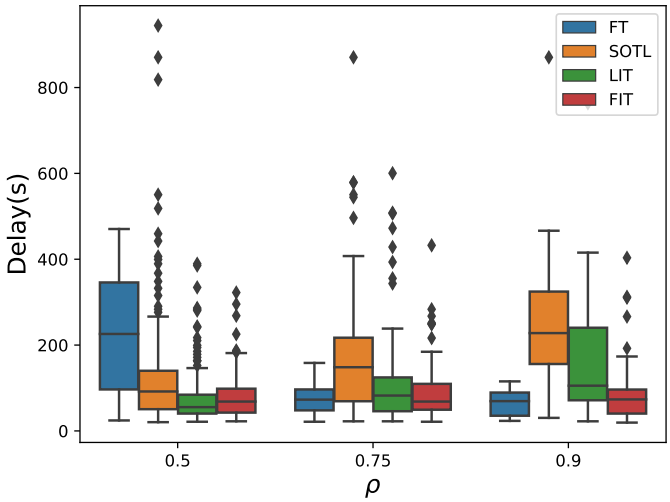


图 3.6　支干道 (N-S 方向) 的车辆延误时间分布

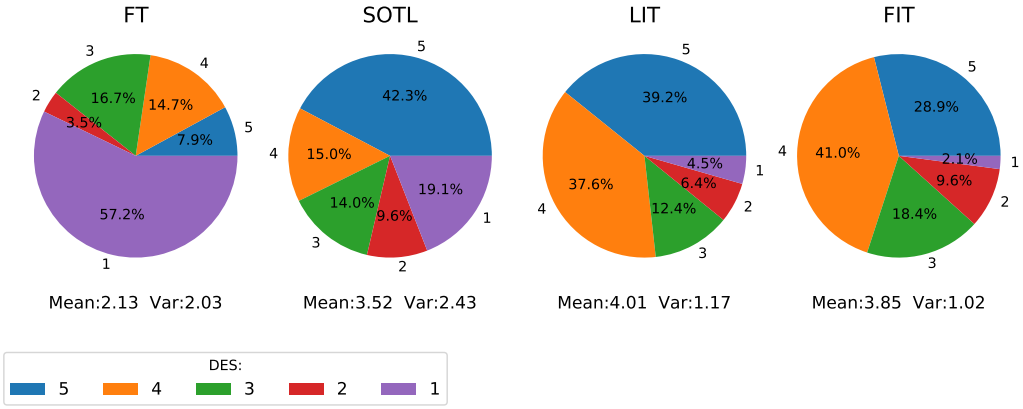
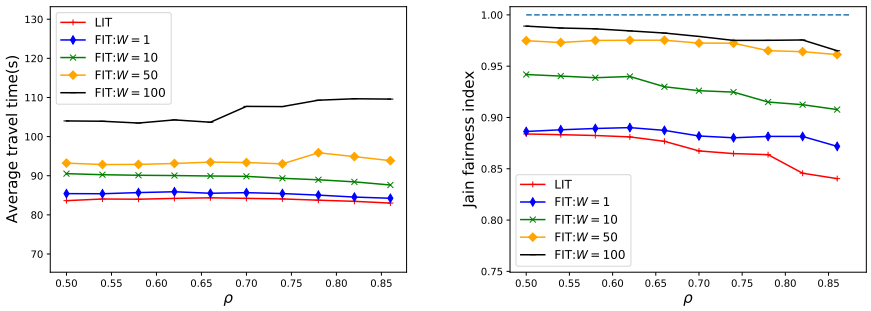


图 3.7　驾驶体验得分统计



(a) 效率　　(b) 公平性

图 3.8　W 对效率和公平性的影响

第四章　多路口场景智能交通信号调度

4.1　相关工作

由于强化学习在单路口交通信号控制上取得了优异的成绩，人们开始致力于使用多智能体强化学习（Multi-Agent Reinforcement Learning, MARL）来解决多路口场景下的交通信号调度。Claus 在 [30] 中将 MARL 分为了两类：联合动作学习（Joint Action Learning）和独立学习（Independent Learning）。

对于多路口信号控制，联合动作学习的思想就是使用一个全局智能体（single global agent）来控制所有的交叉路口，其动作是所有路口动作组合在一起的联合动作，然后通过迭代学习建模多个智能体的联合动作价值函数（Joint Action Value Function）：

$$Q(o_1, o_2, \cdots, o_N, \mathbf{a}) \tag{4.1}$$

其中 o_i 是智能体 i 对路口环境的观测，$\mathbf{a}$ 是所有智能体的联合动作。但是这种方法的缺点是会导致维度灾难（curse of dimensionality），状态动作的联合空间会随着智能体数量的增加呈指数级增长，增加学习的难度。为了缓解这个问题，[23] 使用 max-plus 方法将联合动作价值函数分解为局部子问题的线性组合，如下所示：

$$\hat{Q}\left(o_1, \ldots, o_N, \mathbf{a}\right) = \Sigma_{i,j} Q_{i,j}\left(o_i, o_j, \mathbf{a}_i, \mathbf{a}_j\right) \tag{4.2}$$

其中 i和j 对应于相邻智能体的索引。在 [31–33] 中，将联合 Q 值视为局部 Q 值的加权和：

$$\hat{Q}\left(o_1, \ldots, o_N, \mathbf{a}\right) = \Sigma_{i,j} w_{i,j} Q_{i,j}\left(o_i, o_j, \mathbf{a}_i, \mathbf{a}_j\right) \tag{4.3}$$

