FedCP：通过条件策略分离特征信息以实现个性化联邦学习

摘要

近年来，个性化联邦学习（pFL）在隐私保护、协同学习以及应对客户端（如医院、移动智能手机等）间的统计异质性等方面受到了越来越多的关注。大多数现有的个性化联邦学习方法主要聚焦于利用客户端模型参数中的全局信息和个性化信息，却忽视了数据才是这两类信息的源头。为解决这一问题，我们提出了联邦条件策略（FedCP）方法。该方法为每个样本生成一个条件策略，用于分离其特征中的全局信息和个性化信息，随后分别通过一个全局头和一个个性化头对这两类信息进行处理。与现有的个性化联邦学习方法相比，FedCP 能以样本特定的方式更细粒度地考虑个性化问题。

关键词：联邦学习；统计异质性；个性化；条件计算；特征分离

1 引言

个性化联邦学习（PFL）是联邦学习领域中一个迅速发展的研究方向，它致力于在保护数据隐私的同时，实现模型对不同客户端的个性化适应。随着数据量的不断增长和数据隐私法规的日益严格，传统的集中式机器学习方法面临着数据孤岛和隐私泄露的风险。联邦学习的出现为解决这些问题提供了一种有效的分布式学习框架，允许各客户端在不共享原始数据的情况下共同训练模型 [13]。

然而，联邦学习在实际应用中仍面临诸多挑战。不同客户端的数据往往具有异构性，即数据分布、特征和标签等方面存在差异，这使得单一的全局模型难以在所有客户端上都达到最优性能 [19]。此外，通信开销和计算资源限制也是联邦学习面临的重要问题，频繁的模型参数交换和大规模的模型训练对网络带宽和设备计算能力提出了很高的要求 [9]。

在这样的背景下，个性化联邦学习应运而生。它通过为每个客户端定制个性化的模型，能够更好地适应不同客户端的数据特点，从而提高模型的准确性和泛化能力 [16]。同时，个性化联邦学习也有助于提高联邦学习的效率，减少不必要的通信开销和计算资源浪费 [10]。现有大多数 PFL 方法主要关注客户端级模型参数中的全局和个性化信息，如元学习方法仅微调全局模型参数以适应本地数据 [7]，正则化方法在本地训练期间仅对模型参数进行正则化 [2]，个性化头方法虽将骨干网络拆分为全局和个性化部分，但仍侧重于模型参数中的信息，而忽略了数据作为这些信息的来源 [3]。数据中的异构性包含了全局和个性化信息，而现有方法未能充分挖掘这一点。

而 FedCP 方法基于条件计算技术，FedCP 提出在特征向量层面分离全局和个性化信息，因为原始输入数据维度远大于特征向量，专注于特征向量可提高效率 [6]。通过引入辅助的条

件策略网络（CPN），根据样本和客户端的不同，生成样本特定的策略，从而实现特征信息的分离 [15]。在客户端上，FedCP 将骨干网络拆分为特征提取器和头两部分，并为每个客户端设置全局特征提取器、全局头、个性化特征提取器、个性化头和 CPN。在训练过程中，CPN 根据样本特定输入（由样本特征向量和客户端特定向量组合而成）生成策略，将特征向量分离为全局和个性化特征信息，分别输入全局头和个性化头进行处理。同时，通过冻结全局头保留全局信息，并使用最大平均差异（MMD）损失对齐个性化特征提取器输出与全局特征提取器输出，以确保特征适配 [12]。最终，通过端到端学习，CPN 能够自动学习生成有效的样本特定策略。FedCP 首次在联邦学习中考虑基于样本特定特征信息的个性化，比大多数现有方法仅使用客户端级模型参数更精细，能够更好地满足每个客户端的个性化需求，从而提高模型在个性化任务上的性能 [14]。

FedCP 的主要工作概述如下：

样本级特征信息分离：提出一种基于条件计算技术的联邦条件策略（FedCP）方法。针对特征向量，通过引入辅助的条件策略网络（CPN），为每个样本生成特定策略，以分离特征中的全局信息和个性化信息。CPN 根据样本特定输入（结合样本特征向量和客户端特定向量）生成策略，将特征向量分解为全局和个性化特征信息，分别输入全局头和个性化头进行处理 [17]。

高效的模型架构设计：在客户端模型架构上，将骨干网络拆分为特征提取器和头两部分，并为每个客户端配备全局特征提取器、全局头、个性化特征提取器、个性化头和 CPN。训练时，CPN 学习生成样本特定策略，实现特征信息分离；冻结全局头保留全局信息，通过 MMD 损失对齐个性化特征提取器输出与全局特征提取器输出，确保特征适配 [4]。

2 相关工作

此部分对课题内容相关的工作进行简要的分类概括与描述，二级标题中的内容为示意，可按照行文内容进行增删与更改，若二级标题无法对描述内容进行概括，可自行增加三级标题，后面内容同样如此，引文的 bib 文件统一粘贴到 **refs.bib** 中并采用如下引用方式 [11]。

