通过重叠本地步骤加速去中心化优化

Yijie Zhou数据科学学院 香港中文大学（深圳） yijiezhou@link.cuhk.edu.cn

Shi Pu数据科学学院 香港中文大学（深圳） shipu@cuhk.edu.cn

arXiv:2601.01493v1 [cs.LG] 4 Jan 2026

摘要

去中心化优化已成为分布式学习的重要范式，借助点对点协作，在保护数据隐私的同时实现可扩展训练。然而，现有方法往往因节点间频繁同步而遭遇通信瓶颈。我们提出重叠式局部去中心化随机梯度下降（OLDSGD），这是一种通过计算与通信重叠来加速去中心化训练的新方法，可显著减少网络空闲时间。通过精心设计更新方式，OLDSGD 在避免因通信导致的停滞的同时，保留了与本地随机梯度下降相同的平均更新。从理论上看，我们为光滑非凸目标函数建立了非渐近收敛速率，证明 OLDSGD 在改善每次迭代运行时间的同时，保持了与标准本地去中心化随机梯度下降相同的迭代复杂度。实证结果表明，在不同通信延迟下，OLDSGD 均能持续改善实际运行时间收敛。只需对现有框架进行最小幅度的修改，OLDSGD 就能在不牺牲理论保证的前提下，为加速去中心化学习提供实用方案。

1 引言

在大规模机器学习时代，训练数据集规模不断增长，且频繁分布在多个地点，因此分布式训练已成为必需。传统的集中式方法——将数据汇集到单一地点——往往面临严重瓶颈，包括内存限制、通信开销和隐私问题 [22]。联邦学习 [12] 和数据并行训练 [9] 等分布式训练方法，通过让节点协作进行模型更新，同时将数据保留在本地，缓解了这些问题。然而，这些方法通常依赖中心协调器（例如参数服务器）来同步更新，这可能导致单点故障或通信瓶颈 [6]。去中心化优化（DO）通过在没有中心协调器的情况下实现点对点协作，并利用本地计算和邻居间通信 [14]，解决了这些局限。DO 的目标是优化局部目标之和：

$$\min_{x\in\mathbb{R}^d} f(x) = \sum_{i=1}^{n} f_i(x), \tag{1}$$

预印本。评审中。

其中， n 是智能体的数量， d 是问题的维度， $f_i(x) \triangleq \mathbb{E}_{\xi \sim \mathcal{D}_i} F(x; \xi)$ 表示智能体 i 在数据分布为 $\mathcal{D}_i$ 时的局部损失。智能体通过仅与集合 $\mathcal{N}_i$ 中的邻居通信，协作求解（1）。

分布式训练中的一项关键挑战是通信开销，因为同步会导致等待，从而产生空闲时间 [3]。局部更新 [20]和通信-计算重叠[23] 等技术通过降低同步频率，并利用陈旧信息（例如梯度）进行更新来掩盖通信延迟。然而，陈旧性可能会降低收敛速度，尤其是在非凸目标函数的情况下，因为过时的梯度可能与当前优化方向不一致 [2]。
在去中心化训练中，通信-计算重叠仍未得到充分探索。尽管[25] 提到，采用先融合后适应（CTA）方案的去中心化随机梯度下降（DSGD）能够支持通信与计算重叠，但这一思路尚未得到深入研究；当通信速度较慢时，简单地进行重叠只能带来有限的加速比。在此基础上，发挥 DSGD-CTA 实现重叠的潜力，我们提出了重叠式局部去中心化随机梯度下降（OLDSGD），这是首个将局部更新与通信-计算重叠相结合的去中心化训练算法。通过谨慎修改更新规则，OLDSGD 在不改变平均更新的情况下，完全掩盖通信延迟，从而缓解分布式训练中普遍存在的陈旧性问题 [27]。这使去中心化训练更快速、更稳健。

1.1 相关工作

OLDSGD 是首个将通信-计算重叠与局部更新相结合的去中心化算法，可在维持收敛保证的同时减少通信延迟。为说明其创新之处，我们首先回顾采用类似技术、但依赖集中式协调的分布式训练方法。随后，我们考察不采用重叠的去中心化局部方法，并阐明 OLDSGD 的独特贡献。

1.1.1 分布式重叠局部方法

此前关于重叠局部方法的研究，包括 Overlap-Local-SGD [24], CoCoD-SGD [18],和 Co 2 [21], ，从根本上依赖全局平均模型更新。这种架构选择本身限制了它们在任意网络拓扑中的适用性。在 Overlap-Local-SGD 中，智能体并行执行局部更新，同时定期通信以计算滞后的全局平均值（称为锚点）。这一机制实现了计算与通信的重叠，随后局部模型会向锚点调整，以维持共识。类似地，CoCoD-SGD 允许节点在交换更新的同时使用滞后的梯度进行计算，在全连接拓扑中与 OLDSGD 类似。Co2 则采用不同的方法，通过利用异步通信模式缩短训练时间，但这是在集中式系统架构中实现的。相比之下，OLDSGD 使用邻居的平均模型，因此特别适用于具有任意网络结构的完全去中心化场景。

Co2 和大多数分布式重叠方法 [28, 4, 7] 由于使用陈旧梯度，会在更新中引入偏差。这些方法采用各种补偿技术来减轻这种偏差造成的性能下降：Co2 使用梯度裁剪和陈旧性惩罚，Delayed-SGD [28]采用自适应梯度加权，而 Workie-Talkie [4] 则结合误差补偿与陈旧性感知调度。相比之下，OLDSGD 在去中心化场景中通过先融合后适应（CTA）方案采用了截然不同的方法。这种设计天然能够实现无偏平均模型更新，从而避免了其他方法所需的额外校正机制。

1.1.2 去中心化局部方法

局部去中心化随机梯度下降（LDSGD） [8] 通过允许在通信前执行多个本地步数来扩展 DSGD，如下所示：

$$x_i^t = \begin{cases} x_i^{t-1} - \alpha g_i^{t-1}, \text{ if } t \bmod \tau \neq 0, \\ \sum_{j \in \mathcal{N}_i} w_{ij}(x_j^{t-1} - \alpha g_i^{t-1}), \text{else}, \end{cases} \tag{2}$$

其中， α 为学习率， w_{ij} 为混合权重， τ 为本地步数， $g_i^t \triangleq \nabla F$（$x_i^t$， ξ_i^t）为随机梯度。

局部梯度跟踪方法（如 KGT [11] 和 LUGT [15],）在标准梯度跟踪框架 [16] 的基础上，通过引入局部更新来降低通信频率。这些方法利用校正项跟踪各工作节点上的全局梯度方向，以尽量减轻数据异质性的影响。类似地，Local Exact Diffusion（LED） [1] 在 Exact Diffusion 算法 [26] 的基础上，允许在通信间隔内执行多个本地步数；该算法通过采用经过修改的共识步骤来消除梯度校正偏差。然而，这些方法并未研究通信-计算重叠，而 OLDSGD 能够完全掩蔽通信延迟，并提供收敛保证。

1.2 主要贡献

本文提出了重叠式局部去中心化随机梯度下降（OLDSGD），这是一种新颖的去中心化训练算法，旨在解决大规模机器学习中的通信瓶颈。我们的主要贡献如下：

- **新颖算法**：我们提出了 OLDSGD，这是首个将局部更新与通信-计算重叠相结合的去中心化训练方法，可完全掩盖通信延迟，且不会产生额外开销。
- **理论保证**：我们证明，在非凸场景下，OLDSGD 的收敛速度与局部去中心化随机梯度下降相同 [8]，即使通信重叠，仍能保持理论有效性。
- **实际影响**：我们展示了 OLDSGD 在多种视觉和语言任务中的效率，包括 VGG11、ResNet18 预训练和 GPT2 微调。OLDSGD 始终优于现有去中心化方法，实现显著加速比，并在某些场景下优于本地随机梯度下降。

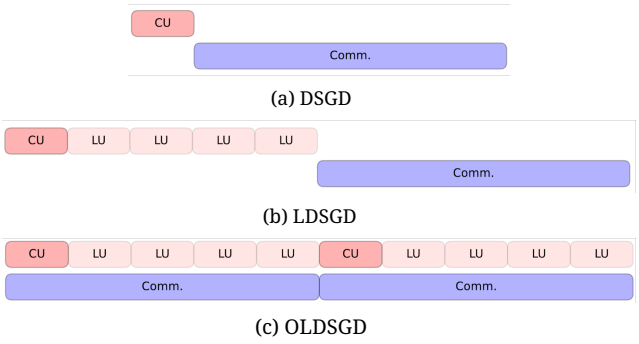
Thes这些贡献使去中心化训练更快速、更具可扩展性，克服了原有方法的局限
赖依赖中式协调或高昂通信开销的方法。

2 算法设计

重叠式局部去中心化随机梯度下降（OLDSGD）通过将局部更新与通信-计算重叠相结合，实现高效的去中心化训练。具体而言，每个智能体在与邻居交换局部模型的同时，执行 τ 次局部梯度更新，交换过程持续 $\mathcal{N}_i$。收到邻居的模型后，智能体使用双随机混合矩阵 $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ 进行混合，并执行梯度更新。算法 1 描述了 OLDSGD 的实现。

算法1 重叠式局部去中心化随机梯度下降（OLDSGD） 要求：智能体数量 N、总迭代次数 T、步长 α、本地步数 τ、混合权重 w_{ij}、所有智能体的初始模型 x_i^0 i。1：初始化：每个智能体 i开始向其在 $\mathcal{N}_i$ 中的所有邻居发送初始模型 x_i^0。2：对于 $t = 1$ 到 T 执行 3：并行地对每个智能体 $i = 1$ 到 N 执行 4：从 $\mathcal{D}_i$ 获取数据批次 ξ_i^{t-1}。5：计算局部随机梯度：$g_i^{t-1} = \nabla F(x_i^{t-1}, \xi_i^{t-1})$。6：如果 t 模 $\tau \neq 0$，则 7：执行局部更新：$x_i^t = x_i^{t-1} - \alpha g_i^{t-1}$。8：否则 9：从 $\mathcal{N}_i$ 中的所有邻居接收 $x_j^{t-\tau}$。10：使用延迟梯度执行共识更新：11：$x_i^t = \sum_{j\in\mathcal{N}_i} w_{ij} x_j^{t-\tau} - \alpha \sum_{k=t-\tau}^{t-1} g_i^k$。12：向 $\mathcal{N}_i$ 中的所有邻居发送 x_i^t。13：结束如果 14：结束循环 15：结束循环 16：返回所有智能体 i 的最终模型 x_i^T。

