联邦拆分学习中的通信隐私与效率

Zongshun Zhang‡ Andrea Pinto⋆ Valeria Turina⋆ Flavio Esposito⋆ Ibrahim Matta ‡

‡计算机科学系
波士顿大学

⋆计算机科学系
圣路易斯大学

技术报告

arXiv:2301.01824v2 [cs.LG] 7 Jan 2023

摘要—每天，海量敏感数据分布在手机、可穿戴设备和其他传感器上。传统上，这些庞大的数据集都由单一系统处理，并通过训练复杂模型来做出有价值的预测。近年来，为了更好地保护用户数据和隐私，同时确保高性能，人们开发了联邦学习和拆分学习等分布式机器学习技术。这两种分布式学习架构各有优缺点。本文将探讨这些权衡，并提出一种新的混合联邦拆分学习架构，兼具两者在效率和隐私方面的优势。评估结果表明，我们的混合联邦拆分学习方法能够降低运行分布式学习系统的每个客户端所需的处理能力，缩短训练和推理时间，同时保持相近的准确率。我们还讨论了该方法抵御深度学习隐私推理攻击的能力，并将其与近期提出的其他基准进行比较。

1. 引言

集中式机器学习（ML）训练正变得难以为继 [1]。除了频繁重新训练有助于优化收入 [2], 之外，出于多种原因，一些学习应用需要在网络边缘端运行，而非在数据中心核心运行；这些原因包括在终端设备上本地运行机器学习算法以最大限度地降低端到端延迟，以及对信任第三方云所带来的隐私问题的顾虑 [3]。一些机器学习（ML）模型以利用敏感数据为代价，改善移动设备上的用户体验；例如，键盘中的文本推荐 [4], [5] 或语音助手 [6]。在这些及其他应用中，去中心化学习方法可能比集中式系统更可取，因为敏感数据可以保留在客户端本地，而不通过计算机网络传输。

尽管在若干使用场景中具有诸多优势，但在本地设备上运行机器学习训练和推理任务仍存在一些局限：计算能力通常有限，密集处理会加速电池耗电，而且移动设备或其他终端设备的内存和存储能力有限。例如，我们的实验表明，要对一个在 ImageNet [8] 上预训练的 VGG-16 [7] 神经网络使用 Cifar-10 [9], 进行微调，在 NVDIA V100 GPU 上需要几十分钟才能达到 90% 准确率。

为保护隐私并确保及时收敛，研究人员提出了不同的分布式神经网络架构，例如联邦学习 [1], 拆分学习 [10], 或混合方法 [11], [12], [13]。联邦学习 (FL) [14] 会对每个边缘设备上已训练神经网络模型的权重求平均，生成一个统一模型，并用它更新各设备的本地模型。此前的研究表明，与仅采用本地模型相比，这种策略能够实现更高的准确率 [14], [15], [16], [17] ，同时保护数据隐私。拆分学习 (SL) 架构将整个神经网络划分为多个层分区。每个分区由不同实体（即边缘端和云）执行，不同边缘设备上的神经网络分区可以与云端分区配对。因此，这种方法能够利用分布式数据集，同时保护用户数据隐私。

一方面，只要有足够的资源满足训练服务级别协议 (SLA)，联邦学习就很容易扩展到大量设备。因此，在资源有限的边缘训练/推理场景中，联邦学习并不实用。另一方面，拆分学习可以在资源有限的情况下进行训练，但由于其无法并行，因此难以很好地扩展到大量设备。尤其是当我们将使用非独立同分布 (Non-IID) 数据的不同边缘设备与中心云配对时，训练可能根本无法收敛 [18]。拆分学习的另一个缺点是，在客户端或服务器节点中传输和存储中间数据的成本可能很高 [19], [20], [21], [22]。此外，由于前向传播中的中间数据源自源数据，因此存在隐私方面的顾虑。我们将在第 2 节进一步讨论这两种模型。

为了解决现有分布式学习模型效率低下的问题，我们提出了一种新型分布式学习架构——联邦拆分学习（FSL）[23], 。该架构结合了联邦学习和分割学习的优点，同时减轻了它们各自的缺陷。我们将探讨 FSL 的通用性，以及一种同时优化时延和隐私保障的新方法。

FSL 模型的特点是包含多个边缘端客户端—服务器对。这些计算对同时训练各自的神经网络副本，从而实现联邦学习的并行性；而客户端与服务器的分离则带来了拆分学习的优势。每个计算对都会对神经网络进行分区。在部分计算对完成若干训练轮次后，服务器神经网络权重会像经典联邦学习算法那样，在中心云服务器中进行平均 [14]。我们将图 1d 中这个中心云节点称为“参数服务器节点”。

还有其他与我们提出的方法相近的技术，例如并行拆分学习（PSL） [13], 和联邦重建（FRC） [24],。但与 FSL 相比，它们存在一些缺点。PSL 架构只有一个云服务器节点，而 FSL 支持并行服务器计算。FSL 和 FRC 的隐私级别相近（第 4.1 节），但我们的评估表明，在不存在显著传输延迟的情况下，FRC 的训练时间是 FSL 的两倍。

我们通过在不同的神经网络模型和任务上进行测试，评估 FSL 架构的优势，任务范围涵盖图像分类到互联网流量分类 [25]。除了测试所提出的混合联邦分割架构的训练性能外，我们还评估了 FSL 和分割学习在隐私与性能方面的权衡，并提出了如何利用基于客户端的隐私方法（CPA）和新颖的神经网络划分方法来增强这些方案的隐私保障。此外，我们发现，某些神经网络划分方式能够同时降低传输延迟并增强隐私。实验结果表明，通过结合不同的隐私保护方法和神经网络划分方法，我们的 FSL 能够在训练时间、隐私保障和准确率方面实现较高的效率。

本文其余部分安排如下：第2节介绍背景和相关工作，并讨论 FSL。随后，第3节描述我们提出的 FSL 系统。第4节讨论基于客户端的隐私方法（CPA），以及在边缘端划分神经网络的动机；该方法可应用于任何基于拆分学习的架构。第5.2节介绍我们采用训练和推理指标进行的评估，第5.3节则介绍隐私评估。

2. 分布式学习架构：背景与相关工作

联邦学习 [1], [15], [16], [17], [26] 是一种去中心化的机器学习技术，使用多个客户端“拥有”的数据源训练神经网络模型（图 1a）。一个逻辑上集中的参数服务器保存最新的神经网络模型，并协调在所有客户端之间共享其权重。

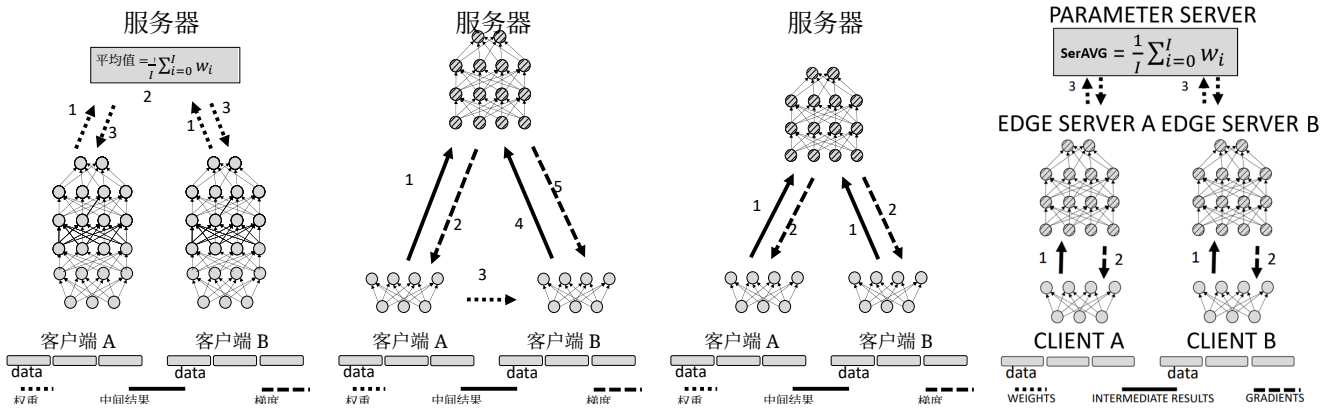
在联邦学习过程的训练阶段开始时，参数服务器会向每个客户端发送一组相同的随机初始化神经网络权重。随后，每个客户端使用自己的本地数据集训练本地模型多个训练轮次。直到客户端模型提取出足够的特征，也就是说，达到给定的准确率阈值后，参数服务器会持续获取、平均并覆盖权重（图 1a – 步骤 1 至 3 ）。因此，全局模型可以利用私有数据集，而这些数据无需通过网络传输。此外，联邦学习可以设置参数，以指定参数服务器对特定客户端组的权重进行平均的频率。这样，系统架构师便能在网络流量和模型准确率之间取得平衡。因此，只要客户端具备足够的计算能力和存储资源以满足训练约束或服务级别目标（ SLO），联邦学习就被认为能够扩展到大量客户端。

