异构联邦双向知识蒸馏传递半监督调制类型识别

齐佩汉，丁渊磊，尹凯，徐佳波，李赞
（西安电子科技大学通信工程学院，陕西 西安 710071）

摘 要：新一代移动通信系统的规模化部署和快速发展支撑巨量多样物联网设备的广泛应用。然而物联网设备的分布式应用导致的私有数据迥异和本地处理模型差别，严重制约全局智能模型的聚合能力。因此，针对认知物联网中智能调制类型识别面临的数据异质、模型异构和标记不足等问题，提出异构联邦双向知识蒸馏传递半监督调制类型识别算法，通过变分自编码器在云端生成公开伪数据集支持上行全局知识蒸馏，自适应共享至本地辅助下行异质知识蒸馏，并在蒸馏过程中内嵌半监督算法。仿真结果表明，在通信信号处理领域中，所提算法的有效性和适用性优于现有的联邦学习算法。

关键词：联邦半监督学习；模型异构；变分自编码器；知识蒸馏
中图分类号：TN92
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023191

Heterogeneous federated bidirectional knowledge distillation transfer semi-supervised modulation recognition

QI Peihan, DING Yuanlei, YIN Kai, XU Jiabo, LI Zan
School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

Abstract: The large-scale deployment and rapid development of the new generation mobile communication system underpin the widespread application of a massive and diverse range of Internet of things (IoT) devices. However, the distributed application of IoT devices results to significant disparities in private data and substantial heterogeneity in local processing models, which severely limits the aggregation capability of global intelligent model. Therefore, to tackle the challenges of data heterogeneity, model heterogeneity, and insufficient labeling faced by intelligent modulation recognition in cognitive IoT, an algorithm was proposed for heterogeneous federated bidirectional semi-supervised modulation recognition, which incorporated bidirectional knowledge distillation. In the proposed algorithm, a public pseudo dataset was generated by variational autoencoder in the cloud for supporting uplink global knowledge distillation, and adaptively sharing to the local devices for downlink heterogeneous knowledge distillation, while integrating a semi-supervised algorithm within the distillation process. The simulation results indicate that the proposed algorithm outperforms current federated learning algorithms in terms of effectiveness and applicability in the field of communication signal processing.

Keywords: federated semi-supervised learning, model heterogeneity, variational autoencoder, knowledge distillation

0 引言

自动调制类型分类（AMC, automatic modulation classification）[1-3]是目前各种通信系统中最基本的技术之一。在通信过程中，为实现最大可能的传输速率，自适应调制方案被广泛应用，以充分利用时变的信道和频谱资源。特定的调制方案由合作通信中的发射机和接收机通过网络协议共享，其代价

收稿日期：2023−06−14；**修回日期：**2023−09−05
通信作者：丁渊磊，dyl85527077@163.com
基金项目：国家自然科学基金资助项目（No.62171334, No.61971337, No.61825104）
Foundation Item: The National Natural Science Foundation of China (No.62171334, No.61971337, No.61825104)

是增加额外的协议开销。然而，当接收机具备调制类型的自主识别能力时，可以消除这种开销。此外，在电子战中需要拦截敌方信号和甄别干扰信号，通过 AMC 可以及时破解敌方信息、保障己方后续通信。因此，AMC 在实现自动接收机配置、干扰抑制和频谱管控等方面都具有重要意义。

目前，基于深度学习（DL, deep learning）的 AMC（DL-AMC）已被证明是非常有效的。然而，基于 DL 的大部分算法都依赖于大量的数据来进行集中学习，DL-AMC 也不例外。DL-AMC 通过上传不同本地设备（如物联网设备）上的调制信号对智能模型进行集中训练，以获得更优的性能。但是，调制信号通常包含用户的信息，如果这些调制信号在上传过程中被截获，可能会导致用户隐私的泄露。此外，如果本地设备仅利用各自私有的调制信号进行训练，那么得到的 AMC 模型通常具有较弱的分类能力。近年来，基于分布式联邦学习（FL，federated learning）[4]的 AMC（FL-AMC）因能够保护用户隐私、不需要集中数据的突出优势受到研究者的广泛青睐。但是，现实物联网设备应用场景的多样性导致获得的调制信号数据分布极难一致，从而产生了数据异质问题[5-7]。其次，由于设备自身计算、处理能力存在差异，不同设备需定制大小合适的智能模型，进而引发了模型异构[8-10]问题。此外，现实场景中调制信号样本标记不足的问题导致完全监督式的 FL-AMC 性能不佳。这些问题严重制约了标准联邦范式下 FL-AMC 在现实认知物联网场景中的应用。

Shi 等[11]通过对标准的联邦平均算法进行部署，实现了 FL-AMC 场景下 11 种调制类型的分类，但并未考虑数据异质和模型异构的情况。Shi 等[12]提出了具有差分隐私的 FL-AMC 算法，进一步保护了调制信号所包含的用户信息，但同样没有考虑数据异质和模型异构问题。Wang 等[13]提出了噪声变化和本地设备所持信号调制类型不平衡条件下的 FL-AMC 算法，考虑了程度较轻的数据异质情况，但并未涉及模型异构。Qi 等[14]提出了一种基于联邦增量的 AMC 算法，通过知识蒸馏（KD, knowledge distillation）[15]来减轻本地模型在优化过程中偏离全局模型的程度，以缓解数据异质问题，但同样未考虑模型异构的情况。FedMD[16]是一种经典的模型异构 FL 算法，但是该算法只能协同本地模型却无法获得全局模型。此外，现实 FL 场景中不仅存在数据异质、模型异构问题，还存在样本标记不足的问题。为此，Jeong 等[17]定义了具体的联邦半监督场景并给出了可实现的 FedMatch 算法，但并未涉及数据异质和模型异构问题。Zhong 等[18]探讨了数据异质、模型异构、样本标记不足场景下的异构联邦半监督学习场景，并提出了 Semi-HFL 算法。表 1 总结了通信信号处理领域和计算机视觉领域中现有的几种联邦学习算法。从表 1 可以看到，在通信信号处理领域中，关于异构联邦半监督调制类型识别的研究相对较少。

鉴于此，针对现实场景中数据异质、模型异构、样本标记不足的问题，本文提出一种异构联邦半监督调制类型识别 BKD-FSSL（bidirectional

表 1　　现有联邦学习算法概述

研究领域	联邦算法	适用的联邦学习场景				创新点
		数据异质	模型异构	完全监督	半监督	
通信信号处理领域	文献[11]	×	×	✓	×	联邦平均算法应用于 FL-AMC 场景，实现 11 种调制类型的分类
	文献[12]	×	×	✓	×	提出 FL-AMC 场景下的差分隐私技术，加强对调制信号数据的保护
	文献[13]	✓	×	✓	×	提出平衡交叉熵损失，实现节点上调制类型不平衡条件下的分类
	文献[14]	✓	×	✓	×	提出联邦增量学习，通过 KD 缓解数据异质，实现调制类型的分类
	本文所提算法	✓	✓	✓	✓	通过 BKD 实现数据异质、模型异构同时存在下的调制类型分类
计算机视觉领域	文献[16]	✓	✓	✓	×	通过对异构模型进行 KD 的方式来等效标准联邦方式下的聚合过程
	文献[17]	×	×	✓	✓	提出模型参数分解方法，平衡标记样本和无标记样本对模型的影响
	文献[18]	✓	✓	✓	✓	通过级联节点上的异构模型来组成全局大模型，解决模型异构问题
	文献[22]	✓	×	✓	✓	通过联邦迁移学习缓解数据异质问题

knowledge distillation for federated semi-supervised learning）算法，利用变分自编码器（VAE, variational auto encoder）[19]在云端生成公开伪样本集作为全局模型和本地模型之间进行参数交互的媒介，通过双向知识蒸馏（BKD, bidirectional knowledge distillation）机制实现协同训练，并在知识蒸馏过程中内嵌基于表示学习[20]和伪标签技术[21]设计的半监督算法。

本文主要的研究工作如下。

1) 建立用于调制信号合成的变分自编码器，构建用于协同训练的公开伪信号样本集。

2) 设计云端和本地设备间互为“师生”的双向知识蒸馏机制，实现协同训练。其中，“本地−云端”的上行蒸馏过程以“多教师−单学生”的形式将不同异构模型的异质知识融合并上传至云端，辅助全局模型的学习；“云端−本地”的下行蒸馏过程则将全局模型的知识以“单教师−多学生”的形式反馈给本地异构模型，提升模型对异质数据的识别能力。

3) 设计内嵌半监督学习模块，仅利用少量有标记样本和大量无标记样本实现异构联邦半监督调制类型识别。

4) 通过与多种现行算法的仿真结果对比，本文所提的 BKD-FSSL 算法在数据异质和模型异构同时存在的场景中具备更出色的信号调制类型识别能力和识别精度。

1　问题定义

在现实边缘物联网系统中，不同物联网设备采集的调制信号数据极难保证分布一致，同时部署的物联网设备也存在着很大的差异，很难保证拥有相近的计算、存储等性能，这意味着不同物联网设备将自主定制合适的本地模型。此外，监督式的联邦学习框架必然带来训练成本的增加，如何平衡训练成本和模型性能之间的效益是在现实场景中部署联邦框架时必须考虑的问题。在现实场景中，对于参与训练的物联网设备来说，无标记的调制信号获取相对容易，而带有专家知识的有标记调制信号却极难获取。因此，针对现实场景中存在的数据异质、模型异构和样本标记不足的问题，本文提出一种异构联邦半监督调制类型识别算法，其学习场景如图 1 所示。

在所提异构联邦半监督场景中，本文不仅考虑了模型异构的情况，还考虑了不同物联网设备所持数据存在极端异质的情况，例如，设备 a 持有类型为 A 的数据，设备 b 持有类型为 B 的数据，且 A 与 B 相异。因此，设备 a 与设备 b 上的数据分布差别会很大，而所提的联邦场景要求设备 a 、b 通过协同训练具备识别全局类别 $\{A,B\}$ 的能力，即设备存在等效增量学习的过程。对于设备 a 而言，类型 B 即缺失的类别。同理，对于设备 b 而言，类型 A 即缺失的类别。

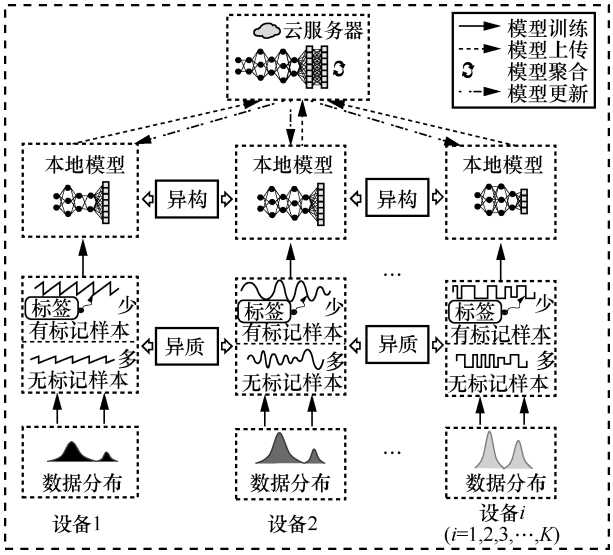


图 1　所提异构联邦半监督学习场景

本文所提的异构联邦半监督调制类型识别场景定义如下：假设存在 K 个参与训练的物联网设备，各自持有无标记私有调制信号集 D_U^k 、有标记私有调制信号集 D_L^k 和定制的本地异构模型 $f_k(\boldsymbol{\theta}_k), k=1,2,\cdots,K$ 。此外，该场景还具备以下条件：本地设备 k 上的有标记样本 $\boldsymbol{x}_i^k \in D_L^k$ 对应真实标签 $y_i^k \in C_k$ 以及经过协同训练后的本地模型对无标记样本 $\boldsymbol{x}_j^k \in D_U^k$ 的预测值为 $f_k(\boldsymbol{x}_j^k;\boldsymbol{\theta}_k) \in C_k$ ，其中， C_k 表示设备 k 持有的私有类别，且满足 $C_i \neq C_j, i \neq j$ 。依据上述条件，经过协同训练后的本地异构模型 $\left\{f_k(\boldsymbol{\theta}_k)\right\}_{k=1}^K$ 需在全局测试集 D_{glb} 上具备识别能力，其中 D_{glb} 对应的调制类别集为全局类别 $C_{\text{glb}} = \bigcup_{k=1}^K C_k$ ，而中央服务器持有的全局模型 $f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}})$ 需要在不直接接触私有调制信号数据的前提下通过协同训练，在全局测试集 D_{glb} 上也具备不错的识别能力。

