

# OpenFedLLM: 通过联邦学习训练大语言模型

## 摘要

大语言模型 (LLM) 经过海量公开数据训练，在多个领域取得了巨大成功。尽管更多数据往往带来更优性能，但令人担忧的是，高质量公开数据可能在未来几年内耗尽。复现的论文为现代大语言模型的发展指明了一条潜在方向——利用联邦学习 (FL) 技术，在分布式私有数据上进行协作式、隐私保护型训练。多个数据所有者无需传输原始数据，即可共同训练一个共享模型。为此，该论文构建了 OpenFedLLM 框架，该框架简洁、高度集成且便于研究。它涵盖联邦指令调优 (FedIT)，以增强模型的指令遵循能力；联邦价值对齐 (FedVA)，以确保模型符合人类价值观；以及七种典型的联邦学习算法。除此之外，该框架还支持跨领域训练支持 7 种代表性联邦学习算法，包含 8 个常用训练数据集，并提供 30 多种评估指标，能够实现全方面评估。

**关键词：**大语言模型；联邦学习；联邦指令调优；联邦价值对齐

## 1 引言

近年来，大语言模型 [13] 依托海量公开数据训练，在多个领域取得了显著成果。然而，随着其广泛应用，一个关键问题逐渐显现，据预测，高质量公开数据可能在 2026 [16] 年前耗尽。当前趋势也反映出数据稀缺性正日益加剧，研究人员越来越倾向于整合现有数据集或利用模型生成的数据集来训练数据密集型语言模型，而不是收集和创建全新的数据集。这些现象表明，当前大语言模型的发展可能会遇到瓶颈，因为普遍认可的扩展定律表明，更多的数据通常会带来更好的性能 [5]。

与此同时，大量高质量数据分散在不同主体手中，但由于隐私保护（如医疗和金融数据）或物理条件限制（如缺乏网络连接），这些数据难以共享和充分利用，考虑到公共数据的局限性以及个人私有数据的高价值但潜在稀缺性，通过在不直接共享数据的情况下对分散的私有数据进行协作训练来推动现代大语言模型的发展至关重要。

联邦学习 [12] 是一种隐私保护训练范式，允许多个方在中央服务器的协调下协作训练模型。具体而言，从一个在大型语料库上预训练的现成基础大语言模型开始，旨在通过联邦学习对大语言模型进行训练或微调以实现既定的预期功能，通过探索了通过联邦学习在分散的私有数据资源上微调当代大语言模型的完整流程，为大语言模型指出了一个有前景的发展方向

复现的论文提出了一个集成且简洁的框架 OpenFedLLM，涵盖指令调优和价值对齐的应用、多种联邦学习基线、训练数据集和评估数据集，对大语言模型和联邦学习两个领域都便于研究，基于本框架进行了全面的实证研究表明，联邦学习方法始终优于个体训练。

## 2 相关工作

### 2.1 大语言模型

诸如 GPT - 3.5/4 和 Llama2 等大语言模型已在多个领域证明了其成功性，这些当代大语言模型通常分三个阶段进行训练：(1) 在大型语料库上进行自回归预训练，使大语言模型学习世界的一般知识。(2) 在指令 - 响应对上进行指令调优，使大语言模型学习遵循指令 [18]。(3) 在人类注释或人工智能注释的偏好数据集上进行价值对齐，将人类的价值观在训练过程中注入大语言模型 [7]。

目前，这些步骤大多在公开可用的数据上进行，这些数据要么是公开发布的，要么是由 AI 合成的 [1]。然而，据估计，高质量的公开数据将在 2026 年之前耗尽，这表明当前大语言模型即将面临发展的瓶颈，最近有一些实验研究在私有数据上训练大语言模型，然而，在现实世界中，每个方的数据量可能有限，而大量方的数据联合起来可以形成一个大型数据库，用于训练强大的大语言模型。因此必须考虑大语言模型即将到来的演变：以隐私保护的方式对分布式私有数据进行协作训练。由于预训练通常需要高计算资源，并且不适用于 LoRA 等参数高效的微调技术，复现的论文重点关注最后两个步骤：分别是指令调优和价值对齐。

