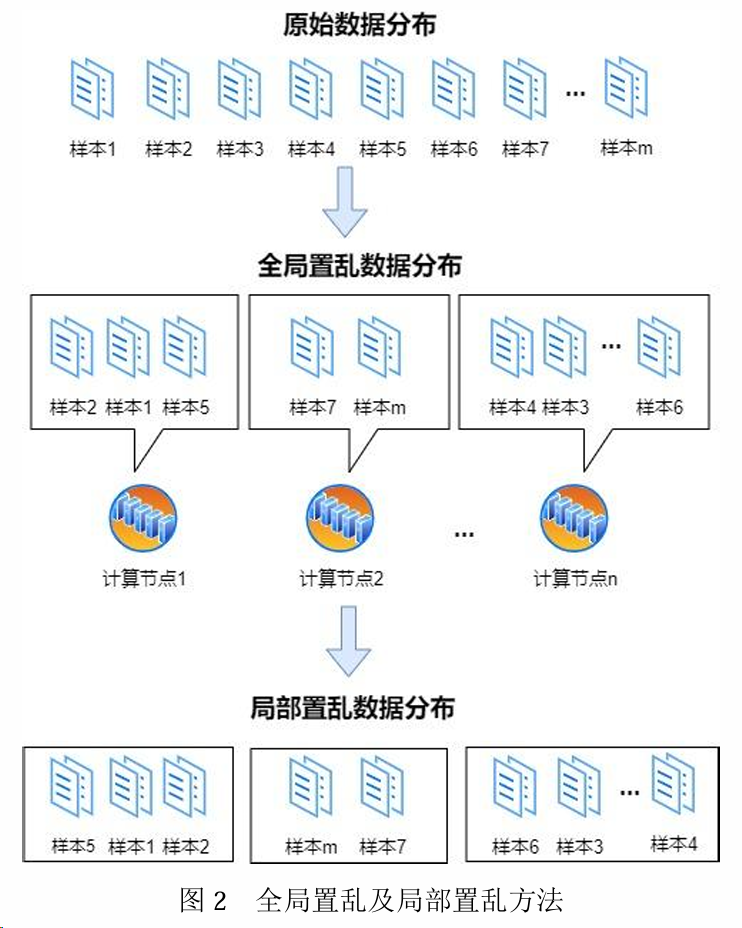
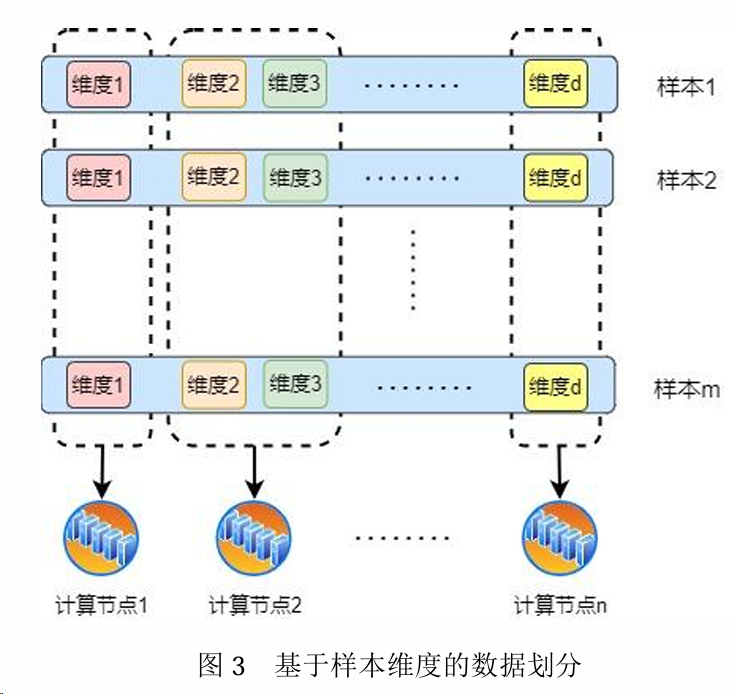
分布式算法、分布式GNN的实现