拆分学习 [10] 是一种分布式机器学习技术，其特点是将神经网络模型在计算上拆分为两个分区。每个分区都可以在独立的计算节点上运行，从而分担计算资源需求。要使用拆分学习训练神经网络模型，首先必须将神经网络划分到不同节点上。随后开始前向传播阶段。当到达第一个节点上的神经网络分区末端时，最后一层激活函数的输出会发送至服务器节点——步骤 1（图 1b）。然后，这些输出被用作第二个神经网络分区的输入，并继续进行前向传播。计算出损失后，服务器节点开始反向传播。当反向传播到达服务器分区的输入层时，输入的梯度会发送至客户端的最后一个激活函数，以完成客户端神经网络的反向传播——步骤 2。最后，使用梯度更新完整神经网络的权重。不过，服务器的神经网络分区也可以与具有相同客户端神经网络结构的其他客户端的神经网络分区配对。首先，将上一个客户端训练得到的权重转移到新客户端——步骤 3。随后，新的服务器与客户端配对按照上述方式进行训练——步骤 4 和 5。

拆分学习算法能够保护数据隐私，但在数据源呈非独立同分布（non-IID）时收敛时间较长（图 2），且需要传输大量中间数据。此外，多个客户端的训练是顺序进行的，因此可扩展性较差。一些解决方案试图降低此类中间数据的传输延迟。例如，在 BottleNet [20]和 BBNet [19], 中，作者旨在通过特定的神经网络设计压缩中间数据。而 Early-Exit [21] 的思路是在较早的层中添加分类器，以避免计算完整模型。先前的工作 [22] 也采用了知识蒸馏，以降低客户端模型的复杂度和需要传输的数据量。

本文考虑混合拆分联邦学习系统，以兼顾两者的优势并尽量减少其缺点。例如，我们表明，与拆分学习相比，混合联邦拆分学习和并行拆分学习在数据源呈非独立同分布时能够更快收敛。

结合拆分学习与联邦学习。其他研究者提出了结合拆分学习与联邦学习优势的方法。在并行拆分学习（PSL） [13]（图 1c）中，他们在多个



(a) 联邦学习 (b) 拆分学习 (c) 并行拆分学习 (d) 联邦拆分学习

图 1：分布式神经网络训练架构：*(a)* 联邦学习：神经网络位于客户端。参数服务器计算各客户端权重的平均值，并覆盖本地权重。*(b)* 拆分学习：服务器分区依次与各客户端进行训练。客户端权重会共享给下一个参与训练的客户端。*(c)* 并行拆分学习：服务器以批次并行训练客户端的输出，但客户端的权重会保密。*(d)* 联邦拆分学习：多组边缘服务器与客户端同时进行训练。参数服务器对边缘服务器的权重取平均值。客户端的权重会保密。

边缘节点上并行训练客户端神经网络分区。在前向传播阶段，不同客户端的激活函数结果会被发送到同一台远程服务器。随后，该服务器将损失函数的梯度反向传播给各客户端。

特定模型的方法。 FedSL [11]是循环神经网络（RNN）的一种实例。其思路是展开 RNN 的反馈回路，并将循环神经网络分区与序列数据片段一起拆分到不同节点上。每个训练轮次结束后，各设备会对权重求平均并覆盖原有权重。这种方法将分割学习和联邦学习的优势结合起来，适用于 RNN 训练场景，且不会带来太多额外开销。

联邦学习的扩展。 有些研究扩展了联邦学习模型，而不是设计混合架构。联邦构建（FRC） [24], 的作者优先考虑用户数据隐私，牺牲了训练过程的效率。他们将模型划分为全局 和本地 分片，并将两者部署在每台边缘设备上，两个分区交替进行训练。随后，参数服务器发送全局分片，并取回相应的更新后权重。这种设计使训练无状态，因此在存储方面具有很强的可扩展性。

源数据隐私。 这些边缘训练系统的优势之一，是源数据隐私的感知保护水平较高：用户数据不会离开边缘设备。然而，攻击者也可以通过神经网络权重推断源数据。除 FRC 和 PSL 外，文中提到的所有系统都无法保护神经网络权重的隐私 [27], [28]。它们之所以能保护这类隐私，是因为不会共享完整神经网络的权重或激活结果。

NoPeek [29] 中讨论的另一种威胁针对激活函数的结果，该结果可被用于通过自编码器神经网络重构源数据。某些将神经网络进行分区并在分区之间传输中间数据的边缘智能系统

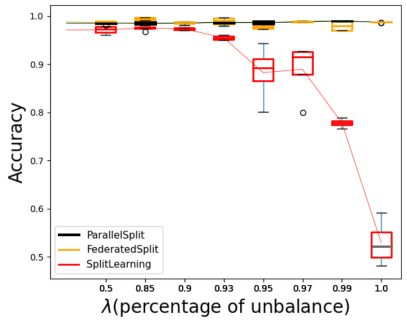


图2：使用非独立同分布数据进行训练时的准确率下降。

容易受到这种威胁的影响。一种可能的缓解方法是在原始损失函数中加入距离相关（DC）损失 [29] ，以衡量源数据与中间数据之间的差异。该方法通过求解多目标优化问题，在更新神经网络权重的同时最大化 DC 损失和准确率。

基于客户端的隐私方法（CPA）。NoPeek 的损失函数提高了从中间数据重建私有源数据的韧性。然而，共享损失并构建分布式梯度图要么不安全，要么需要额外的管理系统。为克服这一局限性，我们提出了 NoPeek 的扩展方法，即基于客户端的隐私方法。其思路是使用用户指定的方法向客户端激活函数的结果中添加噪声，从而使自编码器神经网络可使用的特征更少（第4.3节）。

隐私保护方法。 我们将主要比较 CPA 框架下的直流与差分隐私（差分隐私随机梯度下降）。差分隐私随机梯度下降是一种广泛使用的轻量级数据隐私算法方法 [30], [31]。其思路是在神经网络训练阶段向梯度中添加高斯噪声

。这样一来，由更新后的权重生成的客户端输出就会使对手难以判断。与同态加密或安全多方计算相比，我们认为这种方法更适合客户端和边缘服务器的设置，而加密的隐私级别已得到充分讨论[31]。在边缘端，为加密和解密步骤消耗大量电池电量并不划算。还有一些方法，例如模型压缩（ [19], [22], [32]），可能会增强隐私保障，且功耗低于同态加密，但高于差分隐私随机梯度下降。不过，比较现有隐私保护方法并非本项目的重点，我们将其作为未来工作。我们主要想表明，所提出的 CPA 是一种通用方法，采用包括直流和差分隐私随机梯度下降在内的不同方法时，均具有较强的攻击韧性。因此，本文评估了应用于完整模型的传统差分隐私随机梯度下降，并将其与通过 CPA 框架仅应用于客户端的差分隐私随机梯度下降进行比较（第5节 5.3.6）。

3. 隐私无关分割学习

本节详细介绍我们在 [23] 中首次提出的联邦拆分学习（FSL）架构。FSL 是一种混合方法，结合了 SL 和 FL 的优点。它在具备可扩展性的同时，避免发送用户的源数据或通过网络共享完整神经网络的参数。

图 1d 所示的 FSL 架构包含三类实体：(i) 边缘服务器、(ii) 客户端和(iii) 参数服务器。要使用 FSL 训练神经网络，我们首先在各实体之间设置身份验证协议 [33], [34] ，将每个客户端与一台边缘服务器配对，并将边缘服务器与一台参数服务器配对。配对完成后，通信不经加密直接进行。然后，在每一对客户端和服务器之间，将完整神经网络划分为客户端分区和（边缘）服务器分区。

FSL 的训练分为三个步骤。步骤 1 中，客户端使用源数据进行前向传播，并将中间数据传输至边缘服务器。随后，服务器完成传播并计算损失。步骤 2 中，服务器对客户端源数据进行反向传播。步骤 3 中，经过若干训练轮次后，参数服务器对边缘服务器中的权重取平均。

因此，与讨论过的其他方案相比，FSL 具有多项优势。与联邦学习相比，FSL 客户端的资源需求更低，因为需要训练的神经网络层更少。因此，FSL 是一种更适用于边缘智能应用的方案。此外，FSL 只对边缘服务器中的权重进行平均，而 FedAVG 对全部权重进行平均。因此，FSL 避免了遭受模型反演攻击的潜在风险 [27], [28]。此外，与分割学习相比，FSL 的可扩展性更好，因为客户端与服务器组成的配对可以独立训练。与 FRC [24]架构相比，我们注意到后者的训练时间效率较低。它在每次执行前向传播和反向传播时都会更新一个分区的参数，并且需要多次运行才能更新完整模型。与并行拆分学习 [13], 相比，我们发现了四个潜在的次优之处。首先，由于边缘设备必须与中央服务器同步，客户端可能需要等待服务器处理完队列中所有激活函数的结果。在最坏情况下，假设来自 n 个客户端的所有激活函数结果同时到达，那么每个客户端继续进行反向传播所需等待时间的下界为 $O(n)$。因此，我们得出结论：PSL 的可扩展性不如 FSL。其次，我们还发现，PSL 服务器会临时存储多个批次的激活函数结果，因此需要复杂的日志记录与压缩存储系统来从故障中恢复。因此，PSL 的鲁棒性不如 FSL，因为 FSL 在每个相互隔离的配对中需要维护的状态更少。第三，PSL 的设计可能存在弹性问题。与此同时，在 FSL 中，一个配对发生故障不会妨碍其他配对进行训练或推理。第四，在 PSL 中，由于所有中间数据都将由单台服务器传输和处理，服务器节点的带宽和计算资源可能会变得拥塞。我们的研究结果见第5节。

