基于强化学习的
住宅电力负荷隐私保护协同控制

(申请清华大学工学硕士学位论文)

培 养 单 位 ： 自动化系

学　　科 ： 控制科学与工程

研 究 生 ： 秦 兆 铭

指 导 教 师 ： 曹 军 威　研究员

二〇二二年五月

Privacy-Preserving Reinforcement Learning for Cooperative Residential Load Control

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Control Science and Engineering

by

Qin Zhaoming

Thesis Supervisor: Professor Cao Junwei

May, 2022

学位论文公开评阅人和答辩委员会名单

公开评阅人名单

耿华	教授	清华大学
董炜	副研究员	清华大学

答辩委员会名单

主席	张涛	教授	清华大学
委员	耿华	教授	清华大学
	胡坚明	副教授	清华大学
	黄高	副教授	清华大学
	尚超	助理教授	清华大学
秘书	吕文祥	工程师	清华大学

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）按照上级教育主管部门督导、抽查等要求，报送相应的学位论文。

本人保证遵守上述规定。

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

在可再生能源主导的微电网中，负荷控制是实现功率平衡、降低微网运行成本的重要手段。作为一种数据驱动的方法，强化学习因其能适应不确定的电力需求而在负荷协同控制中被寄予厚望。然而在涉及到住宅用户的场景中，住户对自身数据隐私的担忧阻碍了强化学习的进一步应用。因此，本文针对强化学习在住宅负荷协同控制中的隐私保护问题展开研究，主要研究结果如下：

提出了孤岛微电网场景中适用于强化学习的住宅负荷模型，并探索了强化学习的输入选择问题。设计了结合监督学习和强化学习的有显式预测的能量管理方案，以及基于端到端强化学习的无预测的能量管理方案。实验结果说明，如果显式预测环节不引入强化学习的输入之外的信息，那么预测模块不会对强化学习的性能产生正面影响。该结论给强化学习在能量管理和负荷控制领域的应用提供了指导。

设计了微电网运营商集中管理的住宅负荷隐私保护协同控制方案。为了解决大量可控电力设备引起的高维动作空间难题，引入基于差分奖励的信用分配机制发展出向量强化学习算法，显著降低了值函数估计方差，提高了强化学习训练稳定性和效率。为了缓解微电网运营商对系统状态的部分可观测问题，将循环神经网络融入强化学习的策略网络和值函数网络，增强了强化学习的信息提取能力和决策能力。仿真实验说明了提出的方案相较于其他集中式隐私保护方案以更小的代价实现了对住户部分隐私数据的保护。

设计了云边环境下的分布式住宅负荷隐私保护协同控制方案。为了严格保护住户数据隐私且降低云边通信成本，提出了分散式执行者-分布式评价者（DADC, Decentralized Actors-Distributed Critics）的多智能体强化学习框架，打破了多智能体强化学习集中式训练、分散式执行的传统范式。仿真结果表明，DADC 框架在负荷协同控制效果上显著优于独立执行者-评价者框架，且在隐私保护和通信成本低等优势下，达到了与传统分散式执行者-集中式评价者框架相当的控制效果。

关键词：强化学习；协同控制；电力负荷；数据隐私；住宅微电网

Abstract

In the microgrids dominated by renewable energy, load control is an important means to achieve power balance and reduce the operation cost of microgrid. As a data-driven method, reinforcement learning is highly expected in load cooperative control because it can adapt to uncertain power demand. However, in the scenario involving residential users, residents' concerns about their own data privacy hinder the further application of reinforcement learning. Therefore, this paper studies the privacy protection of reinforcement learning in residential load cooperative control. The main research results are as follows:

A residential load model suitable for reinforcement learning in the isolated island microgrid scenario is proposed, and the state selection of reinforcement learning is explored. An energy management scheme with explicit prediction combined with supervised learning and reinforcement learning and an energy management scheme without prediction based on end-to-end reinforcement learning are designed. The experimental results show that if the explicit prediction link does not introduce information other than the input of reinforcement learning, the prediction module will not have a positive impact on the performance of reinforcement learning. Therefore, in the design of reinforcement learning algorithm, the current observation can be directly used as the input of reinforcement learning network. The conclusion provides guidance for the application of reinforcement learning in the field of energy management and load control.

A cooperative control scheme for privacy protection of residential load centrally managed by microgrid operators is designed. Without obtaining the key privacy data of households such as indoor temperature, microgrid operators coordinate the economic cost of microgrid and household power demand, and control the controllable load of users including air conditioning and electric vehicles. In order to solve the problem of high-dimensional action space caused by a large number of controllable power equipment, the credit allocation mechanism based on differential reward is introduced to develop vector reinforcement learning algorithm, which significantly reduces the estimation variance of value function and improves the stability and efficiency of reinforcement learning training. In order to alleviate the partial observability of system state by microgrid operators, the cyclic neural network is integrated into the strategy network and value function network of

reinforcement learning, which enhances the information extraction ability and decision-making ability of reinforcement learning. Simulation results show that compared with other centralized privacy protection schemes, the proposed scheme realizes the protection of some private data of residents at a lower cost.

The cooperative control scheme of distributed residential load privacy protection in cloud-edge environment is designed. The residential load on the edge side is controlled by its home energy management system. All home energy management systems realize the cooperative control of load through cloud edge communication without exposing local observation. In order to strictly protect the privacy of household data and reduce the cost of cloud side communication, a multi-agent reinforcement learning (MARL) framework, decentralized actors- distributed critic (DADC) is proposed, which breaks the traditional paradigm of centralized training and decentralized execution of MARL. The simulation results show that the DADC framework is significantly better than the independent executor evaluator framework in load cooperative control, and achieves the control effect equivalent to the traditional decentralized executor centralized evaluator framework under the advantages of privacy protection and low communication cost.

Keywords: Reinforcement learning; cooperative control; load control; data privacy; residential microgrid

目 录

摘 要.....I
Abstract.....II
目 录.....IV
插图清单.....VII
附表清单.....VIII
符号和缩略语说明.....IX
第 1 章 引言 1
1.1 研究背景和意义 1
1.2 强化学习研究现状 2
1.2.1 单智能体强化学习 2
1.2.2 多智能体强化学习 4
1.3 住宅电力负荷协同控制研究现状 5
1.3.1 数据驱动的住宅负荷控制方法 5
1.3.2 隐私保护的住宅负荷控制方法 6
1.4 本文主要工作 7
1.4.1 研究思路 7
1.4.2 主要工作 7
第 2 章 系统建模和强化学习输入研究 9
2.1 本章引言 9
2.2 强化学习预备知识 9
2.2.1 马尔可夫决策过程 9
2.2.2 策略评估 10
2.2.3 典型强化学习算法 11
2.3 系统建模 14
2.3.1 光伏功率模型 14
2.3.2 基础负荷模型 14
2.3.3 可调节负荷模型 15
2.3.4 可转移负荷模型 15

2.3.5 可控发电机和储能设备模型……16
2.4 强化学习输入研究……17
2.4.1 一般能量管理问题……17
2.4.2 有预测环节的强化学习……18
2.4.3 无预测环节的强化学习……19
2.4.4 性能评估……22
2.4.5 结论……24
第 3 章 微网运营商集中管理下的隐私保护负荷控制方案……25
3.1 总体思路……25
3.2 场景设计……26
3.2.1 场景描述……26
3.2.2 部分可观测马尔可夫决策过程建模……26
3.3 算法设计……28
3.3.1 向量 A2C 算法……29
3.3.2 隐私保护的负荷协同控制方案……30
3.4 仿真实验……33
3.4.1 实验设置……33
3.4.2 基线方案……34
3.4.3 仿真结果……35
3.5 本章小结……40
第 4 章 云边环境下的分布式隐私保护负荷控制方案……41
4.1 本章引言……41
4.2 场景设计……42
4.2.1 场景描述……42
4.2.2 分散式部分可观测马尔可夫过程建模……43
4.3 算法设计……45
4.3.1 网络结构……45
4.3.2 内部结构……46
4.3.3 分布式训练：在线策略优化……47
4.3.4 分布式训练：离线策略优化……48
4.4 仿真实验……52
4.4.1 实验设置……52
4.4.2 基线框架……53

4.4.3 主要结果.....54
4.4.4 负荷协同控制效果.....56
4.4.5 离线策略和在线策略对比.....58
4.5 本章小结.....58
第 5 章 总结与展望.....60
5.1 研究工作总结.....60
5.2 未来研究展望.....61
参考文献.....62
附录 A 补充内容.....66
致 谢.....67
声 明.....68
个人简历、在学期间完成的相关学术成果.....69
指导教师学术评语.....70
答辩委员会决议书.....71

插图清单

图 2.1 光伏功率数据示意图 14
图 2.2 基础负荷功率数据示意图 15
图 2.3 一般微网能量管理结构图 18
图 2.4 有预测和无预测的微电网能量管理方案 18
图 2.5 监督学习训练曲线和预测效果 22
图 2.6 有预测和无预测的强化学习训练曲线 23
图 2.7 无预测强化学习方案下储能设备充放电功率与电价的关系 24
图 3.1 微网运营商直接管理的负荷协同控制方案 26
图 3.2 基于向量 A2C 算法的隐私保护负荷协同控制方案流程图 32
图 3.3 不同方案的训练曲线 36
图 3.4 向量 A2C 算法和 A2C 算法的值估计损失曲线 36
图 3.5 不同方案控制下住户热舒适度损失和空调耗能 37
图 3.6 不同方案一天内的室内温度变化曲线 38
图 3.7 负荷调度曲线 39
图 3.8 所提方案和基线方案 4 控制下的电动汽车剩余待充电量曲线 40
图 4.1 用于家庭电力负荷协同控制的云边环境 43
图 4.2 云边环境下的多智能体强化学习网络 45
图 4.3 本地 Actor、本地 Critic 和前馈网络的内部结构 47
图 4.4 云边环境下 DADC 框架的离线策略训练示意图 48
图 4.5 各个框架的训练曲线 55
图 4.6 训练过程的值函数估计损失曲线 55
图 4.7 一天内的温度变化 56
图 4.8 一天内的负荷调度情况 57
图 4.9 基于环境采样次数的离线策略和在线策略效果对比 57
图 4.10 基于网络更新次数的离线策略和在线策略效果对比 58

附表清单

表 1.1　强化学习算法分类 2
表 2.1　监督学习的预测效果 23
表 3.1　室内温度转移函数和用户热舒适度损失函数的参数 34
表 3.2　电动汽车充电相关参数 34
表 3.3　测试集上的平均损失 35
表 4.1　边缘侧住宅参数 52
表 4.2　DADC 框架和 IAC 框架的测试效果.....56

符号和缩略语说明

DQN	深度 Q 网络（Deep Q-Network）
AC	执行者-评价者算法（Actor-Critic）
A2C	优势执行者-评价者算法（Advantage Actor-Critic）
PPO	近端策略优化算法（Proximal Policy Optimization）
SAC	软执行者-评价者算法（Soft Actor-Critic）
DDPG	深度确定性策略梯度算法（Deep Deterministic Policy Gradient）
MLP	多层感知器（Multi-Layer Perceptron）
RNN	循环神经网络（Recurrent Neural Network）
GRU	门控循环单元（Gated Recurrent Unit）
MAAC	多注意力执行者-评价者算法（Multi-Attention Actor-Critic）
IGM	个体-全局-最大（Individual-Global-Max）
MDP	马尔可夫决策过程（Markov Decision Process）
POMDP	部分可观测马尔可夫决策过程（Partially Observable Markov Decision Process）
Dec-POMDP	分散式部分可观测马尔可夫决策过程（Decentralized Partially Observable Markov Decision Process）
DADC	分散式执行者-分布式评价者（Decentralized Actors-Distributed Critics）
IAC	独立执行者-评价者（Independent Actor-Critic）
DACC	分散式执行者-集中式评价者（Decentralized Actors-Centralized Critic）
SOC	充电状态（State of Charge）
BES	储能电池（Battery Energy Storage）
CG	可控发电机（Controllable Generator）
PV	光伏（Photovoltaic Panel）
EV	电动汽车（Electric Vehicle）
HEMS	家庭能量管理系统（Home Energy Management System）

第 1 章　引言

1.1　研究背景和意义

传统电力系统以火力发电为主要电力来源，其稳定持续的能量输出有力支撑了大电网的运行。然而，随着煤炭等传统化石能源的枯竭及其引起的全球环境的恶化，传统电力系统亟待向清洁、可持续的新型电力系统转型。在这种背景下，太阳能、风能等可再生能源受到前所未有的重视。根据国家能源局数据[1]，2022 年一季度，我国可再生能源新增装机 2541 万千瓦，占全国新增发电装机的 80%。

微电网是可再生能源的重要载体。与传统能源的集中式发电方式不同，可再生能源发电分布广泛。微电网能够就近利用分布式的可再生发电资源，实现区域内能源的就地消纳，减少远距离电网架设成本和电力传输损耗。据统计，全球已有至少 7643 个微电网项目，配备超过 24000 吉瓦的总容量[2]。

由于可再生能源发电的随机性、不确定性和不稳定性，在可再生能源主导的微电网中，如何维持功率平衡、保证微电网可靠运行是学术界和工业界研究的热点问题。随着消费者用电观念的转变和智能化用电设备的普及，需求侧逐渐具备主动参与能量管理的能力。需求侧存在大量柔性负荷，可以在一定时间范围和幅度内进行灵活调整，为提高可再生能源消纳率、降低微网运行成本提供了广阔空间。

在需求侧的各种负荷中，住宅负荷有着特殊地位。在一些发达国家，住宅负荷占据近 40% 的能源消耗和二氧化碳排放量，超过商业、工业和交通等其他用电终端[3]。此外，住宅负荷包括大量可调节的电力设备，如空调、热水器、洗衣机、电动汽车等。因此，住宅负荷具有突出的管理和调控潜力。然而，单个住宅在需求侧管理中的调节幅度是较小的，要在微电网中达到削峰填谷的效果，需要建立有效的协调机制，使住宅负荷的调控形成聚集效应。因此，住宅负荷协同控制对住宅微电网意义重大，也是学术界关注的焦点。

住宅微电网的负荷控制通常依赖于对住户信息的获取，由此给住户带来了隐私风险。例如，为了提高用户在空调房间的热舒适度，微电网运营商需要收集室内温度来控制空调[4-8]；为了规划调度电动汽车充电过程，住户驾驶电动汽车到达住宅和离开住宅的时间需要提前披露给微电网运营商[9-12]，由此用户的出行隐私便存在泄漏隐患。事实上，大多数现有的住宅微电网控制方法在设计时都假定包括用户行为偏好的先验知识是可供微电网运营商获取并使用的[13-15]，这对住户的隐私构成了严重的威胁。其中一种情况是，用户对温度的偏好可以基于用户的热舒适损失函数进行推断。因此，解决住宅协同负荷控制的隐私问题对促进住宅微电

网供电侧和用电侧良性互动至关重要。

强化学习是解决住宅协同负荷控制隐私问题的潜在方案。借助于深度神经网络的信息提取和表征能力，强化学习有潜力在信息受限的住宅微电网场景中学习负荷协同控制策略。因此，本文试图以强化学习方法为抓手，搭建适用于住宅负荷协同控制场景的隐私保护的解决方案。

1.2　强化学习研究现状

强化学习是一类解决时序决策问题的机器学习方法，其在上世纪末有一段研究高潮[16]，但因为不能很好处理高维空间而没有在实际问题中产生巨大影响。自从深度学习热潮爆发以来，强化学习也乘势而起，通过与神经网络的结合取得了一系列成就：从玩转 Atari 游戏的深度 Q 网络 (DQN)[17]，到完胜世界围棋冠军李世石的 AlphaGo[18]，再到使用多智能体强化学习在星际争霸中击败职业电竞选手的 AlphaStar[19]。在仿真环境中大放异彩的强化学习，在机器人控制等真实世界的难题上也被寄予厚望[20]。

近年来有影响力的强化学习算法几乎都与神经网络结合。因此，如果不额外说明，下文出现的强化学习均指深度强化学习。根据智能体的数量，强化学习可以分为单智能体强化学习和多智能体强化学习两类。表1.1展示了这两类强化学习的典型算法。

表 1.1　强化学习算法分类

算法分类		典型算法
单智能体强化学习	基于值的	DQN[17]、Double DQN[21]、Dueling DQN[22]
	基于策略的	A2C[23]、PPO[24]、SAC[25]
多智能体强化学习	基于值的	VDN[26]、QMIX[27]、QTRAN[28]
	基于策略的	MADDPG[29]、COMA[30]、MAAC[31]

1.2.1　单智能体强化学习

在强化学习发展初期，强化学习研究的场景里只有一个智能体与外部环境进行交互。掀起强化学习研究热潮的标志性工作是 13 年 Volodymyr Mnih 等人提出的 DQN 算法[17]。彼时，深度学习刚刚在计算机视觉和语音识别等领域崭露头角。作者敏锐地抓住了深度神经网络强大的特征提取能力，将其用于近似强化学习的动作值函数，一举解决了传统强化学习无法处理高维状态空间的难题。在这篇文章中，直接以 Atari 游戏图像为输入的 DQN 算法取得了比人类玩家更出色的结果。DQN 算法的出现是强化学习发展的里程碑事件，引发了强化学习研究和应用的高

潮。在 DQN 基础上，后续的研究者着眼于样本利用效率和训练稳定性等因素，又提出了 Double DQN 算法和 Dueling DQN 算法等改进版本。这一系列工作统称为基于值的强化学习算法，因为其核心是用深度神经网络近似动作值函数。在选取动作时，这一类算法遍历动作空间中所有动作的值函数，然后选择令值函数最大的动作。因此，基于值的强化学习算法难以解决高维连续动作空间问题。

为了在连续动作空间中寻找最优策略，人们开始寻求直接对强化学习策略本身进行建模和优化，从而出现了基于策略的强化学习算法，又称策略梯度算法。早期的策略梯度算法以 REINFORCE[32] 为代表。REINFORCE 算法又称蒙特卡洛策略梯度算法。顾名思义，REINFORCE 算法依赖于蒙特卡洛方法估计策略累积奖励的期望，因此智能体从某个状态出发，需要与环境交互到底，得到整个轨迹才能计算该状态的累积奖励。作为一种朴素的策略梯度算法，REINFORCE 算法对累积奖励期望的估计方差较大，对样本的利用效率很低。后来研究者发现在策略梯度算法中引入对值函数的估计可以有效降低累积奖励期望的估计方差。因为研究者习惯将策略网络称为 Actor（执行者），将值函数网络称为 Critic（评价者），这一类算法被称为 Actor-Critic（执行者-评价者，AC）算法。

AC 框架迅速成为后续强化学习研究的标配，涌现出一批至今仍活跃在研究和应用前沿的优秀算法。例如，DDPG（深度确定性策略梯度）算法用深度神经网络近似动作值函数，代替确定性策略梯度算法中的真实动作值函数，成功将传统确定性策略梯度算法的应用场景扩展到高维空间问题。为提高对动作空间的探索能力，DDPG 算法在连续空间中选择动作时加入了噪声。尽管如此，基于确定性策略的算法与随机策略算法相比，更容易陷入局部最优策略。

A2C（优势执行者-评价者）算法采用了随机策略表示，并且将策略梯度定理[33] 中的累积奖励期望更换为优势函数，在保持对策略梯度的无偏估计的同时，显著降低了策略梯度估计的方差。作为一种在线策略算法，A2C 算法采集一批样本用作策略网络和值函数网络更新后，便将样本丢弃。因此，A2C 算法的一个较为突出的不足之处是对样本的利用效率低。

为了提高样本利用效率，PPO（近端策略优化）算法利用同一批样本进行多次策略更新。为了避免更新多次后的策略与产生样本的策略差距过大，PPO 算法对目标函数进行了约束：当某个动作产生的累积奖励高于策略的平均累积期望时，更新后的策略将会增加产生该动作的概率，但不得超过一个上限；当某个动作产生的累积奖励低于策略的平均累积期望时，更新后的策将会略减小产生该动作的概率，但不得低于一个下限。对目标函数的约束有效提高了 PPO 算法的稳定性，PPO 算法已经成为 OpenAI 强化学习研究的默认算法。

尽管对同一批样本使用多次，PPO 算法仍是一种在线策略算法，其样本利用效率无法与离线策略算法相提并论。作为一种出色的离线策略算法，SAC（软执行者-评价者）算法在优化的目标函数中加入了描述策略随机性程度的熵，迫使智能体探索更多的最优策略，从而提高了算法的探索能力。值得说明的是，虽然 AC 算法同时用神经网络表示策略和值函数，但对值函数的估计是为了辅助策略网络的更新优化，因此 AC 算法本质上仍属于基于策略的强化学习。

1.2.2　多智能体强化学习

随着强化学习的发展，研究人员试图将强化学习应用于更广泛和普遍的场景中。然而，单智能体强化学习算法在某些场景中捉襟见肘，甚至完全无法部署。例如，在无人机集群中，以单智能体强化学习代表的集中式控制方式依赖低时延的通信和强大的中心算力，这大大降低了无人机集群的反应速度。因此，多智能体强化学习逐渐成为强化学习研究的热点。

类似于单智能体强化学习算法的分类，多智能体强化学习算法也分为基于值的多智能体强化学习和基于策略的多智能体强化学习。基于值的多智能体强化学习算法的研究基础是 IGM（本地-全局-最大）假设：对某个问题的全局动作值函数，存在一组本地动作值函数，满足每个智能体使本地动作值函数最大的动作组成的联合动作，同样使全局动作值函数最大。为了满足该假设，不同的基于值的多智能体强化学习算法设计了不同的全局动作值函数网络。VDN 算法[26]采用了一种最简单的设计：全局动作值函数等于所有本地动作值函数之和。显然，VDN 算法中全局动作值函数和本地动作值函数的线性关系导致该算法仅仅能表征非常少的一部分问题。QMIX 算法[27]将 VDN 算法的线性关系拓展为单调关系，即全局动作值函数对每个本地动作值函数是单调递增的。为了实现这一目的，QMIX 算法设计了一种特殊的前馈网络将本地动作值函数映射为全局动作值函数：该前馈网络中参与乘法计算的参数非负。显然，QMIX 算法表征全局动作值函数的能力强于 VDN 算法，但仍不能表示满足 IGM 假设的所有全局动作值函数。在 QMIX 算法基础上，后续又发展出其他类似算法[28]。然而，基于值的强化学习算法存在的问题仍然是不能很好处理连续动作空间问题。

基于策略的多智能体强化学习是基于策略的单智能体强化学习在多智能体环境上的拓展。例如，基于 DDPG 算法，MADDPG 算法[29]为每个智能体设计了以全局状态为输入的动作值函数，每个智能体的策略根据对应的动作值函数进行更新。COMA 算法[30]同样使用以全局状态为输入的 Critic 估计全局动作值函数，不同的是，在利用同一个全局 Critic 计算各个智能体的策略梯度时，将全局动作值函数减去每个智能体的反事实基线（Counterfactual Baseline）从而计算出每个智能体

对全局奖励的贡献。COMA 算法的这种信用分配机制有利于协同各个智能体的动作。MAAC 算法[31]将注意力机制融入到中心化 Critic 中，加强了每个智能体的动作值函数考虑其他智能体信息的能力，从而实现智能体间的协作。由上述几种典型算法可以看出，当前基于策略的多智能体强化学习以分散式执行者-集中式评价者（Decentralized Actors-Centralized Critic）为基本框架。

1.3　住宅电力负荷协同控制研究现状

住宅电力负荷的协同控制方案需要综合电力供应侧和住宅住户需求侧的信息，给出住宅电力负荷的控制信号，在满足所有住宅用户的电力需求的同时，降低所控制范围的整体用电成本。根据控制方法的不同，目前住宅电力负荷的协同控制方法可以分为模型驱动和数据驱动两类。模型驱动的方法首先对研究对象进行系统辨识，再以获得的模型为基础进行优化控制。由于可再生能源发电具有很强的不确定性和随机性，对可再生能源主导的微电网精确建模并非易事。此外，随着微电网内住宅数量和可控电力设备的增加，求解高维模型问题非常棘手。因此，本节专注于对数据驱动的住宅电力负荷协同控制方法综述。此外，为了响应本文对隐私保护的考虑，本节还简述了负荷控制中的隐私保护方法。

1.3.1　数据驱动的住宅负荷控制方法

传统的数据驱动方法首推 PID 控制。虽然 PID 控制至今仍在实践中广泛应用，但 PID 控制方法难以处理强非线性、时变性和具有周期性扰动的系统。近年来，以强化学习为代表的数据驱动方法在负荷控制领域大放异彩。一方面，强化学习能够在没有先验知识和显式模型的情况下处理负荷控制问题；另一方面，得益于深度神经网络强大的特征提取能力，即使在隐私限制的环境中，强化学习算法仍然可以凭借有限信息学习到可靠的策略[34]。基于值的深度强化学习算法率先被成功应用于负荷控制领域。例如，DQN 被应用于电动汽车的充电调度[35]，以及优化可中断负荷的需求响应[36]。随着应用场景的拓展，基于策略的强化学习算法在具有连续动作空间的任务中展现出更大的优势。DDPG 算法被用于包含空调在内的家庭能量管理[7]，具有优先级经验重放的改进 DDPG 算法被用于住宅多能系统的优化控制[37]。

随着被控负荷数量的增加，多智能体强化学习算法的优越性逐渐凸显。多智能体强化学习将控制权限由整个微电网运营商下放至住宅甚至负荷，显著降低了每个智能体的动作维度。例如，用 PPO 算法优化采用了 IAC（独立执行者-评论者）框架的多住宅能源管理方案[38]。虽然 IAC 框架下的多智能体强化学习缩小了每个

智能体的观测空间和动作空间，但智能体之间缺乏协调机制，很难共同优化同一个目标。因此，基于多智能体强化学习的协同负荷控制方法大多采用 DACC 框架。例如，同样利用 PPO 算法，以全局状态为输入的 Critic 能够帮助训练智能体间的协同策略，从而降低所有住宅的总用电成本[39]。类似地，MADDPG 算法中分配给每个智能体的用于近似动作值函数的 Critic 也以包含所有智能体观测的全局状态为输入，因此每个智能体能够将其他智能体的状态和策略考察在内，从而做出最有利于自己的决策。该算法可以被用于竞争环境下的住宅负荷控制[40]。

1.3.2　隐私保护的住宅负荷控制方法

尽管对住宅微电网协同负荷控制的研究不在少数[4-15]，但关于住宅微电网住户数据隐私问题的研究却寥寥无几[41]。原因可总结为如下两点：其一，现有隐私保护方法多关注基于激励[42]或基于价格[43]的需求响应方法，用以解决先进测量设施（Advanced Metering Infrastructure）带来的隐私问题，而忽略了直接负荷控制所导致的隐私威胁；其次，传统的负荷控制方法在不获取隐私信息的情况下无法达到理想的控制效果。

通过设计特殊的控制机制，有可能实现对部分隐私数据的保护[41]。例如，在控制室内温度时，微电网运行商可以选择不直接控制空调工作功率，而是向空调所在住宅的家庭能量管理系统发送一个设定温度。家庭能量管理系统根据该设定温度以及当前的室内温度决定空调的工作状态：当室内温度高于该设定温度时，空调以最高工作功率运行；当室内温度低于该设定温度时，空调关闭运行。通过调整设定温度，微电网运行商能够在不获取室内温度信息的条件下控制空调。显然，这种空调控制机制不够精细。首先，该机制控制下的空调只能以完全开启和彻底关闭两种状态运行，无法对空调功率进行连续调整。其次，空调响应控制信号的过程有延迟，只有当室内温度达到设定温度时，空调才会改变工作状态。