其中 $w_{i,j}$ 是预先定义的权重。他们试图通过在单个智能体的学习过程的损失函数中增加一个整形项，并使单个 Q 值的加权和与全局 Q 值的差异最小化，从而确保单个智能体在学习过程中能够考虑到其他智能体的情况。

多路口信号控制的另一条研究路线是使用独立的 RL（IRL）智能体来控制交通信号，其中每个 RL 智能体控制一个路口。与联合动作学习方法不同，每个智能体可以在不知道其他智能体的奖励信号的情况下学习控制策略。根据智能体之间是否进行信息交互进一步分为以下两类：

- IRL without Communication：IRL 单独处理每个交叉口，每个 agent 观察自己的本地环境，不使用显式通信来解决冲突 [25,34–39]。在一些简单的场景中，如动脉网络，这种方法表现良好，可以形成了几个小绿波（Green waves）。然而，当环境变得复杂时，来自相邻 agent 的非平稳影响将被带到环境中，如果 agent 之间没有通信或协调机制，学习过程通常无法收敛到平稳策略。为了应对这一挑战，wei 在 [40] 中提出了一个特定的奖励函数，去描述相邻智能体之间的需求从而实现协调。
- IRL with Communication：这种方法使智能体之间能够就他们的观察进行交流，并作为一个群体而不是个体的集合来完成复杂的任务，在这种情况下，环境是动态的，每个智能体的能力和对世界的可见度是有限的 [41]。典型的方法是直接将邻居的交通状况 [42] 或过去的动作 [43] 加入到自身智能体的观察中，而不是仅仅使用自我观测到的本地交通状况。在这种方法中，不同路口的所有智能体共享一个学习模型，这就需要对相邻的路口进行一致的索引。[44] 试图通过利用图卷积网络的路网结构来消除这一要求，以协作附件的多跳路口的交通，并且通过图卷积网络中定义的固定邻接矩阵来模拟相邻代智能体的影响，这表明他们假设相邻智能体之间的影响是静态的。在其他工作中，[45,46] 提出使用图注意网络来学习相邻智能体和自我智能体的隐藏状态之间的动态相互作用。应该指出的是，利用 max-plus 学习联合行动学习者的方法和利用图卷积

网络学习通信的方法之间有很强的联系，因为它们都可以被看作是学习图上的信息传递，其中前一种方法传递奖励，后一种方法传递状态观测信息。

4.2 已有工作中的不足

目前大多数工作在使用图神经网络 Learn to Communicate 的时候，都是以 intersection 为节点来进行图建模，将每一个路口视作图中的一个节点，每条道路作为连接两个节点的边，很自然地可以将一张交通道路网建模成一个图，如图 4.1所示：在这种建模方式下，每条车道的车辆以及当前的相位将作为该

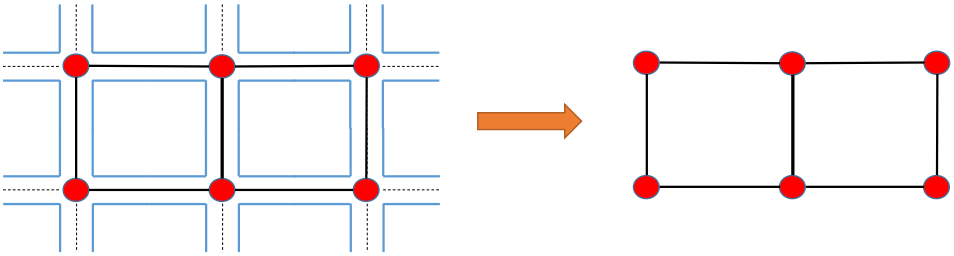


图 4.1 以路口为节点的建图方式示意图

节点的特征。这种建模方式虽然可以很清晰的将多路口场景变成一张图。但是，因为是以一个路口为一个节点，所有车道的状态信息都整合到了一起，有些车道的的信息对目标节点是无用的，如图 4.2所示：路口 B 中只有 2 车道的交通