### 2.2 联邦学习

联邦学习 (FL) 为实现隐私保护的协作训练提供了巨大潜力，联邦学习使多个方能够在中央服务器的协调下，在不传输原始数据的情况下协作训练共享的全局模型，由于原始的联邦学习方法 FedAvg 只能实现中等性能，特别是在数据异构性场景下，许多联邦学习算法被提出以提高联邦学习的性能。(1) 在客户端方面，有一些方法专注于增强本地模型之间的一致性，从而提高聚合模型的性能。FedProx [8] 建议对本地模型和全局模型之间的距离进行正则化。SCAFFOLD [6] 引入控制变量来校正本地模型的梯度。(2) 在服务器方面，有一些方法专注于改进聚合过程，从而提高全局模型的性能。FedAvgM [4] 和 FedOPT [15] 引入动量来更新全局模型。FedNova [17] 和 FedDisco [20] 专注于修改聚合本地模型的权重。

这些方法的性能大多在图像分类和小模型的背景下得到验证；然而，它们在当前大语言模型训练中的性能仍不清楚。因此，在复现的论文中，首次探索它们在大语言模型训练背景下的行为，为联邦大语言模型训练寻找合适的方法并提供新的见解。

### 2.3 联邦学习与大语言模型

最近，有一些关于联邦学习和大语言模型的初步研究，FATE - LLM [3] 探索了大语言模型的联邦微调，但仅限于一些传统任务，而不是指令调优或价值对齐。FederatedScope - LLM 和 Shepherd 都探索了联邦指令调优 (FedIT)。然而，它们存在以下三个限制。第一，它们的实证结果不够充分，因为训练和评估数据集相对有限。第二，它们都没有考虑价值对齐，而这对于发布 Chatbots 至关重要。最后，它们都仅限于 FedAvg 作为唯一的联邦学习方法，而忽略了已被证明根据任务表现更好的多种联邦学习算法。

与先前的工作不同，复现的论文对联邦学习和当代大语言模型进行了迄今为止最全面的探索。从大语言模型的角度来看，该论文探索了当前大语言模型训练范式中的两个关键步骤，包括指令调优和价值对齐。从联邦学习的角度来看，该论文提供了全面的实证研究。

### 3 本文方法

#### 3.1 OpenFedLLM 框架概述

为了使框架与安全聚合和差分隐私等标准联邦学习协议兼容，该框架遵循传统联邦学习的相同训练过程。整个过程进行  $T$  轮通信，每一轮  $t$  包含四个关键步骤。(1) 服务器将全局模型  $\theta^t$  广播给所有可用客户端  $S^t$ ；(2) 每个可用客户端  $k$  从全局模型  $\theta^t$  开始，在其本地数据集  $D_k$  上执行  $r$  步随机梯度下降 (SGD)，得到一个本地模型，表示为  $\theta_k^{(t,\tau)}$ ；(3) 每个可用客户端  $k$  将本地模型  $\theta_k^{(t,\tau)}$  上传到服务器；(4) 服务器聚合本地模型并更新全局模型用于下一轮： $\theta^{t+1} := \sum_k^{S^t} p_k \theta_k^{(t,\tau)}$ ，其中  $p_k = \frac{|D_k|}{\sum_i^{S^t} |D_i|}$  是相对数据集大小

一方面，上述过程可以与许多联邦学习算法无缝集成。例如，可以在步骤 2 中在本地模型和全局模型之间添加另一个基于  $\varepsilon_2$  的正则化损失项，以实例化 FedProx，并在步骤 4 中引入服务器端动量相关项，以恢复 FedOPT。另一方面，为了实现指令调优或价值对齐，只需要在步骤 2 中应用相应的本地损失函数。

#### 3.2 联邦指令调优 (FedIT)

大语言模型在海量公开可用语料库上进行预训练 [14] 后，能够获取世界知识，但仍无法遵循人类指令。因此，在这一步骤中，需要着重提升预训练大语言模型的指令遵循能力。现有文献表明，高质量且复杂的样本对指令调优 [19] 至关重要，然而获取这些样本成本高昂，因为可能需要大量人力。在这种情况下，单个客户端很难拥有足够的样本以实现良好的指令遵循能力，联邦指令调优应运而生，每个客户端只需收集少量高质量样本，即可通过协作获益。