图1展示了在包含 $\tau = 5$ 个本地步数的一个通信轮次内，DSGD、局部去中心化随机梯度下降（LDSGD）和重叠局部DSGD（OLDSGD）的通信与计算时间线。在 DSGD 中，智能体在每次迭代时都要等待邻居模型交换，因此产生了显著的通信开销。LDSGD 将通信频率降至每轮一次，从而缩短了等待时间。OLDSGD 利用通信-计算重叠，通过使用陈旧模型和 CTA方案，完全消除了等待 [1]，使每轮能够进行多达两倍的梯度更新。



(a) DSGD

(b) LDSGD

(c) OLDSGD

Fig图 1：DSGD、LDSGD 和 OLDSGD 的通信-计算示意图。CU：共识 s upd态，LU：局部更新，Comm.：通信。

OLDSGD 的更新规则如下：

$$x_i^t = \begin{cases} x_i^{t-1} - \alpha g_i^{t-1}, & \text{if } t \bmod \tau \neq 0, \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{t-\tau} - \alpha \sum_{k=t-\tau}^{t-1} g_i^k, & \text{otherwise}, \end{cases} \tag{3}$$

其中， x_i^t 是智能体 i在迭代 t时的模型， α 是步长， g_i^k是局部随机梯度。

与局部去中心化随机梯度下降（公式 (2)）相比， [8][2], OLDSGD 的通信步骤有两点不同。首先，它采用先融合后适应（CTA）方案，仅使用局部梯度 $\sum_{k=t-\tau}^{t-1} g_i^k$，因此无需新鲜邻居梯度。其次，它使用陈旧邻居模型 $x_j^{t-\tau}$，而不是 x_j^{t-1}，从而使通信能够与局部梯度计算重叠，进而掩盖通信延迟。

A kOLDSGD 的一个关键性质是，其平均模型 $\bar{x}^t \triangleq \frac{1}{n}\sum_{i=1}^n x_i^t$遵循类似随机梯度下降的更新 e:

$$\bar{x}^t = \bar{x}^{t-1} - \frac{\alpha}{n} \sum_{i=1}^n g_i^{t-1}, \quad \forall t. \tag{4}$$

要理解其中的原因，对于 t 模 $\tau \neq 0$，该关系显然成立。当 t 是 τ的倍数时，更新变为：

$$\begin{aligned} \bar{x}^t &= \frac{1}{n} \sum_{i=1}^n \left(\sum_{j \in \mathcal{N}_i} w_{ij} x_j^{t-\tau} - \alpha \sum_{k=t-\tau}^{t-1} g_i^k \right) \\ &= \bar{x}^{t-\tau} - \frac{\alpha}{n} \sum_{i=1}^n \sum_{k=t-\tau}^{t-1} g_i^k, \end{aligned}$$

其中，第二个等式源于 W 的双随机性（即 $\sum_j w_{ij} = 1$、 $\sum_i w_{ij} = 1$），而（4）则是递归应用局部更新得到的。

类似 SGD 的平均更新解释了 OLDSGD 为何能够避免重叠式分布式方法中常见的梯度不匹配问题 [28, 27]。CTA方案确保与重叠式方法兼容，

[1]只要通信速度不比梯度计算慢超过 τ 倍，通信开销就能被完全掩盖。[2]由于使用了陈旧模型，OLDSGD 不在 [8] 的框架涵盖范围内。

表 1：OLDSGD 分析中的关键符号。

符号	说明	符号	说明
n	智能体数量	$\mathcal{N}_i$	智能体 i的相邻智能体
f_i, f	局部和全局目标函数	$x_i^t, \bar{x}^t$	局部模型和平均模型
$W = [w_{ij}]$	双随机混合矩阵	λ_2	W 的第二大特征值在幅度
τ	本地步数	α	梯度更新的步长
g_i^k	第 k 次迭代的随机梯度	L	光滑常数
σ^2, M	噪声相关常数	$\zeta^2,\ P$	数据异质性相关常数
C, D	定理 3.6 中的常数	p	$1-\lambda_2^2$，谱常数
$C_0, \ldots\ldots C_3$	式 (5) 和 (6) 中的临时常数	c	通信时间与计算时间之比

将性能下降降至最低，使 OLDSGD 尤其适用于去中心化训练。

类似的算法设计可以直接应用于 Local Exact Diffusion [1] 和 Local Update Gradient Tracking [15]。然而，这些算法的重叠版本会遭受显著的收敛退化，体现在收敛速度和允许的本地步数上。由于其更新机制更为复杂，陈旧误差似乎会产生重大影响。此外，KGT [11] 的更新结构从根本上阻碍了重叠实现。这些观察结果解释了为何本文只专注于开发 OLDSGD。不过请注意，只有当所有智能体的计算速度和通信速度都同质时，才能完全掩蔽延迟。否则，同步延迟仍会导致停滞。有关这些比较方面的详细讨论，请参阅附录。

3 收敛分析

3.1 假设

我们陈述 OLDSGD 收敛分析所需的假设。

假设 3.1。 混合矩阵 W 是双随机矩阵，且对应的通信图是无向且连通的。

假设 3.2。 每个损失目标函数 f_i： $\mathbb{R}^d \to \mathbb{R}$ 均为 L-光滑，满足：

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x-y\|, \quad \forall x, y \in \mathbb{R}^d.$$

假设 3.3。 对于每个智能体 i，随机梯度方差满足：

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\xi_i}\|\nabla F(x_i,\xi_i) - \nabla f_i(x_i)\|_2^2 \le \sigma^2 + \frac{M}{n}\sum_{i=1}^n \|\nabla f_i(x_i)\|_2^2.$$

假设 3.4。 智能体之间的数据异质性有界：

$$\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f(x)\|^2 \le \zeta^2 + P\|\nabla f(x)\|^2.$$

假设 3.5。 全局目标函数 f： $\mathbb{R}^d \to f^*$有有限下界 f^*。

3.2 OLDSGD 的收敛性

正文中使用的主要符号汇总于表 1。如第 2 节所述，OLDSGD 的平均更新遵循类似 SGD 的动态，因此可以针对全局目标函数 f 得到如下形式的下降引理：

$$\mathbb{E}_{t+1} f(\bar{x}^{t+1}) \le f(\bar{x}^t) - \alpha C_0 \|\nabla f(\bar{x}^t)\|^2 + \alpha C_1 \sum_{i=1}^n \|x_i^t - \bar{x}^t\|^2 + \alpha^2 C_2, \tag{5}$$

只要控制住共识误差，右端项就能得到控制。我们的关键洞见是，OLDSGD 保留了与 LDSGD 类似的共识误差界。直观来说，这是因为：（1）OLDSGD 仍会定期执行共识平均；（2）陈旧模型与最新模型之间的距离可以得到适当界定。因此，OLDSGD 与 LDSGD 一样，保持了其核心共识性质。具体而言，我们证明 OLDSGD 中的累积期望共识误差满足以下不等式：

$$C_1 \sum_{t=0}^{T-1} \sum_{i=1}^{n} \mathbb{E}\|x_i^t - \bar{x}^t\|^2 \leq \frac{C_0}{2} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\bar{x}^t)\|^2 + \alpha^2 C_3. \tag{6}$$

结合式 (5) 和式 (6)，我们得到 OLDSGD 在非凸情形下的主要收敛定理，以及一个针对特定步长的推论。

定理 3.6. 给定假设 *3.1–3.5*，由式 (3) 生成的平均模型满足 *s*

$$\frac{\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{8(f^0 - f^*)}{\alpha T} + \frac{8\alpha L}{n}(\frac{\sigma^2}{2} + M\zeta^2) + \frac{1024L^2}{n}\frac{D\tau}{p}\alpha^2,$$

其中

$$\alpha \leq \min\{\frac{1}{L}, \frac{n}{4LM(P+1)}, \frac{n}{LM}, \frac{p}{16L\sqrt{3\tau(2\tau+M)}}, \frac{1}{32L}\sqrt{\frac{pn}{2C\tau}}\},$$

and

$$C = \frac{12(2\tau+M)n(P+1)}{p}, D = \frac{6((2\tau+M)n\zeta^2 + n\sigma^2)}{p}.$$

推论 3.7. 在定理 *3.6* 的条件下，令 $\alpha = \sqrt{\frac{n}{T}}$，我们有 —

$$\frac{\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{8(f^0 - f^*)}{\sqrt{nT}} + \frac{8L(\frac{\sigma^2}{2} + M\zeta^2)}{\sqrt{nT}} + \frac{6144nL^2\tau((2\tau+M))\zeta^2 + \sigma^2)}{p^2T},$$

在

$$T \geq \max\{nL^2, \frac{16L^2M^2(P+1)^2}{n}, \frac{L^2M^2}{n}, \frac{768nL^2\tau(2\tau+M)}{p^2}, \frac{24576nL^2\tau(2\tau+M)(P+1)}{p^2}\}.$$

注 3.8. 如推论 *3.7* 所示，*OLDSGD* 达到 $\mathcal{O}(\frac{1}{\sqrt{nT}} + \frac{1}{T})$ 的收敛速率，与此前针对非凸情形建立的局部去中心化随机梯度下降结果 *[8, 10]* 相当。也就是说，与局部去中心化随机梯度下降相比，重叠更新 (3) 不会削弱任何理论保证。

4 实验

我们在视觉和语言任务上评估重叠式局部去中心化随机梯度下降（OLDSGD），并将其与当前先进的去中心化方法进行比较：局部去中心化随机梯度下降（LDSGD） [8], KGT [11], LUGT [15], LED [3] [1], ，以及采用环形全归约通信的本地随机梯度下降（LSGD） [20] 。实验在一台配备 8 块 NVIDIA RTX3090 GPU（每块显存 24GB）的服务器上使用 PyTorch 2.0.1 进行。各实现均遵循原始算法描述，未采用额外加速。我们测试了同质和异质数据分布，并模拟模型通信延迟，以评估运行时间收敛和可扩展性。