多智能体强化学习是住宅负荷隐私保护协同控制的潜在可行方案。以多智能体强化学习为框架，每个住宅作为一个智能体，在执行时根据自身观测独立地控制电力负荷，在训练时联合其他住宅寻找协同控制策略。虽然大部分多智能体强化学习算法采用的“独立执行、集中训练”的范式在训练时不能保护住宅用户的隐私数据，但某些算法通过一些特殊设计，具备了保护数据隐私的功能。例如，在 MAAC（多注意力执行者-评论者）算法[31]中，各个智能体的本地观测先经过一层本地嵌入网络加工后，将编码后的信息上传给中心的注意力机制网络。因此，原始信息在嵌入网络的计算可以被视为一种加密方式。利用该算法，用户的隐私数据得以保护[44]。然而，MAAC 算法很难称得上一种理想的隐私保护协同控制方法。MAAC 的智能体通过嵌入网络将本地观测映射为一个几十维甚至上百维的向量，该向量

依旧包含着大量可被推断的信息。其次，智能体将高维向量上传至中心注意力网络的过程引起通信压力，特别是在大量住宅参与负荷协同控制的情况下。

1.4　本文主要工作

1.4.1　研究思路

针对强化学习在住宅负荷协同控制应用中的隐私保护问题，本文分别从集中式控制和分布式控制两个角度进行研究。

从集中式控制角度出发，本文设计了微电网运营商集中管理的住宅负荷隐私保护协同控制方案。由于标准强化学习算法直接应用于该方案时难以解决高维动作空间和部分可观测问题，本文提出了向量强化学习方法，并将循环神经网络融入强化学习的策略网络和值函数网络。

从分布式控制角度出发,本文设计了云边环境下的分布式住宅负荷隐私保护协同控制方案。为了在强化学习训练过程中加强对住户隐私数据的保护，且降低云边通信成本，本文提出了分散式执行者-分布式评价者（Decentralized Actor-Distributed Critic）的多智能体强化学习框架。

1.4.2　主要工作

第一章首先介绍了微电网负荷协同控制的重要性以及住宅用户对数据隐私的担忧，点明了本文的研究背景和意义；随后从模型驱动的方法、数据驱动的方法和隐私保护的方法三方面综述了住宅电力负荷协同控制的研究进展，并梳理了单智能体强化学习和多智能体强化学习的演进过程；最后展示了本文的研究思路和篇章安排。

第二章首先介绍了强化学习的预备知识，包括马尔可夫决策过程及其变体、策略评估方法以及几种典型的强化学习算法；随后对孤岛微电网协同负荷控制问题涉及的电力设备的功率进行了建模；最后通过仿真实验探讨了能量管理和负荷控制问题如何选择强化学习的输入。

第三章设计了微电网运营商集中管理的住宅负荷隐私保护协同控制方案；在一种标准强化学习算法基础上，引入基于差分奖励的信用分配机制发展了向量强化学习算法，并且在强化学习的策略网络和值函数网络中加入了循环神经网络；仿真实验对比了标准强化学习算法训练的方案、无隐私保护的方案、其他集中式隐私保护方案以及最大化住户用电体验的方案。

第四章设计了云边环境下的分布式住宅负荷隐私保护协同控制方案；提出了分散式执行者-分布式评价者（DADC, Decentralized Actors-Distributed Critics）的多

智能体强化学习框架；将 DADC 框架在负荷协同控制问题上与独立执行者-评价者框架和传统分散式执行者-集中式评价者框架作了比较。

第五章总结了本文的创新和贡献，分析本文的不足之处，并展望了未来研究方向。

第 2 章　系统建模和强化学习输入研究

2.1　本章引言

在应用强化学习算法解决微电网负荷控制问题时，如何选择强化学习神经网络的输入是一个有争议的话题。尽管基于强化学习的负荷控制方法不依赖于对随机变量（如光伏功率和基础负荷功率）的预测，且大部分工作直接将当前时刻智能体的观测作为神经网络的输入，但一些工作仍然将预测方法集成到数据驱动的强化学习中。例如，前馈神经网络被用于预测未来时刻的电价，预测结果作为强化学习网络输入的一部分[45]。类似地，多层感知器被用于建立价格预测模型，从而帮助 DDPG 算法的决策[46]。因此，有必要研究在强化学习输入中加入显式预测结果对控制效果的影响。

本章首先介绍了强化学习的基本知识，包括 MDP 及其变体，强化学习的策略评估方法以及几种典型强化学习算法；接着给出了适用于 MDP 的微电网内各种电力设备的模型，重点是以空调和电动汽车为代表的可控负荷模型；最后通过比较有显式预测和无显式预测的基于强化学习的能量管理方案，研究了如何选择强化学习输入的问题，主要是为了回应关于强化学习输入是否应该加入对未知变量预测的争论。

2.2　强化学习预备知识

2.2.1　马尔可夫决策过程

强化学习解决的标准问题是马尔可夫决策过程。其在某个时间的状态仅仅依赖上一时刻的信息，而独立于之前时刻的信息。把 $\mathcal{P}(\Omega)$ 记为在空间 Ω 中所有概率分布的集合，即：$\mathcal{P}(\Omega) = \{f | \int_{\Omega} f(x)dx = 1\}$。有限步长的马尔可夫决策过程定义为一个五元组 $\langle S, \mathcal{A}, P_s, R, T\rangle$，其中，$S$ 表示状态空间；$\mathcal{A}$ 表示动作空间；$P_s : S \times \mathcal{A} \mapsto \mathcal{P}(S)$ 表示状态转移函数；$R : S \times \mathcal{A} \times S \mapsto \mathbb{R}$ 表示奖励函数；$T \in \mathbb{N}^+$ 表示总步长。在某个时刻 t，智能体观测到状态 $s \in S$，并选择动作 $a \in \mathcal{A}$，智能体因此收到下一个时刻的状态 $s' \sim P_s(s, a)$ 和奖励 $R(s, a, s')$。

然而，很多问题中智能体无法观测到环境的全部状态，而是仅仅观测到状态的一部分或跟状态相关的变量。这种情况下，可以选择部分可观测的马尔可夫决策过程（POMDP）来描述该问题：$\langle S, \mathcal{A}, \mathcal{O}, P_s, P_o, R, T\rangle$，其中，

- S 表示状态空间；

- $\mathcal{A}$ 表示动作空间；
- $\mathcal{O}$ 表示观测空间；
- $P_s : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{P}(\mathcal{S})$ 表示状态转移函数；
- $P_o : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{P}(\mathcal{O})$ 表示观测函数；
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ 表示奖励函数；
- $T \in \mathbb{N}^+$ 表示总步长。

在某个时刻 t，智能体观测到 $o \in \mathcal{O}$，并选择动作 $a \in \mathcal{A}$，环境状态因此从 s 转移到 $s' \sim P_s(s, a)$，同时智能体收到奖励 $R(s, a, s')$ 和下一个时刻的观测 $o' \sim (s', a)$。

在多智能体环境中，上述两种马尔可夫决策过程均无法准确对问题建模。有限步长的分散式部分可观测马尔可夫决策过程（Dec-POMDP）可以用一个元组描述：$\langle \mathcal{D}, \mathcal{S}, \mathcal{A}, \mathcal{O}, P_s, P_o, R, T \rangle$，其中，

- $\mathcal{D}$ 表示所有智能体的集合；
- $\mathcal{S}$ 表示全局状态空间；
- $\mathcal{A} \equiv \times_{i \in \mathcal{D}} \mathcal{A}_i$ 表示联合动作空间；
- $\mathcal{O} \equiv \times_{i \in \mathcal{D}} \mathcal{O}_i$ 表示联合观测空间；
- $P_s : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{P}(\mathcal{S})$ 表示全局状态转移函数；
- $P_o : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{P}(\mathcal{O})$ 表示联合观测函数；
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ 表示全局奖励函数；
- $T \in \mathbb{N}^+$ 表示总步长。

在某个时刻 t，每个智能体 $i \in \mathcal{D}$ 观测到 $o_t^i \in \mathcal{O}_i$，并选择动作 $a_t^i \in \mathcal{A}_i$。组成的联合动作 $a_t = [a_t^i]_{i \in \mathcal{D}}$ 导致全局状态从 s_t 转移到 $s_{t+1} \sim P_s(s_t, a_t)$，同时所有智能体收到全局奖励 $r_t = R(s_t, a_t, s_{t+1})$。此外，环境产生联合观测 $o_{t+1} \sim P_o(s_{t+1}, a_t)$，每个智能体从中得到下一时刻的局部观测 o_{t+1}^i。

由于所研究的隐私保护协同控制不能观测到全部状态信息，下文均在（分散式）部分可观测的马尔可夫决策过程下展开。

2.2.2　策略评估

定义轨迹 $\tau = (o_1, a_1, r_1, o_2, ..., a_{T-1}, r_{T-1}, o_T)$。轨迹 τ 的累积奖励为

$$R(\tau) := \sum_{t=1}^{T-1} r_t. \tag{2.1}$$

强化学习试图通过学习策略来最大化累积奖励 $R(\tau)$。定义策略 $\pi : \mathcal{O} \mapsto \mathcal{P}(\mathcal{A})$，即策略 $\pi(\cdot|o_t)$ 表示在观测到 o_t 时在动作空间上的概率密度函数。注意此处为概率密度函数而非概率分布函数，因为本文涉及的问题具有连续动作空间。如果策略是

确定性的，策略的表示退化为 $\pi : \mathcal{O} \mapsto \mathcal{A}$。本文主要采用随机性策略表示。为了评价策略的好坏，定义策略 π 的值函数为

$$V^{\pi}(o_t) = \mathbb{E}_{s_{t+1:T}\sim P_s, o_{t+1:T}\sim P_o, a_{t:T}\sim\pi}\left[\sum_{t'=t}^{T} r_{t'}|o_t\right], \tag{2.2}$$

表示从时刻 t 观测到 o_t 并一直根据策略 π 选择动作情况下的累积奖励的期望。值函数 V^{π} 只评价了策略 π 的好坏，没有单独对某个动作进行评价。为了弥补这一点，引入策略 π 的动作值函数

$$Q^{\pi}(o_t, a_t) = \mathbb{E}_{s_{t+1:T}\sim P_s, o_{t+1:T}\sim P_o, a_{t+1:T}\sim\pi}\left[\sum_{t'=t}^{T} r_{t'}|o_t, a_t\right], \tag{2.3}$$

表示时刻 t 观测到 o_t 并采取动作 a_t，且之后根据策略 π 选择动作情况下的累积奖励的期望。注意策略的值函数和动作值函数存在关系

$$V^{\pi}(o_t) = \mathbb{E}_{a\sim\pi(o_t)} Q^{\pi}(o_t, a). \tag{2.4}$$

策略 π 的优势函数定义为

$$A^{\pi}(o_t, a_t) = Q^{\pi}(o_t, a_t) - V^{\pi}(o_t), \tag{2.5}$$

表示策略 π 控制下采取某个动作 a_t 的累积奖励期望与平均累积奖励期望之差。

2.2.3　典型强化学习算法

强化学习算法主要分为两种：基于值的方法和基于策略的方法。前者在选择动作时，需要遍历给定观测下的所有动作值函数，然后选择令动作值函数最大的动作，因此该类方法往往只适用于离散动作空间；后者直接学习从观测空间到动作空间（上的所有概率分布的集合）的映射，更适合具有连续动作空间的问题。由于本文研究的负荷协同控制问题的控制信号是连续的，因此本文主要使用基于策略的方法。

基于策略的方法又称为策略梯度方法。策略梯度算法着眼于直接对策略 π 本身进行建模和优化。策略通常用一个带参数 θ 的函数 $\pi(\cdot|\cdot;\theta)$ 表示。关于参数 θ 的目标函数可以表示为

$$J(\theta) = \mathbb{E}_{s_{1:T}\sim P_s, o_{1:T}\sim P_o, a_{1:T}\sim\pi(\cdot;\cdot;\theta)}[\sum_{t=1}^{T} r_t]. \tag{2.6}$$

根据策略梯度定理[33]，（2.6）对参数 θ 的梯度为

$$\nabla_{\theta} J(\theta) = \nabla_{\theta}\mathbb{E}_{\tau}[R(\tau)\sum_{t=1}^{T-1}\nabla_{\theta}\pi(a_t|o_t;\theta)]. \tag{2.7}$$

而后，可以使用各种各样的算法来对参数 θ 进行优化，以达到累积奖励期望最大化的目标。

在策略梯度方法中，通常将策略称为执行者（Actor），将值函数和动作值函数称为评价者（Critic）。AC（Actor-Critic）算法是一类既使用参数化的 Actor、又使用参数化的 Critic 的算法，其在基于策略的强化学习中影响最大、应用最广，已经成为主流策略梯度算法的标配。下面介绍三种常用的 AC 算法。

A2C 算法

A2C 算法以如下的梯度更新策略参数 θ：

$$\nabla_\theta J[\pi(\cdot|\cdot;\theta)] = \mathbb{E}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi(a_t|o_t;\theta)\left(r_t + V(o_{t+1};\phi) - V(o;\phi)\right)\right], \tag{2.8}$$

其中 $V(\cdot,\phi)$ 是以 ϕ 为参数的 Critic，$\left(r_t + V(o_{t+1};\phi) - V(o;\phi)\right)$ 是对优势函数的估计。为了进一步增强采样效率，A2C 算法使用了多步估计

$$A(o_t,a_t;\phi) = \sum_{t'=t}^{t+k-1}\left(r_{t'} + V(o_{t+k};\phi) - V(o_t;\phi)\right), \tag{2.9}$$

其中 k 不超过每次更新的最大步数 $t_{\max}$。因此 Actor 和 Critic 的梯度分别为

$$\Delta\theta = \sum_{t'=t}^{t+t_{\max}} \nabla_\theta\left[\log \pi(a_{t'}|o_{t'};\theta)A(o_{t'},a_{t'};\phi)\right], \tag{2.10}$$

$$\Delta\phi = \sum_{t'=t}^{t+t_{\max}} \nabla_\phi A(o_{t'},a_{t'};\phi)A(o_{t'},a_{t'};\phi). \tag{2.11}$$

PPO 算法

与 A2C 算法相同，该算法也具有一个 Actor $\pi(\cdot|\cdot;\theta)$ 和一个用于估计值函数的 Critic $V(\cdot;\phi)$。Actor 的损失为：

$$\mathcal{L}_a = \hat{\mathbb{E}}_t\left[\min\left(w_t(\theta)\hat{A}_t, \mathrm{clip}(w_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t\right)\right]. \tag{2.12}$$

其中，期望 $\hat{\mathbb{E}}_t[...]$ 表示通过有限采样的经验估计值。概率比 w_t 定义为

$$w_t = \frac{\pi(a_t|o_t;\theta)}{\pi_{\mathrm{old}}(a_t|o_t;\theta_{\mathrm{old}})}, \tag{2.13}$$

其中 π_{old} 表示用于环境交互的策略。超参数 ϵ 用于限制每次参数更新时策略的变化程度。优势函数的估计值根据 $\hat{A}_t$ 通用优势估计（GAE）算法[47]计算：

$$\hat{A}_t = \sum_{t'=t}^{T} \lambda^{t'-t}(-V_{\mathrm{old}}\left(o_{t'};\phi_{\mathrm{old}}\right) + r_{t'} + V_{\mathrm{old}}(o_{t'+1};\phi_{\mathrm{old}})\big), \tag{2.14}$$

其中参数 λ 用于权衡估计的方差和偏差：λ 越大，优势函数估计的方差越大，但估

计的偏差降低。V_{old} 表示采样后未经更新的值函数。Critic $V(\cdot;\phi)$ 的损失为:

$$\mathcal{L}_c = \hat{\mathbb{E}}_t \left[\left(V(o_t;\phi) - V_{\text{old}}(o_t;\phi_{\text{old}}) - \hat{A}_t \right)^2 \right]. \tag{2.15}$$

因此 Actor 和 Critic 的梯度分别为

$$\Delta\theta = \sum_t \nabla_{w_t} \mathcal{L}_a \cdot \nabla_\theta w_t, \tag{2.16}$$

$$\Delta\phi = \nabla_\phi \mathcal{L}_c. \tag{2.17}$$

SAC 算法

与前两种 AC 算法不同，SAC 算法是一种离线策略强化学习算法，其参数更新过程利用的样本数据并非实时采样得到，而是从经验记忆池 $\mathcal{B}$ 中随机选择一批。参数更新完毕后，具有新参数的策略与环境进行交互，交互得到的数据储存于经验记忆池。SAC 算法由值函数 $V(\cdot;\phi)$、soft Q 函数 $Q(\cdot,\cdot;\psi)$ 和策略 $\pi(\cdot|\cdot;\theta)$ 组成，其损失函数分别为:

$$\mathcal{L}_Q = \mathbb{E}_{(o_k,a_k,r_k,o_{k+1})\sim\mathcal{B}} \left[\left(Q\left(o_k,a_k;\psi\right) - r_k - V(o_{k+1};\phi) \right)^2 \right], \tag{2.18}$$

$$\mathcal{L}_V = \mathbb{E}_{o_k\sim\mathcal{B}} \left[\left(V(o_k;\phi) - \mathbb{E}_{a\sim\pi} \left[Q(o_k,a;;\psi) - \log\pi(a|o_k;\theta) \right] \right)^2 \right], \tag{2.19}$$

$$\mathcal{L}_\pi = \mathbb{E}_{o_k\sim\mathcal{B},a\sim\pi} \left[\log\pi(a|o_k;\theta) - Q\left(o_k,a;\psi\right) \right]. \tag{2.20}$$

相应的梯度为

$$\Delta\psi = \mathbb{E}_{(o_k,a_k,r_k,o_{k+1})\sim\mathcal{B}} \left[\nabla_\psi Q\left(o_k,a_k;\psi\right) \left(Q\left(o_k,a_k;\psi\right) - r_k - V(o_{k+1};\phi) \right) \right], \tag{2.21}$$

$$\Delta\phi = \mathbb{E}_{o_k\sim\mathcal{B}} \left[\nabla_\phi V\left(o_k;\phi\right) \left(V(o_k;\phi) - \mathbb{E}_{a\sim\pi} \left[Q(o_k,a;;\psi) - \log\pi(a|o_k;\theta) \right] \right) \right], \tag{2.22}$$

$$\begin{aligned} \Delta\theta = \mathbb{E}_{o_k\sim\mathcal{B},a\sim\pi} \big[& \nabla_\theta \log\pi(a|o_k;\theta) \\ & + \left(\nabla_a \log\pi(a|o_k;\theta) - \nabla_a Q(o_k,a;\psi) \right) \nabla_\theta f(\varepsilon,o_k;\theta) \big]. \end{aligned} \tag{2.23}$$

其中 $f(\varepsilon,\cdot;\theta)$ 是策略 $\pi(\cdot|\cdot;\theta)$ 的显式形式，ε 是与参数 θ 无关的随机变量。给定观测 o，随机策略 $\pi(\cdot|o;\theta)$ 采样动作 a 是通过 $a = f(\varepsilon,o;\theta)$ 完成的。下面以一个具体的例子加以解释。假设策略 $\pi(\cdot|\cdot;\theta)$ 是一个正态分布，给定观测 o，Actor 输出正态分布的均值 $\mu(o;\theta)$ 和方差 $\sigma(o;\theta)$，则

$$\pi(\cdot|o;\theta) = \mathcal{N}\left(\mu(o;\theta),\sigma(o;\theta)\right),$$

从该策略采样的动作可以表示为

$$a = f(\varepsilon,o;\theta) = \sigma(o;\theta)\left(\varepsilon + \mu(o;\theta)\right)$$

其中 $\varepsilon \sim \mathcal{N}(0,1)$。

2.3　系统建模

本文考虑的住宅电力负荷协同控制问题所处的场景是孤岛微电网。从控制的角度, 住宅电力负荷可以分为基础负荷、可调节负荷和可转移负荷等。为了维持电力供需平衡，微电网还配有光伏、储能设备、可控发电机的一种或几种。系统建模过程只考虑功率，忽略频率、电压等因素。假设微电网在离散的时间点进行操作，即 $t \in \mathcal{T} = \{1, 2, ..., T\}$，$T$ 表示总时间步，时间间隔为 Δt。

2.3.1　光伏功率模型

由于可再生能源发电的时变性、波动性和不确定性，利用显式模型准确预测光伏输出功率通常是很困难的。因此本文直接使用真实光伏功率数据来描述光伏发电的动态变化过程。图2.1展示了以 5 分钟为时间间隔的 5 天光伏功率数据。容易看出，由于受天气等因素影响十分严重，光伏输出功率波动性非常强，如第五天中午左右的光伏功率急剧下降为零。光伏输出功率的巨大波动性给孤岛微电网的功率平衡带来了挑战，同时也凸显了负荷调度的重要性。

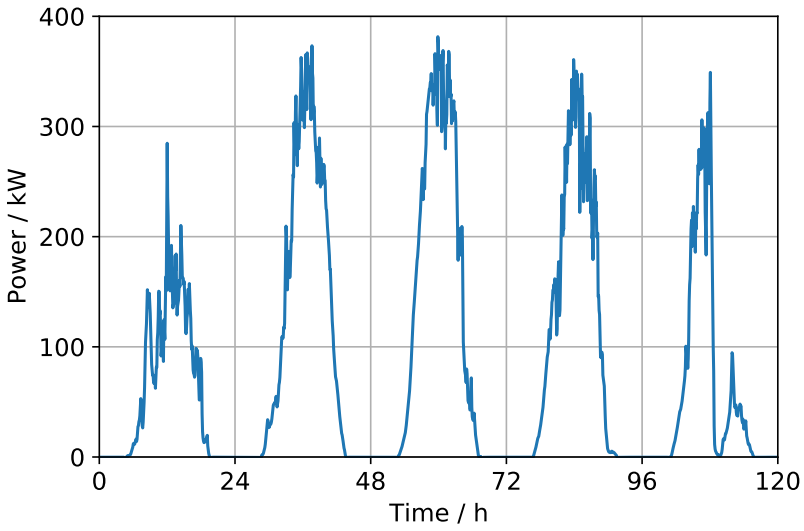


图 2.1　光伏功率数据示意图

2.3.2　基础负荷模型

住宅的基础负荷是一类不可削减、不可转移的负荷，必须在住户指定的时间内提供住户所需的电能。此类负荷通常涉及到住宅的正常运转和住户的即时需求，如照明设备。与光伏发电类似，此类负荷具有随机性和偶发性，因此本文也直接利用真实住宅用电数据进行模拟。图2.2展示了以 5 分钟为时间间隔的 5 天住宅区域基础负荷功率数据。由于聚集了多个住宅的基础负荷，功率的毛刺噪声非常明显。

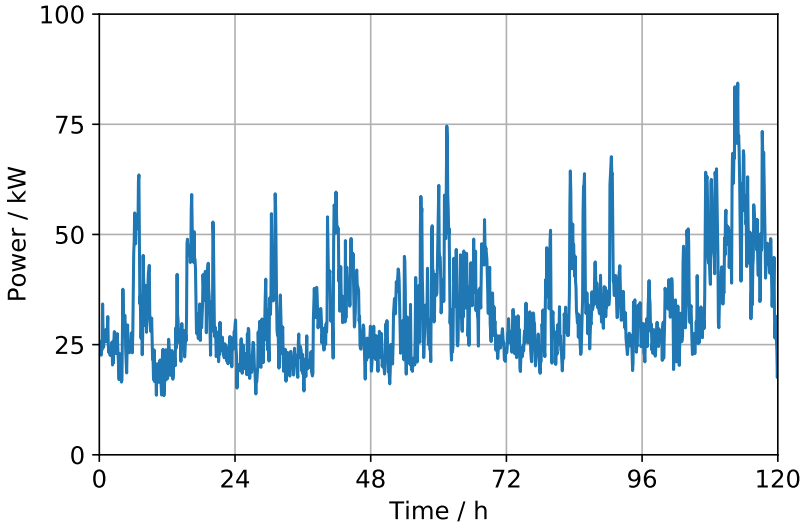


图 2.2　基础负荷功率数据示意图

2.3.3　可调节负荷模型

可调节负荷可在一定范围内和一定约束条件下改变设备的运行功率，如空调、热水器等设备。以空调为例，其目的是控制室内温度来满足住户的热舒适需求。假设住户 i 的热舒适区为 $[T_{i,\min}^{\mathrm{AC}}, T_{i,\max}^{\mathrm{AC}}]$，其中 $T_{i,\min}^{\mathrm{AC}}$ 和 $T_{i,\max}^{\mathrm{AC}}$ 分别表示用户热舒适区的下限温度和上限温度。相应房间的温度动力学过程为

$$T_{i,t+1}^{\mathrm{AC}} = f_i^{\mathrm{AC}}\left(T_{i,t}^{\mathrm{AC}}, T_{i,t}^{\mathrm{out}}, P_{i,t}^{\mathrm{AC}}, \varrho_{i,t}\right). \tag{2.24}$$

公式（2.24）中，$T_{i,t}^{\mathrm{out}}$、$T_{i,t}^{\mathrm{out}}$、$P_{i,t}^{\mathrm{AC}}$ 和 $\varrho_{i,t}$ 分别表示住户 i 所处房间时刻 t 的室内温度、室外温度、空调功率和随机扰动，$f_i^{\mathrm{AC}}(\cdot,\cdot,\cdot,\cdot)$ 是关于上述变量的室内温度动力学函数。获得函数 f_i^{AC} 的形式和参数是一项系统辨识问题，需要一系列的现场实验。由于本文使用的是数据驱动的方法，对函数 f_i^{AC} 的形式没有任何假设，因此在此处不明确给出其形式和参数。空调的功率满足约束

$$0 \leqslant P_{i,t}^{\mathrm{AC}} \leqslant P_{i,\max}^{\mathrm{AC}}, \tag{2.25}$$

其中 $P_{i,\max}^{\mathrm{AC}}$ 表示空调的最大工作功率。

2.3.4　可转移负荷模型

可转移负荷允许在一定时间范围内灵活运行，前提是在指定时间之前完成固定的电力任务。此类负荷的代表是电动汽车的充电需求。令住户 i 的电动汽车在时刻 t 的电量为 $E_{i,t}^{\mathrm{EV}}$，到达时间和离开时间分别为 a_i^{EV} 和 d_i^{EV}，到达时刻的电量为 E_i^{init}，离开时刻的目标电量为 E_i^{targ}，充电效率为 η_i^{EV}。电动汽车充电过程可以表示

为

$$E_{i,t+1}^{\mathrm{EV}}=\begin{cases}E_i^{\mathrm{init}}, & \text{if} \quad t+1=a_i^{\mathrm{EV}},\\ E_{i,t}^{\mathrm{EV}}+\eta_i^{\mathrm{EV}}P_{i,t}^{\mathrm{EV}}\Delta\mathrm{t}, & \text{if} \quad a_i^{\mathrm{EV}}\leqslant t<d_i^{\mathrm{EV}},\\ 0, & \text{otherwise.}\end{cases} \tag{2.26}$$

此外，电动汽车的电量和充电功率分别满足约束

$$E_{i,\min}^{\mathrm{EV}}\leqslant E_{i,t}^{\mathrm{EV}}\leqslant E_{i,\max}^{\mathrm{EV}}, \tag{2.27}$$

$$0\leqslant P_{i,t}^{\mathrm{EV}}\leqslant P_{i,\max}^{\mathrm{EV}}, \tag{2.28}$$