为缓解数据异质、协同异构模型，云端分别存储着与真实有标记私有调制信号集 $\bigcup_{k=1}^K D_L^k$ 分布相

近的有标记公开伪信号样本集 $D_{\text{pseL}}^{\text{glb}}$、与真实无标记私有调制信号集 $\bigcup_{k=1}^{K} D_U^k$ 分布相近的无标记公开伪信号样本集 $D_{\text{pseU}}^{\text{glb}}$，其中，$D_{\text{pseL}}^{\text{glb}}$、$D_{\text{pseU}}^{\text{glb}}$ 均不涉及本地设备的数据隐私且可进行共享，辅助全局和本地模型的训练。

综合上述条件，本地设备上异构联邦半监督调制类型识别的优化问题可表示为

$$\min_{\theta_1,\theta_2,\cdots,\theta_K} \sum_{k=1}^{K} \alpha_k L_k \left(\sum_{i=1}^{\left|D_U^k\right|+\left|D_{\text{pseU}}^{\text{glb}}\right|} l_U\left(\boldsymbol{x}_i^k;\boldsymbol{\theta}_k\right), \sum_{j=1}^{\left|D_L^k\right|+\left|D_{\text{pseL}}^{\text{glb}}\right|} l_L\left(\boldsymbol{x}_j^k, y_j^k;\boldsymbol{\theta}_k\right) \right) \tag{1}$$

其中，$\alpha_k = \dfrac{\left|D_L^k\right|+\left|D_U^k\right|+\left|D_{\text{pseL}}^{\text{glb}}\right|+\left|D_{\text{pseU}}^{\text{glb}}\right|}{\sum_{i=1}^{K}\left(\left|D_L^i\right|+\left|D_U^i\right|+\left|D_{\text{pseL}}^{\text{glb}}\right|+\left|D_{\text{pseU}}^{\text{glb}}\right|\right)}$；$L_k$ 表示本地损失融合函数，用于权衡无标记样本和有标记样本对本地模型参数的贡献度；l_U、l_L 分别表示同一轮优化过程中本地模型参数在无标记样本 $\boldsymbol{x}_i^k \in D_U^k \cup D_{\text{pseU}}^{\text{glb}}$ 上的损失函数和有标记样本 $\boldsymbol{x}_j^k \in D_L^k \cup D_{\text{pseL}}^{\text{glb}}$ 上的损失函数；$y_j^k \in C_{\text{glb}}$ 表示全局类别标签。

此外，本文主要侧重全局模型 $f_{\text{glb}}\left(\boldsymbol{\theta}_{\text{glb}}\right)$ 在不接触本地真实私有调制信号进行训练的前提下，对全局调制类型的识别能力，因此 $f_{\text{glb}}\left(\boldsymbol{\theta}_{\text{glb}}\right)$ 的优化目标就是最大化调制类型识别准确率。考虑调制信号的离散时间基带等效模型，接收信号 $r(n)$ 经过下变频处理后，可表示为

$$H_m: r(n) = s_m(n)h(n)\mathrm{e}^{\mathrm{j}2\pi n\Delta f+\theta} + w(n) \tag{2}$$

其中，$H_m \in \{\text{QPSK}, \text{2FSK}, \cdots\}$ 表示真实的信号调制类型，$m \in \{1,2,\cdots,M\}$ 表示 M 种调制类型的下标索引，同时也是数字化的类别标签，$s_m(n)$ 表示调制信号，$h(n)$ 表示无线信道的脉冲响应，$w(n)$ 表示复数形式下的加性白高斯噪声。为真实地模拟现实信号在传输过程中遭受的非理想因素干扰，在信号模型中引入载波的频率偏移 Δf 和相位偏移 θ。异构联邦半监督调制类型识别就是在式(1)的基础上，最大化所有调制信号样本 s_m^i 在全局模型中的输出 $f_{\text{glb}}\left(s_m^i;\boldsymbol{\theta}_{\text{glb}}\right)$ 与 H_m 的匹配程度，即

$$\max_{\boldsymbol{\theta}_{\text{glb}}} \frac{1}{N}\sum_{i=1}^{N} \Gamma\left(f_{\text{glb}}\left(s_m^i;\boldsymbol{\theta}_{\text{glb}}\right)\middle|H_m\right) \tag{3}$$

其中，s_m^i 表示第 m 种调制类型的第 i 个信号样本，N 表示接收到的所有调制信号样本总数，$\Gamma(\cdot)$ 为指示函数，具体表示为

$$\Gamma(x|y) = \begin{cases} 1, x = y \\ 0, \text{其他} \end{cases} \tag{4}$$

在全局模型评估阶段，式(3)也可作为调制类型识别的评估指标，即平均识别准确率

$$P_{\text{acc}}^{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} \Gamma\left(f_{\text{glb}}\left(s_m^i;\boldsymbol{\theta}_{\text{glb}}\right)\middle|H_m\right)\times 100\% \tag{5}$$

但是本文侧重探究所提算法适用的信噪比（SNR, signal-to-noise ratio）下限，为此定义不同信噪比下的识别准确率

$$P_{\text{acc}}^{\text{snr}} = \frac{1}{N_S}\sum_{i=1}^{N_S} \Gamma\left(f_{\text{glb}}\left(s_{m,i}^{\text{snr}};\boldsymbol{\theta}_{\text{glb}}\right)\middle|H_m\right)\times 100\% \tag{6}$$

其中，N_S 表示测试 SNR 下的样本总数，$s_{m,i}^{\text{snr}}$ 表示测试 SNR 下第 m 种调制类型的第 i 个信号样本。

2 算法设计

本节主要针对第 1 节提出的异构联邦半监督场景进行算法设计并详细说明其工作过程。图 2 给出了所提异构联邦半监督算法 BKD-FSSL 的主要流程，下面将对其中各个环节进行阐述。

2.1 伪样本合成

本文考虑采用变分自编码器（VAE）在本地设备上利用私有数据训练，在云端合成公开伪样本集 $D_{\text{pseL}}^{\text{glb}}$ 和 $D_{\text{pseU}}^{\text{glb}}$。

标准 VAE 由编码器 E 和解码器（生成器）D 构成。如果将 E 的参数设为 $\boldsymbol{\theta}$，那么编码器 $E(\boldsymbol{\theta})$ 可视作概率分布 $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$，将原始数据空间映射到隐变量空间；如果将 D 的参数设为 $\boldsymbol{\varphi}$，那么解码器 $D(\boldsymbol{\varphi})$ 可视作概率分布 $q_{\boldsymbol{\varphi}}(\boldsymbol{x}|\boldsymbol{z})$，使隐变量空间恢复到原始数据空间，这与自编码器的功能一致。为产生原始样本空间中未出现过的新样本，VAE 通过寻找概率分布 $q(\boldsymbol{z})$，迫使 $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ 逼近 $q(\boldsymbol{z})$，从 $q(\boldsymbol{z})$ 分布中采样得到新的隐变量 $\boldsymbol{z}$ 反馈给解码器 D 即能产生新样本。因此，VAE 的核心优化问题可归纳为

$$q(\boldsymbol{z}) = \arg\min_{q(\boldsymbol{z})} \mathrm{KL}\left(q(\boldsymbol{z})\middle|q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right) \tag{7}$$

其中，KL(·) 表示 KL 散度；$q(\boldsymbol{z})$ 通常为高斯分布，

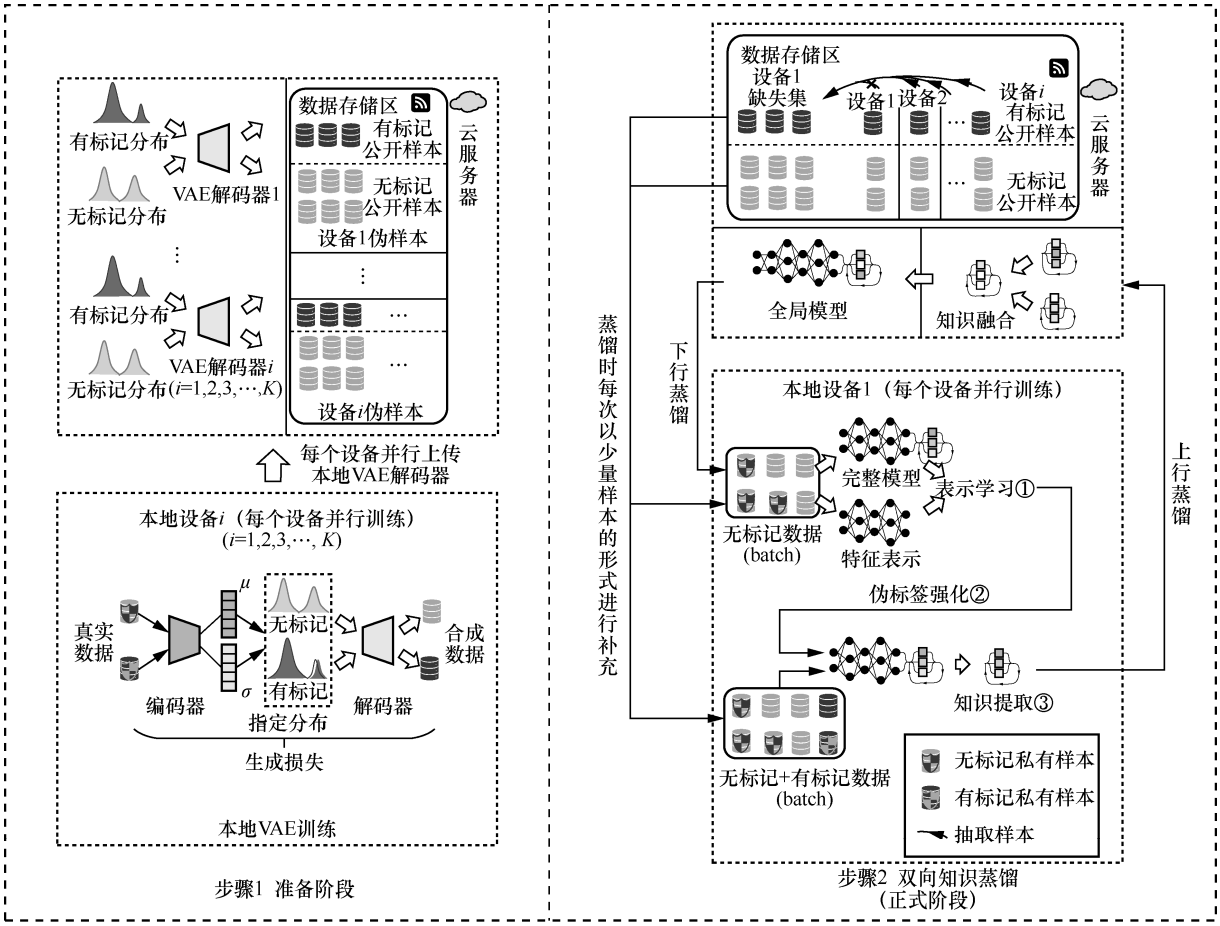


图 2　所提 BKD-FSSL 算法的主要流程

即 $\boldsymbol{z}\sim N(\boldsymbol{\mu},\boldsymbol{\sigma})$，$\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 为高斯分布的统计参数。

在训练过程中，为了保证梯度的传递性，引入重参数化技巧，即 $\boldsymbol{z}=\boldsymbol{\mu}+\boldsymbol{\varepsilon\sigma}$，其中 $\boldsymbol{\varepsilon}\sim N(0,1)$。此外，为保证产生的新样本与原始样本在数据分布上具有相似性，在损失函数中引入生成样本 $\tilde{\boldsymbol{x}}$ 和原始样本 $\boldsymbol{x}$ 相似性的评估指标。因此 VAE 的训练损失函数可表示为

$$L(\tilde{\boldsymbol{x}},\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\varphi})=L_{\text{mse}}(\tilde{\boldsymbol{x}},\boldsymbol{x})+\text{KL}\big(N(\boldsymbol{\mu},\boldsymbol{\sigma}),N(0,1)\big)\tag{8}$$

其中，L_{mse} 表示均方误差函数，用于评估生成样本 $\tilde{\boldsymbol{x}}$ 与原始样本 $\boldsymbol{x}$ 的相似程度。

本文中关于 VAE 编码器和解码器的结构均基于深度卷积神经网络进行设计，针对 IQ 调制信号 $N\times 2$ 的特殊尺寸，其中 N 为信号长度，对卷积核尺寸进行改进，详细参数设置如表 2 所示。从表 2 可知，编码器部分主要由 4 层卷积层和 2 层全连接层组成。其中，2 层全连接层分别用于获取高斯分布的统计参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$。通常，当隐变量以高维向量的形式反馈到解码器时，易生成高质量、高细粒度的样本。

