SpanTrain：基于云边端异构设备的跨域分布式模型训练系统

王锦权[1,2]，刘旭昭[1,2]，廖晓坚[1,2]，肖利民[1,2]，霍志胜[1,2]，索珈顺[1,2]，李云潼[1,2]，沈润楠[1,2,3]，谢喜龙[1,2]，唐熙程[1,2]
1. 北京航空航天大学计算机学院，北京 100191；
2. 北京航空航天大学复杂关键软件环境全国重点实验室，北京 100191；
3. 北京航空航天大学沈元学院，北京 100191

摘要
目前，除云计算中心外，以物联网、固定或移动计算边缘为代表的边、端侧环境中也部署了大量的智能计算设备。将深度神经网络（DNN）模型的训练任务从云计算中心拓展到边、端侧，在新应用模式支持、数据隐私保护、训练成本控制等方面具有显著优势。现有的分布式模型训练系统大多针对同构设备设计，难以适应云边端异构计算环境。为此，设计了基于云边端异构设备的跨域分布式模型训练系统 SpanTrain，通过一种新颖的混合流水线并行机制，实现了云边端异构设备协同的高效 DNN 模型训练。典型云边端异构设备环境中的实验表明，与已有的训练系统相比，SpanTrain 带来了 1.17~3.15 倍的训练吞吐量提升，并将异构设备的资源利用率提升了 39%，这充分说明了 SpanTrain 在云边端异构设备中进行 DNN 训练的高效性。

关键词
云边端异构设备；分布式计算；模型训练；并行训练策略

中图分类号：TP183　　文献标志码：A　　doi：10.11959/j.issn.2096–0271.2025040

SpanTrain: a cross-domain distributed model training system for cloud-edge-end heterogeneous devices

WANG Jinquan[1,2], LIU Xuzhao[1,2], LIAO Xiaojian[1,2], XIAO Limin[1,2], HUO Zhisheng[1,2], SUO Jiashun[1,2], LI Yuntong[1,2], SHEN Runnan[1,2,3], XIE Xilong[1,2], TANG Xicheng[1,2]
1. School of Computer Science and Engineering, Beihang University, Beijing 100191, China
2. State Key Laboratory of Complex & Critical Software Environment, Beihang University, Beijing 100191, China
3. Shen Yuan Honors College, Beihang University, Beijing 100191, China

Abstract
Currently, in addition to cloud computing centers, the edge and end environments represented by the internet of things, fixed or mobile computing edges are also filled with a large number of intelligent computing devices. Expanding the deep neural network (DNN) training from cloud computing centers to the edge and end has significant advantages in aspects such as support for new application patterns, protection of data privacy, and control of training costs. Most existing distributed training systems are designed for homogeneous devices, and they are difficult to adapt to the heterogeneous computing environments of cloud-edge-end. For this reason, a cross-domain distributed training system

named SpanTrain, which is based on the heterogeneous devices of cloud, edge, and end, has been designed. Through a novel hybrid pipeline parallel mechanism, it realizes the efficient DNN model training with the collaboration of the heterogeneous devices of cloud, edge, and end. Moreover, experiments have been carried out in an environment containing typical heterogeneous devices. Experiments in typical cloud-edge-end heterogeneous environments demonstrate that SpanTrain achieves 1.17x~3.15x higher training throughput compared to state-of-the-art systems, while improving resource utilization of heterogeneous devices by 39%. These results validate the efficiency of SpanTrain for DNN training in cloud-edge-end heterogeneous environments.

Key words
cloud-edge-end heterogeneous device, distributed computing, DNN training, parallel training strategy

0 引言

近年来，深度神经网络（deep neural network，DNN）发展迅速，被广泛地应用于自然语言处理（natural language processing，NLP）[1-2]、计算机视觉（computer vision，CV）[3-4]、决策规划等任务中，极大地提升了上述任务的效率。基于DNN的NLP、CV等模型已经被广泛应用于智能制造、智慧交通、智能电网等云边端协同计算环境中，服务了柔性制造[5]、交通态势感知[6]、电网智能巡检[7]等多样化云边端协同任务。

云边端环境包含云计算中心、端侧设备以及靠近端侧设备的边缘服务器或小型数据中心。云边端环境包含大量异构算力，且能执行协同计算。云计算中心通常部署具有高算力的智能计算集群，例如A100、V100等GPU。边缘通常部署具有较高智能计算能力的GPGPU（通用GPU）和神经网络处理器（neural network processing unit，NPU），例如RTX 3090等芯片。端侧通常部署集成了数字信号处理（digital signal processing，DSP）、NPU或者高能效GPU的单片系统（system on chip，SoC）设备[8]。此外，云边端环境中互联网络的带宽达到了10~50 Gbit/s，传输时延降低到了50~150 ms[9]。不断提升的算力和跨域网络为云边端跨域分布式训练提供了良好的硬件基础。

云边端跨域分布式训练具有优秀前景，Google研发的Gemini[10]和OpenAI研发的o1[11]等模型均应用了跨域分布式训练技术[12-13]。首先，云边端跨域训练能充分利用全局算力资源，加速训练过程，避免算力资源的闲置。云计算中心运行着密集的训练任务和卸载到云计算中心的推理任务，计算负载较高；而边缘和端侧的计算资源利用呈现潮汐式趋势，计算资源闲置严重；跨域分布式DNN训练可以充分利用这些闲置资源[14]。其次，云边端协同训练能减少训练数据的传输开销，进一步加速训练过程。在实际场景中，大量数据从边缘服务器和端设备中产生，数据在边缘服务器和端侧设备完成收集、清洗后上传到云端进行DNN训练，造成了大量的传输开销，损害了其他服务的服务质量（quality of service，QoS）与数据隐私；跨域分布式DNN训练可以降低传输开销，保护关键服务的QoS和数据隐私[15]。最后，云边端跨域训练能加速模型的迭代。传统的云侧训练、边端侧推理的应用模式反馈链路长、模型迭代速度低；跨域分布式DNN训练可实现近数据微调，加快DNN模型的训练进

程，提高服务质量和模型迭代速度[16]。

然而，在云边端环境中实现高效的跨域分布式DNN训练仍然存在诸多挑战。云边端设备的计算性能、互联网络呈现异构性，简单地将现有DNN训练框架应用至云边端环境无法充分发挥各设备的性能，甚至会导致训练速度下降、资源利用率不足。例如，DeepSpeed使用平均划分的3D并行实现DNN模型的分布式训练；当拓展到云边端跨域环境中时，训练速度从云计算中心的100 Sample/s左右下降到3~8 Sample/s，且异构计算资源的利用率不足30%。具体来说，将DNN训练拓展到云边端跨域计算环境中时，将面临以下问题。