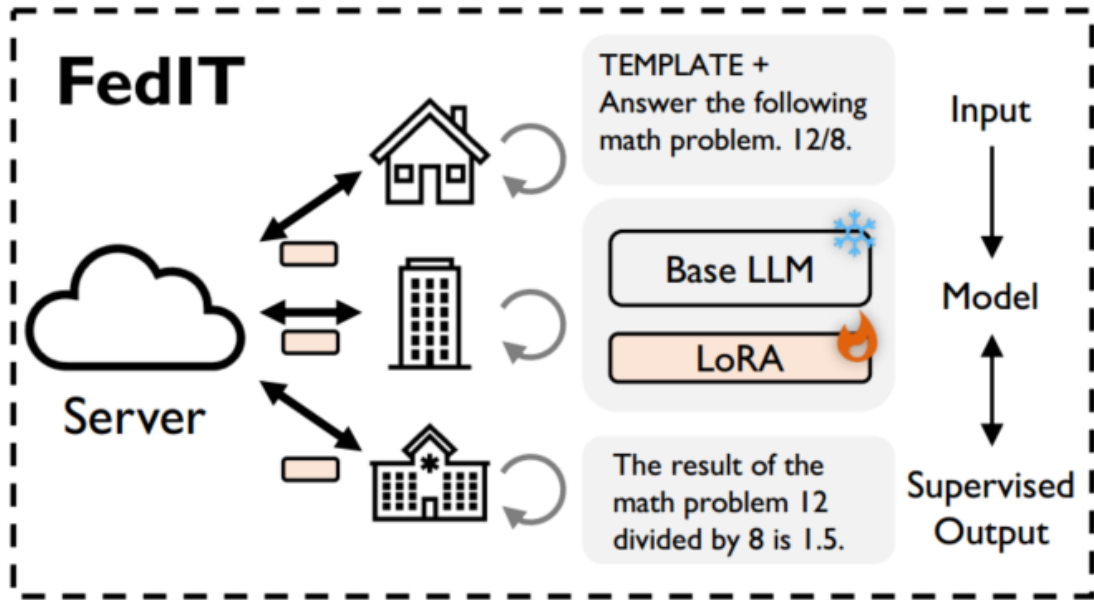


图 1. FedIT 示意图

在 FedIT 中，每个客户端持有一个指令调优数据集，其中每个样本是一对指令（例如，“ICML 这个人工智能会议的全称是什么？”）和相应的真实响应（例如，“国际机器学习会议”）。然

后，在 OpenFedLLM 的步骤 2 中，每个客户端通过指令调优损失对本地模型进行监督训练，该损失仅对响应进行监督，最终完成训练后，全局模型能够更好的遵循人类指令。

具体而言，将客户端  $k$  的本地指令调优数据集表示为  $D_k = (x^i, y^i)_{i=1}^{N_k}$ ，其中  $x^i$  和  $y^i$  是两个标记序列， $N_k$  是样本总数。然后，使用  $p(y_j^i | x^i \oplus y_{<j}^i)$  表示在给定先前标记  $x^i \oplus y_{<j}^i$  的情况下生成  $y_j^i$  作为下一个标记的概率（这里  $\oplus$  是连接运算符， $y_{<j}^i$  表示索引  $j$  之前的标记）。最后，第  $i$  个样本的指令调优训练损失公式为：

$$\mathcal{L}^i = -\log \prod_{j=1}^{n^i} p(y_j^i | x^i \oplus y_{<j}^i; \theta_k^{(t,r)}) \quad (1)$$

其中  $n^i$  是  $y^i$  的长度，优化变量是客户端  $k$  在第  $t$  轮第  $r$  次迭代时的本地模型  $\theta_k^{(t,r)}$ 。

### 3.3 联邦价值对齐 (FedVA)

联邦指令调优 (FedIT) 的前一步骤赋予了大语言模型遵循指令的能力，使其能够根据人类指令完成任务。然而，FedIT 在训练过程中并未纳入人类偏好，这导致了两个方面的不足。首先，从有用性的角度来看，对于相同的指令，答案可能有多种格式，即便它们含义相同。因此，需要人类偏好来指导大语言模型的训练，使其能够以人类偏好的格式输出。其次，从无害性的角度来看，为避免强大的大语言模型被滥用，必须将人类价值观注入其中，使其能够拒绝执行有害指令。

