中国科学技术大学

博士学位论文

异构协同模型推理

作者姓名：　袁牧

学科专业：　计算机科学与技术

导师姓名：　李向阳，张兰

完成时间：　二〇二四年三月二十五日

University of Science and Technology of China

A dissertation for doctor's degree

Heterogeneous Collaborative Model Inference

Author: Mu Yuan

Speciality: Computer Science and Technology

Supervisors: Xiang-Yang Li, Lan Zhang

Finished time: March 25, 2024

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：袁牧　　签字日期：2024-05-27

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

控阅的学位论文在解密后也遵守此规定。

☑ 公开　☐ 控阅（____年）

作者签名：袁牧　　导师签名：

签字日期：2024-05-27　　签字日期：2024-05-27

摘　要

模型推理是支撑诸多人工智能应用的关键，例如交通视频分析依赖于车辆检测模型推理、自然语言问答服务需要基于大语言模型推理实现。将模型推理任务部署于单一设备或同构集群上是最直接和成熟的方式，当下多数智能应用采用这种方案，例如抖音应用基于手机端上部署的视觉模型实现各种视频特效、OpenAI 使用大规模云上 GPU 集群支撑其 ChatGPT 问答服务。然而，随着智能模型愈加复杂、应用场景不断拓宽，基于单一设备或同构集群的模型推理服务显现出很多问题：(1) 环境高度动态，多个数据源分布存在差异，因此静态的推理策略导致大量计算资源浪费；(2) 推理计算卸载引入的通信代价过高，端侧设备尤其是移动设备难以承担；(3) 纯云侧部署的模型推理协议涉及完全明文通信，存在严重的用户端数据隐私泄露风险；(4) 硬件算力受限，如智能物联网系统，单一设备无法部署完整的模型等等。由于软硬件以及部署环境的不同，模型推理任务在真实应用中不可避免地涉及多种异构设备。让这些异构设备合理地、智能地协同执行模型推理，包括协同分担计算任务、协同消除通信数据冗余、协同进行权限分离，是解决单一设备或同构集群面临的资源效率低、隐私安全保护弱等技术挑战的有效途径。因此，本工作聚焦“异构协同模型推理”，通过探索异构设备之间的协同机制，提高模型推理任务的动态自适应性、可扩展性、计算和通信效率、以及对数据隐私安全的保障。具体地，本工作研究了（1）多端协同的并发包门控，通过跨视频流协调解码资源的使用，增强视频实时分析系统中输入源的可扩展性；（2）端边协同的输入过滤，端到端地学习出如何过滤冗余的输入数据，提高通信和计算资源的利用效率；（3）端云协同的安全推理协议，以特征维度的随机置换为基础，赋予模型推理对数据和参数的安全保障；（4）边云协同的自适应模型部署，将原本孤立的模型集合构建为相互关联的模型网络，提高模型部署效率。本工作在理论上分析了所提出技术的性能保障，证明了包门控算法的在线遗憾边界、基于函数族复杂度对比的推理任务可过滤性、以及安全推理协议的隐私泄露上界，并在多个真实系统中进行验证，相较于基线方法，实现显著节省推理开销、大幅提高视频源并发度、优化通信效率等实际优化效果。

关键词：异构计算　端云协同　模型推理　任务调度　安全协议

ABSTRACT

Model inference is crucial for supporting various artificial intelligence applications. For example, traffic video analysis relies on the inference computation of the vehicle detection model, while natural language question-answering services are based on the inference of large language models. Deploying model inference tasks on a single device or homogeneous cluster is the most direct and mature approach. Currently, most intelligent applications adopt this approach. For instance, TikTok implements various video effects based on computer vision models deployed on mobile devices, and OpenAI utilizes large-scale GPU clusters on the cloud to support its ChatGPT question-answering service. However, as intelligent models become more complex and application scenarios continue to expand, model inference services based on a single device or homogeneous cluster present many problems: (1) Highly dynamic environments and distribution shifts of multiple data sources cause static inference strategies to waste significant computing resources; (2) The communication cost introduced by offloading inference computation is prohibitive for end devices (especially mobile devices); (3) The full-cloud model inference protocol involves plain text communication, which brings serious risks of user data privacy leakage; (4) Hardware resources are limited, as seen in AIoT systems where the complete collection of models cannot be deployed on a single device, and so on. Due to differences in software, hardware, and deployment environments, model inference tasks inevitably involve various heterogeneous devices in real-world applications. Enabling these heterogeneous devices to collaborate intelligently in model inference tasks is crucial. The collaboration includes sharing computational workloads, eliminating communication data redundancy, and conducting permission separation collaboratively, etc. It is an effective way to address technical challenges such as resource inefficiency and privacy and security concerns faced by single devices or homogeneous clusters. Therefore, this work focuses on heterogeneous collaborative model inference. This work explores collaboration mechanisms between heterogeneous devices to improve the dynamic adaptability, scalability, computing and communication efficiency, as well as the protection of data privacy and security for model inference tasks. Specifically, this work investigates: (1) Concurrent packet gating with multi-device collaboration. Enhancing the scalability of data sources in real-time video analytics systems by coordinating the use of decoding resources across video streams; (2) Input filtering with device-edge collaboration. End-to-end learning to filter redundant input data to improve the utilization

efficiency of communication and computing resources; (3) Secure inference protocols with device-cloud collaboration. Based on the random permutation in the feature dimensions, providing security guarantees for both data and parameters in model inference; (4) Adaptive model deployment with edge-cloud collaboration. Build an interconnected model network from an originally isolated model set to improve model deployment efficiency. This work theoretically analyzes the performance guarantees of the proposed technologies, demonstrates online regret bounds of the packet gating algorithm, analyzes the filterability of inference tasks based on the complexity comparison of function families, and proves the upper bounds of privacy leakage for the secure inference protocol. This work also evaluates them in multiple real systems. Experimental results show that compared with baseline methods, the proposed techniques achieve practical optimization effects, including reducing inference overhead, improving the concurrency level of video sources, and saving communication bandwidth.

Key Words: heterogeneous computing, device-cloud collaboration, model inference, job scheduling, security protocol

目　录

插 图 清 单

表 格 清 单

第 1 章　引　　言

1.1　研究动机

模型推理，指部署经过训练后的人工智能模型对新数据进行预测的过程，是实现人工智能应用的关键任务。例如，智慧城市系统[1-2]需要部署包括车辆检测模型、人物行为识别模型等等，对实时监控视频数据进行推理，从而支撑下游的交通智能管理等功能。再比如时下热门的智能问答系统，包括 ChatGPT[3]，其服务依赖于大语言模型[4]的自回归推理（指根据提供的输入文本递归预测下一个词）过程。因此，模型推理的性能直接影响人工智能应用的服务质量，如何实现高效率、可扩展、安全的模型推理业已成为系统领域的重要研究方向。

当前主流的、成熟的部署方式是将模型推理任务部署于单一设备或同构的服务器集群上。对于单一设备，例如手机上各种视频特效应用（抖音等等）的实现大多依赖本地部署的计算机视觉模型推理（人体、面部关键点检测[5]等功能）；对于同构集群，典型的例子是智能问答服务，OpenAI 使用微软 Azure 大规模云上 GPU 集群支撑其 ChatGPT 全球数亿用户的使用[6-7]。这种将模型推理任务部署于单一设备或同构集群的方案具有诸多优点，包括端侧本地部署的模型能够保护用户数据隐私[8]，以及云集群的高鲁棒性和可扩展性能够保障云上推理服务的高效和稳定[9]。然而，随着智能模型参数量的增加、应用场景不断拓宽和深入、性能和安全性指标更加严格，现有的基于单一设备或同构集群的模型推理部署方案不再能够满足所有需求，显现出诸多问题：

（1）数据源并发度受限。数据源的高并发度对于大规模的模型推理服务至为重要。以实时智能视频分析为例，在一栋建筑内，视频源并发需求大约在几十路；在一个园区内，可能需要支持数百、数千路的并发；而对于智慧城市，则并发度需求要达到数万甚至十余万的量级[1]。如果视频推理系统无法支持高并发推理，将只能应用于小规模的场景，难以扩展到智慧园区、智慧城市等重要应用。然而，当前提高视频分析并发度的方法一般都是增加硬件，但无论是专用的编解码设备还是通用 GPU，其扩展成本都非常高昂。例如，经实际测算，以 1080p 25FPS 的视频流为例，专用的解码硬件 Kiloview DC230[10]的单路扩展成本约为 62.5 美元，NVIDIA A100 GPU 的单路扩展成本则为 144 美元[11]。除了硬件成本之外，增加设备还会带来额外的通信开销和大量的工程维护工作。因此，本文将针对“数据源并发”研究如何以纯软件解决方法来优化多源推理效率。

（2）通信计算开销过高。随着移动设备的算力不断增强，以及对实时传感器数据分析需求的持续增长，移动人工智能已经成为一种重要趋势。例如，在设备上进行推理的计算机视觉模型为用户提供了越来越丰富的实时增强现实体验。

再比如，通过结合无人机设备和边缘计算，能够实时分析无人机拍摄的视频。对于资源有限的移动设备和对延迟敏感的模型推理任务而言，通信和计算资源利用效率至关重要。然而，即使将模型推理任务卸载部署到边缘服务器上，在高吞吐量推理时，对于端设备来说通信代价仍然过高，且对于边缘服务器而言计算过于密集。因此，本文将针对“端边传算效率”研究如何通过消除待推理的数据冗余性降低通信和计算开销。

（3）数据安全风险严重。基于 Transformer 架构的模型推理服务已经成为最引人注目的人工智能应用，例如，ChatGPT 创下了用户增长最快的记录[12]。为了支撑 Transformer 模型推理，特别是具有数十到数千亿参数的大型模型，云计算平台是一种理想选择。工业界业已发布了多个云原生的 Transformer 推理的框架，例如 NVIDIA NeMo[13]和 Microsoft DeepSpeed[14]。包括 OpenAI 在内的大多数大模型应用提供商选择为其基于 Transformer 的推理服务进行全云部署[7]。然而，将原始数据发送到云端在各种隐私敏感的领域是不可行的，例如，三星在发生了敏感代码泄露事件后正式全面禁止员工使用 ChatGPT[15]。为了解决数据隐私问题，另一种极端化的选择是完全在端设备上部署 Transformer 模型。通过诸如权重量化[16]之类的模型压缩技术，具有数十亿参数的 Transformer 模型可以在端设备上运行推理[17]。然而，完全端侧部署的可扩展性非常有限。浮点操作（FLOPs）和内存占用随参数数量线性增长，而端设备上的计算资源增速远远慢于 Transformer 模型大小的增速[4,18]，而且模型压缩会不可避免地导致准确性损失[19]。因此，本文将针对“模型推理安全性”研究如何通过设计端云参与的协议实现对模型参数和用户数据的隐私保护。

（4）模型部署效率低下。得益于人工智能技术的飞速发展，通过互联网能够获取非常丰富多样的可用模型（例如 HuggingFace 平台在 2024 年 3 月维护了 564,913 个不同的模型 ①）。云上的这些模型能够解决各种数据模态的多样的推理任务，甚至对于完全相同的任务都有成百上千种不同结构、参数量的模型可供选择。然而，当可选项多到一定程度的时候，可能并不是一件好事。当需要在一个具体应用场景中部署模型的时候，选择哪个或哪些模型就成了一个难以解决的技术挑战。在精度方面，由于具体场景的数据分布与云上模型的预训练数据集分布存在偏差，云上的测试精度与本地推理精度往往存在较大偏差[20]。在效率方面，具体场景的部署设备算力有限，难以准确预估云上模型的具体推理延迟. 对于多任务推理，设备内存较低的情况下往往无法并行加载全部的任务模型，导致推理结果召回率低下。因此，本文将针对“模型部署效率”研究如何从云上模型库中选择最优的模型子集部署到边缘设备。