其中，$E_{i,\min}^{\mathrm{EV}}$ 和 $E_{i,\max}^{\mathrm{EV}}$ 表示电动汽车的最大容许电量和最小容许电量，$P_{i\max}^{\mathrm{EV}}$ 表示电动汽车的最大充电功率。

2.3.5　可控发电机和储能设备模型

可控发电机和储能设备对维持孤岛微电网功率平衡具有十分关键的作用。前者在可再生能源发电出现严重不足时提供稳定可靠的电力保证；后者在削峰填谷、降低用电成本方面发挥重要作用。可控发电机输出功率的动态调节过程可以表示为

$$P_{t+1}^{\mathrm{CG}}=f^{\mathrm{CG}}\left(P_t^{\mathrm{CG}},u_t^{\mathrm{CG}}\right), \tag{2.29}$$

其中 P_t^{CG} 和 u_t^{CG} 分别表示可控发电机在时刻 t 的输出功率和控制信号，$f^{\mathrm{CG}}(\cdot,\cdot)$ 是可控发电机关于上述变量的输出功率动力学函数。控制信号 u_t^{CG} 满足约束

$$0\leqslant u_t^{\mathrm{CG}}\leqslant 1. \tag{2.30}$$

储能设备的动态调节过程可以表示为

$$\mathrm{SOC}_{t+1}=f^{\mathrm{BES}}\left(\mathrm{SOC}_t,P_t^{\mathrm{BES}}\right), \tag{2.31}$$

其中 SOC_t 和 u_t^{CG} 分别表示储能设备在时刻 t 的的电量状态和充放电功率，$f^{\mathrm{BES}}(\cdot,\cdot)$ 是储能设备关于上述变量的电量状态动力学函数。充/放电功率满足约束

$$0\leqslant \mathrm{SOC}_t\leqslant 1, \tag{2.32}$$

$$-P_{\mathrm{d,max}}^{\mathrm{BES}}\leqslant P_t^{\mathrm{BES}}\leqslant P_{\mathrm{c,max}}^{\mathrm{BES}}, \tag{2.33}$$

其中 $P_{\mathrm{d,max}}^{\mathrm{BES}}$ 和 $P_{\mathrm{c,max}}^{\mathrm{BES}}$ 分别表示储能设备的最大放电功率和最大充电功率。

2.4　强化学习输入研究

结合前两节可以给出住宅负荷协同控制问题的 MDP 建模。在应用深度强化学习算法解决建立的问题时，如何选择神经网络的输入是一个值得讨论的话题。大部分工作直接将当前观测作为输入，而有些工作将预测方法集成到强化学习中。预测本质上是对原始信息的降维处理，从这个意义上说，桥接预测模块后的强化学习智能体接收到的信息是不完整的。因此，这种方式无法充分利用深度神经网络强大的特征提取能力。尽管预测在基于模型的方法中具有重要作用，但在无模型方法中加入预测可能会破坏控制效果。

为了研究对随机变量进行额外的显式预测是否能提升强化学习的控制效果，本节针对一个简单的能量管理问题对比了使用和不使用预测的强化学习方法。使用预测的强化学习方法包括监督学习和强化学习两部分，监督学习用于训练预测模型，而训练好的预测模型服务于强化学习的训练。不使用预测的强化学习采取端到端的网络进行强化学习训练。我们希望仿真实验的对比结果能够指导强化学习的状态选择。

2.4.1　一般能量管理问题

考虑了微电网中的最一般的能量管理问题。如图2.3所示，微电网由光伏、基础负荷、储能设备和能量管理系统组成。时间跨度 T 为 24 小时，时间步长 Δt 为 1 小时。在每小时开始时，能量管理系统观测到上一小时的可再生发电输出功率和电力需求，从储能设备接收当前电量状态，从公用电网接收上一小时的电价。基于这些信息，能量管理系统确定储能设备的充/放电功率。如果在这一小时里出现能量短缺，微电网会从电网处购买适量的电能；而多余的能量将被丢弃。

由于该问题中的所有信息均能被能量管理系统捕获，可以用 MDP 建模。MDP 的状态 s_t 为能量管理系统观察到的微网状态包括可再生能源发电输出和电力需求、储能设备的电量状态和小时电价：

$$\boldsymbol{s}_t = \left[\mathrm{SOC}_t, P_t^{\mathrm{PV}}, P_t^{\mathrm{BL}}, p_t\right], \tag{2.34}$$

其中 p_t 表示时刻 t 的电价。MDP 的动作 $a_t = P_t^{\mathrm{BES}}$ 为时刻 t 储能设备的充放电功率。状态的前三个分量的转移函数已在上一节介绍，而电价则是通过真实电价数据描述。整个微电网功率平衡由公用电网保证：

$$P_t^{\mathrm{util}} = \begin{cases} P_t^{\mathrm{BL}} - P_t^{\mathrm{PV}} + P_t^{\mathrm{BES}}, & \text{if} \quad P_t^{\mathrm{BL}} - P_t^{\mathrm{PV}} + P_t^{\mathrm{BES}} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.35}$$

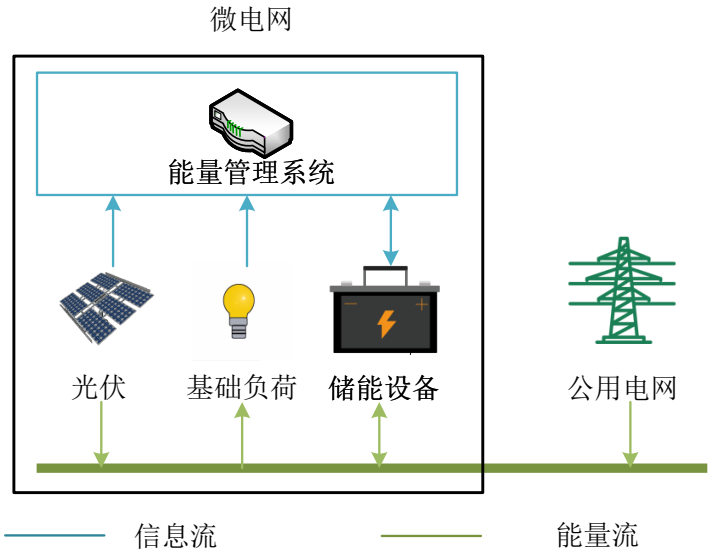


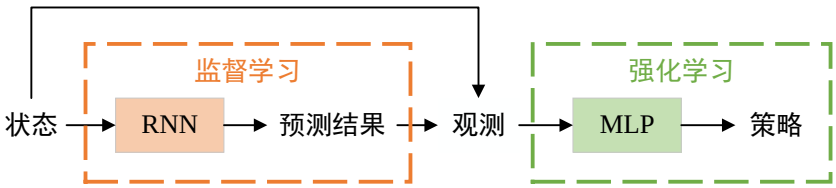
图 2.3　一般微网能量管理结构图

其中 P_t^{util} 是时刻 t 从公用电网输入微电网的功率。MDP 的奖励 r_t 定义为

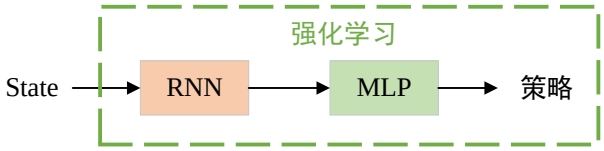
$$r_t = -p_t \cdot P_t^{\text{util}} - g^{\text{BES}}\left(\text{SOC}_t, P_t^{\text{BES}}\right), \tag{2.36}$$

其中 $g^{\text{BES}}(\cdot,\cdot)$ 是储能设备关于当前电量和充放电功率的损失成本函数。

2.4.2　有预测环节的强化学习



(a) 有预测



(b) 无预测

图 2.4　有预测和无预测的微电网能量管理方案

本章设计的有预测的微电网能量管理方案和无预测的微电网能量管理方案如

图2.4所示。在有预测的微电网能量管理方案中，监督学习训练的 RNN 以微电网的状态为输入，输出对未来时刻信息的预测，预测结果和状态串联作为强化学习中 MLP 网络的输入；在无预测的微电网能量管理方案中，由 RNN 和 MLP 级联构成的强化学习策略网络直接将微网状态映射成策略。

本节首先应用监督学习训练 RNN 来预测未来时刻的光伏功率、基础负荷功率和电价。训练好的预测模型被用于强化学习训练。

算法 2.1 k 步预测的监督学习

初始化参数 φ
for epoch = 1 to N **do**
  for $t = 1$ to T **do**
    $\hat{x}_{t+1}, \cdots, \hat{x}_{t+k}, h_{t+1} = \mathrm{GRU}(x_t, h_t; \varphi)$
    $\Delta\varphi \leftarrow \Delta\varphi + \nabla_\varphi \left(\sum_{i=1}^{k} \hat{x}_{t+i} - x_{t+i} \right)^2$
  end for
  用 $\Delta\varphi$ 更新 φ
end for

监督学习的目标是学习映射 f，使得 $y = f(x; \varphi)$，其中 x、y 和 φ 分别表示输入、标签和参数。可预测变量包括未来的光伏功率、电力需求和电价，这属于时间序列预测问题。考虑到 RNN 在时间序列预测任务中的优势，预测模型使用了 GRU，一种门控机制的 RNN。监督学习的训练过程如算法 2.1 所示。输入 x_t 可以是光伏输出功率 P_t^{PV}、基础负荷功率 P_t^{BL} 和电价 p_t。GRU 使用反向传播进行训练，从而最小化 GRU 的输出与目标值之间的均方误差 (MSE) 。

预测模型训练完成后，用作生成强化学习网络的输入。如下式所示，强化学习网络的输入是状态和预测的串联

$$o_t = s_t \cup \left[\hat{P}_{t+1}^{\mathrm{PV}}, \hat{P}_{t+1}^{\mathrm{BL}}, \hat{p}_{t+1}, \cdots, \hat{P}_{t+k}^{\mathrm{PV}}, \hat{P}_{t+k}^{\mathrm{BL}}, \hat{p}_{t+k} \right], \tag{2.37}$$

其中扩展部分由预测模型生成。

强化学习的策略由 PPO 算法[24]训练，细节如算法 2.2中所示。

2.4.3　无预测环节的强化学习

如图2.4所示, 在有预测模块的方案下，RNN 和 MLP 分别用监督学习和强化学习进行训练。而没有预测的方案执行端到端的训练。换句话说，Actor 网络和评论者网络由 RNN 和 MLP 组成，而不仅仅是 MLP。从这个意义上说，在训练过程中，RNN 可以自动学习适当的参数，以便可以捕获先前时间步骤中最重要的信息帮助进行决策。算法2.3展示了没有预测的强化学习方案。状态 s_t 直接作为输入来生成策略和值函数，同时生成 Actor 网络和评论者网络的隐状态。

算法 2.2 有预测的 PPO 训练

初始化 Actor 网络参数 θ 和 Critic 网络参数 ϕ
加载 GRU 参数 φ
for episode = 0 to N **do**
  $h_0 \leftarrow 0$
  for $t = 0$ to T **do**
    执行预测 $x_t, h_{t+1} = \mathrm{GRU}(\hat{x}_{t+1}, \cdots, \hat{x}_{t+k}, h_t; \varphi)$
    $o_t \leftarrow [s_t, \hat{x}_{t+1}, \cdots, \hat{x}_{t+k}]$
    $\mathcal{P} \leftarrow \pi(\cdot|o_t; \theta)$, $v_t = V(o_t; \phi)$
    根据分布 $\mathcal{P}$ 采样动作 a_t
    执行动作 a_t 并观测到 s_{t+1}
    计算概率 $p_t^{\mathrm{old}} \leftarrow \mathcal{P}(a_t)$
  end for
  $\hat{A}_T \leftarrow 0, v_T \leftarrow 0$
  for $t = T - 1$ to 0 **do**
    $\hat{R}_t \leftarrow \gamma\lambda\hat{A}_{t+1} + r_t + \gamma v_{t+1}$
    $\hat{A}_t \leftarrow \hat{R}_t - v_t$
  end for
  for $k = 1$ to K **do**
    $\mathcal{L}_a \leftarrow 0, \mathcal{L}_c \leftarrow 0$
    for $t = 0$ to $T - 1$ **do**
      $w_t \leftarrow \pi(a_t|o_t; \theta)/p_t^{\mathrm{old}}$
      $\mathcal{L}_a += \min\left(w_t\hat{A}_t, \mathrm{clip}\left(w_t, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)$
      $\mathcal{L}_c += \left(V(o_t; \phi) - \hat{R}_t\right)^2$
    end for
    分别用梯度 $\nabla_\theta\mathcal{L}_a$ 和 $\nabla_\phi\mathcal{L}_c$ 更新参数 θ 和 ϕ
  end for
end for

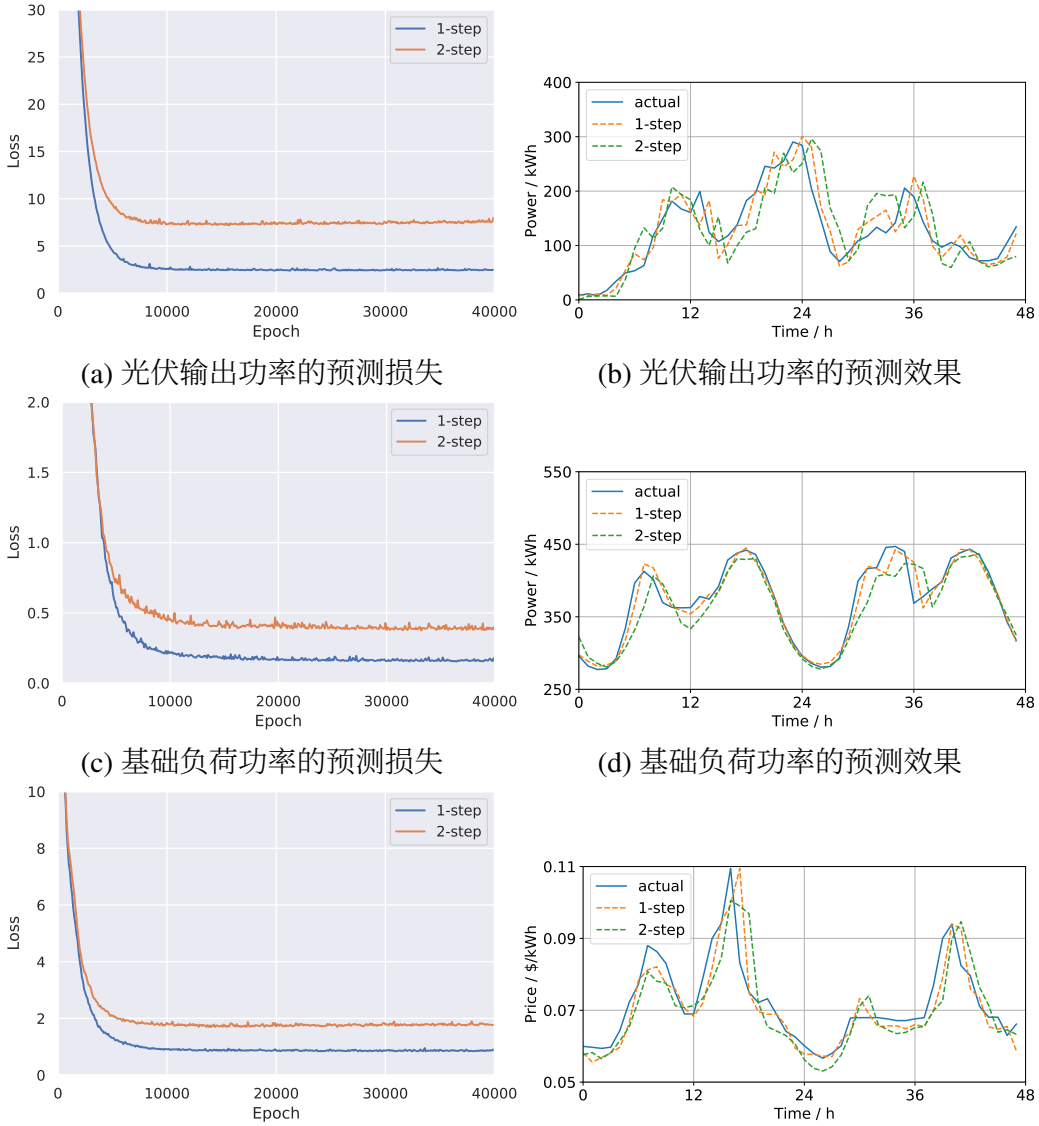
算法 2.3 无预测的 PPO 训练

初始化 Actor 网络参数 θ 和 Critic 网络参数 ϕ
for episode = 0 to N **do**
  $h_0^\pi \leftarrow 0, h_0^V \leftarrow 0$
  for $t = 0$ to T **do**
    $\mathcal{P}, h_{t+1}^\pi = \pi(\mathbf{s}_t, h_t^\pi; \theta)$
    根据分布 $\mathcal{P}$ 采样动作 a_t
    $p_t^{\text{old}} \leftarrow \mathcal{P}(a_t)$
    $v_t, h_{t+1}^V = V(\mathbf{s}_t, h_t^V; \phi)$
    执行动作 a_t 并观测到 $\mathbf{s}_{t+1}$
  end for
  $\hat{A}_T \leftarrow 0, v_T \leftarrow 0$
  for $t = T - 1$ to 0 **do**
    $\hat{R}_t \leftarrow \gamma\lambda\hat{A}_{t+1} + r_t + \gamma v_{t+1}$
    $\hat{A}_t \leftarrow \hat{R}_t - v_t$
  end for
  for $k = 1$ to K **do**
    $\mathcal{L}_a \leftarrow 0, \mathcal{L}_c \leftarrow 0, h_0^\pi \leftarrow 0, h_0^V \leftarrow 0$
    for $t = 0$ to $T - 1$ **do**
      $\mathcal{P}, h_{t+1}^\pi = \pi(\mathbf{s}_t, h_t^\pi; \theta)$
      $V_t, h_{t+1}^V = V(\mathbf{s}_t, h_t^V; \phi)$
      $w_t \leftarrow \mathcal{P}(a_t)/p_t^{\text{old}}$
      $\mathcal{L}_a += \min\left(w_t\hat{A}_t, \text{clip}\left(w_t, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)$
      $\mathcal{L}_c += (V_t - \hat{R}_t)^2$
    end for
    分别用梯度 $\nabla_\theta\mathcal{L}_a$ 和 $\nabla_\phi\mathcal{L}_c$ 更新参数 θ 和 ϕ
  end for
end for

2.4.4　性能评估

本节比较强化学习算法在有预测和没有预测的情况下的控制效果。首先展示了光伏输出功率、基础负荷功率和电价的预测效果，然后给出了两种方案比较的仿真结果和相应的解释和结论。

在监督学习的训练过程中，光伏输出功率、基础负荷功率和电价预测的 MSE 损失如图2.5(a)、图2.5(c)和图2.5(e)，从这三幅图中可以看出，MSE 损失迅速下降并最终收敛，说明 RNN 训练稳定。



(a) 光伏输出功率的预测损失　(b) 光伏输出功率的预测效果

(c) 基础负荷功率的预测损失　(d) 基础负荷功率的预测效果

(e) 电价的预测损失　(f) 电价的预测效果

图 2.5　监督学习训练曲线和预测效果

图2.5(b)、图2.5(d)和图2.5(f)分别展示了两天的光伏输出功率、基础负荷功率和电价预测结果。从这三幅图中可以看出，1 步预测比 2 步预测更准确，这也可以从训练过程中的损失曲线中看出。为定量评估预测效果，表2.1采用平均绝对百分

表 2.1　监督学习的预测效果

	一步预测		两步预测	
指标	MAPE	RMSE	MAPE	RMSE
光伏输出功率	17.7%	31.0	31.0%	47.0
基础负荷功率	3.0%	13.3	5.7%	22.8
电价	8.2%	0.0046	11.4%	0.0058

比误差（MAPE）和均方根误差（RMSE）指标进行评估。该表说明，两步预测的误差相较于一步预测明显上升。此外，由于光伏输出功率的波动性和不确定性相较于基础负荷功率预测和电价预测更强，其预测效果较差。

对于强化学习的训练，Actor 网络和 Critic 网络的学习率分别设置为 3×10^{-4} 和 1×10^{-3}。其他重要的算法参数见表A.1。为展示强化学习训练过程中这两种方案的控制效果，图 2.6绘制了每天平均奖励 ($R = \sum_{t=1}^{T} r_t$)。可以明显看出，没有预测的能量管理方案比有预测的能量管方案具有更高的平均奖励，即微电网的运营成本更低。

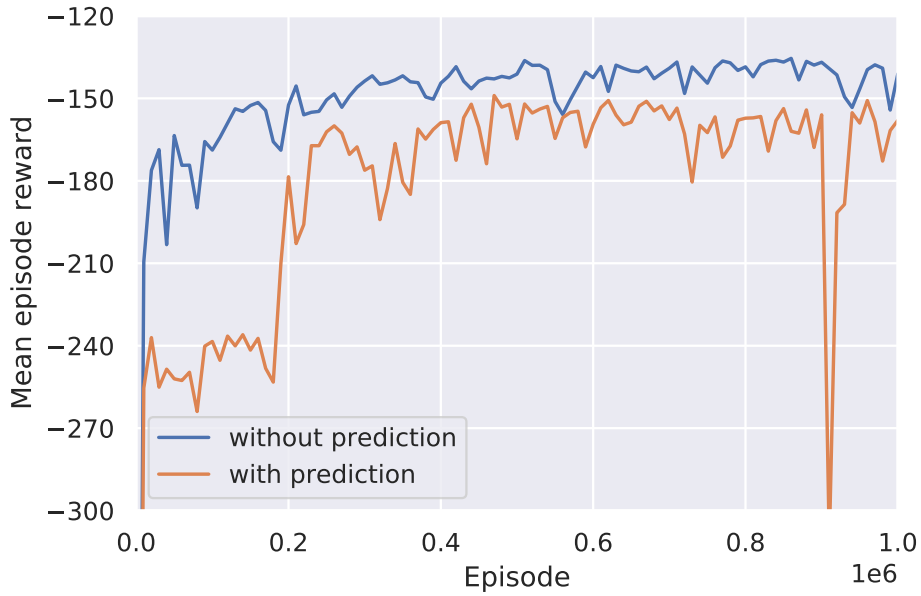


图 2.6　有预测和无预测的强化学习训练曲线

在没有预测的能量管理方案下，储能设备的充放电功率与电价的关系曲线如图 2.7所示。可以看出，没有预测的方案能够学会在电价低谷时充电，在电价高峰时放电。这些充电/放电模式表明，基于强化学习的能量管理方案可以在没有显式预测环节的条件下适应不断变化的电价。

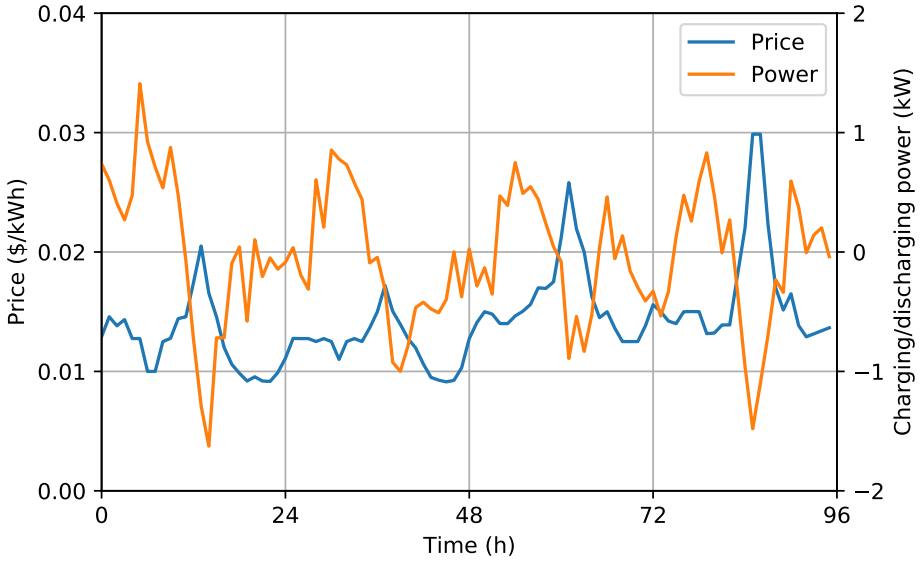


图 2.7　无预测强化学习方案下储能设备充放电功率与电价的关系

2.4.5　结论

通过设计这个简单的实验，我们可以得出结论：如果预测环节仅仅基于强化学习的状态信息，那么在强化学习的输入中加入对未来信息的预测不会对强化学习的性能产生正面影响。因此，在之后的算法设计中，直接将当前观测作为强化学习网络的输入即可。

第 3 章　微网运营商集中管理下的隐私保护负荷控制方案

集中式控制作为理论研究最深入的控制方法，因其可靠性在实践中也应用最广泛。本章提出了一种孤岛微电网内住宅负荷集中式控制方案。在该方案中，由微电网运营商集中管理包括电动汽车和空调在内的可控电力设备。出于保护住户数据隐私的目的，微电网运营商无法观察到诸如电动汽车的到达和离开时间、空调控制的室内温度等住户隐私信息，故而该控制问题为 POMDP。为了适应由大量可控电力设备带来的高维连续动作空间，本章通过引入信用分配机制，提出了一种新的强化学习算法，其中整合了循环神经网络以缓解由隐私保护带来的部分可观测性问题。

3.1　总体思路

本章提出了一种可以最小化孤岛微电网运营成本和住户用电体验损失的隐私保护负荷控制方案。具体来说，空调控制的室内温度信息被完全保留，电动汽车到达和离开时间也不会被事先披露给微电网运营商。为了适应隐私保护导致的部分可观测性，该控制问题被描述为部分可观测马尔可夫决策过程（POMDP），而不是标准马尔可夫决策过程（MDP）。本章以 A2C 算法为基础，提出了向量 A2C 算法，并整合了循环神经网络，从而使得提出的隐私保护负荷控制方案能够解决高维动作空间和部分可观测性问题。本章最后通过仿真实验验证了向量 A2C 算法相较于标准 A2C 算法的优越性，在隐私保护方面，所提方案相较于现有隐私保护负荷控制方案同样具有优势。

具体而言，本章的研究意义和贡献主要集中于如下 4 个方面：

- 本章提出了一种考虑用户体验的隐私保护负荷控制方案。不同于现有的空调控制方案[4-8]，本研究完全保留了室内温度和热舒适损失函数，因而消除了相应的隐私泄露的可能性。同时，与先前研究电动汽车充电调度的文章[9-12]相比，所提方案无需事先获知电动汽车到达和离开时间，进一步加强了对用户隐私的保护。
- POMDP 被用于描述微电网内的隐私保护负荷控制问题。与大多数使用基于标准 MDP 的强化学习算法的负荷控制方案[35-36,48-49]相比，POMDP 提供了一个考虑隐私问题的框架。相应地，一个集成了循环神经网络和多层感知器的深度神经网络被用于从有限的观测中提取信息。
- 基于强化学习的控制方案无需预先获取用户行为和系统模型。此外，通过将

信用分配机制引入 A2C 算法，本章提出了向量 A2C 算法。该算法在处理高维动作空间时，有效减小值函数估计的方差，使强化学习训练过程更稳定高效。

- 仿真结果表明了提出的向量 A2C 算法相较于标准 A2C 算法的优越性。此外，与现有的隐私保护方案相比，本章所提方案以更低的能耗和运行成本给住户提供了更好的用电体验。同时，该方案的灵活性和可扩展性也得以验证。