分布式训练系统的定义为“训练模型的工作负载被分配到多个工作节点，并且这些工作节点通过并行的工作方式加速模型训练。分布式训练不仅用于机器学习模型，而且适用于计算和时间密集型任务”。“分布式训练采用数据并行或模型并行的方式实现模型高效训练。其中，数据并行是指原始数据集被拆分成多个子数据集并分配到不同计算节点，再由神经网络或机器学习算法处理，最后将各个局部计算结果聚合。模型并行是指大模型首先被拆分成若干子模型并分配至相应计算节点，再由各计算节点进行处理，最后将子模型计算中间结果进行聚合”。

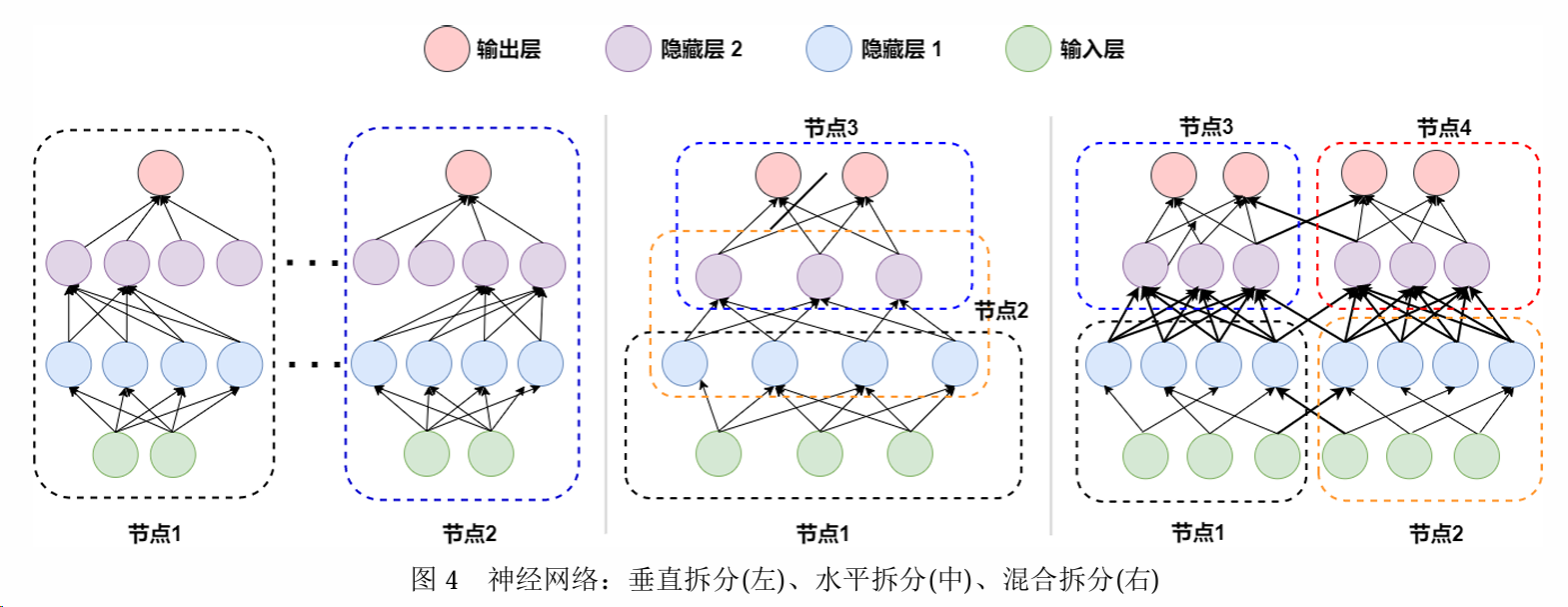
分布式训练系统中的拆分问题一般对于数据集或模型进行拆分，然后将拆分后的子数据或者子模型分配至合适的计算节点。