2.1 传统联邦平均算法

经典联邦学习算法由一个中央服务器协调，数据分布在各个边缘设备，意味着客户端上的数据无需上传到中央服务器，客户端对自己拥有的数据掌握着绝对的控制权。由于客户端数据分布不同以及为了保证效率或隐私保护而采用压缩或加密等特殊计算，联邦学习训练的模型会存在精度损失，其中任意小的正数 用于衡量误差大小。优秀的联邦学习算法旨在尽可能减少这种精度损失 [13]。联邦学习的基本流程可以分为 3 个步骤：问题建模、局部计算和全局聚合。首先由中心服务器根据参与方数据类型和应用需求设计模型并进行初始化，然后将其发送到各个客户端开始训练。联邦学习的训练过程是一个迭代优化的过程，局部计算和全局聚合称为一个计算轮次，在每轮迭代计算过程中，首先由客户端利用其本地的隐私数据集对模型参数进行更新，更新结束之后将其上传至服务器，服务器接收到各个客户端上传的模型信息之后，将统一进行整合得到最新的模型，最终在全局模型收敛到给定精度或者迭代完成固定轮次之后，训练过程停止 [9]。

2.2 联邦学习中的异构数据

联邦学习中，每个客户端上训练数据的分布情况很大程度上取决于这个客户端设备的使用情况，因此，客户端之间的数据分布可能完全不同。这种现象称之为 Non-iid，这种情况会导致客户端之间的模型产生严重的差异性，特别是在横向联邦学习中。具体来说，对于在客户端 k 的一个有监督学习任务，每一个数据样本由输入的属性或者特征以及标签信息来表示，这些样本遵循一个本地的数据分布。在 Non-iid 的情况下，意味着每个客户端之间的数据分布是存在差异性的，这种数据被称作异构数据 [19]。联邦学习中 Non-iid 数据分布主要分为属性倾斜和标签倾斜。属性也被称之为特征，通常作为机器学习模型的输入或者是决策域，属性倾斜意味着每个客户端上的特征分布是彼此不同的。一般来说，属性倾斜主要包括客户端之间数据属性的部分重叠和完全不重叠，完全不重叠的属性倾斜意味着不同客户端之间的数据特征分布是互斥的，部分重叠的属性倾斜意味着不同客户端之间的数据特征的某些部分是可以彼此共享的。标签倾斜表示不同客户端的标签分布是存在差异的。标签倾斜有两种情况，一种是标签分布倾斜，另一种是标签偏好倾斜。标签分布倾斜通常是由于存储类似训练数据的客户端位置的差异所导致的，有关联邦学习的研究中考虑了两种主要的标签分布倾斜，即标签大小不平衡和标签分布不平衡，标签大小不平衡指的是不同客户端拥有的数据的标签类别是不平衡的，标签分布不平衡是另一种情况，这种情况主要是指每个客户端中含有每一类标签的样本个数是不平衡的。标签偏好倾斜考虑到了联邦学习在实际应用中经常遇到的客户端数据样本的交叉问题，假设不同客户端之间的训练数据是水平重叠的，则不同的用户对于同一个数据样本可能给出不一样的标签，因为每个用户都会有自己的个人偏好 [9]。

2.3 个性化联邦学习算法

在模型优化技术方面，参数解耦通过分离局部和全局模型参数来提升个性化，如 Mei 等人及 Ma 等人的相关研究 [12]；知识蒸馏借助从教师模型向学生模型转移知识来优化，Ni 等人和 Chen 等人的研究体现了其应用 [14]；神经架构搜索旨在自动发现有效神经网络架构，但在 PFL 中面临数据和系统异质性等挑战，Yu 等人及 Wan 等人的研究有所涉及 [18]；超参数优化为每个客户端寻找最优超参数，Subramian 等人和 Cheng 等人对此进行了探索 [5]；数据增强通过生成新数据提升模型性能，Lu 等人和 Yang 等人的研究展示了其在处理数据异质性等方面的作用 [8]；正则化用于减轻过拟合，Huang 等人和 Li 等人的框架体现了其价值 [10]；对抗训练增强模型隐私和鲁棒性，Lu 等人及 Zhang 等人的研究关注了相关问题 [17]；元学习通过接触多样化任务提高学习算法，Jiang 等人和 Dinh 等人的工作有所贡献 [7]；聚类通过分组相似客户端提高 FL 效率，Werner 等人和 Yoo 等人的研究涉及此方面 [6]。

然而，这些技术也存在局限。参数解耦在支持模型设计个性化上有局限，知识蒸馏面临模型可解释性等问题，神经架构搜索需解决在 PFL 中的适应性问题，超参数优化受限于通信和计算资源，数据增强可能引入失真等，正则化手动调参困难，对抗训练计算密集且面临对手模型设计难题，元学习在数据获取和计算复杂度上存在挑战，聚类计算和通信成本高 [16]。