4.1 实验设置、任务和超参数

设置。除非另有说明，实验均采用无向环形拓扑。为了比较不同算法相对于时间的收敛情况，我们对通信和计算所产生的延迟进行建模。对于计算，我们假设每个智能体每步花费 1 个时间单位来计算一个本地（随机）梯度，并忽略其他操作（例如通过加法或减法进行模型更新），因为与之相比，这些操作所耗时间可以忽略不计。对于通信，每个智能体在每个共识步骤中需花费 c 个时间单位，将其模型参数传输给邻居。表2给出了各算法每次迭代运行时间与 τ（本地步数）和 c 的函数关系，最后一列显示归一化运行时间

表2：不同算法的每次迭代运行时间，以及在 $\tau = c$ 时相对于 OLDSGD 的归一化值。LSGD 采用环形全归约进行通信。

算法	每次迭代运行时间	归一化值（$\tau = c$）
OLDSGD	$\max\{\tau, c\}$	1
LDSGD	$\tau + c$	2
KGT	$\max\{\tau, c\} + c$	2
LED	$\tau + c$	2
LUGT	$\tau + 2c$	3
LSGD	$\tau + \frac{2(n-1)}{n} c$	$1 + \frac{2(n-1)}{n}$

当 $\tau = c$时，相对于 OLDSGD。OLDSGD 的运行时间得益于通信-计算重叠，与现有方法相比提升了 50% 以上。

任务和数据集。 我们评估三个任务：

1. **VGG11** [19]在 CIFAR-10 上进行训练（50,000 张训练图像，10,000 张测试图像），使用 9 个智能体。2. **ResNet18** [5] 在 CIFAR-10 上进行训练，使用 9 个智能体。3. **GPT2-Small** [17] 在 WikiText-2 [13]（2M 个词元）上进行微调，使用 8 个智能体（仅同质）。

对于 VGG11 和 ResNet18 训练，我们评估了同质和异质数据分布。在同构设置中，我们将数据集均匀分配给各智能体，以确保类别标签表示均衡。在异质设置中，我们按类别标签划分数据集，以引入数据偏斜。具体而言，每个智能体 70% 的数据根据特定标签分配，其余 30% 则从所有标签中均匀采样。对于 GPT2 微调，由于计算限制，我们仅评估了同质数据分布。

超参数。我们设置超参数以确保公平比较。我们为所有任务选择固定学习率 $\alpha = 0.01$，该学习率根据初步实验确定为合适值，并且不采用学习率调度，以保持一致性。对于 GPT2 微调，我们采用最大梯度范数为 1 的梯度裁剪，以稳定大语言模型的训练。本地批次大小设置如下：逻辑回归为 32，VGG11 和 ResNet18 为 8，GPT2 微调为 2。

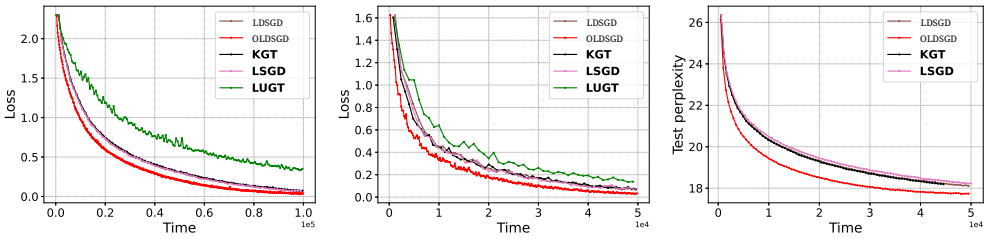
4.2 比较随时间变化的收敛情况

我们分析了通信延迟为 $c = 1$ 和 $c = 5$ 时的收敛性能，并以（模拟的）实际运行时间内的损失下降来衡量。其中，c 表示相对于单次随机梯度计算时间的通信时间（设为 1 个单位）。$c = 1$ 情景假设通信速度与计算速度相同，而 $c = 5$ 则模拟通信速度慢五倍的情况，这反映了具有网络延迟的现实中常见的去中心化训练环境。为确保比较公平，我们针对每种算法评估了广泛的本地步数范围 $\tau \in \{1, 3, 5, 10, 15, 20, 30, 40\}$，并为每种方法选取能使收敛速度最大化的最优 τ，从而近似估计其在给定条件下可达到的最佳性能。

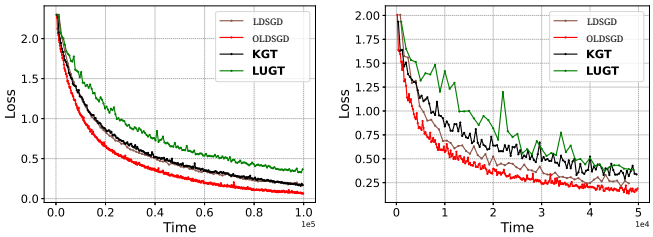
图 2 和图 3 展示了各种算法在不同任务上的收敛性能。顶行显示同质数据分布下的结果，底行则对应异质数据分布。对于 GPT2 微调，鉴于 LUGT 和 LED 在较简单任务中容易发散，我们不纳入这两种方法；同时，为控制计算开销，评估仅限于同构设置。对于 GPT2，纵轴表示平均模型在测试数据集上的困惑度，因为计算训练集指标的开销很大。对于其他任务，纵轴表示平均模型的训练损失，在我们的分析中记为 $f(\bar{x}^k)$。

在所有评估场景中，OLDSGD 的收敛速度始终优于其他算法。值得注意的是，尽管梯度跟踪和精确扩散方法在理论上具有优势，但它们

[3]由于 LED 持续发散，正文中未涵盖 LED 的结果；详情请参阅附录。

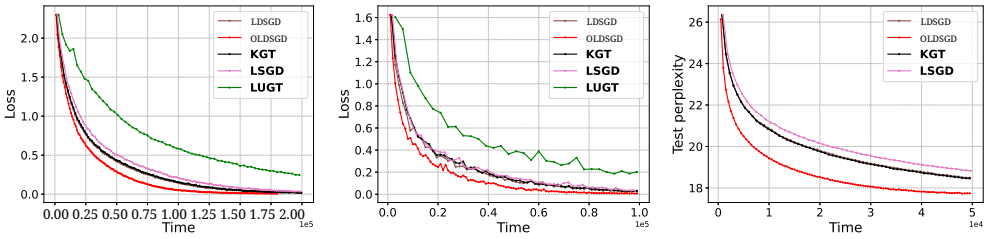


(a) 同质-VGG11 (b) 同质-ResNet18 (c) 同质-GPT2

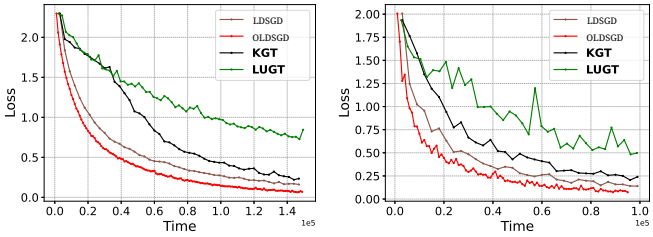


(d) 异质-VGG11 (e) 异质-ResNet18

图 2：不同任务下随时间变化的收敛情况 (c=1)。第一行展示所有同质情况，第二行展示所有异质情况。



(a) 同质-VGG11 (b) 同质-ResNet18 (c) 同质-GPT2



(d) 异质-VGG11 (e) 异质-ResNet18

图3：不同任务（c=5）下的时间收敛情况。第一行展示所有同质情况，第二行展示所有异质情况。

在相同学习率下，通常比原始梯度下降类方法（如 OLDSGD 和 LDSGD）收敛得更慢（LED 则会发散；见附录 B）。具体而言，LUGT 将梯度跟踪简单扩展至包含本地步数后，收敛速度更慢；尤其对于 $\tau > 1$，其损失曲线明显落后于 OLDSGD（见图 2 和图 3；进一步分析请参阅附录）。得益于其更新机制，KGT 更为稳健，在大多数任务（如 VGG11 和 GPT2 微调）中，能够以 $c = 1$ 达到与 LDSGD 相当的收敛速率；但在异质数据下的 ResNet18 任务中，其损失下降明显更慢。在 $c = 5$ 下，KGT 在 VGG11 和 ResNet18 的异质数据场景中的表现进一步下降，其图 3 中更平缓的损失曲线印证了这一点。

表3量化了 OLDSGD 相较于其他算法的加速比，重点关注同质数据分布的情况（由于篇幅限制，异质设置的结果放在附录中）。加速比按达到预设测试准确率所需的相对时间计算（以 OLDSGD 的运行时间为基准）：对于 CIFAR-10 任务，目标是达到预设测试准确率；对于 GPT2 微调，目标是达到预设困惑度。OLDSGD 相较于 LDSGD 的加速比几何平均值超过 1.64，表明只需对算法进行少量修改，就能显著提升性能。值得注意的是，当 $c = 5$ 时，OLDSGD 的效率优势更加明显；此时通信速度比计算慢五倍，这一点可从更高的加速比看出（例如，GPT2 在 $c = 5$ 时为 2.65×，而在 $c = 1$ 时为 1.95×）。这一趋势凸显了 OLDSGD 在通信受限环境中的有效性，而这类环境在现实世界的去中心化系统中十分常见，例如边缘计算或联邦学习。值得一提的是，在 GPT2 微调中，OLDSGD 相较于 LSGD 实现了显著加速（2.08× 和 3.32×）；图3中的困惑度曲线显示，OLDSGD 的收敛速度明显更快，而基线方法的曲线则趋于平缓。这一出色表现凸显了 OLDSGD 扩展至现代 Transformer 架构的潜力，在这类架构中，通信瓶颈至关重要。