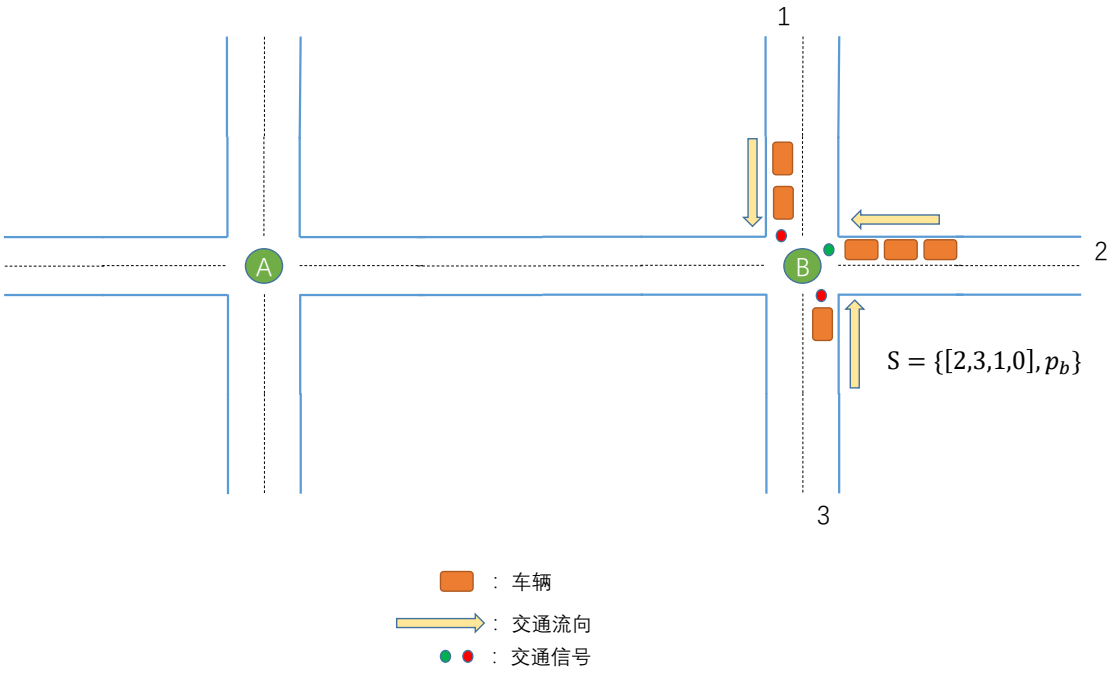


图 4.2　在以路口为节点的建图方式下的信息传递示意图

流向与 A 车道有关，1、3 车道的车辆不会行驶到 A 路口。在信息传递的时候，如果将所有的信息都笼统地传递过去，将会增加 A 提取有效信息的难度，从而降低学习的效率。

4.3　改进

4.3.1　目标

我们的目标是在进行信息交互的时候能够剔除与目标节点无关的信息，从而降低目标节点聚合邻居节点信息的难度，提高学习效率。

4.3.2　基于道路的图建模方式

本文同样采用 IRL with Communication 的框架，与已有工作不同的是，我们采用不同的建模方式：以道路为节点进行图建模，即一条道路就是一个节点，如图 4.3所示：

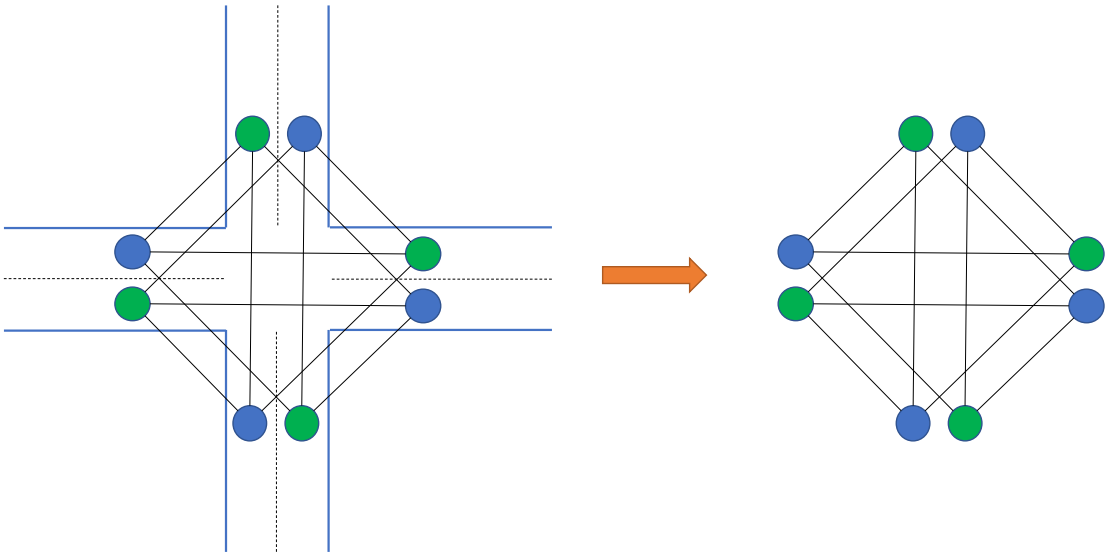


图 4.3　以道路为节点的建图方式示意图

此外，我们根据当前的信号相位对图的边设一个权重。这里我们规定，如果在当前相位下，道路 i 到道路 j 之间的交通是允许通行的，则表示 (i, j) 的状态是'connected'。权重的定义方法如下：

$$w_{i,j} = \begin{cases} 1 & (i, j) \text{ is connected} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

这个权重将被用于剔除对目标节点无用的信息。

4.3.3　Learn to Communicate

这里我们沿用 [45,46] 工作使用图注意网络，但是与之不同的是，我们不是学习相邻智能体和目标智能体的隐藏状态之间的动态相互作用，而是用来做节点回归（Node Regression），即估计目标节点在下一个时间点的特征。具体包括以下几个过程：

- 重要性计算：为了了解节点 j（源节点）的信息在确定节点 i（目标节点）下一时刻的特征的重要性，我们首先嵌入两个节点的特征，然后计算他们之间的相关系数 e_{ij}（节点 j 在确定节点 i 的特征时的重要性），具体操作如下：

$$e_{ij} = a\left(\left[Wh_i \| Wh_j\right]\right) \tag{4.5}$$

其中 W 是一个共享参数，用来进行特征增强，然后用 $[\cdot\|\cdot]$ 对节点 i 和节点 j 增强后的特征进行拼接，最后使用 $a(\cdot)$ 将拼接后的高维特征映射到一个实数上。