3 本文方法

3.1 本文方法概述

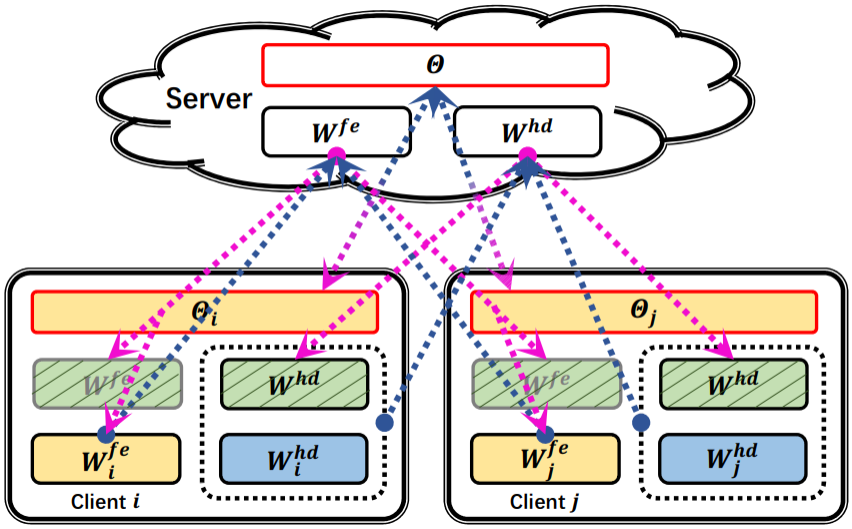

图 1. FedCP 系统中的数据流

首先，在学习场景中，FedCP 被应用于统计异构的 pFL 环境，其中 NN 个客户端各自拥有来自不同分布且无重叠的私有数据集，并通过协作方式训练各自的个性化模型。模型结构方面，FedCP 与部分现有方法相似，将模型主干分为特征提取器和头部，但增加了独特的组件：每个客户端包含全局特征提取器、全局头、个性化特征提取器、个性化头，以及条件策略网络（CPN）。在训练流程中，每次迭代开始时，客户端用新的全局参数覆盖个性化特征提取器参数，同时冻结全局特征提取器和全局头，如图 1 所示。在训练过程中，个性化特征提取器通过最大均值差异（MMD）损失来对齐其输出特征与全局特征提取器的输出。在推理阶段，个性化模型由个性化特征提取器、全局头、个性化头和 CPN 组成。最终，FedCP 的目标函数旨在找到一个个性化模型集合，使整体损失最小化，其中整体损失通常是各客户端局部损失的加权和 [4]。

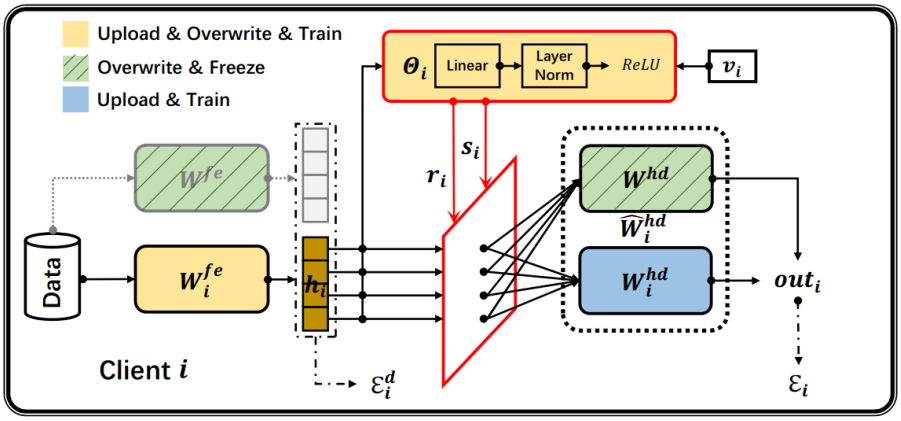

图 2. 转发与客户端 i 上的本地学习相对应的数据流

在客户端 i 上，我们有一个全局特征提取器（参数为）、一个全局头（参数为）、一个个性化特征提取器（参数为）、一个个性化头（参数为）和一个条件策略网络（CPN，参数为）。具体而言，对于特征提取器，我们在每次迭代中用相应的全局参数覆盖来对其进行初始化，然后在本地训练个性化特征提取器。在本地学习过程中，不断变化的个性化特征提取器所生成的特征可能与冻结的全局头不匹配。因此，在接收到参数后，我们会冻结全局特征提取器，并通过 MMD 损失使个性化特征提取器输出的特征与全局特征提取器生成的特征对齐，如图 2 所示。对于全局头，在使用对其初始化后，我们会将其冻结以保留全局信息。简而言之，在每次迭代开始时，我们用新的覆盖，然后冻结和。如图 2 中的非透明模块所示，用于推理的个性化模型（参数为）由个性化特征提取器、全局头、个性化头和 CPN 组成，即。冻结的全局特征提取器仅用于本地学习，并非个性化模型的一部分 [17]。