以上属于基于样本的数据并行，部或全局置乱采样方法成本低并且能够充分利用全体数据样本，避免了一些关键样本未被采样的问题。除了基于样本的数据并行，还有基于样本维度的数据并行。其划分方式如下所示：

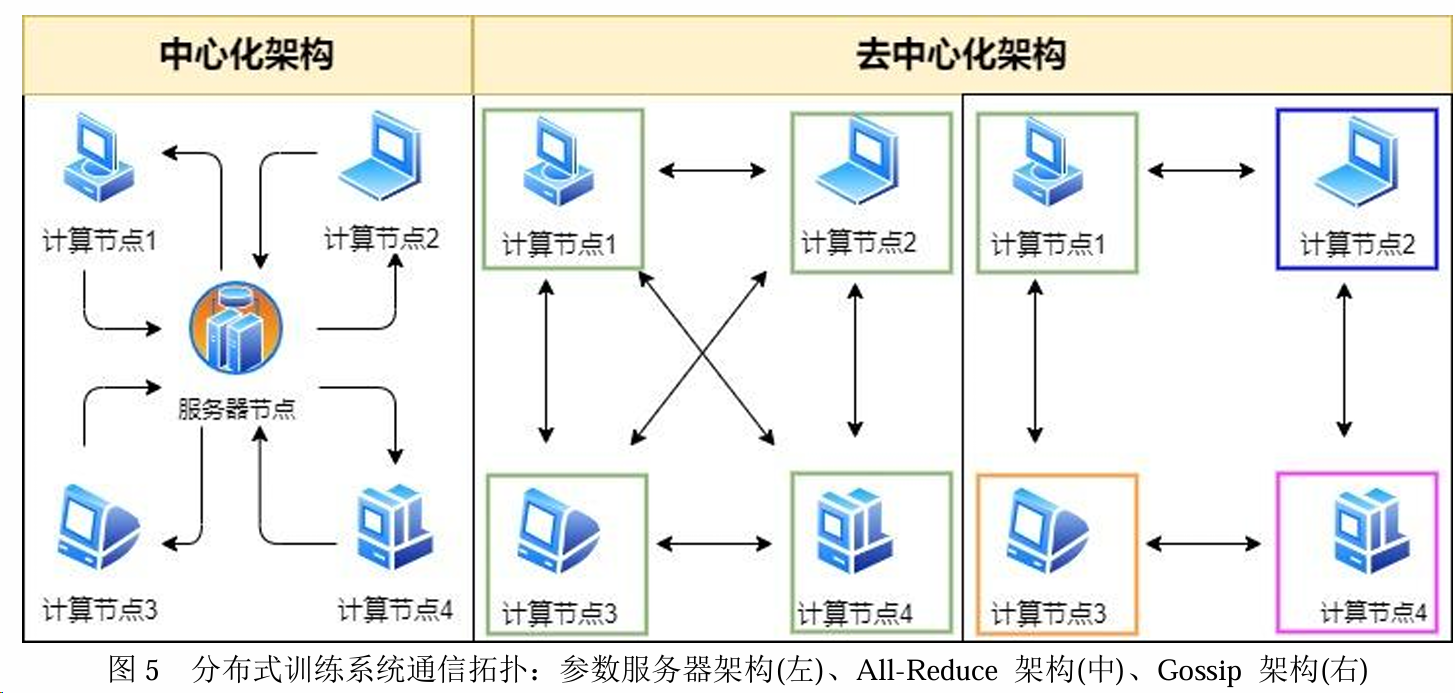


如果训练任务模型过大且无法通过单机方式实现存储，则需要对模型进行拆分。模型并行基于特定的划分规则将原始模型拆分成若干子模型，并且将拆分后的子模型分配至不同的计算节点。根据神经网络的结构特殊性，其模型并行按照不同的拆分方式可以分为：水平拆分、垂直拆分以及混合拆分。如下图所示：