- 特征筛选：由于当前交通信号的影响，道路 j 的车辆不会进入到道路 i，即节点 j 的信息在确定节点 i 的下一时刻的特征时是无用的，因此我们要筛选掉对目标节点无用的信息。这里我们通过之前介绍的边的权重来实现：

$$e_{ij} = e_{ij} * w_{i,j}, \tag{4.6}$$

如果 $WE_{ij} = 1$（即道路 j 到道路 i 的交通在当前相位下是可以通行的），e_{ij} 将维持之前的计算结果。反之，如果 $w_{i,j} = 0$，将清除节点 j 对节点 i 的影响。

- 注意力分布计算：为了重新确定源节点和目标节点之间的注意力值，我们进一步将目标节点 i 和其邻近节点之间的交互等分进行归一化：

$$\alpha_{ij} = \text{softmax}\left(e_{ij}\right) = \frac{\exp\left(e_{ij}/\tau\right)}{\sum_{j\in\mathcal{N}_i}\exp\left(e_{ij}/\tau\right)}, \tag{4.7}$$

其中 τ 是一个温度系数，$\mathcal{N}_i$ 是目标节点 i 邻近范围的节点集合。

- 特征回归：为了确定目标节点在下一个时刻的特征，这里我们将其所有邻近节点的信息按照各自的重要性进行组合：

$$h_i' = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}Wh_j\right) \tag{4.8}$$

其中 $\sigma(\cdot)$ 是激活函数。h_i 是 i 节点融合了邻域信息后的新特征。

- Multi-Head Attention：进一步，我们使用多头注意力机制（Multi-Head Attention）来关注不同相关性下的信息，如下所示：

$$h_i'(K) = \sigma\left(\frac{1}{K}\sum_{k=1}^{k=K}\sum_{j\in N_i}\alpha_{ij}^k W^k h_j\right) \tag{4.9}$$

其中 K 是注意力头的数量，例如当 $k=3$ 时，结构如图图 4.4所示：

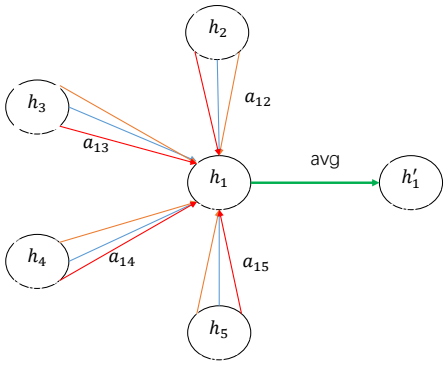


图 4.4　多头注意力计算示意图（$k=3$）

对于每个路口，我们要估计每条引进道路在下一调度时刻的状态，即我们要对四个节点进行计算，如图 4.5中 A 路口的四个红色节点。通过在目标节点（红色节点）的及其邻近节点（绿色节点）构成的子图（如图图 4.5中右上角的图）上对目标节点进行上述的计算。由于我们是以道路为节点进行建图的，所以即便是单个路口也可以表示成一个图，所以当有多个路口的时候，会产生一张很大的图。这里每个节点维护自己路口的子图，包括更新子图中的节点特征以及边的权重（根据当前路口的信号相位确定）。

4.3.4　模型框架

如图 4.6所示，对于单个路口来说，有两个模型，一个是用来进行交通信号控制的 DQN 模型 Q，另一个是用来预测下一调度时刻状态的 GAT 模型 G。

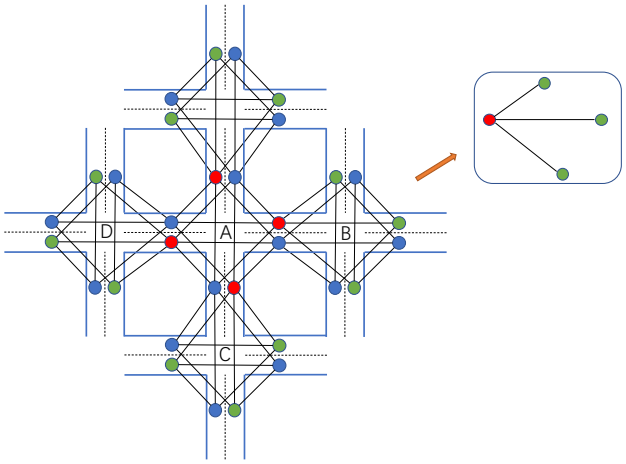


图 4.5　节点回归计算示意图

对于 DQN 模型来说，每次根据输入的状态来选择动作，其中状态由以下几部分组成：

- Queue length：当前路口每条车道的队列长度。
- Traffic volume：当前路口每条车道的车辆数。
- Current Phase：路口当前的相位。
- Next Queue length：通过节点回归估计的下一调度时刻的 Queue length。
- Next Traffic Volume：通过节点回归估计的下一调度时刻的 Volume。

其损失函数如式 4.10所示：

$$\mathcal{L}_t = [r_{t+1} + \gamma \arg\max_{a'} Q(s_{t+1}, a'; \theta) - Q(s_t, a_j; \theta)]^2 \tag{4.10}$$