3.2　场景设计

3.2.1　场景描述

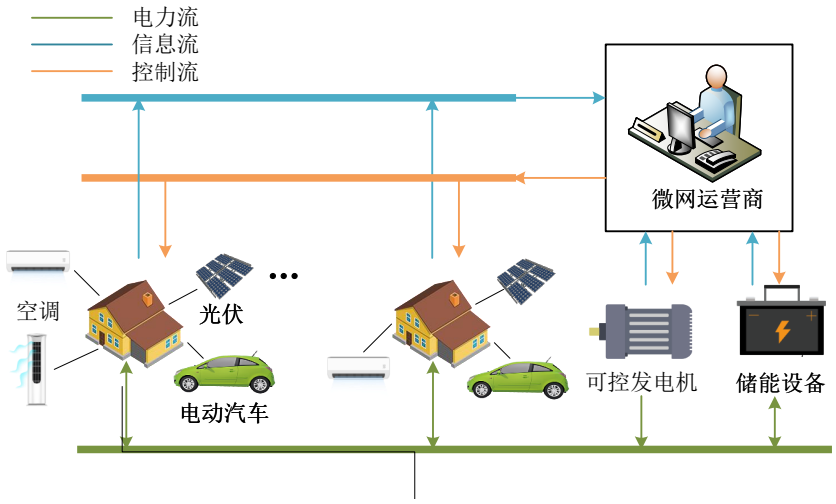


图 3.1　微网运营商直接管理的负荷协同控制方案

如图3.1所示，本章所考虑的负荷协同控制场景由微电网运营商、住宅、可控发电机和储能设备组成。在每个时刻 t，微网运营商从每个住宅和其他电力设备处收集信息，并利用它们做出连续控制信号，从而最小化整个微网的运营成本并保证用户的用电体验。在这个微网中，每个住宅包括空调、电动汽车、光伏和基础负荷。假设整个微网的空调和电动汽车总数分别为 M 和 N，则空调和电动汽车集合分别为 $\{1,...,i,...,M\}$ 和 $\{1,...,j,...,N\}$。简单起见，所有住宅的基础负荷视为一个整体，同样所有住宅的光伏也视为一个不可控发电设备。

3.2.2　部分可观测马尔可夫决策过程建模

由于微网运营商不能观测到系统的全部信息，因此使用 POMDP 对该问题建模。

状态：时刻 t 的状态 s_t 包含微网内全部信息，包括所有光伏的总发电功率 P_t^{PV}，

基础负荷的总功率 P_t^{BL}，发电机的输出功率 P_t^{CG}，电池的充电状态 SOC_t，住宅的室外温度 T_t^{out}，所有电动汽车的电量 $E_t^{\mathrm{EV}} = [E_{1,t}^{\mathrm{EV}}, ..., E_{N,t}^{\mathrm{EV}}]$，所有空调控制的室内温度 $T_t^{AC} = [T_{1,t}^{\mathrm{AC}}, ..., T_{M,t}^{\mathrm{AC}}]$，所有空调的使用状态 $S_t^{AC} = [S_{1,t}^{\mathrm{AC}}, ..., S_{M,t}^{\mathrm{AC}}]$，其中二元变量 $S_{i,t}^{\mathrm{AC}}$ 表示空调 i 在时刻 t 是否被住户使用。注意空调可以被微网运营商在任何时间控制，即使住户不在室内。因此，状态 s_t 表示为

$$s_t = \left[P_t^{\mathrm{PV}}, P_t^{\mathrm{BL}}, T_t^{\mathrm{out}}, P_t^{\mathrm{CG}}, \mathrm{SOC}_t, E_t^{\mathrm{EV}}, T_t^{\mathrm{AC}}, S_t^{\mathrm{AC}}\right], \tag{3.1}$$

观测：在大多数现有的涉及空调控制的文献中，室内温度被默认用作控制空调的信息。同样，电动汽车充电调度也需要提前获知电动汽车的到达时间和离开时间。但出于隐私考虑，上述所有隐私信息均无法被微电网运营商获取。因此，观测 o_t 定义为

$$o_t = \left[P_t^{\mathrm{PV}}, P_t^{\mathrm{BL}}, T_t^{\mathrm{out}}, P_t^{\mathrm{CG}}, \mathrm{SOC}_t, E_t^{\mathrm{EV}}, S_t^{\mathrm{AC}}\right], \tag{3.2}$$

动作：在考虑的微电网中，可控电力设备包括发电机、M 个空调和 N 个电动汽车。在连续动作空间 $\mathcal{A}$ 中的动作 a_t 定义为

$$a_t = \left[u_t^{\mathrm{CG}}, P_{1,t}^{\mathrm{AC}}, ..., P_{M,t}^{\mathrm{AC}}, P_{1,t}^{\mathrm{EV}}, ..., P_{N,t}^{\mathrm{EV}}\right], \tag{3.3}$$

其中，$u_t^{DG}, P_{i,t}^{\mathrm{AC}}, P_{j,t}^{\mathrm{EV}}$ 为发电机、空调 i 和电动汽车 j 的控制信号。控制信号 $u_t^{DG}, P_{i,t}^{\mathrm{AC}}, P_{j,t}^{\mathrm{EV}}$ 只能在范围 $[0,1]$，$[0, P_{i,\max}^{\mathrm{AC}}]$ 和 $[0, P_{j,\max}^{\mathrm{EV}}]$ 内取值，其中 $P_{i,\max}^{\mathrm{AC}}$ 和 $P_{j,\max}^{\mathrm{EV}}$ 分别表示空调 i 最大工作功率和电动汽车 j 最大充电功率。动作 a_t 维度表示为 $K = 1 + M + N$。由于储能设备负责维持孤岛微电网功率平衡，储能设备的充放电功率 P_t^{BES} 不是自由变量，而是由以下等式确定

$$P_t^{\mathrm{BES}} = \sum_{i=1}^{M} P_{i,t}^{\mathrm{AC}} + \sum_{j=1}^{N} P_{j,t}^{\mathrm{EV}} + P_t^{\mathrm{BL}} - P_t^{\mathrm{PV}} - P_t^{\mathrm{CG}}. \tag{3.4}$$

因此 P_t^{BES} 不是动作的一部分。

状态转移动力学：大部分状态分量的转移动力学函数已经在上一章给出，只有空调使用状态 $S_{i,t}^{\mathrm{AC}}$ 尚未讨论。$S_{i,t}^{\mathrm{AC}}$ 的变化反映了住户对空调 i 使用行为的改变，由两个随机变量决定

$$S_{i,t}^{\mathrm{AC}} = \begin{cases} 1, & \text{if } a_i^{\mathrm{AC}} \leqslant t < d_i^{\mathrm{AC}}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

其中，a_i^{AC} 和 d_i^{AC} 表示住户进入和离开空调 i 所在房间的时间。出于隐私保护，a_i^{AC} 和 d_i^{AC} 无法提前被微网运行商获知。

奖励函数：奖励函数设置对引导强化学习策略训练有极为重要的作用。在所设计的负荷协同控制场景中，合理的奖励函数应该既考虑经济成本，又计及住户

的用电体验。因此，时刻 t 的奖励 r_t 由四部分组成：可控发电机的发电成本，储能设备的老化成本，用户的热舒适度损失和电动汽车充电不满意度，

$$r_t = -\left(C_t^{\mathrm{CG}} + C_t^{\mathrm{BES}} + \sum_{i=1}^{M} C_{i,t}^{\mathrm{AC}} + \sum_{j=1}^{N} C_{j,t}^{\mathrm{EV}} \right). \tag{3.6}$$

时刻 t 可控发电机的发电成本 C_t^{CG} 定义为

$$C_t^{\mathrm{CG}} = g^{\mathrm{CG}}\left(P_t^{\mathrm{CG}}\right) \Delta \mathrm{t}, \tag{3.7}$$

其中 $g^{DG}(\cdot)$ 表示可控发电机关于发电功率的单位时间发电成本函数。

时刻 t 储能设备的老化成本 C_t^{BES} 与（2.36）中关于储能设备成本的描述相同，定义为

$$C_t^{\mathrm{BES}} = g^{\mathrm{BES}}\left(\mathrm{SOC}_t, P_t^{\mathrm{BES}}\right) \Delta \mathrm{t}, \tag{3.8}$$

其中 $g^{\mathrm{BES}}(\cdot,\cdot)$ 表示储能设备关于当前电量状态和充放电功率的单位时间老化成本函数，即 $g^{\mathrm{BES}}(\mathrm{SOC}, P^{\mathrm{BES}})$ 表示储能设备在电量状态为 SOC 时以功率 P^{BES} 充/放电单位时间的老化成本。

时刻 t 空调 i 给房间住户带来的热舒适度损失 $C_{i,j}^{\mathrm{AC}}$ 定义为

$$C_{i,t}^{\mathrm{AC}} = \begin{cases} g_{i,t}^{\mathrm{AC}}\left(T_{i,t}^{\mathrm{AC}}\right), & \text{if } S_{i,t}^{\mathrm{AC}} = 1, \\ 0, & \text{if } S_{i,t}^{\mathrm{AC}} = 0, \end{cases} \tag{3.9}$$

其中 $g_{i,t}^{\mathrm{AC}}(\cdot)$ 表示空调 i 所处房间的住户关于室内温度的热舒适度损失函数。公式（3.9）说明只有住户处于空调所在房间，舒适度损失才能发生。

时刻 t 电动汽车 j 的充电过程引起的住户不满意度 $C_{j,t}^{\mathrm{EV}}$ 定义为

$$C_{j,t}^{\mathrm{EV}} = \begin{cases} g_{i,t}^{\mathrm{EV}}\left(E_j^{\mathrm{targ}} - E_{j,t}^{\mathrm{EV}}\right), & \text{if } t = d_i^{\mathrm{EV}}, \\ 0, & \text{if } t \neq d_i^{\mathrm{EV}}, \end{cases} \tag{3.10}$$

其中 $E_j^{\mathrm{targ}} - E_{j,t}^{\mathrm{EV}}$ 表示电动汽车 j 的充电任务剩余电量，$g_{i,t}^{\mathrm{EV}}(\cdot)$ 表示住户在离开住宅时对电动汽车 j 剩余待充电电量的不满意度函数。$C_{j,t}^{\mathrm{EV}}$ 只有当住户离开住宅时才发生。值得一提的是，作为保护住户隐私的措施之一，电动汽车 j 的离开时间 d_i^{EV} 不会被微网运营商提前了解。

3.3　算法设计

上一节将隐私保护的负荷协同控制问题建模成 POMDP，这给强化学习训练带来了两个主要的挑战。第一，随着住宅可控电力设备数量的增加，POMDP 的动作

空间维度也随之增加，寻找状态空间到动作空间最优映射的难度也指数级增加。第二，如何在缺少关键状态信息的条件下做出合理决策从而保障住户的用电体验。为了处理上述挑战，本节首先在 A2C 算法的基础上发展了向量 A2C 算法，有效解决了高维动作空间问题。其次，将循环神经网络融入强化学习网络，一定程度上缓解了部分可观测困境。

3.3.1　向量 A2C 算法

标准的 A2C 算法将动作 a_t 视为一个整体，只考虑总体奖励 r_t 而不考虑动作 a_t 的每个分量对总体奖励的贡献。这会导致优势函数的估计出现非常大的方差，也会严重阻碍最优策略的探索，特别面对高维动作空间问题。例如，在上一节的 POMDP 建模中，假设微网运营商在观测到 o_t 情况下，选择的动作 a_t 中的发电机控制信号 u_t^{CG} 非常差，而动作 a_t 中的其他分量非常好，这会导致总的奖励比预期要好，也就是说，优势函数 A 大于零。在这种情况下，尽管控制发电机的动作分量很糟糕，但 Actor 网络将会向增加这个动作的概率的方向更新。因此，学习各个动作分量之间的协同策略将会变得非常困难。

为了解决这个问题，我们引入了信用分配机制来分解总体奖励 r_t。根据3.6中 r_t 的定义，总体奖励的某些组成部分是可分解的，这给分配总体奖励到每个动作分量提供了可能。具体来说，发电机的发电成本仅由发电机的控制信号 $u\mathrm{CG}_t$ 决定，与 a_t 的其他分量无关。类似地，用户充电任务不满意度和热舒适损失只受充电功率和空调功率的影响。相反，根据功率平衡约束，电池的老化成本受到动作所有分量的影响，不能分配给某个特定的动作分量。至此，我们可以为每个动作分量创建一个单独奖励，从而消除其他动作分量的噪声，体现其本身对总体奖励的影响，

$$\begin{cases} r_t^{\mathrm{CG}} = -C_t^{\mathrm{CG}} - C_t^{\mathrm{BES}}, \\ r_{i,t}^{\mathrm{AC}} = -C_{i,t}^{\mathrm{AC}} - C_t^{\mathrm{BES}}, \quad i = 1, ..., M, \\ r_{j,t}^{\mathrm{EV}} = -C_{j,t}^{\mathrm{EV}} - C_t^{\mathrm{BES}}, \quad i = 1, ..., N. \end{cases} \tag{3.11}$$

该信用分配方法被称为*差分奖励*[50]。

所有动作分量的单独奖励可以组成如下的奖励向量

$$\boldsymbol{r}_t = \left[r_t^{\mathrm{CG}}, r_{1,t}^{\mathrm{AC}}, ..., r_{M,t}^{\mathrm{AC}}, r_{1,t}^{\mathrm{EV}}, ..., r_{N,t}^{\mathrm{EV}} \right], \tag{3.12}$$

基于该奖励向量，时刻 t 到时刻 T 的累积奖励也可以相应地修改为向量形式

$$\boldsymbol{R}_t = \sum_{t'=t}^{T} \boldsymbol{r}_{t'}. \tag{3.13}$$

对累积奖励的期望进行估计的值函数也变更为向量形式

$$\boldsymbol{V}^{\pi}(o_t) = \mathbb{E}_{s_{t+1:T}\sim P_s, o_{t+1:T}\sim P_o, a_{t:T}\sim\pi}\left[\sum_{t'=t}^{T}\boldsymbol{r}_{t'}|o_t\right]. \tag{3.14}$$

在此基础上，A2C 算法的多步优势函数2.9被重新定义为

$$\boldsymbol{A}(o_t, a_t; \phi) = \sum_{t'=t}^{t+k-1}\left(\boldsymbol{r}_{t'} + \boldsymbol{V}(o_{t+k}; \phi) - \boldsymbol{V}(o_t; \phi)\right). \tag{3.15}$$

为了适应向量化的优势函数，对动作 $a_t = [a_t^{(1)}, ..., a_t^{(K)}]$，定义策略 π 的向量表示

$$\boldsymbol{\pi}(a_t|o_t;\theta) = [\pi(a_t^{(1)}|o_t;\theta), ..., \pi(a_t^{(K)}|o_t;\theta)]. \tag{3.16}$$

（2.10）和（2.11）关于 Actor 网络和 Critic 网络参数的梯度相应地修改为

$$\triangle\theta = \sum_{t'=t}^{t+t_{max}}\nabla_{\theta}\left[\log\boldsymbol{\pi}(a_{t'}|o_{t'};\theta)\cdot\boldsymbol{A}(o_{t'}, a_{t'};\phi)\right] \tag{3.17}$$

$$\triangle\phi = \sum_{t'=t}^{t+t_{max}}\partial\left\|\boldsymbol{A}(o_{t'}, a_{t'};\phi\right\|^2/\partial\phi \tag{3.18}$$

注意公式3.17中 $\log\boldsymbol{\pi}(a_{t'}|o_{t'};\theta)$ 和 $\boldsymbol{A}(o_{t'}, a_{t'};\phi)$ 为向量内积，即:

$$\log\boldsymbol{\pi}(a_{t'}|o_{t'};\theta)\cdot\boldsymbol{A}(o_{t'}, a_{t'};\phi) = \sum_{i=1}^{K}\log\pi(a_t^{(i)}|o_t;\theta)A(o_t; a_t^{(i)};\phi). \tag{3.19}$$

算法3.1详细展示了向量 A2C 算法的执行过程。为了提高采样效率，算法设定智能体与多个环境分支同时进行交互。智能体每与环境交互 $t_{\max}$ 步进行一次参数更新，若提前到达终点时刻 $\boldsymbol{T}$，则利用已收集的数据进行参数更新。各个时刻的累积奖励通过反向时间计算：首先计算最后时刻的累积奖励，若为终点时刻，设置为 0，否则用向量 Critic 网络 $\boldsymbol{V}(\cdot;\phi)$ 估计；之后逐步向前迭代计算，直至初始时刻。

3.3.2　隐私保护的负荷协同控制方案

如本节开头所述，部分可观察性为基于策略的强化学习算法解决 POMDP 问题带来了挑战。为了解决该问题，受 RNN 在时序任务中的亮眼表现的启发，RNN 结构来增强神经网络的表征能力。

在这项工作中，如图3.2所示，GRU 与 Actor 网络和 Critic 网络级联。与长短时记忆网络（Long Short Term Memory）类似，GRU 也已经被证明能够解决梯度消失问题。而且 GRU 以更简单的门控结构，具有跟 LSTM 相当的性能表现。以 θ_r 为参

算法 3.1 向量 A2C 算法

随机初始化 θ 和 ϕ
$t \leftarrow 0, T_{\text{total}} \leftarrow 0$
repeat
　重置梯度：$\triangle\theta \leftarrow 0, \triangle\phi \leftarrow 0$
　for 每个环境分支 **do**
　　收到观测 o_t，$t_{\text{start}} \leftarrow t$
　　repeat
　　生成策略 $\pi(\cdot|o_t;\theta)$ 并选择动作 $a_t \sim \pi(\cdot|o_t;\theta)$
　　执行动作 a_t 并收到下一刻的观测 o_{t+1} 和总体奖励 r_t
　　根据（3.11）分解总体奖励 r_t，得到奖励向量 $\boldsymbol{r}_t$
　　$t \leftarrow t+1$
　　until $t = T$ **or** $t - t_{\text{start}} == t_{\max}$
　　if $t = T$ **then**
　　$\boldsymbol{R} = \boldsymbol{0}$
　　else
　　$\boldsymbol{R} = \boldsymbol{V}(o_t;\phi)$
　　end if
　　for $t' \in \{t-1, t-2, ..., t_{\text{start}}\}$ **do**
　　$\boldsymbol{R} \leftarrow \boldsymbol{r}_{t'} + \boldsymbol{R}$
　　$\triangle\theta \leftarrow \triangle\theta + \nabla_\theta \left[\log \boldsymbol{\pi}(a_{t'}|o_{t'};\theta) \cdot \left(\boldsymbol{R} - \boldsymbol{V}(o_{t'}, a_{t'};\phi)\right)\right]$
　　$\triangle\phi \leftarrow \triangle\phi + \partial \left\|\boldsymbol{R} - \boldsymbol{V}(o_{t'}, a_{t'};\phi\right\|^2 / \partial\phi$
　　end for
　end for
　分别利用梯度 $\triangle\theta$ 和 $\triangle\phi$ 更新参数 θ 和 ϕ
until $T_{\text{total}} > T_{\max}$

数的 GRU 可以表示为

$$h_t = \text{GRU}(h_{t-1}, o_t; \theta_r) \tag{3.20}$$

其中 h_{t-1} 为 GRU 上一时刻的隐状态，其与当前时刻的观测 o_t 一起作为 GRU 的输入。得益于循环结构，GRU 能够学会记忆之前时刻的重要信息。

在实践中，随机策略 π 通常表示为具有对角化协方差矩阵的高斯分布 $\mathcal{N}(\mu, \sigma^2)$，其中 $\mu = [\mu_1, ..., \mu_K]$ 和 $\sigma^2 = [\sigma_1^2, ..., \sigma_K^2]$ 为 K 维向量，分别决定动作每个分量的均值和方差。如图3.2所示，均值 μ 和方差 σ^2 分别由两个不同的 Sigmoid 输出层产生，共享两层具以 Tanh 为激活函数的隐藏层。类似地，具有线性输出的三层全连接层用于产生值估计 V。Actor 网络和 Critic 网络共享 GRU 层。

图3.2绘制了基于向量 A2C 算法的隐私保护负荷协同控制方案的流程图。GRU 层以当前的观测 o_t 和上一时刻的隐状态 h_{t-1} 作为输入，输出当前时刻的隐状态 h_t。Actor 和 Critic 分别以 GRU 层当前时刻的隐状态 h_t 为输入，输出策略的均值 μ 和方差 σ^2，以及值函数向量 $\boldsymbol{V}$。微网运行商根据高斯分布 $\mathcal{N}(\mu, \sigma^2)$ 采样得到动作 a_t，

并在负荷协同控制环境中执行。同时，上一时刻的总体奖励 r_{t-1} 被信用分配机制（3.11）扩充为奖励向量 $\boldsymbol{r}_{t-1}$，然后用于计算累积奖励向量 $\boldsymbol{R}$。接下来，值函数向量 $\boldsymbol{V}$ 和累积奖励向量 $\boldsymbol{R}$ 之差，优势向量 $\boldsymbol{A}$，被用于计算值函数损失（优势向量的 2 范数）和策略损失（优势向量和概率向量 $\boldsymbol{\pi}(a_t|o_t;\theta)$）。将值函数损失和策略损失反向梯度传播可以得到网络参数的梯度，据此可进行网络参数更新。

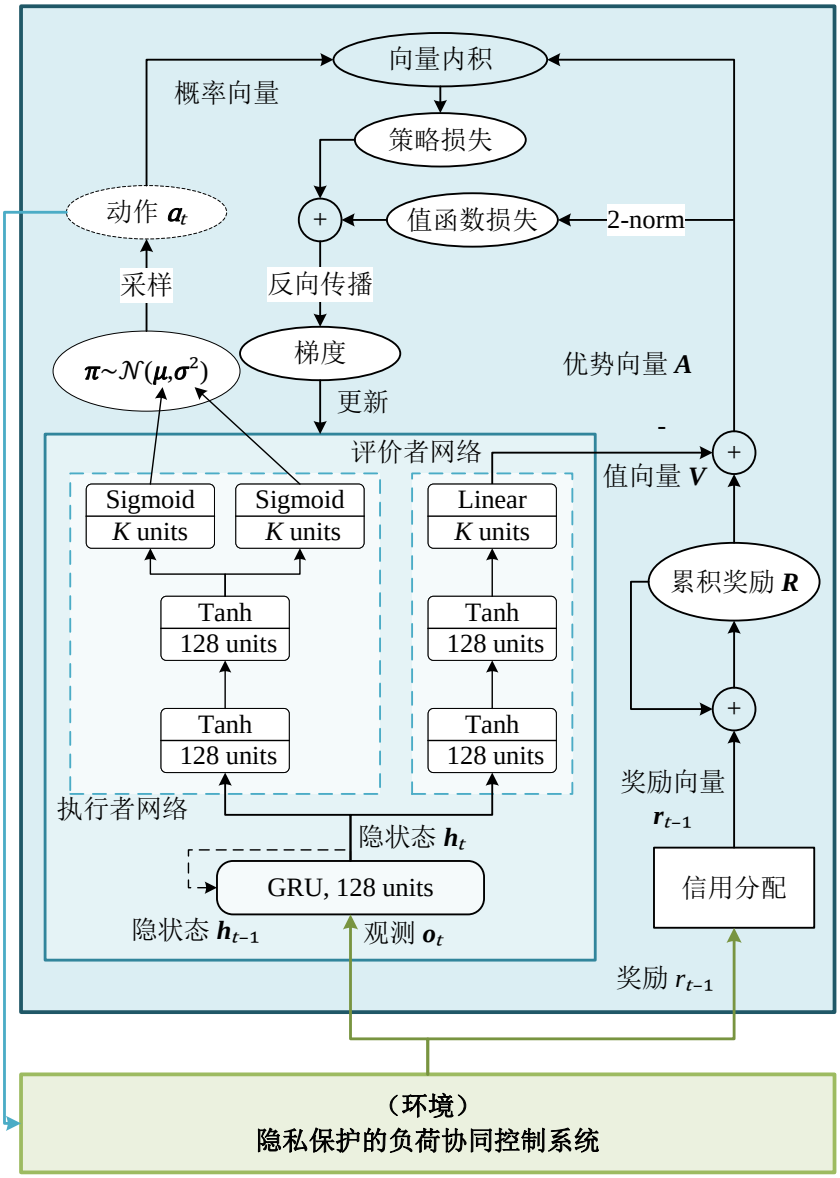


图 3.2　基于向量 A2C 算法的隐私保护负荷协同控制方案流程图

3.4　仿真实验

本节验证了提出的向量 A2C 算法的控制效果，并评估了所提出的隐私保护负荷协同控制方案的有效性。首先描述了仿真实验的设置。然后，介绍了四种用于性能比较的基线方案。最后给出了不同方案的仿真结果和相应的讨论。

3.4.1　实验设置

负荷协同控制的时间间隔 Δt 为 5 分钟，总时长为 24 小时，因此总步数 $T = 288$。通过对真实电动汽车充电功率数据和空调功率数据的分析，电动汽车的到达时间和离开时间以及空调的使用时间基本符合均匀分布。

仿真实验采用了特定形式和参数的转移函数和成本函数，部分展示如下，其他见附录。需要强调的是，微电网运营商不会将这些信息作为先验知识。因此，本章提出的无模型算法也可以解决具有其他函数形式和参数甚至没有显式转移函数的问题。

空调 i 所在房间室内温度的转移函数采用等价热力学模型[6]的离散形式

$$f_i^{\mathrm{AC}}(T, T^{\mathrm{out}}, P, \varrho) = T + \left(1 - \exp\left(-\eta_{i,1}^{AC}\Delta \mathrm{t}\right)\right)\left(T^{\mathrm{out}} - T - \eta_{i,2}^{\mathrm{AC}} P\right) + \varrho, \tag{3.21}$$

其中参数 $\eta_{i,1}^{AC}$ 和 $\eta_{i,2}^{AC}$ 分别与房间和空调的热力学特性有关。如2.3.3所述，控制空调的目的是满足用户的温度需求，即把室内温度控制在热舒适区 $[T_{i,\min}^{\mathrm{AC}}, T_{i,\max}^{\mathrm{AC}}]$ 内。因此热舒适度损失函数仅在该区域之外发生，且距离该温度区越远，热损失越大，仿真实验采取以下热舒适度损失函数

$$g_i^{\mathrm{AC}}(T) = \begin{cases} 0, & T_{i,\min}^{\mathrm{AC}} \leqslant T \leqslant T_{i,\max}^{\mathrm{AC}}, \\ \lambda_{i,1}^{\mathrm{AC}} \exp\left[\lambda_{i,2}^{\mathrm{AC}}(T - T_{i,\max}^{\mathrm{AC}})\right], & T > T_{i,\max}^{\mathrm{AC}}, \\ \lambda_{i,1}^{\mathrm{AC}} \exp\left[\lambda_{i,2}^{\mathrm{AC}}(T_{i,\min}^{\mathrm{AC}} - T)\right], & T < T_{i,\min}^{\mathrm{AC}}, \end{cases} \tag{3.22}$$

其中 $\lambda_{i,1}^{\mathrm{AC}}$ 和 $\lambda_{i,2}^{\mathrm{AC}}$ 是与空调 i 房间的住户有关的权重参数。显然，该热舒适度损失函数关于 $\left(T_{i,\min}^{\mathrm{AC}} + T_{i,\max}^{\mathrm{AC}}\right)/2$ 对称。室内温度转移函数和热舒适度损失函数的具体参数如表3.1所示。室内温度的扰动变量 ϱ 服从均匀分布 $\mathcal{U}(-0.05, 0.05)$。