① https://huggingface.co/models

1.2　领域现状

本文旨在研究异构设备协同的网络架构下的模型推理系统的资源高效性和安全性，在研究目标或方法设计上，本文与下列三类技术存在相关性。

1.2.1　模型推理加速

模型推理加速技术[21]主要分为如下三种:

轻量化模型设计。针对计算能力非常有限的移动物联网设备，现有工作设计多种计算效率极高的卷积轻量架构：ShuffleNet 通过引入分组卷积技术降低深度可分离卷积模块中计算量较大的逐点卷积开销[22]；MCUNet[23]使用了高效的神经网络架构搜索方法，并针对微控制单元（MCU）设计了高效推理引擎，其重点优化了代码生成、内存调度、卷积循环展开、算子融合等操作，大幅降低了推理内存开销。

神经网络剪枝。神经网络剪枝[24]是一种优化神经网络结构的技术，旨在减少网络参数和计算量，从而提高模型的效率和推理速度，同时保持模型的性能。其核心思想是通过去除冗余的连接、神经元或层，以达到减少模型复杂度的目的。这种技术的出发点是稀疏性原理[25]，即神经网络中许多参数是冗余的，不是所有的参数都对模型的性能起到重要作用。因此，通过剪枝可以去除这些冗余参数而不影响模型的性能。剪枝技术通常分为结构化剪枝和非结构化剪枝两种类型[26]，其中结构化剪枝是在网络的特定结构上进行剪枝，如整个层、通道或滤波器的剪枝，而非结构化剪枝则是直接在参数级别上进行剪枝，不考虑其位置。剪枝策略包括权重剪枝、通道剪枝、模型剪枝等，它们通过不同的方式来实现对网络结构的优化。剪枝后的网络通常会经过微调（Fine-tuning），以恢复或进一步提高模型性能。虽然神经网络剪枝可以显著减少模型的计算量和存储需求，提高模型的推理速度，并在一定程度上防止过拟合，但剪枝可能会引入误差，需要通过微调等手段进行补偿，并且剪枝过程本身可能需要大量的计算资源和时间。

参数量化。神经网络参数量化[27]是一种将神经网络中的参数表示为低精度的形式的技术，旨在减少模型的存储需求和计算复杂度，同时保持或最小程度地影响模型的性能。在参数量化中，通常将原始的浮点数参数转换为更小的整数或者低位宽的浮点数，以减少存储空间和计算开销。这种技术的主要思想是，在保持模型精度的前提下，通过量化参数可以大幅减少模型的存储需求和计算复杂度。参数量化通常分为两种类型：权重量化和激活量化[28]。权重量化是将网络中的权重参数转换为低位宽的表示形式，例如将 32 位浮点数参数量化为 8 位整数；而激活量化则是将网络中的激活值进行量化。常见的量化方法包括定点数表示、二值化、三值化等[29]。参数量化的优势在于能够显著减少模型的存储空间和计算开销，从而使得模型可以在资源受限的设备上进行部署和运行，比如移动

设备、嵌入式系统等。然而，参数量化也可能会引入精度损失，从而降低模型的性能。因此，在量化过程中需要权衡模型的性能和资源消耗，并通过一系列技术手段来尽可能地减少量化带来的影响，如量化感知训练、自适应量化等。

关联和创新性：本文优化目标之一是提高模型的推理效率，这一点与现有模型推理加速的方法相同。但是在方法上，本文聚焦如何利用异构设备协同提高效率，这一点与专注模型本身的模型结构设计、神经网络剪枝、参数量化等思路完全不同。本文提出的多模型部署、输入过滤、多流包门控算法在工作原理上可以与模型加速方案互补，即本文提出的方法可以作用于已经完成加速优化后的神经网络模型。

1.2.2　端云协同学习和推理

作为分布式机器学习的新兴范式，拆分学习（Split Learning）[30] 是为资源受限的数据所有者提出的，通过让算力充裕的服务器承担大部分计算，数据所有者（端侧设备）承担小部分模型计算的方法，实现数据不出域的模型训练。在拆分学习中，机器学习模型（神经网络）通常被分为客户端模型和服务器模型，其中服务器模型占模型的大部分参数。客户端模型以隐私数据为输入，并将中间隐层输出发送给服务器完成后续计算，实现本地隐私数据的保护。在拆分学习背景下，现有工作探索了如何使多个端设备共享同一个集中式服务器[31]、如何结合横向联邦学习技术以提高参数更新效率[32]、如何利用梯度信息优化通信开销[33]等技术方向。随着大模型研究的发展，一些近期的工作将拆分学习与大语言模型相结合，设计了对弱算力友好的大语言模型微调方案[34]。自然地，模型拆分推理指的是将神经网络模型分拆到多个设备（例如端设备和云服务器）上进行推理。现有技术重在研究如何通过合理地选择拆分位置，实现充分利用端侧设备算力的情况下降低通信开销和时延，包括使用回归模型预测不同类型层的计算开销和输出数据传输开销，之后线性搜索总代价最小的二拆分方案[35]；在神经网络不同位置增加提前退出推理过程的分支，并在推理阶段尝试对所有早退出分支线性搜索总代价最小的二拆分方案[36]；基于图最小割选择满足内存约束且总推理时延最小的拆分方案[37]等多种角度。

关联和创新性：本文考虑异构设备协同的模型推理场景，在模型部署方式会考虑将一个完整的推理模型拆分到端侧和云侧进行部署，这与模型拆分的思路相同。本文的创新性在于，将冗余过滤的思路首次引入视频解码阶段并提出多端协同的包门控算法（第 2 章）、以及提出了第一个端到端可学的全模态输入过滤框架（第 3 章）、针对异构设备协同推理任务提出了新颖的基于模型链接的多模型部署算法（第 5 章）。

1.2.3 深度学习模型安全推理

当前针对深度学习模型、大模型的安全推理协议的研究都是在“安全多方计算”(Secure Multi-Party Computation)[38]的理论框架下进行的，通过同态加密(Homomorphic Encryption)[39]等方案, 实现对双方（数据方和模型方）的隐私保护。Iron[40]基于定制的同态加密方案实现安全的矩阵乘法，并针对 SoftMax、GeLU 激活和 LayerNorm 等复杂非线性层设计了安全计算协议。THE-X[41]也是基于同态加密技术, 对 Transformer 模型中的复杂非线性层进行近似计算。Iron 和 THE-X 都考虑 BERT 模型用于验证，模型参数量在数亿级别。CipherGPT[42]为 GPT 架构的 Transformer 模型定制了安全矩阵乘法，以及针对自回归生成任务的 top-k 采样协议。这些基于加解密的方案具有较强的安全性保障，但是也不可避免地带来了高开销，且不能够支持产品级别推理框架（例如 DeepSpeed 中使用的 KV-cache 优化技术[14]），对于千亿级别的大模型而言可行性较低。另一方面，这些方法对于复杂非线性层的近似会带来推理精度损失，这一问题在诸多应用场景下是不可接受的，且难以扩展到其他大模型变种架构。因此，研究需要解决这些挑战，寻找更有效的方法来保护隐私、降低计算开销，并同时确保推理的精度无损。这将有助于推动大模型在隐私敏感领域的更广泛应用。

隐私计算的另一个重要技术是可信执行环境（Trusted Execution Environment, TEE）[43]，这是一种在计算设备中创建隔离和受保护的环境，确保代码和数据的机密性和完整性。TEE 的主要目的是保护敏感信息和代码免受恶意软件和未授权访问的影响。TEE 具有四个核心特点：（1）隔离性：TEE 运行在独立于操作系统和其他应用程序的环境中，确保其内部的数据和代码不受外界干扰。（2）机密性：保护内部数据的机密性，防止未经授权的访问和泄露。（3）完整性：保证代码和数据在传输和执行过程中不被篡改。（4）可信执行：通过安全启动（Secure Boot）和硬件信任根（Root of Trust），确保仅可信的代码可以在 TEE 中运行。TEE 通常通过硬件支持实现，主要的实现方式包括：ARM TrustZone[44]：一种常见的 TEE 实现方案，分隔普通世界和安全世界，允许在相同的处理器上运行可信代码和非可信代码。Intel SGX (Software Guard Extensions)[45]：提供硬件加密保护的内存区域，在这些区域中运行的代码和数据受到保护。AMD SEV (Secure Encrypted Virtualization)[46]：为虚拟机提供内存加密，保护虚拟机中的数据和代码免受外部攻击。针对智能模型的执行，TEE 可以实现的效果主要包括：（1）数据保护：通过在 TEE 中执行智能模型，确保输入数据、模型参数和输出结果的机密性。敏感数据在进入 TEE 之后不会泄露到外部世界。（2）模型保护：保护智能模型本身的知识产权，防止模型被逆向工程或复制。（3）可信推理：确保推理过程的完整性和可信度，防止模型执行过程中被篡改。基于这些特性，TEE 技术能够被应

用于隐私敏感的人工智能场景，包括（1）隐私保护机器学习[47-48]：在医疗、金融等领域，通过 TEE 执行模型推理，确保用户数据隐私，允许数据在加密状态下直接进行计算。（2）边缘计算和物联网[49]：在智能设备和传感器上使用 TEE，保护设备中的智能模型和数据，保证设备的安全性和数据的隐私性。（3）联邦学习[50-51]：在多个参与方之间共享模型而不共享数据，通过 TEE 保护各方的数据隐私和模型参数。通过 TEE 技术，智能模型可以在保护数据隐私和安全的前提下，发挥其强大的计算和分析能力。

关联和创新性：本文实现了一种针对自注意力机制模型（即 Transformer）的安全推理协议，在安全性上与现有工作一致，即具有理论的安全保障性。本文的创新性在于，首次在三方（模型开发方、云平台方、用户方）场景下研究精度完全无损的安全推理方案，并且仅引入轻量化的特征置换操作，而非高开销的同态加解密运算，且不需要特殊的计算硬件。

1.3 研究内容

针对上述模型推理任务涉及的四个关键技术问题，即数据源并发度受限、通信计算开销过高、数据安全风险严重、模型部署效率低下，本文具体进行了四个方面的研究。如图 1.1所示，以一个视频分析系统作为应用实例，展示了本文的四个主要研究内容。在这样一个视频分析系统中，部署的推流节点将实时监控视频流推送给边缘服务器，边缘服务器完成视觉模型推理得到分析结果。在用户交互方面，基于大语言模型实现了一个自然语言问答接口，用户（例如安保工作人员）只需询问如“当前某某会议室是否有人”之类的自然语言问题，部署在云数据中心的大语言模型将自动地根据查询到的推理结果形成自然语言回复。边缘服务器上部署的具体视觉模型由开发者从云端视觉模型库里做选择。本文的四个主要研究问题皆是以“异构设备相互协同”为基本思路，解决了上述复杂的智能系统中的技术挑战。