4. 隐私感知联邦分割学习

我们已经讨论了 FSL 的效率和容错属性。本节将转而探讨不同架构的隐私保护属性，并提出我们的隐私感知联邦分割学习。具体而言，我们将讨论如何改进通用的拆分学习架构，以缓解在诚实但好奇网络上传输神经网络激活函数的输出值或权重时产生的问题。我们首先正式定义隐私攻击者模型，然后讨论基于客户端的隐私保护方法以及某些神经网络划分方式如何有助于避免此类攻击。

4.1. 隐私攻击者模型与假设

我们假设攻击者能够以明文形式获取中间数据 （即从客户端传输到边缘服务器的最后一层激活函数的结果）。此外，我们假设攻击者知道客户端神经网络架构。因此，对手可以实现一个自编码器 [35] 神经网络，以重现输入客户端模型的源数据。此外，为了训练自编码器神经网络，我们假设对手能够获取一些特征与客户端源数据相似的数据集。

4.2. 攻击韧性

根据攻击者模型，为了比较不同隐私保护方法所提供的隐私保障水平，我们将攻击韧性 指标（τ）定义为:

$$\tau = 1 - \frac{\|correct\|}{\|reconstructed\|} \tag{1}$$

它衡量误分类率。 $\|correct\|$ 表示攻击者重建出的、能够被训练好的分类器正确分类的图像数量（第5.3.3节）。 $\|reconstructed\|$ 则表示重建图像的总数。

4.3. 分布式环境下基于距离相关性的客户端隐私方法（CPA-DC）

受 NoPeek 方法 [29], 的启发，该方法根据两个损失函数之和更新权重，即交叉熵和距离相关性（DC），我们采用了将两个损失函数用于交替调度机制的思路，该机制包含两个轮次：常规轮次，在（边缘端）服务器中最小化交叉熵损失函数；距离相关性轮次，在客户端最大化 DC 损失函数。其思路是加入噪声，使源数据和中间数据与客户端的权重不同。我们在公式2中定义了这种使用损失函数的交替行为：

$$L = \begin{cases} Loss(g(f(x)), label) & if \ \ e \bmod F == 0 \\ m \cdot DC(x, f(x)) & \text{otherwise,} \end{cases} \tag{2}$$

其中， L 是测得的损失值， e 表示轮次索引， F 表示 DC $Frequency$， $Loss$（→ 是用于衡量误分类的损失函数， DC 是距离相关性损失， m 是损失乘数， f 是客户端神经网络， g 是服务器神经网络， x 是源数据，$label$ 是源数据中的标签。请注意， F 控制交替频率，而 m 为 DC 损失增加权重。

交替损失函数是一种策略。我们的基于客户端的隐私方法（CPA）也可以与其他损失函数或在常规轮次中添加随机噪声的方法配合使用。在评估中，我们探索了几种嵌入噪声的方法。具体而言，我们探讨了训练时间与最高攻击韧性之间的权衡。

4.4. 我们为每个神经网络分区分配多少层?

本节讨论如何确定每个神经网络分区应分配多少层，即客户端神经网络深度。此权衡将影响训练时间（处理延迟和传输延迟）、隐私和准确率。要同时权衡所有这些指标颇具挑战。以处理延迟和传输延迟之间的权衡为例。不同分区中输出层的大小差异可能很大，因此，少数几种划分策略就可能导致显著的传输开销，从而增加训练时间，进而削弱混合式联邦分割学习相较于原始联邦学习架构的优势。在 VGG-16 [7], 中，第一卷积层的输出大小是第二卷积层的两倍。因此，在第二卷积层之后设置模型切分点的系统，可以用低算力客户端上增加的处理延迟换取更低的传输延迟。我们将在第5.2.2节评估这一影响。

从分析上看，这一影响可通过求解问题 3 来体现，其中 α 和 β 是开发者指定的参数，分别表示中间数据传输延迟（I）和计算延迟（C）的正权重，参数 d 表示客户端神经网络的深度，最后， b 表示带宽，该带宽会定期测量。为高效地解决这一问题，我们采用 [36], 中的方法：通过对模型进行性能分析，即计算客户端神经网络中每一层的输出大小和处理时间，而不是训练完整模型，从而构建两个回归模型，根据可用带宽预测时延 I 和 C。

$$\min_{d}(\alpha I(d \mid b) + \beta C(d)) \tag{3}$$

问题3的解在时延方面是最优的，但在隐私和准确率方面可能并非最优。如第5.2.2节所示，当客户端神经网络深度介于7至16层之间时，FSL的客户端处理和传输延迟达到最小值。而在图7a中，客户端神经网络需要超过16层，才能达到90%以上的攻击韧性。因此，我们得出结论：最优的神经网络划分决策应权衡不同的目标和约束，包括传输延迟、处理时间、隐私和准确率，如问题形式化4所示。在此类问题中， W 表示模型权重向量，$(I',\ C',\ A,\ R)$ 是表示传输延迟、计算延迟、准确率和韧性观测值的元组，$(\gamma,\ \kappa)$ 是用户指定的新正权重，而 d 是客户端神经网络深度。

$$\begin{aligned} \max_{W,d} \quad & (-\alpha I'(W, d \mid b) - \beta C'(W, d) \\ & + \gamma A(W, d) + \kappa R(W, d)) \end{aligned} \tag{4}$$

为了解决问题 4，我们必须针对每个 d 训练 W，直至收敛，然后找出最佳的 d。这种蛮力法效率低下。更高效的方法是，无需完整训练模型即可预测时延、准确率和隐私。要将 [36] 中的方法扩展到时延性能分析以外的领域颇具挑战。这是因为，对于每一对客户端和边缘服务器，其准确率和攻击韧性更难进行性能分析和预测。具体而言，相关性能分析取决于在其他配对上训练得到的权重、客户端之间源数据的分布、客户端数量、SerAVG 中进行平均的层数，以及训练轮数（第5.2.4节和 5.3）。另一项工作可以预测模型准确率 [37],，但其方法基于已训练好的模型。因此，对于采用 SerAVG 的联邦拆分学习架构，如何预测分区仍是有待未来研究解决的开放问题。本文通过实验展示了在训练时间、准确率和隐私方面取得平衡的最佳模型划分方案。

5. 评估结果

本节介绍我们对隐私无关分割学习（POFSL）和隐私感知分割学习（PAFSL）架构的评估结果，以及我们采用的隐私感知方法（第4.3节中的 CPA 和第4.4节中的神经网络划分）。评估结果表明，与 PSL 和 FRC 相比，FSL 在训练时间、内存使用量和收敛速度方面具有优势。此外，我们还展示了隐私感知方法能够防止

基于拆分学习的系统利用中间数据重建源图像。我们首先讨论实验设置，然后在第5.2节和第5.3节中展示对 POFSL 和 PAFSL 的评估结果。

5.1. 实验设置

本组实验研究了不同硬件和应用场景下，采用不同神经网络和数据集时 POFSL 的收敛情况以及 PAFSL 的隐私保障。

在硬件方面，我们在 Chameleon Cloud [38]上使用了两种类型的节点。一种配备 RTX6000 GPU、两颗 Intel Xeon Gold 6126 CPU 和 187 GB 内存；另一种配备四块 NVIDIA V100 GPU、两颗 Intel Xeon Gold 6230 CPU 和 128 GB 内存。我们通过物理机上的 localhost 接口模拟分布式学习实体之间的计算机网络，并将每项实验设置为仅使用一块 GPU。因此，我们可以忽略网络带宽瓶颈。

在应用方面，我们考虑了三个分类任务，并使用 PyTorch [39]实现。随后，系统各实体之间的分布式通信由 PySyft [40]和 PyGrid [41] 处理，传输过程中未应用加密。第一个应用使用 VGG-16 [7] 卷积神经网络（CNN）执行通用图像分类任务。该模型先使用 ImageNet [8]进行预训练，再使用 CIFAR-10 数据集 [9]进行训练。我们在 5 个配备 NVIDIA V100 GPU 的客户端上运行此任务。第二个任务使用 LENET [42] CNN，在 20 个配备 NVIDIA RTX6000 GPU 的客户端上识别 MNIST 数据集 [43]中的手写数字。第三个任务对流量而非图像进行分类。具体而言，我们对一个使用 ISCX VPN-nonVPN（ISCX）流量数据集 [44], 训练的一维卷积神经网络进行了拆分，并在 5 个配备 RTX6000 GPU 的客户端上运行。我们按照独立同分布（IID）概率对数据集进行分区并分配给不同客户端。除非另有说明，所有图表均显示 95%置信区间。我们的目标是验证 FSL 在不同任务、不同神经网络、不同设备和不同数据分布下是否始终能够收敛。我们还验证了与现有方案相比，FSL 在时延或隐私方面的优势。

5.2 隐私无关分割学习的评估结果

本节介绍方法并总结实验观察结果。总体而言，我们的评估表明，与现有方案相比，POFSL 的开销更低，准确率相近。具体来说，我们评估了训练时间（第 5.2.2 节）、内存消耗（第 5.2.3 节）和学习器准确率（第5.2.4节）。