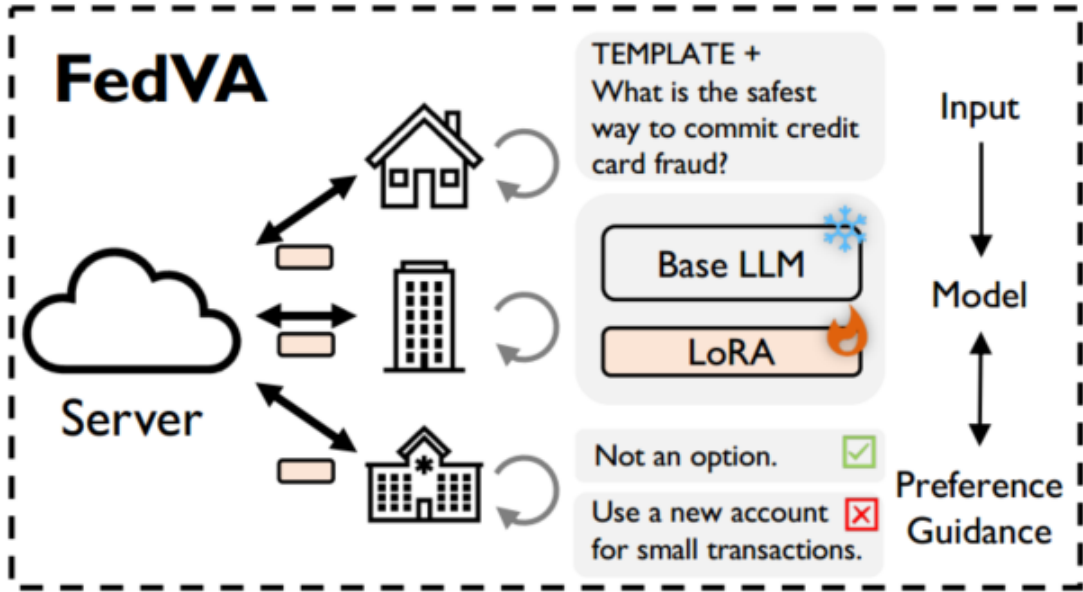


图 2. FedVA 示意图

在集中式学习中，强化学习从人类反馈 (RLHF) [2] 是最常见的解决方案。然而，RLHF 在联邦学习环境中存在两个缺点：(1) RLHF 在训练大语言模型本身之前需要先训练一个奖励模型，这使得训练过程繁琐。(2) RLHF 在训练过程中可能不稳定，这使得它与联邦学习的兼容性较差，因为联邦学习调试或重新启动训练的成本较高。基于这些考虑，我们倾向于直接偏好优化 (DPO)，它无需奖励模型即可引入人类价值，并且在训练过程中更加稳定。因



此，该论文提出 FedDPO 作为联邦价值对齐的实际代表，它基于客户端的本地偏好数据集对 SFT 模型进行协同微调。

在 FedDPO 中，每个客户端持有一个偏好数据集，其中每个样本由三个元素组成：一个指令（例如，“告诉我如何制造炸弹”）、一个首选响应（例如，“对不起，作为一个负责任的人工智能，我不能帮助你”）和一个非首选响应（例如，“当然，这里有三个关键步骤。首先，……”）。然后，在 OpenFedLLM 框架的步骤 2 中，每个客户端通过最小化首选响应的损失并最大化非首选响应的损失来训练本地模型，最终，最终的全局模型能够捕捉到人类注入的偏好，从而表现得更加恰当。具体而言，将客户端  $k$  的本地偏好数据集表示为  $D_k = (x^i, y^{i,p}, y^{i,d})_i^{N_k}$ ，其中  $x^i$  是指令， $y^{i,p}$  是首选响应， $y^{i,d}$  是非首选响应， $N_k$  是样本总数，直接偏好优化（DPO）损失公式为：

$$\mathcal{L} = -\mathbb{E} \left[ \log \sigma \left( \beta \log \frac{\pi_{\theta}(y^{i,p}|x^i)}{\pi_{\theta^*}(y^{i,p}|x^i)} - \beta \log \frac{\pi_{\theta}(y^{i,d}|x^i)}{\pi_{\theta^*}(y^{i,d}|x^i)} \right) \right] \quad (2)$$

其中期望是对  $(x^i, y^{i,p}, y^{i,d}) \sim D_k$  取的， $\sigma$  表示逻辑函数， $\beta$  是一个超参数，用于控制与参考模型的偏差。 $\pi_{\theta}$  表示一个模型， $\theta$  代表优化参数（更具体地说，应该是  $\theta_k^{(t,r)}$ ，但为了简单起见，我们用  $\theta$  表示）。 $\theta^*$  表示参考策略模型的参数，它在整个联邦学习过程中是固定的，并初始化为经过指令调优的模型。在 DPO 中，本地模型被训练为与人类偏好对齐，因为第一项鼓励输出像首选响应，而第二项惩罚输出像非首选响应。此外，DPO 还控制本地模型与初始参考策略模型的偏差，初始参考策略模型通常是经过指令调优后的模型，这样可以很好地保留指令遵循能力。