对于 GAT 模型来说，首先我们规定节点的特征 $f_t = \{q_t, v_t\}$，其中 q_t 和 v_t 分别时队列长度（Queue length）和交通流量（Traffic volume）。当环境转移到新的状态 s_{t+1} 时，更新节点特征（$f_{t+1} = \{q_{t+1}, v_{t+1}\}$）以及边的权重，其损失函数如下所示：

$$\mathcal{L}_G = [f_{t+1} - f'_{t+1}]^2 \tag{4.11}$$

其中 f'_{t+1} 是在 t 调度时刻，利用邻近节点信息预估的下一时刻的节点特征，是预测值，而 f_{t+1} 是 $t+1$ 时刻的真实节点特征。

具体的算法流程如算法 2所示。

Algorithm 2 multi-intersection

输入： E：学习片段数

T：每个学习片段的步数

b：学习经验数

ϵ：随机选择动作概率

γ：折扣因子

Δt：信号维持时间

for $episode = 1, E$ **do**

  初始化环境。

  for $t = 1, T$ **do**

    从环境中获取当前状态观测 $s_t = q_t, v_t, p_c$。

    使用 GAT 估计下一调度时刻的节点特征 $f'_{t+1} = \{q'_{t+1}, v'_{t+1}\}$。

    生成一个 0 到 1 之间的随机数 $rand$。

    if $rand < \epsilon$ **then**

      从动作空间中随机采样一个动作 a_t。

    else

      使用 DQN 模型选择动作：$a_t = \arg\max_a Q((s_t || f'_{t+1}), a; \theta)$。

    end if

    将当前信号更改为 a_t 并维持 δt 秒时间。

    环境转移到新的状态 s_{t+1} 并返回一个奖励 r_{t+1}。

    更新节点特征 $f_{t+1} = q_{t+1}, v_{t+1}$。

    计算 GAT 损失函数 $\mathcal{L}_G$: $\mathcal{L}_G = [f_{t+1} - f'_{t+1}]^2$

    更新 GAT 模型参数。

    将经验 $(s_t, a_t, r_{t+1}, s_{t+1})$ 存储到经验回放池 M 中。

18: **if** $|M| > b$ **then**

19: 从经验回放池 M 中随机采样 b 条经验数据。

20: 计算 DQN 损失函数 $\mathcal{L}_Q$: $\mathcal{L}_Q = [r_{t+1} + \gamma \arg\max_{a'} Q(s_{t+1}, a'; \theta) - Q(s_t, a_t; \theta)]^2$;

21: 更新 DQN 模型参数。

22: **end if**

23: **end for**

end for

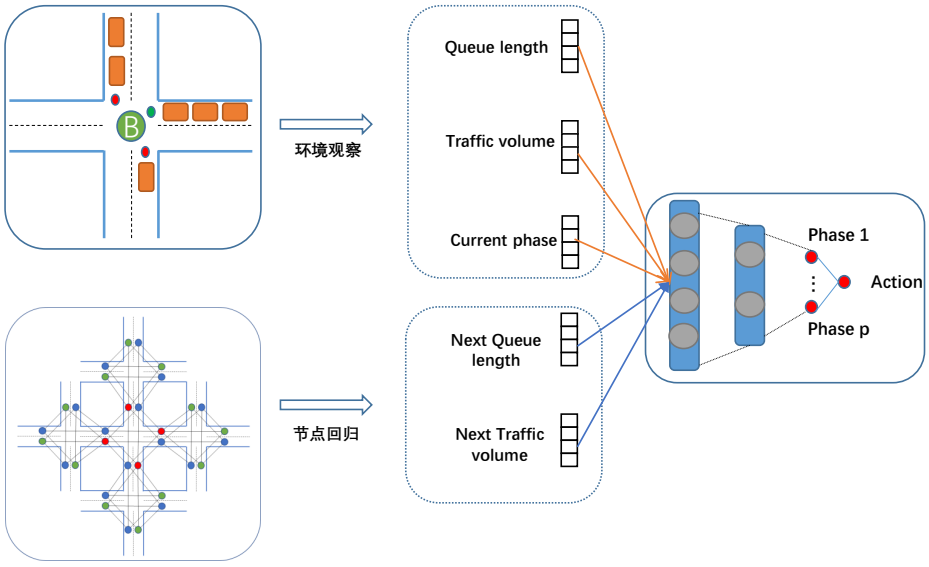


图 4.6 模型框架

4.4 实验

我们在三个合成的数据集以及两个真实数据集上对我们的方法进行了实验，以评估我们方法的性能。由于实验是针对大规模的多路口场景，我们这里的仿真工具使用的是 CityFlow[1]，其对大规模交通信号控制的支持更加出色。

4.4.1 数据集介绍

合成数据：延用 [45] 的做法，我们使用了以下类中合成的交通数据：

- $Grid_{6x6} - b$：6x6 的路网结构，其中每个路口有四个方向（西 → 东、东 → 西、南 → 北以及北 → 南），每个方向有三条车道（长 300 米，宽 3 米）。车辆在西 ↔ 东方向的生成速率是 300 辆/车道/小时，在南 ↔ 北方向的生成速率是 90 辆/车道/小时。
- $Grid_{6x6}$：与 $Grid_{6x6}$ 的路网结构相同，但是只有单向的车辆流动，即只有西 → 东和北 → 南的单向交通流动。