3.2 联合条件策略（FedCP）

由于统计异质性，h R 包含全局和个性化特征信息。为了分别利用这两种类型的信息，我们提出了 FedCP，它以端到端的方式学习特定于样品的分离，在冻结的全局 HEAD 和个性化 HEAD 中的个性化信息中，CPN（FedCP 的核心）可以学习自动生成样本特定策略，并在分离全局和个性化信息。在统计异构的联邦学习设置中，我们致力于分离特征向量中的信息。设 $h_i = f(x_i; W_i^{fe})$，其中 $(x_i, y_i) \in \mathcal{D}_i$，由于数据的统计异构性，$h_i \in \mathbb{R}^K$ 包含了全局和个性化的特征信息。为了分别利用这些信息，我们提出了 FedCP 方法，它以端到端的方式学习样本特定的分离策略。

CPN（条件策略网络）是 FedCP 的核心，它由一个全连接层、一个层归一化层和 ReLU 激活函数串联组成。在客户端 i 上，CPN 根据样本特定输入 C_i 生成样本特定策略 $\{r_i, s_i\}$，具体过程如下：

设 $C_i \in \mathbb{R}^K$ 为 CPN 的样本特定输入，其生成方式为结合样本特定向量 h_i 和客户端特定向量 v_i。客户端特定向量 v_i 通过降低个性化头参数 w_i^{hd}（在 FedCP 中，头为全连接层，即 $W_i^{hd} \in \mathbb{R}^{C \times K}$）的维度获得，具体计算为 $v_i := \sum_{c=1}^{C} w_c^T$，其中 w_c 是 w_i^{hd} 的第 c 行，从而得到与 h_i 形状相同且具有特征语义的向量。然后，$C_i := (v_i / \|v_i\|_2) \odot h_i$，其中 $\|v_i\|_2$ 是 v_i 的 ℓ_2-范数，$\odot$ 表示哈达玛积。在每次迭代的本地学习前获取 v_i，并在训练期间将其视为常数，推理时重用最新的 v_i。

CPN 生成中间变量 $a_i \in \mathbb{R}^{K \times 2}$，其中 $a_i^k = \{a_{i,1}^k, a_{i,2}^k\}$，$k \in [K]$，且 $a_{i,1}^k$ 和 $a_{i,2}^k$ 无约束。然后通过 softmax 操作得到 r_i^k 和 s_i^k，公式为：

$$r_i^k = \frac{\exp(a_{i,1}^k)}{\sum_{j \in \{1,2\}} \exp(a_{i,j}^k)}, \quad s_i^k = \frac{\exp(a_{i,2}^k)}{\sum_{j \in \{1,2\}} \exp(a_{i,j}^k)}$$

满足 $r_i^k + s_i^k = 1$ 且 $r_i^k, s_i^k \in (0, 1)$，$\forall k \in [K]$。通过将策略 $\{r_i, s_i\}$ 与特征向量 h_i 相乘，得到全局特征信息 $r_i \odot h_i$ 和个性化特征信息 $s_i \odot h_i$。将分离后的全局特征信息 $r_i \odot h_i$ 和个性化特征信息 $s_i \odot h_i$ 分别输入全局头和个性化头。全局头的输出为 $out_i^r = g(r_i \odot h_i; W^{hd})$，个性化头的输出为 $out_i^s = g(s_i \odot h_i; W_i^{hd})$。最终输出定义为 $out_i := out_i^r + out_i^s$，局部损失为 $\mathcal{E}_i = \mathbb{E}_{(x_i, y_i) \sim \mathcal{D}_i} \mathcal{L}(out_i, y_i)$，其中 $\mathcal{L}$ 为交叉熵损失函数。从每个样本的角度来看，提取的特征由全局头和个性化头共同处理。为简化起见，我们通过平均聚合这两个头来形成上传头 $\widehat{W}_i^{hd}$，公式为 $\widehat{W}_i^{hd} = \frac{W^{hd} + W_i^{hd}}{2}$。在每次迭代中，将 $\{W_i^{fe}, \widehat{W}_i^{hd}, \Theta_i\}$ 上传到服务器。为了使个性化特

征提取器输出的特征与冻结的全局头相匹配，我们通过最大平均差异（MMD）损失 $\mathcal{E}_i^d$ 来对齐两者的输出，公式为 $\mathcal{E}_i^d = \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}_i}\kappa[h_i, f(x_i; W^{fe})]$，其中 κ 为径向基函数（RBF）核。最终的局部损失为 $F_i = \mathcal{E}_i + \lambda\mathcal{E}_i^d$，其中 λ 是一个超参数，用于控制 MMD 损失的重要性。

3.3 隐私分析

在联邦学习场景中，隐私保护至关重要。本部分旨在剖析 FedCP 方法在保护数据隐私方面的能力与机制。

在 FedCP 里，参数共享方式对隐私保护意义重大。服务器与客户端之间共享一个特征提取器、一个头以及一个条件策略网络（CPN）的参数。对于头部分，于每个客户端上，我们通过特定方式聚合全局头参数 w^{hd} 与个性化头参数 w_i^{hd} 得到 $\widehat{W}_i^{hd}$ 后再上传。此过程可类比为在上传的个性化头参数中融入全局参数所形成的“噪声”，从而在上传与下载环节为隐私提供一定程度的保护。