• 如何在云边端域间的异构设备中合理地划分模型训练任务？在云边端跨域环境中，根据网络连接关系，不同的异构设备被天然地分成了多个不同的计算域。这些计算域之间具有不同的计算能力和网络互联速度。算网资源的异构性将会影响模型的计算效率和通信效率，从而影响模型训练的速度。因此，云边端域间合理的模型训练任务划分成为亟待解决的问题。

• 如何在云边端域内的异构设备中合理地划分模型训练任务？在云边端跨域环境中，不同的计算域内部同样会包含多种具有不同计算能力的异构计算设备。DeepSpeed等计算框架将在这些异构设备中平均分配计算任务，无法发挥域内异构计算设备的计算性能，且容易出现瓶颈节点。因此，云边端域内合理的模型训练任务划分成为亟待解决的问题。

为了应对上述挑战，本文提出了SpanTrain，这是一个基于云边端异构设备的跨域分布式模型训练系统。SpanTrain可以在云边端的异构设备中执行高效的跨域分布式DNN训练。首先，SpanTrain针对云边端异构设备实现了跨域并行策略生成技术，结合评估域间计算、通信性能，生成跨域并行策略，在不同的计算域之间合理地拆分DNN模型。其次，SpanTrain实现了一种域内非对称多维并行技术，通过非对称的张量分配技术，在高效适配跨域并行策略的同时最大化域内的计算效率。本文在具有代表性的云边端异构环境中实现了SpanTrain，评估表明，SpanTrain带来了1.17~3.15倍的训练吞吐量提升，对异构设备的资源利用率提升了39%。

总的来说，本文的主要贡献如下。

• 本文介绍了将DNN训练扩展到云边端环境的潜在优势，此外，本文还分析了在云边端异构设备中进行DNN训练面临的挑战。

• 本文提出了一种名为SpanTrain的基于云边端异构设备的跨域分布式模型训练系统，用于在云边端异构设备上进行高效的跨域分布式DNN训练。

• 本文基于PyTorch实现了SpanTrain，并通过对比实验检验了SpanTrain的有效性。实验结果表明，SpanTrain可以显著提升云边端DNN的训练速度和资源利用率。

1 研究背景

本节将从云边端计算环境、云边端异构硬件训练策略两个方面介绍研究背景，并阐述现有云边端计算环境中的软硬件配置、常见模型训练策略等。

1.1 云边端计算环境

云边端计算环境已经成为新一代信息基础设施，服务了国家关键领域数智化转

型，在智能制造、智慧交通、智能电网等领域发挥了重要作用。目前，云边端计算环境中异构设备的计算能力已经得到了长足的进步。

在智能制造、智能电网等云边端协同场景中，需要构建由数百台计算、存储服务器组成的云计算中心；在各个工厂、变电站中需要部署由10~20台边缘服务器组成的计算集群，具备RTX3090级别的算力；端侧设备主要包含智能摄像设备、巡检机器人、机械臂等，其中加装了具有一定计算能力的Jetson T4、寒武纪等芯片。云边使用高速广域网（wide area network，WAN）进行连接，互联带宽为1 Gbit/s左右，边端、边边使用高速局域网（local area network，LAN）进行连接，互联带宽为10~50 Gbit/s。

但是，笔者在云边端计算环境中发现了两种计算资源未得到最佳利用的现象。如**图1**所示，笔者采样了24 h内云计算集群和边缘服务器的负载情况。结果显示，云计算机群的负载处于持续性高位，平均负载达到了78%，运行了大量的智算类、服务类、管理类应用和系统。而边缘服务器仅持续运行一定量的推理任务，平均负载为20%~30%。

此外，在云边端计算环境中，大多数应用遵循端侧收集数据、边侧清洗数据、云侧训练模型的工作模式。大量的高清音视频数据需要上传到云计算中心，进行集中的存储以进行模型的训练，占用了大量的云边上行带宽，在高负载情况下会影响云计算中心管理命令下发、模型更新等操作，严重降低了云边端协同应用的QoS。

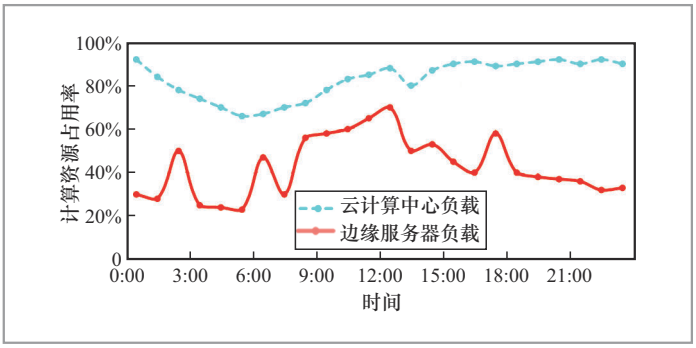


图1 24 h云计算集群与边缘服务器负载情况

将DNN训练从云计算中心拓展到云边端异构计算设备，是解决上述两个问题的有效手段。针对上述问题，本文设计了基于云边端异构设备的跨域分布式模型训练系统SpanTrain。

1.2 云边端异构硬件训练策略

云边端异构硬件实施分布式训练主要有两种思路：基于数据并行（data parallelism，DP）的混合数据并行（hyper data parallelism，HDP）和基于流水线并行（pipeline parallelism，PP）的混合流水线并行（hyper pipeline parallelism，HPP）。

在DP中，各工作节点（Worker）维护完整的模型副本，模型输入（即训练数据）被切分并分配至各工作节点。同时，Worker之间通过参数服务器（parameter server，PS）或者AllReduce等集合通信库（collective communication library，CCL）原语定期同步梯度。DP的简单性为其带来了良好的拓展性，可以支持一定规模的跨域分布式模型训练；但是，DP存在显存占用大、资源利用率低、通信规模大等问题。**图2**展示了云边端环境中不同模型采用数据并行进行模型训练的通信、计算占比。不难发现，由于云边端计算环境中网络连接速度无法与云计算中心相提并论，将DNN模型训练拓展到云边端异构设备中并采用DP进行模型训练时，通信开销将会主导整个