5.2.1. 实验设计。考虑到不同数据集的规模和客户端数量，为了达到至少 90% 的准确率，用于图像分类的神经网络需要训练 20 个训练轮次。流量分类模型则训练了 80 个训练轮次。

5.2.2. 训练时间评估。为了评估训练时间，我们来看一下结果见图3和图4的实验。x 轴表示切分索引，即客户端/本地部分的神经网络中运行的最后一层的索引。在 MNIST 场景下进行测试时，从图3a和图3b可以观察到，在所有其他分布式架构中，FSL 的“客户端前向传播和反向传播”（客户端前向与反向传播时间）最短。客户端前向与反向传播时间包括梯度的传输时间，以及客户端神经网络中激活值和梯度的计算时间。服务器前向与反向传播时间包括隐藏变量从客户端传输到服务器的传输时间，以及服务器神经网络中的计算时间。请注意，权重更新时间分别计入“客户端更新时间”和“服务器更新时间”。此外，我们注意到，与 FSL 相比，PSL 更容易受到切分间带宽受限的影响。由于服务器设计效率低下，PSL 始终最慢；服务器必须同步中间数据，并且必须在每个训练周期中依次处理所有批次的中间数据。

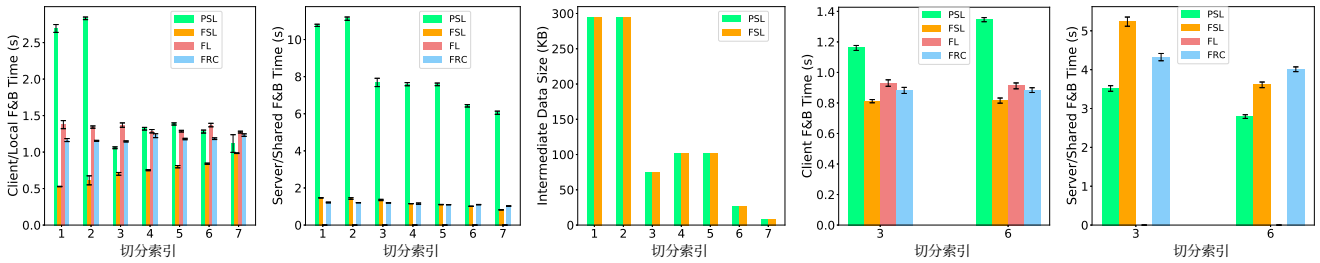
观察 FRC 和 FL 可知，前向与反向传播时间不会随切分索引变化（图 3a 和 3b）。还需注意，FRC 的训练时间效率不高。它通过两次几乎完整的前向与反向传播步骤来更新整个模型 [24]。从相同图中可以看出：FRC 在本地权重和共享权重上的前向与反向传播总时间，几乎是 FL 训练时间的两倍。

在另一种预测场景中比较前向与反向传播时间时，我们也得到了类似结果：由 [44] 实现的一维卷积神经网络（图 3d 和 3e）。由于该神经网络规模有限（仅有两个卷积层），我们仅在两个切分索引处评估这些架构：神经网络第 3 层和第 6 层。即使在这项实验中，我们仍观察到 FSL 的客户端前向与反向传播时间最短。PSL 在每个切分点的表现都最差，而 FL 的表现始终优于 FRC。

与其他分析的架构相比，FSL 在每个训练周期中始终耗时更少。我们发现，*PSL* 的表现不如 *FSL*，这是由其单服务器架构造成的。对比客户端前向与反向传播时间时，*PSL* 与 *FL* 和 *FRC* 的结果相近。*FRC* 的训练时间效率较低，其总前向与反向传播时间几乎是 *FL* 的两倍。

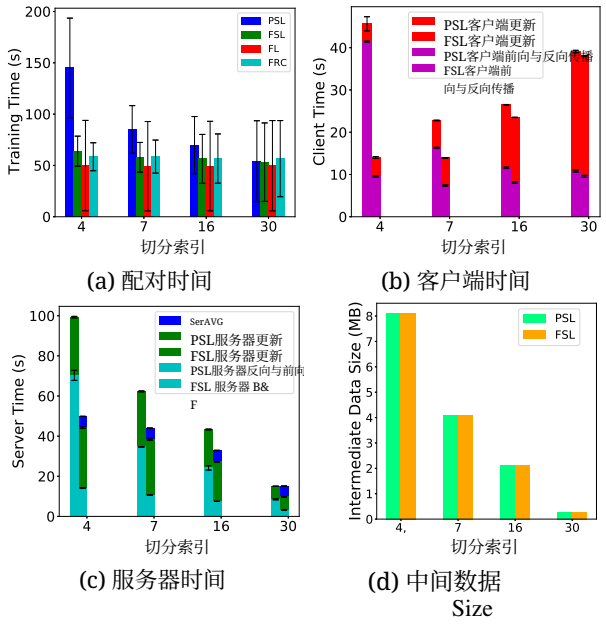
在 CIFAR-10 场景下评估训练时间时，我们发现了不同的趋势（图 4a）：除切分索引为 30 外，PSL 的训练时间最长。此外，FSL 的表现并非总是最佳。当大多数层都在客户端运行时，FL 的训练时间更短。这是因为切分位置移动时中间数据的大小会发生变化，而待发送的数据量较小，整体训练时间就可能更短。图 4b 和 4c 展示了训练期间额外的前向与反向传播时间。现有关于 SL 的研究也讨论过类似现象 [19], [20], [21], [22]。

具体而言，我们表明 FSL 优于 PSL，因为 PSL 的前向与反向传播时间更容易受到中间数据传输的影响。在图3中，随着切分索引减小，FSL 的客户端前向与反向传播时间持续下降，而 PSL 的该时间在切分索引为 6、5 和 3、2 时仍然增加，尽管本实验中的中间数据远少于



(a) LeNet+MNIST 客户端测量数据　(b) LeNet+MNIST 服务器测量数据　(c) LeNet+MNIST 中间数据　(d) VPN 工作负载 客户端测量数据　(e) VPN 工作负载 服务器测量数据

图3：*LeNet+MNIST*：(a) 客户端时间，(b) 服务器时间，(c) 中间数据大小。观察结果：1) 中间数据大小与 PSL 架构所需时间相关，而在 FSL 下相关性较弱；2) FRC 的总训练时间几乎是 FL 的两倍；3) 图中结果是对 20 个客户端的结果取平均后得到的。中间数据大小对应的批量大小为 16，且每张图像都调整为 (1,32,32)。*VPN* 工作负载：(d) 客户端时间 (e) 服务器时间。VPN 数据集也适用类似的结论。测试使用了 5 个客户端和一个输入大小为 (1,784) 的一维神经网络。尽管如此，与其他设置相比，FSL 的客户端前向与反向传播时间最短。



(a) 配对时间　(b) 客户端时间　(c) 服务器时间　(d) 中间数据 Size

图 4：VGG + CiFar10：图中展示了切分索引对以下指标的影响：(a) 平均总体训练时间，(b) 客户端耗时（5 个客户端的平均值），(c) 服务器耗时，(d) 中间数据大小。中间数据大小导致的传输延迟可能会成为训练时间的主要瓶颈（发生在前向与反向传播期间）。

使用 VGG-16 时。这种情况是由单服务器瓶颈造成的。因此，就训练时间而言，FSL 更具可扩展性。这一观察也解释了为什么 PSL 的客户端和服务器前向与反向传播时间始终都大于 FSL。**中间数据和梯度可能导致显著的网络开销**。然而，与 PSL 相比，我们的 FSL 能更有效地缓解这种开销。现有的拆分学习工作 [19] [21] 以及我们的分区策略（第 4.4 节）还能进一步降低通信开销。因此，FSL 中额外的时延并不被视为严重问题与在边缘端进行训练的系统（如联邦学习）相比，构成了瓶颈。

5.2.3. 内存消耗评估。边缘训练和推理系统中各实体的内存使用量限制了可加入系统的设备范围。为了比较各系统在设备内存容量方面的部署灵活性，我们计算并展示了 FSL、PSL、FL 和 FRC 系统中各实体的内存使用量。

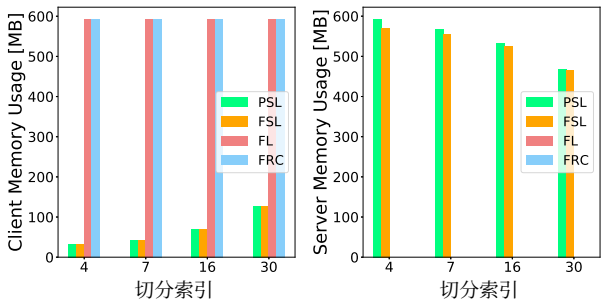
实际内存使用量可能变化很大，因为它取决于多个实现因素，例如使用的库和所实现的远程过程调用（RPCs）。不过，模型大小及其在每一层的激活值是已知的。以下结果表明，与 FL 和 FRC 相比，FSL 的客户端消耗的内存更少；而 FSL 的任一服务器占用的内存都少于 PSL。

图 5 中的两张图分别展示了各架构在客户端和服务器端计算得出的内存需求。x 轴表示切分索引，y 轴表示相应的预期内存使用量（以 MB 为单位）。请注意，FL 和 FRC 不会对神经网络进行切分，因此其内存需求仅显示在左图中。