电动汽车充电的不满意度函数定义为二次函数

$$g_j^{\mathrm{EV}}(E) = \lambda_{j,1}^{\mathrm{EV}} E + \lambda_{j,2}^{\mathrm{EV}} E^2, \tag{3.23}$$

其中 E 为剩余待充电电量，$\lambda_{j,1}^{\mathrm{EV}}$ 和 $\lambda_{j,2}^{\mathrm{EV}}$ 为与电动汽车 j 车主对未完成充电任务忍耐度有关的权重参数。包括最大充电功率在内的具体参数如表3.2所示。

算法层面，为提高数据采样效率，8 个环境分支同时与环境进行交互。为了防止过拟合，100 天的历史数据被随机分为训练集、验证集和测试集，比例分别为

表 3.1　室内温度转移函数和用户热舒适度损失函数的参数

空调编号 i	$P_{i,\max}^{\mathrm{AC}}$	$\eta_{i,1}^{\mathrm{AC}}$	$\eta_{i,2}^{\mathrm{AC}}$	a_i^{AC}	d_i^{AC}	$T_{i,\min}^{\mathrm{AC}}$	$T_{i,\max}^{\mathrm{AC}}$	$\lambda_{i,1}^{\mathrm{AC}}$	$\lambda_{i,2}^{\mathrm{AC}}$
1	1 kW	2.50	17.7	$\mathcal{U}(11:00,12:00)$	$\mathcal{U}(13:00,15:00)$	23.1	26.3	0.352	0.396
2	1 kW	2.27	15.4	$\mathcal{U}(12:45,13:45)$	$\mathcal{U}(14:45,16:45)$	21.0	24.7	0.694	0.349
3	2 kW	2.91	8.50	$\mathcal{U}(11:05,12:05)$	$\mathcal{U}(13:05,15:05)$	24.4	26.7	0.467	0.317
4	2 kW	2.38	8.45	$\mathcal{U}(11:55,12:55)$	$\mathcal{U}(13:55,15:55)$	22.8	26.3	0.775	0.289
5	2 kW	2.85	8.77	$\mathcal{U}(13:50,14:50)$	$\mathcal{U}(15:50,17:50)$	22.5	26.1	0.454	0.331
6	2 kW	1.71	8.25	$\mathcal{U}(12:20,13:20)$	$\mathcal{U}(14:20,16:30)$	22.8	26.7	0.421	0.376
7	2 kW	1.71	8.50	$\mathcal{U}(10:35,11:35)$	$\mathcal{U}(12:35,14:35)$	22.6	26.6	1.24	0.212
8	3 kW	2.71	5.13	$\mathcal{U}(12:25,13:25)$	$\mathcal{U}(14:25,16:25)$	20.8	24.5	1.10	0.215
9	3 kW	2.10	5.03	$\mathcal{U}(12:40,13:40)$	$\mathcal{U}(14:40,16:40)$	23.9	26.5	1.03	0.263
10	3 kW	1.75	5.86	$\mathcal{U}(12:50,13:50)$	$\mathcal{U}(14:50,16:50)$	20.6	24.0	0.670	0.303

表 3.2　电动汽车充电相关参数

电动汽车编号 j	$P_{j,\max}^{\mathrm{EV}}$	$\lambda_{j,1}^{\mathrm{EV}}$	$\lambda_{j,2}^{\mathrm{EV}}$	a_j^{EV}	d_j^{EV}
1	3.4 kW	1.12	0.020	$\mathcal{U}(8:15,9:15)$	$\mathcal{U}(11:15,13:15)$
2	3.4 kW	1.70	0.034	$\mathcal{U}(8:55,9:55)$	$\mathcal{U}(11:55,13:55)$
3	6.8 kW	1.57	0.023	$\mathcal{U}(8:00,9:00)$	$\mathcal{U}(11:00,13:00)$
4	6.8 kW	1.56	0.033	$\mathcal{U}(9:20,10:20)$	$\mathcal{U}(12:20,14:20)$
5	2.0 kW	1.14	0.029	$\mathcal{U}(9:10,10:10)$	$\mathcal{U}(12:10,14:10)$

80%、10% 和 10%。Actor 网络和 Critic 网络参数的学习率设置为 0.0002 和 0.0001。每次更新的步数 $t_{\max}$ 设置为 24。强化学习算法利用 Python 的 PyTorch 实现。仿真实验在 8 核 AMD Ryzen 7 3700X 处理器和一个 GeForce RTX 2080 GPU 上进行。

3.4.2　基线方案

本章提出的隐私保护负荷协同控制方案将与四种基线方案进行对比:

- 基线方案 1：使用标准 A2C 算法的隐私保护负荷协同控制方案。A2C 算法的网络结构与图3.2完全相同，除了 Critic 网络输出层的单元数量（向量 A2C 算法是 K 个，标准 A2C 算法是 1 个）。通过将本章提出的方案与基线方案 1 进行比较，向量 A2C 算法的优势得以显现。
- 基线方案 2：不考虑隐私保护的负荷协同控制方案。该方案允许微网运营商获取微网内的全部信息，即微网状态是完全可观测的，因此不必使用循环神经网络。通过将本章提出的方案与基线方案 2 进行比较，可以揭示由隐私保护机制引起的损失。
- 基线方案 3：一种现有的隐私保护负荷协同控制方案。该方案使用了一种无需室内温度信息的空调控制策略，温度设定控制[51]。在该方案下，微电网运营商设定每台空调的温度，然后由住宅根据所设定的温度决定每台空调的工作功率：当室内温度高于设定点时，空调以最大功率运行; 否则空调关闭。这

种控制策略数学上可以表示如下

$$P_{i,t}^{\mathrm{AC}}=\begin{cases}P_{i,\max}^{\mathrm{AC}}, & T_{i,t}^{\mathrm{AC}}>T_{i,t,\mathrm{set}}^{\mathrm{AC}}\\ 0, & T_{i,t}^{\mathrm{AC}}\leqslant T_{i,t,\mathrm{set}}^{\mathrm{AC}}\end{cases} \tag{3.24}$$

其中 $T_{i,\mathrm{set}}^{\mathrm{AC}}$ 表示空调 i 的温度设定值。该方案通过向量 A2C 算法进行优化。算法在每个时刻的动作为

$$a_t=\left[u_t^{DG},P_{1,t,\mathrm{set}}^{\mathrm{AC}},...,T_{M,t,\mathrm{set}}^{\mathrm{AC}},P_{1,t}^{\mathrm{EV}},...,P_{N,t}^{\mathrm{EV}}\right], \tag{3.25}$$

- 基线方案 4: 最大化用户体验的隐私保护负荷协同控制方案。在该方案下，微网运营商决定电动汽车和空调的功率时不考虑经济成本，全力满足用户的用电需求。因此，（3.11）中的信用分配机制更改为

$$\begin{cases}r_t^{\mathrm{CG}}=-C_t^{\mathrm{CG}}-C_t^{\mathrm{BES}}\\ r_{i,t}^{\mathrm{AC}}=-C_{i,t}^{\mathrm{AC}},i=1,...,M\\ r_{j,t}^{\mathrm{EV}}=-C_{j,t}^{\mathrm{EV}},j=1,...,N\end{cases} \tag{3.26}$$

与（3.11）相比，（3.26）中分配给电动汽车控制信号和空调控制信号的奖励舍弃了储能设备的老化成本，因此该分配机制下的学到的策略会最小化用户舒适度损失和不满意度。

3.4.3　仿真结果

为了评估不同方案在训练过程的控制效果，图3.3展示了 10 个不同的随机数种子在验证集上的平均累积奖励。训练过程结束后，在测试集上评估了训练好的策略，结果如表3.3所示。经济成本包括发电机的发电成本和储能设备的老化成本，住户用电体验损失包括空调控制引起的住户热舒适度损失和电动汽车充电引起的住户不满意度。

表 3.3　测试集上的平均损失

方案	经济成本	住户用电体验损失	总成本
提出的方案	68.03	20.59	88.62
基线方案 1	54.26	122.81	177.07
基线方案 2	57.23	18.36	75.61
基线方案 3	87.12	30.14	117.26
基线方案 4	115.37	5.12	120.49

算法优势：首先聚焦于本章提出的向量 A2C 算法与标准 A2C 算法的比较。需要指出的是，基线方案 1 是用标准 A2C 算法实现的，而其他方案是用提出的向量 A2C 算法实现的。从图 3.3 可以看出，除基线方案 1 外的所有方案都能收敛，这显

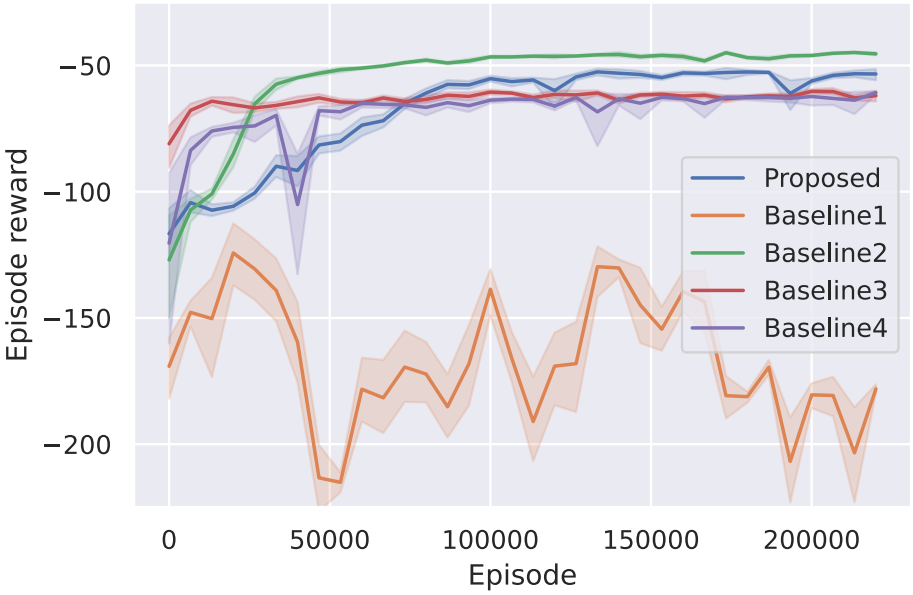


图 3.3　不同方案的训练曲线

示了向量 A2C 算法在训练稳定性方面的优势。

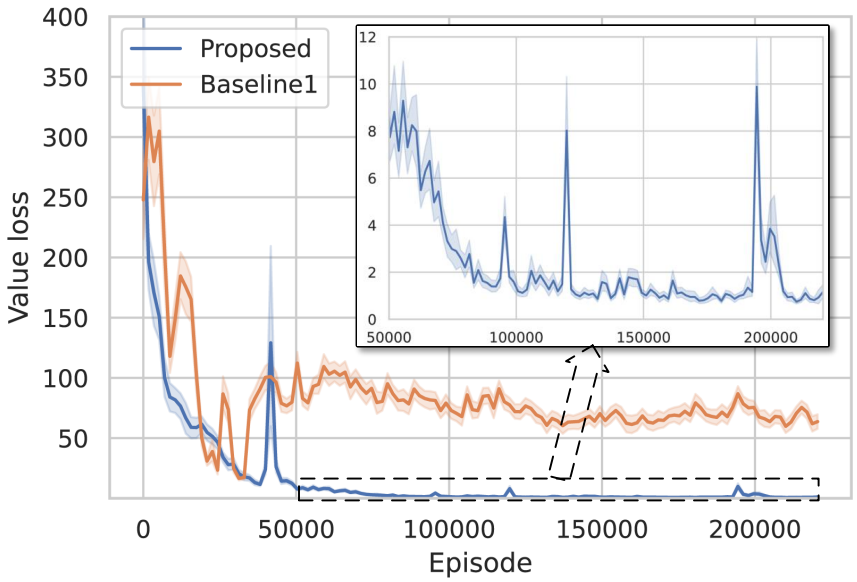


图 3.4　向量 A2C 算法和 A2C 算法的值估计损失曲线

向量 A2C 算法和 A2C 算法的值估计损失如图 3.4 所示。在训练过程中，A2C 算法的平均值估计损失大于 50。极大的值估计偏差，导致图 3.3中基线方案 2 训练过程的不稳定。相比之下，向量 A2C 算法的平均值估计损失在训练初期迅速下降，最终收敛到 2 以下。此外，表3.3显示，本章提出的使用向量 A2C 算法的隐私保护协同控制方案的平均成本与使用 A2C 算法基线方案 1 相比降低了 50%。特别是住户体验损失方面降低了 83%。这些结果都表明，向量 A2C 算法在隐私保护负荷协

同控制问题上的控制效果明显优于标准 A2C 算法。这种算法优越性可以从两个角度来解释。首先，借助于信用分配机制，向量 A2C 算法的价值函数被设计为估计一系列累积奖励而不是总体累积奖励，这减少了估计的方差。其次，Actor 网络的更新方向是最大化与动作分量相关的优势而不是联合动作的优势，这提高了训练的效率和稳定性。

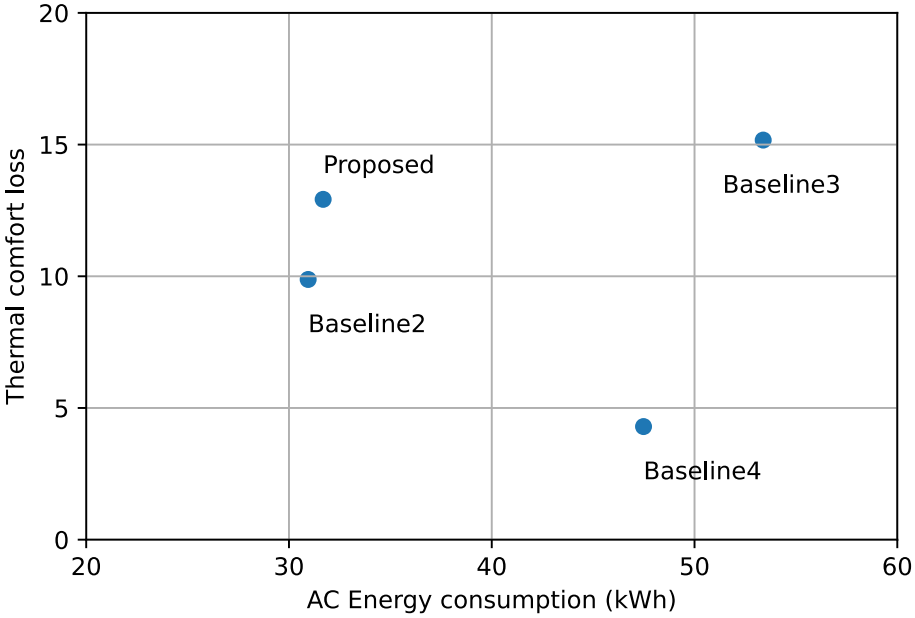


图 3.5　不同方案控制下住户热舒适度损失和空调耗能

隐私保护成本：下面关注本章提出的方案与现有隐私保护方案（基线方案 3）和不考虑隐私保护的方案（基线方案 2）的性能比较。三种方案下训练过程中的控制效果如图 3 所示。收敛后，提出方案的性能优于基线方案 3，略低于基线方案 2，同样的结果也反映在表3.3中。相较于基线方案 2，提出方案的总成本增加了 14.7%，而基线方案 3 的总成本增加了 55.1%。为了更具体分析负荷控制的效果，图3.5 展示了不同方案下空调控制的总体效果：所有空调的耗能和所有住户的热舒适度损失的关系图。可以看出，本章提出的隐私保护方案的控制效果更接近于无隐私保护方案的控制效果；与另一种隐私保护方案——基线方案 3——相比，提出的方案下的空调控制造成的住户热舒适损失更低，但仅仅消耗基线方案 3 下空调耗能的 60%。

为了具体分析空调的温度控制效果，图3.6展示了测试集上某一天的室内温度曲线图。一方面，与基线方案 2 比较，由于微网运营商在决策空调运行功率时缺失室内温度信息，提出的方案存在过度控制的风险：室内温度在空调作用下降低到住户热舒适区的最低温度以下。这会引起用户的热舒适度损失，并且增加空调的耗能。另一方面，与另一种隐私保护方案相比，提出的方案的室内温度控制更稳

定，并且能耗更低，其中一个原因是基线方案 3 的温度设定控制策略只有两种工作模式：以最大功率运行和完全关闭，因此其没有完全利用空调功率的连续动作空间。图3.6中还有一个有趣的现象：提出方案的控制策略在不知道住户进入房间

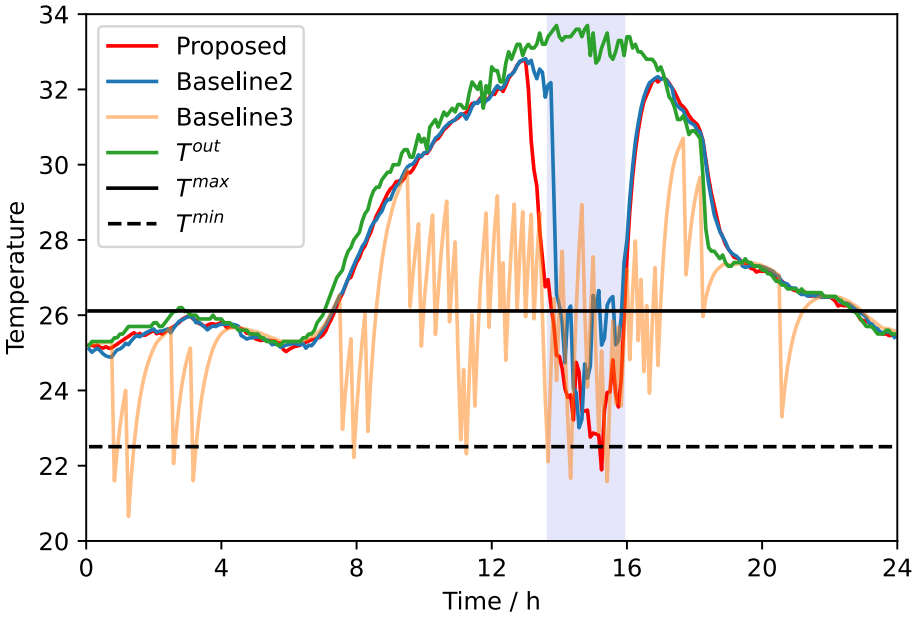


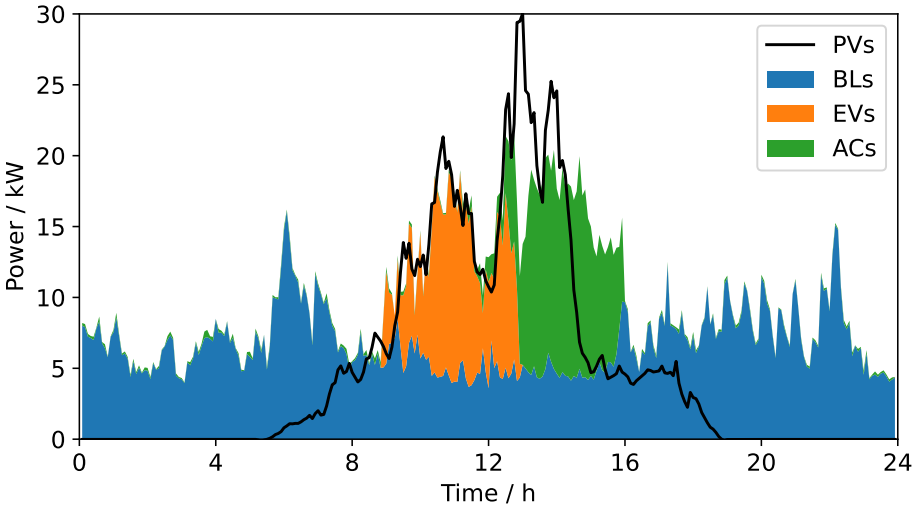
图 3.6　不同方案一天内的室内温度变化曲线

的准确时间的情况下，提前将室内温度降低到零舒适度损失区。这一进步主要得益于 RNN 架构，该架构具有记忆之前信息的能力。总体而言，与基线方案 2 和基线方案 3 相比，所提出的方案以相对较小的成本保护了用户隐私数据。

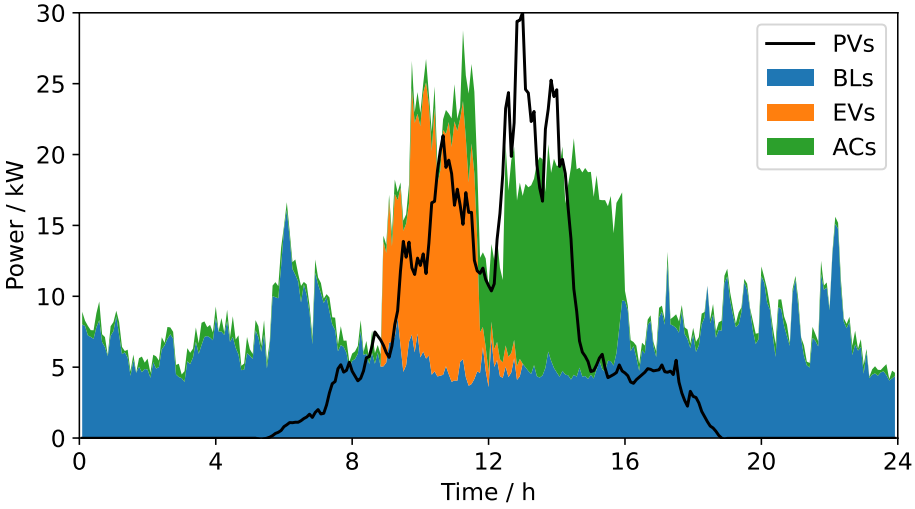
灵活性和可扩展性：所提方案的目标是最小化包括住宅微电网经济成本和住户用电体验损失在内的总成本，但在实践中可以通过调整向量 A2C 算法中的信用分配机制来达到具有不同偏好的控制效果。例如，基线方案 4 旨在通过采用如公式（3.26）所示的特定信用分配机制来最小化住户用电体验损失。表3.3的结果表明，基线方案 4 最大限度地降低了住户用电体验损失，甚至比不考虑隐私保护的方案（基线方案 2）低 72%。但付出的代价是，基线方案 2 的经济成本远高于其他所有方案。

为了进一步比较所提方案和基线方案 4 的控制效果，图3.7展示了一天内所有电动汽车的总充电功率和所有空调的总工作功率。可以看出，所提方案调度下的电动汽车充电过程更匹配光伏的输出功率，从而节省发电机的发电成本和储能设备的老化成本。相比之下，基线方案 4 调度下的电动汽车旨在尽快完成住户的充电任务以避免引起住户对充电过程的不满。基线方案 4 对空调的控制也反映了最小化住户用电体验损失的目标。基线方案 4 控制下的空调造成的住户热舒适度损失仅为所提方案的 29%；同时，基线方案 4 的空调耗能增加了 50% 。因此，基线

方案 4 达到了预期目标：尽可能保证住户的用电体验。



(a) 所提方案控制下的负荷调度曲线



(b) 基线方案 4 控制下的负荷调度曲线

图 3.7　负荷调度曲线

为了更深入地研究所提方案和基线方案 4 下的电动汽车充电调度策略，图3.8显示了一天内两辆电动汽车的剩余待充电量。可以看出，在基线方案 4 下，电动汽车 3 和电动汽车 4 都以几乎最大功率充电，直到充电任务完成。这种充电调度策略旨在充分满足住户的充电需求，而不考虑微电网的功率平衡和经济成本。相比之下，在所提方案调度下，两辆电动汽车以适度和可变的功率充电，从而充分利用光伏发电，降低运营成本。但这种充电调度策略有引起住户不满的风险。例如，在所提方案下，电动汽车 4 在住户离开住宅时充电任务尚未完成。基线方案 4 和所提方案的控制效果表明，通过设计适当的信用分配机制可以实现不同的控制效果。

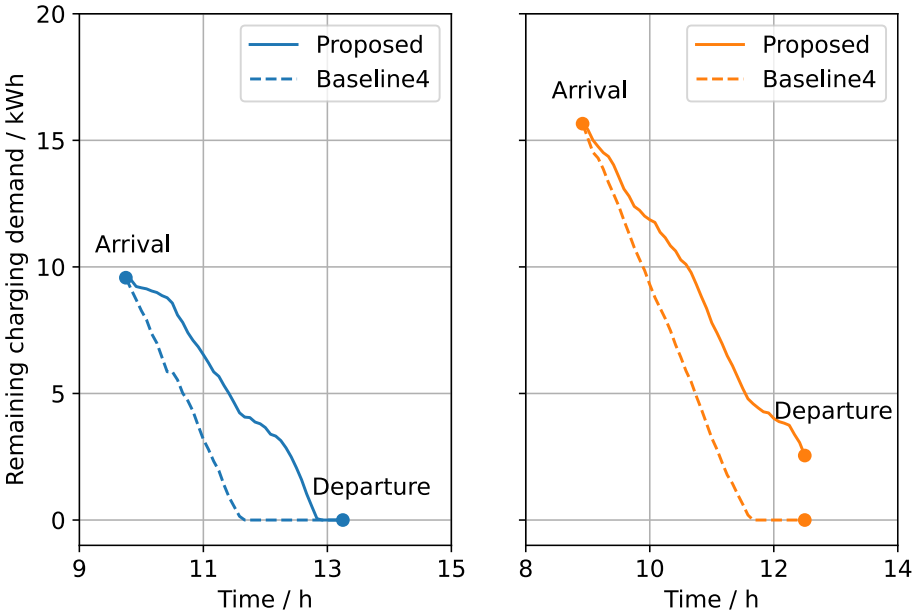


图 3.8　所提方案和基线方案 4 控制下的电动汽车剩余待充电量曲线

3.5　本章小结

本章针对孤岛微电网住宅负荷协同控制问题设计了由微网运营商集中管理的隐私保护方案。在缺失住户关键隐私数据和动作空间维度过高等挑战下，通过设计向量 A2C 算法和融入循环神经网络等方法实现了住宅负荷的有效隐私保护控制。仿真实验说明了本章提出的隐私保护方案相较于其他隐私保护方案达到了更好的控制效果，并且该方案可以灵活扩展以实现不同的控制目的。

第 4 章　云边环境下的分布式隐私保护负荷控制方案

4.1　本章引言

上一章提出的微网运营商集中管理的隐私保护负荷控制方案能够有效保护住户的部分关键隐私数据，但住户的隐私担忧并未完全消除。例如，尽管微网运营商无法提前获取电动汽车的到达时间和离开时间，但电动汽车到达或离开后会立刻告知微网运营商以方便其调度充电过程。此外，上一章将住户的用电需求作为目标函数，这事实上造成了住户用电需求得不到满足的可能，仿真结果也验证了这种可能性。在以用户为中心的场景中，牺牲用户体验换取经济利益是不可接受的。即便住户默许这一行为，在实际部署该方案时，住户很可能会恶意夸大自己的舒适度损失函数使得自身的用电需求被优先考虑，这最终会导致负荷协同控制方案失效。

完全合作的多智能体强化学习方法为解决上述一系列问题带来了希望。在多智能体强化学习设定中，每个智能体根据自身的观测决定自己的动作。受多智能体强化学习概念的启发，本章为孤岛微电网中隐私保护的住宅负荷协同控制设计了新的场景。第一，场景中每个住户的可控负荷由其家庭能量管理系统（HEMS）控制，每个 HEMS 仅仅根据所在住宅的本地信息和微电网的公共信息决定可控负荷的控制信号。第二，该场景下住户的用电需求被设定为必须满足的硬性约束条件，而非出现在目标函数中。