训练过程。HDP 训练方法（如 Hetpipe）基于 DP 对设备进行分组，在组间采用 DP，组内采用 PP。然而，Hetpipe[17]针对同构设备设计，难以适应云边端异构环境。

PP 是另一种得到广泛应用的训练方式。在 PP 中，DNN 模型被划分为多个阶段（Stage），各 Stage 被映射到一个单独的处理进程，并通过流水线的形式并行执行前向传播和后向传播。通常情况下，各 Stage 工作负载的大致相同，确保最高的并行性。如**图 3** 所示，对于语言模型（如 Bert、GPT 等）和经过适当切分的模型，采用 PP 进行训练时的吞吐量要显著高于 DP，这是因为相比于 DP，PP 提供了更好的通信性能。但是，在云边端计算环境中，不合适的 Stage 划分将会给 PP 带来极大的通信开销。根据文献[15]所述，由于 Stage 划分过多，异构计算硬件计算能力不统一，不合适的 Stage 划分将带来高达 24 倍于计算时延的通信时延。在 PP 的基础上拓展出了 Asteroid[15]等 HPP 训练方法，HPP 对设备进行分组，通过组间采用 PP、组内采用 DP 的方式实现 DNN 的分布式训练，但是 HPP 大多针对同构设备和均衡网络设备，难以直接在云边端环境中使用。

基于上述原因，本文选择 HPP 作为 SpanTrain 的基础计算策略，并通过精心设计的域内并行策略、域间并行策略最大化资源利用率，提升模型训练性能。

1.3 相关工作

为了实现基于云边端异构硬件的跨域分布式训练，工业界和学术界已经开展了许多研究。Zhu 等人[18]的 PockEngine 为嵌入式设备微调大型模型提供了解决方案，通过稀疏反向传播和模型编译，优化训练图，降低计算开销并保持模型质量。Jiang

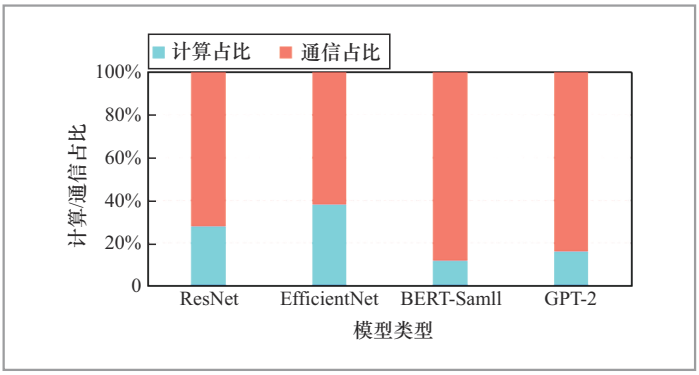


图 2 云边端环境中不同模型采用数据并行进行模型训练的通信、计算占比

等人[19]开发的 MNN（mobile neural network）引擎针对异构设备进行算子优化，支持轻量级和 IoT 设备运行模型，并兼容多种后端。Patil 等人[20]提出的 Poet 系统利用重计算和分页机制减少内存消耗，并设计了能量优化的 MILP 规划器，实现微小设备上的模型微调。Wang 等人[21]的 AtRec 技术通过识别和优化 CPU 中的耗时算子，消除冗余计算，提升内存访问效率。Zhang 等人提出的 2PGraph[22]和 AGL[23]技术分别针对图神经网络训练中的内存访问延迟和消息传递模式进行优化。Mao 等人[24]的 MoDNN 系统将 DNN 模型划分到多个移动设备，降低计算成本和内存使用。Zeng 等人[25]的 CoEdge 系统根据设备计算能力和网络条件动态划分 DNN 工作负载，

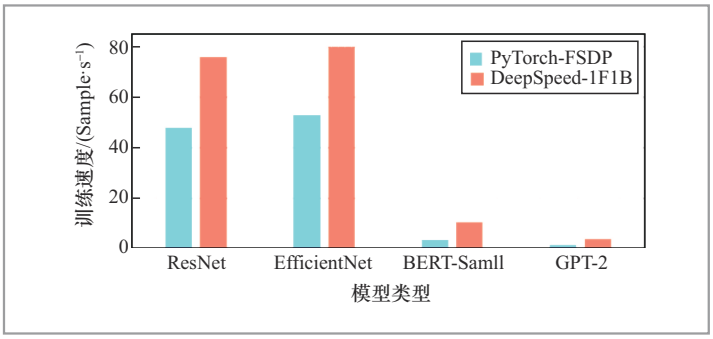


图 3 使用 PyTorch 和 DeepSpeed 在云边端计算环境中的进行模型训练的吞吐率

协同多个边缘设备。Zhao等人[26]的DeepThings技术采用文件传输协议（file transfer protocol，FTP）和分布式工作窃取方法，实现IoT设备和边缘集群的协同。Hao等人[27]的EDDL系统设计了动态数据分发和自适应通信机制，加速边缘模型训练。Yang等人[28]的AutoSF技术适应时变边缘资源，优化聚合结构和频率。Chen等人[29]的FTPipeHD技术动态优化分区点，并提出权重重新分配方法，解决训练期间的故障问题。这些研究共同推动了在资源受限的异构设备上高效运行和训练深度学习模型的技术发展。

但是，这些方法没有充分考虑云边端域间、域内的计算特征，无法在云边端不同的计算域之间合理地分配模型计算任务，导致无法最大化模型的训练性能和云边端异构资源的利用效率。

2 SpanTrain方法介绍

针对上述关键问题和挑战，笔者设计并实现了基于云边端异构硬件的跨域分布式训练系统SpanTrain。SpanTrain包含2个重要组件，分别是云边端跨域模型拆分组件和域内非对称多维并行组件。本节将详细介绍SpanTrain的运行流程以及各个组件的工作原理。

2.1 SpanTrain概览

图4展示了SpanTrain的工作流程和生成的并行方案。SpanTrain在模型训练启动前，通过运行时间评估进行云边端跨域模型拆分，并基于拆分结果生成域内非对称多维并行策略。首先，SpanTrain运行云边端跨域模型拆分组件，分析模型结构，构建运行时间模型，结合云边端计算域的计算能力和通信能力对模型进行拆分，得到跨域模型拆分结果。然后，根据域间模型拆分结果，运行域内非对称多维并行组件，根据域内异构计算节点不同的计算能力、显存大小生成非对称的PP、DP、TP并行策略，以充分利用域内的异构计算资源。

