GenFedRL：面向深度强化学习智能体的通用联邦强化学习框架

金彪[1,2]，李逸康[1]，姚志强[1,2]，陈瑜霖[1]，熊金波[1,2]
（1. 福建师范大学计算机与网络空间安全学院，福建 福州 350007；2. 大数据分析与应用福建省高校工程研究中心，福建 福州 350007）

摘 要：针对智能物联网中，搭载深度强化学习智能体的智能设备缺乏有效安全数据共享机制的问题，提出一种面向深度强化学习智能体的通用联邦强化学习（GenFedRL）框架。GenFedRL 不需要共享深度强化学习智能体的本地私有数据，而通过模型共享技术实现共同训练，在保护各智能体私有数据隐私的同时，有效地利用其数据资源和计算资源。为应对现实通信环境的复杂性与满足加速训练的需要，为 GenFedRL 设计了基于同步并行的模型共享机制。结合常见深度强化学习算法自身的模型结构特点，基于 FedAvg 算法设计了适用于单网络结构与多网络结构的通用联邦强化学习算法，进而实现了具有同种网络结构的智能体间的模型共享机制，更好地保护各类智能体的私有数据。仿真实验表明，即使在大部分数据节点无法参与训练的恶劣通信环境下，常见深度强化学习算法智能体在所提框架上仍表现出良好的性能。

关键词：智能物联网；联邦学习；联邦强化学习；深度强化学习
中图分类号：TN92
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023122

GenFedRL: a general federated reinforcement learning framework for deep reinforcement learning agents

JIN Biao[1,2], LI Yikang[1], YAO Zhiqiang[1,2], CHEN Yulin[1], XIONG Jinbo[1,2]
1. College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350007, China
2. Fujian Provincial Colleges and University Engineering Research Center of Big Data Analysis and Application, Fuzhou 350007, China

Abstract: To solve the problem that intelligent devices equipped with deep reinforcement learning agents lack effective security data sharing mechanisms in the intelligent Internet of things, a general federated reinforcement learning (GenFedRL) framework was proposed for deep reinforcement learning agents. The joint training through model-sharing technology was realized by GenFedRL without the need to share the local private data of deep reinforcement learning agents. Each agent device's data and computing resources could be effectively used without disclosing the privacy of its private data. To cope with the complexity of the real communication environment and meet the need to accelerate the training speed, a model-sharing mechanism based on synchronization and parallel was designed for GenFedRL. Combined with the model structure characteristics of common deep reinforcement learning algorithms, general federated reinforcement learning algorithm suitable for single network structure and multi-network structure was designed based on the FedAvg algorithm, respectively. Then, the model sharing mechanism among agents with the same network structure was implemented to protect the private data of various agents better. Simulation experiments show that common deep reinforcement learning algorithms still perform well in GenFedRL even in the harsh communication environment where most data nodes cannot participate in training.

Keywords: intelligent Internet of things, federal learning, federal reinforcement learning, deep reinforcement learning

收稿日期：2022−12−19；**修回日期：**2023−03−29
通信作者：姚志强，yzq@fjnu.edu.cn
基金项目：国家自然科学基金资助项目（No.62272103）
Foundation Item: The National Natural Science Foundation of China (No.62272103)

0 引言

物联网作为一种极具潜力的通信模式，在工业、交通、环境监测、智能家居等领域有着广泛的应用。在物联网中，各种智能设备通过局域网或互联网相互连接，实现信息共享，其中不乏搭载具有自学习能力的深度强化学习智能体的智能设备。智能设备利用其所携带的传感器采集大量的、丰富的环境状态信息。若能共享这些数据，将有助于物联网服务提供商训练出更优秀的深度强化学习模型，进而为用户提供更高质量的服务。

然而，物联网中智能设备的拥有者彼此间很难建立起信任，同时其对物联网服务提供商的信任度也十分有限。一方面，在未经用户允许的情况下，用户的隐私数据不应被其他用户获得，物联网服务提供商也不应收集用户的隐私数据。另一方面，即使智能设备的拥有者愿意共享数据，他们也不愿意在没有安全数据共享机制的情况下共享自己的数据。因此，为了激励物联网中具有深度强化学习智能体的智能设备的拥有者参与数据共享，必须提供一种有效的、安全的数据共享方案。

作为一种分布式安全数据共享的解决方案，联邦学习[1]一经提出就受到广泛关注。最早的联邦学习基于C/S架构，参与者仅需基于其自身拥有的数据在本地完成训练任务，不需要共享本地数据，从而降低隐私数据泄露的风险。将联邦学习引入深度强化学习领域有助于实现包括智能设备在内的多种实体间的安全数据共享。

近年来，深度强化学习在众多领域取得了优异的成绩[2-4]。它被认为是解决现实世界中顺序决策问题的一种颇具前景的解决方案[5]。深度强化学习算法的一大特点是它的训练数据由智能体自身与环境交互而来。智能体将训练数据存储于经验池等结构中，用于迭代优化自身的策略。这些训练数据包含环境状态、智能体采取的动作以及获得的奖励值等重要信息。

在现实中，个人或组织机构各自拥有的内嵌深度强化学习智能体的物联网智能设备往往分布于不同的地理位置。这些智能设备拥有的计算资源具有总量大但分布分散的特点。当它们面临相同或相似但互不干扰的任务环境时，就存在共同合作训练一个决策人工智能的可能性。例如，当不同工厂使用了相同或相似的燃煤锅炉系统时，它们就具备了共同合作训练深度强化学习智能体进行最优控制的基础条件[6]。

尽管并不是所有由智能设备生成的数据都有保护的必要性，个人或组织机构之间智能设备的合作仍面临数据隐私保护的挑战。以分属于不同个人或组织机构的环境监测机器人之间的合作为例，携带多种传感器的环境监测机器人在与环境的互动中，通过深度强化学习智能体习得最佳的行动策略，并在交互过程中收集具有机密性质的地理环境信息。此时，分布式强化学习并不适用于该种场景。

尽管分布式强化学习技术在近几年得到了长足的发展，其仍然面向集中式分布的高性能计算机集群，通过高效的并行化技术快速完成高难度的深度强化学习任务。具体而言，分布式强化学习令多个经验收集者高效并行地探索同一类环境并收集训练数据。训练数据被发送至中心化的经验学习者，由经验学习者学习最新策略。经验学习者每隔一段时间将最新策略发送给经验收集者。经验收集者再以最新策略探索环境，重复上述过程直到策略达到指定要求。显然，在此过程中分布式强化学习并未考虑如何防止隐私泄露。当拥有内嵌深度强化学习智能体的物联网智能设备的个人或组织机构面临相同或相似且互不干扰的任务环境，并希望在不共享彼此数据的前提下共同训练同一模型时，联邦学习技术为他们的合作提供了可能性。联邦学习与深度强化学习相结合就产生了联邦强化学习这一概念。

然而，联邦学习与深度强化学习相结合的过程中存在一些挑战。联邦学习中通常有多个数据节点与单个协调服务器，数据节点常代表分布于不同地理位置的参与者。在传统联邦学习中，参与者在最初就已拥有一定数量的本地数据，且在本地训练过程中其本地数据往往不会发生变化。而在联邦强化学习中，由于参与者为内嵌深度强化学习的智能体，其所进行的是一个从零开始生成本地经验的自学习过程，最初通常并不具有本地数据。同时，本地数据由参与者与环境的不断交互而产生，并随着深度强化学习智能体中神经网络的更新而丰富多样。这些差异导致深度强化学习智能体难以被直接应用于传统的联邦学习框架。

目前，针对传统的联邦学习框架，学术界与工业界已有不少的研究成果，但深度强化学习智能体

在联邦学习框架下的研究应用仍处于初级阶段，并且大都缺乏对框架通用性的考虑。

为此，本文针对内嵌深度强化学习智能体的参与者，提出了一种面向深度强化学习智能体的通用联邦强化学习（GenFedRL, general federated reinforcement learning）框架。在初步实现隐私保护功能的同时，所提框架具有高通用性以及应对恶劣通信环境的能力，为将来引入高级安全多方计算技术奠定基础。

值得强调的是，本文提出的通用联邦强化学习框架继承了联邦学习的基本定义，采用 C/S 架构，在参与方使用相同算法、相同的神经网络结构，以及面对相同或相似且互不干扰的任务环境的基础上，兼容常见的深度强化学习算法。

本文主要的研究贡献如下。

1) 提出 GenFedRL 框架，以联邦学习为基础，初步实现了对包含深度强化学习智能体的数据节点的隐私保护，为将来引入高级安全多方计算技术奠定基础。

2) 为应对现实通信环境的复杂性以及满足加速训练速度的需要，设计基于同步并行的模型共享机制。

3) 结合常见深度强化学习算法自身的模型结构特点，基于 FedAvg（federated averaging）算法[1]设计了适用于单网络结构与多网络结构的通用联邦强化学习算法。

4) 在 GenFedRL 上对常见深度强化学习智能体进行性能评估。仿真实验表明，即使在大部分数据节点无法参与训练的恶劣通信环境下，GenFedRL 仍表现出良好的性能。