真实数据：我们还使用了杭州和济南两个城市某个路段下采集到的真实数据[2]进行了实验，表 4.1统计了这两个数据集的关键信息。

表 4.1 数据统计

数据集	路口数量	车辆到达率（300 辆/s）			
		均值	方差	最大值	最小值
$D_{Hangzhou}$	16	526.63	86.70	676	256
D_{Jinan}	12	250.70	38.21	335	208

- $D_{Hangzhou}$：这个数据集中有 16 个路口，其中交通数据是由路侧监控摄像头拍摄产生，每条数据包含时间、摄像头 ID 和车辆信息。通过使用摄像头位置分析这些记录，记录车辆通过道路交叉口时的轨迹。我们以通

[1] http://cityflow-project.github.io

[2] https://traffic-signal-control.github.io/#open-datasets

过这些路口的车辆数作为实验的交通量。

- D_{Jinan}：与 $D_{Hangzhou}$ 类似，数据集中包含了 12 个路口。

:

4.4.2 比较方法

我们将我们的方法与传统交通控制方法和基于强化学习的几种方法进行了对比：

- FT[47]: 这种方法以预先设定的方式循环改变信号。
- MaxPressure[1]：交通领域最先进的网络级的交通信号控制方法，每次调度时，选择压力最大的相位。
- Individual RL[20]：一种基于深度强化学习的交通信号控制方法，不考虑邻居信息。每个路口由一个智能体控制，智能体之间不共享参数，而是独立更新自己的网络。
- GCN[23]：一种基于深度强化学习的交通信号控制方法，使用图卷积神经网络（GCN）提取相邻路口的交通特征，不过其图建模方式是以路口为节点。
- Colight[45]：与 GCN 方法类似，都是以路口为节点的图建模方法，不过其使用 GAT 来学习不同路口之间的动态交互。
- GAT-Road：我们的方法，基于一种新的图建模方式（以道路为节点）。

4.4.3 性能表现

我们在合成数据和真实数据上进行了大量实验，并和已有方法进行了对比，然后在通行效率和模型收敛性上进行了分析。

4.4.3.1 性能比较

效率比较：表 4.2列出了所有方法在合成数据和真实数据上路口的平均通行时间，可以看出，无论是在合成数据还是真实数据集上，我们的方法都取得了最好的表现。进一步的观察数据我们可以看出：与传统交通控制中最先进的方法 (MaxPressure) 相比，我们的方法在合成数据和真实数据上都取得了一致

表 4.2 不同方法在合成数据集和真实数据集上关于平均通行时间的表现

方法	$Grid_{6x6}-Uni$	$Grid_{6x6}-Bi$	$D_Hangzhou$	D_Jinan
Fixedtime	209.68	209.68	728.79	869.85
MaxPressure	186.07	194.96	422.15	361.33
Individual RL	314.82	261.60	345.00	325.56
GCN	205.40	272.14	768.43	625.66
CoLight	173.79	170.11	297.26	291.14
GAT-Road	**156.32**	**161.56**	**290.44**	**279.67**

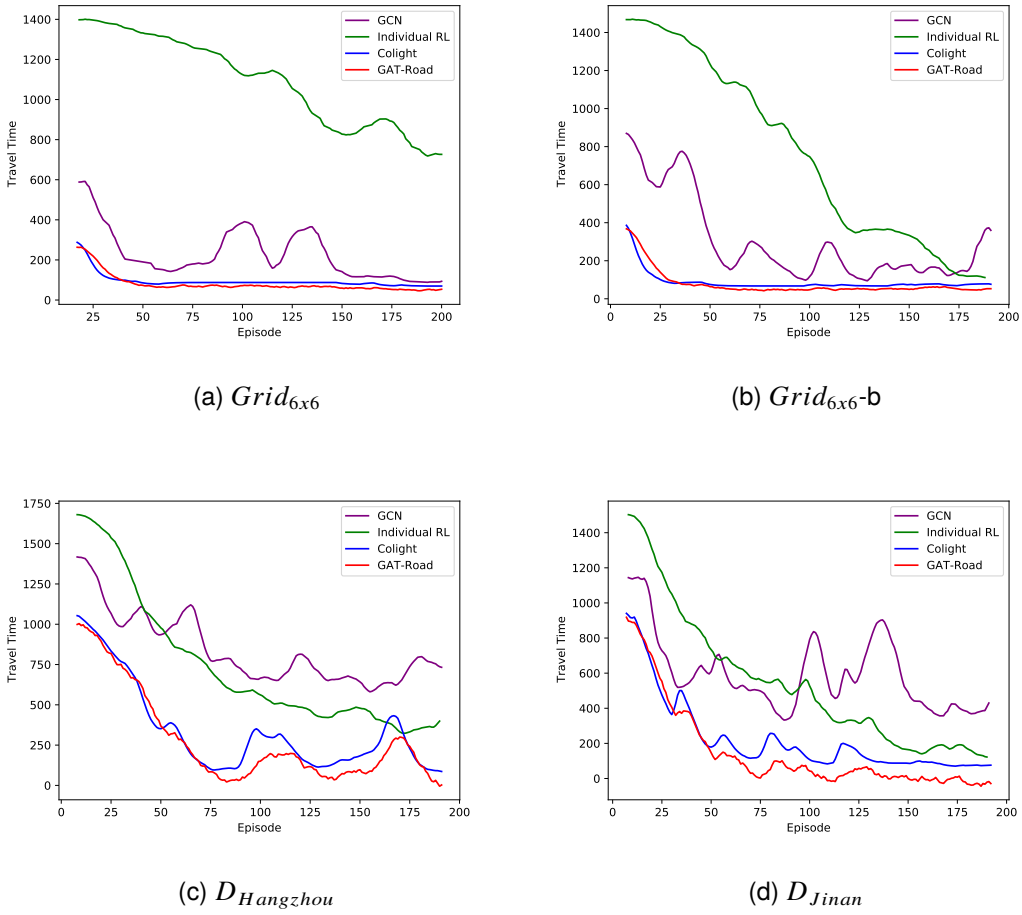
的性能改进，其中在合成数据上的平均改进率是 16.6%，在真实数据上的平均改进率是 26.9%。