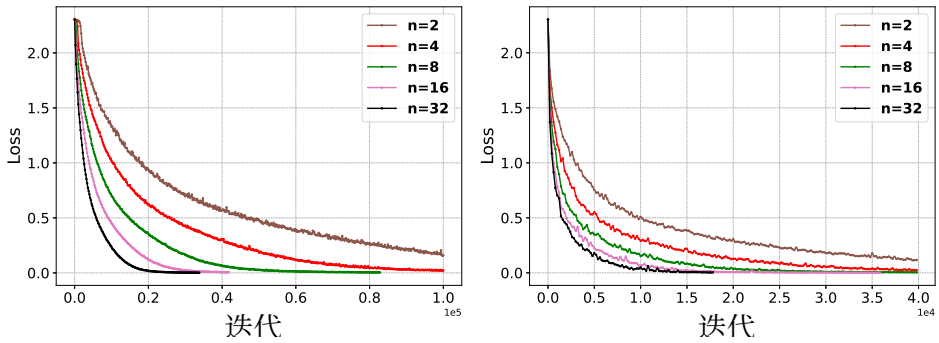
表3：OLDSGD 相较于现有方法的加速比（加速比越高越好）

	VGG11		ResNet18		GPT2		
算法	$c = 1$	$c = 5$	$c = 1$	$c = 5$	$c = 1$	$c = 5$	**几何平均值**
LDSGD	1.23×	1.26×	1.50×	1.62×	1.95×	2.65×	1.64×
KGT	1.26×	1.30×	1.42×	1.73×	1.86×	2.54×	1.63×
LSGD	1.34×	1.64×	1.84×	2.18×	2.08×	3.32×	1.98×
LUGT	2.68×	3.53×	1.98×	4.62×	-	-	3.05×

4.3 可扩展性

先前的实验表明，OLDSGD 在各种任务中都表现稳健，但计算资源限制了可测试的智能体数量。为评估其可扩展性，我们在 CIFAR-10 上训练 VGG11 和 ResNet18，并测量加速比随智能体数量的变化，其中本地步数固定为 $\tau = 5$。我们在环形拓扑中测试了从 2 到 32 个智能体，并观察到显著的加速效果，如图4所示。OLDSGD 在最多 16 个智能体时实现了近线性加速；与单个智能体相比，VGG11 和 ResNet18 的收敛速度分别提升约 14× 和 13×。超过 16 个智能体后，加速比的增长略有放缓。这主要是因为环形拓扑的图连通性较差，随着网络规模扩大，限制了信息的高效交换。

这些结果凸显了 OLDSGD 的可扩展性及其在大规模去中心化训练中的潜力。



(a) VGG11 (b) ResNet18

图4：OLDSGD 在 VGG11 (a) 和 ResNet18 (b) 上的可扩展性（CIFAR-10，环形拓扑，$\tau = 5$）。在最多 16 个智能体时实现了近线性加速比（VGG11 为 14×，ResNet18 为 13×），由于环形拓扑的限制，超过这一数量后收益递减。损失曲线表明，随着并行度提高，收敛速度加快。

5 结论

我们提出了重叠式局部去中心化随机梯度下降（OLDSGD），这是一种新颖的去中心化训练算法，通过重叠通信与计算来最大限度地减少同步开销。与局部去中心化随机梯度下降相比，OLDSGD 将每次迭代运行时间缩短至最多 50%，同时在非凸条件下保持相同的理论保证。在实验中，OLDSGD 相较于 LDSGD 实现了 1.64× 的几何平均加速比，在 GPT2 微调中加速比最高达 3.32×；它在联邦学习和边缘计算中普遍存在的通信受限场景（$c = 5$）下表现出色。只需对算法进行最小幅度的修改，OLDSGD 就能显著缩短运行时间，同时不牺牲理论鲁棒性，因此是去中心化训练中一种极具吸引力、可直接替换的方案。

参考文献

[1] S. A. ALGHUNAIM，《去中心化优化与学习的局部精确扩散》，IEEE自动控制汇刊，（2024）。[2] Y. ARJEVANI、O. SHAMIR 和 N. SREBRO，《具有延迟更新的随机梯度下降的紧致收敛分析》，载于《算法学习理论》，PMLR，2020年，第111–132页。[3] T. BEN-NUN 和 T. HOEFLER，《揭秘并行与分布式深度学习：深入的并发分析》，《ACM计算调查》（CSUR），52（2019），第1–43页。[4] R. CHEN、Q. WAN、P. PRAKASH、L. ZHANG、X. YUAN、Y. GONG、X. FU 和 M. PAN，《Workie-Talkie：通过对比正则化重叠计算与通信，加速联邦学习》，载于《IEEE/CVF国际计算机视觉大会论文集》，2023年，第16999–17009页。[5] K. HE、X. ZHANG、S. REN 和 J. SUN，《用于图像识别的深度残差学习》，载于《IEEE计算机视觉与模式识别会议论文集》，2016年，第770–778页。[6] P. KAIROUZ、H. B. MCMAHAN、B. AVENT、A. BELLET、M. BENNIS、A. N. BHAGOJI、K. BONAWITZ、Z. CHARLES、G. CORMODE、R. CUMMINGS 等，《联邦学习的进展与开放问题》，《机器学习基础与趋势》，14（2021），第1–210页。[7] S. KALE、A. DOUILLARD 和 Y. DONCHEV，《DiLoCo中重叠通信与计算的主动更新》，arXiv预印本 arXiv:2502.12996，（2025）。[8] A. KOLOSKOVA、N. LOIZOU、S. BOREIRI、M. JAGGI 和 S. STICH，《具有变化拓扑和局部更新的去中心化随机梯度下降统一理论》，载于《国际机器学习大会》，PMLR，2020年，第5381–5393页。[9] S. LI、Y. ZHAO、R. VARMA、O. SALPEKAR、P. NOORDHUIS、T. LI、A. PASZKE、J. SMITH、B. VAUGHAN、P. DAMANIA 等，《PyTorch分布式：加速数据并行训练的实践经验》，arXiv预印本 arXiv:2006.15704，（2020）。[10] X. LI、W. YANG、S. WANG 和 Z. ZHANG，《通信高效的局部去中心化随机梯度下降方法》，arXiv预印本 arXiv:1910.09126，（2019）。[11] Y. LIU、T. LIN、A. KOLOSKOVA 和 S. U. STICH，《利用本地步数进行去中心化梯度跟踪》，《优化方法与软件》，（2024），第1–28页。[12]B. MCMAHAN、E. MOORE、D. RAMAGE、S. HAMPSON 和 B. A. Y ARCAS，《基于去中心化数据的深度网络通信高效学习》，载于《人工智能与统计学》，PMLR，2017年，第1273–1282页。[13] S. MERITY、C. XIONG、J. BRADBURY 和 R. SOCHER，《指针哨兵混合模型》，arXiv预印本 arXiv:1609.07843，（2016）。[14] A. NEDIC 和 A. OZDAGLAR，《面向多智能体优化的分布式次梯度方法》，《IEEE自动控制汇刊》，54（2009），第48–61页。

[15] E. D. H. NGUYEN、S. A. ALGHUNAIM、K. YUAN 和 C. A. URIBE, 《局部更新下梯度跟踪的性能》, 载于《2023年第62届IEEE决策与控制会议》（CDC）, IEEE, 2023年, 第4309–4313页。[16] S. PU 和 A. NEDI ´C, 《分布式随机梯度跟踪方法》, 《数学规划》, 187（2021）, 第409–457页。[17] A. RADFORD、J. WU、R. CHILD、D. LUAN、D. AMODEI、I. SUTSKEVER 等, 《语言模型是无监督多任务学习器》, OpenAI博客, 1（2019）, 第9页。[18] S. SHEN、L. XU、J. LIU、X. LIANG 和 Y. CHENG, 《更快的分布式深度网络训练：计算与通信解耦的随机梯度下降》, arXiv预印本 arXiv:1906.12043, （2019）。[19] K. SIMONYAN 和 A. ZISSERMAN, 《用于大规模图像识别的超深卷积网络》, arXiv预印本 arXiv:1409.1556, （2014）。[20]S. U. STICH, 《本地随机梯度下降快速收敛且通信量少》, arXiv预印本 arXiv:1805.09767, （2018）。[21] W. SUN、Z. QIN、W. SUN、S. LI、D. LI、X. SHEN、Y. QIAO 和 Y. ZHONG, 《Co2：具有完全通信-计算重叠的高效分布式训练》, arXiv预印本 arXiv:2401.16265, （2024）。[22] J. VERBRAEKEN、M. WOLTING、J. KATZY、J. KLOPPENBURG、T. VERBELEN 和 J. S. RELLERMEYER, 《分布式机器学习综述》, 《ACM计算调查》（CSUR）, 53（2020）, 第1–33页。[23] J. WANG 和 G. JOSHI, 《协作式随机梯度下降：通信高效随机梯度下降算法设计与分析的统一框架》, arXiv预印本 arXiv:1808.07576, （2018）。[24] J. WANG、H. LIANG 和 G. JOSHI, 《重叠式本地随机梯度下降：隐藏分布式随机梯度下降中通信延迟的算法方法》, 载于《ICASSP 2020—2020年IEEE国际声学、语音与信号处理会议》（ICASSP）, IEEE, 2020年, 第8871–8875页。[25] Z. WANG、J. ZHANG、X. WU 和 M. JOHANSSON, 《从承诺到实践：实现高性能去中心化训练》, arXiv预印本 arXiv:2410.11998, （2024）。[26] K. YUAN、B. YING、X. ZHAO 和 A. H. SAYED, 《用于分布式优化与学习的精确扩散——第一部分：算法开发》, 《IEEE信号处理汇刊》, 67（2018）, 第708–723页。[27] Y. ZHOU、Q. YE 和 J. LV, 《采用补偿式重叠联邦平均的通信高效联邦学习》, 《IEEE并行与分布式系统汇刊》, 33（2021）, 第192–205页。[28] L. ZHU、H. LIN、Y. LU、Y. LIN 和 S. HAN, 《延迟梯度平均：容忍联邦学习中的通信延迟》, 载于《神经信息处理系统进展》, 34（2021）, 第29995–30007页。

A 迭代运行时间分析

表2展示了不同算法的每次迭代运行时间，下面将详细分析各算法逐次迭代的运行时间。设有 n个智能体，每个智能体需要1个时间单位来计算（随机）梯度，并需要 c 个时间单位来传输模型大小的数据。每个通信轮次包含 τ 个本地步数。讨论时沿用参考文献中的原始记号。

- LSGD：环形全归约将模型划分为 n 个数据块，每个数据块传输2($n-1$)次，因此通信时间为 $^{2\ (n-1)\ n}\ c$。计入局部梯度计算后，总运行时间为 $\tau + ^{2\ (n-1)\ n}\ \ c$。
- OLDSGD：如图1所示。

• LDSGD：如图1所示。

• KGT：包括模型 $\mathbf{x}_{(t)\,i}$ 和累积梯度$\mathbf{z}_{(t)\,i}$。尽管参考文献中没有提及，模型$\mathbf{x}_{(t)\,i}$ 的通信实际上可以与梯度计算重叠，因为每次更新都使用上一个通信轮次的模型。这会产生 $\max\{\tau, c\}$ 项。然而，传递累积梯度$\mathbf{z}_{(t)\,i}$ 必须等到某个智能体完成本地步数后才能进行，因此会使总运行时间增加 c。