1　相关研究

在联邦学习被提出以前，为进一步提高学习效率，强化学习与分布式机器学习方法相结合形成了分布式强化学习。分布式强化学习将单个强化学习任务分配到多个计算节点上，以同步或者异步的方式并行化来提高训练速度。近年来，已经有 IMPALA[7]、R2D2[8]、SEED[9]、Acme[10]等性能强大的分布式强化学习算法被开发出来。

IMPALA 算法通常由多个经验生产者 Actor 与单个经验学习者 Learner 组成。Actor 负责使用策略网络不断探索环境，在此过程中收集训练数据，并将训练数据发送至 Learner。Learner 根据训练数据训练自身以得到最新策略网络，并且每隔一段时间将最新策略同步至 Actor。由于 Actor 直接发送训练数据给 Learner，Actor 没有保护自身的训练数据。

R2D2 算法通常由多个经验生产者 Actor 与单个经验学习者 Learner 组成，并具有优先经验回放经验池[11]。与 IMPALA 算法不同，R2D2 中的多个 Actor 与环境交互，收集训练数据并发送至同一经验池中保存。由于 Actor 直接发送训练数据到中心化的经验池，Actor 没有保护自身的训练数据。

SEED 算法通常由多个经验生产者 Actor 与单个经验学习者 Learner 组成。SEED 中的 Actor 只负责观测环境，及时地将环境状态发送给 Learner。Learner 做出决策并将动作返回给 Actor，Actor 在环境中执行动作。Learner 在训练过程中会将训练数据保存于自身的经验池。由于 Actor 直接发送环境状态给 Learner，Actor 没有保护自身的训练数据。

Acme 算法指出，多个经验生产者 Actor 与单个经验学习者 Learner 中应存在一个数据存储传输系统，作为它们沟通的桥梁。进而，Acme 算法提出了一个高效、灵活、稳定的数据存储系统 Reverb。该系统显著增强了分布式强化学习的通用性。不过，Actor 仍然需要将训练数据发送给一个中心化的数据存储传输系统，使 Actor 同样没有保护自身的训练数据。

由此可见，分布式强化学习往往注重于以并行化技术实现加速训练而忽视了并行化带来的隐私保护问题。

自 Mcmahan 等[1]提出联邦学习以来，研究者意识到通过共享本地模型参数而不需要共享本地数据来共同训练一个模型是可行的。在人们日益增长的数据隐私保护需求下，联邦学习在各行各业的应用迅速发展，越来越多原本用于单机环境或者传统分布式环境的机器学习应用被考虑用于联邦学习，其中包括物联网与深度强化学习。

物联网中存在大量分属于不同个人或组织机构且缺乏隐私保护的边缘智能计算设备。当这些设备试图在不共享数据的情况下共同训练一个机器学习模型时，将非常契合联邦学习的适用范围。联邦学习也因此逐渐成为物联网边缘计算的新范式[12-13]。然而，现有的研究普遍忽略了物联网中可能存在的搭载深度强化学习智能体的智能设备。更准确地说，目前深度强化学习在联邦学习中最主要的应用是利

用深度强化学习技术帮助联邦学习选择优质参与方数据节点[14-15]，而对于联邦学习参与方自身进行深度强化学习任务的研究仍处于初级阶段。为方便后续描述，本文将联邦学习参与方自身执行深度强化学习任务的情形称为联邦强化学习。

目前已存在一些基于模型共享但仅针对某一种特定深度强化学习智能体和特定应用领域而设计的联邦强化学习方法。根据智能体内使用的算法类型，这些研究可分为基于价值的联邦强化学习算法与基于策略的联邦强化学习算法。

基于价值的深度强化学习算法中，最具有代表性的算法是 DQN（deep Q-network）算法。DQN 由 Mnih 等[16]提出，并被进一步完善。它是一种基于价值的时序差分算法[17]，利用深度神经网络学习最优 Q 值函数，也是第一个将强化学习与深度学习相结合的深度强化学习算法。由于其结构相对简单，DQN 也最先被应用于联邦强化学习的研究，形成了基于价值的联邦强化学习算法。

基于价值的联邦强化学习算法的主要研究如下。文献[6]提出，2019 年出现了第一种适用于 DQN 智能体的联邦强化学习框架，独创性地引入了高斯差分隐私加密方法和一个共享的全连接层以加密智能体与服务器之间的信息，使参与训练的智能体可以在不泄露本地数据与本地模型的情况下，仅共享有限的信息即可帮助智能体训练各自的本地模型。Nadiger 等[18]提出一种适用于 DQN 智能体的带有分组策略的联邦强化学习框架，分组策略使框架内具有多个模型聚合中心，以进行差异化训练。同时，该框架在服务器和客户端之间引入一个通信策略以决定更新服务器上的全局模型或者将全局模型共享给客户端，仿真实验表明，该框架可以缩短智能体适应用户个性化行为所需的时间。Liu 等[19]提出一种适用于内嵌 DQN 智能体的机器人的终身联邦强化学习框架，运用知识融合算法来训练云端共享模型，并采用高效的迁移学习方法来共享云端共享模型，仿真实验表明，该框架极大地提高了机器人适应实际环境的学习效率。Mowla 等[20]提出一种以自适应联邦强化学习为基础，适用于 DQN 智能体的干扰攻击防御策略，并将其应用于无人机的分散通信网络——飞行自组网，以抵御来自敌方的干扰攻击，仿真实验表明，该防御策略的平均准确率比传统的分布式干扰检测机制的平均准确率高 39.9%。Wang 等[21]提出一种基于 Double DQN[22]智能体的联邦协同边缘缓存框架，用于解决移动网络中的大规模内容访问带来的通信困难性，Double DQN 是 DQN 的一种改进，仿真实验表明，该框架能有效降低性能损失和平均时延，减少回程流量和提高命中率。

基于策略的强化学习算法中，最具有代表性的算法是 Actor 类算法与 Actor Critic 类算法。Actor 类算法只有策略网络 Actor，直接用神经网络学习参数化的策略。Actor 类算法包括 reinforce 算法[23]、VPG（vanilla policy gradient）算法[24]等。目前 Actor 类算法已经被更优秀的 Actor Critic 类算法取代。Actor Critic 类算法结构相对简单，最简单的 Actor Critic 类算法仅有 2 个神经网络，包括策略网络 Actor 与评价网络 Critic。类似于 DQN，Actor Critic 被应用于联邦强化学习的研究同样具有潜在的可行性。目前已有一些基于 Actor Critic 类算法的联邦强化学习研究，形成了基于策略的联邦强化学习算法。下文中提及的 Actor Critic 算法[25]、PPO（proximal policy optimization）算法[26]、DDPG（deep deterministic policy gradient）算法[27]和 SAC（soft actor critic）算法[28]属于 Actor Critic 类算法。

基于策略的联邦强化学习算法的主要研究如下。Lim 等[29]提出一种适用于 PPO 智能体的联邦强化学习框架，使多个强化学习智能体可以在类型相同但状态略有不同的物联网设备上学习最优控制策略，仿真实验表明，该框架可以有效地加速多个物联网设备的学习过程。Yoo 等[30]提出一种新型的空中接入网，其核心是基于 PPO 算法的联邦强化学习框架 FRL，仿真实验表明，该空中接入网提供的通信资源约为仅使用卫星的空中接入网的 3.25 倍，并降低了 5.1%的通信时延。Hu 等[31]提出现有的联邦强化学习方法很少利用强化学习算法的结构，而仅局限于特定的场景或算法。针对该种情况，他们提出一种基于奖励塑形技术[32]的通用联邦强化学习算法。该算法根据每个客户端发送的加密状态与下一个状态生成个性化的奖励信号。该信号作为协调信息在不同任务的客户端之间共享，以提高每个客户端的训练速度和策略质量。

联邦强化学习作为一种特殊的分布式强化学习，因其能保护隐私的特点而备受关注，并逐渐应用于物联网、通信、机器人等具有一定隐私保护需求的领域。但正如 Hu 等[31]提出的那样，目前联邦

强化学习的相关研究的主要形式是针对单一深度强化学习智能体而设计的联邦强化学习框架或单一深度强化学习算法在某一领域的联邦强化学习应用。这些联邦强化学习方法往往包含许多独有的、只适用于各自研究领域的额外机制，并未设计出具有高通用性的联邦强化学习框架。尽管 Hu 等[31]旨在提出一个具有高通用性的联邦强化学习算法，但该算法核心并非典型的联邦强化学习算法，其是否属于联邦强化学习仍值得商榷。

针对当前联邦强化学习的研究现状，本文对相关研究工作进行提炼与抽象，继承 FedAvg 算法的设计思想，并将其与多种常见的深度强化学习算法相结合，最大程度地摒弃各种不必要的设计。最终，本文提出一种结构简单但通用性强的联邦强化学习框架，并在所提出的通用联邦强化学习框架上实验分析多种深度强化学习算法的表现，据此对框架的有效性和通用性进行验证。