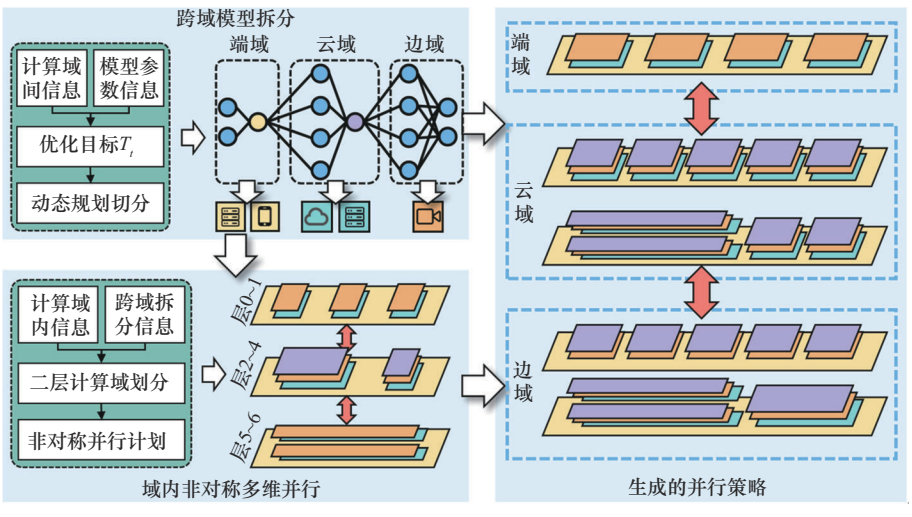


图4 SpanTrain方法总体架构

SpanTrain在运行上述组件后，生成如**图4**所示的并行策略。在云、边、端各计算域之间，SpanTrain采用流水线并行的方式编排模型，通过合理的模型跨域拆分协调不同计算域的运行速度，在各个计算域之间形成混合流水线并行结构。计算域内，SpanTrain则根据异构计算设备的计算能力，采用不同的非对称多维并行训练策略编排模型训练。如在云域中，SpanTrain首先将模型按层切分，实现均衡的流水线并行；在云域的第二个流水线阶段，进一步将张量的第一维、第二维进行非对称切分，实现了非对称的张量并行和数据并行。而在边域中，SpanTrain则非对称地划分了模型，2层模型为第一个阶段，3层模型为第二个阶段。

2.2 云边端跨域模型拆分

为了在云边端各计算域之间合理地划分模型，SpanTrain首先对模型训练运行时间进行建模，之后通过线性规划算法将不同的模型层放置于最合适的云边端计算域中，最终形成云边端跨域的混合流水线并行结构。

如**图5**所示，SpanTrain基于HPP的模型训练主要包含3个过程，分别是流水线填充阶段、流水线运行阶段和权重同步阶段。SpanTrain将每一个计算域d内这3个阶段所用的时间定义为$T_{w,d}$、$T_{r,d}$和$T_{s,d}$，并将最终的优化目标定义为：

$$T_t = \min \max \prod_{d \in D} \left(T_{w,d} + T_{r,d} + T_{s,d} \right) \quad (1)$$

其中，D代表所有的计算域形成的集合，SpanTrain的目标是缩短任意一个计算域内，流水线填充阶段运行时间$T_{w,d}$、流水线运行阶段运行时间$T_{r,d}$和权重同步阶段运行时间$T_{s,d}$的和，使总体运行时间T_t尽可能小，缩短每一轮训练所需的时间。

对于任意一个计算域d_k，SpanTrain使用计算域$d_1 \sim d_{i-1}$中的模型层的前向传播时间、激活值的传输时间估算其流水线填充阶段的运行时间$T_{w,d}$：

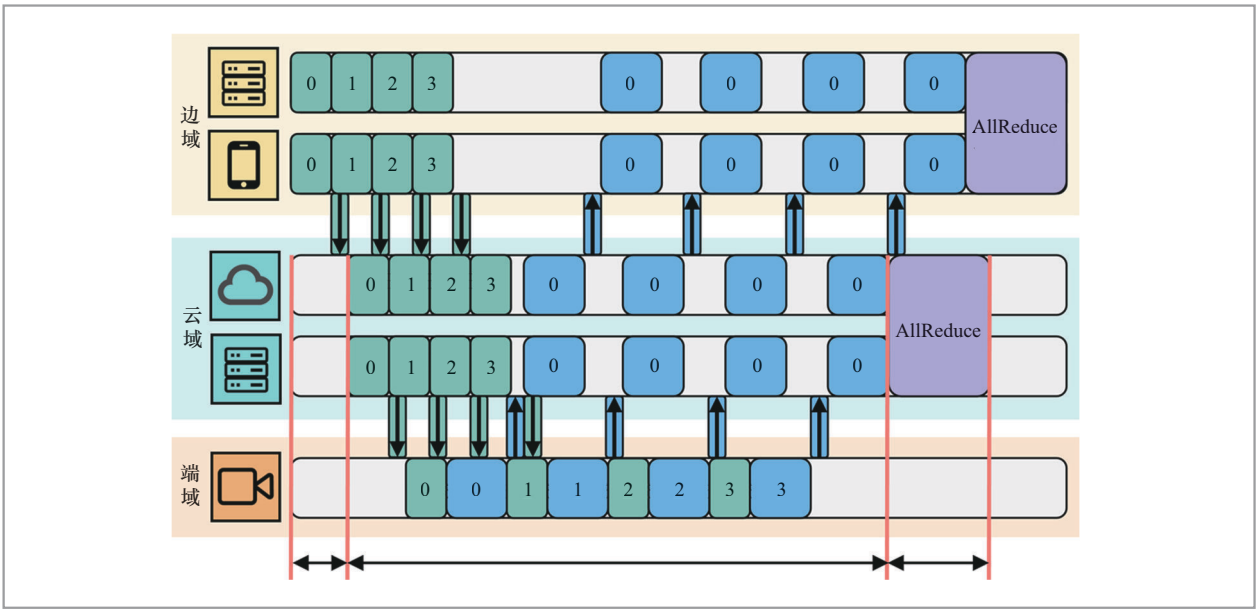


图5 SpanTrain中HPP训练的3个主要阶段

$$T_{w,d_k}=\sum_{d_j\in\{d_1,\cdots,d_{k-1}\}}b_j\left(\sum_{l\in d_j}\left(\frac{C(l)}{t_f(l,d_j)}\right)+\frac{a_j}{\beta_{n,m}}\right) \quad (2)$$