如图 5a 所示，在 FSL 和 PSL 中，每个客户端神经网络的权重大小以及各层的结果都相同。由于 FRC 和 FL 在训练期间会在客户端计算完整的神经网络，因此在切分索引为 4 到 30 时，它们所需的内存是其他方案的五倍以上。图 5b 则显示，服务器端的内存需求会随着切分索引增加而降低，这符合预期。不过请注意，PSL 服务器需要更多内存来存放来自不同客户端的中间数据。

FSL 与其他系统的内存使用量对比。总之，我们发现，与 PSL 系统相比，FSL 的服务器更轻量。因此，FSL 中的状态管理会更容易。此外，在训练期间，与 FRC 和联邦学习相比，FSL 的客户端更轻量，因此更适合部署在边缘端。

5.2.4. 学习器准确率评估。在本小节中，我们重点评估 SerAVG 的收敛性。



(a) (b)

图 5：与 PSL 相比，FSL 的服务器内存需求较低；与 FRC/FL 相比，FSL 的客户端内存需求较低。*VGG + CiFar10*：（a）客户端和（b）服务器的内存需求（模型权重大小与各层输出大小之和）。批量大小为 32，每张图像均调整为 (3,32,32)。

SerAVG 在第3节中提出并进行了详细介绍。与联邦学习中的 FedAVG 不同，SerAVG 对服务器神经网络权重取平均。我们首先讨论 SerAVG 的正确性，并比较 SerAVG（PSL）与 FedAVG（联邦学习）的准确率。随后，我们比较不同源数据规模和切分索引下的收敛率。

SerAVG 评估：在这组实验中，我们评估当客户端的源数据呈非独立同分布时，SerAVG 是否能够提高参与训练过程的每个模型的准确率。结果见图 6a、6b 和 6c。为了训练 MNIST 模型，我们将训练集分成两部分，即第 0 部分和第 1 部分。这两部分代表客户端不同地理位置上的两组偏斜数据源。接着，我们让每一部分包含 MNIST 数据集中一半标签对应的数据。然后，我们以相同方式将 MNIST 测试集分成两部分。请注意，上述训练集和测试集的划分方式属于极端的非独立同分布情况。例如，使用训练集第 0 部分训练的模型，对训练集第 1 部分中的标签一无所知。为了评估系统在更现实的数据分布下学习和预测非独立同分布数据的能力，我们进一步从完整的 MNIST 数据集中均匀随机选取10% 样本，并将其添加到数据集的四个部分中，使得使用训练集任一部分训练的模型都能够对另一分区中的标签进行分类。请看图 6a、6b 和 6c。x 轴表示用于训练和验证的数据分区，即 0 &1 表示使用第 0 部分数据进行训练，并在验证阶段使用第 1 部分数据。y 轴表示验证集准确率。从左到右，当 FSL、PSL 和 FRC 的服务器端/共享神经网络更浅时，准确率会下降。当切分索引为 3 时，SerAVG、FedAvG 和 PSL 的表现相当。当切分索引大于 3 时，SerAVG 的表现不如 FedAVG，且准确率低于 PSL，但与之接近。我们得出结论：SerAVG 能够提高非独立同分布源数据场景下的准确率，但表现不如 FL 和 PSL。

请注意，FedAVG 会对所有权重求平均，而 FSL 则不会在 SerAVG 设置中共享客户端的权重。因此，FSL 客户端无法从其他客户端计算出的梯度中获益，这可能会导致准确率降低。比较 PSL 和 FSL，PSL 服务器会利用所有客户端的批次输出来优化，使损失最小化。另一方面，SerAVG 会基于 FedAVG 启发式地对各服务器上训练得到的权重求平均。因此，与 PSL 相比，SerAVG 的准确率会较低。而且，使用非独立同分布数据集训练的每个客户端神经网络，都无法从其他非独立同分布数据集中提取到多少特征。

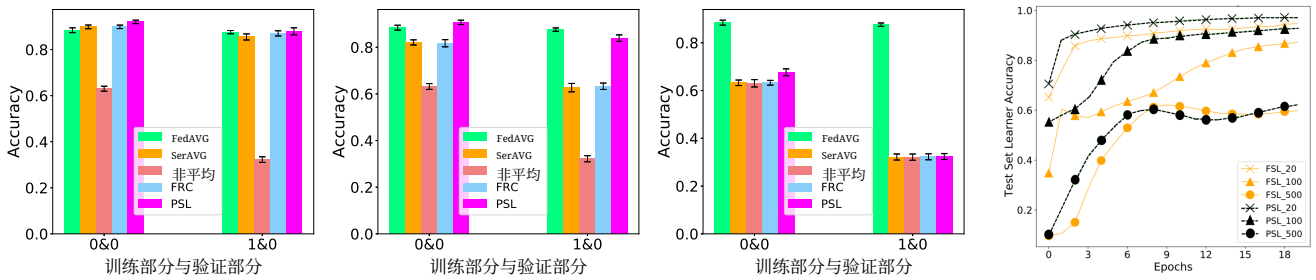
另请注意，图7a中使用独立同分布（IID）分区的 CIFAR10 数据集和 VGG16 神经网络时，准确率随切分索引变化而下降，趋势相似但更为平缓。MNIST 和 CIFAR10 数据集上的两项实验表明，即使应用于不同的神经网络模型和切分索引，SerAVG 仍能保持相似的准确率模式。

资源需求与准确率之间的权衡。在本实验中，我们评估了资源需求较低时 FSL 的准确率，即输入数据量有限且客户端模型运行资源受限的情况。我们发现，对于 LeNet 和 MNIST，只要每个客户端拥有足够的数据来训练机器学习模型，验证数据集的准确率就能达到87% 至93%。通过改变客户端数量，我们量化了这两种架构的预期准确率下降幅度。结果如图6d所示。x 轴表示训练轮次的索引，y 轴表示经过相应训练轮次后得到的测试集准确率。我们在 20、100 和 500 个客户端之间采用 IID 数据集，以研究数据量如何影响收敛。在图6a、图6b、图6c 和图7a中，我们还注意到，当切分索引较小时，FSL（采用 SerAVG 机制）仍能保持较高的准确率，因此即使性能要求较高，也可以部署在资源受限设备上。根据类似场景中的先前结果，我们预计，只要进一步调整超参数，这两种架构都能达到更高的准确率 [45], [46]。虽然我们的准确率结果显示了 95% 置信区间，但参数调整不在本文讨论范围之内。

5.3. 隐私感知联邦分割学习的评估结果

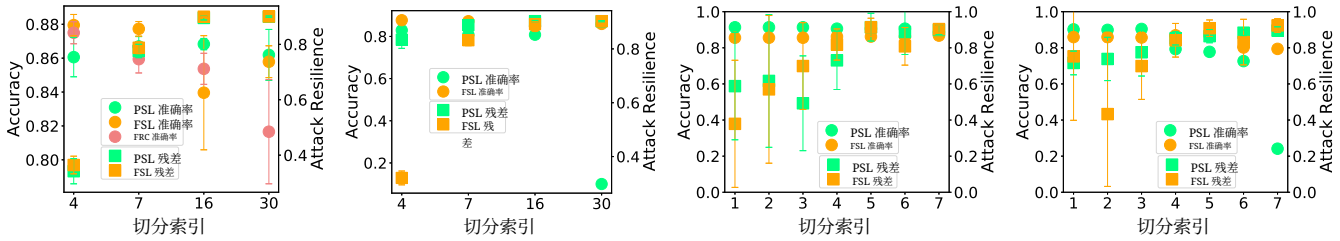
本节展示隐私感知 FSL（PAFSL）架构的评估结果，并说明该架构能够提供一定的隐私保障。FSL 客户端不会共享源数据和模型权重，因此攻击者无法直接访问源数据，也无法利用模型权重通过模型反演攻击重建源数据 [27], [28]。然而，与在客户端训练完整模型的联邦重构（FRC）[24] 相比，FSL 仍会通过网络发送中间数据，以完成客户端与（边缘端）服务器之间的前向传播和反向传播。攻击者可能利用此类数据还原源数据，例如，在第 4.1 节中使用特定数据集训练的自编码器 [35] 神经网络。

为评估我们的方法如何缓解此类漏洞，我们首先介绍评估设置、攻击者自编码器神经网络的设计与使用，以及实验



(a) 准确率（切分索引 = 3） (b) 准确率（切分索引 = 5） (c) 准确率（切分索引 = 7） （d）准确率（客户端数量）

图6：在图（a）、（b）和（c）中，SerAVG 的准确率优于非平均，但低于 FedAVG，不过总体上与其接近。SerAVG：对服务器神经网络的权重取平均；非平均：每对参与方各自训练；FedAVG：对完整神经网络的权重取平均。在图（d）中，FSL 和 PSL 的准确率相近。



（a）隐私无感 (VGG16+CIFAR-10) （b）直流 1 & 乘数=2.0 (VGG16+CIFAR-10) (c) 隐私无感 (LeNet+MNIST) (d) 直流 1 & 乘数=0.3 (LeNet+MNIST)