FedCP 的样本特定特征进一步增强了隐私保护能力。一方面，由于样本特定输入 C_i 动态生成且不与服务器共享，这使得攻击者难以借助 CPN 或模型反转攻击恢复样本特定策略。另一方面，若缺失样本特定策略，特征提取器与头之间的连接将被切断，极大增加了基于共享模型参数展开攻击的难度。

4 复现细节

4.1 与已有开源代码对比

文章作者公开了实验的源代码,为复现实验提供了基础: https://github.com/TsingZ0/FedCP [1]。然而，为了更高效地进行复现相关的对比实验，并拓展对不同数据集的支持，本人对作者的源代码进行了重构，整合出一套更为通用和灵活的实验框架。原代码结构在功能模块的划分上不够清晰，这给后续的扩展和维护带来了一定困难。本人重新组织了代码结构，将其划分为数据处理、模型构建、训练过程、评估指标计算以及结果可视化等独立的模块。这样的划分使得代码逻辑更加清晰，每个模块的功能单一且明确，便于理解和修改。例如，在数据处理模块中，将数据集的加载、预处理（如归一化、数据增强等）操作封装在独立的函数中，提高了代码的可读性和复用性。原实验仅针对特定的数据集进行测试，为了拓展对数据集的支持，本人在新的实验框架中设计了通用的数据接口。通过这个接口，能够方便地接入其他常见的联邦学习数据集，如 TinyImageNet、Cifar100 等图像数据集，以及 AgNews 等文本数据集。在数据加载过程中，根据数据集的特点自动进行相应的预处理操作，确保数据能够适用于模型的训练和评估。

4.2 实验环境搭建

代码基于 tytorch 深度学习框架实现，使用 2.0.1 版本，cuda 版本 11.7, 本地运行在单张 4060ti 16GB 显卡上，Python==3.11，采用了 Conda 虚拟环境来确保实验环境的独立性和稳定性。

4.3 使用说明

代码使用方法说明:

以下是数据集预处理的使用流程。在执行代码之前，请确保已处于正确的工作目录下，即./dataset 目录中，以手写数字数据集 MNIST 为例。

1. 生成独立同分布（IID）且不平衡的 MNIST 数据集

   - 执行命令：python generate_MNIST.py iid。此命令将按照独立同分布且不平衡的方式生成 MNIST 数据集。在这种场景下，数据虽然是从相同的分布中采样，但在各个客户端之间可能存在数据量不平衡的情况。这对于模拟现实中客户端数据分布不均匀的联邦学习场景具有重要意义，例如不同的设备或用户可能拥有不同数量的数据样本。

2. 生成独立同分布（IID）且平衡的 MNIST 数据集

   - 执行命令：python generate_MNIST.py iid balance。通过添加 balance 参数，代码将生成独立同分布且平衡的 MNIST 数据集。这意味着每个客户端将获得大致相同数量的数据样本，且数据均来自相同的分布。这种场景常用于对比实验，以评估在理想数据平衡情况下模型的性能，从而更好地理解数据不平衡对模型的影响。

3. 生成病理非独立同分布（Pathological Non - IID）且不平衡的 MNIST 数据集

   - 执行命令：python generate_MNIST.py noniid - pat。该命令会根据病理非独立同分布的策略生成 MNIST 数据集，并且数据集是不平衡的。病理非独立同分布是一种特殊的非独立同分布情况，常用于研究联邦学习算法在极端数据分布差异下的性能。例如，不同客户端的数据可能集中在不同的类别或特征空间，模拟了现实中数据分布严重倾斜的情况。

4. 生成实际非独立同分布（Practical Non - IID）且不平衡的 MNIST 数据集

   - 执行命令：python generate_MNIST.py noniid - dir。这将使用实际非独立同分布策略（通常基于 Dirichlet 分布）生成不平衡的 MNIST 数据集。这种策略更贴近现实世界中的数据分布情况，即客户端的数据虽然来自不同分布，但具有一定的相似性和关联性。在实际的联邦学习应用中，如跨不同地区或用户群体的数据协作学习，数据往往呈现这种实际非独立同分布且不平衡的特征。

5. 生成扩展 Dirichlet 策略（Extended Dirichlet Strategy）下的 MNIST 数据集

   - 执行命令：python generate_MNIST.py noniid - exdir。此命令将采用扩展 Dirichlet 策略生成 MNIST 数据集，用于探索在更复杂的数据分布假设下联邦学习模型的行为。扩展 Dirichlet 策略可能在数据分布的多样性、复杂性或动态性方面进行了进一步的拓展，以研究算法在更具挑战性的数据环境中的适应性和性能表现。

通过运行这些不同的命令，可以根据具体的实验目的和需求生成相应的 MNIST 数据集，为后续的联邦学习模型训练、测试和分析提供合适的数据基础。

实验通过运行 python main.py 命令来启动主程序，并通过一系列参数来定制实验的具体设置。

1. -data MNIST 参数：

   - 该参数指定了在本次实验中使用的数据集为 MNIST 数据集。MNIST 数据集是一个广泛用于机器学习和深度学习实验的手写数字图像数据集。它包含了大量的训练和测试样本，这些样本涵盖了从 0 到 9 的手写数字图像，适用于多种图像分类任务的研究和模型评估。无论是对基础图像分类算法的测试，还是对新提出的模型改进策略的验证，MNIST 数据集都提供了一个标准且丰富的实验数据来源。