2　GenFedRL 框架

本节提出一种基于联邦学习且适用于深度强化学习智能体的模型同步共享方案，并称其为 GenFedRL 框架。

2.1　深度强化学习智能体的基本结构与工作原理

本文重点研究深度强化学习智能体在联邦学习框架内的应用。深度强化学习算法可分为异策略算法和同策略算法。异策略算法可以利用过往策略或其他策略生成的经验数据来更新策略，因而需要使用经验池来存储大量经验数据。异策略算法主要包括 DQN、DDPG、SAC 等算法。同策略算法每次更新策略必须使用由当前策略生成的经验，因而只需要临时缓存来存储少量最新经验数据，并在使用完毕后重置。同策略算法主要包括 VPG、reinforce、Actor Critic、PPO 等算法。为此，联邦强化学习中的数据节点可以搭载异策略或同策略的深度强化学习算法智能体。

搭载异策略/同策略深度强化学习算法智能体的数据节点的工作原理如图 1 所示。

图 1 中，{状态,动作,奖励,下一个状态}可称为经验组合。异策略算法重复利用经验池中的经验组合，而同策略算法单次利用临时缓存中的经验组合。由于经验组合中包含了大量的隐私数据，在注重隐私的应用场景下，它们不应被轻易分享。这也暴露了分布式强化学习缺乏隐私保护的缺点。

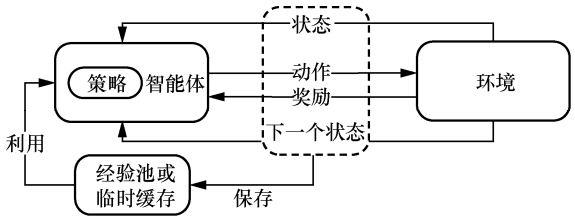


图 1　搭载异策略/同策略深度强化学习算法智能体的数据节点工作原理

2.2　GenFedRL 框架的基本结构与工作原理

本文针对具有深度强化学习算法智能体的数据节点，在传统联邦学习框架的基础上对其进行通用性改造，提出如图 2 所示的 GenFedRL 框架。

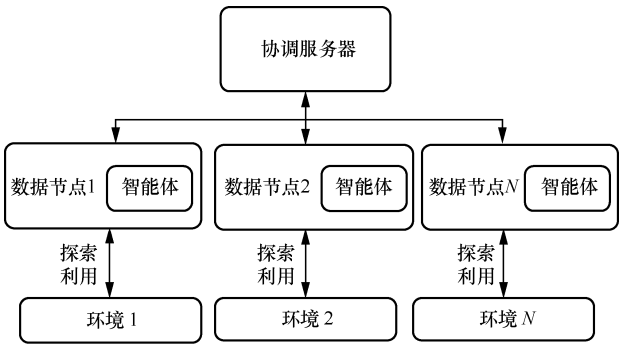


图 2　GenFedRL 框架

GenFedRL 框架包括数据节点和协调服务器，数据节点本质上是深度强化学习智能体。在 GenFedRL 中，数据节点的所有者不允许直接共享私有数据但允许共享模型。数据节点可以是执行深度强化学习算法的智能设备或智能机器人。协调服务器主要负责与数据节点的通信和模型安全聚合。

鉴于每一个数据节点可能有着截然不同的计算能力或网络连接状态（在线或离线），从而导致数据节点完成本地训练所需的时间不同，与协调服务器的通信时延也存在差异，本文提出一种基于同步并行的模型共享机制。该机制的核心主要表现为两方面：一方面，数据节点并行地进行本地训练，在本地训练结束后将本地模型参数发送到服务器，并等待接收新的全局模型；另一方面，协调服务器接收到一定数量的数据节点的模型参数后便开始模型聚合，与要求接收到所有数据节点的模型参数才进行模型聚合相比更符合实际情况，能节省模型聚合的时间。

具体而言，协调服务器将维护一个容量适中的模型缓冲区，用于保存每轮联邦强化学习过程中接收到的来自数据节点的本地模型。该模型缓冲区设

置了预期接收模型数量和最大等待时间。在训练前期，协调服务器计算最大等待时间内收集到的模型数量，估计并设置较大模型缓冲区的预期接收模型数量。随着训练的进行，协调服务器将根据实际的模型接收状况，动态调整模型缓冲区的预期接收模型数量。若模型缓冲区接收的模型数量达到预期接收模型数量且未超出最大等待时间，协调服务器将对模型缓冲区中的模型进行全局聚合并令数据节点停止训练。协调服务器将根据最近若干轮次的实际模型接收情况，动态调整预期接收模型数量。若模型缓冲区为空且超出最大等待时间，则上一轮全局模型将作为新的全局模型，并认为此时训练中可能存在故障，需要进一步检查。在生成新的全局模型后，模型缓冲区将被清空，等待下一轮联邦强化学习。通过引入最大等待时间以及动态调整预期模型接收数量，可以避免协调服务器长期处于等待状态。

GenFedRL 的联邦强化学习流程简述如下。首先，协调服务器将初始全局模型分发给在线数据节点。然后，在每一轮联邦强化学习过程中，在线数据节点使用深度强化学习智能体与环境进行交互，获得经验数据并优化自身的策略，即更新本地模型参数。在线数据节点与环境交互的过程是同步并行的。为避免通信过于频繁，在线数据节点进行一定轮次的本地训练后，其本地模型参数才被发送至协调服务器。协调服务器收集到足够而非全部的本地模型参数后，执行模型聚合算法形成全局模型，并根据接收到的模型数量动态调整模型缓冲区预期接收模型数量，完成本轮联邦强化学习。下一轮联邦强化学习的开始，协调服务器将全局模型再次发送给在线数据节点以进行本地训练。聚合−分发−训练−聚合过程迭代执行，直至全局模型达到要求或联邦强化学习的轮次达到指定数量。

3 通用联邦强化学习算法设计

在传统联邦学习 FedAvg 算法[1]的基础上，本文设计了 GeFedRL 框架中的通用联邦强化学习算法。该算法适用于具有单网络结构与多网络结构的深度强化学习智能体，由算法 1～算法 5 组成。

算法 1 协调服务器上的联邦强化学习算法

定义 M 为深度强化学习智能体包含的模型数量；深度强化学习智能体的全局模型为 $w=\{w^1,w^2,\cdots,w^M\}$，w^i 为全局模型内的第 i 个神经网络；w_r 为第 r 轮联邦强化学习轮次的全局模型；数据节点 k 的智能体的本地模型为 $w_{k_{\text{local}}}=\{w_{k_{\text{local}}}^1,w_{k_{\text{local}}}^2,\cdots,w_{k_{\text{local}}}^M\}$，$w_{k_{\text{local}}}^i$ 为数据节点 k 的智能体的本地模型内的第 i 个神经网络；共进行 N 轮联邦强化学习，定义数据节点 k 每次本地训练共执行 n_k 轮强化学习过程；B_r 为模型缓冲区，预期接收模型数量为 V_p；$\text{Wait}_{\max}$ 为最大等待时间。

1) $w_0 \to \{w_0^1,w_0^2,\cdots,w_0^M\}$
2)for 联邦强化学习轮次 r from 0 to $N-1$
3) 初始化 S_r 为在线数据节点的集合
4) 初始化 B_r 为模型缓冲区
5) 分发全局模型 w_r 给 S_r 中的数据节点
6) for 数据节点 k in S_r 并行执行
7) ModelReplace(k , w_r)
8) $\left\{w_{k_{\text{local}}},n_k\right\}=\text{DataNodeUpdate}(k)$
9) end for
10) while 模型缓冲区 B_r 未达到指定条件
11) 等待数据节点 k 回传 $\left\{w_{k_{\text{local}}},n_k\right\}$
12) $\{w_{k_{\text{local}}},n_k\}$ 被保存至模型缓冲区 B_r
13) end while
14) 令所有在线数据节点停止本地训练
15) $w_{r+1}=\text{ModelAggregate}\ (B_r)$
16) ModelBufferAdjust(B_r)
17) if w_{r+1} 达到指定条件
18) break
19) end if
20) end for

模型缓冲区 B_r 未达到指定条件是指 B_r 接收的模型数量未达到预期接收模型数量 V_p，或本轮等待时间未达到最大等待时间 $\text{Wait}_{\max}$。每一轮联邦强化学习结束时，协调服务器会重新计算本轮 V_p 的值。w_{r+1} 达到指定条件是指 w_{r+1} 的性能达到要求。

算法 2 数据节点上的模型替代算法

Model Replace(k,w_r)
1) $w_r \to \{w_r^1,w_r^2,\cdots,w_r^M\}$
2) $w_{k_{\text{local}}} \to \left\{w_{k_{\text{local}}}^1,w_{k_{\text{local}}}^2,\cdots,w_{k_{\text{local}}}^M\right\}$
3) for i from 1 to M
4) $w_{k_{\text{local}}}^i=w_r^i$
5) end for
6) $w_{k_{\text{local}}}=\left\{w_{k_{\text{local}}}^1,w_{k_{\text{local}}}^2,\cdots,w_{k_{\text{local}}}^M\right\}$

算法 2 将本地模型替换为全局模型，以保证接下来的模型更新是在全局模型的基础上进行的。算法 2 包含了对智能体中可能包含多个模型的考虑，即单个智能体中的每一个本地模型均对应其全局模型，不可混淆。