• LUGT：该方法需要通信模型和跟踪变量，因此通信时间为 $2c$。且无法进行重叠，因此运行时间为 $\tau + 2c$。

• LED：与精确扩散[26],类似，LED 的通信开销与模型大小相同，且无法进行重叠，因此运行时间为 $\tau + c$。

B LED/LUGT 中重叠的可行性

如第 2 节所述，重叠原理可以扩展到 LED 和 LUGT。算法2和算法3（使用其原始记号，可能与我们的约定有所不同）展示了完全重叠的通信-计算实现，并且保留了与其非重叠对应方法相同的平均更新性质。然而，如图5所示，这种重叠方法会导致显著的性能下降；图中呈现了低数据异质性水平下 ResNet18 训练的损失曲线。由于数据异质性较低，该任务更为简单，这一点也体现在 OLDSGD 更快的收敛速度上。虽然带有 $\tau = 1$ 的 OLGT 保持了与标准 LUGT 相当的性能，但增加到 $\tau = 5$ 后会导致显著减速，其影响超过了每次迭代运行时间方面的优势。图5还突显了 LED 固有的发散问题，这也解释了我们为何在第 4.2 节的比较分析中排除 LED。在所有测试配置下，OLED 的发散速度甚至快于原始LED。这些结果表明，对于 LED 和 LUGT，朴素的重叠实现均不可行。开发稳定的重叠变体仍然是一项开放的研究挑战，需要采用比直接组合方法更复杂的方案。

算法 2 重叠式 LUGT（OLGT）

Input: $x_i^0 = 0 \in \mathbb{R}^m$, $y_i^0 = \alpha \nabla f_i(x_i^0)$, $\alpha > 0$, $\eta > 0$, $T_0 \in \mathbb{Z}_{\geq 0}$, $K \in \mathbb{Z}_+$
Define: $\tau = \{0, T_0, 2T_0, 3T_0, \ldots\}$
for $k = 0$ **to** $K - 1$ **do**
  if $k \in \tau$ **then**
    $x_i^{k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{k+1-T_0} - \eta \sum_{t=k+1-T_0}^{k} y_i^t$
    $y_i^{k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} y_j^{k+1-T_0} + \alpha \sum_{t=k+1-T_0}^{k} (\nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t))$
  else
    $x_i^{k+1} = x_i^k - \eta y_i^k$
    $y_i^{k+1} = y_i^k + \alpha \nabla f_i(x_i^{k+1}) - \alpha \nabla f_i(x_i^k)$
  end if
end for

算法 3重叠式本地精确扩散（OLED）

Input: x_i^0, $\alpha > 0$, $\beta > 0$, and τ.
Initialize: $y_i^0 = x_i^0 - \sum_{j \in \mathcal{N}_i} w_{ij} x_j^0$ (or $y_i^0 = 0$)
for $r = 0, 1, 2, \ldots$ **do**
  Set $\phi_{i,0}^r = x_i^r$
  for $t = 0$ **to** $\tau - 1$ **do**
    $\phi_{i,t+1}^r = \xi_{i,t}^r - \alpha \nabla F_i(\phi_{i,t}^r; \xi_{i,t}^r) - \beta y_i^r$
  end for
  $x_i^{r+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^r + (\phi_{i,\tau}^r - x_i^r)$
  $y_i^{r+1} = y_i^r + \nabla f_i(x_i^{r+1}) - \nabla f_i(x_i^r)$
end for

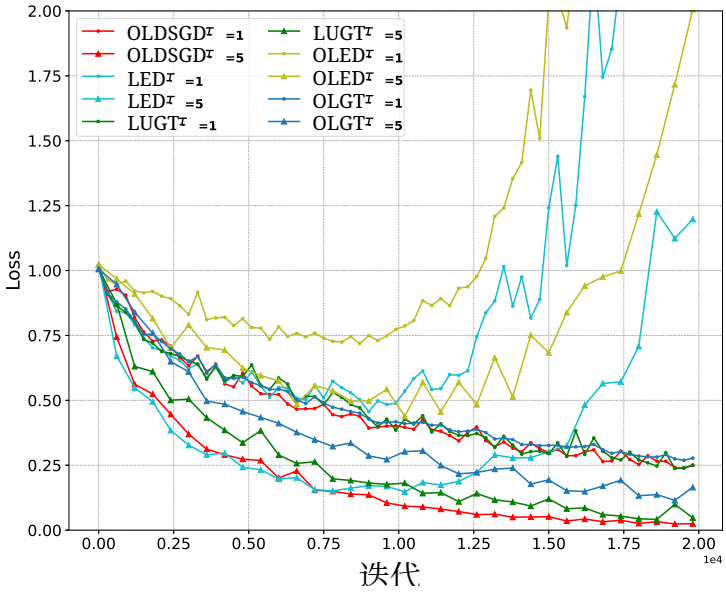


图5：低数据异质性水平下，ResNet18 训练损失随迭代变化的曲线。在所有情况下，LED/OLED 均发散；采用本地步数时，OLGT 的表现不如 LUGT。

C OLDSGD 的理论收敛性

本节将在非凸条件下从理论上证明重叠局部DSGD的收敛性。

C.1 符号说明

我们介绍以下分析中使用的符号。读者也可参阅表 1 作为补充。

首先，我们定义对于任意迭代 t,

$$t' \triangleq (\lfloor \frac{t}{\tau} \rfloor - 1)\tau \quad \text{and} \quad t'' \triangleq \lfloor \frac{t}{\tau} \rfloor \tau,$$

其中， t'' 是最近一次共识迭代。在 t''时，使用在 t' 更新的模型。由于重叠，OLDSGD 的递归范围从 τ 扩展到 2τ，如下文所示。

为得到紧凑矩阵表示，我们定义

$$X_t = [x_t^1, ..., x_t^n] \quad \text{and} \quad G_t = [g_t^1, ..., g_t^n],$$

其中 $g_t^i = \nabla F(x_t^i, \ \xi_t^i)$。

我们将期望共识误差定义为

$$E_t \triangleq \mathbb{E}\|X_t - \bar{x}_t \mathbf{1}_n^T\|_F^2,$$

其中 $\|\cdot\|_F$ 表示弗罗贝尼乌斯范数。

令 λ_2 表示 W 中绝对值第二大的特征值。混合后共识误差的单步收缩率为 $p = (1-\lambda_2)^2$。

C.2 收敛分析

我们首先介绍由更新式（3）得到的几个有用等式。首先，我们重申平均模型的更新

$$\bar{x}^{t+1} = \bar{x}^t - \frac{\alpha}{n} \sum_{i=1}^{n} g_i^t. \tag{7}$$

正如正文所详述，尽管通信与计算存在重叠，平均更新仍与随机梯度下降相似。因此，重叠所导致的陈旧性不会影响平均更新，从而避免了大多数分布式方法中的梯度不匹配问题 [28, 7, 4]。

接下来，我们给出任意迭代时模型与 t' 之间的关系，

$$x_t^i = \sum_j w_{ij} x_{t'}^j - \alpha \sum_{k=t'}^{t-1} g_k^i \tag{8}$$

and

$$\bar{x}_t = \bar{x}_{t'} - \frac{\alpha}{n} \sum_{k=t'}^{t-1} \sum_{i=1}^{n} g_k^i. \tag{9}$$

方程 (8) 和 (9) 描述了模型在一个 2τ 周期内的演化，这对于分析共识误差至关重要。从某种意义上说，OLDSGD 与采用 2τ 本地步数的局部去中心化随机梯度下降非常相似。然而，OLDSGD 不属于 [8], 提出的框架，该框架

$$X^{t+1} = (X^t - G^t)W^t,$$

以 W^t 为周期，分别采用 I 或 W。这是因为，与 OLDSGD 使用的陈旧模型 $\{x_{t'}^j\}_{j\in\mathcal{N}_i}$ 相比，该框架采用最新邻居模型。
根据 (7)，可以直接推导出以下下降引理。

引理 C.1. （下降引理）给定假设 *3.2*、*3.3* 和 *3.4*，我们有

$$\mathbb{E}_{t+1} f(\bar{x}^{t+1}) \le f(\bar{x}^t) - \frac{\alpha}{4}\|\nabla f(\bar{x}^t)\|^2 + \frac{2\alpha L^2}{n}\sum_{i=1}^{n}\|x_i^t - \bar{x}^t\| + \frac{\alpha^2 L}{n}(\frac{\sigma^2}{2} + M\zeta^2),$$

其中 $\alpha \le$ 最小值$\{\frac{1}{L}; \frac{n}{4LM^{(P+1)}}\ \frac{n}{LM}\}$。

证明。 由式（7）可得

$$\begin{aligned}
&\mathbb{E}_{t+1} f(\bar{x}^{t+1}) \\
&= \mathbb{E}_{t+1} f\left(\bar{x}^{(t)} - \frac{\alpha}{n}\sum_{i=1}^{n}\nabla F(x_i^t, \xi_i^t)\right) \\
&\le f(\bar{x}^t) - \alpha\mathbb{E}_{t+1}\langle\nabla f(\bar{x}^t), \frac{1}{n}\sum_{i=1}^{n}\nabla F(x_i^t,\xi_i^t)\rangle + \frac{\alpha^2 L}{2}\mathbb{E}_{t+1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla F(x_i^t,\xi_i^t)\right\|^2 \\
&\le f(\bar{x}^t) - \alpha\langle\nabla f(\bar{x}^t), \frac{1}{n}\sum_{i=1}^{n}\nabla f(x_i^t)\rangle + \frac{\alpha^2 L}{2}\left(\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}(\nabla F(x_i^t,\xi_i^t) - \nabla f_i(x_i^t))\|^2 + \|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^t)\|^2\right) \\
&\le f(\bar{x}^t) - \frac{\alpha}{2}\|\nabla f(\bar{x}^t)\|^2 - \frac{\alpha}{2}(1-\alpha L)\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^t)\|^2 + \frac{\alpha}{2}\|\nabla f(\bar{x}^t) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^t)\|^2 \\
&+ \frac{\alpha^2 L}{2}\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}(\nabla F(x_i^t,\xi_i^t) - \nabla f_i(x_i^t))\|^2 \\
&\overset{\alpha\le 1/L}{\le} f(\bar{x}^t) - \frac{\alpha}{2}\|\nabla f(\bar{x}^t)\|^2 + \frac{\alpha L^2}{n}\sum_i\|x_i^t - \bar{x}^t\| + \frac{\alpha^2 L}{2n^2}\mathbb{E}_{t+1}\sum_{i=1}^{n}\|(\nabla F(x_i^t,\xi_i^t) - \nabla f_i(x_i^t))\|^2 \\
&\le f(\bar{x}^t) - \frac{\alpha}{2}\|\nabla f(\bar{x}^t)\|^2 + \frac{\alpha L^2}{n}\sum_i\|x_i^t - \bar{x}^t\| + \frac{\alpha^2 LM}{2n^2}\sum_i\|\nabla f_i(x_i^k)\|^2 + \frac{\alpha^2 L\sigma^2}{2n},
\end{aligned}$$