其中，b_j代表计算域内d_j内的微批次（micro batch）的大小，l代表分配在计算域d_j中的所有模型层，$C(l)$代表模型层l的计算量，$t_f(l,d_j)$代表模型层l模型在计算域d_k中的前向传播速度，a_j代表计算域d_j最后一个模型层数据的单位激活值大小，$\beta_{n,m}$代表计算域d_j中的设备n到计算域d_{j+1}中的设备m的最大通信带宽。通过上述计算式，SpanTrain计算了计算域d_1~d_{k-1}中各模型层的计算速度和激活值的传输时间，从而得出计算域d_k流水线填充阶段的耗时。

云边端计算环境中不平衡的网络带宽会导致计算和通信的重叠率下降，使流水线运行阶段的运行时间$T_{r,d}$难以估算，SpanTrain通过寻找关键计算域的方式解决这个问题。如**图5**所示，在端域中前向传播、反向传播紧密排布。SpanTrain将这样的计算域作为关键计算域。SpanTrain计算端域流水线运行阶段的运行时间$T_{r,d}$，并以此为依据计算边域和云域的流水线运行阶段的运行时间。具体来说，SpanTrain中将关键计算域流水线运行阶段的运行时间定义如下：

$$T_{r,d_c}=nb_c\sum_{l\in d_c}\frac{C(l)}{t_f(l,d_c)}+\frac{C(l)}{+t_b(l,d_c)} \quad (3)$$

其中，n是关键计算域中微批次的个数，b_c是关键计算域中微批次的大小，l代表分配在计算域d_c中的所有模型层，$C(l)$代表模型层l的计算量，$t_b(l,d_c)$代表模型层l模型在关键计算域d_c中的前向传播速度。通过计算式（3），SpanTrain计算了关键计算域中运行所有微批次前向传播和反向传播的时间和，获得了关键计算域中流水线运行阶段的运行时间，并以这个时间更新其前序计算域的运行时间。

对于计算域d_k来说，如果d_k是关键计算域前向传播的后序计算域，因为总耗时最长的计算域将不可能是d_k，SpanTrain将不会计算其流水线运行阶段的运行时间。反之，SpanTrain通过式（4）计算其对应的流水线运行阶段运行时间$T_{r,d}$：

$$T_{r,d_k}=T_{w,d_c}+T_{r,d_c}+\sum_{d_j\in\{d_{c-1},\cdots,d_k\}}b_j\left(\sum_{l\in d_j}\left(\frac{C(l)}{t_b(l,d_j)}\right)+\frac{g_j}{\beta_{n,m}}\right)-T_{w,d_k} \quad (4)$$

其中，b_j代表计算域内d_j内的微批次的大小，l代表分配在计算域d_j中的所有模型层，$C(l)$代表模型层l的计算量，$t_b(l,d_j)$代表模型层l模型在计算域d_k中的反向传播速度，g_j代表计算域d_{j-1}最后一个模型层数据的单位梯度大小，$\beta_{n,m}$代表计算域d_j中的设备n到计算域d_{j-1}中的设备m的最大通信带宽。通过式（4），SpanTrain估算了计算域d_{c-1}~d_k中各模型层的计算速度和梯度值的传输时间，从而得出了计算域d_k流水线运行阶段耗时。

如**图5**所示，在权重同步阶段计算域内的异构设备会通过AllReduce与计算域内的其他异构设备进行通信，且不与其他计算域内的节点进行通信。SpanTrain使用式（5）计算权重同步阶段的用时：

$$T_{s,d_k}=\frac{2(|d_k|-1)\cdot\sum_{l\in d_j}|w_l|}{|d_k|\cdot\min_{n,m\in d_k}\beta_{n,m}} \quad (5)$$

其中，$|d_k|$代表计算域d_k内的设备总数，w_l代表模型层l的张量传输大小，$\beta_{n,m}$代表计算域d_k内两个异构设备n、m之间的互联带宽。值得注意的是，SpanTrain虽然提供了域内非对称多维并行，实现了计算域内张量的不均衡分配，但是总的通信量是相同的，为了简单地预估权重同步阶段的执行时间，SpanTrain仍然使用标准的

AllReduce通信时间计算式得到权重同步阶段的执行时间。通过上述计算式，SpanTrain计算了在计算域内进行AllReduce权重同步操作所消耗的时间，从而得到权重同步阶段的消耗时间。综上所述，SpanTrain完成了优化目标的构建，得到了对计算过程耗时的完整建模。

接下来，SpanTrain使用动态规划算法对上述优化问题进行求解，图6所示为SpanTrain求解的动态规划问题。首先，对于模型层l，SpanTrain尝试将其放置在不同计算域d_k内，并寻找这种放置下的关键计算域。其次，计算将模型层l放置在不同计算域d_k时的优化目标T_t'，并将模型拆分策略更新为T_t'最小时对应的模型拆分策略。通过有限次数的迭代计算，SpanTrain能获得最佳跨域模型拆分计划。

2.3 域内非对称多维并行

数据中心内的DNN训练通常会采用同构的设备，对数据、张量进行均等划分，实现DP和TP，这样做可以确保每一个计算设备的性能被充分发挥。但是，当DNN训练被拓展到云边端环境中时，异构计算设备具有不同的计算能力，传统的TP、PP无法最大化利用这些异构计算资源。为了进一步提升计算域内的资源利用效率和计算性能，SpanTrain实现了域内非对称多维并行策略。首先，SpanTrain在计算域内根据设备的计算能力进行二次聚合，形成二层计算域；其次，计算二层计算域之间的性能比例，并以此为依据在张量的第一维和第二维进行非对称拆分，并将它们划分到不同的异构设备中。

SpanTrain根据设备的计算能力、显存大小，对计算域内异构设备进行二次设备聚合，形成二层计算域。然后，在一个计算域d_k内，SpanTrain会分两种情况进行处理，分别是进行二层计算域划分的计算域与没有进行二层计算域划分的计算域。

对于进行二层计算域划分的计算域来说，SpanTrain实现了非对称的PP策略。首先，SpanTrain进行流水线的划分，提供计算域内的PP并行能力。SpanTrain获取二层计算域的异构设备计算性能，分别是前向传播时间$t_{f,s}$、反向传播时间$t_{b,s}$和显存大小M_s，并计算二层计算域之间的计算性能比例；在显存空间允许的情况下，按照计算性能的比例划分模型层到不同的二层计算域内。其次，SpanTrain将在二层计算域内采用标准的TP和PP策略，最大化异构计算设备的性能。如图7（a）所示，5台计算设备被划分为2个二层计算域，SpanTrain在此基础上进行PP划分的示例。SpanTrain评估各个二层计算域之