算法 3　数据节点上的模型更新算法

定义 P 为经验池；数据节点 k 每次本地训练一共进行 n_k 轮强化学习过程；D_t 为时间步 t 时探索的完成标记。更新 $w_{k_{\text{local}}}$ 的具体方式由数据节点的算法类型决定，常使用 Adam 算法[33]更新神经网络的参数。

Data Node Update (k)

1) for 本地训练轮次 I from 1 to n_k
2) 定义时间步 $t=0$
3) 初始化临时缓存 Temp
4) 定义本轮探索完成标记 Flag=False
5) while Flag==False
6) 探索完成标记 $D_t=$ False
7) 智能体从环境中观测到状态 S_t
8) 智能体根据自身策略选择动作 A_t
9) 智能体在环境中执行动作 A_t
10) 智能体将获得的奖励值 R_t
11) 智能体从环境中观测到新状态 S_{t+1}
12) if 达到探索结束条件
13) 探索完成标记 Flag = D_t = True
14) end if
15) if 算法类型为异策略
16) 经验组合 = $\{S_t, A_t, R_t, S_{t+1}, D_t\}$
17) 经验组合存储于经验池 P
18) end if
19) if 经验池 P 中经验达到一定数量
20) 从经验池 P 中随机采样训练集 B
21) 训练集 B 被用于更新 $w_{k_{\text{local}}}$
22) end if
23) if 算法类型为同策略
24) 经验组合 = $\{S_t, A_t, R_t, S_{t+1}, D_t\}$
25) 经验组合顺序存储于 Temp
26) end if
27) 时间步 $t=t+1$
28) end while
29) if 算法类型是同策略
30) 临时缓存 Temp 被用于更新 $w_{k_{\text{local}}}$
31) 重置临时缓存 Temp
32) end if
33) end for
34) $w_{k_{\text{local}}}=\left\{w_{k_{\text{local}}}^1, w_{k_{\text{local}}}^2, \cdots, w_{k_{\text{local}}}^M\right\}$
35) return $\{w_{k_{\text{local}}}, n_k\}$

算法 3 体现了异策略和同策略的深度强化学习算法进行本地训练时的差异。异策略深度强化学习算法具有经验池，主要用于保存过往经验数据以供智能体学习。当经验池已满时，智能体将自动淘汰旧的或低质量的经验数据[11]。而同策略的深度强化学习算法每次本地训练只利用本轮强化学习产生的经验数据，只需使用临时缓存。

算法 4　协调服务器上的模型聚合算法

定义实际参与聚合的模型数量为 $V_r(0<V_r\leqslant V_p)$。

Model Aggregate (B_r)

1) if B_r 为空
2) return w_r
3) end if
4) $B_r \rightarrow \left\{w_{1_{\text{local}}}, w_{2_{\text{local}}}, \cdots, w_{n_{\text{local}}}\right\}$
5) $B_r \rightarrow \{n_1, n_2, \cdots, n_k\}$
6) for i from 1 to V_r
7) $w_{i_{\text{local}}} \rightarrow \left\{w_{i_{\text{local}}}^1, w_{i_{\text{local}}}^2, \cdots, w_{i_{\text{local}}}^M\right\}$
8) end for
9) for i from 1 to M
10) $$w_{r+1}^i=\frac{n_k}{\sum_{k=1}^{V_r} n_k}\sum_{k=1}^{V_r} n_k w_{k_{\text{local}}}^i$$
11) end for
12) $w_{r+1}=\left\{w_{r+1}^1, w_{r+1}^2, \cdots, w_{r+1}^M\right\}$
13) return w_{r+1}

算法 4 在 FedAvg 算法基础上，增加了对智能体中可能包含多个模型的考虑，即单个智能体中的每一个本地模型均对应其全局模型，不可混淆。同时，算法 4 还增加了对各个客户端本地的计算能力的考虑，即进行相对多本地训练的客户端生成的模型将在聚合过程中具有更大的权重。

算法 5　协调服务器上的模型缓冲区调整算法

定义全局最大容量 $V_{\max}$，$V_{\max}$ 为一个较大的固定值，V_p 为可调整值。算法 5 第一次被调用时，预期接收模型数量 V_p 初始化为全局最大容量为 $V_{\max}$。定义 NumSet 为记录每轮次模型缓冲区共计收集到

的模型数量的集合。

Model BufferAdjust(B_r)

1) $\text{Num}_r = B_r$ 共计收集到的模型数量

2) Num_r 顺序存储于 NumSet

3) if $\text{Num}_r == V_p$

4) 预期接收模型数量 V_p 保持不变

5) else

6) 预期接收模型数量 $V_p = \text{RMean(NumSet)}$

7) end if

RMean(NumSet) 可计算 NumSet 中最近存储的若干个模型缓冲区实际收集到的模型数量的平均值，蕴含了滑动平均的思想。当模型缓冲区 B_r 接收的模型数量达到 V_p 后，实际上仍会接收到一部分本地模型，但这些模型不会参与聚合，其数量被记录，用于算法 5 调整 V_P 的值，当 V_p 被修改后，将影响算法 1 中的等待时间。

算法 1～算法 5 共同实现了通用联邦强化学习算法，使具有单网络结构与多网络结构的深度强化学习智能体均可自适应地应用于 GenFedRL。只要参与联邦强化学习的智能体的模型具有相同的网络结构，使用相同的算法，就可以参与联邦强化学习。同时，保证了每轮联邦强化学习均在可接受的时间内完成。

4 实验评估

4.1 实验环境

本文实验选取 2 种常见的强化学习测试环境 CartPole 和 Pendulum 作为 GenFedRL 的测试环境，它们由 OpenAI Gym 具体实现并封装[34]。其中，CartPole 对应于 Barto 等[35]描述的推车问题，Pendulum 则基于控制理论中的经典问题。

实验中，本文使用 CartPole-v0、CartPole-v1 和 Pendulum-v1 作为 GenFedRL 数据节点中的智能体所面对的环境。本文的仿真实验在装有 Windows11 系统的计算机上完成，机器配置为 12 代 Intel Core i512400 处理器，内存容量 32 GB，CPU 工作频率为 4.2 GHz。

4.2 实验方法

4.2.1 符号定义

实验描述中涉及的主要符号定义如表 1 所示。

一次完整的联邦强化学习过程 P 包含多轮联邦强化学习的过程 P_i，即 $P=\{P_1,P_2,\cdots,P_N\}$。每一轮 P_i 有 m 个在线数据节点参与训练，在线数据节点包含一个智能体 A_{ij}，构成了 P_i 对应的智能体集合 A_i，即 $A_i=\{A_{i1},A_{i2},\cdots,A_{im}\}$。每一个智能体 A_{ij} 对应一个环境 Env_{ij}，智能体 A_{ij} 在环境 Env_{ij} 中进行本地训练 E_{ij}。

本地训练 E_{ij} 包含 n 轮次本地训练，即 $E_{ij}=\{E_{ij1},E_{ij2},\cdots,E_{ijk},\cdots,E_{ijn}\}$。$E_{ij}$ 实质是 n 轮次强化学习训练。每一个 E_{ijk} 结束后，将得到一个得分 S_{ijk}，因此也构成了 E_{ij} 对应的得分集合 $S_{ij}=\{S_{ij1},S_{ij2},\cdots,S_{ijk},\cdots,S_{ijn}\}$。得分 S_{ijk} 的实质为一轮强化学习训练的累积奖励值，在本文实验中由环境自动计算得出。

4.2.2 主要实验参数

本文实验参考经典联邦学习[1]，共设置了 50 个数据节点，并令每轮联邦强化学习过程中只有 10%

表 1 符号定义

符号	含义
P_i	第 i 轮联邦强化学习过程
P	一次完整的联邦强化学习过程 $P=\{P_1,P_2,\cdots,P_N\}$，N 为最大联邦强化学习轮次
A_{ij}	第 i 轮参与训练的智能体集合中的第 j 个智能体
$\boldsymbol{A}_i$	第 i 轮联邦强化学习过程中参与训练的智能体集合，$A_i=\{A_{i1},A_{i2},\cdots,A_{im}\}$，$m$ 为第 i 轮联邦强化学习过程中参与训练的智能体总数
Env_{ij}	第 i 轮参与训练的智能体集合中的第 j 个智能体所对应的环境
Env_i	第 i 轮联邦强化学习过程中参与训练的智能体所对应的环境集合 $\text{Env}_i=\{\text{Env}_{i1},\text{Env}_{i2},\cdots,\text{Env}_{im}\}$，$m$ 为第 i 轮联邦强化学习过程中参与训练的智能体总数
E_{ijk}	第 i 轮参与训练的第 j 个智能体的第 k 轮本地训练
$\boldsymbol{E}_{ij}$	第 i 轮参与训练的第 j 个智能体在环境 Env_{ij} 中的本地训练集合，$E_{ij}=\{E_{ij1},E_{ij2},\cdots,E_{ijk},\cdots,E_{ijn}\}$，$n$ 为最大本地训练轮次
S_{ijk}	E_{ijk} 对应的得分
$\boldsymbol{S}_{ij}$	本地训练 $\boldsymbol{E}_{ij}$ 对应的得分集合，$\boldsymbol{S}_{ij}=\{S_{ij1},S_{ij2},\cdots,S_{ijk},\cdots,S_{ijn}\}$，$n$ 为最大本地训练轮次