2. -m CNN 参数：

   - 此参数明确了实验所采用的模型架构为卷积神经网络（CNN）。CNN 在处理图像数据方面具有独特的优势。其卷积层能够自动提取图像的特征，例如边缘、纹理等局部特征，通过卷积核在图像上滑动并进行卷积操作，有效地捕捉图像中的关键信息。这一特性大大减少了模型的参数数量，相比于全连接网络，降低了计算复杂度和过拟合风险。同时，CNN 的结构设计使其能够更好地适应图像数据的二维结构，提高了模型的训练效率和泛化能力。在 MNIST 数据集的图像分类任务中，CNN 可以有效地学习到手写数字的各种特征模式，如数字的笔画形状、书写风格等，从而实现准确的分类。

3. -algo FedAvg 参数：

   - 该参数表示本次实验将使用 FedCP 算法（此处原描述有误，已修正为 FedCP 算法）。在这个过程中，每个客户端在本地使用自己的数据训练模型，充分利用客户端本地的数据资源，保护数据隐私的同时实现分布式训练。然后客户端将模型参数上传到服务器，服务器对这些参数进行平均聚合，这种聚合方式综合了各个客户端模型的信息，得到一个新的全局模型参数，并将其发送回客户端，开始下一轮的训练。通过多次迭代，使得全局模型能够不断优化，逐渐适应不同客户端的数据分布，提高模型在整个联邦学习系统中的性能。

4. -gr 200 参数：

   - 此参数设定了实验的训练迭代次数为 200 次。训练迭代次数是影响模型收敛和性能的一个重要因素。在足够的迭代次数下，模型有更多的机会学习到数据中的复杂模式和特征，例如 MNIST 数据集中手写数字的各种变体和细微差别。然而，过多的迭代次数也可能导致过拟合，即模型过度适应训练数据，而对未见过的数据表现不佳。因此，通过合理设置训练迭代次数，如本次设置为 200 次，可以在模型的训练效果和计算资源消耗之间找到一个平衡，确保模型在学习到数据关键特征的同时，保持较好的泛化能力。

4.4 创新点

本次除了原始论文中的数据集，还成功实现了对真实世界数据集 HCI HAR（Human - Computer Interaction Human Activity Recognition，人机交互人类活动识别数据集）的支持，HCI HAR 数据集包含了来自真实世界场景中人类各种活动的多模态数据，如行走、跑步、上下楼梯、站立、坐下等多种活动状态下的传感器数据（如加速度计、陀螺仪等）。这些数据具有丰富的多样性和复杂性，与传统的实验数据集（如 MNIST 等）相比，更能真实地反映现实世界中联邦学习应用场景下的数据特性。在许多实际的联邦学习应用中，如智能家居、健康监测等领域，数据往往以类似 HCI HAR 数据集的形式存在，即来自多个分布式源的传感器数据，用于监测和分析人类的行为活动。通过支持 HCI HAR 数据集，我们的实验环境能够更直接地模拟这些实际应用场景，使得研究成果更具现实意义和应用价值。研究人员可以利用我们的框架在更真实的环境下评估联邦学习算法的性能，例如研究在不同活动模式下模型的准确性、实时性以及对隐私保护的要求。这有助于推动联邦学习技术在实际场景中的落地应用，为解决实际问题提供更有效的解决方案。HCI HAR 数据集涉及人机交互和人类活动识别领域，与联邦学习的结合为跨领域研究提供了新的机遇。这一创新点吸引了来自不同领域的研究人员，如计算机视觉、传感器技术、机器学习以及人机交互设计等领域的专家。通过共同研究基于 HCI HAR 数据集的联邦学习问题，促进了不同领域之间的知识交流和技术融合，有望催生新的研究思路和方法。例如，人机交互领域的专家可以提供关于如何更好地采集和标注活动数据的建议，而机器学习专家则可以专注于开发更高效的联邦学习算法来处理这些数据。这种跨领域的合作将加速联邦学习技术在多领域的创新应用，推动整个领域的发展。

5 实验结果分析

本次对论文中的图像分类任务进行了实验，同时与本领域的最新方法 FedFomo，Per-FedAvg 进行对比实验，还与联邦学习的经典算法 FedAvg 进行对比，表现个性化联邦学习的效果。

5.1 实验设置

本次实现了一个具有 20 台设备，1 个中央服务器的联邦学习系统，每一轮选择 10 个设备进行训练，采用了 4 层 CNN 网络作为模型，在 Cifar10，Cifar100 两个经典图像识别数据集上进行实验，为了模拟不同设备的数据异构情况，数据划分采用了狄利克雷系数 =0.1 来模拟数据异构情况，证明 FedCP 在狄利克雷系数 alpha = 0.1 的情况下优于其他个性化联邦学习算法。