（1）多端协同的并发包门控。本研究开发了一个校园实时视频分析系统，处理安装在公共区域的 1000 多个摄像头。为了实现校园安防功能，部署了包括物体检测和人物动作识别在内的多个视觉模型[52-53]，并应用了多种模型加速（包括 TensorRT[54]）技术以提高资源效率。在本研究使用的的边缘 GPU 服务器上，这些方法有效地将系统吞吐量从 27 FPS 提高到 3,500 FPS。然而，在开发过程中，本研究发现了一个先前被忽视的瓶颈：视频源并发度，即可以同时处理的视频流的数量。实验结果显示，端到端的并发受到视频解码模块的限制（该模块将编码后的视频包作为输入，并输出解码后的 RGB 帧）：解码模块只能支持 35 个流，而推理模块的并发度是 3015。本文提出在视频推理流程的解码器之前添加一个

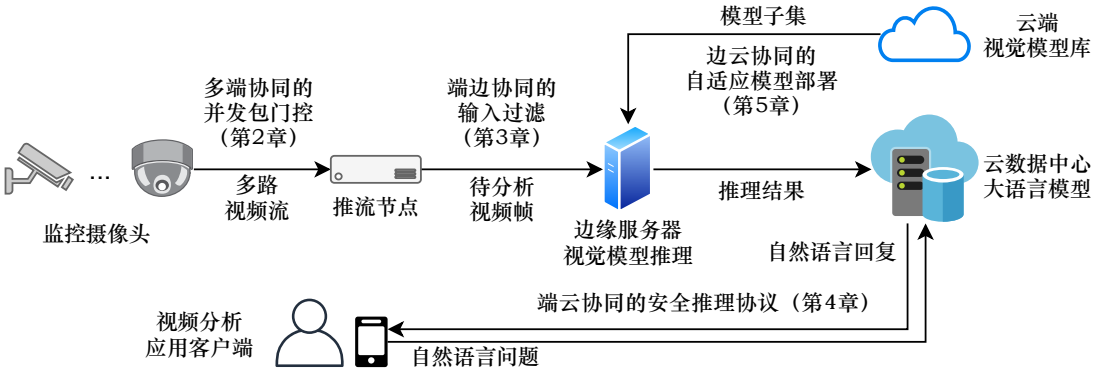


图 1.1　研究内容在视频分析系统上的应用实例

筛选模块，命名为视频包门控，从所有流中仅选择一部分视频包进行解码。视频包门控不仅可以减轻推理模型的计算开销，更能减轻视频解码器的计算开销。首先，本文形式化了多流视频包门控问题并分析了算法结构，并提出了一个通用框架。本研究设计了一个基于滑动窗口的时序估计器，使用在线反馈和决策历史预测每个流的选择概率，以及一个多视图神经网络，作为上下文预测器。上下文预测器学习将独立帧和依赖帧的视频包大小嵌入为不同的特征视图，并融合时序估计器返回的概率，得到最终的视频包置信度。基于来自多个流的置信度，本文提出了一个组合优化器，并证明了其 $1-c/B$ 的最优性能近似比，其中 B 是解码预算，c 是一个视频包的最大解码成本。基于多臂赌博机理论[55-56]，本文证明了整体算法的 $\tilde{O}(\sqrt{T})$ 的遗憾上界，其中 T 是决策轮数。

（2）端边协同的输入过滤。在端边协同推理场景下，本文研究如何过滤输入数据中的冗余[57-63]来提高计算和通信效率。本文将这一系列方法统称为输入过滤，并将其分类为跳过和重用两类：(1) 跳过方法[57,62] 的目标是过滤掉会导致无用推理结果的输入数据，例如，对于人脸检测器，没有脸部的图像是无用的，而对于语音识别器，没有有效命令的音频也是无用的。(2) 重用方法[59-60] 则试图过滤那可以重复使用先前推理结果的输入数据，例如，相同动作的运动信号和具有相同车辆检测数量的视频帧。与模型优化相比，输入过滤在准确性和效率之间提供了更灵活的权衡，例如调整跳过方法中的阈值以及重用方法中使用的缓存大小。本文首先形式化了输入过滤问题和有效过滤器条件。然后，本文从理论上定义了“可过滤性”，并通过比较推理模型及其输入过滤器的假设族复杂性[64-65]对两种最常见的推理任务（分类和回归）的可过滤性进行了证明。本文提出了第一个端到端可学习的输入过滤框架，统一了跳过和重用方法[57-59]。端到端的可学习性以一种与任务无关的方式提供了具有鲁棒辨别力的特征嵌入，从而显著拓宽了适用范围。基于统一的框架，本文设计了一个名为 InFi 的输入过滤系统，对各种数据模态和部署方式都提供了支持。

（3）端云协同的安全推理协议。现有研究通过同态加密（HE）和安全两方计算

（2PC）设计了理论上安全的 Transformer 推理协议[40-42]。然而，这些协议产生了巨大的计算和设备与云端通信开销，特别是在具有非线性复杂层（如 LayerNorm 层和 ReLU 层）的情况下。例如，在 CipherGPT 协议下，使用 GPT2 模型生成单个词花费了 25 分钟的处理时间和超过 90 GiB 的流量[42]。为了克服效率障碍，本文使用第一性原理思维重新思考基本的两方假设：模型所有者和数据所有者。本文从两个真实模型推理服务中得到了一致的经验：模型开发者 $\neq$ 模型服务器。在开发这两个服务的过程中，本研究使用收集到的数据微调[66]开源参数[67-68]Transformer 模型。本研究有足够的计算能力进行离线模型开发，但缺乏为众多用户提供大规模、长期服务的算力。因此，模型开发者需要依赖第三方云平台为完成开发的模型提供服务。通过与真实开发经验对齐，本文提出了一个新的三方威胁模型。在这个模型中，本文将模型所有者分解为两个实体：模型开发者和模型服务器。由于开发的模型是专有的，模型开发者必须保护他们的模型参数免受来自模型服务器的潜在攻击，因此本文假设它们不会共谋。基于本文引入的三方威胁模型，开发了安全推理协议 STIP，即 Secure Transformer Inference Protocol 的缩写。首先，本文采用高效的特征空间随机置换进行安全且等价的 Transformer 推理。由于推理是在不受信任的服务器上执行的，模型参数和设备上的数据必须在上传到云端之前进行转换。基于特征空间的高效随机置换，本文设计了一种 Transformer 层的数据和参数转换方法。本文证明了提出的转换进行计算的数学等价性，从而确保没有准确性损失。其次，本文设计了一种模型开发者和数据所有者之间的半对称保护方案，这个洞察源于神经网络的顺序结构。模型开发者只需要与数据所有者共享第一层和最后一层的相同置换，就可以保护中间层转换的信息。本文通过距离相关性[69]展示了 STIP 的隐私保护性，并证明其对暴力和已知明文攻击的抵抗性。

（4）边云协同的自适应模型部署。本文从一个新的角度来解决如何从云端选择模型部署到边缘设备的问题：链接黑盒模型。其基本思想是，将原本无关联的、独立训练的机器学习模型视为节点，通过构建这些模型节点之间的语义关联性，将其链接为互相关联的“模型网络”结构。产生这一想法的动机是，现有工作观察到机器学习模型容易出现过拟合[70]，因此即使机器学习模型在输入模态、学习任务、架构等方面有所不同，它们也可以相互共享知识[71]。如果能有效地在机器学习模型之间建立知识桥梁，则可以基于执行模型的输出直接预测其余模型的推理结果。如果这种预测的成本较低，与独立推理的原始工作流程相比（无法获取未执行模型的结果），基于模型链接的方法有望在有限的成本预算下显著提高部署的模型推理结果的准确性。

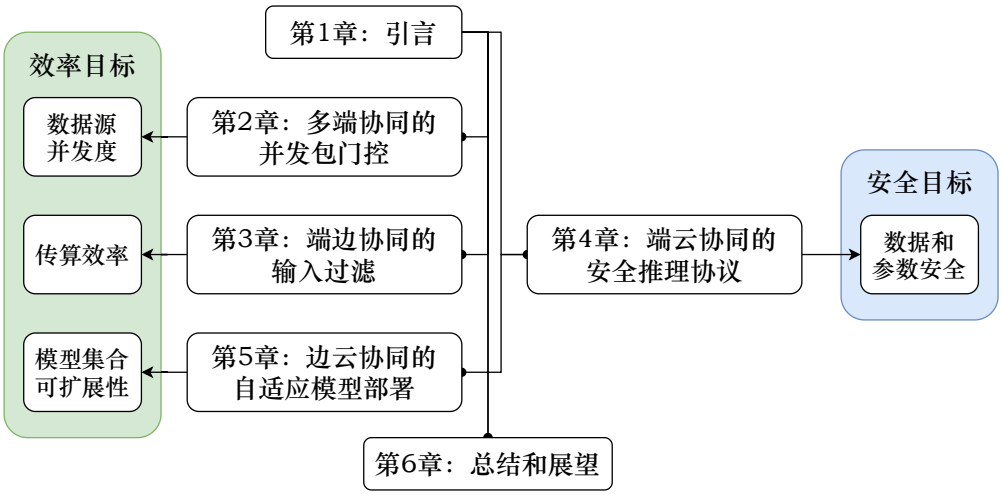


图 1.2　论文组织架构

1.4　主要贡献

本文围绕异构协同模型推理进行研究，组织架构如图 1.2所示。本文的主要贡献总结如下：

• 在第2章中，本文首先阐述了如何从实际系统中发现视频推理流程中被忽视的并发度瓶颈。针对视频分析端到端并发度提升，本文提出了一个新的思路：视频包门控，它对现有优化方法在解码效率方面进行了补充。本文提出了第一个用于多流视频包门控的框架 PacketGame，其利用轻量级的时序估计器和上下文预测器来自适应地表示视频包。同时，本文设计了一个组合优化器，并证明了其在有效的跨流协调方面具有最优近似比。本文证明了 PacketGame 的整体性能具有在线遗憾上界，并在 1108 个摄像头的真实系统和公共视频上进行了四个推理任务的验证。实验结果显示，与原始推理方法相比，PacketGame 节省了 52.0-79.3% 的解码成本，实现了 2.1-4.8 倍的并发性。与四种最先进的互补方法[54,58,63,72]的比较表明，在端到端并发性和广泛适用性方面，PacketGame 具有显著的优越性。相关代码于https://github.com/yuanmu97/PacketGame开源。

• 在第3章中，本文首先对输入过滤问题进行了形式化，并提供了过滤器的有效性条件。本文基于推理任务和输入过滤器的假设族之间的复杂性比较进行了分析，这可以指导和解释输入过滤技术的应用。本文提出了第一个端到端可学习的输入过滤框架，统一了跳过和重用方法。得益于端到端可学习性，提出的框架具有鲁棒辨别力的特征嵌入，支持更多的输入模态和推理部署方式。本文设计并实现了一个名为 InFi 的输入过滤系统。对包括 8 种输入模态、14 个推理任务进行全面验证表明，InFi 具有更广泛的适用性，并在准确性和效率方面优于基线方法。对于移动平台上的视频分析应用，InFi 相对于原本的车辆计数任务，可以实现高达 8.5 倍的吞吐量提升，并节省 95% 的带宽，同时保持超过 90% 的准确性。相关代码于https://github.com/yuanmu97/infi开源。