其中，第三个不等式由极化恒等式得出，第四个和第五个不等式由假设 3.2 和假设 3.3 得出。

根据假设 3.4，我们有

$$\begin{aligned}\sum_i \|\nabla f_i(x_i^t)\|^2 &= \sum_i \|\nabla f_i(x_i^t) \mp \nabla f_i(\bar{x}^t)\|^2 \\ &\le 2L^2 \sum_i \|x_i^t - \bar{x}^t\|^2 + 2\sum_i \|\nabla f_i(\bar{x}^t)\|^2 \\ &\le 2L^2 \sum_i \|x_i^t - \bar{x}^t\|^2 + 2n\zeta^2 + 2n(P+1)\|\nabla f(\bar{x}^t)\|^2,\end{aligned}$$

其中 $P+1$源自 $\|\nabla f_i(\bar{x}^t)\| \le 2\|\nabla f_i(\bar{x}^t) - \nabla f(\bar{x}^t)\| + 2\|\nabla f(\bar{x}^t)\|$。
合并所有内容

$$\begin{aligned}&\mathbb{E}_{t+1} f(\bar{x}^{t+1}) \\ &\le f(\bar{x}^t) - \alpha(\frac{1}{2} - \frac{\alpha LM(P+1)}{n})\|\nabla f(\bar{x}^t)\|^2 + (\frac{\alpha L^2}{n} + \frac{\alpha^2 L^3 M}{n^2})\sum_i \|x_i^t - \bar{x}^t\| + \frac{\alpha^2}{n}(\frac{L\sigma^2}{2} + LM\zeta^2) \\ &\overset{\alpha \le \frac{n}{4LM(P+1)}}{\le} f(\bar{x}^t) - \frac{\alpha}{4}\|\nabla f(\bar{x}^t)\|^2 + (\frac{\alpha L^2}{n} + \frac{\alpha^2 L^3 M}{n^2})\sum_i \|x_i^t - \bar{x}^t\| + \frac{\alpha^2 L}{n}(\frac{\sigma^2}{2} + M\zeta^2) \\ &\overset{\alpha \le \frac{n}{LM}}{\le} f(\bar{x}^t) - \frac{\alpha}{4}\|\nabla f(\bar{x}^t)\|^2 + \frac{2\alpha L^2}{n}\sum_i \|x_i^t - \bar{x}^t\| + \frac{\alpha^2 L}{n}(\frac{\sigma^2}{2} + M\zeta^2)\end{aligned}$$

□

为了界定引理 C.1 的右端项，我们需要界定共识误差。回顾一下，在 OLDSGD 中，每次迭代 t，每个智能体都会在 t''执行共识步骤，并使用在 t'更新的陈旧模型。因此，我们先建立 t' 与 t处模型之间的递推关系。

引理 C.2。（共识误差部分 *1*）给定假设 *3.1–3.4*，当 $\alpha \le \frac{p}{16L\sqrt{3\tau(2\tau+M)}}$时，我们有

$$E_t \le (1 - \frac{p}{2})E_{t'} + \frac{p}{64\tau}\sum_{k=t'}^{t-1} E_k + C\mathbb{E}\sum_{k=t'}^{t-1}\alpha^2\|\nabla f(\bar{x}_k)\|^2 + D\sum_{k=t'}^{t-1}\alpha^2,$$

其中 $C = \frac{12}{(2\tau+M)n(P+1)\,p}$, $D = \frac{6}{((2\tau+M)n\zeta^2 + n\sigma^2)\,p}$，且 $p = 1 - \lambda_2^2$。

证明。根据方程 (8) 和 (9)，我们有

$$X_t - \bar{x}_t \mathbf{1}_n^T = (X_{t'}W - \bar{x}_{t'}\mathbf{1}_n^T) - \alpha\sum_{k=t'}^{t-1} G_k(I - J),$$

其中 $J = \frac{\mathbf{1}_n\mathbf{1}_n^T}{n}$和 $t' = (\lfloor \frac{t}{\tau} \rfloor - 1)\tau$。取平方范数并求期望

$$\begin{aligned}\mathbb{E}\|X_t - \bar{x}_t\mathbf{1}_n^T\|_F^2 &= \mathbb{E}\|(X_{t'}W - \bar{x}_{t'}\mathbf{1}_n^T) - \alpha\sum_{k=t'}^{t-1} G_k(I-J)\|_F^2 \\ &\le \lambda_2^2(1+\epsilon)\mathbb{E}\|X_{t'} - \bar{x}_{t'}\mathbf{1}_n^T\|_F^2 + (1 + \frac{1}{\epsilon})\alpha^2 \underbrace{\mathbb{E}\|\sum_{k=t'}^{t-1} G_k(I-J)\|_F^2}_{T_1} \qquad (10)\end{aligned}$$

我们对 T_1 作如下界定

$$
\begin{aligned}
T_1 &= \mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla F(x_k^i,\xi_k^i)-\frac{1}{n}\sum_{k=t'}^{t-1}\sum_{j=1}^{n}\nabla F(x_k^j,\xi_k^j)\right\|^2 \\
&= \mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla F(x_k^i,\xi_k^i)\right\|^2 - n\mathbb{E}\left\|\frac{1}{n}\sum_{k=t'}^{t-1}\sum_{j=1}^{n}\nabla F(x_k^j,\xi_k^j)\right\|^2 \\
&\le \mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla F(x_k^i,\xi_k^i)\right\|^2 \\
&\le 2\mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}(\nabla F(x_k^i,\xi_k^i)-\nabla f_i(x_k^i))\right\|^2 + 2\mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla f_i(x_k^i)\right\|^2 \\
&= 2\mathbb{E}\sum_{i=1}^{n}\sum_{k=t'}^{t-1}\left\|(\nabla F(x_k^i,\xi_k^i)-\nabla f_i(x_k^i))\right\|^2 + 2\mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla f_i(x_k^i)\right\|^2 \\
&\le 2\sum_{k=t'}^{t-1}(n\sigma^2+M\sum_i\|\nabla f_i(x_k^i)\|^2) + 2\mathbb{E}\sum_{i=1}^{n}\left\|\sum_{k=t'}^{t-1}\nabla f_i(x_k^i)\right\|^2 \\
&\overset{t'-t\le 2\tau}{\le} (4\tau+2M)\sum_{k=t'}^{t-1}\underbrace{\sum_i\|\nabla f_i(x_k^i)\|^2}_{T_2} + 2\sum_{k=t'}^{t-1}n\sigma^2,
\end{aligned}
$$

其中，第三个等式是因为 ζ_t^i 与 x_{t-1}^i相互独立。我们进一步将 T_2 界定如下

$$
\begin{aligned}
T_2 &\le 2\mathbb{E}\sum_i\left(\left\|\left(\nabla f_i(x_k^i)-\nabla f_i(\bar{x}_k)\right)\right\|^2+\|\nabla f_i(\bar{x}_k)\|^2\right) \\
&\le 2L^2\mathbb{E}\|X_k-\bar{x}_k\mathbf{1}_n^T\|_F^2 + 2n(\zeta^2+(P+1)\|\nabla f(\bar{x}_k)\|^2) \\
&= 2L^2E_k + 2n(P+1)\|\nabla f(\bar{x}_k)\|^2 + 2n\zeta^2
\end{aligned}
$$

因此，我们有

$$
T_1 \overset{t-t'\le 2\tau}{\le} 4(2\tau+M)L^2\sum_{k=t'}^{t-1}E_k + 4n(2\tau+M)(P+1)\sum_{k=t'}^{t-1}\|\nabla f(\bar{x}_k)\|^2 + 2n(2\tau+M)\sum_{k=t'}^{t-1}\zeta^2 + 2\sum_{k=t'}^{t-1}n\sigma^2
$$

结合公式（10），我们有

$$
\begin{aligned}
E_t &\le \lambda_2^2(1+\epsilon)E_{t'} \\
&\quad + 2(1+\frac{1}{\epsilon})\alpha^2\left(2(2\tau+M)L^2\sum_{k=t'}^{t-1}E_k + 2(2\tau+M)n(P+1)\mathbb{E}\sum_{k=t'}^{t-1}\|\nabla f(\bar{x}_k)\|^2 + n(2\tau+M)\sum_{k=t'}^{t-1}\zeta^2 + \sum_{k=t'}^{t-1}n\sigma^2\right) \\
&\overset{\epsilon=\frac{1-\lambda_2^2}{2}}{\le} (1-\frac{p}{2})E_{t'} \\
&\quad + \frac{6}{p}\alpha^2\left(2(2\tau+M)L^2\sum_{k=t'}^{t-1}E_k + 2(2\tau+M)n(P+1)\mathbb{E}\sum_{k=t'}^{t-1}\|\nabla f(\bar{x}_k)\|^2 + n(2\tau+M)\sum_{k=t'}^{t-1}\zeta^2 + \sum_{k=t'}^{t-1}n\sigma^2\right) \\
&\overset{\alpha\le\frac{p}{16L\sqrt{3\tau(2\tau+M)}}}{\le} (1-\frac{p}{2})E_{t'} + \frac{p}{64\tau}\sum_{k=t'}^{t-1}E_k + C\mathbb{E}\sum_{k=t'}^{t-1}\alpha^2\|\nabla f(\bar{x}_k)\|^2 + D\sum_{k=t'}^{t-1}\alpha^2,
\end{aligned}
$$