场景上的两点变化引起了新的挑战。首先，目前完全合作的多智能体强化学习方法采用集中式训练、分散式执行的范式。该范式在执行时无需智能体分享各自的观测信息，但在训练时需要收集所有智能体的观测和动作。因此在使用多智能体强化学习解决新场景中负荷协同控制问题时，需要提出新的多智能体强化学习框架，从而在训练阶段加强对住户隐私数据的保护。其次，将住户用电需求由目标函数变为约束条件，将导致全局奖励变得完全不可分，基于差分奖励的显式信用分配机制失效。因此，需要设计隐式的信用分配机制，引导各个 HEMS 学习推断各自对全局奖励的贡献。

为了给多智能体强化学习的部署提供物理支撑，本章引入了云边环境。各个 HEMS 对住宅负荷的控制在边缘侧执行，各个 HEMS 的部分协同训练过程在云侧完成。随着住宅数量的增加，多智能体强化学习在云边环境中的大规模部署给云边通信带来了极大压力。因此新的多智能体强化学习框架也需要降低云边环境的通信成本。

基于上述考量，本章打算在通信受限的云边环境中最小化住宅微电网的运营成本，同时有效地保护住宅的本地信息。该场景中的住宅负荷协同控制问题被表述为有限步长的分散式部分可观测马尔可夫决策过程（Dec-POMDP）。我们提出了分散式执行者-分布式评价者（DADC）框架，一种新的隐私保护多智能体强化学习框架。在这个框架中，每个边缘侧的 HEMS 具有单独的 Actor 和 Critic，两者都只以住宅的本地观测和微电网公共信息为输入。本地 Critic 输出标量的本地值函数并上传至云端。全局值函数通过一个前馈网络来估计，该网络将所有本地值函数的串联作为输入。各个 HEMS 的 Critic 网络和前馈网络可以通过反向传播全局时序差分误差获得梯度并进行更新。

本章的贡献可以总结如下：

- 本章为隐私保护的住宅负荷协同控制设计了云边环境场景。该场景中，边缘侧的 HEMS 控制对应住宅的电力负荷，并且通过云边通信实现所有住宅的协同负荷控制。
- 本章提出了一种新的多智能体强化学习框架 DADC，用于解决通信受限的云边环境中住宅微电网的协同负荷控制问题。DADC 框架中的每个 HEMS 在训练阶段的单个时间步仅与云端共享一个经过本地 Critic 编码的标量值，有效地保护住户的数据隐私。
- DADC 框架中全局值函数的计算只需要来自住宅的标量本地值函数，显著降低了云边通信量；而 DADC 框架中的前馈网络计算量与住宅数量成线性复杂度，这两者都有利于 DADC 框架在云边环境中的大规模部署。
- 基于真实数据的仿真实验说明了 DADC 框架明显优于 IAC 框架（隐私保护基线框架），并且与没有隐私保护设计的通用多智能体强化学习框架 DACC 相比，DADC 框架可以在有效保护住户数据隐私的同时达到与其相当的控制效果。

4.2　场景设计

4.2.1　场景描述

考虑一个用于家庭电力负荷协同控制的云边环境。如图4.1所示，假设边缘侧有一个孤岛微网，包含 n 个住宅和可控发电机。住宅和发电机可以与云端双向通信，从而实现负荷协同控制和功率平衡。不失一般性，假设每个住宅配备基础负荷、一个空调和一个电动汽车。此外，每个住宅有一个家庭能量管理系统（HEMS）来直接调度包括空调和电动汽车在内的所有可调节电力设备。为了达到隐私保护的目的，每个 HEMS 只能获取到来自发电机的公共信息和自己住宅的本地信息。

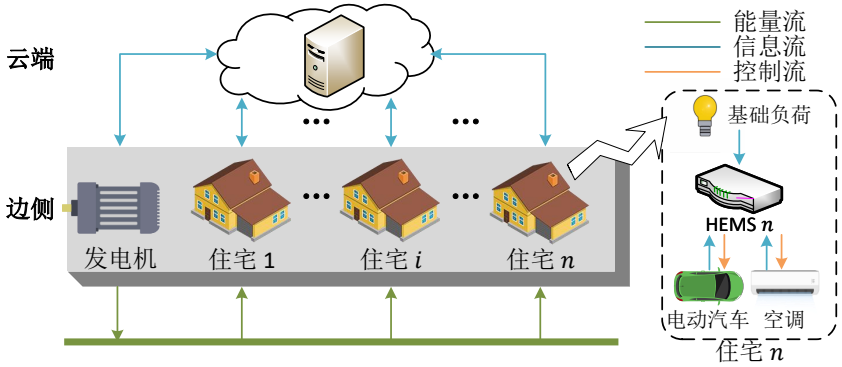


图 4.1　用于家庭电力负荷协同控制的云边环境

上一章将室内温度控制和电动汽车充电调度任务设置为损失函数，即用户允许室内温度超出用户的舒适区，电动汽车电量在用户离开时可以达不到用户的设定量。而本章将其定义为硬性约束：

$$T_{i,\min}^{\mathrm{AC}} \leqslant T_{i,t}^{\mathrm{AC}} \leqslant T_{i,\max}^{\mathrm{AC}}, \tag{4.1}$$

$$E_{i,t}^{\mathrm{EV}}\big|_{t=d_i^{\mathrm{EV}}} \geqslant E_{i,\mathrm{targ}}^{\mathrm{EV}}, \tag{4.2}$$

此外，本章的场景假设住宅 i 的空调房间始终有人，因此约束4.1需要在全时间段满足。

在每个时刻 t，发电机自动调整来满足微电网的功率平衡条件，

$$P_t^{\mathrm{CG}} = \sum_{i\in\mathcal{D}} \left(P_{i,t}^{\mathrm{BL}} + P_{i,t}^{\mathrm{AC}} + P_{i,t}^{\mathrm{EV}}\right), \tag{4.3}$$

其中 $P_{i,t}^{\mathrm{BL}}$ 表示住宅 i 在时刻 t 的基础负荷功率。

4.2.2　分散式部分可观测马尔可夫过程建模

本小节将云边环境下的电力负荷协同控制问题建模分散式部分可观测马尔可夫过程。分散式部分可观测马尔可夫过程中的多个智能体被指定为微电网中的多个 HEMS。由于无模型的多智能体强化学习不依赖于全局状态和状态转移概率分布，因此我们将重点放在三个部分上，即观测、动作和全局奖励函数。

观测：为了使住宅的协同调度成为可能，发电机功率被视为公共信息提供给每个 HEMS。HEMS 的观测定义为

$$o_t^i = \left[t, P_t^{\mathrm{CG}}, P_{i,t}^{\mathrm{BL}}, T_{i,t}^{\mathrm{out}}, T_{i,t}^{\mathrm{AC}}, E_{i,t}^{\mathrm{EV}}, E_{i,\mathrm{targ}}^{\mathrm{EV}}, d_i^{\mathrm{EV}}\right], \tag{4.4}$$

式（4.4）中的后 6 个分量是住宅 i 的本地信息，需要得到保护。

动作：为了便于多智能体强化学习的训练，所有连续动作空间统一为 $[0, 1]$。为此，引入空调和电动汽车的控制信号。为了满足室内温度约束、保证住户的热舒适

度，定义如下空调控制机制，

$$P_{i,t}^{\mathrm{AC}} = P_{i,\max}^{\mathrm{AC}} \begin{cases} 1, & \text{if } T_{i,t}^{\mathrm{AC}} \geqslant T_{i,\max}^{\mathrm{AC}}, \\ 0, & \text{if } T_{i,t}^{\mathrm{AC}} \leqslant T_{i,\min}^{\mathrm{AC}}, \\ u_{i,t}^{\mathrm{AC}}, & \text{otherwise.} \end{cases} \tag{4.5}$$

该机制下，空调采取相应的模式来改善热条件：如果当前室内温度高于用户最高的理想温度，则空调以最大功率运行；如果当前室内温度低于用户最低的理想温度，则空调关闭。

为了保证住户的电动汽车充电任务在用户离开之前完成，需要在所有时刻 $a_i^{\mathrm{EV}} \leqslant t < d_i^{\mathrm{EV}}$ 检查如下不等式，

$$E_{i,t}^{\mathrm{EV}} + \eta_i^{\mathrm{EV}} P_{i,\max}^{EV} \left(d_i^{\mathrm{EV}} - t\right) \Delta \mathrm{t} \geqslant E_{i,\mathrm{targ}}^{\mathrm{EV}}, \tag{4.6}$$

其中左式表示电动汽车在离开时的最大理论电量（剩余充电时间内始终保持最大充电功率）。一旦不等式（4.6）不满足，电动汽车必须以最大功率充电。因此，电动汽车在可充电时间的充电机制可以表示为

$$P_{i,t}^{\mathrm{EV}} = P_{i,\max}^{\mathrm{EV}} \begin{cases} 1, & \text{if (4.6) not satisfied,} \\ 0, & \text{else if } E_{i,t}^{\mathrm{EV}} \geqslant E_{i,\max}^{\mathrm{EV}}, \\ u_{i,t}^{\mathrm{EV}}, & \text{otherwise.} \end{cases} \tag{4.7}$$

（4.7）中的第二个条件是为了保障住宅 i 的电动汽车电量满足约束（2.27），即电池电量不超过最大容许电量。

通过设计空调控制机制（4.5）和电动汽车充电机制（4.7），室内温度和电动汽车充电任务的硬性约束（4.1）和（4.2）得以满足。HEMS i 在时刻 t 的动作可以表示为

$$a_t^i = \left[u_{i,t}^{\mathrm{AC}}, u_{i,t}^{\mathrm{EV}}\right] \in [0,1]^2 \, .i \in \mathcal{D}. \tag{4.8}$$

那么所有 HEMS 在时刻 t 的联合动作为

$$a_t = \left[a_t^1, \cdots, a_t^n\right] \in [0,1]^{2n} \, . \tag{4.9}$$

奖励函数：与上一章微网运营商直接管理的场景相比，本章的场景中没有储能设备，这导致发电机将独自承担维持孤岛微网功率平衡的重任。因此本章假设发电机功率在每个时刻任意指定，而不是服从转移函数2.29。此外，本章场景中住户用电需求为硬性约束，而不是损失函数一部分。因此微网的成本只与发电机有关，包括发电机的发电成本和调整成本。前者由发电机的输出功率决定，后者取决于发电机发电功率的波动，因为频繁的功率调整会损耗发电机的使用寿命。因此

时刻 t 的全局奖励可以表示为

$$r_t = -\left(g_1^{\mathrm{CG}}\left(P_t^{\mathrm{CG}}\right) + g_2^{\mathrm{CG}}\left(P_t^{\mathrm{CG}} - P_{t-1}^{\mathrm{CG}}\right)\right), \tag{4.10}$$

其中 $g_1^{\mathrm{CG}}(\cdot)$ 是发电机当前输出功率的发电成本函数，$g_2^{\mathrm{CG}}(\cdot)$ 是发电机当前发电功率与下一时刻发电功率之差的调节成本函数。值得一提的是，函数 $g_1^{\mathrm{CG}}(\cdot)$ 和 $g_2^{\mathrm{CG}}(\cdot)$ 根据实际情况设置，可以是非线性的，因此无法把全局奖励直接分配给每个 HEMS，上一章的信用分配机制失效。

4.3　算法设计

如前文所述，目前存在的 AC 框架不能既实现有效的负荷协同控制，又严格保护住宅用户的数据隐私。本节提出了分散式 Actor-分布式 Critic（DADC）框架，该隐私保护框架允许所有 HEMS 严格保存本地观测且进行高效的联合强化学习训练。本节还详述了使用在线策略算法和离线策略算法的 DADC 框架分布式训练。

4.3.1　网络结构

DADC 框架具有分散式 Actor 和分布式 Critic 的结构。该框架中的 Critic 可以用于估计值函数和动作值函数。为了便于演示，图4.2以估计值函数的 Critic 为例展示 DADC 框架的结构。

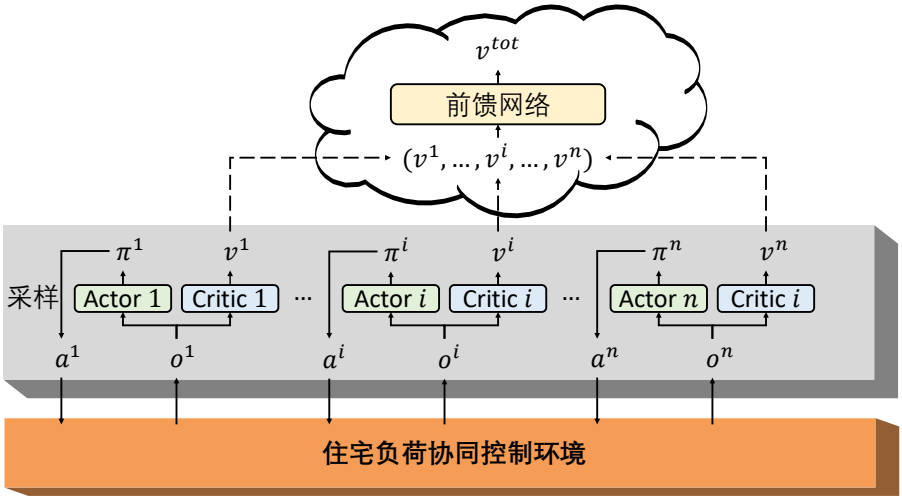


图 4.2　云边环境下的多智能体强化学习网络

在 DADC 框架中，每个 HEMS i 在边缘侧都有一个 Actor 和一个 Critic。Actor 和 Critic 分别以本地观测 o^i 为输入，输出本地策略 π^i 和本地值函数 v^i。HEMS i 的动作根据随机策略 π^i 采样产生，而本地值函数被上传至云端。在云端，可学习的前馈网络将所有 HEMS 估计的本地值函数组成的 n 维向量映射为全局值函数。

分散式 Actor: 每个 HEMS 学习以 θ^i 为参数的随机策略 $\pi^i : \mathcal{O}_i \times \mathcal{A}_i \mapsto [0, +\infty)$。该策略将每个 HEMS 的本地观测映射为其动作空间上的一个概率分布。注意策略 π^i 只取决于本地观测 o^i。整个微电网的联合策略由所有 HMES 的本地策略构成:

$$\pi(a|o) := \prod_{i=1}^{n} \pi^i(a^i|o^i;\theta^i), \tag{4.11}$$

其中 $a = (a^1, ..., a^n)$，$o = (o^1, ..., o^n)$。

分布式 Critic: 大多数合作多智能体 AC 算法采用的 DACC 框架有一个集中式的 Critic 网络来估计联合策略的全局值函数。这意味着虽然每个 HEMS 可以独立地执行自己的策略，但在训练时需要将包含所有 HEMS 观测的全局状态提供给中心化的 Critic。为了保护数据隐私，本章提出的 DADC 框架仅仅使用本地值函数来估计全局值函数，

$$V^{\boldsymbol{\pi}}(o) \approx V^{\text{tot}}\left(V^1(o^1;\phi^1), ..., V^n(o^n;\phi^n);\varphi\right), \tag{4.12}$$

其中 $V^i(\cdot;\phi^i) : \mathcal{O}_i \to \mathbb{R}$ 是在边缘侧的以 ϕ^i 为参数的本地 Critic, $V^{\text{tot}}(\cdot;\varphi) : \mathbb{R}^n \to \mathbb{R}$ 是在云端的以 φ 为参数的前馈神经网络。

通过这种设计，云端全局值函数的计算只需要收集本地值函数，这些本地值函数是由边缘侧中 HEMS 的 Critic 编码的标量值。这种设计有三个优势。首先，标量的值函数极大降低了原始信息被分析或窃取的可能，从而严格保留了 HEMS 的本地观测。其次，云端和边缘侧之间的通信负担显著降低。比如将 DADC 框架与 MAAC 算法比较时，MAAC 算法中智能体的本地嵌入函数将维度为 d 的向量上传至中心的注意力机制网络，因此使用 MAAC 算法的云边环境的总通信复杂度为 $O(nd)$，而使用 DADC 框架的云边环境的总通信复杂度为 $O(n)$。第三，前馈网络减少了云端的计算负担。如果前馈网络的隐藏层具有固定数量的单元，则计算复杂度为 $O(n)$，而 MAAC 算法受累于自注意力网络，其计算复杂度为 $O(n^2d)$。因此，提出的 DADC 框架有助于在云边环境中进行大规模部署。

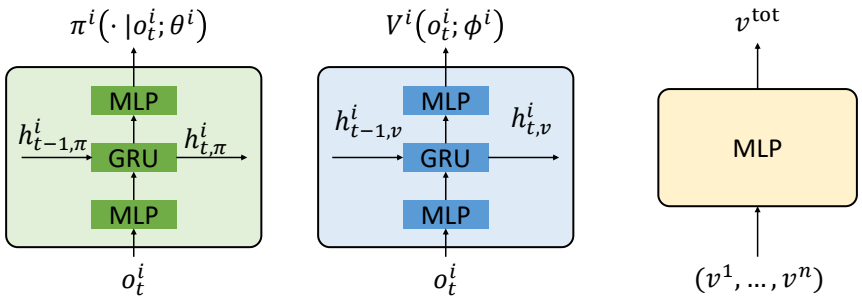
4.3.2　内部结构

本地 Actor、本地 Critic 和前馈网络的内部结构如图4.3所示。每个 HEMS 的策略网络由一个 GRU 和两个 MLP 组成。GRU 的隐状态 $h^i_{t-1,\pi}$ 编码了时刻 t 之前的信息，这可以缓解 HEMS 的部分可观测问题。在时刻 t，Actor i 以 HEMS i 的本地观测 o^i_t 和上一时刻的隐状态 $h^i_{t-1,\pi}$ 为输入，输出本地动作空间上的一个概率分布 $\pi^i(\cdot|o^i_t;\theta^i)$ 和当前时刻的隐状态 $h^i_{t,\pi}$。

本地 Critic 的内部结构类似于本地 Actor，也是由 GRU 和 MLP 组成。在时刻 t，Critic i 以 HEMS i 的本地观测 o^i_t 和上一时刻的隐状态 $h^i_{t-1,v}$ 为输入，输出本地

值函数 $V^i(o_t^i;\phi^i)$ 和当前时刻的隐状态 $h_{t,v}^i$。

云端的前馈网络完全由 MLP 构成。其以边缘侧的 HEMS 上传的本地值函数为输入，输出全局值函数。



(a) Actor i　　(b) Critic i　　(c) 前馈网络

图 4.3　本地 Actor、本地 Critic 和前馈网络的内部结构

4.3.3　分布式训练：在线策略优化

目前存在的绝大多数 AC 强化学习算法均适用于 DADC 框架的分布式训练。本小节以 PPO 算法[24]为例说明如何用在线策略强化学习算法进行 DADC 框架的训练。

在 DADC 框架中，全局值函数根据（4.12）估计。因此全局 Critic 的损失为

$$\mathcal{L}_c=\hat{\mathbb{E}}_t\left[\left(V^{\text{tot}}(v_t^1,\ldots,v_t^n;\varphi)-V_{\text{old}}^{\text{tot}}(v_{t+1,\text{old}}^1,\ldots,v_{t+1,\text{old}}^n;\varphi_{\text{old}})-\hat{A}_t\right)^2\right]. \tag{4.13}$$

根据梯度传播的链式法则，云端前馈网络的梯度为

$$\Delta\varphi=\hat{\mathbb{E}}_t\left[\nabla_\varphi V^{\text{tot}}(v_t^1,\ldots,v_t^n;\varphi)\left(V^{\text{tot}}-V_{\text{old}}^{\text{tot}}-\hat{A}_t\right)\right], \tag{4.14}$$

而边缘侧 Critic i 的梯度为

$$\Delta\phi^i=\hat{\mathbb{E}}_t\left[\nabla_{\phi^i}V^i(o_t^i;\phi^i)\nabla_{v^i}V^{\text{tot}}(v_t^1,\ldots,v_t^n;\varphi)\left(V^{\text{tot}}-V_{\text{old}}^{\text{tot}}-\hat{A}_t\right)\right]. \tag{4.15}$$

注意梯度 $\nabla_{v^i}V^{\text{tot}}(v_t^1,\ldots,v_t^n;\varphi)\left(V^{\text{tot}}-V_{\text{old}}^{\text{tot}}-\hat{A}_t\right)$ 由云端下发给 HEMS i。通过这种方式，每个本地 Critic 通过反向传播全局时序差分误差来学习。因此，$V^i(\cdot;\phi^i)$ 是隐式学习的，而不是通过单独分配给 HEMS i 奖励来学习。

本地 Actor 的损失为

$$\mathcal{L}_a^i=\hat{\mathbb{E}}_t\left[\min\left(w_t^i(\theta^i)\hat{A}_t,\text{clip}\left(w_t^i(\theta^i),1-\epsilon,1+\epsilon\right)\hat{A}_t\right)\right], \tag{4.16}$$

其中

$$w_t^i(\theta^i)=\frac{\pi^i(a_t^i|o_t^i;\theta^i)}{\pi^i(a_t^i|o_t^i;\theta_{\text{old}}^i)}. \tag{4.17}$$

注意该损失是完全由相应的 HEMS 在边缘侧本地计算。因此相应的梯度为

$$\Delta\theta^i = \hat{\mathbb{E}}_t \left[\nabla_{\theta^i} \pi^i(a_t^i | o_t^i; \theta^i) \nabla_{w_t^i} \mathcal{L}_a^i \right]. \tag{4.18}$$

云边环境下分布式训练的详细过程如算法4.1所示。总体上，一次迭代的训练可以分为三个部分：与环境交互、计算全局优势函数和执行参数更新。灰色区域表示在边缘侧中的操作，而黄色区域表示云端中的操作。

如算法4.1的第 4-10 行所示，第一部分由 HEMS 独立操作。在每个时刻 t，每个 HEMS i 都通过独立选择和执行自己的动作来与环境进行交互。HEMS 上传到云端的唯一信息是每个 HEMS 计算的本地值函数。

第二部分由云端执行。如第 14 行所示，全局值函数的计算只依赖于 HEMS 上传的本地值函数，而不是包括所有 HEMS 的观测在内的全局状态。全局优势函数的计算过程是反向的，即计算时刻 t 的优势函数需要使用时刻 $t+1$ 的优势函数。

第三部分由边缘侧和云端交替执行。网络的参数更新 K 次，在每次更新过程中，边缘侧的 HEMS 首先根据新的参数计算各自的值函数，并将其上传到云端。云端使用前馈网络来计算全局 Critic 的损失，该损失对前馈网络参数的梯度用于更新前馈网络，而对本地值函数的梯度分发给相应的 HEMS。HEMS 使用接收到的梯度和全局优势函数更新本地 Actor 和 Critic。

4.3.4　分布式训练：离线策略优化

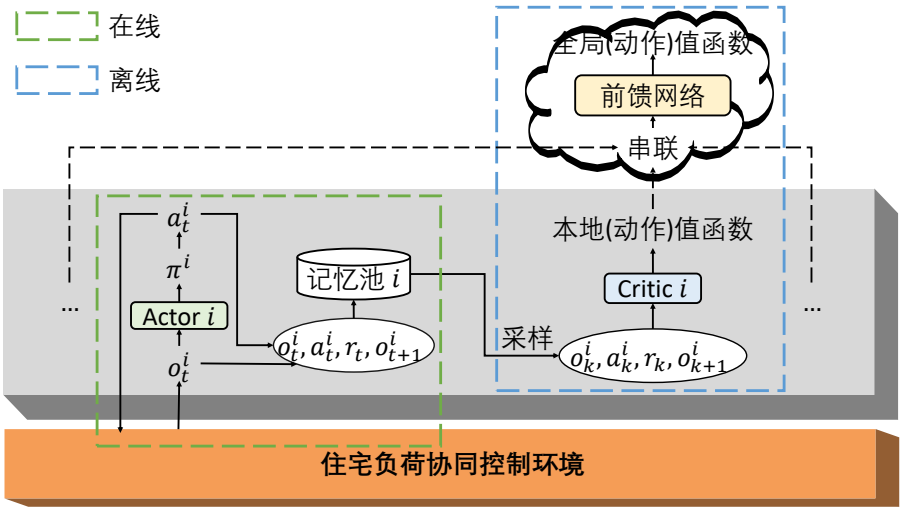


图 4.4　云边环境下 DADC 框架的离线策略训练示意图

DADC 框架也支持离线策略训练。离线策略强化学习算法一般需要用到经验记忆池。为了保护用户的数据隐私，联合经验 (o_t, a_t, r_t, o_{t+1}) 以分布式的方式储存在边缘侧，而不是储存在集中式记忆池。具体来说，每个 HEMS 的记忆池用于储存其本地经验 $(o_t^i, a_t^i, r_t, o_{t+1}^i)$。为了从分布式的记忆池中取出联合经验，所有 HEMS

算法 4.1 使用 PPO 算法的 DADC 框架分布式训练

为每个 HEMS 初始化 θ^i 和 ϕ^i；为前馈网络初始化 φ
for $episode = 1$ to $episode_{\max}$ **do**
  % 与环境交互
  for $t = 1$ to $T-1$ **do**
    for all HEMS i **do**
      生成策略 $\pi^i(\cdot|o_t^i;\theta^i)$
      选择动作 $a_t^i \sim \pi^i(\cdot|o_t^i;\theta^i)$
      执行动作 a_t^i 并收到下一时刻的观测 o_{t+1}^i.
      $p_{t,\text{old}}^i \leftarrow \pi^i(a_t^i|o_t^i;\theta^i)$, $v_{t,\text{old}}^i \leftarrow V^i(o_t^i;\phi^i)$
      上传 $v_{t,\text{old}}^i$ 至云端
    end for
  end for
  % 计算全局优势函数
  $\hat{A}_T \leftarrow 0, v_T^{\text{tot}} \leftarrow 0$
  for $t = T-1$ to 1 **do**
    $v_{t,\text{old}}^{\text{tot}} \leftarrow V^{\text{tot}}(v_{t,\text{old}}^1, ..., v_{t,\text{old}}^n;\varphi)$
    $\hat{A}_t \leftarrow \lambda\hat{A}_{t+1,\text{old}} + r_t + v_{t+1}^{\text{tot}} - v_{t,\text{old}}^{\text{tot}}$
  end for
  发送 $\{\hat{A}_t\}_{t=1}^{T-1}$ 到每个 HEMS
  % 参数更新
  for $k = 1$ to K **do**
    % 边缘侧
    for all HEMS i **do**
      $\{v_t^i\}_{t=1}^{T-1} \leftarrow \{V^i(o_t^i;\phi^i)\}_{t=1}^{T-1}$
      上传 $\{v_t^i\}_{t=1}^{T-1}$
    end for
    $\mathcal{L}_c \leftarrow \sum_{t=1}^{T-1}\left(V^{\text{tot}}(v_t^1, ..., v_t^n;\varphi) - v_{t,\text{old}}^{\text{tot}} - \hat{A}_t\right)^2$
    用梯度 $\partial\mathcal{L}_c/\partial V^{\text{tot}} \cdot \partial V^{\text{tot}}/\partial\varphi$ 更新 φ
    发送 $\{\partial\mathcal{L}_c/\partial v_t^i\}_{t=1}^{T-1}$ 至 HEMS i
    % 边缘侧
    for all HEMS i **do**
      $\Delta\phi^i \leftarrow \sum_{t=1}^{T-1}\partial\mathcal{L}_c/\partial v_t^i \cdot \partial v_t^i/\partial\phi^i$
      用梯度 $\Delta\phi^i$ 更新 ϕ^i
      for $t = 1$ to $T-1$ **do**
        $w_t^i \leftarrow \pi^i(a_t^i|o_t^i;\theta^i)/p_{t,\text{old}}^i$
      end for
      $\mathcal{L}_a^i = \sum_{t=1}^{T-1}\min(w_t^i\hat{A}_t, \text{clip}(w_t^i, 1-\epsilon, 1+\epsilon)\hat{A}_t)$
      $\Delta\theta^i \leftarrow \sum_{t=1}^{T-1}\partial\mathcal{L}_a^i/\partial w_t^i \cdot \partial w_t^i/\partial\theta^i$
      用梯度 $\Delta\theta^i$ 更新 θ^i
    end for

  end for
end for

需要根据相同的采样索引取出自己的本地经验。记 $\mathcal{B}^i$ 为 HEMS i 的记忆池，则所有分布式记忆池组成的联合记忆池为 $\mathcal{B}^{\text{tot}}$。