表 2　VAE 编码器和解码器参数设置

结构名	网络层名	输出尺寸	参数配置
编码器	卷积层 1	32×512×1	Conv,32,15×2,S=(1,2)
	卷积层 2	64×256×1	Conv,64,5×1,S=(2,1)
	卷积层 3	128×128×1	Conv,128,3×1,S=(2,1)
	卷积层 4	256×64×1	Conv,256,3×1,S=(2,1)
	展平层	16 384×1	—
	输出层	128×1	全连接层
		128×1	全连接层
解码器	扩展层	16 384×1	全连接层
	变形层	256×64×1	—
	反卷积层 1	128×128×1	TConv,128,2×1,S=(2,1)
	反卷积层 2	64×256×2	TConv,64,2×1,S=(2,2)
	反卷积层 3	32×512×2	TConv,32,2×1,S=(2,1)
	反卷积层 4	1×512×2	TConv,1,1×1,S=(1,1)

本文考虑采用 128 维的隐变量 $\boldsymbol{z}$ 来生成伪样本，将编码器的输出层设置为 2 层含有 128 个神经元的全连接层，分别来获得 128 维的 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$，进而根据重参数化技巧转变成 128 维的 $\boldsymbol{z}$ 用于伪信号样本的生成。

表 2 中，“$\mathrm{Conv},32,15\times2,S=(1,2)$”的含义如下：“Conv”表示由卷积层−批量归一化层−ReLU 层组成的序列化模块，“32”表示卷积核的通道数，“15×2”表示卷积核的尺寸，“$S=(1,2)$”表示卷积核在信号样本长度和宽度上移动的步幅。

解码器主要对编码器的输出进行逆处理，因此，表 2 中解码器的结构采用与编码器对应的逆变换。其中，“TConv”表示由反卷积层−批量归一化层−ReLU 层组成的序列化模块。

2.2 半监督训练模块

对于半监督训练模块的构建，本文考虑采用表示学习和伪标签技术进行设计。

半监督训练模块部署在云端和本地设备上，云端利用有标记公开集 $D_{\mathrm{pseL}}^{\mathrm{glb}}$ 和无标记公开集 $D_{\mathrm{pseU}}^{\mathrm{glb}}$ 进行训练，而本地设备则利用有标记的 $D_L^k \cup D_{\mathrm{pseL}}^{\mathrm{glb}}$ 和无标记的 $D_U^k \cup D_{\mathrm{pseU}}^{\mathrm{glb}}$ 进行训练。因此，为方便本节叙述，定义有标记样本集为 D_L，无标记样本集为 D_U，两者对应的类别集均为 $C_{\mathrm{glb}}=\{1,2,3,\cdots,M\}$。

在所提半监督模块中，考虑额外引入数据增强来提升模型的泛化能力。下面优先对本文设计的增强方式进行介绍。

1) 旋转增强

旋转增强就是对 IQ 信号进行一定角度的旋转，本文考虑对信号进行随机 $\theta\in\left\{0,\dfrac{\pi}{2},\pi,\dfrac{3\pi}{2}\right\}$ 角度的旋转，变换过程为

$$\widehat{\boldsymbol{x}}=\mathrm{Aug}(\boldsymbol{x})=\begin{bmatrix}\widehat{\boldsymbol{I}}\\ \widehat{\boldsymbol{Q}}\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{Q}\end{bmatrix} \tag{9}$$

2) MixUp 增强

MixUp 增强是一种强增强方式而上述旋转增强是一种弱增强方式。具体过程如下：给定有标记子集 $B_L=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^N$，其中 $\boldsymbol{x}_i\in D_L$，y_i 是对应的标签，N 为样本数量。首先创建 $\left(\boldsymbol{x}_i,\boldsymbol{x}_i^{'}\right)$ 样本对，其中 $\boldsymbol{x}_i\in B_L,\boldsymbol{x}_i^{'}\in B_L^{'},i\in\{1,2,\cdots,N\}$，$B_L^{'}$ 是 B_L 随机排序后的集合。然后，将样本对中的 2 个样本按照混合比例 λ_{ii} 进行混合，混合过程为

$$\widehat{\boldsymbol{x}}_{ii}=\mathrm{MixUp}\left(\boldsymbol{x}_i,\boldsymbol{x}_i^{'}\right)=\lambda_{ii}\boldsymbol{x}_i+\left(1-\lambda_{ii}\right)\boldsymbol{x}_i^{'} \tag{10}$$

其中，混合系数 $\lambda_{ii}\sim\mathrm{Beta}(\alpha,\alpha)$，$\alpha$ 为 Beta 分布的统计参数，本文中 $\alpha=1$。

最后将混合样本 $\widehat{\boldsymbol{x}}_{ii}$ 输入模型中进行强化训练。训练过程中的损失函数为

$$L_{\underset{\boldsymbol{x}_i,y_i\in B_L,\boldsymbol{x}_i^{'}\in B_L^{'}}{\mathrm{Smix}}}\left(\boldsymbol{x}_i,y_i,\boldsymbol{x}_i^{'};\boldsymbol{\theta}\right)=L_{\mathrm{CE}}(\boldsymbol{x}_i,y_i;\boldsymbol{\theta})+L_{\mathrm{MixUp}}\left(\widehat{\boldsymbol{x}}_{ii};\boldsymbol{\theta}\right) \tag{11}$$

其中，$L_{\mathrm{CE}}(\boldsymbol{x}_i,y_i;\boldsymbol{\theta})$ 表示交叉熵损失，L_{MixUp} 表示混合样本损失。事实上，MixUp 会记录与样本对匹配的标签对，即 $\left(\widehat{\boldsymbol{x}}_{ii},y_i,y_i^{'}\right)$，因此 L_{MixUp} 可表示为

$$L_{\mathrm{MixUp}}\left(\widehat{\boldsymbol{x}}_{ii};\boldsymbol{\theta}\right)=\lambda_{ii}L_{\mathrm{CE}}\left(\widehat{\boldsymbol{x}}_{ii},y_i;\boldsymbol{\theta}\right)+\left(1-\lambda_{ii}\right)L_{\mathrm{CE}}\left(\widehat{\boldsymbol{x}}_{ii},y_i;\boldsymbol{\theta}\right) \tag{12}$$

少量有标记样本通过 MixUp 增强可以减缓模型训练时过拟合的速度，提升有标记数据对模型的贡献度。

此外，所提算法对大量的无标记样本也进行了 MixUp 增强，但由于缺少带有先验知识的标签数据，无法对模型的预测结果进行有效约束，因此监督式的 MixUp 方式并不可取。本文考虑将无标记样本的 MixUp 增强用于表示学习，并对训练过程中的损失函数进行重新设计，具体过程介绍如下。

① 表示学习

如图 3 所示，首先构建特征表示模块 $f\left(\boldsymbol{\theta}\right)$，然后随机选取无标记子集 $B_U=\{\boldsymbol{x}_i\}_{i=1}^N$，其中 $\boldsymbol{x}_i\in D_U$，但由于缺乏真实标签数据，无法监督训练过程，为此考虑引入虚拟标签集 $V=\{1,2,\cdots,N\}$。虚拟标签 $v_i\in V$ 根据样本 $\boldsymbol{x}_i$ 在子集 B_U 中的位置进行创建。随后进行 MixUp 增强，同样创建 $\left(\boldsymbol{x}_i,\boldsymbol{x}_i^{'}\right)$ 样本对，其中 $\boldsymbol{x}_i\in B_U,\boldsymbol{x}_i^{'}\in B_U^{'}$，$B_U^{'}$ 是 B_U 随机排序后的集合。将样本对 $\left(\boldsymbol{x}_i,\boldsymbol{x}_i^{'}\right)$ 中的 2 个样本根据式(9)进行 MixUp 得到样本 $\widehat{\boldsymbol{x}}_{ii}$，记为 $\left(\widehat{\boldsymbol{x}}_{ii},v_i,v_i^{'}\right)$，其中 $v_i^{'}$ 是样本 $\boldsymbol{x}_i^{'}$ 的虚拟标签，也就是 $\boldsymbol{x}_i^{'}$ 在 B_U 中的位置。然后输入 $f(\boldsymbol{\theta})$ 获取特征表示 $\hat{\boldsymbol{z}}_{ii}$，记为 $\left(\hat{\boldsymbol{z}}_{ii},v_i,v_i^{'}\right)$。其次，对具有虚拟标签的样本 $\boldsymbol{x}_i$ 进行旋转增强，根据前文所述，旋转增强不会改变样本在集合中的位置，因此将旋转后的样本 $\widehat{\boldsymbol{x}}_i$ 和对应的虚拟标签 v_i 记

为$\left(\hat{\boldsymbol{x}}_i, v_i\right)$，同样输入表示模块获取特征表示$\hat{\boldsymbol{z}}_i$，记为$\left(\hat{\boldsymbol{z}}_i, v_i\right)$。

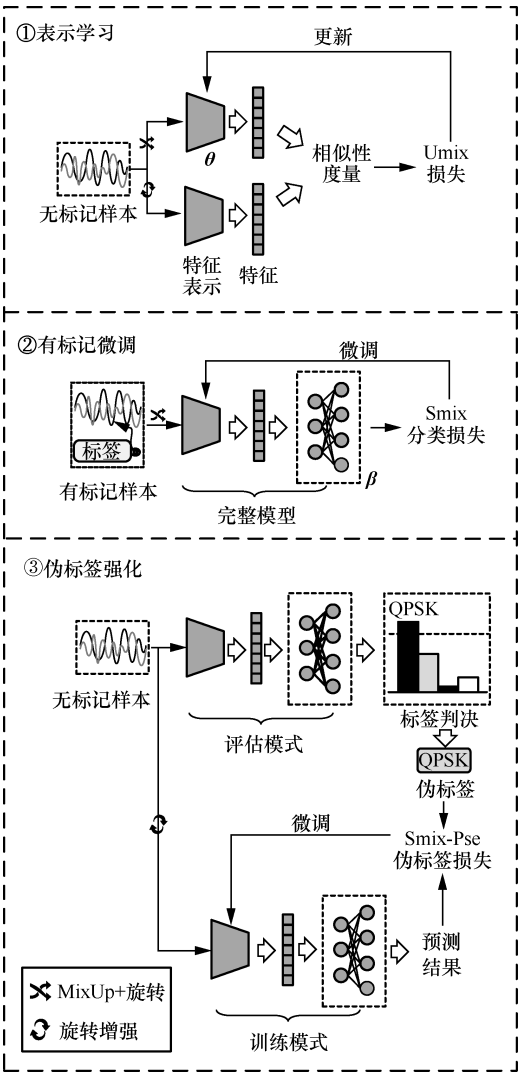


图 3　内嵌半监督模块流程

最后，对$\hat{\boldsymbol{z}}_{ii}$和$\hat{\boldsymbol{z}}_i$进行余弦相似度计算，具体表示为

$$\operatorname{Sim}\left(\hat{\boldsymbol{z}}_{ii}, \hat{\boldsymbol{z}}_i\right)=\frac{\hat{\boldsymbol{z}}_{ii}\ \hat{\boldsymbol{z}}_i^{\mathrm{T}}}{\left\|\hat{\boldsymbol{z}}_{ii}\right\|_2\left\|\hat{\boldsymbol{z}}_i\right\|_2} \tag{13}$$

其中，$\|\cdot\|_2$表示 L2 范数。训练过程中的损失函数为

$$L_{\mathrm{Umix}}=\lambda_{ii} L_{\mathrm{CE}}\left(\operatorname{Sim}\left(\hat{\boldsymbol{z}}_{ii}, \hat{\boldsymbol{z}}_i\right), v_i\right)+\left(1-\lambda_{ii}\right) L_{\mathrm{CE}}\left(\operatorname{Sim}\left(\hat{\boldsymbol{z}}_{ii}, \hat{\boldsymbol{z}}_i\right), v_i^{'}\right) \tag{14}$$

② 有标记微调

表示学习使$f(\boldsymbol{\theta})$初步具备了对无标记数据的特征表示能力，然而对于易混淆的数据表示能力仍有所欠缺，因此利用有标记数据对其进行微调，同时更新分类模块的参数$\boldsymbol{\beta}$。通过对有标记数据进行 MixUp 增强，根据式(11)计算$L_{\mathrm{Smix}}(\boldsymbol{x}_i;\boldsymbol{\theta},\boldsymbol{\beta})$优化完整的模型$f(\boldsymbol{\theta},\boldsymbol{\beta})$，其中$\boldsymbol{x}_i \in D_L$。

③ 伪标签强化

通过有标记数据的微调，固然会使模型参数得到改善，但是即使经过 MixUp 增强，少量的有标记数据对模型的贡献度仍然有限。为充分利用无标记数据，本文考虑对模型$f(\boldsymbol{\theta},\boldsymbol{\beta})$进行伪标签强化训练。