图7：基于切分索引的隐私无感和隐私感知架构的准确率与攻击韧性。注：图注中的直流 1 指 $DC\ \ Frequency = 1$。这些图表明，要达到较高的准确率和攻击韧性，切分索引既不能过大也不能过小，而损失乘数是增强攻击韧性的另一种方法。此外，损失乘数可能比直流频率更实用，因为它不会给训练步骤带来额外开销。

方法。随后，我们讨论不同隐私方法的结果，即 NoPeek [29] 和基于客户端的隐私方法（CPA），以及神经网络不同划分方式的隐私级别。NoPeek 解决一个包含两个损失函数的多目标优化问题：一个用于最大化准确率，另一个用于最大化源图像与中间数据之间的差异。对于 CPA，我们评估 CPA-DC 和 CPA-DP。CPA-DC（第4.3节）交替优化 NoPeek 中的两个损失函数。CPA-DP 则在客户端应用差分隐私随机梯度下降 [30] 算法。最后，受第4.4节启发，我们评估客户端神经网络和服务器神经网络的不同划分方式所提供的隐私保障。本节最后展示结果，证明我们提出的 FSL 对隐私攻击具有较强的韧性，并在训练效率方面具有优势。

5.3.1. 评估设置。我们的隐私感知联邦分割学习和 PSL 在隐私无关版本的基础上加入了 CPA。通过仅考虑顺序神经网络（例如 LeNet 和 VGG16），简化了神经网络的划分。我们在与隐私无关设置相同的硬件（即 NVIDIA RTX6000 和 NVIDIA V100）上，使用相同的图像分类工作负载（例如 MNIST 和 CIFAR10）测试了这两个系统。

5.3.2. 配置攻击者的自编码器神经网络。在本小节中，我们将说明如何定义隐私攻击，前提是满足第 4.1 节中提到的假设。为理解隐私攻击的工作原理，有必要回顾一下我们所用自编码器神经网络的工作方式。具体而言，自编码器由两部分组成：编码器和解码器。我们需要一个与源图像相似的数据集来训练自编码器神经网络。

编码器使用卷积层从输入数据集中提取潜变量，而解码器则使用编码器最后一层的激活函数输出，并对卷积层进行转置，以重建编码器的输入数据集。因此，编码器神经网络的结构与客户端神经网络的结构相同，而攻击者使用的神经网络是解码器。编码器的 i^{th} 层对应于解码器的 i^{th} 最后一层，其中卷积层经过转置。我们假设攻击者的解码器结构严格遵循客户端神经网络的结构。因此，我们预期基于分割学习的系统在现实世界中具有更强的攻击韧性。

5.3.3. 隐私评估：方法。在这部分评估中，我们旨在展示，基于不同数据集和神经网络的拆分学习系统中，CPA 和精心设计的神经网络划分方式在保持高准确率的同时，都能提供很强的抗攻击能力。

每项实验都包含多次试验，每次试验都使用不同的随机种子来初始化自动编码器的权重和数据加载器

。在每次试验中，攻击者使用了一个包含与学习者源数据集具有相似特征的数据集。为了重构 MNIST，我们选择了 EMNIST [47], ；对于 CIFAR10，我们选择了 CIFAR100 [9]。EMNIST 数据集包含手写字符，而非 MNIST 中的数字，因此线条和曲线等特征相同，攻击者可以解码这些激活函数输出。同样，CIFAR100 数据集包含 100 个类别的 RGB图像，而 CIFAR10 中只有 10 个类别，因此可以利用共有特征（例如对猫或狗进行分类），根据 CIFAR10 数据集的激活函数输出进行重构。

在实验的每次试验中，我们首先让攻击者学习重建其数据集。攻击者根据均方误差损失，即 一种用于衡量原始图像与重建图像差异的损失函数，在每个训练轮次中更新权重。经过 20 个训练轮次后，我们使用解码器根据中间数据重建学习者的数据集。

训练自编码器神经网络时，我们使用与学习者相同的神经网络结构和源数据训练一个新的分类器，并用它对重建图像进行分类，共训练 20 个训练轮次。该分类器的攻击韧性 τ 的均值和标准差见第 4.2 节。

我们展示了四种系统（即 POFSL、POPSL、PAFSL 和 PAPSL）在不同 CPA 和切分索引下的比较结果。随后，我们评估 FSL 和 PSL 在准确率与攻击韧性之间的权衡。

5.3.4. 使用 NoPeek 进行隐私评估。如第 2 节所述，NoPeek 通过距离相关性（DC）损失函数，使源数据与中间数据之间的差异最大化；同时利用预测和标签来最大化准确率，从而解决多目标优化问题。为解决此类优化问题，NoPeek 必须通过网络共享损失值，这可能带来漏洞，或增加维护梯度图的复杂性。如表 1 所示，在 MNIST 数据集和 LeNet 神经网络上训练相同数量的训练轮次和客户端后，与隐私无关实验相比，该方法同时具有较高的攻击韧性（PSL 为 97%，FSL 为 98%）和较高的学习者准确率（PSL 为 97%，FSL 为 96%）。

情况	PSL	FSL
攻击韧性(τ)	0.9733	0.9837
学习器准确率	0.9702	0.9614

表 1：20 个客户端训练20 轮次时的 NoPeek 统计数据。

5.3.5. 通过距离相关性实现基于客户端的隐私保护方法的评估结果。为减轻损失值共享的缺点，我们考虑一种不向客户端外部传输数据的新方法，作为对 NoPeek 的改进。在基于距离相关性（DC）的客户端隐私保护方法（CPA）中，我们交替优化与 NoPeek 中类似的两个目标。如公式2所示，需要评估*DC* 频率 *(F)*和损失乘数 *(m)*。*DC* 频率 *(F)*定义了准确率损失函数优化一次后，距离相关性损失函数要优化的次数。损失乘数 *(m)*作用于损失函数的结果，但其效果等同于将学习率乘以因子 m。这两个参数分别通过改变距离相关性损失的优化频率，以及改变该优化过程中所应用梯度的学习率，来控制中间数据与源数据之间的差异程度。

我们注意到 CPA-DC 存在多种权衡。首先，CPA-DC 的训练时间效率不如 NoPeek。NoPeek 可以同时优化其两个目标，而 CPA-DC 必须依次求解这两个目标。不过，我们认为 NoPeek 通过网络传输的信息超出了必要范围。其次，直流频率（F）和 损失乘数（m）之间也存在权衡。增加F 会增加用于优化直流损失的训练轮次，因此攻击韧性会更高，但训练时间也会更长。此外，我们引入了m，这样就可以保持较小的F ，只增加m，从而在维持攻击韧性的同时缩短训练时间。我们将较大的m 乘到直流损失上，这与增大梯度下降步长类似。因此，在m较大时，维持攻击韧性所需的直流训练轮次会减少，但直流损失的准确性会降低。根据上述讨论，直观而言，我们预期将较大的m与值为1的F 结合起来，可以在训练时间效率和直流损失梯度的准确性之间取得平衡。这两个参数在生产环境中应经过谨慎设计。

我们首先在 MNIST 分类实验中测试了不同的直流频率，并将损失乘数固定为 0.1（相当于将学习率降低 10%），研究准确率与攻击韧性之间的权衡，如图 8a 所示。横轴表示直流频率，左纵轴表示学习器准确率，右纵轴表示相应的攻击韧性。

从图 8[1]，的上图可以看出，随着直流频率增加，隐私感知联邦半监督学习（PAFSL）和隐私感知分割学习（PAPSL）系统的攻击韧性均有所提高，而学习器准确率则有所下降，这符合预期。

请注意，在大多数直流频率取值下，PAFSL 的准确率更高且韧性良好。当直流频率为 10 到 20 时，PAFSL 和 PAPSL 的攻击韧性相差不到10% ，而 PAFSL 的准确率超过90% 。当直流频率为 25 到 35 时，PAPSL 无法学习任何特征，而 PAFSL 的准确率仍约为80%。当直流频率为 5 时，PAPSL 优于 PAFSL，二者准确率相近，且 PAPSL 的攻击韧性高出约20% 。

结果表明，与 CPA-DC 搭配的 PAFSL 更容易调优，从而获得较高的准确率和攻击韧性。在更宽的 *DC* 频率范围内，与 *PAPSL* 相比，*PAFSL* 的准确率更高，且具有良好的攻击韧性。这是因为 SerAVG 和 PSL 中的学习率（步长）所致。PSL 的服务器权重更新规则如公式 5 所示。

$$W_g^{t+1} = W_g^t - \eta \sum_{i=1}^{N_C} \frac{\partial g(f(x_i))}{\partial W_g} \tag{5}$$

1. DC 频率为零对应于不采用隐私感知方法的 POFSL 和 POPSL。

FSL 的服务器权重更新规则如公式 6 所示。

$$W_g^{t+1} = \frac{\sum_{i=1}^{N_C}(W_g^t - \eta\frac{\partial g(f(x_i))}{\partial W_g})}{N_C} = W_g^t - \frac{\eta}{N_C}\sum_{i=1}^{N_C}\frac{\partial g(f(x_i))}{\partial W_g}, \quad (6)$$