除了数据并行和模型并行，还有研究人员提出混合并行的模式，将数据并行模式和模型并行模式集合起来。

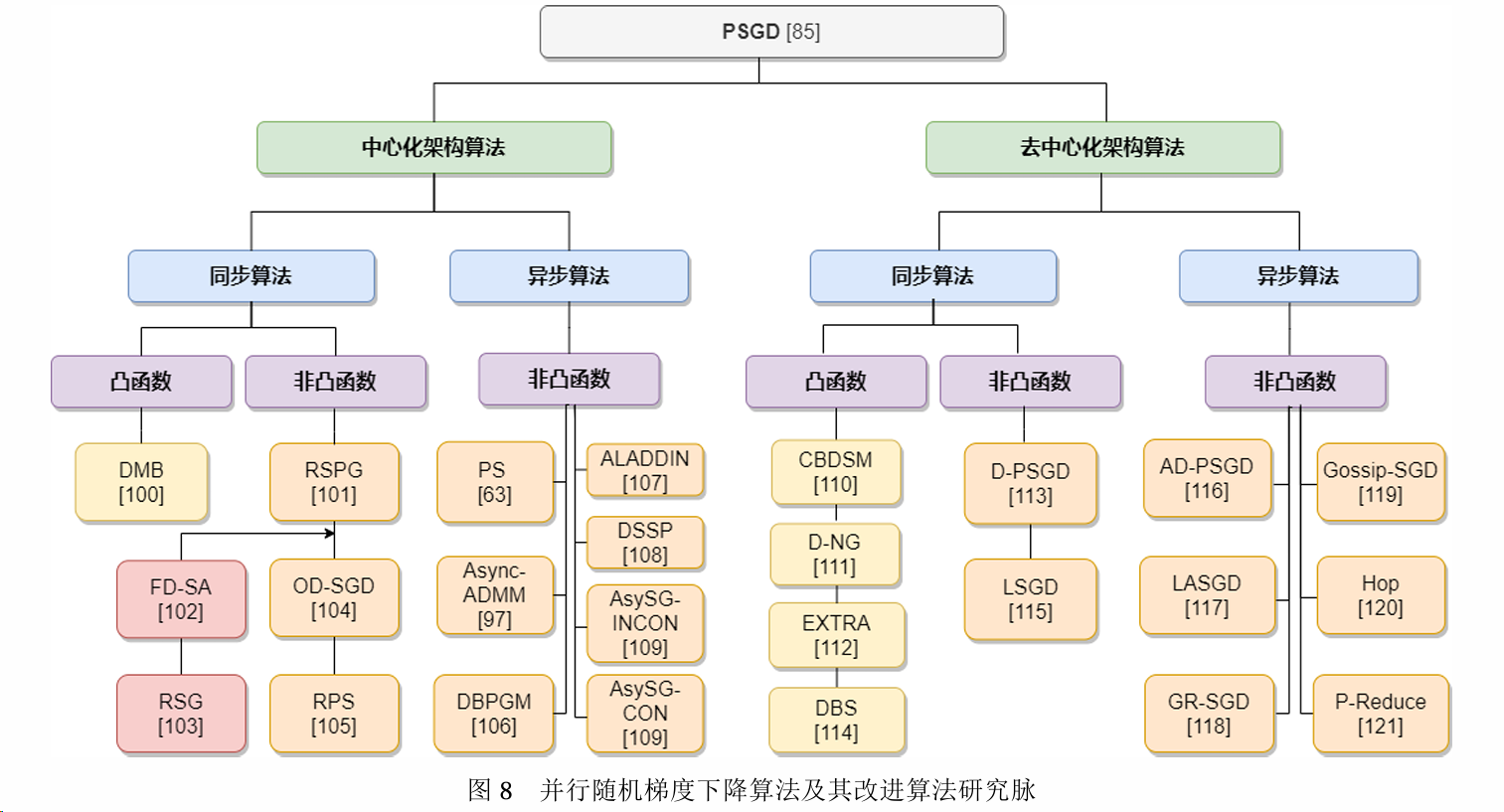
分布式系统的并行模式决定了对数据集和模型的拆分方式，影响了计算资源以及训练效果。而分布式训练系统的通信拓扑影响了其通信开销、资源利用率、训练精度等。不同的分布式系统架构产生了不同的通信方式，即分布式训练网络拓扑架构决定了通信方式。分布式训练系统的通信拓扑可以分为中心化架构和去中心化架构两大类，如下图所示：