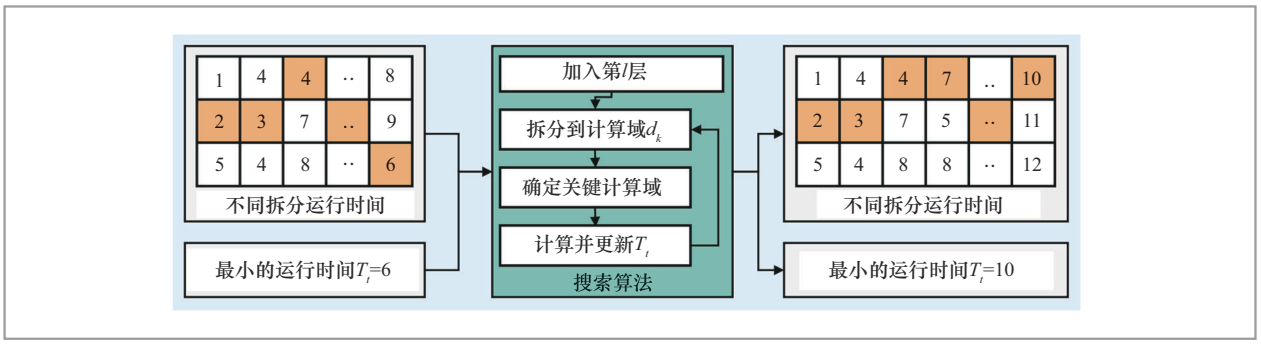


图6 SpanTrain求解的动态规划问题

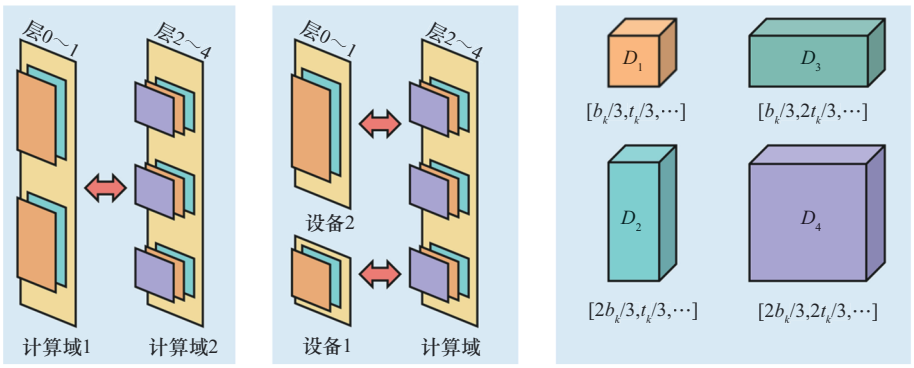


图7 域内非对称多维并行示意图

间计算性能比例为2∶3，然后通过模型编译技术获取了分配到当前计算域内的模型的计算量，并按照二层计算域计算性能的比例进行了划分，充分发挥了不同异构计算设备的计算性能。如图7（b）所示，5台设备包含1个二层计算域，剩余2台设备的计算性能差距较大，SpanTrain优先将尽可能多的模型层划分到二层计算域内，触发标准的DP、TP策略；然后将剩余模型层划分到其他两台设备中，并且在这两台设备中采取非对称的DP和TP策略。

针对没有进行二层计算域划分的计算域，SpanTrain在整个计算域内采取非对称的DP和TP策略。具体来说，SpanTrain通过异构设备算网能力评估组件获取异构计算设备性能，即前向传播时间$t_{f,i}$、反向传播时间$t_{b,i}$和显存大小M_i；通过微批次非平衡调度组件获取当前计算域d_k内的微批次大小b_k。然后，SpanTrain计算每个异构计算设备应该分配张量的比例，SpanTrain对齐各个异构计算设备的计算时间和显存开销，避免出现瓶颈节点，确保所有节点的运算在同时完成。图7（c）展示了在域内进行非对称DP、TP划分的一个例子。当计算域内d_k异构设备的计算能力满足1∶2∶2∶4，那么对于一个$[b_k, t_k, \cdots]$维的张量，SpanTrain将会将其划分为$[b_k/3,t_k/3,\cdots]$、$[2b_k/3,t_k/3,\cdots]$、$[b_k/3,2t_k/3,\cdots]$、$[2b_k/3,2t_k/3,\cdots]$4个张量，并将它们分布在这4台设备中。

3 性能实验

本节从实验设置、模型训练时间对比、异构资源利用率对比3个方面，阐述SpanTrain在跨域分布式训练中的性能表现。

3.1 实验设置

（1）模型与数据集

本文使用4种典型的、具有代表性的模型进行实验，分别为经典的CV模型ResNet101[30]、EfficientNet-B1[31]和两个NLP模型BERT-Small[32]和GPT-2[33]。在CV数据集方面，本文使用Cifar-10数据集进行测试，并将输入数据的张量维度统一为3×224×224；在NLP模型数据集方面，本文使用oscar-en-10k数据集进行测试，

并将输入数据集的大小调整为32×512。

（2）云边端环境设置

本文使用10台设备模拟了一个云边端异构计算环境，具体的配置信息见**表1**。其中，云计算域与边计算域之间的互联带宽为100 Mbit/s，边端计算域之间的互联带宽为1 Gbit/s。

（3）对比方法

本文选择了两个具有代表性的方法进行对比，分别是PyTorch和DeepSpeed。其中，PyTorch使用了完全分片数据并行（FSDP）特性，DeepSpeed使用了基于一次前向传播-一次后向传播（1F1B）的流水线并行特性。

表1 云边端异构计算集群模拟配置

计算域	设备名称	CPU配置	内存配置/GB	显卡配置
云	设备A	22vCPU	32	4×4090
	设备B	22vCPU	32	4×4090
	设备C	22vCPU	32	4×4090
边	设备D	12vCPU	32	2×3090
	设备E	12vCPU	32	2×3090
	设备F	12vCPU	32	2×3090
端	设备G	1CPU	16	1×3090
	设备H	1CPU	16	1×3090
	设备I	1CPU	16	1×3090
	设备J	1CPU	16	1×3090