• 在第4章中，本文首先确定了现有的针对 Transformer 模型安全推理协议的两方设置（模型所有者和数据所有者）中固有的效率瓶颈及其与现实应用的不一致之处。不同于传统同态加密和安全两方计算框架，本文提出了一个新的三方威胁模型，将模型所有者分解为两个不同的实体：模型开发者和模型服务器。基于此本文提出了第一个用于三方 Transformer 推理的安全协议 STIP，证明了其具有隐私泄漏的理论界限和精度无损的保证。本文实现了 STIP 并在实际系统上对各种 Transformer 模型（最大的具有多达 700 亿参数）进行了验证。实验结果展示了 STIP 的效率能够与未保护的全云推理相媲美，超过了最先进的安全两方协议[40-42]数百万倍。相关代码于https://github.com/yuanmu97/secure-transformer-inference开源。

• 在第5章中，本文首先形式化了模型链接任务，并提出了支持异构黑盒机器学习模型的模型链接设计。本文提出了模型链接的适应和聚合方法，涵盖了在线动态和跨领域分布偏移两个方面。本文开发了一种基于模型链接的算法，命名为 MLink，用于在有限成本预算下优化多模型推理的部署任务。本文在一个包含七个不同机器学习模型的多模态数据集上验证了 MLink 的设计，涵盖了五类学习任务和三种输入模态。结果表明，本文提出的模型链接可以有效地在异构黑盒模型之间建立。本文还在两个真实的视频分析系统上验证了 MLink，一个用于智能建筑，另一个用于城市交通监控，包括六个视觉模型和来自 58 台摄像机的 3264 小时视频。实验结果显示，相较于原始的离线训练，提出的在线自适应训练方法有效地提高了 MLink 的性能提出的聚合方法实现了比原始模型高 7.9% 的平均准确度。在 GPU 内存预算下，MLink 显著优于多个基线（多任务学习[73]，基于深度强化学习的调度器[74]和帧过滤[58]），可以节省 66.7%的推理计算同时保持 94%的输出准确度。相关代码于https://github.com/yuanmu97/MLink开源。

第 2 章　多端协同的并发包门控

对于各种来源（IP 摄像头[58]、无人机[75]、移动直播[76] 和用户生成内容[77]）进行视频推理（基于人工智能模型推理的视频分析）的需求迅速增长。例如，智慧城市系统将计算机视觉模型应用于来自数万摄像头的视频，用于紧急响应和环境保护[78]；Twitch 平台每时每刻都有超过 100,000 个并发直播流[79]，近期的研究提出使用神经超分辨率（Super-Resolution）来提高视频质量[80]。

典型的视频推理流水线[61,63,78,81-83]（见图 2.1）首先从实时网络流（例如 RTSP）或本地文件系统（例如 MP4）中解析视频，然后解码数据包并在 RGB 图像帧上运行人工智能模型。现有的优化视频推理流水线效率的工作可以分为四类：(1) 摄像头端帧过滤[58] 是在在线视频流分析的开始阶段对帧进行过滤。在每台摄像头上，它基于连续帧之间的特征差异选择帧，并仅对选定的帧进行编码以传输到服务器。(2) 视频压缩[72]。与常见的视频编码方法[84]（例如 H.264 和 VP9）旨在为人类视觉感知设计不同，视频压缩旨在最小化推理模型的感知损失。因此，它可以有效提高针对模型推理任务的视频传输效率。(3) 服务器侧帧过滤[63,83] 与摄像头端帧过滤有相同的思想，但将过滤器移到服务器上。在解码视频后，这一系列方法根据神经网络分类器决定是否对每一帧执行推理。(4) 模型加速[54,85] 侧重于流水线的最终推理阶段。通过修剪和融合深度神经网络中的算符，它提高了推理模型的计算效率。

系统观察：并发瓶颈。本工作开发了一个实时视频分析系统，处理安装在公共区域的 1000 多个摄像头。为了支持人物轨迹分析和紧急响应功能，本工作部署了先进的视觉模型[52-53]，并应用了服务器侧帧过滤（InFi[63]）和模型加速（TensorRT[54]）技术以提高资源效率。在开发用的边缘 GPU 服务器上，这些方法有效地将系统吞吐量从 27 FPS 提高到 3,500 FPS。然而，在系统一年的运行中，本工作发现了一个先前被忽视的瓶颈：并发水平，即可以同时处理的视频流的数量。实验结果显示，端到端的并发受到视频解码模块的限制（该模块将编码后的视频包作为输入并输出解码的 RGB 视频帧）。使用 12 个 CPU 的解码模块只能支持 35 个流，一个 GPU 能够支持 18 个流，而帧过滤和推理模块的并发水平分别是 143 和 3015，显著高于解码模块。原因是解码器和帧过滤器需要处理所有帧，而推理模块只需要处理通过过滤器的帧的一个更小的部分（<2%）。

这项工作提出在视频推理流程的解码器之前添加一个选择器模块，命名为包门控（即从所有流中选择一部分需要解码的视频包）。与之前利用 RGB 图像的低级或学习特征的帧过滤方法不同[58,63,83]，本文尝试根据解析视频流的包元数据来进行选择决策。包门控不仅可以减轻推理模型的计算开销，更重要的是减轻

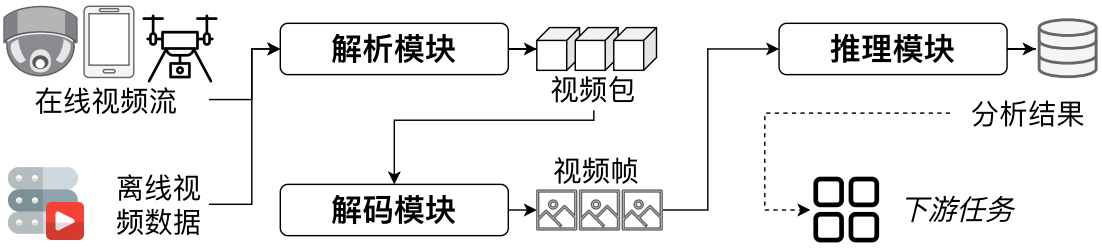


图 2.1　视频推理流程

视频解码器的计算开销。此外，包门控不需要修改视频编码和传输协议，因此可以支持通用摄像头和离线存储的视频。在摄像头帧过滤和视频压缩方法中，缺乏广泛适用性和可插拔性。请注意，包门控的设计旨在提高并发水平。因此，在本文的设计中，跨流协调（即基于有限解码能力选择多个流的包）是自然而然的重要考虑因素，而之前的工作对此关注较少。

构建一个有效的多流包门控框架涉及两个关键挑战：

(1) 非自适应视频包表示。在解码之前，仅在视频包的一些元数据可用时，例如视频编解码器、图像类型、包大小等，就要开发适应各种推理任务和视频内容的视频包表示是具有挑战性的。本文可以参考的现有想法例如包分类[86-87]、网络流量分类[88-89]以及视频帧的重要性建模[90-91]。然而，这些表示方法并非专为通用的视频推理流程而设计。实验结果表明，它们无法有效地适应不同的视频推理应用。

(2) 低效的跨流协调。要提高并发水平，仅对单个视频流进行包门控是不够的。对于并发流，动态内容和非均匀的解码开销（由视频编解码器设置引起）使得现有的资源调度器性能极低。实验结果表明，在并发流数量增加时，经典的轮询策略的性能显著下降。

克服包门控带来的这些挑战需要多方面的技术进步。首先，本章形式化了多流包门控问题并分析了算法结构。本章提出了一个通用框架，并找出了主要的理论与实践差距，即视频包序列嵌入和图片组（Group of Pictures / GOP）中的解码依赖性。其次，本章设计了一个基于滑动窗口的时序估计器，使用在线反馈和决策历史预测每个流的选择概率。本章设计了一个多视图神经网络，作为上下文预测器。上下文预测器学习将独立和依赖帧的视频包大小嵌入为不同的特征视图，它还融合了时序估计器返回的概率，得到最终的视频包置信度。基于来自多个流的视频包置信度，本章提出了一个组合优化器，并证明了其 $1-c/B$ 的最优近似比，其中 B 是解码预算，c 是一个视频包的最大解码成本。基于多臂赌博机理论[55-56]，本章证明了提出的整体算法的 $\tilde{O}(\sqrt{T})$ 的遗憾上界，其中 T 是决策轮数。本章实现了这一具有理论支持的算法，命名为 PacketGame，作为插件在视频推理流程的视频包解析器和解码器之间工作。实验在 1108 个摄像头的真实系统和公开视频数据集上进行了四个推理任务的验证。实验结果显示，与原始工作

负载相比，PacketGame 节省了 52.0-79.3% 的解码成本，并实现了 2.1-4.8 倍的并发性。与四种最先进的互补方法[54,58,63,72]的比较表明，在端到端并发性和广泛适用性方面，PacketGame 具有优越性。

2.1　背景

本节首先介绍动机用例：具有大规模并发需求的广泛视频推理任务。然后，本节讨论现有工作为高效视频推理而做的努力。接下来，基于运行实际系统的经验和定量分析，确定了并发水平的瓶颈。然后，本节提出了一个新的思路，即包门控，并阐明设计范围和独特性。

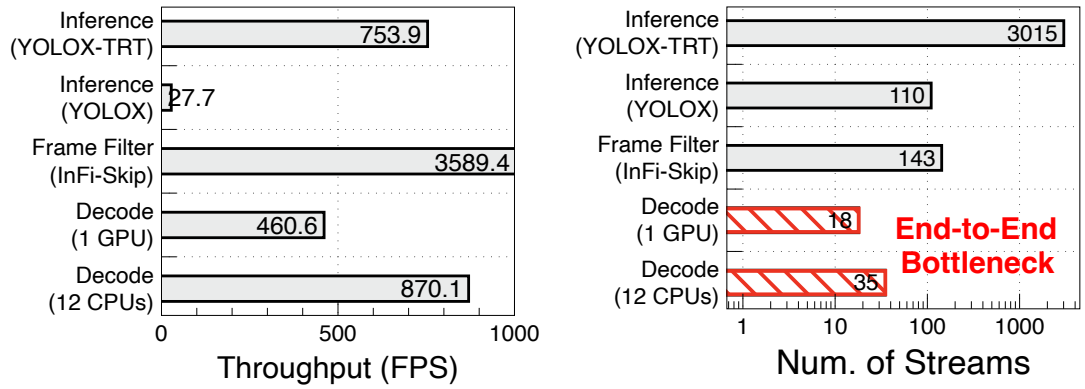
动机用例。(1) 监控视频推理。监控摄像头在当今社会随处可见，广泛用于家庭和公共区域的安全。人工智能模型已经赋予了许多城市中数万个 IP 摄像头的视频以紧急响应等分析功能[78]。(2) 移动视频推理。各种移动设备，如手机、无人机和机器人，都配备了摄像头。由于通信和计算资源有限，许多应用将视频推理卸载到边缘和云服务器上进行处理[92]，例如基于工人身上佩戴的摄像头进行施工现场管理[93]。(3) 离线视频推理。视频分享平台存储了大量的视频（例如 YouTube 上至少有 8 亿个视频[94]）。为了提高服务质量，开发了各种人工智能模型，用于活动级别的广告[77]、基于内容的检索[95]和分辨率增强[80]等功能。随着硬件和用户数量的增加，无论视频源是什么，这些应用对规模化并发处理有着共同的需求。