的随机数据节点作为在线数据节点参与训练以模拟恶劣通信环境。此外，为提高实验可控性，保证实验结果的稳定性与可重复性，实验过程中，模型接收缓冲区的预期接收模型数量 V_p 恒等于 10%的数据节点数量，且模型接收缓冲区在每轮联邦强化学习中总能接收到足够多的模型。因此，模型接收缓冲区的预期接收模型数量 V_p 未进行动态调整，从而避免引入更多的随机变量。设定联邦强化学习通信 N=100（SAC 算法中 N=20），并令所有智能体 A_{ij} 的最大本地训练轮次 n 相同，在实验中分别尝试 n 的不同取值。

另外，实验过程中还考虑了 P_i 中每个 A_{ij} 面对的环境初始状态是否相同这一因素。为方便描述，本文将 P_i 中每个 A_{ij} 面对的环境的初始状态相同的情形称为 S1，P_i 中每个 A_{ij} 面对的环境的初始状态各不相同的情形称为 S2。S2 比 S1 更具挑战性。

4.3　实验评价指标

在 P_i 中，A_{ij} 与环境的每次交互都会获得奖励值。本文将 A_{ij} 通过 $\boldsymbol{E}_{ij}$ 所获得奖励值的平均值作为 P_i 的评价指标，即 $S_{P_i}=\frac{1}{m}\sum_{j=1}^{m}\mathrm{Mean}(S_{ij})$。经统计可得总得分集合 $S_P=\{S_{P_1},S_{P_2},\cdots,S_{P_{100}}\}$。$S_P$ 经过移动平均处理后用于图表绘制以便观察分析 GenFedRL 的性能表现，得分越高，表示性能越好。

4.4　实验结果与分析

本文分别针对具有单网络、双网络以及多网络结构的深度强化学习智能体在 GenFedRL 下进行模拟实验，考虑 10～35 范围内的最大本地训练轮次 n，以及各个智能体对应的初始环境状态是否相同 2 个因素，并对实验结果进行逐一记录和分析。

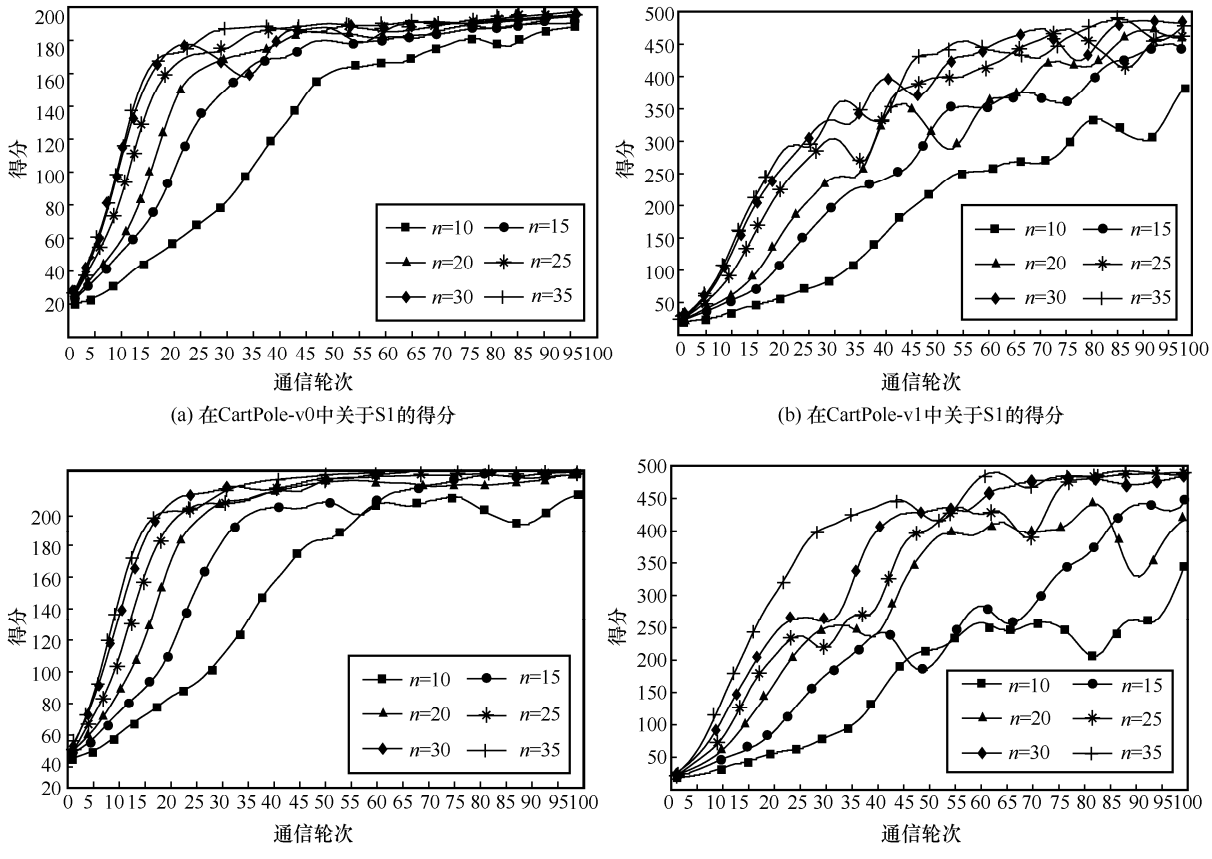
4.4.1　GenFedRL 下单网络结构深度强化学习智能体

本节实验在 CartPole-v0、CartPole-v1 环境中让 GenFedRL 执行具有单网络结构的深度强化学习算法 reinforce[23]，以评估具有单网络结构的深度强化学习智能体应用于 GenFedRL 的可行性。

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[10,35]$ 时，在 CartPole-v0、CartPole-v1 中 GenFedRL 的得分曲线如图 3 所示。

4.4.2　GenFedRL 下双网络结构深度强化学习智能体

本节实验在 CartPole-v0、CartPole-v1 环境中让 GenFedRL 执行具有双网络结构的深度强化学



(a) 在CartPole-v0中关于S1的得分　(b) 在CartPole-v1中关于S1的得分

(c) 在CartPole-v0中关于S2的得分　(d) 在CartPole-v1中关于S2的得分

图 3　搭载 reinforce 智能体的 GenFedRL 的得分

习算法如 DQN 算法[16]、Actor Critic 算法[25]和 PPO 算法[26]，以评估具有双网络结构的深度强化学习智能体应用于 GenFedRL 的可行性。

1) DQN 算法

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[10,35]$ 时，在 CartPole-v0、CartPole-v1 中 GenFedRL 的得分曲线如图 4 所示。

2) Actor Critic 算法

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[10,35]$ 时，在 CartPole-v0、CartPole-v1 中 GenFedRL 的得分曲线如图 5 所示。

3) PPO 算法

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[10,35]$ 时，在 CartPole-v0、CartPole-v1 中 GenFedRL 的得分曲线如图 6 所示。

4.4.3 GenFedRL 下多网络结构深度强化学习智能体

本节实验在 Pendulum-v1 环境中让 GenFedRL 执行具有多网络结构的深度强化学习算法 DDPG[27]和 SAC[28]，以评估具有多网络结构的深度强化学习算法的智能体应用于 GenFedRL 的可行性。其中，DDPG 算法具有 4 个神经网络，SAC 算法具有 5 个神经网络。

1) DDPG 算法

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[5,35]$ 时，在 Pendulum-v1 中 GenFedRL 的得分曲线如图 7 所示。