在合成数据上，Fixedtime 的表现相对比较出色，但是在真实数据上，和基于学习的方法的性能差异变得明显。这是因为人为合成的数据是相对比较规则的数据，车辆流动的随机性没有真实数据中强，交通相对更加稳定。

GCN 在真实数据上的表现并不出色是因为其在聚合邻居节点的信息时，是根据预先定义的静态权重处理邻居节点的信息，而不是根据实时的交通情况进行动态调整，当面对到变化性较强的真实数据时，这种静态处理方式会导致无法正确地聚合信息。

虽然 CoLight 也是使用 GAT 学习邻居节点的动态交互，但是由于其是以路口为节点来进行建图的，在数据交互的时候传递的是路口的所有交通信息，导致目标节点在聚合信息时难以挖掘有效的信息。而我们的方法 GAT-Road 以道路为节点进行建模，并且根据相位信息剔除掉了对目标节点无效的信息，因此表现更加出色。

收敛性比较：我们将我们的方法 GAT-Road 与其他三种基于学习的方法（Individual、GCN 和 Colight）分别合成数据和真实数据上的学习收敛速度进行了比较，结果如图 4.7所示通过对比我们可以看出：与同样是使用 IRL with



(a) $Grid_{6x6}$ (b) $Grid_{6x6}$-b

(c) $D_{Hangzhou}$ (d) D_{Jinan}

图 4.7 GAT-Road 和其他三种基于学习的方法的收敛速度

Communication 框架的 GCN 和 Colight 相比，我们的方法的收敛速度更快，这得益于我们提出的新的建图方式，在这种建图方式下，节点在进行信息聚合时可以剔除与目标节点无关的信息，从而降低了学习的难度，模型更容易收敛。

GCN 方法在合成数据上收敛效果较好，但是在真实数据上收敛效果差的原因和上一节效率结果分析中的一样，静态的处理邻居路口的交通导致其难以聚合准确的信息。

虽然 Individual 最终也能收敛到最佳性能，但是由于其是独立优化单个路口的策略，没有考虑到周围路口环境的交通信息，因此其刚开始时的性能表现

相较于其他三种方法更差，并且收敛速度更慢。

第五章 总结与展望

本文研究了基于深度强化学习的智能交通信号控制，并对已有工作在不同场景下的不足进行了分析，并各自提出了新的方法。

对于单路口场景下的交通信号控制，已有的基于学习的方法更多的注重于提高通行效率，而忽略了公平性问题。在本文中，我们提出了一个新的模型可以在保证通行效率的同时，兼顾到对公平性的考虑。通过大量的实验验证了我们方法的有效性。

对于多路口场景下的交通信号控制，我们使用了 IRL with communication 的框架，并提出了一种新的将道路网建模成图的建模方式，在这种建模方式下，智能体在提取邻近节点的信息时可以剔除那些对自己无用的信息。通过实验验证，我们的方法在通行效率和收敛性上都优于已有的方法。

虽然目前有很多使用强化学习来解决智能交通信号调度的工作，但是这些动作都只停留在仿真阶段，即实验场景的搭建以及效果的验证都是通过仿真器完成的。要想将这些模型部署到现实生活中的信号灯上还需要更多的研究和实地测试。

参考文献

[1] Varaiya P. The max-pressure controller for arbitrary networks of signalized intersections[M]. . Proceedings of Advances in Dynamic Network Modeling in Complex Transportation Systems. Springer, 2013: 27–66.

[2] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015..

[3] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]. Proceedings of International Conference on Machine Learning. PMLR, 2018. 1587–1596.

[4] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]. Proceedings of International conference on machine learning. PMLR, 2018. 1861–1870.

[5] Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains[C]. Proceedings of Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 2. IEEE, 2005. 729–734.

[6] 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.

[7] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017..

[8] Lee J B, Rossi R, Kong X. Deep graph attention model[J]. arXiv preprint arXiv:1709.06075, 2017..

[9] Zhang J, Shi X, Xie J, et al. Gaan: Gated attention networks for learning on large and spatiotemporal graphs[J]. arXiv preprint arXiv:1803.07294, 2018..

[10] Kipf T N, Welling M. Variational graph auto-encoders[J]. arXiv preprint arXiv:1611.07308, 2016..

[11] Pan S, Hu R, Long G, et al. Adversarially regularized graph autoencoder for graph embedding[J]. arXiv preprint arXiv:1802.04407, 2018..