随机选取无标记子集$B_U=\{\boldsymbol{x}_i\}_{i=1}^N$，获取样本$\boldsymbol{x}_i \in B_U$的预测概率向量$\boldsymbol{p}_i=f(\boldsymbol{x}_i;\boldsymbol{\theta},\boldsymbol{\beta})$。当向量$\boldsymbol{p}_i$中的最大元素$p_{ij}=\arg\max(\boldsymbol{p}_i)$超过设定的阈值$p_d$时，该样本$\boldsymbol{x}_i$会被判决为第$j$类，其中$j \in C_{\mathrm{glb}}$，$p_{ij}$表示第$i$个样本预测概率向量$\boldsymbol{p}_i$中的第$j$个元素。具体判决过程为

$$y_{\mathrm{pse}}^i=\mathrm{PseLab}(p_{ij}, p_d)=\begin{cases} j, p_{ij} \geqslant p_d \\ 0, p_{ij}<p_d \end{cases} \tag{15}$$

本文中标签判决门限$p_d=0.75$。然后，将标注伪标签的B_U与有标记子集B_L合并，对标签$y_i>0$的样本再次进行 MixUp，同样根据式(11)计算损失$L_{\text{Smix-Pse}}(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta},\boldsymbol{\beta})$来强化模型$f(\boldsymbol{\theta},\boldsymbol{\beta})$，其中$\boldsymbol{x}_i \in D_L, \boldsymbol{x}_j \in D_U$。半监督模块的完整伪代码如附录 1 所示。

2.3　“多教师–单学生”的上行蒸馏

上行蒸馏的实质是本地到云端的“多教师–单学生”蒸馏体系，将本地设备上的模型视作“教师”网络，云端的全局模型视为“学生”网络，通过云端存储的公开伪样本集$D_{\mathrm{pseL}}^{\mathrm{glb}}$和$D_{\mathrm{pseU}}^{\mathrm{glb}}$将本地设备的异质知识融合，传递给全局模型，强化全局模型对真实数据集D_{glb}的识别能力。

在进行上行蒸馏之前，本地设备将通过 2.2 节中的半监督学习模块，利用本地有标记样本集$D_L^k \cup D_{\mathrm{pseL}}^{\mathrm{glb}}$和无标记样本集$D_U^k \cup D_{\mathrm{pseU}}^{\mathrm{glb}}$对模型$f_k\left(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)$进行预热。然后，开始上行蒸馏：云端随机指定子集$B_U \subseteq D_{\mathrm{pseU}}^{\mathrm{glb}}$和$B_L \subseteq D_{\mathrm{pseL}}^{\mathrm{glb}}$，每个本地模型在相同的$B_L \cup B_U$上进行评估，为简化叙述，这里的样本$\boldsymbol{x}_i$不区分设备索引。将提取的“异质知

识”上传到云端，对于第 k 个模型，有

$$\boldsymbol{p}_{k,i}=f_k\left(\boldsymbol{x}_i;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)[-2] \tag{16}$$

其中，$\boldsymbol{p}_{k,i}$ 表示第 k 个本地模型对样本 $\boldsymbol{x}_i\in B_L\cup B_U$ 提取的“异质知识”，$f_k(\cdot)[-2]$ 表示模型倒数第 2 层的输出。

当 K 个本地模型并行上传“异质知识”结束后，云端对收集到的“异质知识”进行分组，对于有标记的知识，记为 $\text{llogits}=\left\{\left(\boldsymbol{x}_i,\boldsymbol{p}_{k,i},y_i\right)\right\}_{i=1}^{|B_L|}$，其中 $y_i\in C_{\text{glb}}=\cup_{k=1}^{K}C_k$；而对于无标记的知识则记为 $\text{ulogits}=\left\{\left(\boldsymbol{x}_j,\boldsymbol{p}_{k,j}\right)\right\}_{j=1}^{|B_U|}$。然后对不同设备的“异质知识”进行融合，具体表示为

$$\boldsymbol{p}_i=\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}_{k,i} \tag{17}$$

其中，$\boldsymbol{p}_{k,i}\in\text{llogits}\cup\text{ulogits}$。上述集合更新为 $\text{llogits}'=\left\{(\boldsymbol{x}_i,\boldsymbol{p}_i,y_i)\right\}_{i=1}^{|B_L|}$ 和 $\text{ulogits}'=\left\{(\boldsymbol{x}_j,\boldsymbol{p}_j)\right\}_{j=1}^{|B_U|}$。

模型的异构性往往会加重“异质知识”的差异性。通过平均这些“异质知识”，可以消融部分差异，锐化本地模型基于真实私有类别 C_k 训练获得的参数在融合知识中的体现，从而通过蒸馏的方式转移到全局模型上，具体过程为

$$L_d=T_{\text{l}}^2\text{KL}\left(\text{softmax}\left(\frac{\boldsymbol{p}_i}{T_{\text{l}}}\right),\text{softmax}\left(\frac{\boldsymbol{p}_{\text{glb},i}}{T_{\text{l}}}\right)\right) \tag{18}$$

其中，T_{l} 表示上行蒸馏中的温度参数，用于软化融合知识；$\text{softmax}(\cdot)$ 表示 softmax 激活函数，$\boldsymbol{p}_{\text{glb},i}=f_{\text{glb}}\left(\boldsymbol{x}_i;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)[-2]$ 表示全局模型提取的知识，$\boldsymbol{p}_i\in\text{llogits}'\cup\text{ulogits}'$。然后将半监督学习过程嵌入蒸馏过程中，可得上行蒸馏的完整损失函数为

$$\begin{aligned}L_{\text{up}}=&L_{\text{Umix}}\left(\boldsymbol{x}_i;\boldsymbol{\theta}_{\text{glb}}\right)+L_{\text{Smix}}\left(\boldsymbol{x}_j;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)+\\&L_{\text{Smix-Pse}}\left(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)+L_d\left(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)\end{aligned} \tag{19}$$

其中，$\boldsymbol{x}_i\in D_{\text{pseU}}^{\text{glb}},\boldsymbol{x}_j\in D_{\text{pseL}}^{\text{glb}}$。当有标记样本充足时，即完全监督场景，式(19)可退化为

$$L_{\text{up}}=L_{\text{CE}}\left(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)+L_d\left(\boldsymbol{x}_i;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right) \tag{20}$$

其中，$\boldsymbol{x}_i$ 是标记样本，y_i 是对应的标签。上行蒸馏的完整伪代码如附录 2 所示。

2.4 “单教师–多学生”的下行蒸馏

本文所提场景考虑了不同本地设备存在缺失类别的情况。因此，下行蒸馏主要提升本地模型对缺失类别的识别能力。与上行蒸馏相反，下行蒸馏是云端到本地的“单教师–多学生”蒸馏体系，将全局模型学习到的隐含多方真实私有类别 $\{C_k\}_{k=1}^{K}$（即 C_{glb}）信息的全局知识提取回本地，辅助本地模型 f_k 在本地数据集上通过蒸馏的方式，等效识别其他设备上的私有类别 $C_{\text{glb}}-C_k$，也就是本地缺失类别，从而提升本地模型对缺失类别的识别能力。本地私有类别的识别提升主要依靠内嵌的半监督过程。整体而言，最终提升本地模型对全局类别 C_{glb} 的识别能力。

在下行蒸馏中，本地设备重新筛选、排列出与本地类别不重叠，即缺失类别的伪样本集 $D_{\text{pseL}}^{\text{glb}'}$ 和 $D_{\text{pseU}}^{\text{glb}'}$，组成 $D_L^k\cup D_{\text{pseL}}^{\text{glb}'}$ 和 $D_U^k\cup D_{\text{pseU}}^{\text{glb}'}$。随机选取子集 $B_L^k\subseteq D_L^k\cup D_{\text{pseL}}^{\text{glb}'}$ 和 $B_U^k\subseteq D_U^k\cup D_{\text{pseU}}^{\text{glb}'}$，计算全局模型的知识 $\boldsymbol{p}_{k,i}^{\text{glb}}=f_{\text{glb}}\left(\boldsymbol{x}_i^k;\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}}\right)[-2]$，其中 $\boldsymbol{x}_i^k\in B_L^k\cup B_U^k$，$\boldsymbol{p}_{k,i}^{\text{glb}}$ 表示全局模型对第 k 个设备上第 i 个样本提取的知识。与上行蒸馏不同的是，全局模型的知识不需要全部转移到本地模型上，根据前面所述，下行蒸馏是优化本地缺失类别对应的模型参数，如图 4 所示。私有类别的差异性导致不同本地模型关于缺失类别的输出并非连续，如图 4 中“情况 2”所示。

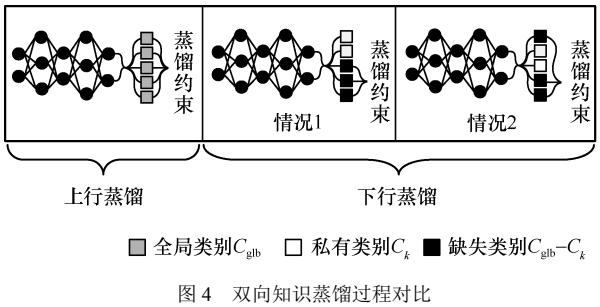


图 4 双向知识蒸馏过程对比

为此，设计选择性蒸馏机制，具体过程如下：首先，对全局模型知识 $\boldsymbol{p}_{k,i}^{\text{glb}}$ 中关于本地缺失类别的部分进行软化，具体表示为

$$p_{k,ij}^{\text{glb}}=\text{soft}\left(\frac{\boldsymbol{p}_{k,i}^{\text{glb}}}{T_2}\right)^{\{C_{\text{glb}}-C_k\}}=\frac{\mathrm{e}^{\frac{p_{k,ij}^{\text{glb}}}{T_2}}}{\sum\limits_{j\in\{C_{\text{glb}}-C_k\}}\mathrm{e}^{\frac{p_{k,ij}^{\text{glb}}}{T_2}}} \tag{21}$$

其中，T_2 表示下行蒸馏中的温度参数，$p_{k,ij}^{\mathrm{glb}}$ 表示 $\boldsymbol{p}_{k,i}^{\mathrm{glb}}$ 中的第 j 个元素，$\left\{C_{\mathrm{glb}}-C_k\right\}$ 表示本地缺失类别的数字标签集。同样地，本地模型也将在相同的样本 $\boldsymbol{x}_i^k$ 上提取知识并根据式(21)软化为 $p_{k,ij}^{\mathrm{loc}}$。最后可得损失 L_D^k，具体表示为

$$L_D^k=-T_2^2\sum_{j\in\left\{C_{\mathrm{glb}}-C_k\right\}}p_{k,ij}^{\mathrm{glb}}\log\left(p_{k,ij}^{\mathrm{loc}}\right) \tag{22}$$

此外，同样在蒸馏过程中嵌入半监督学习过程，于是可得下行蒸馏的总体损失 L_{down}^k

$$\begin{aligned}L_{\mathrm{down}}^k=&L_{\mathrm{Umix}}^k\left(\boldsymbol{x}_i^k;\boldsymbol{\theta}_k\right)+L_{\mathrm{Smix}}^k\left(\boldsymbol{x}_j^k;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)+\\&L_{\mathrm{Smix\text{-}Pse}}^k\left(\boldsymbol{x}_i^k,\boldsymbol{x}_j^k;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)+L_D^k\left(\boldsymbol{x}_i^k,\boldsymbol{x}_j^k;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)\end{aligned} \tag{23}$$

其中，$\boldsymbol{x}_i^k\in B_U^k,\boldsymbol{x}_j^k\in B_L^k$。同样地，在完全监督场景下，式(23)也可退化为

$$L_{\mathrm{down}}^k=L_{\mathrm{CE}}^k\left(\boldsymbol{x}_i^k,y_i^k;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right)+L_D^k\left(\boldsymbol{x}_i^k;\boldsymbol{\theta}_k,\boldsymbol{\beta}_k\right) \tag{24}$$

其中，$\boldsymbol{x}_i^k$ 是标记样本，y_i^k 是对应的标签。下行蒸馏 DownKD 算法的完整伪代码如附录 3 所示。

3　仿真与性能分析

为验证所提算法的性能效果，本节通过设置不同仿真场景，对比多种算法，对所提算法的性能进行评估。

3.1　仿真调制信号生成

本文考虑通过 MATLAB 生成 10 种调制类型的信号，分别为 QPSK、4FSK、16QAM、16PAM、4CPM、8ASK、MSK、AM、FM 和 OOK。此外，在生成信号过程中考虑的非理想因素如下。归一化载波频率偏差（相对于采样频率）Δf 在−0.1 到 0.1 的范围内随机选择，相位偏移 θ 在 0 到 2π 之间随机选择。同时，对脉冲信号进行成形处理，成形滤波器的滚降系数在 0.2 到 0.7 之间随机选择。信噪比（SNR）范围为−18 dB 到 20 dB，间隔为 2 dB。每个信号样本包含 64 个符号，并且过采样率为 8。在每个 SNR 下，对每种调制类型生成 1 000 个信号作为无标记样本和 20 个信号作为有标记样本，用于训练；生成 500 个信号样本用于测试。