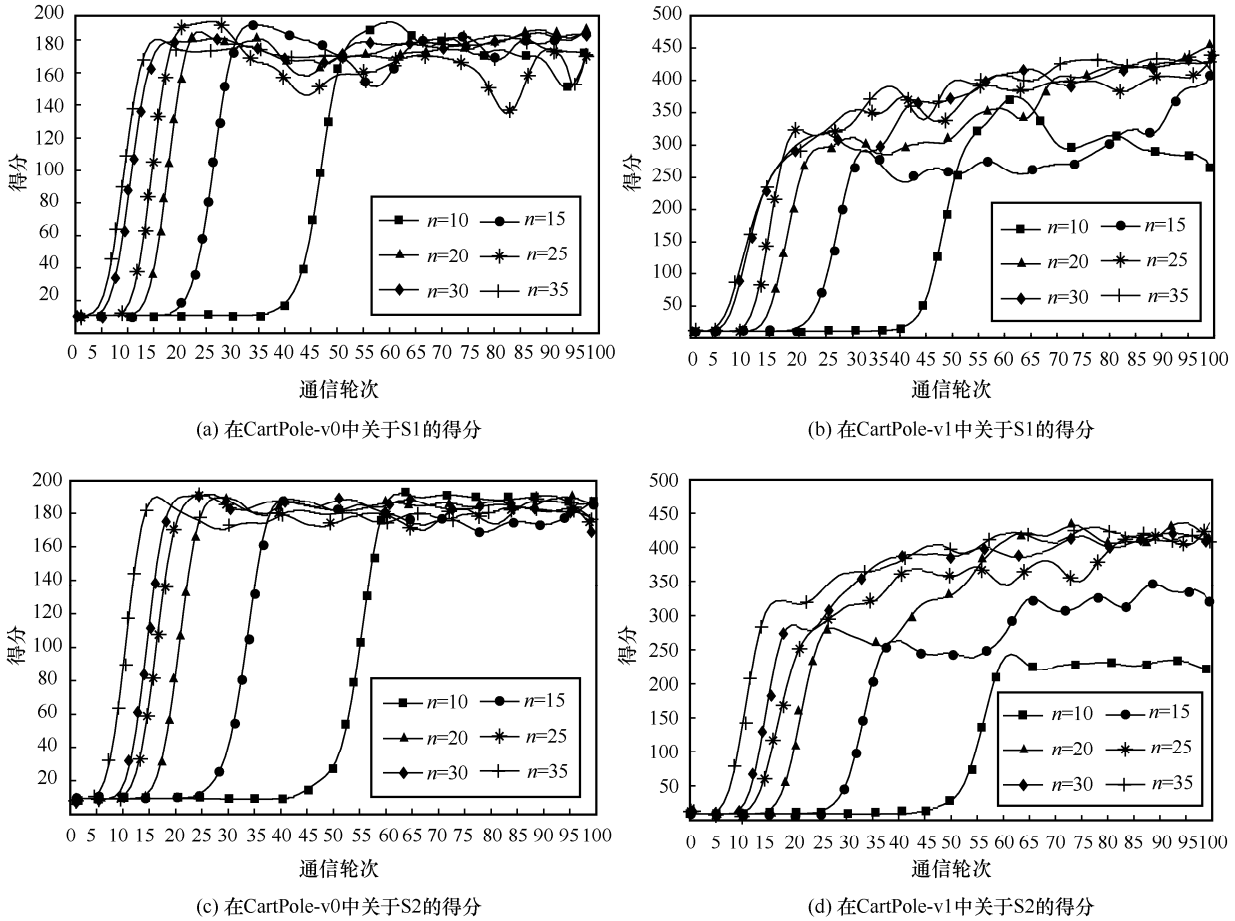
2) SAC 算法

当 P_i 中 A_{ij} 的 E_{ij} 的最大训练轮次 $n\in[5,35]$ 时，在 Pendulum-v1 中 GenFedRL 的得分曲线如图 8 所示。

4.4.4 实验分析

实验在 CartPole-v0、CartPole-v1 和 Pendulum-v1 环境中测试了搭载不同智能体的 GenFedRL 的表现。

图 3～图 8 表明，在 CartPole-v0 与 CartPole-v1 中，在情形 S1 和 S2 下，GenFedRL 的得分均随着联邦强化学习通信轮次的增加而逐步上升，意味