为了展示离线情况下 DADC 框架的分布式训练，本小节使用一种典型的离线策略强化学习算法, SAC 算法[25]。要适配 DADC 框架, SAC 算法应该包括全局值函数 V^{tot}，全局 soft-Q 函数 Q^{tot} 和联合策略 π。全局值函数和联合策略已经在上一小节定义过，只需将原算法中的 soft-Q 函数 Q 扩展到 DADC 框架下的全局 soft-Q 函数 Q^{tot} 即可。类似于全局值函数 V^{tot} 的定义，全局 soft-Q 函数 $Q^{\text{tot}}(\cdot;\omega):\mathbb{R}^n\to\mathbb{R}$ 是以 ω 为参数的前馈神经网络，其在云端将 HEMS 上传的本地 soft-Q 函数组成的向量映射为全局 soft-Q 函数。处于边缘侧的本地 soft-Q 网络 $Q^i(\cdot;\psi^i):\mathcal{O}_i\times\mathcal{A}_i\to\mathbb{R}$ 以 ψ^i 为参数，将本地观测和动作映射为本地 soft-Q 函数。

相应地，(2.21)-(2.23) 中的损失函数修改为

$$\mathcal{L}_Q=\mathbb{E}_{(o_k,a_k,r_k,o_{k+1})\sim\mathcal{B}^{\text{tot}}}\left[\left(Q^{\text{tot}}(q_k^1,...,q_k^n;\omega)-r_k-V^{\text{tot}}(v_{k+1}^1,...,v_{k+1}^n;\varphi)\right)^2\right],\tag{4.19}$$

$$\mathcal{L}_V=\mathbb{E}_{o_k\sim\mathcal{B}^{\text{tot}}}\left[\left(V^{\text{tot}}(v_k^1,...,v_k^n;\varphi)-\mathbb{E}_{a^i\sim\pi^i,i\in\mathcal{D}}\left[Q^{\text{tot}}(q_k^1,...,q_k^n;\omega)-\sum_{i\in\mathcal{D}}\log p_k^i\right]\right)^2\right],\tag{4.20}$$

$$\mathcal{L}_\pi^i=\mathbb{E}_{o_k^i\sim\mathcal{B}^i}\left[\mathbb{E}_{a\sim\pi^i(\cdot|o_k^i;\theta^i)}\left[\log\pi^i(a|o_k^i;\theta^i)-Q^i(o_k^i,a;\psi^i)\right]\right],\ \forall i\in\mathcal{D}.\tag{4.21}$$

其中，

$$p_k^i=\pi^i(a^i|o_k^i;\theta^i),\ q_k^i=Q^i(o_k^i,a_k^i;\psi^i),\ v_k^i=V^i(o_k^i;\phi),\ v_{k+1}^i=V^i(o_{k+1}^i;\phi),\ \forall i\in\mathcal{D}.\tag{4.22}$$

使用 SAC 算法的 DADC 框架离线策略训练详细过程如算法4.2所示。训练过程总体上分为两部分，在线环境交互和离线参数更新。下面着重介绍离线策略训练和在线策略训练的不同之处。

第一部分如算法4.2的第 3-9 行所示。在每个时刻 t，HEMS i 在与环境交互之后，将经验 $(o_t^i,a_t^i,r_t,o_{t+1}^i)$ 存储于本地记忆池 $\mathcal{B}_i$ 中。

第二部分如算法4.2的第 12-38 行所示。首先云端随机选择样本索引并下发至边缘侧的每个 HEMS。HEMS 根据收到的索引从自己的记忆池中采样出对应的本地经验。随后，HEMS 计算在给定观测下以当前策略采样出给定动作的概率，并计算本地值函数和 soft Q 函数。这些中间计算结果被上传至云端。云端接收并利用这些计算结果计算全局值函数网络和全局 soft Q 网络的损失。损失对云端前馈网络参数的梯度 $\Delta\varphi$ 和 $\Delta\omega$ 直接用于更新前馈网络；对各个本地值函数和本地 soft Q 网络的梯度下发至相应的 HEMS。边缘侧的 HEMS 根据接收的中间梯度计算全局损失对本地 Critic 参数的梯度，并更新本地 ciritic。同时，HEMS 利用本地 soft Q

算法 4.2 使用 SAC 算法的 DADC 框架分布式训练

边缘侧的每个 HEMS 初始化 θ^i, ϕ^i 和 ψ^i；云端为前馈网络初始化 φ 和 ω
for $episode = 1$ to $episode_{\max}$ **do**
  % 在线交互
  for $t = 1$ to T **do**
    for all HEMS i **do**
      生成策略 $\pi^i(\cdot|o_t^i;\theta^i)$
      选择动作 $a_t^i \sim \pi^i(\cdot|o_t^i;\theta^i)$
      执行动作 a_t^i 并收到 o_{t+1}^i, r_t
      将 $\left(o_t^i, a_t^i, r_t, o_{t+1}^i\right)$ 储存于 $\mathcal{B}_i$
    end for
  end for
  % 离线更新
  随机选择 K 个索引
  将索引下发至每个 HEMS
  % 边缘侧分布式计算
  for all HEMS i **do**
    根据索引从 $\mathcal{B}^i$ 采出样本
    $\{p_k^i\}_{k=1}^K \leftarrow \{\pi^i(a_k^i|o_k^i;\theta^i)\}_{k=1}^K$
    $\{v_k^i\}_{k=1}^K \leftarrow \{V^i(o_k^i;\phi^i)\}_{k=1}^K$
    $\{q_k^i\}_{k=1}^K \leftarrow \{Q^i(o_k^i,a_k^i;\psi^i)\}_{k=1}^K$
    上传 $\{p_k^i\}_{k=1}^K, \{v_k^i\}_{k=1}^K, \{q_k^i\}_{k=1}^K$ 至云端
  end for
  % 云端参数更新
  $\left\{v_k^{\text{tot}}\right\}_{k=1}^K \leftarrow \left\{V^{\text{tot}}(v_k^1, ..., v_k^n;\varphi)\right\}_{k=1}^K$
  $\left\{q_k^{\text{tot}}\right\}_{k=1}^K \leftarrow \left\{Q^{\text{tot}}(q_k^1, ..., q_k^n;\omega)\right\}_{k=1}^K$
  $\mathcal{L}_V \leftarrow \sum_{k=1}^K \left(v_k^{\text{tot}} - q_k^{\text{tot}} + \sum_{i=1}^n \log p_k^i\right)^2$
  $\mathcal{L}_Q \leftarrow \sum_{k=1}^K \left(q_k^{\text{tot}} - v_k^{\text{tot}} - r_k\right)^2$
  用梯度 $\partial\mathcal{L}_V/\partial V^{\text{tot}} \cdot \partial V^{\text{tot}}/\partial\varphi$ 更新 φ
  用梯度 $\partial\mathcal{L}_Q/\partial Q^{\text{tot}} \cdot \partial Q^{\text{tot}}/\partial\omega$ 更新 ω
  下发 $\{\partial\mathcal{L}_Q/\partial q_k^i, \partial\mathcal{L}_V/\partial v_k^i\}_{k=1}^K$ 至 HEMS i
  % 边缘侧参数更新
  for all HEMS i **do**
    $\Delta\phi^i \leftarrow \sum_{k=1}^K \partial\mathcal{L}_V/\partial v_k^i \cdot \partial v_k^i/\partial\phi^i$
    $\Delta\psi^i \leftarrow \sum_{k=1}^K \partial\mathcal{L}_Q/\partial q_k^i \cdot \partial q_k^i/\partial\psi^i$
    用梯度 $\Delta\phi^i$ 和 $\Delta\psi^i$ 更新 ϕ^i 和 ψ^i
    $\mathcal{L}_\pi^i \leftarrow \sum_{k=1}^K \mathbb{E}_{a\sim\log\pi^i(o_k^i,\cdot;\theta^i)}$
      $\left[\log\pi^i(o_k^i,a;\theta^i) - Q^i(o_k^i,a;\psi^i)\right]$
    用梯度 $\partial\mathcal{L}_\pi^i/\partial\theta^i$ 更新 θ^i
  end for

end for

函数计算策略网络的梯度并更新策略网络。策略网络的更新不依赖于来自云端的中间计算结果。

4.4　仿真实验

本节进行了仿真实验并展示了仿真结果。提出的 DADC 框架与现有的 AC 框架作了比较，并分析了强化学习训练后的策略在住宅电力负荷协同负荷控制中的效果。此外，本节也对比了在线策略算法和离线策略算法的 DADC 框架分布式训练。

4.4.1　实验设置

仿真环境使用 Open Gym[52] 搭建而成，包括 10 个各不相同的家庭住宅。时间间隔 Δt 为 15 分钟，总时长为 24 小时，因此 Dec-POMDP 的总时间步 T 为 96。基础负荷功率和室外温度的建模采用了真实的电力数据和温度数据，分别可以从 Pecan Street Database 和 NOAA 获取。因为同一微电网下的住宅空间距离较近，因此假设同一时刻所有住宅的室外温度相同。部分状态转移函数、损失函数和参数展示如下，其他在附录中提供。

室内温度的转移函数采取（3.21）的等价简化形式

$$f_i^{\mathrm{AC}}(T, T^{\mathrm{out}}, P, \varrho) = T + \eta_{i,1}^{\mathrm{AC}}(T^{\mathrm{out}} - T) - \eta_{i,2}^{\mathrm{AC}} P + \varrho, \tag{4.23}$$

其中参数 $\eta_{i,1}^{\mathrm{AC}}$ 和 $\eta_{i,2}^{\mathrm{AC}}$ 分别与房间和空调的热力学特性有关。扰动因子 ϱ 服从均匀分布 $\mathcal{U}[-0.1, 0.1]$。

表 4.1　边缘侧住宅参数

住宅编号 i	$T_{i,\min}^{\mathrm{AC}}$	$T_{i,\max}^{\mathrm{AC}}$	$P_{i,\max}^{\mathrm{AC}}$	$\eta_{i,1}^{\mathrm{AC}}$	$\eta_{i,2}^{\mathrm{AC}}$	$P_{i,\max}^{\mathrm{EV}}$	$E_{i,\max}^{\mathrm{EV}}$
1	22.0	26.0	3.0	0.2	0.667	6.67	40
2	22.2	26.2	3.1	0.2	0.645	7.00	42
3	22.4	26.4	3.2	0.2	0.625	7.33	44
4	22.6	26.6	3.3	0.2	0.606	7.67	46
5	22.8	26.8	3.4	0.2	0.588	8.00	48
6	23.0	26.0	3.5	0.2	0.571	8.33	50
7	23.2	27.2	3.6	0.2	0.555	8.67	52
8	23.4	27.4	3.7	0.2	0.541	9.00	54
9	23.6	27.6	3.8	0.2	0.526	9.33	56
10	23.8	27.8	3.9	0.2	0.513	9.67	58

发电机的发电成本函数和调整成本函数分别设置为

$$g_1^{\mathrm{CG}}(P) = \lambda_1^{\mathrm{CG}} P + \lambda_2^{\mathrm{CG}} P^2, \tag{4.24}$$

$$g_2^{\mathrm{CG}}(\Delta P) = \lambda_3^{DG} \Delta P. \tag{4.25}$$

其中 $\lambda_1^{\mathrm{CG}}, \lambda_2^{\mathrm{CG}}, \lambda_3^{\mathrm{CG}}$ 为发电机的成本参数，分别设置为 0.5，0.0125 和 0.1。

所有本地 Critic 具有相同的结构，由三部分组成，如图4.3所示，一个具有两层 128 单元隐藏层和 tanh 激活函数的全连接 MLP，一个具有 64 个单元的 GRU，以及具有一层 128 单元隐藏层和一层输出层的 MLP。随机策略由高斯分布 $\mathcal{N}(\mu, \sigma^2)$ 表示。因此，本地 Actor 有一个 tanh 输出层用来生成均值 μ，还有另一个 sigmoid 输出层用来生成方差 σ^2。本地 Actor 的非输出层与本地 Critic 结构相同。云端前馈网络的 MLP 由一层 tanh 激活的 64 单元隐藏层和一层输出层构成。

4.4.2　基线框架

仿真实验将 DADC 框架与两种基线框架做了比较，分别是 IAC 框架和 DACC 框架。需要强调的是 DADC 框架设计的一个核心目的是为了保护住户的数据隐私。与提出的 DADC 框架和 IAC 框架相比，DACC 框架在实践部署中会引起隐私问题和通信成本问题，因此，DACC 框架的控制效果只在主要结果中报告，不进行具体的分析和比较。下面展示 IAC 框架和 DACC 框架以 PPO 为优化算法的训练。

IAC 框架中的每个 HEMS 具有完全独立的 Actor 和 Critic，即每个 HEMS 的 Actor $\pi_{\mathrm{I}}^i\left(\cdot|\cdot;\theta_{\mathrm{I}}^i\right):\mathcal{O}^i\times\mathcal{A}^i\mapsto[0,+\infty)$ 完全基于该 HEMS 的 Critic $V_{\mathrm{I}}^i\left(\cdot;\phi_{\mathrm{I}}^i\right):\mathcal{O}^i\mapsto\mathbb{R}$ 训练，而不与其他 HEMS 或云端通信。$\pi_{\mathrm{I}}^i\left(\cdot|\cdot;\theta_{\mathrm{I}}^i\right)$ 和 $V_{\mathrm{I}}^i\left(\cdot;\phi_{\mathrm{I}}^i\right)$ 的损失函数分别为

$$\mathcal{L}_{\mathrm{I,a}}^i=\hat{\mathbb{E}}_t\left[\min\left(w_t^i\left(\theta_{\mathrm{I}}^i\right)\hat{A}_t^i,\mathrm{clip}\left(w_t^i\left(\theta_{\mathrm{I}}^i\right),1-\epsilon,1+\epsilon\right)\hat{A}_t^i\right)\right],\forall i\in\mathcal{D},$$
$$\mathcal{L}_{\mathrm{I,c}}^i=\hat{\mathbb{E}}_t\left[\left(V_{\mathrm{I}}^i\left(o_t^i;\phi_{\mathrm{I}}^i\right)-V_{\mathrm{I,old}}^i\left(o_{t+1}^i;\phi_{\mathrm{I,old}}^i\right)-\hat{A}_t^i\right)^2\right],\forall i\in\mathcal{D},$$

其中概率比 $w_t^i(\theta_{\mathrm{I}}^i)$ 和优势函数 $\hat{A}_t^i$ 完全在本地计算：

$$w_t^i=\frac{\pi_{\mathrm{I}}^i\left(a_t^i|o_t^i;\theta_{\mathrm{I}}^i\right)}{\pi_{\mathrm{I,old}}^i\left(a_t^i|o_t^i;\theta_{\mathrm{I,old}}^i\right)},\hat{A}_t^i=\sum_{t'=t}^{T-1}\lambda^{t'-t}\left(-V_{\mathrm{I,old}}^i\left(o_{t'}^i;\phi_{\mathrm{I,old}}^i\right)+r_{t'}+V_{\mathrm{I,old}}^i\left(o_{t'+1}^i;\phi_{\mathrm{I,old}}^i\right)\right).$$

相应地，参数 θ_{I}^i 和 ϕ_{I}^i 的梯度为

$$\Delta\theta_{\mathrm{I}}^i=\hat{\mathbb{E}}_t\left[\nabla_{\theta_{\mathrm{I}}^i}\pi_{\mathrm{I}}^i\left(a_t^i|o_t^i;\theta_{\mathrm{I}}^i\right)\nabla_{w_t^i}\mathcal{L}_{\mathrm{I,a}}^i\right],\forall i\in\mathcal{D},$$
$$\Delta\phi_{\mathrm{I}}^i=\hat{\mathbb{E}}_t\left[\nabla_{\phi_{\mathrm{I}}^i}V_{\mathrm{I}}^i\left(o_t^i;\phi_{\mathrm{I}}^i\right)\left(V_{\mathrm{I}}^i-V_{\mathrm{I,old}}^i-\hat{A}_t^i\right)\right],\forall i\in\mathcal{D}.$$

为了公平比较，IAC 框架的 Actor 和 Critic 与 DADC 框架的本地 Actor 和 Actor 网络结构相同。

DACC 框架为分散式 Actor 和集中式 Critic 结构。分散的 Actor $\pi_{\mathrm{C}}^i\left(\cdot|\cdot;\theta_{\mathrm{C}}^i\right):\mathcal{O}^i\times\mathcal{A}^i\mapsto[0,+\infty)$ 以 HEMS i 的本地观测为输入，而中心化的 Critic $V_{\mathrm{C}}\left(\cdot;\phi_{\mathrm{C}}\right):\mathcal{O}\mapsto\mathbb{R}$ 以所有 HEMS 的观测作为输入。DACC 框架的 Actor 和 Critic 的损失函数分别为

$$\mathcal{L}_{\mathrm{C,a}}^i=\hat{\mathbb{E}}_t\left[\min\left(w_t^i\left(\theta_{\mathrm{C}}^i\right)\hat{A}_t,\mathrm{clip}\left(w_t^i\left(\theta_{\mathrm{I}}^i\right),1-\epsilon,1+\epsilon\right)\hat{A}_t\right)\right],\forall i\in\mathcal{D},$$

$$\mathcal{L}_{\mathrm{C,c}} = \hat{\mathbb{E}}_t \left[\left(V_{\mathrm{C}} \left(o_t; \phi_{\mathrm{C}} \right) - V_{\mathrm{C,old}} \left(o_{t+1}; \phi_{\mathrm{C,old}} \right) - \hat{A}_t \right)^2 \right],$$

其中概率比 w_t^i 在本地计算，

$$w_t^i = \frac{\pi_{\mathrm{C}}^i \left(a_t^i | o_t^i; \theta_{\mathrm{C}}^i \right)}{\pi_{\mathrm{C,old}}^i \left(a_t^i | o_t^i; \theta_{\mathrm{C,old}}^i \right)},$$

而全局优势函数 $\hat{A}_t$ 在云端计算，

$$\hat{A}_t = \sum_{t'=t}^{T-1} \lambda^{t'-t} \left(-V_{\mathrm{C,old}} \left(o_{t'}; \phi_{\mathrm{C,old}} \right) + r_{t'} + V_{\mathrm{C,old}} \left(o_{t'+1}; \phi_{\mathrm{C,old}} \right) \right).$$

相应地，参数 θ_{C}^i 和 ϕ_{C} 的梯度为

$$\Delta\theta_{\mathrm{C}}^i = \hat{\mathbb{E}}_t \left[\nabla_{\theta_{\mathrm{C}}^i} \pi_{\mathrm{C}}^i \left(a_t^i | o_t^i; \theta_{\mathrm{C}}^i \right) \nabla_{w_t^i} \mathcal{L}_{\mathrm{C,a}}^i \right], \forall i \in \mathcal{D},$$

$$\Delta\phi_{\mathrm{C}} = \hat{\mathbb{E}}_t \left[\nabla_{\phi_{\mathrm{C}}} V_{\mathrm{C}} \left(o_t; \phi_{\mathrm{C}} \right) \left(V_{\mathrm{C}} - V_{\mathrm{C,old}} - \hat{A}_t \right) \right].$$

实验中，DACC 框架的 Actor 和 DADC 框架中的 Actor 具有相同的内部结构，而 DACC 框架的集中式 Critic 处于云端，采用了图4.3(b) 中的结构，但输入为所有 HEMS 观测的串联。

仿真实验分别使用学习率为 1×10^{-4} 和 3×10^{-4} 的 Adam 优化器[53]对 Actor 和 Critic 进行优化更新。HEMS 每与环境交互 120 个时间步，网络参数进行一轮更新。为了提高训练效率，同时并行 10 个环境。仿真实验在 8 核 AMD Ryzen 7 3700X 处理器和一个 GeForce RTX 2080 GPU 上进行。

4.4.3　主要结果

以 PPO[24]为优化算法，DADC 框架与其他 AC 框架在住宅负荷协同控制问题上进行了对比。每种框架使用不同的随机数种子进行六次训练。PPO 算法的 GAE 参数设置为 0.95，网络参数在一次采样期间更新 3 次。

为了评估训练过程的控制效果，采用如下评估过程：每经过 1000 天仿真暂停一次，运行 10 天的仿真环境进行评估，评估时每个 HEMS 执行独立的动作选择。

训练曲线如图4.5所示，实线对应于六次训练的平均值，阴影区域对应于六次训练的最小和最大奖励。从图中可以看出，IAC 框架未能学习稳定的协同控制策略，导致控制效果始终不佳。其不稳定的训练可归因于 IAC 框架完全独立的训练过程。在训练时，IAC 框架下的 HEMS 只能观察到其本地信息，无法了解其他 HEMS 的控制行为变化，因此单个 HEMS 面临的是非平稳环境。相比之下，由于集中式的全局 Critic，DACC 框架可以了解所有 HEMS 的信息，因此有能力协调 HEMS 之间的负荷控制。本章提出的 DADC 框架学到的控制策略达到与 DACC 相当的效

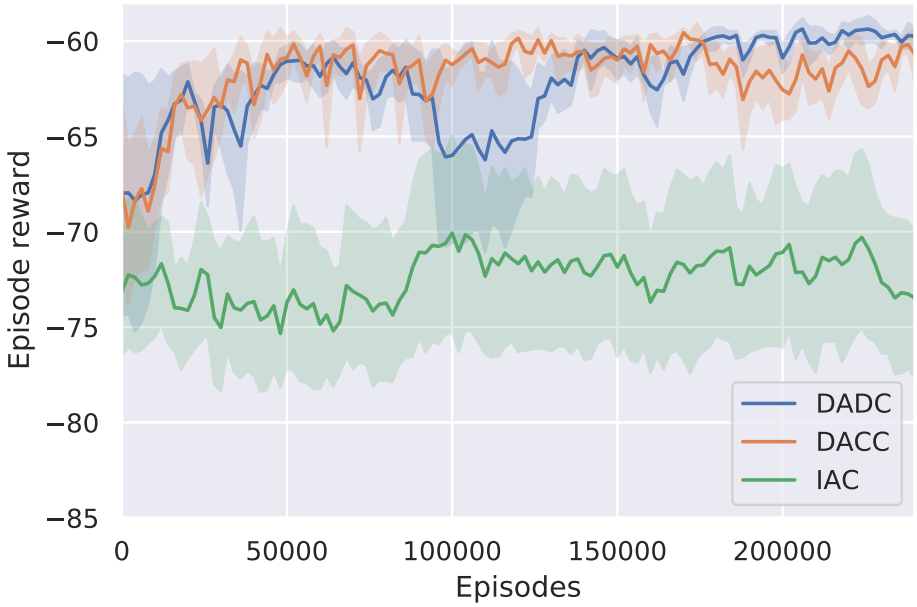


图 4.5　各个框架的训练曲线

果，这说明了 DADC 框架在协同负荷控制问题上的有效性。值得强调的是，DADC 框架能有效保护 HEMS 的本地信息，而 DACC 框架无法做到。因此，图4.5显示了 DADC 框架优于其他 AC 框架的性能。

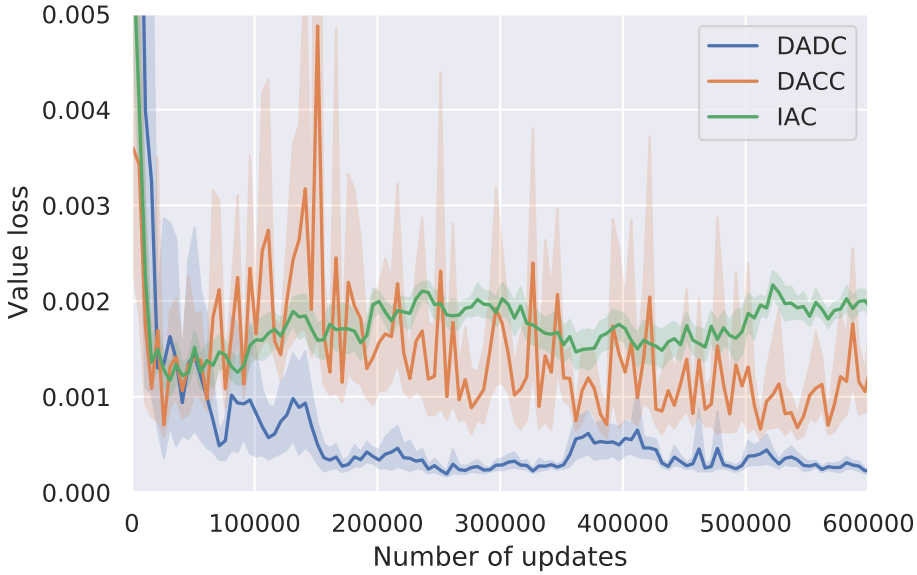


图 4.6　训练过程的值函数估计损失曲线

如本章概述中所言，DADC 框架能够隐式地学习 HEMS 之间的信用分配。为了说明这一点，图4.6绘制了训练过程中各个框架的 Critic 估计损失。可以看出 IAC 框架的独立 Critic 对值函数的估计偏差最大，因为 IAC 框架中完全分散的 HEMS 无法捕获其他 HEMS 在训练期间的控制策略变化。而受益于集中式 Critic，DACC 框架的值函数估计损失相对于 IAC 框架较小。然而，在训练过程中，DACC 框架对

值函数的估计是非常不稳定的。当某个特定 HEMS 改变其控制策略导致全局奖励和全局值函数发生改变时，DACC 框架无法将这种变化迅速归结于该 HEMS，因为 DACC 框架的集中式 Critic 以所有 HEMS 的观测为输入。而在 DADC 框架的设计中，所有 HEMS 的本地 Critic 和云端的前馈网络协作估计全局值函数。某个 HEMS 控制策略的改变明确地反映在其本地值函数的变化上，而本地值函数的更新是通过端到端的梯度更新完成的。在图4.6中，可以观察到 DADC 的值损失比 IAC 和 DACC 小得多，这表明了隐式信用分配在 DADC 框架中的作用。

4.4.4　负荷协同控制效果

表 4.2　DADC 框架和 IAC 框架的测试效果

指标	DADC	IAC
平均成本	58.201 ± 0.953	65.390 ± 1.198
平均发电成本	55.790 ± 1.038	60.535 ± 1.502
平均调整成本	2.411 ± 0.334	4.855 ± 0.776

接下来以 IAC 框架作为对比，研究 DADC 框架对负荷协同调度问题的控制效果。经过分布式训练，在训练时具有最佳评估效果的策略被部署于最终的测试环境。表4.2中的测试结果表明，DADC 框架的平均成本比 IAC 框架降低了 11%，特别是 DADC 框架下的发电机调整成本降低了 50% 以上。发电机调整成本的大幅降低有力说明了 DADC 框架能够有效协调各个 HEMS 的负荷控制策略。