本文所提算法的实现过程是在 NVIDIA GeForce GTX 3070Ti 上进行的，并在 PyTorch 1.13.1 中进行了模拟。在仿真过程中，首先进行一轮本地设备预热，然后进行 20 轮的双向联邦知识蒸馏，初始学习率为 0.001，在联邦蒸馏中每经过 4 轮，初始学习率降低一半。此外，本文还采用 Adam 优化器对分类损失和蒸馏损失进行了优化。训练过程中的批量大小根据仿真场景灵活设置。

3.2　合成信号质量分析

本文所提算法需要通过云端存储的有标记伪样本集和无标记伪样本集来协同异构模型，因此需要合成高质量的伪信号样本，但是如果合成信号与真实信号完全相同，则会存在隐私泄露的风险。为此，本文对合成信号的质量进行了分析，结果如图 5 和图 6 所示。

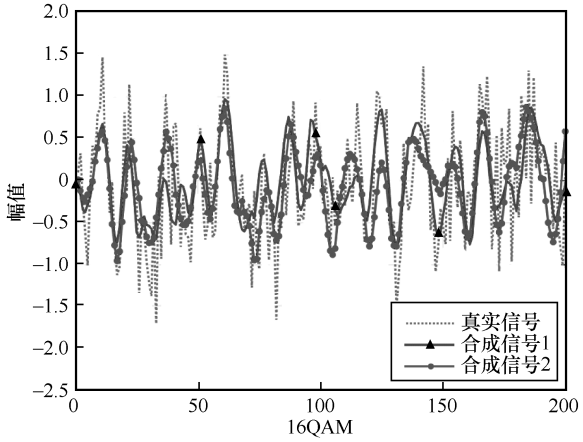


图 5　信噪比为 0 dB 时 16QAM 真实信号与两次合成信号的可视化对比

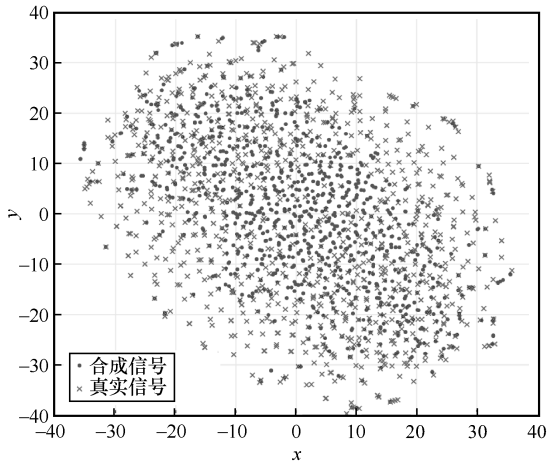


图 6　真实信号与合成信号的统计分布对比

从图 5 中可以看到，真实信号与合成信号在时域波形上存在很大不同，这减小了隐私泄露的风险。而且两次合成产生的信号也有所不同，这体现了合成样本的多样性。此外，图 6 给出了在真实样

本集和合成样本集中各随机抽样 1 000 个样本并经过 T 分布随机近邻嵌入（T-SNE）降维可视化后的对比，也就是两者统计分布的比较。从图 6 中可以看到，合成信号分布与真实信号分布存在重叠部分，这确保了合成信号与真实信号存在相似性。以上结果表明，合成信号的质量较高，为后续的仿真实验奠定了基础。本文考虑在云端生成与真实信号数量一致的伪样本作为公开集。

3.3 不同数量客户端的性能分析

为验证所提算法的性能，本文通过设置以下 6 种场景来对比所提算法的性能。

1) 少量有标记合成样本直接训练。通过每类 20 个有标记伪样本直接对全局模型进行训练。

2) 少量有标记合成样本的模型异构 FedMD。FedMD 是经典的监督式联邦模型异构算法，仅通过不同本地模型间的 KD 进行协同训练。本文利用有标记样本对多个客户端进行 FedMD 训练。但是，由于标准的 FedMD 算法不具备全局模型，本文考虑在不影响其正常工作的情况下，以普通客户端的形式嵌入全局模型来对比所提算法。

3) 半监督 FedMatch。FedMatch 是现行模型同构场景下的联邦半监督算法，本文考虑将其部署在所提异构场景中，对于数据异质问题，采用本文所提共享伪样本数据集的方式解决；对于异构模型的聚合过程，则以 FedMD 的方式进行。

4) 半监督 Semi-HFL。Semi-HFL 是现行数据异质和模型异构同时存在场景下的异构联邦半监督算法，通过级联本地异构模型组成全局模型，对相同的结构进行平均聚合，以此来解决模型异构问题。在本地模型训练过程中，附加约束项来缓解数据异质的情况。

5) 半监督 BKD-FSSL（所提算法）。在该算法中，首先将 10 类调制信号划分为若干簇，来模拟 5 个客户端持有的异质数据，具体划分如表 3 所示。其次，分配对应的异构模型，具体结构如表 4 所示。其中，Dw 表示深度卷积，Pw 表示逐点卷积，AvgPool 表示平均池化，Fc×M 表示含有 M 个神经元的全连接层。其余参数配置含义与表 2 相同。此外，本文所提的“特征表示模块”由表 4 中“分类模块”前的子模块构成。最后，将上行蒸馏过程中的温度参数 T_1 设置为 20，下行蒸馏过程中的温度参数 T_2 设置为 1。

表 3 不同客户端所持私有类别及异构模型

客户端名称	私有类别		模型名称
	本文生成	RML2016.10a	
客户端 1	QPSK、4FSK	8PSK、BPSK	模型 4
客户端 2	16QAM、16PAM	CPFSK、GFSK	模型 1
客户端 3	4CPM、8ASK	4PAM、16QAM	模型 2
客户端 4	MSK、AM	64QAM、QPSK	模型 3
客户端 5	FM、OOK	AM-DSB、WBFM	模型 5

6) 足量标记样本的 BKD-FSSL。在本文所提半监督算法的基础上，进行足量有标记样本的仿真对比，来指示所提算法的性能上限。

此外，为验证所提算法在调制信号识别方面的普适性，本文在 RML2016.10a 调制信号公开集上也进行了仿真实验。RML2016.10a 数据集中 SNR 范围为−20 dB 到 18 dB，间隔为 2 dB，调制类型和具体划分如表 3 所示。同时，根据所提半监督场景的设置，对 RML2016.10a 数据集中每个 SNR 下每类 1 000 个样本进行划分：随机采样 15 个样本作为有标记样本，735 个样本作为无标记样本，250 个样本作为测试样本。

本文设置的对比场景均比较的是全局模型的性能，为对比所提算法在不同数量客户端下的性能，分别进行了 2（客户端 1、2）、3（客户端 1~3）、5（客户端 1~5）个客户端的仿真实验，以式(6)中的 $P_{\text{acc}}^{\text{snr}}$ 作为评估指标，对全局模型在 20 种信噪比下的分类准确率进行评估，仿真结果如图 7 和图 8 所示。

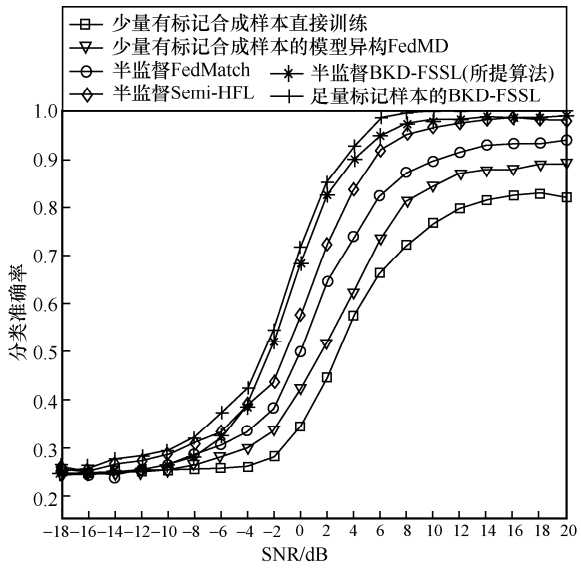
从图 7 中可以看到，所提算法的性能均优于其他对比算法，这是合理的：一方面根据 3.2 节的仿真结果可知，合成样本的数据分布与真实样本相近，却并非一致，同时仅少量合成样本具备标签，导致直接通过少量有标记合成样本集中训练获得的全局模型过度拟合伪样本的分布，削弱了在真实样本上的泛化能力。而 FedMD 模型异构算法通过协同训练，让全局模型交互到基于真实私有样本训练获得的本地模型参数信息，并且在协同聚合的过程中隐式扩充了有标记样本数量，因此相比于直接训练，FedMD 仅通过少量有标记样本也能获得性能较大提升的全局模型。但是每类 20 个有标记样本的设定对于 FedMD 来说仍然不充足，导致其性能提升十分有限。得益于数据增强处理，半监督的

表 4　　异构模型结构及参数配置

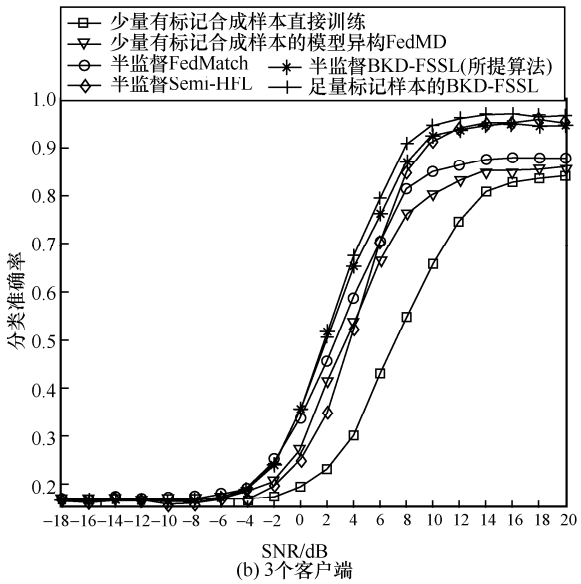
模块名称	参数配置	本地模型					全局模型
		模型 1	模型 2	模型 3	模型 4	模型 5	
Public Block	$\begin{bmatrix}\text{Conv},48,15\times2\\ \text{Conv},64,7\times1,S=(2,1)\end{bmatrix}$	✓	✓	✓	✓	✓	✓
Common Block	$\begin{bmatrix}\text{Conv},64,3\times1,S=(2,1)\\ \text{Conv},128,3\times1,S=(2,1)\\ \text{Conv},256,3\times1,S=(2,1)\end{bmatrix}$	×	×	×	×	✓	×
Residual Block 1	$\begin{bmatrix}\text{Conv},64,3\times1\\ \text{Conv},64,3\times1,S=(2,1)\\ \text{Conv},64,3\times1\end{bmatrix}$	✓	✓	✓	✓	×	×
Residual Block 2	$\begin{bmatrix}\text{Conv},64,3\times1\\ \text{Conv},64,3\times1\\ \text{Conv},64,3\times1\end{bmatrix}$	✓	×	×	×	×	×
Residual Block 3	$\begin{bmatrix}\text{Conv},128,3\times1\\ \text{Conv},128,3\times1,S=(2,1)\\ \text{Conv},128,3\times1\end{bmatrix}$	✓	✓	✓	✓	×	×
Residual Block 4	$\begin{bmatrix}\text{Conv},128,3\times1\\ \text{Conv},128,3\times1\\ \text{Conv},128,3\times1\end{bmatrix}$	✓	✓	×	×	×	×
Residual Block 5	$\begin{bmatrix}\text{Conv},256,3\times1\\ \text{Conv},256,3\times1,S=(2,1)\\ \text{Conv},256,3\times1\end{bmatrix}$	✓	✓	✓	✓	×	×
Residual Block 6	$\begin{bmatrix}\text{Conv},256,3\times1\\ \text{Conv},256,3\times1\\ \text{Conv},256,3\times1\end{bmatrix}$	✓	✓	✓	×	×	×
Mobile Block 1	$\begin{bmatrix}\text{Conv},64,3\times1,\text{Dw},S=(2,1)\\ \text{Conv},64,1\times1,\text{Pw}\end{bmatrix}$	×	×	×	×	×	✓
Mobile Block 2	$\begin{bmatrix}\text{Conv},128,3\times1,\text{Dw},S=(2,1)\\ \text{Conv},128,1\times1,\text{Pw}\end{bmatrix}$	×	×	×	×	×	✓
Mobile Block 3	$\begin{bmatrix}\text{Conv},256,3\times1,\text{Dw},S=(2,1)\\ \text{Conv},256,1\times1,\text{Pw}\end{bmatrix}$	×	×	×	×	×	✓
分类模块	$\left[\text{AvgPool},\text{Fc}\times M\right]$	✓	✓	✓	✓	✓	✓