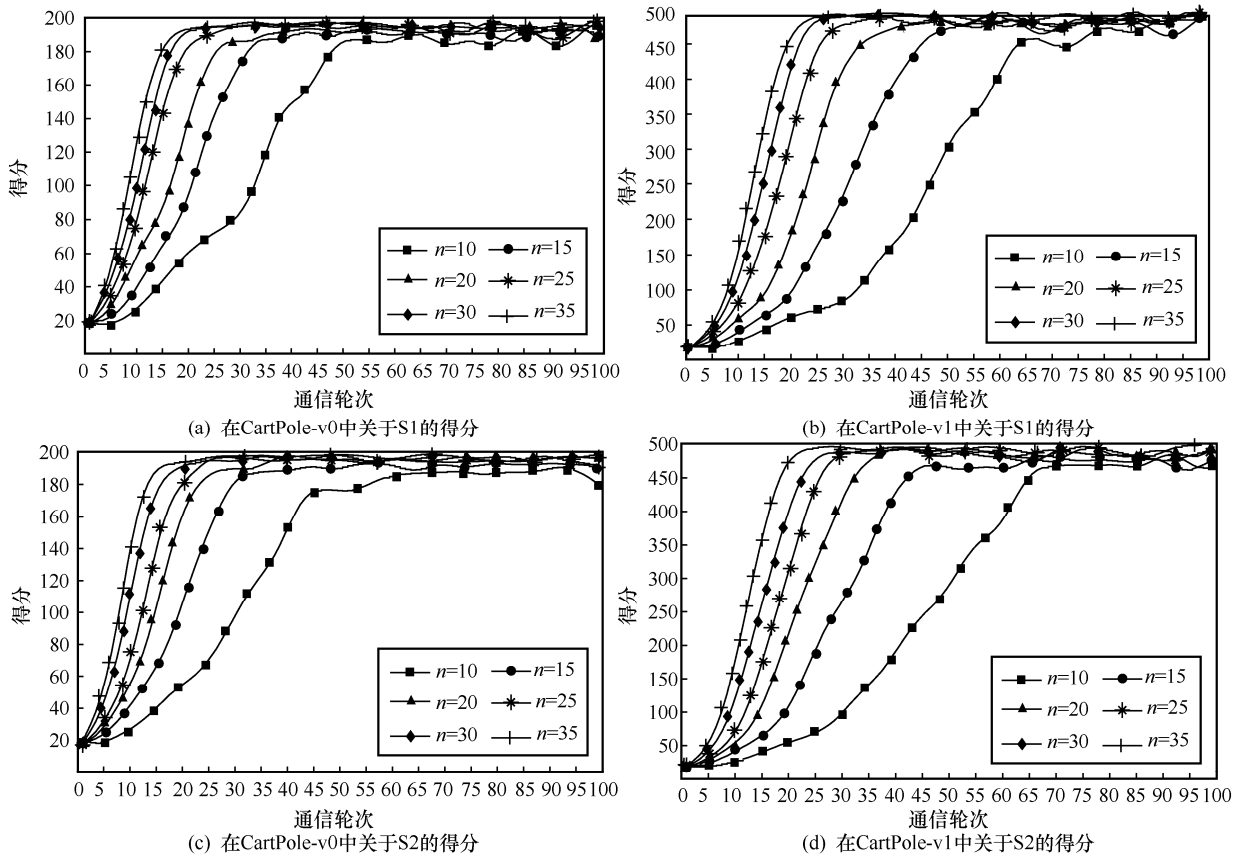
(a) 在CartPole-v0中关于S1的得分

(b) 在CartPole-v1中关于S1的得分

(c) 在CartPole-v0中关于S2的得分

(d) 在CartPole-v1中关于S2的得分

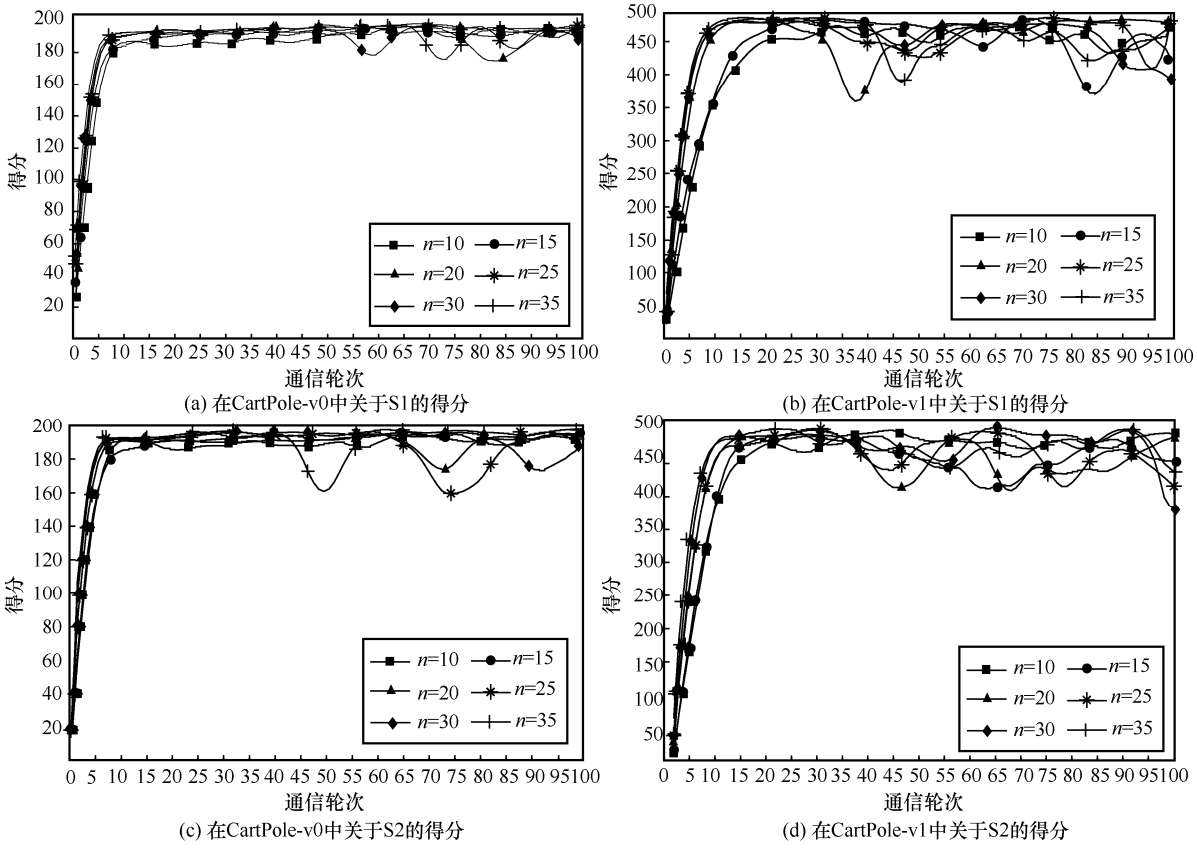
图 4 搭载 DQN 智能体的 GenFedRL 的得分



(a) 在CartPole-v0中关于S1的得分　(b) 在CartPole-v1中关于S1的得分

(c) 在CartPole-v0中关于S2的得分　(d) 在CartPole-v1中关于S2的得分

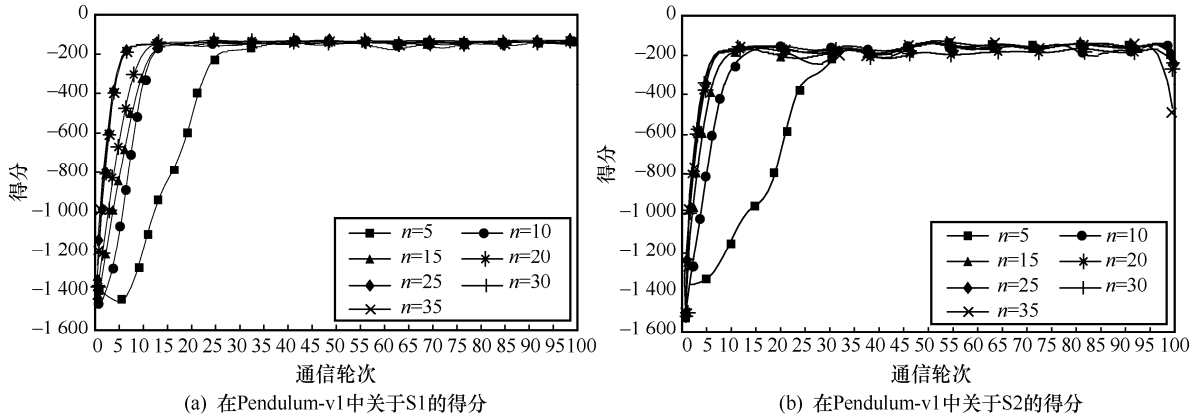
图 5　搭载 Actor Critic 智能体的 GenFedRL 的得分



(a) 在CartPole-v0中关于S1的得分　(b) 在CartPole-v1中关于S1的得分

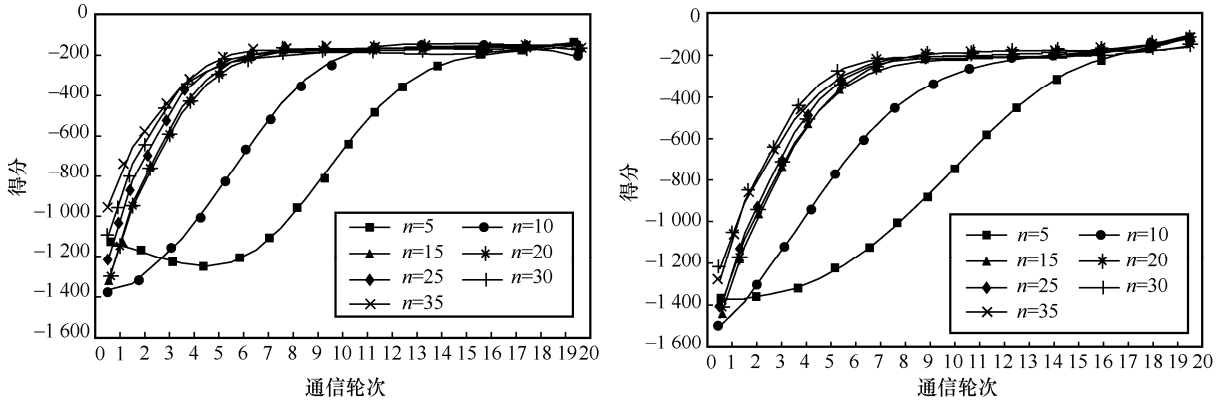
(c) 在CartPole-v0中关于S2的得分　(d) 在CartPole-v1中关于S2的得分

图 6　搭载 PPO 智能体的 GenFedRL 的得分



(a) 在Pendulum-v1中关于S1的得分　　(b) 在Pendulum-v1中关于S2的得分

图 7　搭载 DDPG 智能体的 GenFedRL 的得分



(a) 在Pendulum-v1中关于S1的得分　　(b) 在Pendulum-v1中关于S2的得分

图 8　搭载 SAC 智能体的 GenFedRL 的得分

着 GenFedRL 中的智能体逐渐习得可获得最大奖励值的策略。对于该现象的一种合理解释如下：这些运用了深度强化学习算法的智能体具有从旧本地策略中，通过探索环境而学习到更优的新本地策略的能力。GenFedRL 进行分发−聚合−分发的迭代过程本质上是以间接方式从旧的本地策略产生新的本地策略。因此，当这些新的本地策略被聚合成新的全局策略后，新的全局策略优于旧的全局策略是可预期的。

为方便比较和观察，设得分 190、400、−200 分别为 CartPole-v0、CartPole-v1 以及 Pendulum-v1 环境下的比较基准。根据图 3～图 8，在情形 S1 和 S2 下，搭载不同智能体的 GenFedRL 在 CartPole-v0、CartPole-v1 与 Pendulum-v1 中首次达到对应基准所需的通信轮次如表 2～表 4 所示。表 2～表 4 中，括号外数据和括号内数据分别表示 GenFedRL 在情形 S1 以及情形 S2 下首次达到某一等级时所需的通信轮次，—表示在情形 S1 下 GenFedRL 无法达到对应的等级，(—)表示在情形 S2 下 GenFedRL 无法达到对应的等级。

表 2 和表 3 表明，GenFedRL 在 CartPole-v0 和 CartPole-v1 中关于情形 S1 和 S2 的表现总体差异较小。当 n 较低时，S2 的困难性体现得更明显。但随着 n 的增加，GenFedRL 中的智能体逐渐克服这种不利影响甚至有更好的表现。表 4 反映了 GenFenRL 在 Pendulum-v1 环境中的类似现象。

表 2～表 4 表明，搭载不同强化学习算法智能体的 GenFedRL 在相同环境中的表现不同，这可能是具有不同强化学习算法的智能体本身的算法特性与超参数差异导致的。同时，表 2～表 4 显示，搭载不同强化学习算法智能体的 GenFedRL 在相同环境中均表现出 E_{ij} 的最大本地训练轮次 n 越大，GenFedRL 首次达到对应基准所需要的通信轮次越少的特点。对于该现象的一种合理解释如下：E_{ij} 可视为普通的强化学习过程，智能体的得分会随着智能体的训练轮次的增加而提高，这使智能体更可

表 2　GenFedRL 在 CartPole-v0 中首次达到对应基准所需通信轮次

算法	n=10	n=15	n=20	n=25	n=30	n=35
reinforce	91 (—)	73(68)	45(44)	36(47)	42(31)	27(33)
DQN	57 (61)	32 (—)	99 (93)	20 (23)	99 (21)	—(16)
Actor Critic	83 (90)	40 (42)	35 (33)	24 (23)	19 (19)	17 (16)
PPO	49 (10)	19 (17)	7 (8)	10 (7)	8 (7)	6 (6)

表 3　GenFedRL 在 CartPole-v1 中首次达到对应基准所需通信轮次

算法	n=10	n=15	n=20	n=25	n=30	n=35
reinforce	—(—)	81 (84)	69 (60)	55 (49)	50 (40)	44 (28)
DQN	—(—)	99 (—)	69 (59)	87 (81)	57 (70)	50 (45)
Actor Critic	59 (61)	40 (38)	29 (29)	23 (23)	19 (20)	17 (16)
PPO	13 (10)	11 (9)	7 (7)	6 (7)	5 (6)	5 (5)

表 4　GenFedRL 在 Pendulum-v1 中首次达到对应基准所需通信轮次

算法	n=5	n=10	n=15	n=20	n=25	n=30	n=35
DDPG	26 (31)	12 (12)	12 (8)	10 (9)	6 (7)	6 (7)	6 (7)
SAC	16 (17)	10 (14)	7 (9)	6 (9)	6 (16)	5 (11)	8 (15)

能从旧的全局策略中习得更优的、新的本地策略，进而聚合为更优的、新的全局策略。

总体而言，搭载具有单网络结构与多网络结构的强化学习智能体的 GenFedRL 可以通过适当增加最大训练轮次 n 来应对智能体所面对的环境初始状态不同的情况。这说明即使在通信恶劣的情况下，具有单网络结构的强化学习算法的智能体应用于 GenFedRL 是可行的。

5　结束语

本文提出一种联邦强化学习框架 GenFedRL，并设计了通用联邦强化学习算法，使具有单网络结构、多网络结构的深度强化学习算法的智能体均可自适应地应用于 GenFedRL。仿真实验表明 GenFedRL 具有高通用性，可以在保护隐私的同时实现安全数据共享。此外，GenFedRL 还能够有效应对大量数据节点频繁离线的恶劣通信环境。

然而，继承自联邦学习的最初定义，虽然本文希望联邦强化学习的参与方各自面临相同或相似的任务环境，但这也限制了框架的适用范围。在现实的联邦强化学习场景下，可能存在联邦强化学习的参与方各自面临截然不同任务环境的情形。这对联邦强化学习的模型共享机制提出了更高的要求。

从隐私保护的角度而言，向 GenFedRL 中增加额外的、成熟的安全机制以进一步提升其隐私保护能力也值得探讨，如安全多方计算[36]、同态加密技术[37-38]、差分隐私技术[39-40]。

此外，在现实的工程运用中，联邦强化学习框架的顺利运作离不开模型缓冲区的良好实现。本文设计了一个动态调整模型缓冲区的预期接收模型数量 V_p 的机制，但出于提高实验可控性，保证实验结果的稳定性与可重复性的考虑，本文实验中并未充分验证这一机制。因此，在未来的研究工作中，设计并验证更成熟的模型缓冲区调整机制也是值得研究的工作。

参考文献：

[1] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv Preprint, arXiv: 1602.05629, 2016.