高效视频推理。现有工作主要探索了四类方法来提高视频推理的效率。四种代表性方法（也是在后续验证实验中用于比较的方法）总结如下：(1) 视频压缩。Grace[72] 提出了一种视频压缩算法，显著节省了网络带宽，同时没有降低推理性能。Grace 通过分析空间频率和颜色优化了目标推理模型的编解码器压缩策略。(2) 摄像头端帧过滤。Reducto[58] 通过根据低级视觉特征（例如像素和面积）自适应地设置帧差阈值，在摄像头端过滤帧。Reducto 仅对选择的帧进行编码和传输，从而节省了网络带宽和后端推理计算。(3) 服务器侧帧过滤。InFi[63] 使用轻量级卷积神经网络学习过滤解码帧。其端到端的可学性为不同推理任务提供了具有强鲁棒性的特征嵌入。(4) 模型加速。TensorRT[54] 对 NVIDIA GPU 实现了许多推理加速技术，包括权重量化、层融合、并行执行等。

2.1.1　并发瓶颈

调研发现，大多数现有工作侧重于视频推理的延迟和吞吐量指标，但整体并发水平的瓶颈仍未被深入探讨。

实际系统经验。在开发一个接入 1000 多个 IP 摄像头并发视频流的视频分



(a) 独立吞吐量　　**(b) 潜在并发度**

图 2.2　使用 25FPS 1080p 视频流的推理性能基准测试结果

析系统时，本工作应用了 TenosrRT[54] 和 InFi[63] 来提高效率。如图 2.2a 所示，TensorRT 显著提高了推理模型（YOLOX[52]）的吞吐量，从 27.7 提高到 753.9 FPS。InFi 则在保持 90% 以上准确率的同时，实现了 99% 的过滤率。然而，当进行实时处理时，本文发现视频解码器成为了端到端的瓶颈。如图 2.2b 所示，使用 12 个 CPU / 一块 TITAN X GPU 在边缘只能支持 35 / 18 个并发流。下游过滤器和推理模型的潜在并发水平分别为 143 和 3015，比解码的数量级高得多。原因非常明显：解码器和过滤器需要处理每个包，而推理模型仅需要在通过过滤器的帧的一个小部分上运行。

昂贵的硬件解决方案。部署更多的硬件进行解码是缓解瓶颈的一种直接但昂贵的方式。考虑到 1080p 25FPS 的视频流，专用的解码器硬件（例如 Kiloview DC230[10]）每个流的成本约为 62.5 美元。最先进的 NVIDIA A100[11] GPU 在 Azure 上每年每个流的成本为 144 美元。使用 Azure vCores CPU 每年每个流的成本为 132 美元。除了硬件成本之外，部署专用解码器或更多带有 CPU / GPU 的机器还带来额外的通信开销和大量的工程工作。以本工作的系统为例，为了分析 1000 个摄像头而每年额外花费超过 10 万美元对于大多数组织来说是不可行的。因此，本文寻求一种纯软件的解决方案。

定量条件。除了上述具体案例之外，现在给出一个定量条件，即解码是并发瓶颈的条件：$T_{inference} > (1-r)T_{decode}$，其中 $r \in [0,1]$ 是过滤率，$T_{inference}, T_{decode}$ 分别是推理模型和解码器的吞吐量。请注意，对于下游应用程序的不同分析频率，可以直接配置解码器按照固定频率进行解码[84]。过滤率则取决于视频内容和推理任务，作为一个经验参考，先前的研究[58,63,83] 报告称，各种视频推理任务的潜在过滤率约为 80-99%。在应用模型加速技术[54,85] 后，这个条件通常成立。

表 2.1　视频包门控与相关互补方法在四个设计目标上的比较

方法	减少解码	支持通用摄像头	支持离线视频	跨流协同
视频压缩	✓	✗	✗	✗
摄像头端帧过滤	✓	✗	✗	✗
服务器侧帧过滤	✗	✓	✓	✗
模型加速	✗	✓	✓	✗
PacketGame	✓	✓	✓	✓

2.1.2　设计空间

范围。本文关注视频推理工作负载的通用摄入阶段，即从接收视频到获取推理结果（见图 2.1）。在推理之后，下游应用可能以各种方式使用分析结果。下游应用中的潜在优化机会不在本工作的考虑范围内。

设计目标。本文有四个主要的设计目标：

• 减少解码。首先，必须减少解码开销，同时保持高推理精度。

• 支持通用摄像头。通用摄像头通常不支持二次编程。支持传统摄像头和新的商用摄像头对于现有的视频压缩和摄像头端帧过滤技术来说是费时的，甚至是不可行的。

• 支持离线视频。这些场景中，假设已经使用某种视频编解码器对离线存储的视频进行了编码。一个理想的包门控解决方案应该是编解码器无关的，并且不需要额外的转码开销。

• 跨流协调。用于大规模并发流的包门控策略应该具有全局优化视图和对于并发视频流的弹性可扩展性。

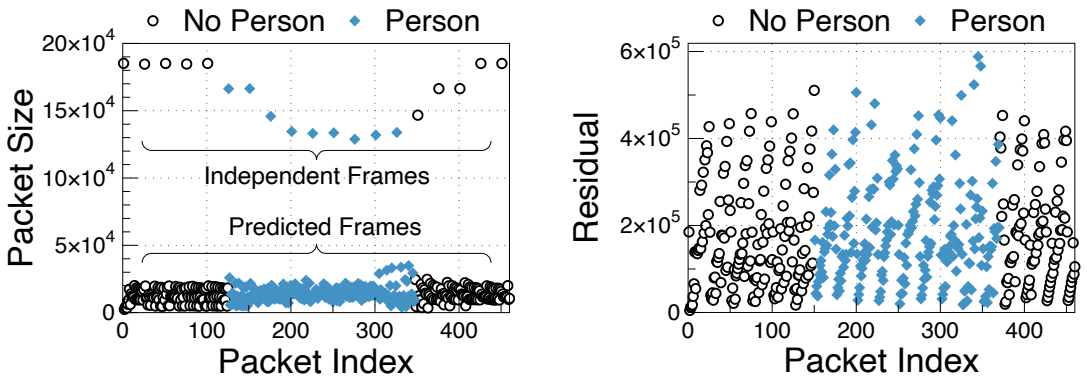
核心思想。为了解决发现的并发瓶颈并满足所有设计目标，本文提出了一个新的思路：在解码器之前为解析后的包添加一个名为包门控的过滤模块，该模块从所有流中选择一部分需要解码的视频包。从上述四个设计目标的角度来看，表 2.1 展示了本文提出的多流包门控方法 PacketGame 与现有方法相比之下的新颖性。请注意，本文提出的 PacketGame 与列出的现有方法没有冲突，并且可以作为它们的补充进行工作。

2.2　挑战

基于本文提出的数据包门控理念构建有效的方法涉及两个技术挑战，即非自适应数据包表示和低效的跨流协调。

2.2.1　非自适应数据包表示

由于数据包门控是在解析器之后执行的，因此只有一些视频数据包的元数据可用，例如视频编解码器、图像类型、数据包大小等。因此，原则上，本文的目标



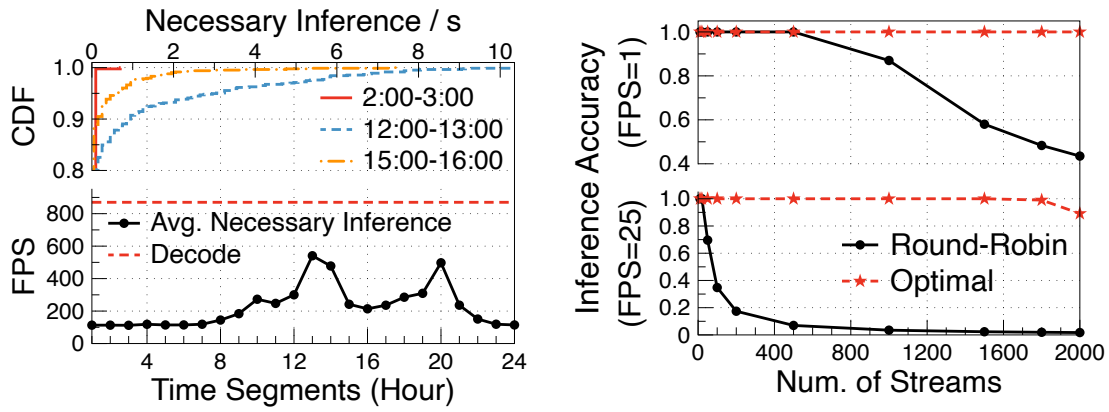
(a) 数据包大小的时序和非线性关联　　**(b) 残差特征的区分性差**

图 2.3　用于人物检测推理任务的数据包大小和残差特征的分布

是建立从数据包元数据到是否需要解码此数据包的映射。这项工作是实现这一目标的第一步。但从高层次的角度来看，数据包门控是一个数据包决策模型。在网络管理中已经探索过的许多想法可以作为设计参考，包括数据包分类（Packet Classification）[86-87] 和网络流量分类（Traffic Classification）[88-89]。为了在有限的网络资源下保证视频传输质量，相关工作提出了几种方法[90-91] 来选择性地丢弃通过度量如峰值信噪比（PSNR）和多尺度结构相似性（MS-SIMM）量化的视频帧。最近的工作[76] 还提出了一种基于残差的特征，可以使用视频数据包大小来对视频帧选择性执行超分辨率任务。然而，实验结果显示这些方法要么不能区分必要和冗余的数据包，要么不能适应多种推理任务。例如，将最大假阳性率设置为 10%，基于残差的选择结果只能达到 6.1% 的真阳性率，而本文提出的方法 PacketGame 达到了 76.6%。对于人物计数推理任务，图 2.3 中绘制了视频片段的数据包大小和残差特征[76]。要区分具有和没有检测到人的数据包需要对数据包大小进行时序和非线性表示。而必要和多余数据包的手工残差特征呈现高度难以区分的模式。端到端学习可以为数据和任务相关表示提供适应性[25,63,96]。因此，本文利用一个超轻量级的神经网络，通过学习实现基于数据包元数据对各种视频内容和推理任务的预测。另一方面，下游推理模型可以为数据包门控的预测表现提供在线反馈。因此，本文提出将元数据和反馈结合为视频数据包的融合表示。

2.2.2　低效的跨流协调

为了最大程度地提高在多个视频流上的整体分析并发性，还需要仔细协调跨流的数据包解码资源。使用现有的调度程序，例如轮询（Round-Robin）方法，会导致显著的性能下降。本工作在视频分析系统上进行了基准测试，使用了 1108 个流。图 2.4a 显示了一天中用于人物计数任务的必要推理的分布。对于给定的



(a) 一天中必要推理的分布　　**(b) 两种分析频率下的推理精度**

图 2.4　跨流协同调度性能存在大量优化空间

视频流，如果计数结果与最新数字不同，则将推理视为必要。人物计数任务的必要推理呈现出与常识一致的两个峰值（早晨和晚上）。实验结果表明，如果可以完美地从所有流中识别必要的数据包（最多为 540.8 FPS），解码能力（870 FPS）实际上是完全足够的。如图 2.4b 所示，与最佳的跨流策略相比，轮询方法在流的数量增加时性能骤降，其原因在于它并没有考虑多个流解码的必要性。例如，给定 25 FPS 的分析频率和 90% 的目标准确率，最佳策略支持 2000 个并发流，而轮询方法只能支持 30 个流。作为一个在线决策过程，需要在所有流之间使用有限的解码资源仔细权衡探索（Exploration）和利用（Exploitation）。因此，本文设计了一种组合算法，考虑数据包门控的置信度（即选择概率）和异构的解码开销作为流状态。