5.2 在图像数据集上进行实验

Cifar10 和 Cifar100 均用于图像分类任务。模型方面采用 4 层 CNN ，通过 Dirichlet 分布创建实际设置场景，设置 =0.1。之后将每个客户端的数据按 75% 和 25% 的比例分别划分为训练数据集和测试数据集。设置本地批量大小为 10，本地学习轮数为 1 。

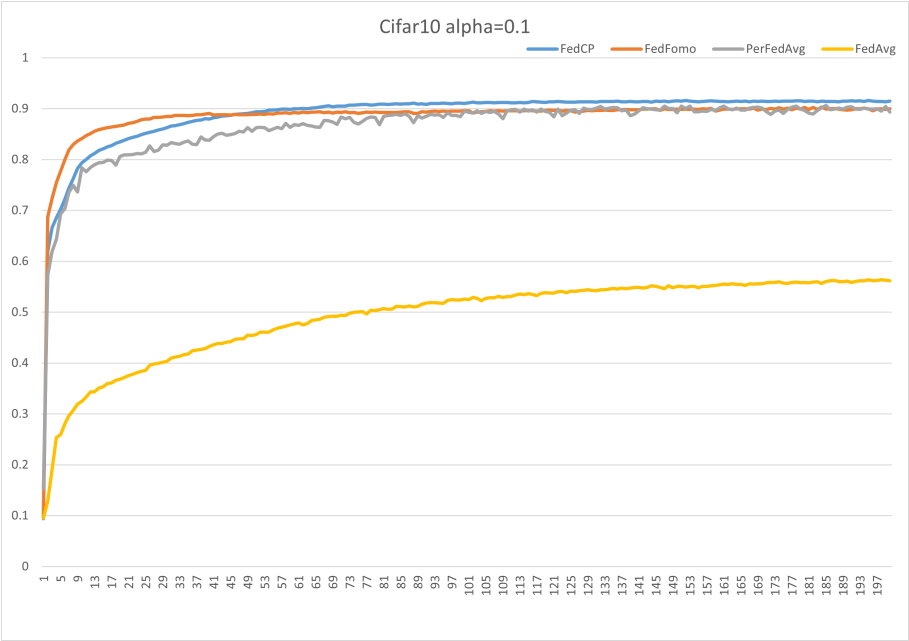

图 3. Cifar10 实验结果示意

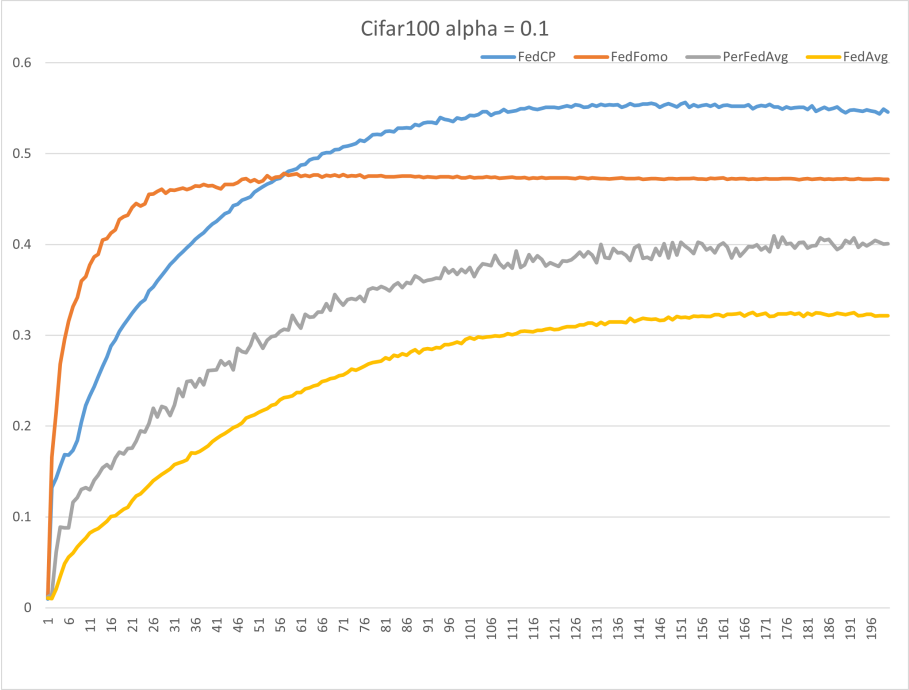

图 4. Cifar100 实验结果示意

在主实验环节，FedCP 在表现卓越，尤其在相对具有挑战性的任务上优势更为突出。例如在 Cifar100 的默认实际设置中，FedCP 的准确率显著超出基线方法。

为了进一步拓展对不同数据集的支持，以及验证 FedCP 是否能在保持个性化联邦学习优势的情况下兼顾均衡分布的情况，本次又在 HCI-HAR 数据集上进行了实验，采用了 6 层 CNN，设置 20 个客户端数据均衡划分，可以看到 FEDCP 在均衡分布的情况下仍然保持最优的性能。