3.2 模型训练时间对比

为了充分验证SpanTrain的有效性，本文将SpanTrain与PyTorch、DeepSpeed进行了对比。**图8**所示为ResNet、EfficientNet、BERT-Small、GPT-2 4种模型分别使用PyTorch、DeepSpeed与SpanTrain进行训练的速度。不难发现，在云边端异构设备中，SpanTrain的速度都明显优于其他方法，带来了26%~47%的性能提升。

具体来说，和PyTorch相比，SpanTrain在训练ResNet、EfficientNet、BERT-Small、GPT-2时，分别带来了2.11倍、3.15倍、1.84倍和1.98倍的速度提升。这主要有两方面的原因：首先，PyTorch中基于FDSP的数据并行没有充分地考虑节点的异构性，在各节点间平均分配数据，造成各节点训练时长不统一，需要等待其他节点完成训练，才可以进行参数同步；其次，在数据并行中计算与通信的重叠度较低，在所有节点的训练完成后，基于AllReduce的参数同步操作消耗了大量的时间。与DeepSpeed相比，SpanTrain在训练ResNet、EfficientNet、BERT-Small、GPT-2时，分别带来了1.57倍、2.22倍、1.17倍、1.38倍的速度提升。这主要因为DeepSpeed是针对数据中心的同构集群设计的，导致流水线的划分不能良好地匹配异构设备的计算能力与云边端环境中的网络差异。

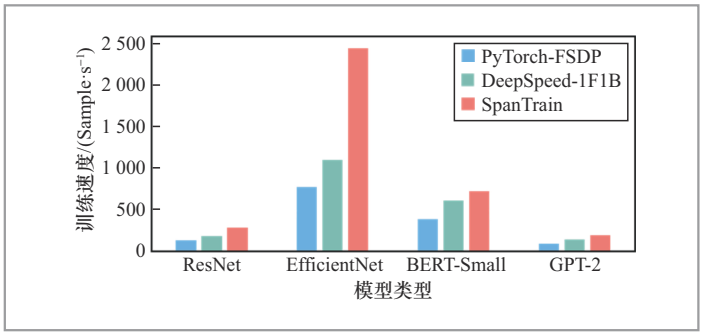


图8 不同模型使用不同方法进行云边端跨域分布式训练的训练速度

3.3 异构资源利用率对比

为了充分验证SpanTrain的资源使用

效率，本文将SpanTrain与PyTorch、DeepSpeed进行了对比。图9所示为ResNet、EfficientNet、BERT-Small、GPT-2 4种模型，分别使用PyTorch、DeepSpeed与SpanTrain进行训练时的计算资源利用效率。不难发现，在云边端异构设备中，SpanTrain的异构资源利用效率都明显优于其他方法，带来了27%~44%的资源利用提升。

具体来说，与PyTorch相比，SpanTrain在训练ResNet、EfficientNet、BERT-Small、GPT-2时，分别带来了1.92倍、2.17倍、2.45倍和3.3倍的资源利用率提升。这主要由于SpanTrain采用了HPP的并行方式，相比于PyTorch的FSDP极大地缩短了通信时间，提升了计算密度，进而提升了资源的利用效率。与DeepSpeed相比，SpanTrain在训练ResNet、EfficientNet、BERT-Small、GPT-2时，分别带来了1.44倍、1.27倍、1.41倍、1.44倍的资源利用率提升。这主要是因为SpanTrain结合了云边端计算域间和域内算网性能差异，引入了跨域模型拆分和非对称多维并行机制，相比于DeepSpeed实现了更加紧凑的流水线，因此得到了更优的资源使用效率。

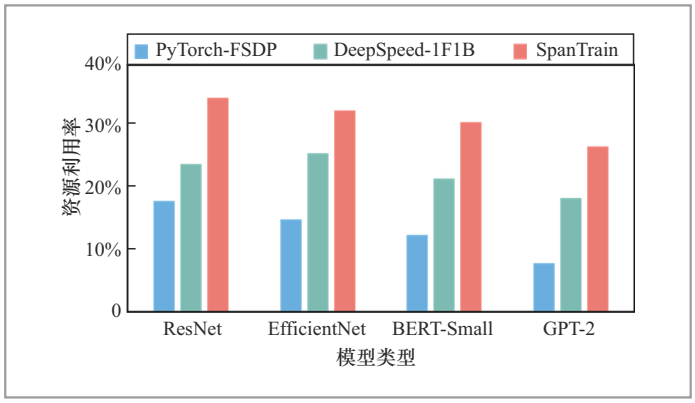


图9 不同模型使用不同方法进行云边端跨域分布式训练的计算资源利用率

4 结束语

云边端协同模型训练在新应用模式支持、数据隐私保护、训练成本控制等方面具有显著优势。针对云边端协同计算环境中的异构算网资源，本文设计了一种基于云边端异构设备的跨域分布式模型训练系统SpanTrain，通过云边端跨域模型拆分、域内非对称多维并行，实现了高效的混合流水线并行，支持云边端协同的高效DNN模型训练。

后续可持续开展下面两个方面的研究：首先，SpanTrain可结合细粒度的通信、计算-通信重叠、AllReduce异步化、梯度压缩等通信优化方法，进一步提升云边端协同环境中模型训练的效率；其次，可以进一步研究不同模型（如语音、推荐、图神经网络等）的算子特征和并行需求，探索不同模型在云边端环境中进行模型训练的可行性和技术挑战。

参考文献：

[1] BROWN B T, MANN B, RYDER N, et al. Language models are few-shot learners[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2020: 1877-1901.

[2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2017.

[3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB]. arXiv preprint, 2014, arXiv: 1409.1556.

[4] SOHL-DICKSTEIN J, WEISS E A,

MAHESWARANATHAN N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//Proceedings of the 32nd International Conference on International Conference on Machine Learning. New York: ACM, 2015: 2256-2265.

[5] DOU R L, ZHUANG G Y, LIU X, et al. Potential of AI for service performance of manufacturers: Analytical and empirical insights[J]. Advanced Engineering Informatics, 2024, 60: 102383.

[6] WATANABE Y, SATO K, TAKADA H. DynamicMap2.0: a traffic data management platform leveraging clouds, edges and embedded systems[J]. International Journal of Intelligent Transportation Systems Research, 2020, 18(1): 77-89.

[7] MELONI A, PEGORARO P A, ATZORI L, et al. Cloud-based IoT solution for state estimation in smart grids: exploiting virtualization and edge-intelligence technologies[J]. Computer Networks, 2018, 130: 156-165.

[8] IDC CORPORATE. Artificial intelligence computing power migrating to the edge will drive the steady growth of edge servers in China[Z]. 2024.