中心化架构的分布式训练系统是具有一个中心主节点来协调各个计算节点的架构。典型实例是参数服务器(Parameter Server, PS)架构。在参数服务器架构中存在两种角色：计算节点(Worker)和服务器节点(Server)。计算节点主要负责如下操作：(1)基于其局部数据样本完成本地训练任务。(2)通过客户端接口与服务器节点进行通信，即从服务器节点处下载最新的全局模型参数，并将其自身的局部参数上传到服务器节点处。服务器节点作为参数服务器架构的核心部件主要完成如下操作：(1)接收并聚合各个计算节点发送来的局部梯度或模型参数等信息。(2)更新全局模型参数并返回至各个计算节点处。

为了缓解中心化架构的通信拥塞问题，研究人员提出了去中心化架并设计诸如全规约(All-Reduce)和 Gossip 等复杂通信机制。All-Reduce机制内计算节点间通过全规约(All-Reduce)操作实现信息交换，全体计算节点在无中心服务器节点的情况下进行通信。与参数服务器架构相比，全规约机制放弃了中心服务器节点，规避了服务器节点的通信拥塞问题，实现了去中心化，已广泛应用于各类分布式训练系统。然而，全规约机制也存在以下两点不足：一是由于缺乏中心服务器节点的全局统一调度与管理，需要引入大量的计算节点间的协调策略，从而导致通信开销仍是挑战。二是全规约机制不支持异步通信，从而一定程度上降低了训练系统信息同步方式的灵活性。Gossip 机制与全规约机制相比，该机制中各计算节点仅与邻居节点进行信息交互，从而加快通信。另外，为保证训练精度，该机制的挑战是如何保持模型参数的一致性，即“共识问题”(Consensus Problem)。

通信同步方式直接决定了模型参数及梯度等信息交互的过程，从而影响分布式算法的收敛性质。近些年在分布式训练优化算法领域有许多突破性成果，其中以并行随机梯度下降(Parallel Stochastic Gradient Descent, PSGD) 算法为代表。该算法的主要贡献是将经典的串行随机梯度下降算法实现了并行化，促进了分布式并行优化算法的发展。基于PSGD算法的中心化架构以及去中心化架构研究脉络如下图所示。



根据分布式系统的通信拓扑可以将PSGD算法划分为中心化架构算法和去中心化架构算法。二者均有相关的同步算法和异步算法的研究。典型的**同步算法是整体同步并行(Bulk Synchronous Parallel, BSP)算法**。在 BSP算法中，当一个计算节点完成当前迭代训练任务后，需要通过各种通信拓扑逻辑与其他计算节点同步模型参数或梯度等信息。然后，全部计算节点以相同的“起点”进入下一轮次迭代训练过程。该算法引入了一个全局同步障碍(Synchronization Barrier)。同步障碍要求那些处理能力较强且迭代速度快的计算节点在同步障碍处强制停止，并等待其他处理能力较弱且迭代速度慢的计算节点完成其当前轮次迭代训练任务后，系统才会执行下一轮次迭代训练任务。目的维护全局一致性。但是该算法存在拖累者问题。**异步算法各节点可按照自身迭代速度独立执行本地训练与参数更新。**当一个计算节点完成当前轮次梯度计算后，它将局部梯度发送至服务器并从服务器处获取当前全局参数。之后，继续进行下一轮次迭代而无需等待其他计算节点。异步通信虽然避免了计算节点间不必要的等待，但会导致梯度过时。**除此之外，还有关于同步算法和异步算法的平衡的研究。**研究人员提出了一种兼顾同步和异步通信的延时同步并行(Stale Synchronous Parallel, SSP) 算法。该算法的主要思想是控制迭代速度最快与最慢计 算节点之间的迭代间隔，算法要求两者迭代间隔不能超过预先设定的阈值。超过阈值就会停止工作等待其他节点，这个阈值的设定困难，太小会导致频繁停顿和资源闲置，太大容易陷入梯度过时。