FedMatch 和 Semi-HFL 相比于 FedMD 来说，性能又有所提升，如图 7 所示。

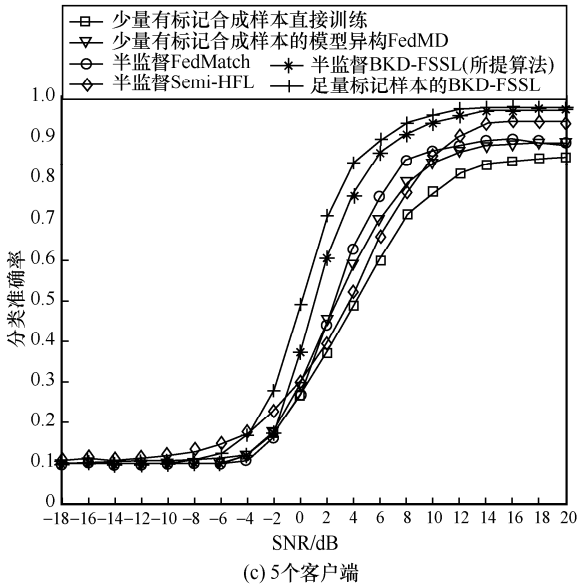
另一方面，所提算法通过双向知识蒸馏不仅能让全局模型交互到基于真实数据训练获得的本地模型参数信息，还能将其反馈至本地以选择性蒸馏的方式提升本地模型对异质数据的识别能力，而这是现行半监督 FedMatch 算法所欠缺的。Semi-HFL 虽然在本地训练环节中增加了约束项，减轻了本地模型偏离全局模型的程度，但是并未对异质数据进行有效的分类性能优化，当数据异质较轻时，仅靠约束确实能提升全局模型的分类能力。但是，当数据异质严重时，简单的约束对全局模型性能的提升作用一般。从图 7 中可以看到，随着客户端数量的增加，数据异质性逐渐加重，半监督 FedMatch 和 Semi-HFL 的性能逐渐衰退，尤其是 Semi-HFL 的性能衰退极其明显，而本文所提算法性能依旧稳定。此外，Semi-HFL 通过级联本地异构模型形成全局大模型的方法，只是从表面解决了模型异构的



(a) 2个客户端

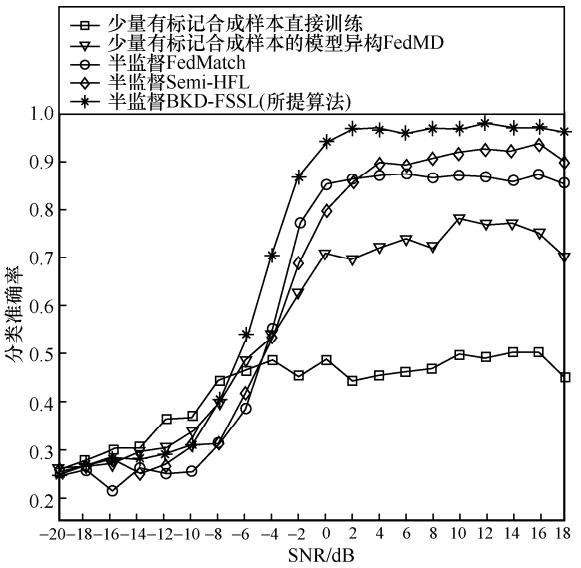


(b) 3个客户端

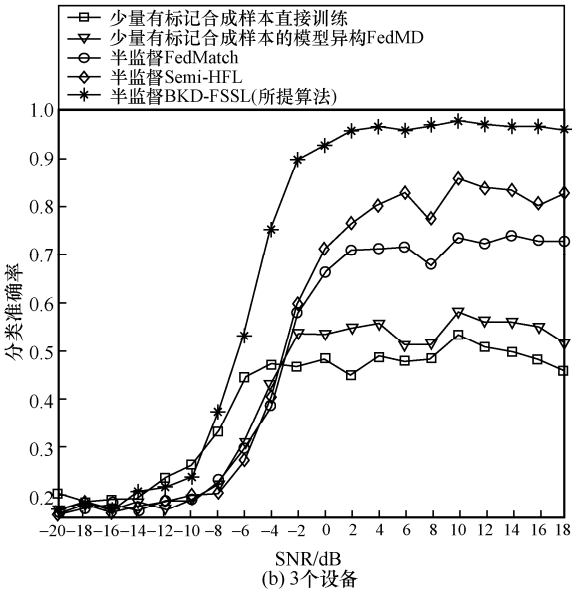


(c) 5个客户端

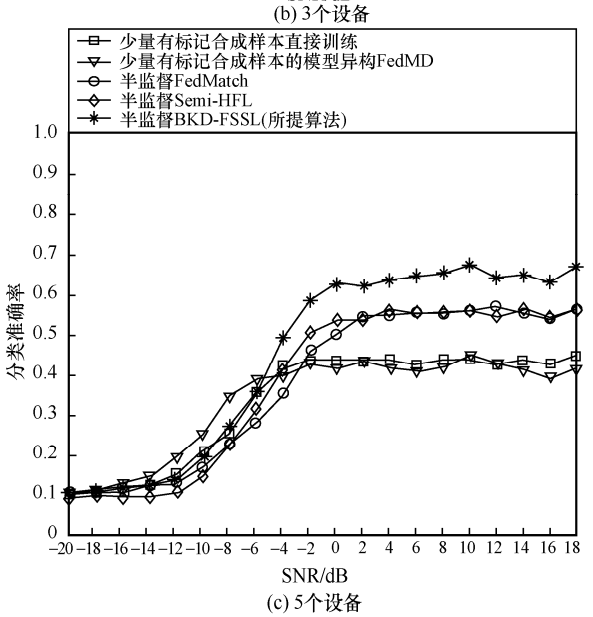
图 7　不同数量客户端下各场景的识别性能曲线



(a) 2个设备



(b) 3个设备



(c) 5个设备

图 8　不同数量客户端下各算法在 RML2016.10a 上的识别性能曲线

问题，并不能在协同训练中有效提升全局模型的分类能力。但是，本文所提算法通过融合不同设备上的异质知识来对全局模型进行蒸馏，这对全局模型的分类能力是有增强作用的，同时在下行蒸馏过程中，全局模型对本地模型进行选择性蒸馏，能提升本地模型对异质数据的分类能力。可见，双向蒸馏能够产生正向反馈作用，这也是本文所提算法优于 Semi-HFL 的原因。

此外，从图 8 中可以看到，所提算法在 RML2016.10a 调制信号公开集上也具备出色的调制类型识别能力。在 2、3 个客户端的场景下，所提算法性能稳定，且在 0 dB 时便达到了 90%以上的识别准确率。但是随着数据异质性的加重，在 5 个设备的场景下，所提算法的性能有所衰退，主要原因是：在 5 个设备时，全局模型的识别类型增加到 10 类，尤其是增加了 64QAM 等相对较难区分的高阶调制，这会大大增加模型的分类难度，同时 RML2016.10a 公开集中信号长度只有 128 个采样点，且归一化处理导致信号幅度范围过小，使信号特征稀疏，模型无法学到充足的调制参数信息去识别这些易混淆的高阶调制类型。但是，所提算法依旧优于对比算法。

3.4　不同数量标记样本的性能分析

为探究所提算法中有标记样本的数量下限，本文在 5 个客户端的场景中通过设置每个 SNR 下每类 1 个标记样本、每类 20 个标记样本、每类 40 个标记样本以及足量标记样本共 4 种场景来研究不同数量有标记样本对所提算法的性能影响，仿真结果如图 9 所示。

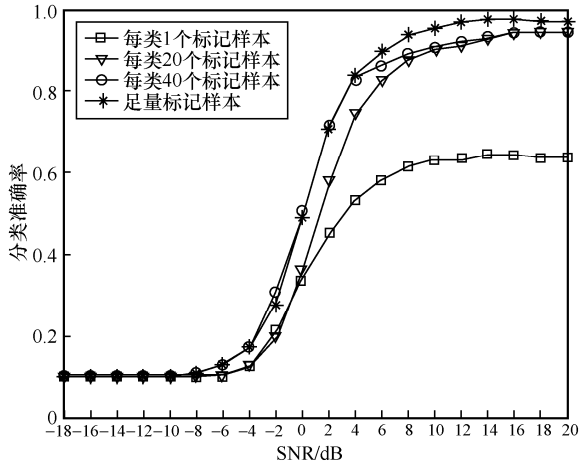


图 9　不同数量标记样本下所提算法的识别性能曲线

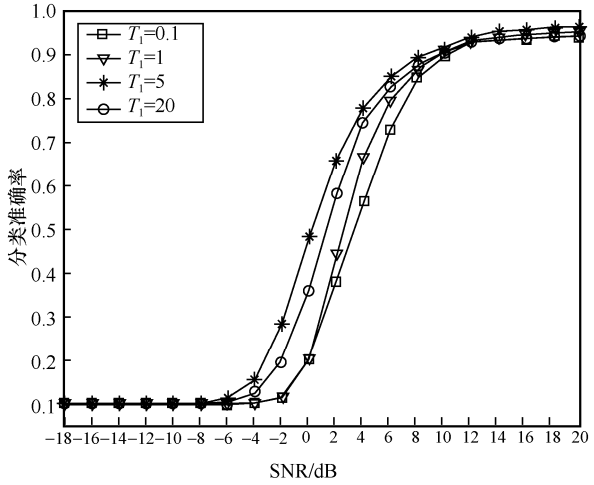
从图 9 中可以看到，所提算法在每类 20 个，总计 4 000 个有标记样本的情况下已经接近足量样本监督训练的性能，这证明了所提算法的有效性。

3.5　双向知识蒸馏过程中温度参数的影响

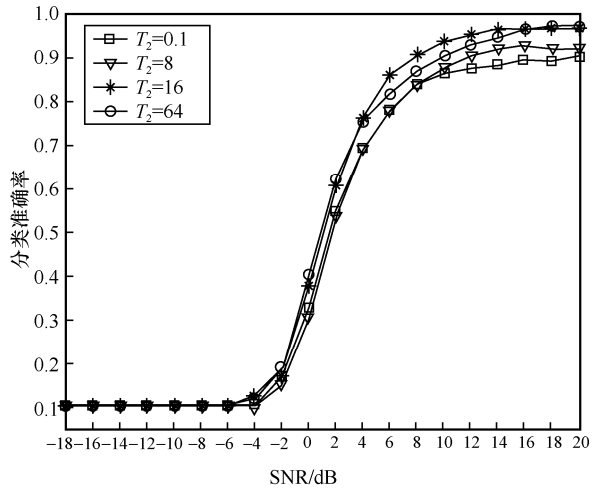
为解释双向知识蒸馏过程中，上行蒸馏温度参数 T_1 和下行蒸馏温度参数 T_2 在所提算法中的作用。本文同样在 5 个客户端的场景下，通过仿真验证不同温度参数对所提算法性能的影响，其中验证上行蒸馏温度参数 T_1 时，T_2 保持为 1；验证下行蒸馏温度参数 T_2 时，T_1 保持为 20。仿真结果如图 10 所示。

从图 10(a)中可以看到，上行蒸馏过程中的温度参数 T_1 主要影响中低信噪比下的调制类型识别性能。此外，随着 T_1 的增大，识别性能先上升后衰退，这有如下解释。一方面，中低信噪比下的有标记合成样本相对于高信噪比而言，合成质量有所欠缺，因此全局模型对这部分样本的识别能力主要依靠各个携带真实样本信息的本地模型来提升，也就是上行蒸馏。根据式(18)可知，当 T_1 增大时，蒸馏部分的损失会提升，迫使全局模型在更新时更倾向于蒸馏过程，以获取更多关于真实样本的蒸馏信息，从而提升对中低信噪比下真实信号的识别能力。这与实际生活中的蒸馏理念一致。然而当 T_1 超过一定范围时，根据式(18)亦可知，知识被软化过度导致蒸馏信息失真，反而干扰了全局模型的参数优化，识别性能有所衰退。另一方面，对于高信噪比的调制信号而言，合成样本的质量较高，特征清晰且更接近真实样本，全局模型通过这些样本根据式(19)进行参数更新时，内嵌的半监督过程对模型的贡献度远大于蒸馏过程，使全局模型更多地依靠自身的分类学习来提升性能，因此全局模型对于高信噪比下真实信号的识别能力相对稳定，不易受 T_1 的影响。