2.3　问题定义

本节定义了多流数据包门控问题，并分析了其算法结构。本节提出了一个具有理论性能保证的框架，并确定了从理论到实践的关键差距。

给定 m 个并发视频流，在每一轮 t 下，解码资源预算为 B，需要选择从到达的 m 个数据包中解码的子集。定义 $c_{t,i}$ 为在轮次 t 解码流 i 的数据包的成本。为了直观理解这些变量的实际意义，将以本工作开发的视频分析系统作为一个示例。本工作开发的系统中有 m=1000 个来自 IP 摄像头的并发 RTSP 流（拍摄频率为 25 FPS）。如果将一秒分为 25 轮，则在每一轮会收到 1000 个数据包。常见的视频编解码器（例如 H.264 和 VP9）有两种图片类型（Picture Type）的编码帧，独立帧（I-frame）和预测帧（P/B-frame）[84]，它们的解码成本是不同的。本工作使用的边缘服务器的计算资源支持在每一轮解码 11 个 I-frame 数据包或 32 个 P/B-frame 数据包。令 $x_{t,i}$ 表示在轮次 t 时第 i 个视频流信息的特征向量，它包括数据包大小和图片类型。自然而然地，与先前的工作[58,63,83]一样，本文假设是

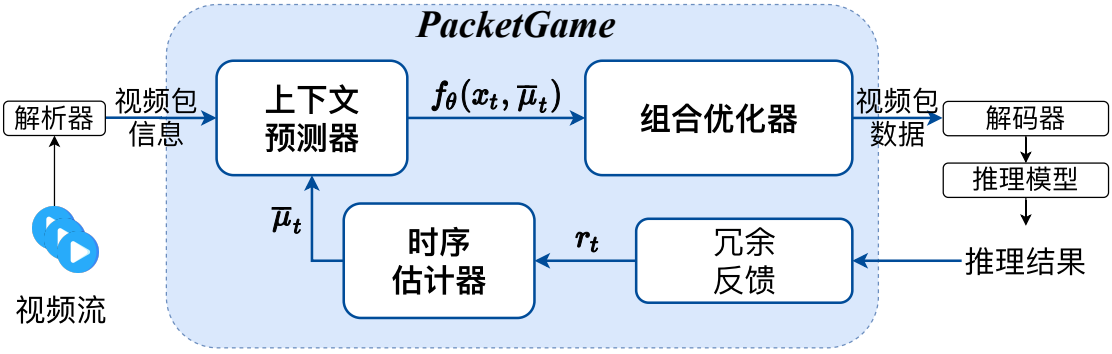


图 2.5　PacketGame 框架概览

否运行冗余推理的在线反馈是可用的。定义一组伯努利变量 $r_{t,i}$ 作为在轮次 t 时来自视频流 i 的数据包的冗余反馈。例如，将后端推理模型视为异常行为的检测器，如果解码的帧返回为“正常”，将反馈设置为 0；如果返回为“异常”，将反馈设置为 1。

本文研究的包门控问题的优化目标是最大化必要解码的数据包数量：

$$\max \sum_{t=1}^{T} \sum_{i \in S_t^{\mathcal{A}}} r_{t,i} \quad \text{s.t.} \quad \forall t \in [T], \quad \sum_{i \in S_t^{\mathcal{A}}} c_{i,t} \leqslant B, \tag{2.1}$$

其中 $S_t^{\mathcal{A}}$ 表示算法 $\mathcal{A}$ 在轮次 t 选择的数据包集合。最大化这个目标函数意味着选择更多必要解码的数据包，从而带来更少的推理精度损失。请注意，在实时操作中，除非解码每一帧，否则无法知道假阴性（本应解码但未解码的帧）情况。建立一个并行流水线，定期解码所有帧并验证召回率（类似于 LiveNet 中的快慢路径设计[97]）是对本研究使用的选择性反馈的一种有前景的补充方法。

框架概述。为了解决上述优化问题，本研究借鉴多臂赌博机（Multi-Armed Bandit）理论[55-56]，并提出了一个名为 PacketGame 的框架。如图 2.5所示，PacketGame 由三个主要模块组成。首先，使用一个时序估计器（Temporal Estimator）使用收集到的反馈历史估计一个反馈期望 $\bar{\mu}_t$。其次，使用一个上下文预测器（Contextual Predictor）结合了数据包元数据和计算的反馈期望的信息。本研究提出构建一个基于神经网络的预测器 $f_\theta(x_t, \bar{\mu}_t)$，用于预测各个视频流必要解码的置信度。第三，鉴于所有流的置信度和解码成本，需要解决有约束的组合优化问题。优化器返回最终要解码的数据包 $S_t^{\mathcal{A}}$。解码器将选定的帧推送给到推理模型，推理输出用于计算冗余反馈 r_t，时序估计器则在新的反馈到达时更新。

经理论分析，PacketGame 框架在理论上具有良好的性能保证，即有界的在线遗憾（Online Regret），详见第 2.4.4节。然而在实际系统中，仍存在如下两个必须通过仔细设计填补的技术性差距。

（1）元数据特征嵌入。上述形式化中，假设已经给定了一个有效的特征向量 x。然而在实践中，需要设计如何嵌入视频数据包的元数据[98]。以数据包大小

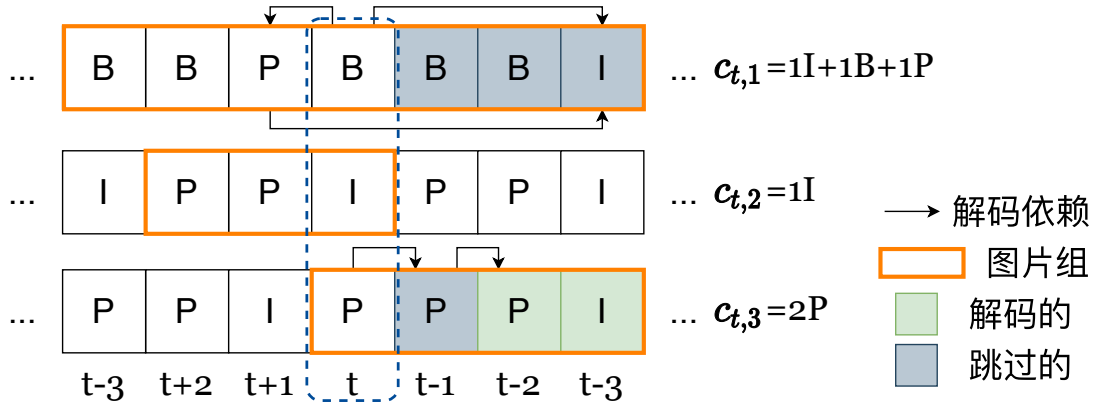


图 2.6　多个视频流中数据包的动态解码成本

（Packet Size）为例，它取决于许多配置，包括编解码器算法、比特率、图片类型等。嵌入先验知识（例如，I 帧和 P/B 帧具有不同包大小的模式）作为归纳偏见（Inductive Bias）对于基于神经网络的学习性能至关重要。

（2）异构的解码开销。如果解码成本是均匀的，那么定义的有约束优化将很容易解决，因为贪心算法即是最优的。解码不同图片组中的视频数据包会产生异构的开销。如图 2.6所示，当前数据包的成本取决于图片类型、图片组大小、解码依赖关系和先前的决策。对于第一个流，解码当前的 B 帧（双向预测图片）数据包取决于图片组中的第一个 I 帧和接下来的 P 帧。由于假设第一个 I 帧被跳过（未解码），第一个流的当前成本是 $1I + 1B + 1P$。而对于第二个流，成本是 $1I$，因为当前 I 帧没有解码依赖。对于第三个流，需要依赖第一个解码的 P 帧，导致解码成本为 $2P$。包门控策略需要权衡各种情况，例如解码当前的 P 帧还是等待下一个 I 帧，特别是当图片组很大时（通常在实时流媒体应用中）。

2.4　PacketGame 设计

本节介绍 PacketGame 框架中时序估计器、上下文预测器和组合优化器的设计，并证明了整体算法的性能保证。

2.4.1　时序估计器

许多推理任务的必要性模式具有时间连续性。例如，异常事件一般在视频帧中会持续出现一段时间。再比如，在网络出现问题的那段时间，需要使用超分辨率模型对实时视频进行画质增强。因此，由推理模型返回的在线冗余反馈对于数据包门控是有用的。

冗余反馈。与先前的帧过滤工作[58,63,83] 一样，本文假设存在一个冗余度量（Redundancy Measurement）可用。这一假设是自然而然且普遍成立的：对于物体计数任务，如果推理结果与之前的结果相同，将其视为冗余；对于检测任务，如

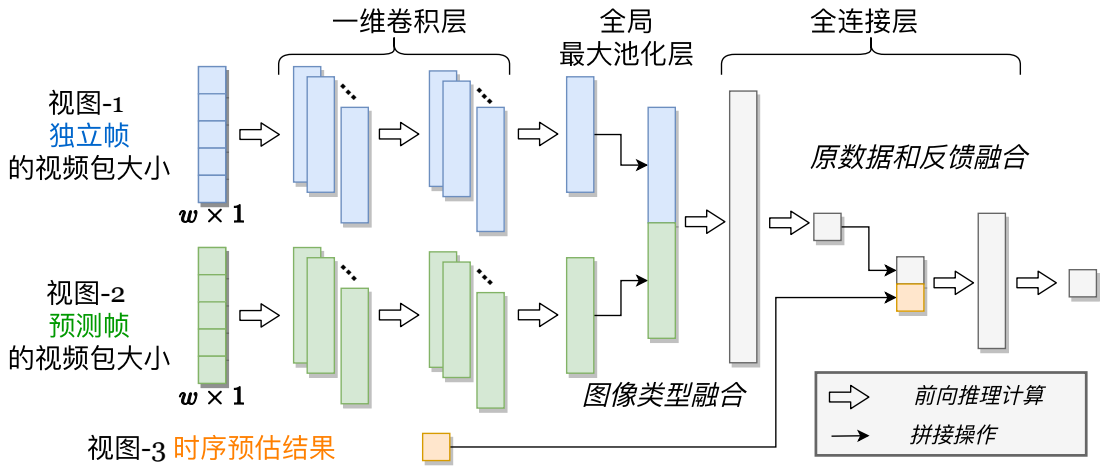


图 2.7　上下文预测器神经网络结构

果与之前检测出的边界框重叠度（IoU）高于阈值，则推理是冗余的；对于分类任务，可以将一些标签子集设置为冗余，或者检查标签是否发生变化。一旦收到关于所选数据包的冗余反馈，就可以为每个流记录历史反馈。

利用-探索（Exploitation-Exploration）权衡。本文提出设置一个长度为 w 的时间窗口，并通过 $\bar{\mu}_{t,i} = \frac{1}{w}\sum_{j=1}^{w} r_{t-j,i} + \sqrt{\frac{3\ln T}{2T_{t,i}}}$ 计算下一轮选择的概率，其中 $T_{t,i}$ 表示在最近的 w 轮中选择流 i 的次数，即 $T_{t,i} \leftarrow \sum_{j=1}^{w} 1(i \in S_{t-j})$。第一项，是为了利用。时态窗口内的平均奖励较大的流下一轮选择的概率较高。第二项是为了探索。直观地说，如果对某个流的尝试次数很少，那么应该以较高的优先级选择它。相关多臂赌博机理论结果[56] 表明，这种形式的利用-探索权衡对于在线决策具有良好的性能保证。消融实验（见第 2.5.2 节）也显示了所提出的时序估计器的有效性。