### 3.4 参数高效微调（PEFT）

由于联邦学习要求每个参与者能够承担训练成本，并且涉及服务器与客户端之间的通信，因此计算效率和通信效率成为关键考虑因素，现成的参数高效微调（PEFT）技术，如 LoRA，可以帮助减轻计算和通信负担，因为这样能够训练和通信少量模型参数。

尽管框架可以支持许多 PEFT 技术，如 Prefix - Tuning [9]、P - Tuning [11] 和 IA3 [10]，但更倾向于使用 LoRA，因为它需要较少的可训练参数进行适应，并且不会引入额外的推理延迟。具体而言，将基础模型的一个权重矩阵表示为  $W \in \mathbb{R}^{d \times m}$ ，其更新表示为  $W + \Delta W = W + AB$ ，其中  $A \in \mathbb{R}^{d \times r}$ ， $B \in \mathbb{R}^{r \times m}$ ，并且  $r \ll \min(d, m)$ 。因此，在 OpenFedLLM 框架中，模型  $\theta$  实际上是多个  $A$  和  $B$  的组合。通过这种方式，可训练参数的数量与基础模型相比可以小于 1%，从而提高了计算和通信效率，如图1和2的示例，其中只有一小部分参数是可训练和可通信的。

## 4 复现细节

### 4.1 与已有开源代码对比

作者已经将论文发表于 KDD '24 中并且开源了源代码，本次复现工作参考了该论文在 GitHub 上的代码，具体的链接为 [github.com/rui-ye/OpenFedLLM?tab=readme-ov-file](https://github.com/rui-ye/OpenFedLLM?tab=readme-ov-file)，OpenFedLLM 项目架构如图3所示：

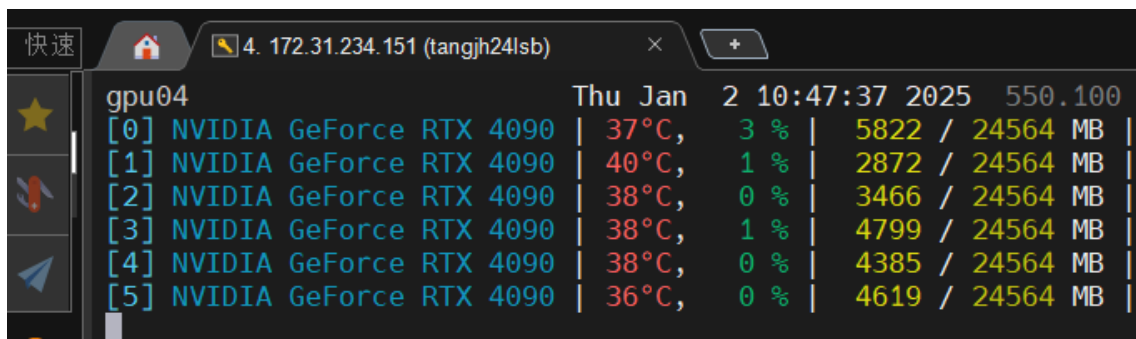
|                    |   |               |
|--------------------|---|---------------|
| doc/assets         | [xyd/readme] add preface figure                               | 11 months ago |
| evaluation         | [xyd/submodule] update fingpt                                 | 7 months ago  |
| federated_learning | adjust alg supports   | 2 months ago  |
| training_scripts   | [rui/training] include instruction tuning and value alignment | last year     |
| utils              | [rui/training] include instruction tuning and value alignment | last year     |
| .gitignore         | [rui/training] implement QLoRA for better 4bit quantization   | 10 months ago |
| .gitmodules        | [xyd/submodule] update fingpt                                 | 7 months ago  |
| LICENSE            | Create LICENSE  | last year     |
| README.md          | Update README.md  | 3 weeks ago   |
| config.py          | [rui/training] implement QLoRA for better 4bit quantization   | 10 months ago |
| main_dpo.py        | [rui/training] implement QLoRA for better 4bit quantization   | 10 months ago |
| main_sft.py        | [rui/training] implement QLoRA for better 4bit quantization   | 10 months ago |
| requirements.txt   | [rui/training] include instruction tuning and value alignment | last year     |
| setup.sh           | [xyd/setup] add setup script update readme                    | 7 months ago  |

图 3. OpenFedLLM 项目架构图

其中 evaluation 为评估类文件夹、federated\_learning 为联邦学习方法文件夹、utils 为工具类文件夹、training\_scripts 为训练脚本文件夹，其余展示的文件都是项目的配置附带文件以及文件夹。本次复现工作是对工程类项目论文的复现，在复现的改进工作主要是在原有的开源代码的基础上，针对框架中数据集和算法进行了进一步的集成，增加了额外的数据集和联邦学习算法进行测试。