[9] IDC CORPORATE. China edge cloud market tracking research[Z]. 2024.

[10] GEMINI TEAM, GOOGLE. Gemini 1.5: unlocking multimodal understanding across millions of tokens of context[Z]. 2024.

[11] OPENAI. OpenAI o1 system card[Z]. 2024.

[12] PATEL D, NISHBALL D, ONTIVEROS J E. Multi-datacenter training: OpenAI's ambitious plan to beat Google's infrastructure[Z]. 2024.

[13] PALAK, GANDHI R, TANDON K, et al. Improving training time and GPU utilization in geo-distributed language model training[EB]. arXiv preprint, 2024, arXiv: 2411.14458.

[14] XU D L, XU M W, LOU C H, et al. SoC-Flow: efficient and scalable DNN training on SoC-clustered edge servers[C]//Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2024: 368-385.

[15] YE S Y, ZENG L K, CHU X W, et al. Asteroid: resource-efficient hybrid pipeline parallelism for collaborative DNN training on heterogeneous edge devices[C]//Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. New York: ACM, 2024: 312-326.

[16] BHARDWAJ R, XIA Z X, ANANTHANARAYANAN G, et al. Ekya: continuous learning of video analytics models on edge compute servers[C]//Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation. [S. l.: s. n.], 2022: 119-135.

[17] PARK J H, YUN G, YI C M, et al. HetPipe: enabling large DNN training on (whimpy) heterogeneous GPU clusters through integration of pipelined model parallelism and data parallelism[EB]. arXiv preprint, 2020, arXiv: 2005.14038.

[18] ZHU L G, HU L X, LIN J, et al. PockEngine: sparse and efficient fine-tuning in a pocket[C]//Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture. New York: ACM, 2023: 1381-1394.

[19] JIANG X T, WANG H, CHEN Y L, et al. MNN: a universal and efficient infer-

ence engine[EB]. arXiv preprint, 2020, arXiv: 2002.12418.

[20] PATIL S G, JAIN P, DUTTA P, et al. POET: training neural networks on tiny devices with integrated rematerialization and paging[C]//Proceedings of the 39th International Conference on Machine Learning. [S.l.: s.n.], 2022: 17573-17583.

[21] WANG S Q, FENG T Y, YANG H L, et al. AtRec: accelerating recommendation model training on CPUs[J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(6): 905-918.

[22] ZHANG L Z, LU K, LAI Z Q, et al. Accelerating GNN training by adapting large graphs to distributed heterogeneous architectures[J]. IEEE Transactions on Computers, 2023, 72(12): 3473-3488.

[23] ZHANG D, HUANG X, LIU Z, et al. AGL: a scalable system for industrial-purpose graph machine learning[J]. Proceedings of the VLDB Endowment, 2020, 13(12): 3125-3137.

[24] MAO J C, CHEN X, NIXON K W, et al. MoDNN: local distributed mobile computing system for deep neural network[C]//Proceedings of the Design, Automation & Test in Europe Conference & Exhibition, 2017. Piscataway: IEEE Press, 2017: 1396-1401.

[25] ZENG L K, CHEN X, ZHOU Z, et al. CoEdge: cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices[J]. IEEE/ACM Transactions on Networking, 2021, 29(2): 595-608.

[26] ZHAO Z R, BARIJOUGH K M, GERSTLAUER A. DeepThings: distributed adaptive deep learning inference on resource-constrained IoT edge clusters [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(11): 2348-2359.

[27] HAO P Z, ZHANG Y F. EDDL: a distributed deep learning system for resource-limited edge computing environment[C]//Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing. Piscataway: IEEE Press, 2021: 1-13.

[28] YANG L, GAN Y Q, CHEN J R, et al. AutoSF: adaptive distributed model training in dynamic edge computing[J]. IEEE Transactions on Mobile Computing, 2024, 23(6): 6549-6562.

[29] CHEN Y H, YANG Q Q, HE S B, et al. FTPipeHD: a fault-tolerant pipeline-parallel distributed training approach for heterogeneous edge devices[J]. IEEE Transactions on Mobile Computing, 2024, 23(4): 3200-3212.

[30] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2016: 770-778.

[31] HUMPHREY E J, BELLO J P. Rethinking automatic chord recognition with convolutional neural networks[C]//Proceedings of the 2012 11th International Conference on Machine Learning and Applications. Piscataway: IEEE Press, 2012: 357-362.

[32] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[EB]. arXiv preprint, 2018, arXiv: 1810.04805.

[33] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[Z]. 2024.

作者简介

王锦权（1998– ），男，北京航空航天大学计算机学院博士生，主要研究方向为分布式训练与并行计算。

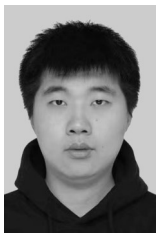
刘旭昭（2000– ），男，北京航空航天大学计算机学院硕士生，主要研究方向为分布式训练与并行计算。

廖晓坚（1994– ），男，博士，北京航空航天大学计算机学院副教授，主要研究方向为内存/存储系统、分布式系统与智能计算系统。

肖利民（1970– ），男，博士，北京航空航天大学计算机学院教授，主要研究方向为计算机体系结构和系统软件、高性能计算机和服务器系统、系统虚拟化与云计算、大数据存储和分布式文件系统、智能计算芯片架构及技术。

霍志胜（1983– ），男，博士，北京航空航天大学计算机学院助理研究员，主要研究方向为大数据存储、分布式/并行存储系统的框架研究及性能优化、大数据环境中重复数据删除。

索珈顺（1997– ），男，北京航空航天大学计算机学院博士生，主要研究方向为机器学习系统。



李云潼（2000– ），男，北京航空航天大学计算机学院硕士生，主要研究方向为分布式训练与通信优化。

沈润楠（1998– ），男，北京航空航天大学计算机学院博士生，主要研究方向为并行计算与分布式文件系统。

谢喜龙（2000– ），男，北京航空航天大学计算机学院博士生，主要研究方向为训推加速器。

唐熙程（2002– ），男，北京航空航天大学计算机学院博士生，主要研究方向为分布式训练与分布式容错。

收稿日期: 2025-02-19

通信作者: 肖利民，xiaolm@buaa.edu.cn；廖晓坚，liaoxj@buaa.edu.cn

基金项目: 国家重点研发计划项目(No.2023YFB4503100)

Foundation Item: The National Key Research and Development Program of China (No.2023YFB4503100)