其中 $p \triangleq 1-\lambda_2^2$，第二个不等式由 $(1-2x)(1+x) \le 1-x$ 和 $p<1$，以及$C = {}^{12(}$ $^{2\tau+M)n(P+1)\,p}$，~~$D =$~~ $^{6}{}_{((2\tau+M)n\zeta^2+n\sigma^2)\,p}$得到。

□

为了界定共识误差之和，我们需要将 E_t 展开到之前的时间步。然而，引理 C.2 仅刻画了在 t 和 t'之间存在共识步骤的情况。我们需要一个类似的引理，将 E_t 展开到 $E_{t''}$，其中 $t''=\lfloor \frac{t}{\tau} \rfloor \tau$。也就是说，在 t 和t''之间不存在共识步骤。

引理 C.3。（共识误差部分 *2*）在假设 *3.2*、*3.3* 和 *3.4* 下，当 $\alpha \leq \frac{p}{16L\sqrt{3\tau(2\tau+M)}}$ 时，我们有

$$E_t \leq (1+\frac{p}{2})E_{t''} + \frac{p}{64\tau}\sum_{k=t''}^{t-1} E_k + C\mathbb{E}\sum_{k=t''}^{t-1}\alpha^2\|\nabla f(\bar{x}_k)\|^2 + D\sum_{k=t''}^{t-1}\alpha^2$$

其中 $t''=\lfloor \frac{t}{\tau} \rfloor \tau$。

证明。 与引理 C.2 的证明类似

$$X_t - \bar{x}_t\mathbf{1}_n^T = (X_{t''} - \bar{x}_{t''}\mathbf{1}_n^T) - \alpha\sum_{k=t''}^{t-1} G_k(I-J),$$

其中 $J=\frac{\mathbf{1}_n\mathbf{1}_n^T}{n}$且 $t''=\lfloor \frac{t}{\tau} \rfloor \tau$。取平方范数并求期望

$$\begin{aligned}\mathbb{E}\|X_t - \bar{x}_t\mathbf{1}_n^T\|_F^2 &= \mathbb{E}\|(X_{t''} - \bar{x}_{t''}\mathbf{1}_n^T) - \alpha\sum_{k=t''}^{t-1} G_k(I-J)\|_F^2 \\ &\leq (1+\epsilon)\mathbb{E}\|X_{t''} - \bar{x}_{t''}\mathbf{1}_n^T\|_F^2 + (1+\frac{1}{\epsilon})\alpha^2\underbrace{\mathbb{E}\|\sum_{k=t''}^{t-1} G_k(I-J)\|_F^2}_{T_3} \qquad (11)\end{aligned}$$

T_3 也可以用类似方法进行界定，唯一的区别是 $t-t''\leq\tau$,

$$T_3 \overset{t-t''\leq\tau}{\leq} 4(\tau+M)L^2\sum_{k=t''}^{t-1}E_k + 4n(\tau+M)(P+1)\sum_{k=t'}^{t-1}\|\nabla f(\bar{x}_k)\|^2 + 2n(\tau+M)\tau\zeta^2 + 2\tau n\sigma^2$$

因此，当 $\alpha \leq \frac{p}{16\sqrt{3\tau(2\tau+M)}}$ 且 $\epsilon=\frac{1-\lambda_2^2}{2}$ 时，

$$E_t \leq (1+\frac{p}{2})E_{t'} + \frac{p}{\tau}\sum_{k=t''}^{t-1} E_k + C\mathbb{E}\sum_{k=t''}^{t-1}\alpha^2\|\nabla f(\bar{x}_k)\|^2 + D\sum_{k=t''}^{t-1}\alpha^2.$$

□

结合引理 C.2 和 C.3，我们得到总体共识误差的以下界。**引理 C.4**。（总体共识误差）给定假设 *3.2*、*3.3* 和 *3.4*，我们有

$$\frac{2L^2}{n}\sum_{t=0}^{T-1}E_t \leq \frac{1}{8}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2 + \frac{128L^2}{n}\frac{DT\tau}{p}\alpha^2,$$

其中 $\alpha \leq$最小值 $\{\frac{p}{16L\sqrt{3\tau(2\tau+M)}}, \frac{1}{32L}\sqrt{\frac{pn}{2C\tau}}\}$。

证明。 将 [8] 中的引理 14 应用于引理 C.2 和 C.3，我们有

$$B\sum_{t=0}^{T-1}w_tE_t \leq \frac{1}{8}\sum_{t=0}^{T-1}w_t\mathbb{E}\|\nabla f(\bar{x}^t)\|^2 + 64BD\frac{\tau}{p}\sum_{t=0}^{T-1}w_t\alpha^2,$$

其中 $\alpha \leq \frac{1}{32}\sqrt{\frac{p}{BC\tau}}$和 $\{w_t\}_t$ 是 $\frac{p}{16\tau}$-缓慢递增的。

设为 $B=\frac{2L^2}{n}$，$w_t=1$，我们得出了结论。

□

引理 C.4 的关键观察是，我们可以用累积梯度平方范数来界定累积共识误差。现在，我们将引理 C.4 应用于引理 C.1，以得到收敛定理。

定理 C.5 （定理 3.6）。 在假设 *3.1–3.5* 下，有

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{8(f^0-f^*)}{\alpha T} + \frac{8\alpha L}{n}(\frac{\sigma^2}{2}+M\zeta^2) + \frac{1024L^2}{n}\frac{D\tau}{p}\alpha^2,$$

其中， $\alpha \leq \min\{\frac{1}{L}, \frac{n}{4LM(P+1)}, \frac{n}{LM}, \frac{p}{16L\sqrt{3\tau(2\tau+M)}}, \frac{1}{32L}\sqrt{\frac{pn}{2C\tau}}\}$ 以及 C、D 定义于引理 *C.2*.

证明。 对引理 C.1 取全期望，并对 t 求和，得

$$\frac{\alpha}{4}\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{f^0-f^*}{T} + \frac{2\alpha L^2}{n}\frac{\sum_{t=0}^{T-1}E_t}{T} + \frac{\alpha^2 L}{n}(\frac{\sigma^2}{2}+M\zeta^2)$$

根据引理 C.4，有

$$\frac{\alpha}{4}\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{f^0-f^*}{T} + \frac{\alpha}{8}\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} + \frac{128L^2}{n}\frac{D\tau}{p}\alpha^3 + \frac{\alpha^2 L}{n}(\frac{\sigma^2}{2}+M\zeta^2).$$

整理各项后，得证。 □

推论 C.6 （推论 3.7）。 在定理 *3.6* 的相同假设下，令 $\alpha = \sqrt{\frac{n}{T}}$，则有

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} \leq \frac{8(f^0-f^*)}{\sqrt{nT}} + \frac{8L(\frac{\sigma^2}{2}+M\zeta^2)}{\sqrt{nT}} + \frac{6144nL^2\tau((2\tau+M))\zeta^2+\sigma^2)}{p^2T},$$

给定

$$T \geq \max\{nL^2, \frac{16L^2M^2(P+1)^2}{n}, \frac{L^2M^2}{n}, \frac{768nL^2\tau(2\tau+M)}{p^2}, \frac{24576nL^2\tau(2\tau+M)(P+1)}{p^2}\}.$$

证明。 将 $\alpha = \sqrt{\frac{n}{T}}$ 代入定理 3.6，可得

$$\begin{aligned}\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}{T} &\leq \frac{8(f^0-f^*)}{\sqrt{nT}} + \frac{8L(\frac{\sigma^2}{2}+M\zeta^2)}{\sqrt{nT}} + \frac{6144nL^2\tau((2\tau+M))\zeta^2+\sigma^2)}{p^2T} \\ &= \mathcal{O}(\frac{1}{\sqrt{nT}} + \frac{n}{T}).\end{aligned}$$

为满足步长规则，我们需要

$$\sqrt{\frac{n}{T}} \leq \min\{\frac{1}{L}, \frac{n}{4LM(P+1)}, \frac{n}{LM}, \frac{p}{16L\sqrt{3\tau(2\tau+M)}}, \frac{1}{32L}\sqrt{\frac{pn}{2C\tau}}\},$$

由此可得

$$T \geq \max\{nL^2, \frac{16L^2M^2(P+1)^2}{n}, \frac{L^2M^2}{n}, \frac{768nL^2\tau(2\tau+M)}{p^2}, \frac{24576nL^2\tau(2\tau+M)(P+1)}{p^2}\}.$$

□

D 实验补充说明

由于篇幅限制，正文仅展示了有限的实验结果。本节将提供全面的分析，以补充这些发现。

D.1 计算资源

实验在一台配备 8 块 NVIDIA 3090 GPU 的服务器上进行，每块 GPU 的显存为 24GB。整个实验耗时约 1500 多 GPU 小时。大量计算主要来自各种本地步数 τ，我们考察了 8 种可能的取值，并在同构和异质情况下分别进行了 VGG11 和 ResNet18 训练。

D.2 逐迭代收敛比较

为扩展第 4.2 节中的模拟实际运行时间分析，我们考察了重叠式局部去中心化随机梯度下降（OLDSGD）与各基线方法（包括局部去中心化随机梯度下降（LDSGD）、KGT、LUGT、本地随机梯度下降（LSGD）和 Local Exact Diffusion（LED））的逐次迭代收敛情况。图 6、7 和 8 分别展示了在同构和异质数据分布下，VGG11、ResNet18 和 GPT2 微调在 CIFAR-10 及语言数据集上的训练损失、测试准确率和测试困惑度。如前所述，由于计算限制，GPT2 微调仅在同构设置下进行。为便于说明，我们重点关注本地步数为 $\tau \in \{1, 3, 5, 10\}$ 时的结果，尽管采用更大的 τ 进行的实验也呈现出一致的趋势。分析证实了 OLDSGD 在理论上与 LDSGD 等价，凸显了基于梯度下降的方法的稳定性，表明随着本地步数增加，加速比显著提升，并揭示了 OLDSGD 在测试性能方面的优势，尤其是在 GPT2 上的表现，彰显了其应用于现代基于 Transformer 的模型的潜力。