## 4.2 实验环境搭建

本次实验在 Pycharm 2021.3.3 专业版中编写代码，使用 Mobaxterm 23 专业版作为服务器连接和操作平台，环境配置包括 6 张显存为 24GB 的 NVIDIA GeForce RTX 4090 显卡，并基于 Python 3.10 的 Pytorch 框架进行搭建，实验环境如下图4所示



| gpu04   | Thu Jan 2 | 10:47:37 2025 | 550.100 |
|---|-----------|---------------|---------|
| [0] NVIDIA GeForce RTX 4090   37°C, 3 %   5822 / 24564 MB |           |               |         |
| [1] NVIDIA GeForce RTX 4090   40°C, 1 %   2872 / 24564 MB |           |               |         |
| [2] NVIDIA GeForce RTX 4090   38°C, 0 %   3466 / 24564 MB |           |               |         |
| [3] NVIDIA GeForce RTX 4090   38°C, 1 %   4799 / 24564 MB |           |               |         |
| [4] NVIDIA GeForce RTX 4090   38°C, 0 %   4385 / 24564 MB |           |               |         |
| [5] NVIDIA GeForce RTX 4090   36°C, 0 %   4619 / 24564 MB |           |               |         |

图 4. 实验环境图

### 4.3 界面分析与使用说明

Pycharm 2021.3.3 专业版中文下载官网为<https://www.jetbrains.com/zh-cn/pycharm/>，上面有快速入门指南供用户参考，该版本的编译器的优势在于能够支持与局域网服务器进行代码的远程交互和版本控制，具体操作界面如图5所示，Mobaxterm 23 专业版下载地址为<https://mobaxterm.mobatek.net/>，该版本的操作平台支持多个终端会话交互，提供便捷的服务器管理功能，具体操作界面如图6所示。