2.4.2　上下文预测器

除了利用历史冗余反馈外，当前数据包的元数据也可能有所帮助。例如，突发火灾会导致相对静态的帧发生显著变化，导致编码数据包的大小波动。与传统的基于元数据（如端口号和大小）设计有效规则的数据包分类任务不同[89]，元数据和推理冗余标签之间的相关性是非线性且复杂的。因此，本文设计了一个专门针对视频数据包的元数据的神经网络。

利用与视频编码相关的归纳偏差。由于不同的编码机制[84]，独立帧可以通过数据包本身解码，而预测帧需要参考其他数据包。例如，在一个图片组中，B 帧依赖于 I 帧和下一个 P 帧（参见图 2.6 中的第一个流）。因此，这两种类型的数据包的大小的范围和分布是不同的。另一方面，直观地说，这两种类型的数据包大小具有不同的含义。对于独立帧，数据包大小反映了当前帧的丰富程度。而对于预测帧，数据包大小反映了与依赖帧的相对变化。这些输入信息中的差异和互

补性启发了本文采用多视图学习（Multi-View Learning）[99]，这种方法在许多学习任务中取得了成功。本文使用单独的嵌入层来学习两种类型帧的数据包大小的特征。

时序特征嵌入。与时序估计器类似，本文设置一个长度为 w 的时态窗口。因此，输入设定为一个 w 维向量，记录了最近的 w 个数据包大小。本文利用一维卷积层和全局最大池化（Global Maxpooling）层作为特征嵌入块，这是时间序列分类的常见做法[100]。事实上，本文还尝试了其他类型的神经网络层，包括全连接、循环（Recurrent）层和 LSTM 层。作为概念验证，考虑到参数效率和实验性能，本文最终选择了一维卷积层。卷积层之后连接两个特征嵌入的视图，并使用一个全连接层来预测冗余标签。

元数据-反馈融合。时序估计器和神经网络都可以预测冗余的概率。因此可以在拼接它们的输出之后使用全连接层来融合它们的预测。图 2.7 展示了提出的上下文预测器的架构，它具有三个输入信息视图，即独立和预测帧的数据包大小以及时序估计的输出。消融实验表明，这种元数据-反馈融合带来了显著的改进。

多任务扩展。在同一视频流上运行多个推理模型是复杂分析系统（如智能城市[2]）的常见需求。本文的神经网络设计可以灵活扩展以支持多任务数据包门控，只需将最后一个全连接层的长度设置为任务数量。由于跨任务的相关性，实验结果（见图 2.11）显示多任务上下文预测器优于单任务。现有的多任务学习工作[101] 也报告了类似的结果。

参数优化。本文的上下文预测器的监督标签是冗余反馈，并将其归一化到 0-1 范围，这一设定与先前的工作一致[63,83]。本文采用二元交叉熵（Binary Cross-Entropy）损失，形式上为 $L(r,y) = -(r\log(y) + (1-r)\log(1-y))$，其中 r, y 分别表示真实和预测的标签。原则上，本文提出的神经网络可以通过任何基于梯度下降的算法进行端到端的优化。作为概念验证并考虑到实现的效率，在这项工作中，本文使用离线推理记录训练上下文预测器。然后，将训练好的权重转化为二进制运行时文件，并将其部署用于实时数据包门控（无在线参数更新）。在未来的工作中，将探索在线优化和领域自适应等学习相关的进展。

2.4.3　组合优化器

基于上下文预测器计算得到的流门控置信度，需要在解码预算下选择数据包的子集。PacketGame 是为了大规模的并发分析而设计的，其计算效率和可扩展性必须非常高。因此，本文提出首先根据置信度与成本的比率进行贪心选择数据包，即 $f_{\theta_t}(x_{t,i}, \bar{\mu}_{t,i})/c_{t,i}$。然后使用剩余的预算，尽可能多地解码当前优先的数据包所依赖的数据包。图片组中的解码依赖关系以有向图的形式存在，并且可以快速解析。这个任务特定的组合算法具有 $O(m\log(m))$ 的计算复杂度，并且对于

并发流的数量 m 具有线性可扩展性。这种基于贪心的优化器在一般的组合问题中不存在近似比。幸运的是，解码视频数据包的成本是近似可分的（Fractional）。基于这个特性，可以证明它具有 $1-\frac{c}{B}$ 的近似比，其中 B 是解码预算，c 是单帧的最大解码成本。

引理 2.1 (近似比)　对于近似可分背包问题，贪心算法具有近似比 $1-\frac{c}{B}$。

给定预测 $f_\theta(x,\bar{\mu})$ 作为价值, c_i 作为成本, 最大化在成本预算 B 下的累积价值是一个背包问题。根据价值与成本之比贪心地选择可能在一般情况下性能可能是任意差的。幸运的是,解码的成本是近似可分的,例如在图 2.6 中 $c_{t,1}=1I+1B+1P$。直观地说，当剩余预算低于下一个成本时，仍然可以解码部分帧。本文假设解码依赖帧所获取的价值遵循相同的比例。在这个实用的假设下，可以证明算法的最优近似比如下。

证明　设 $V_{\mathcal{A}}$ 表示算法返回的价值。考虑两个最优解：opt 为近似可分背包问题，opt_F 为严格可分背包问题。那么有 $V_{\mathcal{A}} \leqslant \text{opt} \leqslant \text{opt}_F$。定义 b 为剩余预算，r 为下一个物品的价值-成本比，c 为单帧的最大成本，因此 $b<c$。由于是根据价值-成本比贪心地选择物品，$V_{\mathcal{A}} \geqslant (B-b)r$。

$$\frac{V_{\mathcal{A}}}{\text{opt}_F} = \frac{V_{\mathcal{A}}}{V_{\mathcal{A}}+br} \tag{2.2}$$

$$= \frac{1}{1+\frac{br}{V_{\mathcal{A}}}} \tag{2.3}$$

$$\geqslant \frac{1}{1+\frac{br}{(B-b)r}} \tag{2.4}$$

$$= \frac{1}{1+\frac{b}{B-b}} \tag{2.5}$$

$$= 1-\frac{b}{B} \tag{2.6}$$

$$\geqslant 1-\frac{c}{B}. \tag{2.7}$$

因此，近似比为：$\frac{V_A}{\text{opt}} \geqslant \frac{V_A}{\text{opt}_F} \geqslant 1-\frac{c}{B}$。 ∎

在实践中，c/B 通常低于 0.05，这意味着相对于最优结果有超过 95% 的接近程度。

2.4.4　性能保证

基于提出的三个模块，算法 2.1 展示了整体算法，其中 $\bar{\mu}$ 表示时序估计器，f_θ 表示上下文预测器。时间窗口的长度是根据经验设置的，在实验中（图 2.13），窗口长度对性能有所影响。在每一轮中，算法 2.1 首先解析数据包特征（数据包大小和图片类型），并为每个流预测置信度 $p_{t,i}$。接下来，PacketGame 从 m 个流

算法 2.1　多流数据包门控算法

input 轮数 T，窗口长度 w
:
for $t = 1, ..., T$ **do**
　解析数据包特征 $\{x_{t,i}\}_{i=1}^{m}$;
　for $i = 1, ..., m$ **do**
　　$T_{t,i} \leftarrow \sum_{j=1}^{w} 1(i \in S_{t-j})$;
　　$\bar{\mu}_{t,i} \leftarrow \frac{1}{w}\sum_{j=1}^{w} r_{t-j,i} + \sqrt{\frac{3\ln T}{2T_{t,i}}}$;
　　$p_{t,i} \leftarrow \frac{f_\theta(x_{t,i}, \bar{\mu}_{t,i})}{c_{t,i}}$;
　end
　$P_t \leftarrow$ 按照 $p_{t,i}$ 的降序排列后的索引;
　$b_t \leftarrow 0, k \leftarrow 0, S_t \leftarrow \varnothing$;
　while $b_t < B$ **do**
　　$S_t \leftarrow S_t \cup \{P_t[k]\}$;
　　$b_t \leftarrow b_t + c_{t,i}$;
　　$k \leftarrow k + 1$;
　end
　使用剩余预算解码 S_t 中的所有数据包和 $P_t[k]$ 所依赖的尽可能多的数据包;
　接收冗余反馈 $r_{t,i}, \forall i \in S_t$;
end

中选择数据包并将它们发送到解码器。然后推理模型处理解码的帧并将冗余反馈返回给 PacketGame 。使用引理 2.1 和现有结果[55-56]，可以证明算法 2.1 的遗憾上界。

定理 2.2 (遗憾上界)　算法 2.1 在 T 轮中的遗憾至多为 $\tilde{O}(\sqrt{T})$。

本文将形式化后的问题视为一个 m 臂的组合上下文赌博机(Contextual MAB）问题，其中总轮数 T 是已知的。在第 $t \in [T]$ 轮，观察包含元数据和反馈估计的上下文：$\{x_{t,i}, \bar{\mu}_{t,i} | i \in [m]\}$。提出的的算法选择一个子集 S_t 的视频流（即一个赌博机的臂）进行解码，并接收反馈（奖励）$r_{t,S_t} = \{r_{t,i} | i \in S_t\}$。定义遗憾如下：

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(r_t^* - r_{t,S_t})\right], \tag{2.8}$$

其中 $r_t^* = \max_{S \subseteq 2^{[m]}} \mathbb{E}\left[r_{t,S}\right]$ 是第 t 轮的最大期望奖励。

证明　使用引理. 2.1, 得到一个 (α, β)-近似的预测器, 其中 $\alpha = 1 - c/B$, $\beta = 1$。基于这个预测器，并利用现有的结果[55-56,102]，可以推导出一个 $\tilde{O}(\sqrt{T})$ 的遗憾上界。∎

这样的理论保证对于实际应用非常重要。实验结果表明了算法 2.1 的有效性，例如在相同预算下，实现 2.1-4.8 倍的并发，同时保持超过 90% 的推理精度。

其他可能的设计选择。原则上，任何在线决策方法都有潜力解决数据包门控问题，例如深度强化学习（Deep Reinforcement Learning）[103]。深度强化学习在许多网络和资源管理任务中取得了成功[104-105]。然而，由于其组合性质，行动

表 2.2　数据集和推理任务总结

数据集	视频来源	推理任务
Campus1K	IP 摄像头	人数统计 (PC) 异常检测 (AD)
YT-UGC	离线视频	超分辨率 (SR)
FireNet	移动摄像头	火灾检测 (FD)

空间具有指数复杂性。此外，对固定观察空间和行动空间的要求使得深度强化学习方法缺乏伸缩弹性。例如，当并发流的数量发生变化时，需要重建和重新训练深度神经网络。另一方面，这项工作在不考虑排队的情况下定义了数据包门控问题，并为每轮设置了固定的解码预算。调度具有两个正交维度的数据包，即时间和视频流，会更为复杂，这是一个可能的未来研究方向。