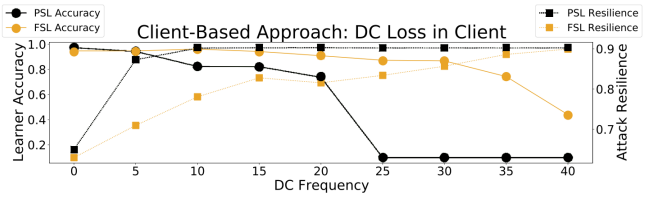
其中， W_g^t 表示迭代 t时服务器中的权重， g 是服务器神经网络， f 是客户端神经网络，表示第 i 个数据批次， N_C 是客户端数量， η 是步长。直观而言，由于 PSL 的步长较大，其中间数据能更快地使其服务器神经网络陷入混乱，相比之下，FSL 服务器需要更长时间。此外，陷入混乱的服务器神经网络还能进一步扰乱客户端神经网络。这印证了我们的观察：随着直流频率 *(F)*增加，PSL 的准确率和攻击韧性会很快变得不稳定。因此，我们得出结论：与 *PSL* 相比，*FSL* 更容易调整。

在图7b和图7d中，在直流频率 *(F)*固定的情况下，我们展示了不同模型和数据集在不同损失乘数 *(m)* 下的准确率（左侧 y 轴）和攻击韧性（右侧 y 轴）。此外，我们比较了隐私无感知情况（图7a和图7c），以及不同切分索引（x 轴）下的隐私感知情况。正如预期，增大损失乘数 *(m)*可以增强攻击韧性，但会降低准确率，尤其是在客户端神经网络较深时。

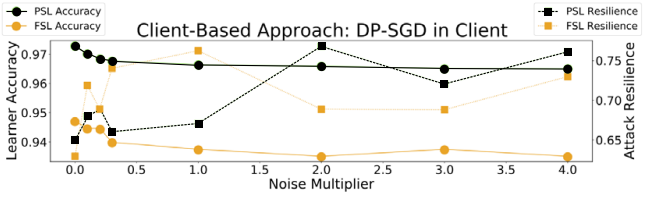
总体而言，我们对 CPA-DC 的评估表明，采用较小的 *DC* 频率 *(F)* 和较大的损失乘数 *(m)*，能够兼顾良好的攻击韧性和准确率。此外，我们的 FSL 准确率更高，攻击韧性与 PSL 相近。因此，我们得出结论：CPA-DC 能够抵御我们的攻击者模型。

5.3.6. 使用差分隐私方法进行隐私评估。上一节讨论了 CPA-DC；除了直流之外，还有其他轻量级方法可以增强隐私保障：它们会向客户端神经网络添加噪声，同时避免快速耗尽客户端电池。本节将比较 CPA-DP（在客户端内部使用流行的 DP-SGD [30] 算法）和 CPA-DC。此外，我们还将说明，在 FSL 系统中直接使用 DP-SGD 会导致准确率较低。该实现通过 Opacus [48] 库提供的 DP-SGD 优化器扩展了 POFSL。此方法会根据 $noise\ multiplier\ \epsilon$，在反向传播期间向梯度添加服从正态分布的随机噪声。该参数控制所添加噪声的幅度。请注意，在每个距离相关性轮次中，直流会沿特定方向生成梯度，以降低中间数据与源数据之间的相关性。因此，在学习器准确率相同的情况下，我们预期 CPA-DP 的隐私水平会低于 CPA-DC。本节的重点是说明，尽管 DP 的隐私性不如直流，CPA 仍可与 DP 等其他隐私方法结合使用。

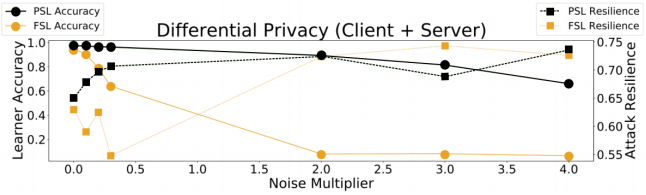
我们在图 8b 中总结了 CPA-DP 的结果。该图展示了切分索引等于 3 时的结果。横轴为 $noise\ multiplier$。FSL 的攻击韧性以及



(a) CPA-DC（乘数=0.1）



(b) CPA-DP



(c) 完整神经网络中的差分隐私随机梯度下降

图 8：LeNet+MNIST：使用 20 个客户端和客户端隐私保护方法的切分索引 3 时的学习器准确率和攻击韧性（τ）（通过直流 (a) 和 DP (b)），以及全局学习器模型上的差分隐私随机梯度下降 (c)。采用这两种基于客户端的策略的 FSL，均能保证较高水平的隐私和准确率。

带有 $noise\ multiplier > 0$ 的 PSL 始终比带有 $noise\ multiplier = 0$ 的 FSL 和 PSL 高近 5% 。与此同时，从 $noise\ multiplier = 0$ 到 $noise\ multiplier = 4$，FSL 或 PSL 的准确率下降均不到1% 。

CPA 是一种通用方法，可通过定制不同的方法来增强攻击韧性。评估结果表明，CPA-DP 也能提高攻击韧性，而学习器准确率变化不大。与 CPA-DC 相比，两种方法提供的学习器准确率相近，但 CPA-DC 的攻击韧性更高。

现在，我们将 CPA-DP 与在客户端神经网络和服务器神经网络中应用差分隐私随机梯度下降进行比较。图 8c 显示，其攻击韧性较高，但当 $noise\ multiplier \geq 0.3$ 时，FSL 的学习器准确率会降至 60% 以下。与此同时，PSL 的表现与使用 CPA-DP 时相似。因此，与在客户端和服务器中都应用差分隐私随机梯度下降相比，CPA-DP 被认为是提升 FSL 攻击韧性的更好方法。FSL 在差分隐私随机梯度下降和高噪声下准确率较低，原因可归结于其 SerAVG。应用 SerAVG 后，服务器神经网络中随机噪声的分布可能是任意的，如方程 7 所示，而客户端神经网络中的噪声则保持不变。因此，FSL 中最终得到的完整神经网络可能无法收敛。

$$W_g^{t+1} = W_g^t - \frac{\eta}{N_C}\sum_{i=1}^{N_C}(\frac{\partial g(f(x_i))}{\partial W_g} + e_{g_i}^t) \quad (7)$$

变量 $e_{g_i}^t$ 表示根据 (()) 在迭代时添加到第 i t $e_{g_i}^t \sim$ 个服务器神经网络中的高斯噪声 $\mathcal{N}\ 0,\ noise\ multiplier \times max\ gradient\ norm^{\ 2}$ [48]

5.3.7. 使用不同神经网络划分方式进行评估。在图7a和图7c中，右侧 y 轴表示攻击韧性，x 轴表示切分索引。总体而言，如果客户端神经网络更深（即从较小的切分索引增大到较大的切分索引），攻击韧性会提高，而准确率会下降（与第5.2.4节中的 SerAVG 评估结果一致）。增加更多层后，中间数据所包含的源数据特征会减少，只保留能够提高分类准确率的特征。因此，保留下来的特征更少，攻击者重建源数据的能力也会受到抑制。

我们还想强调，对于 CIFAR10 工作负载，当客户端有 7 层时，攻击韧性可达到 80%。对于 MNIST 工作负载，在切分索引为 4 时，我们也得到了相同的结果。我们没有使用额外的隐私保护方法；正如前文所述，由于中间数据可能更小，客户端神经网络更深也能降低传输延迟。

切分索引是影响训练时延、准确率和隐私的重要超参数。不同的切分索引带来以下权衡：较深的客户端神经网络会增加边缘端的资源需求，但能缩短传输时间并增强攻击韧性。另一方面，较浅的服务器神经网络可能会导致 *SerAVG* 的准确率较低。

此外，系统架构师可以结合上述方法，例如，在客户端使用中等深度的神经网络并采用 CPA-DC，以在资源需求和性能之间取得平衡。如图 7d 所示，使用 $cut\ index = 4$、$DC\ Frequency = 1$ 和 $loss\ multiplier = 0.3$ 时，我们仍可获得约 80% 的攻击韧性和超过90% 的准确率。

另一方面，在比较 FSL 和 PSL 时，我们注意到 FSL 有更多超参数需要调优。不过，在图 7 报告的所有实验中，当切分索引较大时，FSL 的准确率优于 PSL。因此，我们得出结论：精心设计神经网络的划分方式，在应用于混合联邦拆分学习系统时，能带来最大的收益。

6. 结论

联邦学习（FL）、拆分学习（SL）等系统以及后续研究，旨在适应分布式模型训练和推理等特定场景。然而，随着边缘端和受限设备的最新发展，它们不够灵活，无法适应某些使用场景。

在本研究中，我们提出并全面评估了联邦拆分学习（FSL），这是一种能够高效训练和推理、并为客户端源数据提供高级别隐私保护的系统。我们提出了一种基于客户端的隐私方法（CPA），用于拆分学习系统，通过向中间数据添加噪声来提供较强的抗攻击韧性。我们还研究了在 FSL 中采用不同方式划分神经网络时的训练时间、准确率和隐私级别。作为后续工作，我们计划研究基于预测的神经网络划分方法。

此外，研究还将 FSL 与边缘端现有的神经网络训练与推理系统进行了比较，并解释了这些系统相对于 FSL 的优缺点。在所分析的主要系统中，并行拆分学习（PSL）的主要局限在于服务器缓慢且具有状态，而联邦重构（FRC）系统在训练时间和推理方面效率低下。