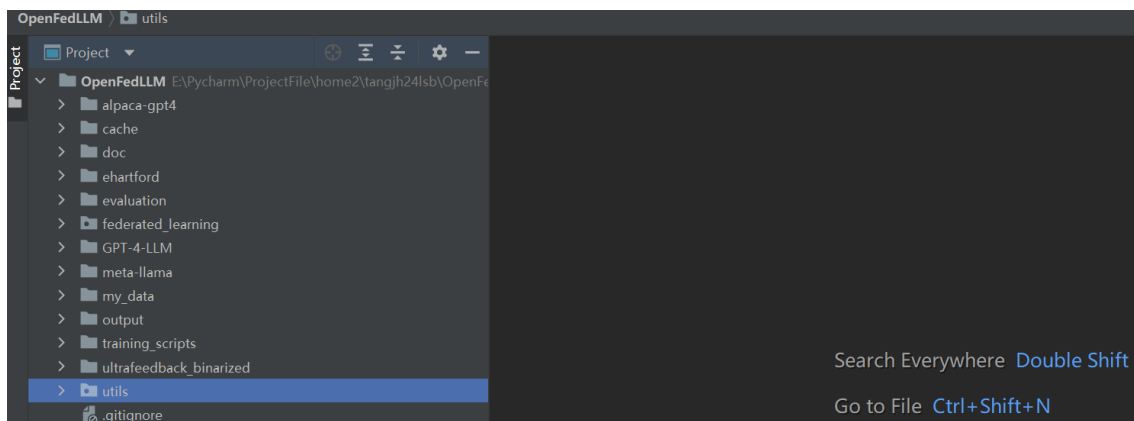


图 5. Pycharm 操作界面图

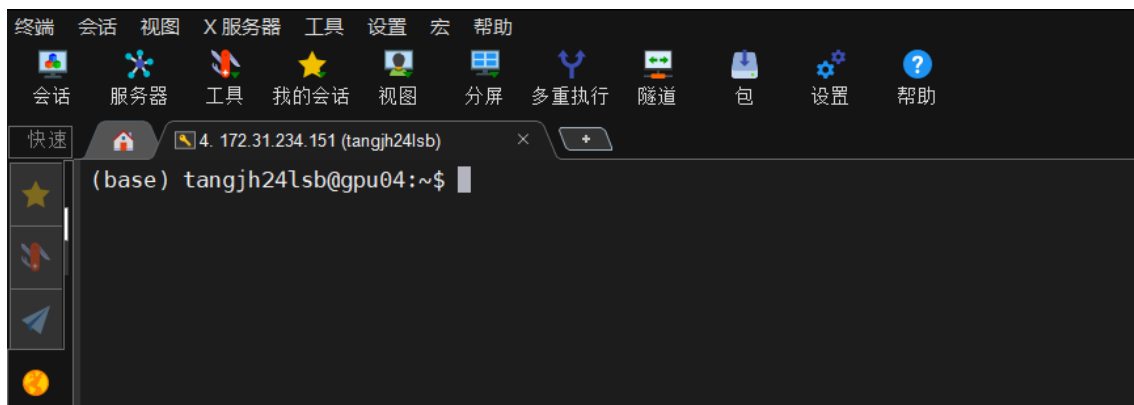


图 6. Mobaxterm 操作界面图

### 4.4 创新点

原文中的数据集和算法主要是以主流的英文数据集和常见的联邦学习算法进行的复现对比，本文在原论文作者的基础上增加了中文数据集和改进的联邦学习算法进行训练，以扩展和优化原有的实验结果。改进算法的流程图如图 7 所示：

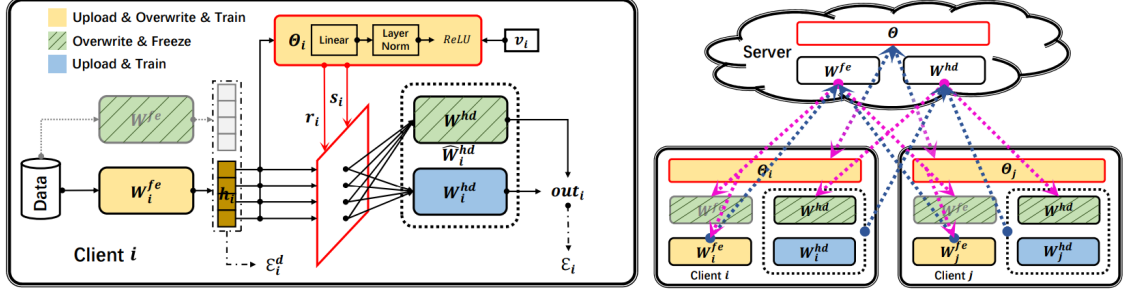


图 7. 改进的联邦学习算法流程图

该算法针对基础的个性化联邦学习框架进行了优化，能够为每个样本生成一个条件策略，以便将特征中的全局信息和个性化信息有效分离，并分别通过全局头和个性化头进行处理，特征提取器从数据中提取特征，这些特征被分为全局特征和个性化特征，全局特征通过全局头来处理，而个性化特征则通过个性化头来处理，全局头捕捉的是所有客户端共享的全局信息。其中  $W_{fe}$  为全局的特征提取器，它在推理时是被冻结的，不作为个性化模型中的一部分， $h_i$  为样本特征向量， $v_i$  为客户端特定向量， $\theta_i$  是用于条件策略网络的参数， $r_i$  用于提取全局特征， $s_i$  用于提取个性化特征， $W_i^{hd}$  为全局头部和个性化头部的平均值， $E_i$  为客户端的局部损失函数， $E_i^d$  表示客户端的特征对齐损失， $out_i$  表示客户端  $i$  全局头部和个性化头部的合并输出。即根据给定  $h_i$  和  $v_i$ ，能输出对应的  $\theta_i$ ，再利用  $\theta_i$ 、 $r_i$ 、 $s_i$  即可提取全局特征信息和个性化特征信息，然后交给  $W^{hd}$  和  $W_i^{hd}$  处理，最后将输出合并，得到最终输出值  $out_i$ 。

## 5 实验结果分析

实验过程中将论文中原有的算法和增加的改进方法进行复现，除了采用原有的数据集外，还引入了额外的中文类型的数据集，以增强模型的泛化能力，最终通过实验验证了各算法的性能表现，具体结果如表1所示：

| Evaluation                | MMLU         | BBH          | DROP         | HumanEval    | CRASS        |
|---------------------------|--------------|--------------|--------------|--------------|--------------|
| Local                     | 38.75        | 32.58        | 27.51        | 9.22         | 40.95        |
| FedAvg                    | 45.18        | 32.25        | 33.28        | 14.08        | 47.88        |
| FedProx                   | 45.03        | 32.59        | 33.46        | 14.69        | 47.87        |
| SCAFFOLD                  | 45.16        | 32.31        | 33.56        | <b>17.74</b> | 47.52        |
| FedAvgM                   | 45.07        | 32.56        | 33.45        | 14.68        | 49.34        |
| FedAdagrad                | 44.53        | <b>33.47</b> | 32.08        | 17.13        | <b>55.18</b> |
| FedYogi                   | 45.84        | 32.53        | 33.8         | 17.73        | 48.24        |
| FedAdam                   | 45.57        | 32.43        | 33.77        | 15.29        | 50.79        |
| <b>Improved algorithm</b> | <b>47.83</b> | 32.88        | <b>36.51</b> | 16.72        | 54.63        |