在 VGG11 和 ResNet18 训练中，OLDSGD 的逐迭代收敛表现与 LDSGD 高度一致，尤其是在本地步数较少时。这印证了定理 3.6：由于 OLDSGD 采用类似 SGD 的平均更新，因此在非凸场景下两者具有相同的收敛速率。在 $\tau = 1$，同构设置下，经过数百次迭代后，两种算法的训练损失几乎相同，差异极小。对于 $\tau \in \{5, 10\}$，损失也保持相近；在异质设置下，OLDSGD 的表现紧跟 LDSGD，但由于陈旧误差而略有落后。随着训练推进、梯度逐渐减小，这种陈旧性影响会逐渐消退，使 OLDSGD 达到与 LDSGD 同等精度的最终损失，这与其理论收敛速率的预测一致。测试准确率也呈现类似趋势：在同构设置下，OLDSGD 的表现与 LDSGD 相当；而在异质设置下，OLDSGD 有时甚至优于 LDSGD，尤其是在 ResNet18 训练的 $\tau = 1$，OLDSGD 的准确率更高。

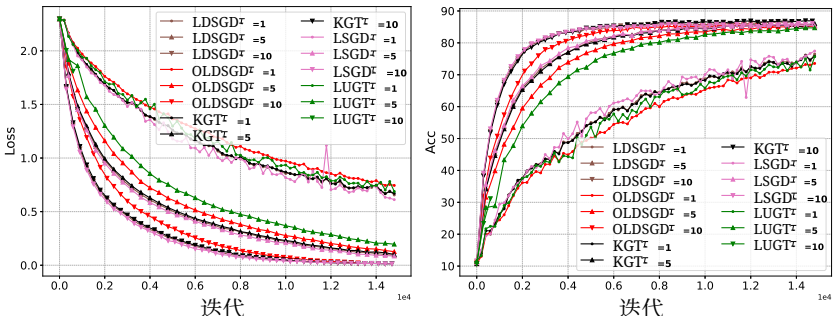
在 GPT2 微调中，OLDSGD 在测试困惑度方面始终优于所有基线方法，尽管在理论上与 LDSGD 等价，仍展现出实际优势。在 $\tau = 3$，OLDSGD 的困惑度低于 LDSGD 和 KGT，且随着 τ 增大，差距进一步扩大。在 $\tau = 5$ 和 $\tau = 10$，如图8所示，OLDSGD 的困惑度曲线下降得更快，在所有迭代过程中始终明显领先于 LDSGD、KGT 和 LSGD。这种在所有测试的 τ 中都能观察到的稳定优势，凸显了 OLDSGD 在基于 Transformer 的架构中的效能；在这类架构中，通信瓶颈至关重要。

与基于梯度跟踪的方法或精确扩散方法相比，基于梯度下降的方法（包括 OLDSGD、LDSGD 和 LSGD）在不同本地步数下表现出显著的稳定性。对于 VGG11 和 ResNet18，这些方法在 $\tau = 10$ 均能保持稳定的损失下降和准确率提升，即使 τ 较大时也未观察到发散。相比之下，KGT 在异质条件下收敛缓慢，损失明显更高；如图6所示，在异质 VGG11 训练中，LUGT 则在 $\tau = 10$ 完全发散。对于两个任务，LED 同样在 $\tau \geq 10$ 失效，这凸显了基于跟踪的方法 [4] 的局限性，尽管它们具有诸如在异质条件下精确收敛等理论保证。

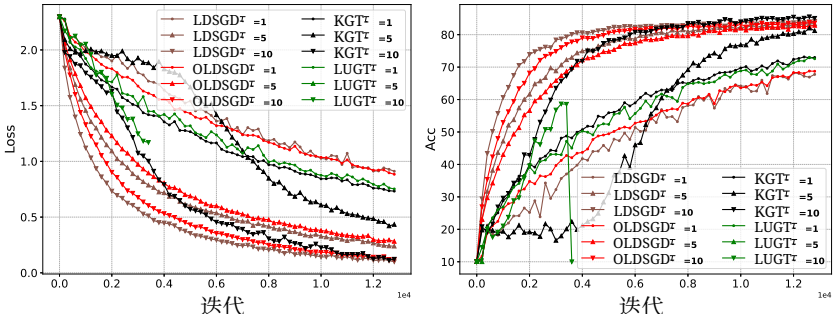
随着本地步数增加，OLDSGD 的加速比也随之提升，这是其一项关键优势，并且在所有任务中均有所体现。在 VGG11 和 ResNet18 训练中，随着 τ 增加，OLDSGD 每次迭代的损失下降得更快，在同质和异质设置下均是如此。对于 GPT2，这种加速效果更加显著：与较小的 τ 相比，在 $\tau = 10$ 时困惑度下降明显更快。针对更大的 τ（例如 $\tau = 20$）进行的实验也证实了这一趋势，表明 OLDSGD 能够应对远超计算时间的通信延迟，因此适用于网络延迟较高的实际应用。

这些发现表明，即使在高度异质且存在通信延迟的环境中，OLDSGD 仍是去中心化训练的一种稳健且可扩展的解决方案。其收敛表现与 LDSGD 高度吻合，验证了理论预测；而它在 VGG11 和 ResNet18 测试准确率方面偶尔更胜一筹，并且在 GPT2 困惑度方面始终占优，这些都凸显了其实际优势。基于梯度下降的方法具有稳定性，再加上 OLDSGD 能够掩蔽通信延迟，使其特别适用于现代基于 Transformer 的模型和联邦学习场景。

[4] 基于跟踪的方法　还需要更多内存和通信预算。　　预算。

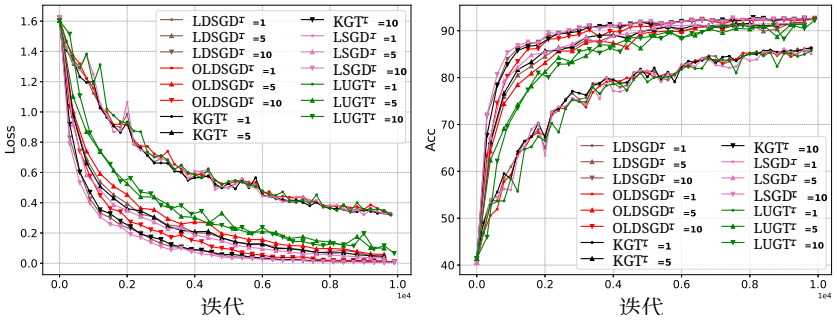


(a) 训练损失 - 同质。 (b) 测试准确率 - 同质。

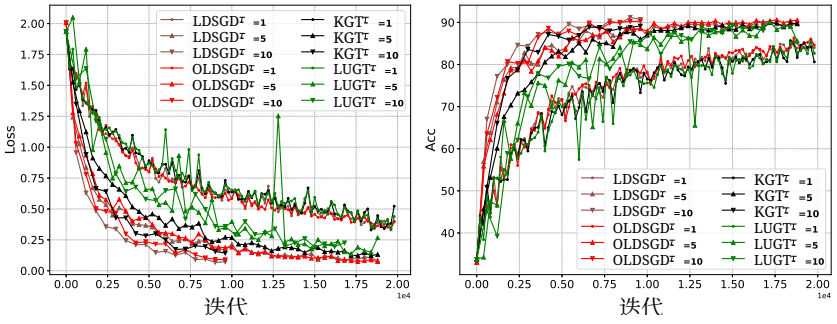


(c) 训练损失 - 异质。 (d) 测试准确率 - 异质。

图6：VGG11 训练的逐次迭代收敛。展示了不同数量的本地步数。

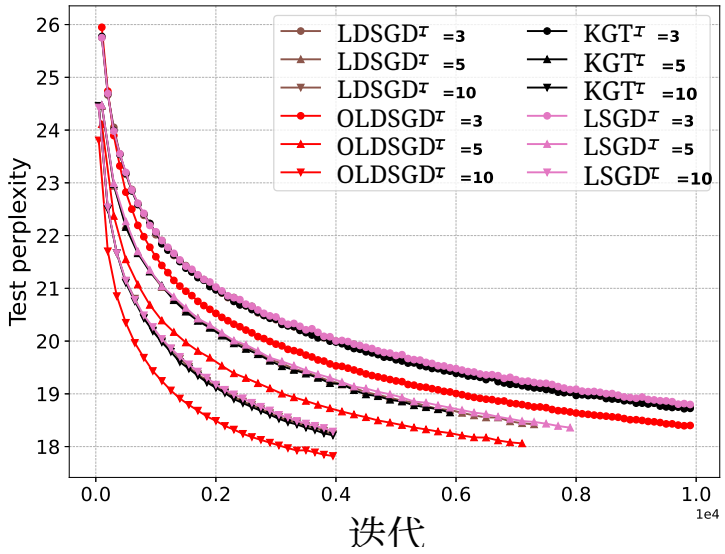


（a）训练损失 - 同质。 （b）测试准确率 - 同质。



（c）训练损失 - 异质。 （d）测试准确率 - 异质。

图7：ResNet18 训练的逐次迭代收敛。展示了不同数量的本地步数。



Fig图8：GPT2 微调训练相对于迭代次数的测试困惑度曲线。OLDSGD 始终 y achi取得比其他方法更低的困惑度。

D.3 异质情况下的加速比

在正文中，我们通过表3展示了 OLDSGD 在同质设置下的加速比。如上一小节所示，在数据异质性条件下，OLDSGD 往往表现得更好（相比其他方法）。下面的表4量化了这一趋势。最后一列表示同构情况下的加速比几何平均值，仅根据 VGG 和 ResNet 训练计算得出，以便进行比较。在异质数据条件下，OLDSGD 平均比 LDSGD 实现了 1.64× 的加速比，超过了同质数据下观察到的 1.39× 加速比。在数据异质性条件下，OLDSGD 相对于 KGT 和 LUGT 的加速比也以类似方式提升。这种在数据异质性条件下增强的性能凸显了 OLDSGD 对数据分布变化的鲁棒性。该特性有利于实际部署，尤其是在由于隐私问题等限制而难以采用同质数据的场景中。

表4：OLDSGD 相较于现有方法的加速比（数值越高越好）。最后一列给出了同构情况下加速比的几何平均值（不包括 GPT2 微调），以供比较。

	VGG11		ResNet18			
算法	$c=1$	$c=5$	$c=1$	$c=5$	**几何平均值**	**同质情况几何平均值**
LDSGD	1.41×	1.56×	1.54×	1.71×	1.64×	1.39×
KGT	1.34×	2.25×	2.63×	2.25×	1.63×	1.42×
LUGT	2.14×	6.20×	3.41×	7.43×	4.28×	3.05×