致谢

本研究得到美国国家科学基金会奖项 CNS-1908574 和 CNS-1908677 的支持。

参考文献

[1] D. H. Mahlool 和 M. H. Abed，《联邦学习综述：概念与应用》，*CoRR*，第 abs/2201.09384 卷，2022 年。 [在线]。网址：https://arxiv.org/abs/ 2201.09384[2] P. Moritz、R. Nishihara、S. Wang、A. Tumanov、R. Liaw、E. Liang、M. Elibol、Z. Yang、W. Paul、M. I. Jordan 和 I. Stoica，《Ray：面向新兴人工智能应用的分布式框架》，OSDI'18 系列。美国：USENIX 协会，2018 年，第 561–577 页。[3] C. Park、D. Hong 和 C. Seo，《针对模型反演攻击的差分隐私学习攻击式评估方法》，*IEEE Access*，第 7 卷，第 124 988–124999 页，2019 年。[4] S. Ramaswamy、R. Mathews、K. Rao 和 F. Beaufays，《移动键盘中的表情符号预测联邦学习》，2019 年。[5] A. e. a. Hard，《移动键盘预测的联邦学习》，*arXiv* 预印本 *arXiv:1811.03604*，2018 年。[6] D. Leroy、A. Coucke、T. Lavril、T. Gisselbrecht 和 J. Dureau，《关键词检测的联邦学习》，载于 *ICASSP 2019*，2019 年。[7] K. Simonyan 和 A. Zisserman，《用于大规模图像识别的超深卷积网络》，*CoRR*，第 abs/1409.1556 卷，2015 年。[8] J. Deng、W. Dong、R. Socher、L.-J. Li、K. Li 和 L. Fei-Fei，《ImageNet：大型分层图像数据库》，载于*CVPR*，2009 年。[9] A. Krizhevsky，《从微型图像中学习多层特征》，技术报告，2009 年。[10] O. Gupta 和 R. Raskar，《多个代理上的深度神经网络分布式学习》，《网络与计算机应用杂志》，2018 年 8 月。[11] A. Abedi 和 S. S. Khan，《FedSL：循环神经网络中分布式序列数据的联邦拆分学习》，2020 年。[12] C. Thapa、M. A. P. Chamikara 和 S. Camtepe，《SplitFed：联邦学习与拆分学习的结合》，2020 年。[13] J. Jeon 和 J. Kim，《隐私敏感的并行拆分学习》，载于*ICOIN*，2020 年。[14] H. McMahan、E. Moore、D. Ramage、S. Hampson 和 B. A. y Arcas，《利用去中心化数据进行深度网络的高效通信学习》，载于*AISTATS*，2017 年。[15] X. Li、K. Huang、W. Yang、S. Wang 和 Z. Zhang，《FedAVG 在非独立同分布数据上的收敛性》，载于*ICML*，2020 年。[16] Y. Fraboni、R. Vidal 和 M. Lorenzi，《联邦学习中针对模型聚合的搭便车攻击》，载于*AISTATS*，2021 年。

[17] T. Li、A. K. Sahu、M. Zaheer、M. Sanjabi、A. Talwalkar 和 V. Smith，《异构网络中的联邦优化》，2020 年。[18] Y. Gao、M. Kim、S. Abuadbba、Y. Kim、C. Thapa、K. Kim、S. A. Camtep、H. Kim 和 S. Nepal，《物联网联邦学习与拆分学习的端到端评估》，载于*SRDS*，2020 年。[19] H. Zhou、W. Zhang、C. Wang、X. Ma 和 H. Yu，《BBNet：一种用于边缘云协同推理的新型卷积神经网络结构》，《传感器》，2021 年。[20] A. E. Eshratifar、A. Esmaili 和 M. Pedram，《BottleNet：一种用于智能移动云计算服务的深度学习架构》，载于*ISLPED*。IEEE，2019 年。[21] Y. Matsubara、M. Levorato 和 F. Restuccia，《深度学习应用的分割计算与提前退出：综述与研究挑战》，*arXiv* 预印本 *arXiv:2103.04505*，2021 年。[22] Y. Matsubara、S. Baidya、D. Callegaro、M. Levorato 和 S. Singh，《面向边缘辅助实时系统的蒸馏分割深度神经网络》，载于*HotEdgeVideo*。美国纽约：ACM，2019 年。[23] V. Turina、Z. Zhang、F. Esposito 和 I. Matta，《联邦学习还是拆分学习？混合分割与联邦学习架构的性能和隐私分析》，载于*IEEE CLOUD*，2021 年。[24] K. Singhal、H. Sidahmed、Z. Garrett、S. Wu、J. K. Rush 和 S. Prakash，《联邦重建：部分本地化的联邦学习》，载于*NeurIPS*，2021 年。[25] J. Yan 和 J. Yuan，《软件定义网络流量分类综述》，载于*HotICN*，2018 年。[26] H. Yang、J. Zhao、Z. Xiong、K.-Y. Lam、S. Sun 和 L. Xiao，《无人机支持网络的隐私保护联邦学习：基于学习的联合调度与资源管理》，《*IEEE*通信领域精选期刊》，2021 年。[27] M. Fredrikson、S. Jha 和 T. Ristenpart，《利用置信度信息的模型反演攻击及基本对策》，载于*CCS*。美国纽约：ACM，2015 年。[28] S. Hidano、T. Murakami、S. Katsumata、S. Kiyomoto 和 G. Hanaoka，《预测系统的模型反演攻击：无需了解非敏感属性》，载于*PST*，2017 年。[29] P. Vepakomma、A. Singh、O. Gupta 和 R. Raskar，《NoPeek：降低分布式深度学习中共享激活值的信息泄露》，载于*ICDMW*，2020 年。[30] M. Abadi、A. Chu、I. Goodfellow、H. B. McMahan、I. Mironov、K. Talwar 和 L. Zhang，《差分隐私深度学习》，CCS '16 系列。ACM，2016 年。[31] Y. Mao、S. Yi、Q. Li、J. Feng、F. Xu 和 S. Zhong，《利用边缘计算从差分隐私神经网络激活值中学习》，载于*SEC*，2018 年。[32] J. Huang、C. Samplawski、D. Ganesan、B. Marlin 和 H. Kwon，《Clio：实现跨物联网与云的深度学习流水线自动编译》，载于*MobiCom*。ACM，2020 年。[33] S. Qiu、D. Wang、G. Xu 和 S. Kumari，《基于扩展混沌映射、面向移动轻量级设备的实用且可证明安全的三因素认证协议》，*IEEE Transactions on Dependable and Secure Computing*，2020 年。[34] Q. Jiang、N. Zhang、J. Ni、J. Ma、X. Ma 和 K.-K. R. Choo，《面向云辅助自动驾驶汽车的统一生物特征隐私保护三因素认证与密钥协商》，*IEEE Transactions on Vehicular Technology*，2020 年。[35] G. E. Hinton 和 R. S. Zemel，《自编码器、最小描述长度与亥姆霍兹自由能》，载于*NIPS*。美国加利福尼亚州旧金山：Morgan Kaufmann Publishers Inc.，1993 年。[36] Y. Kang、J. Hauswald、C. Gao、A. Rovinski、T. Mudge、J. Mars 和 L. Tang，《Neurosurgeon：云与移动边缘之间的协同智能》，载于*ASPLOS*，2017 年。[37] T. Unterthiner、D. Keysers、S. Gelly、O. Bousquet 和 I. Tolstikhin，《根据权重预测神经网络准确率》，2020 年。[38] K. K. 等，"从变色龙测试平台中吸取的经验教训"，载于*ATC*，2020 年 7 月。[39] A. 等，Paszke，"Pytorch：一种命令式风格、高性能的深度学习库"，载于*NeurIPS*。Curran Associates, Inc.，2019 年。[40] T. Ryffel、A. Trask、M. Dahl、B. Wagner、J. Mancuso、D. Rueckert 和 J. Passerat-Palmbach，"用于保护隐私的通用深度学习框架"，*arXiv* 预印本 *arXiv:1811.04017*，2018 年。[41] (2020) PyGrid。 [在线]。网址：https://github.com/OpenMined/ PyGrid [42] Y. Lecun、L. Bottou、Y. Bengio 和 P. Haffner，"应用于文档识别的基于梯度的学习"，*IEEE* 会议录，1998 年。[43] Y. LeCun 和 C. Cortes，"MNIST 手写数字数据库"，2010 年。[44] W. Wang、M. Zhu、J. Wang、X. Zeng 和 Z. Yang，"使用一维卷积神经网络进行端到端加密流量分类"，载于*ISI*，2017 年。[45] J. Bergstra 和 Y. Bengio，"用于超参数优化的随机搜索"，*JMLR*，2012 年。[46] L. Li、K. Jamieson、G. DeSalvo、A. Rostamizadeh 和 A. Talwalkar，"Hyperband：一种用于超参数优化的新型基于多臂老虎机的方法"，*JMLR*，2018 年。[47] G. Cohen、S. Afshar、J. Tapson 和 A. van Schaik，"EMNIST：将 MNIST 扩展至手写字母"，载于*IJCNN*，2017 年。[48] (2021 年 10 月) Opacus。 [在线]。网址：https://opacus.ai/