[2] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of GO without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[3] HESSEL M, SOYER H, ESPEHOLT L, et al. Multi-task deep reinforcement learning with PopArt[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2019: 3796-3803.

[4] ANDRYCHOWICZ O M, BAKER B, CHOCIEJ M, et al. Learning dexterous in-hand manipulation[J]. International Journal of Robotics Research, 2020, 39(1): 3-20.

[5] HAO J Y, YANG T P, TANG H Y, et al. Exploration in deep reinforcement learning: from single-agent to multiagent domain[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023: doi.org/10.1109/TNNLS.2023.3236361.

[6] 杨强, 刘洋, 程勇, 等. 联邦学习[M]. 北京: 电子工业出版社, 2020.
YANG Q, LIU Y, CHENG Y, et al. Federated learning[M]. Beijing: Publishing House of Electronics Industry, 2020.

[7] ESPEHOLT L, SOYER H, MUNOS R, et al. IMPALA: scalable distributed deep-RL with importance weighted actor-learner architectures[C]//Proceedings of International Conference on Machine Learning. New York: ACM Press, 2018: 1407-1416.

[8] KAPTUROWSKI S, OSTROVSKI G, QUAN J, et al. Recurrent experience replay in distributed reinforcement learning[C]//Proceedings of International Conference on Learning Representations. [s.l.]: Open Review, 2019: 1-19.

[9] ESPEHOLT L, MARINIER R, STANCZYK P, et al. SEED RL: scalable and efficient deep-RL with accelerated central inference[J]. arXiv Preprint, arXiv: 1910.06591, 2019.

[10] KRISHNAN S, LAM M, CHITLANGIA S, et al. QuaRL: quantization for fast and environmentally sustainable reinforcement learning[J]. arXiv Preprint, arXiv: 1910.01055, 2019.

[11] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[J]. arXiv Preprint, arXiv: 1511.05952, 2015.

[12] 王志勤, 江甲沫, 刘沛西, 等. 6G 联邦边缘学习新范式: 基于任务导向的资源管理策略[J]. 通信学报, 2022, 43(6): 16-27.
WANG Z Q, JIANG J M, LIU P X, et al. New design paradigm for federated edge learning towards 6G:task-oriented resource management strategies[J]. Journal on Communications, 2022, 43(6): 16-27.

[13] ZHOU Z, TIAN Y L, XIONG J B, et al. Blockchain-enabled secure and trusted federated data sharing in IIoT[J]. IEEE Transactions on Industrial Informatics, 2023, 19(5): 6669-6681.

[14] 贺文晨, 郭少勇, 邱雪松, 等. 基于 DRL 的联邦学习节点选择方法[J]. 通信学报, 2021, 42(6): 62-71.
HE W C, GUO S Y, QIU X S, et al. Node selection method in federated learning based on deep reinforcement learning[J]. Journal on Communications, 2021, 42(6): 62-71.

[15] MIAO Q, LIN H, WANG X, et al. Federated deep reinforcement learning based secure data sharing for Internet of things[J]. Computer Networks, 2021, 197: 108327.

[16] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[17] TESAURO G. Temporal difference learning and TD-Gammon[J]. Communications of the ACM, 1995, 38(3): 58-68.

[18] NADIGER C, KUMAR A, ABDELHAK S. Federated reinforcement learning for fast personalization[C]//Proceedings of 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering. Piscataway: IEEE Press, 2019: 123-127.

[19] LIU B Y, WANG L J, LIU M. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 4555-4562.

[20] MOWLA N I, TRAN N H, DOH I, et al. AFRL: adaptive federated reinforcement learning for intelligent jamming defense in FANET[J]. Journal of Communications and Networks, 2020, 22(3): 244-258.

[21] WANG X F, WANG C Y, LI X H, et al. Federated deep reinforcement learning for Internet of things with decentralized cooperative edge caching[J]. IEEE Internet of Things Journal, 2020, 7(10): 9441-9455.

[22] HASSELT H V, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2016: 2094-2100.

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

[24] SUTTON R S, MCALLESTER D, SINGH S, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Proceedings of the 12th International Conference on Neural Information Processing Systems. New York: ACM, 1999: 1057-1063.

[25] KONDA V, TSITSIKLIS J. Actor-critic algorithms[J]. Advances in Neural Information Processing Systems, 1999, 12: 1008-1014.

[26] ZHAO G Q, XU J M, LIU A D, et al. Research on proximal policy optimization algorithm based on N-step update[C]//Proceedings of International Conference on Communications, Information System and Computer Engineering. Piscataway: IEEE Press, 2021: 854-857.

[27] SEYED M S M, BAGHI V, MIANDOAB E M, et al. Duplicated replay buffer for asynchronous deep deterministic policy gradient[C]//Proceedings of the 26th International Computer Conference, Computer Society of Iran. Piscataway: IEEE Press, 2021: 1-6.

[28] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//2018 International Conference on Machine Learning. New York: ACM Press, 2018: 1861-1870.

[29] LIM H K, KIM J B, HEO J S, et al. Federated reinforcement learning for training control policies on multiple IoT devices[J]. Sensors, 2020, 20(5): 1359.

[30] YOO S, LEE W. Federated reinforcement learning based AANs with LEO satellites and UAVs[J]. Sensors, 2021, 21(23): 8111.

[31] HU Y Q, HUA Y, LIU W Y, et al. Reward shaping based federated reinforcement learning[J]. IEEE Access, 2021, 9: 67259-67267.

[32] NG A Y, HARADA D, RUSSELL S J. Policy invariance under reward transformations: theory and application to reward shaping[C]//Proceedings of the 16th International Conference on Machine Learning. New York: ACM Press, 1999: 278-287.

[33] KINGMA D P, BA J. Adam: a method for stochastic optimization[J]. arXiv Preprint, arXiv: 1412.6980, 2014.

[34] LE G T, MARJOU X, LEMLOUMA T, et al. A multi-agent OpenAI gym environment for telecom providers cooperation[C]//Proceedings

of the 24th Conference on Innovation in Clouds, Internet and Networks and Workshops. Piscataway: IEEE Press, 2021: 28-32.

[35] BARTO A G, SUTTON R S, ANDERSON C W. Neuronlike adaptive elements that can solve difficult learning control problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1983, SMC-13(5): 834-846.

[36] MOHASSEL P, ZHANG Y P. SecureML: a system for scalable privacy-preserving machine learning[C]//Proceedings of IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2017: 19-38.

[37] ACAR A, AKSU H, ULUAGAC A S, et al. A survey on homomorphic encryption schemes: theory and implementation[J]. ACM Computing Surveys, 2018, 51(4): 1-35.

[38] EL-YAHYAOUI A, ECH-CHERIF E K M D. A verifiable fully homomorphic encryption scheme for cloud computing security[J]. Technologies, 2019, 7(1):21.

[39] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C]//Proceedings of Third Theory of Cryptography Conference. Berlin: Springer, 2006: 265-284.

[40] DWORK C, FELDMAN V, HARDT M, et al. Preserving statistical validity in adaptive data analysis[C]//Proceedings of the 47th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 2015: 117-126.

[作者简介]

金彪（1985− ），男，安徽六安人，博士，福建师范大学副教授、硕士生导师，主要研究方向为信息安全、隐私保护等。

李逸康（1998− ），男，广东广州人，福建师范大学硕士生，主要研究方向为联邦学习与深度强化学习的交叉应用。

姚志强（1967− ），男，福建莆田人，博士，福建师范大学教授、博士生导师，主要研究方向为信息安全、隐私保护等。

陈瑜霖（1996− ），男，福建泉州人，福建师范大学硕士生，主要研究方向为深度学习技术。

熊金波（1981− ），男，湖南益阳人，博士，福建师范大学教授、博士生导师，主要研究方向为安全深度学习、移动智群感知、隐私保护。