然而，图 10(b)中下行蒸馏的温度参数 T_2 主要影响高信噪比下的调制类型识别性能，但程度较轻。同样地，随着 T_2 的增大，识别性能先上升后衰退，解释如下。从式(21)可知，下行蒸馏优化的是本地模型对缺失类别的识别能力，而真实的私有类别并不会受到全局模型的约束，也就是不受 T_2 影响。私有类别的识别性能主要由内嵌的半监督过程通过本地持有的真实私有样本进行提升，因此本地模型对私有类别的识别能力相对稳定。根据前文所述，中低信噪比下的合成样本质量相对较差，当全局模型通过这些样本去优化本地模型关于缺失类别的参数时，缺失类别的识别性能并不会有很大的提升。总体而言，本地模型对中低信噪比下全局类别的识别性能也就相对稳定，不易受 T_2 影响。



(a) 不同T_1下所提算法的识别性能曲线



(b) 不同T_2下所提算法的识别性能曲线

图 10　双向知识蒸馏过程中不同温度对所提算法性能的影响

相反，高信噪比下的合成样本质量相对较高，当全局模型通过这些样本去优化本地模型关于缺失类别的参数时，融合了多方真实私有类别信息的全局模型会将这些信息通过合成样本反馈给本地，从而提升本地模型对缺失类别的识别能力。同样地，T_2 的增大会导致式(22)蒸馏损失增大，进而提升本地模型在优化过程中对蒸馏部分的倾向性，以获得更多关于缺失类别的蒸馏信息。整体来说，全局识别性能也随之提升，而过度增大T_2 也会导致蒸馏信息的失真，削弱识别性能。

3.6　不同数量私有类别的性能分析

为研究所提算法的适用性，本文依旧在 5 个客户端的场景下，通过设置不同数量的私有类别来对比所提算法的性能，为此在这里重新划分不同客户端持有的私有类别，如表 5 所示。仿真结果如图 11 所示。

表 5　不同客户端所持数量不等的私有类别

客户端名称	类别不等 1	类别不等 2
客户端 1	QPSK	QPSK、4FSK
客户端 2	4FSK、16QAM、16PAM	16QAM、16PAM
客户端 3	4CPM、8ASK	4CPM
客户端 4	MSK	8ASK、MSK、AM 、FM
客户端 5	AM 、FM、OOK	OOK

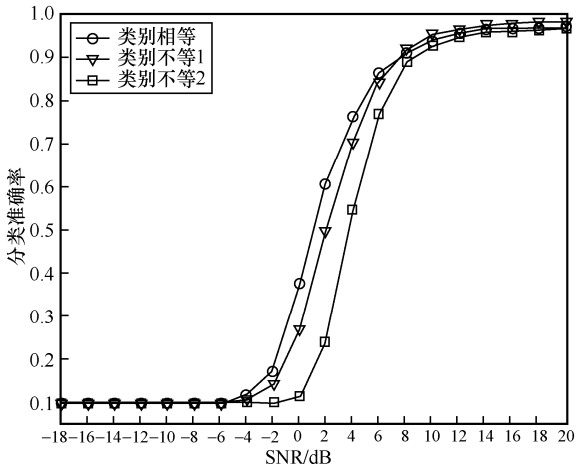


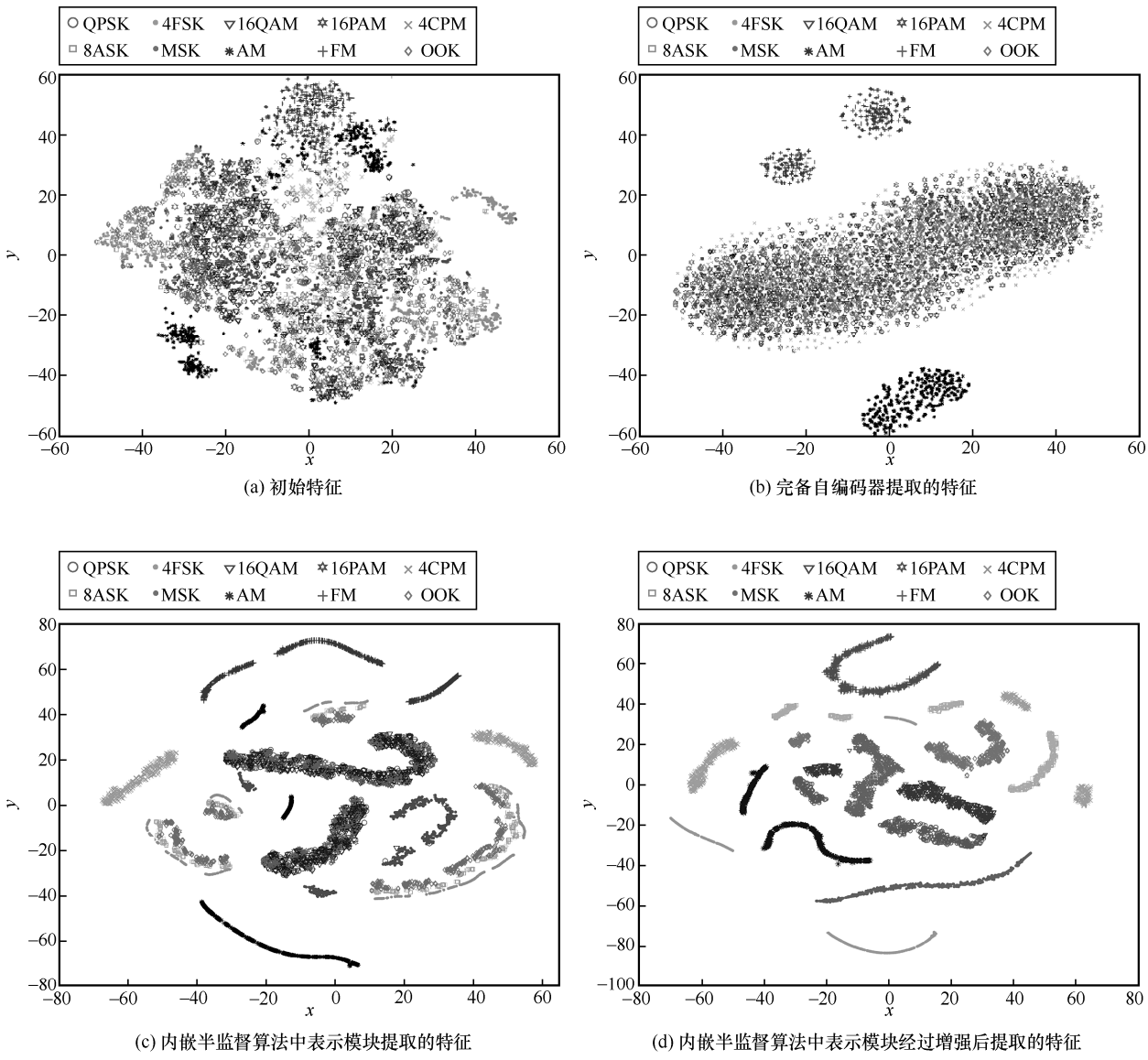
图 11　不同数量的私有类别下所提算法的识别性能曲线

从图 11 中可以看到，在 2 种私有类别数量不等的情况下，所提算法依然可以保持不错的识别能力。然而当私有类别数量存在严重不平等时，例如表 5 中“类别不等 2”的情况，客户端 4 持有 4 种私有类，客户端 3 和 5 仅持有 1 种私有类，这会加剧数据的异质性，恶化模型的性能。

3.7　特征表示效果分析

为验证内嵌半监督算法的有效性，本文通过提取 SNR=20 dB 时 5 000 个信号样本的特征，分别设置初始特征、完备自编码器提取的特征、内嵌半监督算法中表示模块提取的特征以及表示模块经过增强后提取的特征，共 4 种场景进行对比，并通过 T-SNE 降维可视化来评估内嵌半监督算法的有效性。仿真结果如图 12 所示。

自编码器是经典的无监督表示学习方法，通过对原始样本降维得到特征并以此重建样本的方式进行训练，最终获得关于样本的特征表示。从图 12(a)中可以看到，本文生成的 10 类信号样本的初始特征经过 T-SNE 降维后呈现出比较混乱的态势，而图 12(b)中自编码器经过训练后提取的特征分布有序，但并未呈现出聚类的态势。一般而言，高度区分的聚类态势能够帮助分类模块快速收敛，有效提升分类性能。由此可知，自编码器提取的特



(a) 初始特征　　(b) 完备自编码器提取的特征

(c) 内嵌半监督算法中表示模块提取的特征　　(d) 内嵌半监督算法中表示模块经过增强后提取的特征

图 12　T-sne 降维可视化后的特征表示效果对比

征不适合分类任务。得益于无标记数据中虚拟标签的引入，本文内嵌半监督算法中的表示模块仅通过无标记样本的初步表示学习，提取的特征便能呈现出聚类的态势。但是 16QAM 与 16PAM 这 2 种调制信号的特征存在混叠，如图 12(c)所示。

事实上，虚拟标签的引入迫使表示模块在训练过程中以识别样本所在集合中的位置来代替传统自编码器中以重建样本的方式进行表示学习，而在表示学习时引入虚拟分类过程，如区分样本的旋转角度等，可以让样本特征呈现出聚类的态势。因此，表示模块提取的特征聚类效果优于传统的自编码器。

此外，本节还对经过有标记样本和伪标签增强后的表示模块提取的特征进行了可视化，如图 12(d)所示。从图 12(d)中可以看到，经过增强后的表示模块提取的特征能够清晰地呈现出聚类的态势，16QAM 与 16PAM 这 2 种调制信号的特征也能清楚地分开。以上结果均证明了内嵌半监督算法的有效性。

3.8　算法复杂度分析

本文所提算法主要经历了本地设备上的样本 MixUp 增强和表示学习，以及本地与云端间的双向知识蒸馏过程，算法复杂度相对较高，为此对其进行具体分析，并同现有算法进行对比。考虑到 FedMD 算法仅利用了少量的标记样本，并未利用大量的无标记样本，算法复杂度是最低的，不具备可比性，因此并未将其纳入对比范围。本文所提 BKD-FSSL 算法中的双向蒸馏过程包含了 MixUp 增强、表示学

习等处理过程，因此算法的主要复杂度为

$$2\sum_{i=1}^{N}\left(M_i+T_i+K_i\right)\left(E_s+E_u\right) \tag{25}$$

其中，N 表示参与训练的模型数量；M_i 表示异构模型 i 在每轮训练中的浮点运算次数（FLOP, floating point of operation），用于衡量模型复杂度，由模型自身的结构、神经元算子的种类和数量等决定；E_s、E_u 分别表示模型在有标记样本集、无标记样本集下训练的轮数，$E_s \ll E_u$；T_i 表示 MixUp 增强中的矩阵运算复杂度，$T_i \ll M_i$；K_i 表示单向蒸馏的复杂度，$K_i \approx M_i$。加权系数“2”表示双向蒸馏。同理，可得 FedMatch 算法的主要复杂度为

$$\sum_{i=1}^{N}\left(M_iE_s+M_iE_u+\sum_{j=1}^{K}M_jE_u+\sum_{j=1}^{K}D_jE_u\right) \tag{26}$$

其中，$\sum_{j=1}^{K}M_j$ 表示 FedMatch 中协同 K 个设备进行伪标记时的复杂度，$2 \leqslant K \leqslant N$；$\sum_{j=1}^{K}D_jE_u$ 表示一致性正则化时的复杂度，$D_j \approx M_j$。根据式(25)和式(26)可得

$$\frac{2\left(M_i+T_i+K_i\right)\left(E_s+E_u\right)}{M_iE_s+M_iE_u+\sum_{j=1}^{K}M_jE_u+\sum_{j=1}^{K}D_jE_u} \approx \frac{2M_i+2K_i}{M_i+\sum_{j=1}^{K}M_j+\sum_{j=1}^{K}D_j} < 1 \tag{27}$$

因此，本文所提 BKD-FSSL 算法的复杂度低于 FedMatch。而 Semi-HFL 的复杂度为

$$\sum_{i=1}^{N}\left(M_iE_s+M_iE_u+C\right) \tag{28}$$

其中，C 表示 Semi-HFL 中级联本地异构模型的复杂度 $C \ll M_i$。同样地，根据式(25)和式(28)可得

$$\frac{M_iE_s+M_iE_u+C}{2\left(M_i+T_i+K_i\right)\left(E_s+E_u\right)} \approx \frac{M_i}{2\left(M_i+K_i\right)} < 1 \tag{29}$$

因此，Semi-HFL 的复杂度低于所提算法。

综上可得，不同算法的复杂度关系为

$$O\left(\text{Semi-HFL}\right) < O\left(\text{BKD-FSSL}\right) < O\left(\text{FedMatch}\right) \tag{30}$$