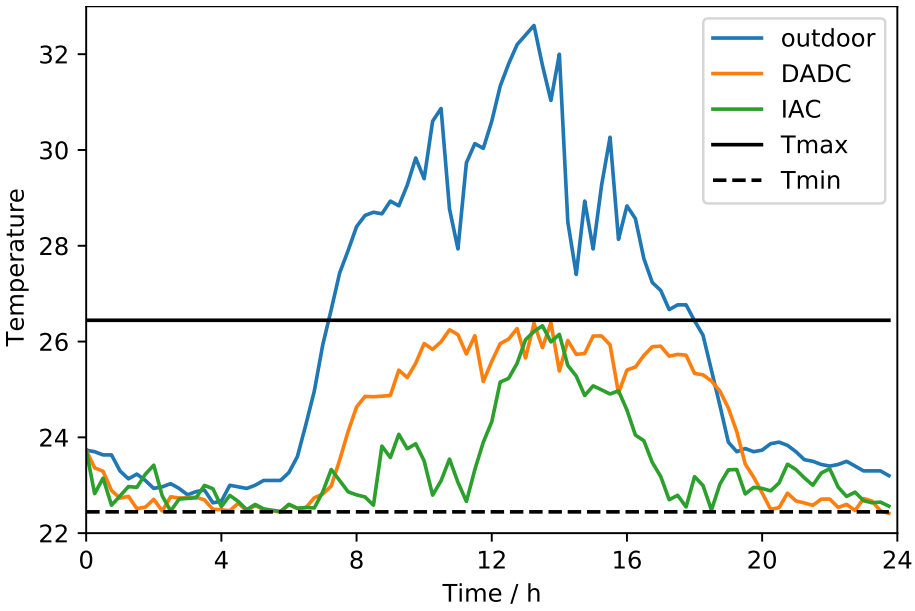


图 4.7　一天内的温度变化

为了展示对空调的控制效果，图4.7绘制了一天中空调控制下的室内温度曲线。黑色实线和虚线分别表示住户温度舒适区的上限温度和下限温度。橙色和绿色线

分别表示由 DADC 框架和 IAC 框架的策略控制的一天室内温度曲线。从图中可以看出，当室外温度较高时，采用 DADC 框架的室内温度更接近上限温度约束，与 IAC 框架相比可以节约能源，降低成本。此外，可以看出使用 DADC 框架的室内温度曲线相对平滑，这意味着与 IAC 框架相比，对空调的功率调整更少。

为了展示总体负荷调度效果，图4.8绘制了基本负荷功率、发电机功率、所有电动汽车的总充电功率和所有空调的总工作功率，分别用蓝色、橙色和绿色区域表示。从图中可以看出使用 DADC 框架时，发电机输出功率的调整相对平稳且发电机输出功率的峰值小于 IAC 框架下的发电机峰值功率。从这个意义上讲，DADC 框架使得各个 HEMS 能够协同地调度电力负荷从而降低微网总成本。

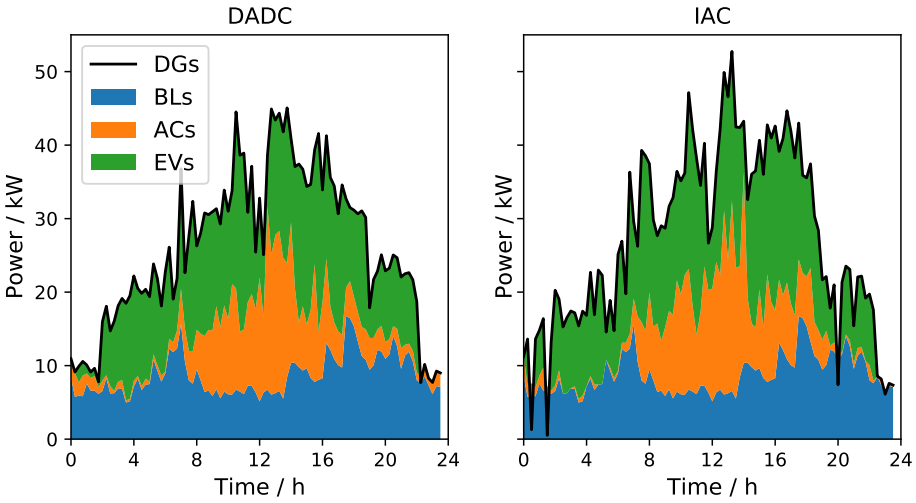


图 4.8　一天内的负荷调度情况

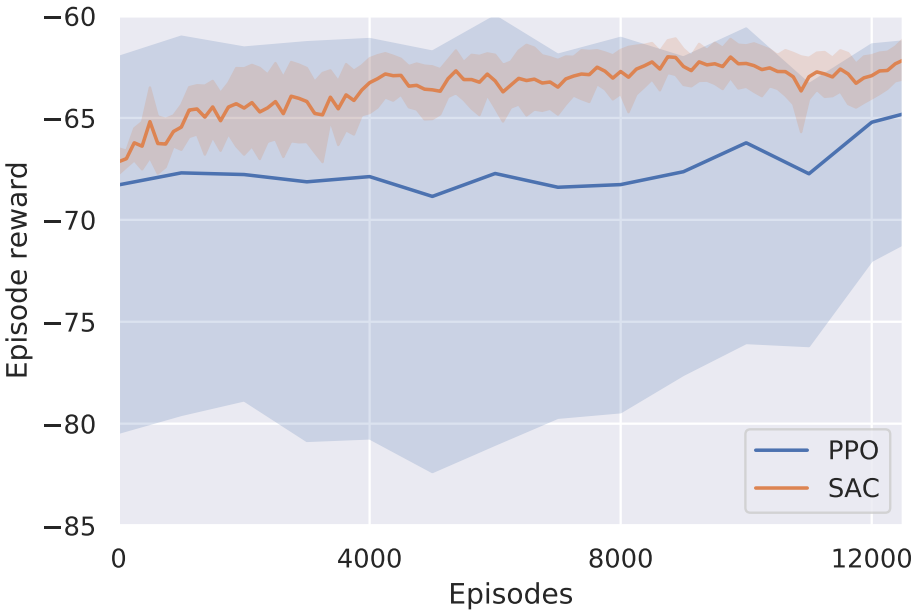


图 4.9　基于环境采样次数的离线策略和在线策略效果对比

4.4.5　离线策略和在线策略对比

DADC 框架可以使用在线策略算法和离线策略算法进行分布式训练。为了研究在线策略和离线策略训练的效果，仿真实验分别以在线策略的 PPO 算法和离线策略的 SAC 算法 DADC 框架进行了优化。SAC 算法的记忆池大小设置为 1×10^5，网络参数每 120 个环境步骤更新 60 次。

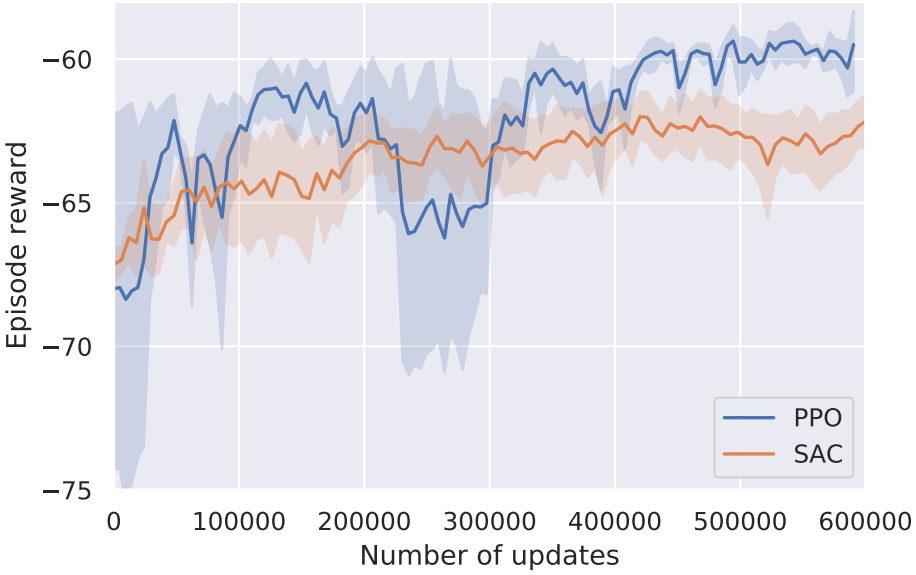


图 4.10　基于网络更新次数的离线策略和在线策略效果对比

从参数更新的频率上可以看出在线策略算法和离线策略算法的不同：离线策略算法对样本的利用率远高于在线策略算法。因此需要从不同尺度进行比较。第一个尺度基于样本数量，即负荷协同控制的仿真天数，如图4.9所示。可以看出 DADC 框架的离线策略训练会以更少的环境交互次数收敛。第二个尺度基于参数的更新次数，如图4.10所示。虽然离线策略训练过程更稳定，但其收敛到控制效果较差的局部最优策略，而在线策略训练可以学习具有更好控制效果的策略。综合两种尺度上的比较结果可以得出结论：因此，离线策略训练在数据效率方面优于在线策略训练，而在线策略训练在学习效率方面优于离线策略训练。

4.5　本章小结

本章针对孤岛微电网住宅负荷协同控制问题设计了云边环境下的分布式隐私保护方案。为了协调调度所有住宅负荷、保护住户数据隐私且降低云边通信成本，本章提出了分散式执行者-分布式评价者（DADC）的多智能体强化学习框架，并分别展示了云边环境中该框架的在线策略优化和离线策略优化。仿真结果表明，提出的 DADC 框架在负荷协同控制效果上远优于独立执行者-评价者（IAC）框架，且

在具有隐私保护和通信成本低的特点下，达到了与多智能体强化学习中通用的分散式执行者-集中式评价者（DACC）框架相当的控制效果。此外，仿真结果还说明了 DADC 框架在线策略优化和离线策略优化的优劣：离线策略训练在数据利用效率方面占优，而在线策略训练在学习效率方面占优。

第 5 章　总结与展望

5.1　研究工作总结

本文围绕强化学习在住宅负荷协同控制应用中的隐私问题，分别设计了集中式和分布式的住宅负荷隐私保护协同控制方案，并提出了适应相应方案的强化学习算法或框架。本文的主要工作和结论总结如下:

- **提出了孤岛微电网场景中适用于强化学习的住宅负荷模型，并通过实验得出了关于如何选择强化学习输入的结论**。
  本文比较了有显式预测和无显式预测的基于强化学习的能量管理方案，其中显式预测环节通过监督学习训练获得。实验结果说明，在加入显式预测的方案下，微电网的运营成本增加了 10% 左右。因此，将强化学习算法应用于能量管理或负荷控制问题时，应直接将当前时刻的观测作为输入，无需通过预测在输入中加入额外的信息。
- **设计了微电网运营商集中管理的住宅负荷隐私保护协同控制方案，并提出了融入循环神经网络的向量强化学习算法以解决微电网运营商面临的高维动作空间和部分可观测问题**。
  相较于标准强化学习算法，提出的向量强化学习算法能够更准确地估计值函数，因而在高维动作空间问题上有更好的训练稳定性。与其他集中式隐私保护方案相比，提出的方案以更小的代价实现了对住户部分隐私数据的保护，并且该方案可以通过灵活调整信用分配机制实现不同的控制目标。
- **设计了云边环境下的分布式住宅负荷隐私保护协同控制方案，并提出分散式执行者-分布式评价者（DADC, Decentralized Actors-Distributed Critics）的多智能体强化学习框架，打破了多智能体强化学习集中式训练、分散式执行的传统范式**。
  与可实现隐私保护的独立执行者-评价者框架相比, DADC 框架通过负荷协同控制使微电网运营成本降低了 11%；在云边环境中，DADC 框架与传统分散式执行者-集中式评价者的多主体强化学习框架相比，以更佳的隐私保护性能和更低的通信成本，实现了相同的控制效果。DADC 框架在线策略优化和离线策略优化的实验结果说明，离线策略训练具有更好的数据利用效率，而在线策略训练具有更佳的学习效率。

5.2　未来研究展望

本研究试图以简单直接的方式解决强化学习在住宅负荷协同控制应用中的隐私保护问题。由于本人能力和精力有限，本文的研究结果在很多方面还有细化和提高的空间，下面简要讨论其中最突出的两点：

- 提高强化学习的真实样本利用效率。本文使用的强化学习算法的训练步数在十万步以上，这在仿真环境中可以实现，但在真实孤岛微电网的协同控制场景中是不可接受的。因此，有必要利用离线强化学习、元强化学习等方法，先在仿真环境中训练出可迅速适应新场景的策略，再将其部署于真实物理场景的训练。
- 结合隐私分析的理论框架。本文提出的数据隐私保护基于常识和直观感受，缺乏理论支撑。因此，有必要结合差分隐私、同态加密等理论框架给出隐私的严格数学定义，在此基础上进行相应的隐私保护设计和分析。

参考文献

[1] 国家能源局. 截至 3 月底全国发电装机容量约 24 亿千瓦 3 月份可再生能源发电量较快增长[EB/OL]. (2022-04-20)[2022-04-22]. http://www.nea.gov.cn/2022-04/22/c_1310569074.htm.

[2] Insights G. Microgrid deployment tracker 1q22[EB/OL]. (2022-4)[2022-04]. https://guidehouseinsights.com/reports/microgrid-deployment-tracker-1q22.

[3] Administration U E I. Electric power annual 2020[EB/OL]. (2021-10-29)[2022-03-10]. https://www.eia.gov/electricity/annual/.

[4] Vanouni M, Lu N. Improving the centralized control of thermostatically controlled appliances by obtaining the right information[J/OL]. IEEE Transactions on Smart Grid, 2015, 6(2): 946-948. DOI: 10.1109/TSG.2014.2357211.

[5] Cui Q, Wang X, Wang X, et al. Residential appliances direct load control in real-time using cooperative game[J/OL]. IEEE Transactions on Power Systems, 2016, 31(1): 226-233. DOI: 10.1109/TPWRS.2015.2391774.

[6] Luo F, Kong W, Ranzi G, et al. Optimal home energy management system with demand charge tariff and appliance operational dependencies[J/OL]. IEEE Transactions on Smart Grid, 2020, 11(1): 4-14. DOI: 10.1109/TSG.2019.2915679.

[7] Yu L, Xie W, Xie D, et al. Deep reinforcement learning for smart home energy management [J/OL]. IEEE Internet of Things Journal, 2020, 7(4): 2751-2762. DOI: 10.1109/JIOT.2019.2957289.

[8] Du Y, Zandi H, Kotevska O, et al. Intelligent multi-zone residential hvac control strategy based on deep reinforcement learning[J]. Applied Energy, 2021, 281: 116117.

[9] Liu Z, Wu Q, Huang S, et al. Optimal day-ahead charging scheduling of electric vehicles through an aggregative game model[J/OL]. IEEE Transactions on Smart Grid, 2018, 9(5): 5173-5184. DOI: 10.1109/TSG.2017.2682340.

[10] Hou H, Xue M, Xu Y, et al. Multi-objective economic dispatch of a microgrid considering electric vehicle and transferable load[J]. Applied Energy, 2020, 262: 114489.

[11] Mohammadi J, Hug G, Kar S. A fully distributed cooperative charging approach for plug-in electric vehicles[J]. IEEE Transactions on Smart Grid, 2016, 9(4): 3507-3518.

[12] Yang Y, Jia Q S, Deconinck G, et al. Distributed coordination of ev charging with renewable energy in a microgrid of buildings[J]. IEEE Transactions on Smart Grid, 2017, 9(6): 6253-6264.

[13] Ahmadi M, Rosenberger J M, Lee W J, et al. Optimizing load control in a collaborative residential microgrid environment[J/OL]. IEEE Transactions on Smart Grid, 2015, 6(3): 1196-1207. DOI: 10.1109/TSG.2014.2387202.

[14] Ahmed N, Levorato M, Li G P. Residential consumer-centric demand side management[J/OL]. IEEE Transactions on Smart Grid, 2018, 9(5): 4513-4524. DOI: 10.1109/TSG.2017.2661991.

[15] Baniasadi A, Habibi D, Bass O, et al. Optimal real-time residential thermal energy management for peak-load shifting with experimental verification[J/OL]. IEEE Transactions on Smart Grid, 2019, 10(5): 5587-5599. DOI: 10.1109/TSG.2018.2887232.

[16] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of artificial intelligence research, 1996, 4: 237-285.

[17] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[A]. 2013.

[18] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. nature, 2016, 529(7587): 484-489.

[19] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.

[20] Gu S, Holly E, Lillicrap T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]//2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017: 3389-3396.

[21] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]// Proceedings of the AAAI conference on artificial intelligence: volume 30. 2016.

[22] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1995-2003.

[23] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning [C]//International conference on machine learning. PMLR, 2016: 1928-1937.

[24] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[A]. 2017.

[25] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//ICML. PMLR, 2018: 1861-1870.

[26] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward[C]//AAMAS. 2018: 2085-2087.

[27] Rashid T, Samvelyan M, Schroeder C, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]//ICML. PMLR, 2018: 4295-4304.

[28] Son K, Kim D, Kang W J, et al. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2019: 5887-5896.

[29] Lowe R, WU Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. NIPS, 2017, 30: 6379-6390.

[30] Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients[C]//AAAI: volume 32. 2018.

[31] Iqbal S, Sha F. Actor-attention-critic for multi-agent reinforcement learning[C]//ICML. PMLR, 2019: 2961-2970.

[32] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning, 1992, 8(3): 229-256.

[33] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018.

[34] Hausknecht M, Stone P. Deep recurrent q-learning for partially observable mdps[C]//2015 aaai fall symposium series. 2015.

[35] Wan Z, Li H, He H, et al. Model-free real-time ev charging scheduling based on deep reinforcement learning[J]. IEEE Transactions on Smart Grid, 2018, 10(5): 5246-5257.

[36] Wang B, Li Y, Ming W, et al. Deep reinforcement learning method for demand response management of interruptible load[J]. IEEE Transactions on Smart Grid, 2020, 11(4): 3146-3155.

[37] Ye Y, Qiu D, Wu X, et al. Model-free real-time autonomous control for a residential multi-energy system using deep reinforcement learning[J]. IEEE Transactions on Smart Grid, 2020, 11(4): 3068-3082.

[38] Zhang C, Kuppannagari S R, Xiong C, et al. A cooperative multi-agent deep reinforcement learning framework for real-time residential load scheduling[C]//International Conference on Internet of Things Design and Implementation. 2019: 59-69.

[39] Lee J, Wang W, Niyato D. Demand-side scheduling based on deep actor-critic learning for smart grids[A]. 2020.

[40] Chung H M, Maharjan S, Zhang Y, et al. Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids[J/OL]. IEEE TII, 2021, 17(4): 2752-2763. DOI: 10.1109/TII.2020.3007167.

[41] Halder A, Geng X, Kumar P R, et al. Architecture and algorithms for privacy preserving thermal inertial load management by a load serving entity[J/OL]. IEEE Transactions on Power Systems, 2017, 32(4): 3275-3286. DOI: 10.1109/TPWRS.2016.2628055.

[42] Gong Y, Cai Y, Guo Y, et al. A privacy-preserving scheme for incentive-based demand response in the smart grid[J/OL]. IEEE Transactions on Smart Grid, 2016, 7(3): 1304-1313. DOI: 10.1109/TSG.2015.2412091.

[43] Zhang Q, Dehghanpour K, Wang Z, et al. A learning-based power management method for networked microgrids under incomplete information[J/OL]. IEEE Transactions on Smart Grid, 2020, 11(2): 1193-1204. DOI: 10.1109/TSG.2019.2933502.

[44] Ye Y, Papadaskalopoulos D, Yuan Q, et al. Multi-agent deep reinforcement learning for coordinated energy trading and flexibility services provision in local electricity markets[J/OL]. IEEE Transactions on Smart Grid, 2022: 1-1. DOI: 10.1109/TSG.2022.3149266.

[45] Xu X, Jia Y, Xu Y, et al. A multi-agent reinforcement learning-based data-driven method for home energy management[J/OL]. IEEE Transactions on Smart Grid, 2020, 11(4): 3201-3211. DOI: 10.1109/TSG.2020.2971427.

[46] Liu B, Akcakaya M, Mcdermott T E. Automated control of transactive hvacs in energy distribution systems[J/OL]. IEEE Transactions on Smart Grid, 2021, 12(3): 2462-2471. DOI: 10.1109/TSG.2020.3042498.

[47] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation[A]. 2015.

[48] Lin L, Guan X, Peng Y, et al. Deep reinforcement learning for economic dispatch of virtual power plant in internet of energy[J]. IEEE Internet of Things Journal, 2020, 7(7): 6288-6301.

[49] Mocanu E, Mocanu D C, Nguyen P H, et al. On-line building energy optimization using deep reinforcement learning[J]. IEEE transactions on smart grid, 2018, 10(4): 3698-3708.

[50] Devlin S, Yliniemi L, Kudenko D, et al. Potential-based difference rewards for multiagent reinforcement learning[C]//Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. 2014: 165-172.

[51] Halder A, Geng X, Kumar P R, et al. Architecture and algorithms for privacy preserving thermal inertial load management by a load serving entity[J/OL]. IEEE Transactions on Power Systems, 2017, 32(4): 3275-3286. DOI: 10.1109/TPWRS.2016.2628055.

[52] Brockman G, Cheung V, Pettersson L, et al. Openai gym[A]. 2016. arXiv: 1606.01540.

[53] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//ICLR. 2015.

[54] Qin Z, Liu D, Hua H, et al. Privacy preserving load control of residential microgrid via deep reinforcement learning[J/OL]. IEEE TSG, 2021: 1-1. DOI: 10.1109/TSG.2021.3088290.

附录 A　补充内容

储能设备的电量状态转移函数和成本函数指定如下[54]。

$$f^{\mathrm{BES}}\left(\mathrm{SOC}, P^{\mathrm{BES}}\right)=\begin{cases}\mathrm{SOC}+\frac{\eta_c}{C}P^{\mathrm{BES}}, & \text{if } P^{\mathrm{BES}}\geqslant 0,\\ \mathrm{SOC}+\frac{1}{C\eta_d}P^{\mathrm{BES}}, & \text{otherwise,}\end{cases} \tag{A.1}$$

$$g^{\mathrm{BES}}\left(\mathrm{SOC}, P^{\mathrm{BES}}\right)=\begin{cases}\lambda_1|P^{\mathrm{BES}}|, & \text{if SOC}<0.5,\\ \lambda_2|P^{\mathrm{BES}}|, & \text{otherwise,}\end{cases} \tag{A.2}$$

C、η_c 和 η_d 表示储能设备的容量、充放电效率系数，λ_1 和 λ_2 分别是每千瓦时的最大和最小退化成本，分别对应于低电量状态和高电量状态。参数设置见表 A.1.

表 A.1　环境和算法设置

参数	值	参数	值	参数	值
d_{max}	400 kW	c_{max}	400 kW	C	2000 kWh
η_d	0.95	η_c	0.95	λ_1	0.013
λ_2	0.005	ϵ	0.2	K	3

致　谢

衷心感谢导师曹军威研究员的精心指导。加入曹老师实验室是我的幸运。在科研方面，曹老师提供了宽松的研究环境和丰富的研究资源，使我有机会根据自己的研究兴趣充分探索未知领域；在生活方面，曹老师处处体现出对学生的尊重，桃李不言，下自成蹊。

感谢华昊辰博后和秦钰超师兄在我进入实验室初期给予的大量指导。在他们的耐心帮助下，我得以窥探学术研究的全貌，从而真正迈入科研的大门。感谢刘迪博后与我在学术上的讨论和合作。感谢牛津大学的董楠卿学长对本文第四章相关研究的帮助。感谢实验室李宇童和丁晓可同学与我在生活和学习上的交流。感谢实验室所有工作人员提供的技术支持。感谢清华大学和自动化系的培养。

感谢王静同学对我学习和科研的鼓励、支持和鞭策。逆耳忠言，永感于心。

最后感谢我的父母，他们给了我战胜困难、战胜自己的勇气。想到他们，我内心充满了宁静、温暖和力量。希望我永远是他们骄傲的孩子。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间完成的相关学术成果

个人简历

1998 年 9 月 9 日出生于山东省商河县。

2015 年 9 月考入北京航空航天大学自动化科学与电气工程学院自动化专业，2019 年 7 月本科毕业并获得工学学士学位。

2019 年 9 月考入清华大学自动化系控制科学与工程专业，攻读工学硕士至今。

在学期间完成的相关学术成果

学术论文：

[1] Qin Z, Liu D, Hua H, Cao J. Privacy Preserving Load Control of Residential Microgrid via Deep Reinforcement Learning. IEEE Transactions on Smart Grid, vol. 12, no. 5, pp. 4079-4089, Sept. 2021. （中科院一区 SCI，影响因子：8.96）

[2] Hua H*, Qin Z*, Dong N, et al. Data-Driven Dynamical Control for Bottom-up Energy Internet System. IEEE Transactions on Sustainable Energy, vol. 13, no. 1, pp. 315-327, Jan. 2022. （中科院一区 SCI，影响因子：7.917，* 共同一作）

[3] Qin Z, Zhang H, Zhao Y, Xie H, Cao J. Does Explicit Prediction Matter in Deep Reinforcement Learning-Based Energy Management? 2021 IEEE International Conference on Energy Internet, 2021, pp. 13-19.（EI，检索号：20221311862237，最佳会议论文奖）

[4] Qin Z, Hua H, Liang H, Herzallah R, Zhou Y and Cao J. Optimal Electricity Trading Strategy for a Household Microgrid. 2020 IEEE 16th International Conference on Control & Automation, 2020, pp. 1308-1313.（EI，检索号：20205209687492）

专利：

[5] 秦兆铭, 曹军威. 一种负荷协同控制方法及装置：中国，202111284855.8.（已受理）

[6] 秦兆铭，曹军威. 基于元强化学习的大规模负荷需求响应策略、系统及设备：中国，202111284855.8.（已受理）

指导教师学术评语

负荷控制是维持微电网功率平衡、提高可再生能源消纳率的重要手段，而隐私数据保护是推广住宅电力负荷控制的关键。论文由作者独立完成，针对强化学习在住宅电力负荷协同控制应用中的隐私问题展开研究，具有重要的理论意义和应用价值。

论文取得的主要成果包括：

(1) 针对强化学习在能量管理和负荷控制应用中的争议，比较了基于预测的强化学习和端到端的强化学习在一般能量管理问题上的控制效果，得出了“不引入额外信息的预测对强化学习性能没有正面影响”的结论。

(2) 设计了微电网运营商集中管理的负荷控制方案，并针对方案存在的高维动作空间和部分可观测问题，提出了基于差分奖励的向量强化学习算法，并融合了循环神经网络，从而稳定了强化学习的训练过程。

(3) 设计了云边环境下的分布式负荷控制方案，并针对方案中存在的隐私保护和通信成本问题，提出了分散式执行者-分布式评价者的多智能体强化学习框架，在严格保护住户数据、降低通信成本的同时，有效地协同了各个住宅的负荷控制。

论文工作表明秦兆铭同学具备扎实的专业基础知识，具有独立从事科学研究工作的能力，学术作风严谨求实。论文写作规范，逻辑性强，结构合理，层次分明，表述清晰，达到了硕士学位论文的要求。同意对秦兆铭同学安排硕士学位论文答辩。

答辩委员会决议书

论文提出了……

论文取得的主要创新性成果包括:

1. ……

2. ……

3. ……

论文工作表明作者在 ××××× 具有 ××××× 知识，具有 ×××× 能力，论文 ××××，答辩 ××××。

答辩委员会表决，（× 票/一致）同意通过论文答辩，并建议授予 ××× （姓名）××× （门类）学博士/硕士学位。