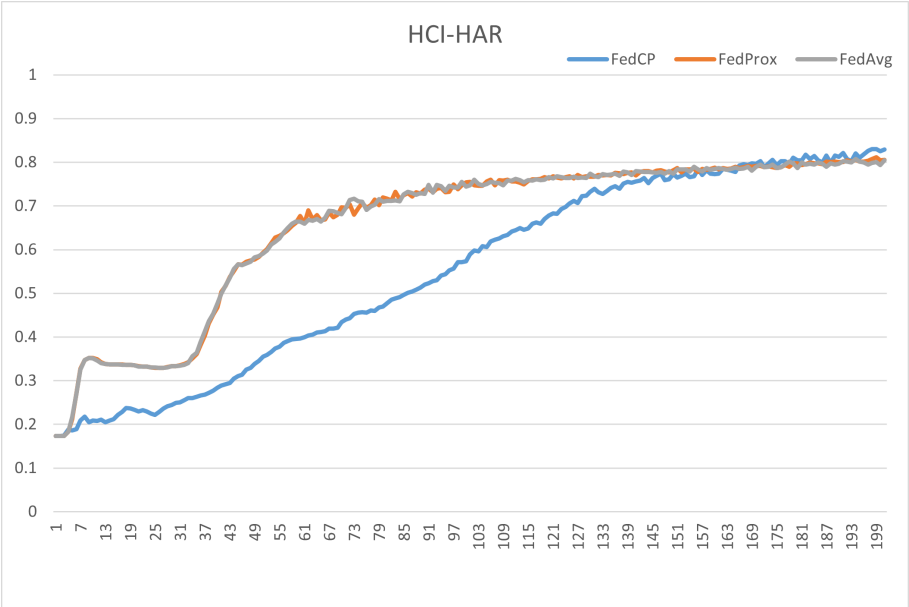

图 5. HCI-HAR 实验结果示意

6 总结与展望

本文聚焦个性化联邦学习领域，提出了 FedCP 方法，致力于解决现有 pFL 方法在处理数据中全局和个性化信息方面的不足，取得了显著成果。在方法创新上，现有 pFL 方法多关注客户端级模型参数中的全局和个性化信息，忽略了数据这一信息源头。FedCP 则另辟蹊径，基于条件计算技术，通过为每个样本生成条件策略，分离特征中的全局和个性化信息，并分别由全局头和个性化头进行处理，实现了样本级的个性化考量，粒度更精细。尽管 FedCP 取得了突出成绩，但仍有进一步探索的空间。在模型优化上，可继续研究 CPN 网络设计，尝试更多架构组合，寻找更优的参数配置，进一步提升模型性能和效率。考虑到现实应用场景的多样性和复杂性，未来研究可拓展到更多实际领域，如医疗健康、金融风控等，在不同的数据分布和任务类型下验证和改进 FedCP，提升其泛化能力和适应性。

本次复现工作存在一些不足，首先是受实验设备的性能限制好可分配时间有限，，未能完成原文中的 2000 轮完整训练，也没有实现原文所有的数据集上进行的实验，在与其他先进方法的对比实验中，我们对对比方法的配置和参数设置可能不够精准。不同方法在特定数据集和任务上可能需要针对性的参数调整才能发挥出最佳性能，若不能保证这些对比方法处于最优状态，可能会影响对 FedCP 优势的准确评估，无法清晰界定其真正的领先程度。

参考文献

[1] Fedcp code repository.

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[3] Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin. Federated user representation learning. *arXiv preprint arXiv:1909.12535*, 2019.

[4] Yiqiang Chen, Wang Lu, Xin Qin, Jindong Wang, and Xing Xie. Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[5] Anda Cheng, Zhen Wang, Yaliang Li, and Jian Cheng. Hpn: Personalized federated hyperparameter optimization. *arXiv preprint arXiv:2304.05195*, 2023.

[6] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4805–4814, 2019.

[7] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[8] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):567–580, 2022.

[9] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[10] Xin-Chun Li, De-Chuan Zhan, Yunfeng Shao, Bingshuai Li, and Shaoming Song. Fedphp: Federated personalization with inherited private models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 587–602. Springer, 2021.

[11] Yilin Liu, Jiale Chen, Shanshan Pan, Daniel Cohen-Or, Hao Zhang, and Hui Huang. Split-and-Fit: Learning B-Reps via Structure-aware Voronoi Partitioning. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 43(4):108:1–108:13, 2024.

[12] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10092–10101, 2022.

[13] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[14] Xuanming Ni, Xinyuan Shen, and Huimin Zhao. Federated optimization via knowledge codistillation. *Expert Systems with Applications*, 191:116310, 2022.

[15] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

[16] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

[17] Tianchun Wang, Wei Cheng, Dongsheng Luo, Wenchao Yu, Jingchao Ni, Liang Tong, Haifeng Chen, and Xiang Zhang. Personalized federated learning via heterogeneous modular networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1197–1202. IEEE, 2022.

[18] Sixing Yu, Phuong Nguyen, Waqwoya Abebe, Justin Stanley, Pablo Munoz, and Ali Jannesari. Resource-aware heterogeneous federated learning using neural architecture search. *arXiv preprint arXiv:2211.05716*, 2022.

[19] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.