[12] Yu W, Zheng C, Cheng W, et al. Learning deep network representations with adversarially regularized autoencoders[C]. Proceedings of Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 2663–2671.

[13] Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations[C]. Proceedings of Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.

[14] Wang D, Cui P, Zhu W. Structural deep network embedding[C]. Proceedings of Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016. 1225–1234.

[15] Tu K, Cui P, Wang X, et al. Deep recursive network embedding with regular equivalence[C]. Proceedings of Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018. 2357–2366.

[16] De Cao N, Kipf T. MolGAN: An implicit generative model for small molecular graphs[J]. arXiv preprint arXiv:1805.11973, 2018..

[17] Li Y, Vinyals O, Dyer C, et al. Learning deep generative models of graphs[J]. arXiv preprint arXiv:1803.03324, 2018..

[18] You J, Ying R, Ren X, et al. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models[J]. 2018..

[19] Bojchevski A, Shchur O, Zügner D, et al. NetGAN: Generating Graphs via Random Walks[J]. 2018..

[20] Wei H, Zheng G, Yao H, et al. Intellilight: A reinforcement learning approach for intelligent traffic light control[C]. Proceedings of Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 2496–2505.

[21] Steingrover M, Schouten R, Peelen S, et al. Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion.[C]. Proceedings of BNAIC, 2005. 216–223.

[22] Kuyer L, Whiteson S, Bakker B, et al. Multiagent reinforcement learning for urban traffic control using coordination graphs[C]. Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2008. 656–671.

[23] Pol E, Oliehoek F A. Coordinated deep reinforcement learners for traffic light control[J]. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016), 2016..

[24] Brys T, Pham T T, Taylor M E. Distributed learning and multi-objectivity in traffic light control[J]. Connection Science, 2014, 26(1):65–83.

[25] Pham T T, Brys T, Taylor M E, et al. Learning coordinated traffic light control[C]. Proceedings of Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-13), volume 10. IEEE, 2013. 1196–1201.

[26] Zheng G, Zang X, Xu N, et al. Diagnosing reinforcement learning for traffic signal control[J]. arXiv preprint arXiv:1905.04716, 2019..

[27] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540):529–533.

[28] Miller A J. Settings for fixed-cycle traffic signals[J]. Journal of the Operational Research Society, 1963, 14(4):373–386.

[29] Cools S B, Gershenson C, D' Hooghe B. Self-organizing traffic lights: A realistic simulation[M]. . Proceedings of Advances in applied self-organizing systems. Springer, 2013: 45–55.

[30] Claus C, Boutilier C. The dynamics of reinforcement learning in cooperative multiagent systems[J]. AAAI/IAAI, 1998, 1998(746-752):2.

[31] Zhang Z, Yang J, Zha H. Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization[J]. arXiv preprint arXiv:1909.10651, 2019..

[32] Chu T, Wang J, Codecà L, et al. Multi-agent deep reinforcement learning for large-scale traffic

signal control[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(3):1086–1095.

[33] Tan T, Bao F, Deng Y, et al. Cooperative deep reinforcement learning for large-scale traffic grid signal control[J]. IEEE transactions on cybernetics, 2019, 50(6):2687–2700.

[34] Mannion P, Duggan J, Howley E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control[J]. Autonomic road transport support systems, 2016. 47–66.

[35] Casas N. Deep deterministic policy gradient for urban traffic light control[J]. arXiv preprint arXiv:1703.09035, 2017..

[36] Zheng G, Liu H, Xu K, et al. Learning to simulate vehicle trajectories from demonstrations[C]. Proceedings of 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020. 1822–1825.

[37] Liu X Y, Ding Z, Borst S, et al. Deep reinforcement learning for intelligent transportation systems[J]. arXiv preprint arXiv:1812.00979, 2018..

[38] Calvo J A, Dusparic I. Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights Control.[C]. Proceedings of AICS, 2018. 2–13.

[39] Gong Y, Abdel-Aty M, Cai Q, et al. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning[J]. Transportation Research Interdisciplinary Perspectives, 2019, 1:100020.

[40] Wei H, Chen C, Zheng G, et al. Presslight: Learning max pressure control to coordinate traffic signals in arterial network[C]. Proceedings of Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019. 1290–1298.

[41] Sukhbaatar S, Fergus R, et al. Learning multiagent communication with backpropagation[J]. Advances in neural information processing systems, 2016, 29:2244–2252.

[42] Xu M, Wu J, Huang L, et al. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning[J]. Journal of Intelligent Transportation Systems, 2020, 24(1):1–10.

[43] Ge H, Song Y, Wu C, et al. Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control[J]. IEEE Access, 2019, 7:40797–40809.

[44] Nishi T, Otaki K, Hayakawa K, et al. Traffic signal control based on reinforcement learning with graph convolutional neural nets[C]. Proceedings of 2018 21st International conference on intelligent transportation systems (ITSC). IEEE, 2018. 877–883.

[45] Wei H, Xu N, Zhang H, et al. Colight: Learning network-level cooperation for traffic signal control[C]. Proceedings of Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019. 1913–1922.

[46] Wang Y, Xu T, Niu X, et al. STMARL: A spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control[J]. IEEE Transactions on Mobile Computing, 2020..

[47] Koonce P, Rodegerdts L. Traffic signal timing manual.[R]. Technical report, United States. Federal Highway Administration, 2008.

致　谢

在此感谢对本论文作成有所帮助的人。