此外，本文还通过算法达到收敛时的迭代平均耗时来验证算法的复杂度，具体结果如表 6 所示。此外，表 6 中还列出了不同算法在收敛时的平均识别准确率 $P_{\text{acc}}^{\text{avg}}$，来进行直观对比。

从表 6 中可以看到，所提算法的分类性能是最优的，平均耗时是中等的。虽然所提算法看似复杂，但是部分处理过程，例如，MixUp 增强仅是矩阵的加减操作，并不涉及模型中大量神经元的推理过程；FedMatch 中的伪标签环节需要依靠多个设备共同完成，也就是说当前设备上的模型在训练过程中，还会存在其他多个本地模型的推理过程，这会大大增加算法的时间复杂度，因此该算法的运行耗时是最长的。相反，所提算法和 Semi-HFL 中的伪标签环节都是依靠自身模型完成的，同时在该环节中还会并行处理其他过程。此外，Semi-HFL 仅级联本地异构模型和聚合相同结构的处理过程相对较复杂，其他环节均与普通训练类似，因此该算法的平均耗时是最短的。

4 结束语

本文提出了一种适用于数据异质和模型异构同时存在场景下的异构联邦半监督调制类型识别算法，设计了内嵌半监督算法的双向知识蒸馏机制实现本地模型和全局模型之间的协同训练。通过“本地−云端”的上行蒸馏来集成携带真实私有样本信息的本地异构模型，辅助全局模型的参数优化；“云端−本地”的下行蒸馏反馈全局模型关于本地设备上异质数据的知识，以选择性蒸馏的方式提升本地模型对异质数据的识别能力。内嵌半监督算法通过对无标记数据进行虚拟分类获得高质量的特征表示并利用少量有标记数据和伪标签强化，同步提升模型性能。此外，本文还利用 VAE 在云端构建可共享的有标记公开集和无标记公开集，缓解不同本地设备上的数据异质性，并作为云端−本地协同训练的媒介，辅助双向知识蒸馏的运行。仿真结果表明，在数据异质和模型异构同时存在的场景中，本文所提算法比现有算法具备更出色的调制类型识别精度。

表 6 不同算法的迭代平均耗时和平均识别准确率对比

算法	2 个设备		3 个设备		5 个设备	
	平均耗时/s	$P_{\text{acc}}^{\text{avg}}$	平均耗时/s	$P_{\text{acc}}^{\text{avg}}$	平均耗时/s	$P_{\text{acc}}^{\text{avg}}$
FedMatch	973.55	58.57%	1 980.11	48.81%	5 403.51	46.14%
Semi-HFL	**123.97**	63.56%	**161.66**	49.28%	**432.94**	46.87%
BKD-FSSL	263.91	**65.25%**	536.89	**52.03%**	1 368.76	**51.38%**

附录 1　半监督 SSL 算法

输入　有标记数据集 D_L，无标记数据集 D_U，测试集 D_T，初始化模型 $f(\boldsymbol{\theta},\boldsymbol{\beta})$，停止条件控制参数 T

输出　模型 $f(\boldsymbol{\theta},\boldsymbol{\beta})$

$t=0$;
while $t<T$ do
　随机选取子集 $B_L\subseteq D_L$，$B_U\subseteq D_U$；对 B_U 进行随机排序获得 $B_U^{'}$；
　for all $\boldsymbol{x}_i\in B_U,\boldsymbol{x}_i^{'}\in B_U^{'}$ do
　　根据 $\boldsymbol{x}_i$ 在 B_U 中的位置创建虚拟标签 v_i，根据式(9)、式(10)、式(13)和式(14)计算 Umix 损失 L_{Umix} 更新表示模块 $f(\boldsymbol{\theta})$；
　end for
　for all $(\boldsymbol{x}_j,y_j)\in B_L$ do
　　根据式(10)和式(11)计算 Smix 损失 L_{Smix} 更新完整模型 $f(\boldsymbol{\theta},\boldsymbol{\beta})$；
　end for
　根据式(15)对 B_U 标注伪标签，得到 B_U^{pseL}；
　for all $(\boldsymbol{x}_i,y_i)\in B_L\cup B_U^{\text{pseL}}$ do
　　if $y_i>0$ then
　　　根据式(10)和式(11)计算伪标签损失 $L_{\text{Smix-Pse}}$ 强化完整模型 $f(\boldsymbol{\theta},\boldsymbol{\beta})$；
　　end if
　end for
　$t=t$+1;
end while

附录 2　上行蒸馏 UpKD 算法

输入　有标记伪样本集 $D_{\text{pseL}}^{\text{glb}}$，无标记伪样本集 $D_{\text{pseU}}^{\text{glb}}$，测试集 D_{glb}，初始 $f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}})$ 和 K 个本地模型 $f_k(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k)$，其中 $k\in\{1,2,\cdots,K\}$，停止条件控制参数 T

输出　全局模型 $f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}})$

$t=0$;
while $t<T$ do
云端服务器初始化：
　随机选取子集 $B_L\subseteq D_{\text{pseL}}^{\text{glb}}$，$B_U\subseteq D_{\text{pseU}}^{\text{glb}}$；llogits $\leftarrow\varnothing$，ulogits $\leftarrow\varnothing$；
本地设备：
　for $k=1:1:K$ do
　　for all $\boldsymbol{x}_i\in B_L$ do
　　　根据式(16)计算 $\boldsymbol{p}_{k,i}$ 并上传至云端，更新 llogits $\leftarrow$ llogits $\cup\{(\boldsymbol{x}_i,\boldsymbol{p}_{k,i},y_i)\}$；
　　end for
　　for all $\boldsymbol{x}_j\in B_U$ do
　　　根据式(16)计算 $\boldsymbol{p}_{k,j}$ 并上传至云端，更新 ulogits $\leftarrow$ ulogits $\cup\{(\boldsymbol{x}_j,\boldsymbol{p}_{k,j})\}$；
　　end for
　end for
云端服务器更新：
　for all $(\boldsymbol{x}_i,\boldsymbol{p}_{k,i},y_i)\in$ llogits & $(\boldsymbol{x}_j,\boldsymbol{p}_{k,j})\in$ ulogits do
　　根据式(17)和式(18)计算 L_d；
　　计算 $L_{\text{Umix}},L_{\text{Smix}},L_{\text{Smix-Pse}}\leftarrow\text{SSL}\left(\boldsymbol{x}_i,\boldsymbol{x}_j,f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}})\right)$；
　　根据式(19)计算 L_{up} 更新全局模型 $f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}})$；
　end for
　$t=t$+1;
end while

附录 3　下行蒸馏 DownKD 算法

输入　有标记样本集 $D_L^k,D_{\text{pseL}}^{\text{glb}'}$，无标记样本集 $D_U^k,D_{\text{pseU}}^{\text{glb}'}$，测试集 D_{glb}，全局 $f_{\text{glb}}(\boldsymbol{\theta}_{\text{glb}},\boldsymbol{\beta}_{\text{glb}})$ 和 K 个本地模型 $f_k(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k)$，其中 $k\in\{1,2,\cdots,K\}$，停止条件控制参数 T

输出　本地模型集 $\{f_k(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k)\}_{k=1}^K$

$t=0$;
while $t<T$ do
本地设备：
　随机选取子集：$B_L^k\subseteq D_L^k\cup D_{\text{pseL}}^{\text{glb}'}$，$B_U^k\subseteq D_U^k\cup D_{\text{pseU}}^{\text{glb}'}$；
　for $k=1:1:K$ do
　　for all $\boldsymbol{x}_i^k\in B_L^k,\boldsymbol{x}_j^k\in B_U^k$ do
　　　根据式(16)、式(21)和式(22)计算缺失类别 $C_{\text{glb}}-C_k$ 的蒸馏损失 L_D^k；
　　　计算 $L_{\text{Umix}}^k,L_{\text{Smix}}^k,L_{\text{Smix-Pse}}^{\text{k}}\leftarrow\text{SSL}\left(\boldsymbol{x}_i^k,\boldsymbol{x}_j^k,f_k(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k)\right)$；
　　　根据式(23)计算 L_{down}^k 更新本地模型 $f_k(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k)$；
　　end for
　end for
　$t=t$+1;
end while

参考文献：

[1] DOBRE O A, ABDI A, BAR-NESS Y, et al. Survey of automatic modulation classification techniques: classical approaches and new trends[J]. IET communications, 2007, 1(2): 137-156.

[2] XU J L, SU W, ZHOU M C. Likelihood-ratio approaches to automatic modulation classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2011, 41(4): 455-469.

[3] ZHANG Z F, WANG C, GAN C Q, et al. Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD[J]. IEEE Transactions on Signal and Information Processing Over Networks, 2019, 5(3): 469-478.

[4] BONAWITZ K, EICHNER H, GRIESKAMP W, et al. Towards federated learning at scale: system design[J]. arXiv Preprint, arXiv: 1902.01046, 2019.

[5] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning[J]. Foundations and Trends in Machine Learning, 2021, 14(1/2): 1-210.

[6] WANG X M, CHEN W, XIA J Z, et al. HetVis: a visual analysis approach for identifying data heterogeneity in horizontal federated learning[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(1): 310-319.

[7] LI G H, HU Y, ZHANG M, et al. FedGosp: a novel framework of gossip federated learning for data heterogeneity[C]//Proceedings of 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Piscataway: IEEE Press, 2022: 840-845.

[8] LONG G D, SHEN T, TAN Y, et al. Federated learning for privacy-preserving open innovation future on digital health[M]. Cham: Springer, 2022.

[9] LU X F, LIAO Y Y, LIU C, et al. Heterogeneous model fusion federated learning mechanism based on model mapping[J]. IEEE Internet of Things Journal, 2022, 9(8): 6058-6068.

[10] HU L, YAN H, LI L, et al. MHAT: an efficient model-heterogenous aggregation training scheme for federated learning[J]. Information Sciences, 2021, 560: 493-503.

[11] SHI J B, ZHAO H J, WANG M Y, et al. Signal recognition based on federated learning[C]//Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE Press, 2020: 1105-1110.

[12] SHI J B, QI L, LI K X, et al. Signal modulation recognition method based on differential privacy federated learning[J]. Wireless Communications and Mobile Computing, 2021, 2021: 1-13.

[13] WANG Y, GUI G, GACANIN H, et al. Federated learning for automatic modulation classification under class imbalance and varying noise condition[J]. IEEE Transactions on Cognitive Communications and Networking, 2022, 8(1): 86-96.

[14] QI P H, ZHOU X Y, DING Y L, et al. Collaborative and incremental learning for modulation classification with heterogeneous local dataset in cognitive IoT[J]. IEEE Transactions on Green Communications and Networking, 2023, 7(2): 881-893.

[15] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv Preprint, arXiv: 1503.02531, 2015.

[16] LI D, WANG J. FedMD: heterogenous federated learning via model distillation[J]. arXiv Preprint, arXiv: 1910.03581, 2019.

[17] JEONG W, YOON J, YANG E, et al. Federated semi-supervised learning with inter-client consistency & disjoint learning[C]//Proceedings of 9th International Conference on Learning Representations (ICLR 2021). Piscataway: IEEE Press, 2021: 1-5.

[18] ZHONG Z Y, WANG J, BAO W D, et al. Semi-HFL: semi-supervised federated learning for heterogeneous devices[J]. Complex & Intelligent Systems, 2023, 9(2): 1995-2017.

[19] KINGMA D P, WELLING M. An introduction to variational autoencoders[M]. [S.l.]: Now Publishers, 2019.

[20] BENGIO Y, COURVILLE A, VINCENT P. Representation learning: a review and new perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.

[21] SOHN K, BERTHELOT D, LI C, et al. FixMatch: simplifying semi-supervised learning with consistency and confidence[J]. arXiv Preprint, arXiv: 2001.07685, 2020.

[22] FENG S, LI B, YU H, et al. Semi-supervised federated heterogeneous transfer learning[J]. Knowledge-Based Systems, 2022, 252: 109384.

[作者简介]

齐佩汉（1986− ），男，河南永城人，博士，西安电子科技大学教授，主要研究方向为电磁空间智能感知、电磁频谱空间安全与综合利用、通信信号处理与对抗等。

丁渊磊（1998− ），男，浙江嘉兴人，西安电子科技大学硕士生，主要研究方向为联邦学习、数字信号处理等。

尹凯（1994− ），男，河南平顶山人，西安电子科技大学硕士生，主要研究方向为通信电子对抗、数字信号处理等。

徐佳波（2000− ），男，浙江杭州人，西安电子科技大学硕士生，主要研究方向为异常检测、数字信号处理等。

李赞（1975− ），女，陕西西安人，博士，西安电子科技大学教授，主要研究方向为电磁频谱认知、高安全高可靠通信、数字信号处理、无线通信系统等。