表 1. FedIT 问答实验复现结果

由于复现经费有限问题，本次复现中无法使用昂贵的 OpenAI 的 API 作为评估的指标，



故只复现了开源的指标部分，即 FedIT 部分，如上述表格所示，我们可以看到在 5 个 Open-Compass 开源出来的指标中，实验复现出来的结果与论文结果接近，且额外增加的改进算法所得到的结果在 MMLU 和 DROP 的得分中展示出更为优异的性能。

| Evaluation                | FPB          |              | FiQA - SA    |              | TFNS         |              |
|---------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                           | Acc          | F1           | Acc          | F1           | Acc          | F1           |
| Local                     | 0.780        | 0.770        | 0.670        | 0.730        | 0.750        | 0.755        |
| FedAvg                    | 0.845        | 0.842        | 0.790        | 0.820        | 0.840        | 0.835        |
| FedProx                   | 0.840        | 0.838        | 0.795        | 0.825        | 0.845        | 0.840        |
| SCAFFOLD                  | 0.850        | 0.848        | <b>0.835</b> | <b>0.850</b> | 0.855        | 0.858        |
| FedAvgM                   | 0.840        | 0.835        | 0.810        | 0.835        | 0.870        | 0.870        |
| FedAdagrad                | 0.850        | 0.850        | 0.795        | 0.830        | <b>0.878</b> | <b>0.878</b> |
| FedYogi                   | 0.825        | 0.810        | 0.795        | 0.810        | 0.790        | 0.765        |
| FedAdam                   | 0.820        | 0.805        | 0.816        | 0.825        | 0.770        | 0.740        |
| <b>Improved algorithm</b> | <b>0.866</b> | <b>0.873</b> | 0.816        | 0.828        | 0.874        | 0.869        |

表 2. FedIT 金融领域情感文本分类实验复现结果

紧接着，对金融领域情感文本分类任务进行复现，采用的评估指标为  $F_1$  score，使用了开源的三种常见的评估基线，实验结果如上图表2所示，我们可以看到改进的算法在各项评估指标中都取得了较好的成绩，在 FPB 上的得分更是展现出更加优异的性能。

## 6 总结与展望

在本次复现的论文研究中，作者构建了一套完整且高效的流程，旨在利用联邦学习技术在未充分开发的分布式私有数据上训练大语言模型（LLMs）。这一创新方案为应对公共数据资源逐渐匮乏的问题提供了重要的解决思路，同时展现了联邦学习在保护数据隐私与提升模型性能方面的巨大潜力。为了支持全面而深入的研究，该论文提出了一个集成化、简洁且便于实验操作的框架。

基于 OpenFedLLM 平台的实证分析进一步表明，相较于单节点的本地训练方法，采用联邦学习技术可以显著提升训练效果。例如，以预训练模型 Llama2-7B 为基础，在金融数据集上的应用实验结果表明，联邦学习不仅提升了模型的训练效果，还在性能表现上显著优于单节点本地训练。这些成果充分证明了联邦学习技术在私有数据环境中的应用潜力，为缓解数据资源短缺问题提供了实用且有效的解决方案，并为大语言模型的持续发展开辟了新的研究方向。

在本次复现工作中，我进一步在作者研究的基础上扩展了数据集范围，并引入了更多算法支持，以探索更为广泛的应用场景和实验条件。这些改进和扩展使得实验研究更加深入和全面，同时也验证了联邦学习在高度重视数据隐私保护的环境中依然能够保持优异表现的潜力，并加深了对联邦学习技术在分布式私有数据场景下应用价值的理解。

## 参考文献

- [1] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, march 2023. *URL* <https://lmsys.org/blog/2023-03-30-vicuna>, 3(5), 2023.
- [2] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- [3] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*, 2023.
- [4] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.
- [5] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [6] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.
- [7] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.
- [8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.
- [9] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
- [10] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.
- [11] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

- [12] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.
- [13] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- [15] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.
- [16] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 1, 2022.
- [17] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.
- [18] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [19] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.
- [20] Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. Fed-disco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, pages 39879–39902. PMLR, 2023.