基于中心化架构的同步并行随机梯度下降算法的关键迭代步骤如下：1. 各计算节点下载保存在服务器节点处的全局模型参数，并使用该全局模型参数进行梯度计算。2. 计算节点将各自的局部梯度上传到服务器节点处。3. 服务器节点聚合所有计算节点的梯度并更新全局模型参数。其中，关于凸目标函数算法，以分布式小批量 (Distributed Mini-Batch, DMB)算法为代表，该算法将串行在线预测算法转化为分布式算法。非凸目标函数算法有随机投影梯度(Randomized Stochastic Projected Gradient, RSPG)算法，这种算法的返回解稳定性较差。该作者在后续也提出了改良算法2-RSPG。

区别于同步算法，异步算法中各计算节点完成当前轮次迭代训练后无需等待其他计算节点，因而提升了计算资源的利用率。ALADDIN算法在服务器节点与计算节点之间采用非对称通信方式，解决服务器节点的通信瓶颈问题。除此之外还有自适应动态同步阈值算法

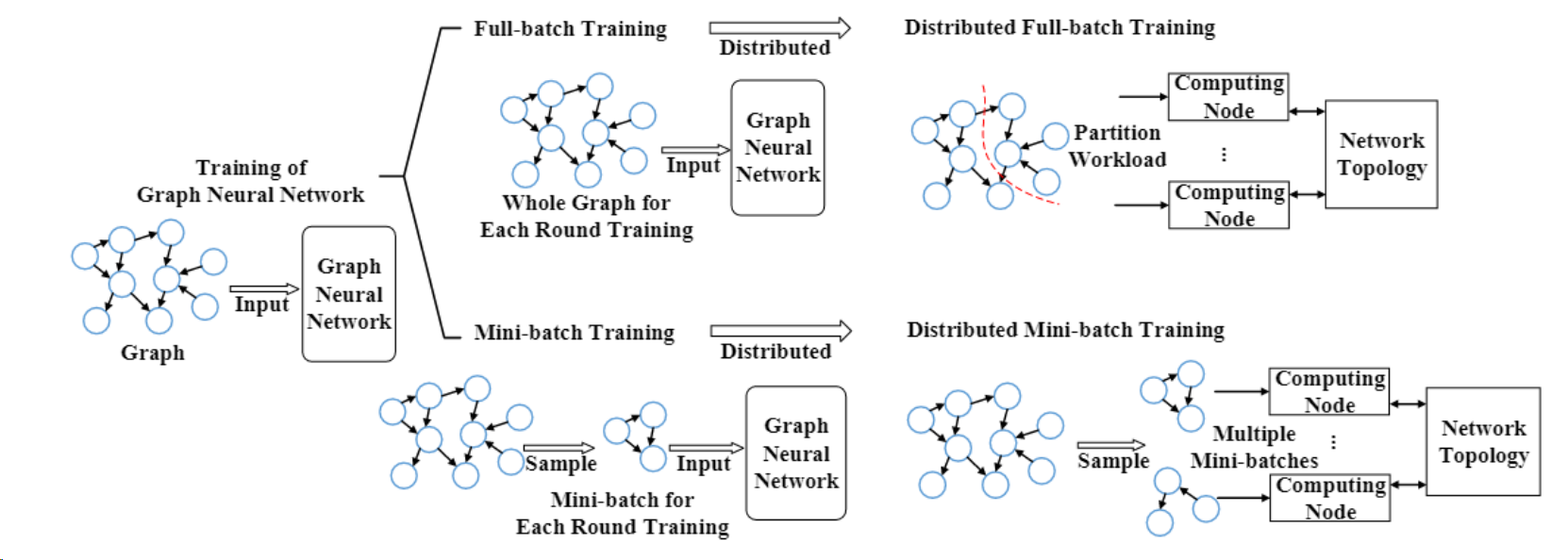
(Dynamic Stale Synchronous Parallel, DSSP)。算法在运行时自适应地调整各次迭代的同步阈值，以减少速度较快计算节点对全局模型参数同步的等待时间，从而增加迭代吞吐量并加快收敛。

与中心化架构算法相比，去中心化架构算法通常具有更多限制性的假设使得其收敛速度较差。为了解决这个问题，去中心化并行随机梯度下降(Decentralized Parallel Stochastic Gradient Descent, D-PSGD) 算法被提出。D-PSGD算法在迭代中采用固定数量的样本子集，后续有从小批量样本容量的角度出发提出的动态样本容量(Dynamic Batch Size, DBS)算法。

虽然D-PSGD算法在收敛速度方面与中心化架构算法具有相似的特性，但由于各个计算节点均需要同其所有邻居节点在当前迭代中进行信息同步，从而导致D-PSGD算法的通信开销仍然较高。为了克服上述问题，研究人员提出异步去中心化并行随机梯度下降(Asynchronous Decentralized Parallel Stochastic Gradient Descent, AD-PSGD)算法并应用

于非凸目标函数。

分布式GNN是加快GNN训练的流行解决方案。它试图通过向具有并行执行策略的计算系统添加更多计算资源或“节点”来加速整个计算过程。2019年提出的NeuGraph[44]是分布式GNN训练的第一个已发表的工作。之后有工作负载划分、传输规划、缓存策略等优化技术。GNN可以分为全批训练和小批量训练，全批训练用整个图形进行更新，小批量训练用图形中的部分顶点和边进行更新。



分布式全批训练是GNN全批训练的分布式实现。除了图分区之外，一个主要的区别是多个计算节点需要在更新模型参数之前同步梯度，以便计算节点之间的模型保持统一。因此，一轮分布式全批训练包括两个阶段：模型计算（正向传播+反向传播）和梯度同步。其主要采用工作负载划分方法，分割图以生成小工作负载，并将其移交给不同的计算节点以解决大量的计算需求和内存占用。

分布式全批量训练的工作流程概括为具有工作负载划分的协同计算。由于每一轮的计算都涉及到整个图形，因此计算节点需要在本地缓存它，从而导致高内存容量需求。此外，分布式全批量训练的通信量很大。在每一轮中，Aggregate函数都需要收集每个顶点的邻居的特征，这会导致大量的节点间通信请求，因为**图被划分并存储在不同的节点**上。考虑到通信基于不规则图形结构，分布式全批量训练的通信不规则性很高。通信的另一个特点是高度不确定性。生成通信请求的时间是不确定的，因为每个计算节点根据其自身计算过程中当前涉及的顶点发送通信请求。因此，分布式全批量训练的主要挑战是工作负载不平衡和大量传输。

分布式小批量训练通过同时处理多个小批量（每个计算节点一个）来并行化训练过程。可以由计算节点本身或其他设备（例如专门用于采样的另一个节点）对小批量进行采样。每个计算节点在其自己的小批量上执行前向传播和后向传播。然后，节点同步并累积梯度，并相应地更新模型参数。

分布式小批量训练的工作流被概括为具有周期同步的独立计算。主要传输内容是**从采样节点（或组件）发送到负责当前小批量的计算节点（或部件）**。因此，这些传输具有低不规则性和低不确定性，因为传输的方向和内容是确定性的。由于每一轮的计算只涉及小批量，因此它触发的通信量更少，需要的内存容量也更少。然而，额外的采样阶段可能会带来一些新的挑战。由于采样阶段的计算是不规则的，并且需要访问给定顶点的相邻信息的整个图，因此很可能会遇到采样性能不足的问题，导致后续计算节点（或组件）由于缺少输入而暂停，从而导致性能损失。

二者一个是采样，一个是对图进行划分。全批量保存的信息更多，但是负载更大，传输要求更高，小批量训练能缓解负载和传输要求，但是采用可能会导致性能损失。