2.5　验证实验

实验使用真实分析系统和公共数据集对 PacketGame 原型进行各种视频推理任务的验证。实验的亮点如下：

• 与原始视频推理工作负载相比，PacketGame 节省了 79.3% 的解码预算，并在超过 90% 的推理准确度下实现了 4.8 倍的并发水平。

• PacketGame 在涉及的变量方面表现出强大的鲁棒性，包括训练大小、窗口长度、图片组大小和视频编解码器。

• PacketGame 在提高端到端并发性方面优于其他最先进的方法，并具有更广泛的适用性。

2.5.1　实验设置

实现。本工作基于 FFmpeg[84] 和 TensorFlow[106] 库实现了 PacketGame 。为了表明 PacketGame 的设计不依赖于特定的框架，本工作还基于 MindSpore[107] 完成了另一实现。PacketGame 使用 *av_parser_parse*2 API 解析二进制视频，并通过访问 *size* 和 *pict_type* 属性获得数据包大小和图片类型。PacketGame 使用 RMSprop 优化器对上下文预测器进行训练。如果没有专门提及，本节的实验使用相同的超参数：窗口大小为 5，2 个 32 个单元的卷积层，128 个全连接单元，2048 批大小和 0.001 的学习率。

数据集和推理任务。为了验证 PacketGame 的性能，实验选择了三个视频数据集，如表 2.2所总结。(1) *Campus1K*。该数据集包含从本工作在大学校园部署的 1108 台 IP 摄像头采集的 h265 格式视频。摄像头每小时以 10 秒的频率捕捉画面，为期 24 小时，总计产生 4432（$1108 \times 10 \times 24/60$）小时的视频。图 2.8显示

表 2.3　在目标推理准确度为 90% 的情况下的总体并发度提升

方法	预算节省 / 并发水平			
	人数统计	异常检测	超分辨率	火灾检测
时序	52.6%/2.3x	71.8%/3.6x	75.8%/4.1x	50.5%/1.9x
上下文	68.1%/2.9x	38.9%/1.7x	14.4%/1.1x	31.0%/1.5x
PacketGame	75.2%/3.6x	79.3%/4.8x	76.2%/4.3x	52.0%/2.1x

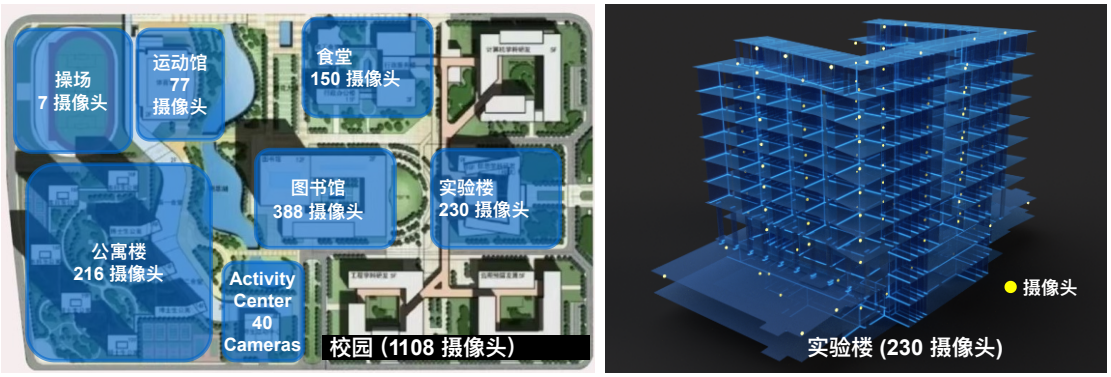
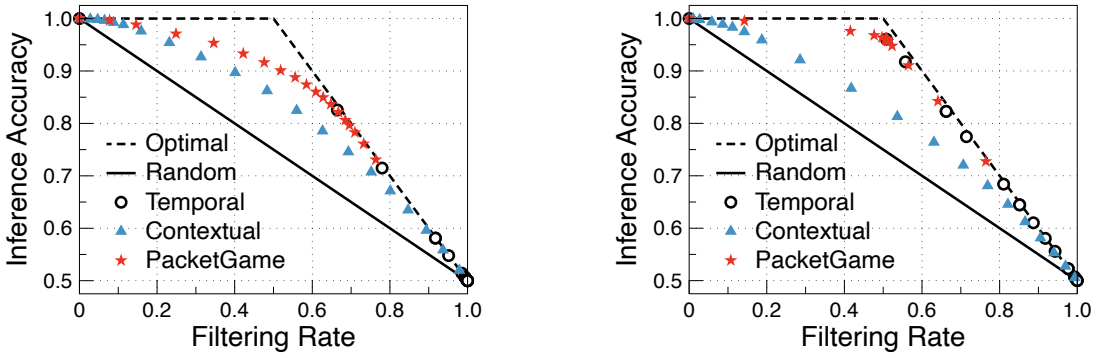


图 2.8　校园实时视频分析系统中 1108 台摄像头的分布情况

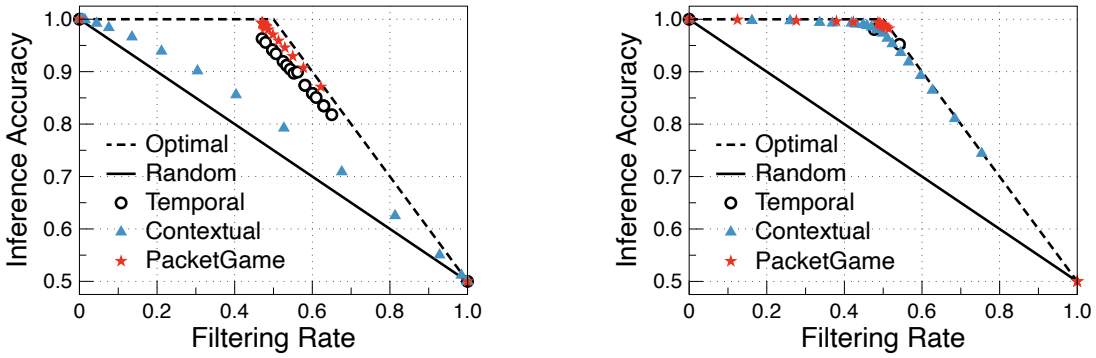
了校园内这些摄像头的分布情况。实验部署了一个人物检测模型（PC）和一个基于姿势的动作分类模型（AD）进行异常检测。(2) *YT-UGC*[108]。这个大规模数据集包含由 YouTube 用户生成的 1179 个 h264 格式的视频。为了模拟由带宽问题引起的视频质量波动，实验手动使用较低的比特率重新编码部分视频片段。该数据集涵盖了多样的内容和视频质量。对于推理任务，实验在视频片段上部署了一个超分辨率模型[109] 以提高视频质量。(3) *FireNet*[110]。该数据集包含由手机拍摄的 47 个具有火灾和 17 个没有火灾的视频。由于原始视频片段只包含有火灾或没有火灾的帧，实验随机将火灾片段插入没有火灾的视频中。该数据集为火灾检测提供了一个具有挑战性的场景。对于这个推理任务，实验部署了一个火灾检测模型[110]。

基线方法。 PacketGame 是首个用于视频数据包门控的方法，因此实验考虑以下基线和 PacketGame 的削弱版本进行端到端比较：(1) 随机。在预算下随机选择要解码的数据包。(2) 时序。仅使用本文提出的时序估计器。(3) 上下文。仅使用本文提出的上下文预测器（删除时序视图）。对于为视频推理优化设计的补充方法，实验考虑了四种最先进的方法：(4) Grace[72]，一种视频压缩方法。(5) Reducto[58]，一种在摄像头上的帧过滤方法。(6) InFi[63]，一种在服务器上的帧过滤方法。(7) TensorRT[54]，一种模型加速方法。

设备。对于摄像头部署实验，实验使用一部手机（小米 Mi 5）。对于所有其他实验，实验使用一台运行 Ubuntu 20.04 的边缘服务器，配备 12 个 Intel Core i7-5930K CPU 3.50 GHz 和 1 个 NVIDIA TITAN X GPU。



(a) 人数统计　　(b) 异常检测



(c) 超分辨率　　(d) 火灾检测

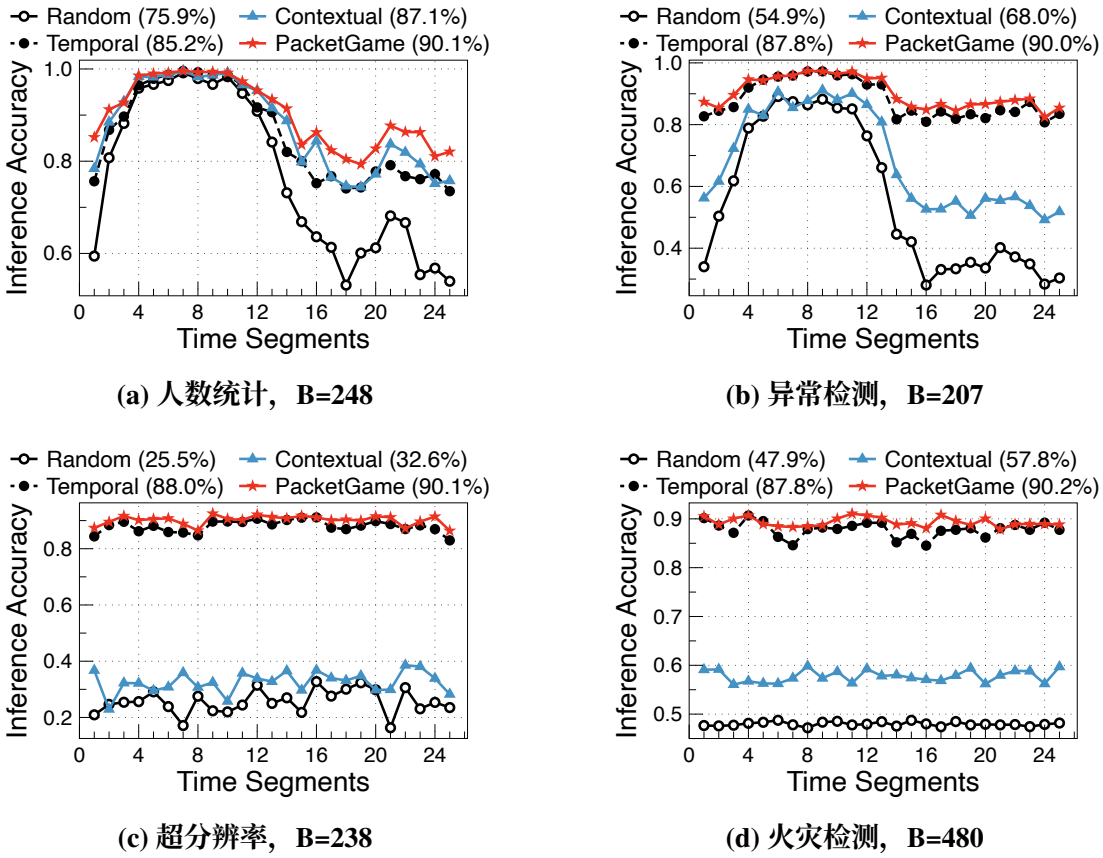
图 2.9　四个任务的离线过滤率和推理准确度

2.5.2　整体性能

在验证 PacketGame 的整体性能时，实验考虑了离线和在线两个角度。

离线。在离线验证中，实验采用正负样本比例为 1:1，并测量过滤率和推理精度。为了分析这些指标，实验调整预测置信度的阈值从 0.0 到 1.0，并绘制曲线以说明性能。最优曲线（标记为 Optimal）是使用真实标签计算的。形式上，设 a, r 表示推理精度和过滤率，TN 表示测试集中真负例（冗余数据包）的比例。最优曲线的公式是 $a = 1 - \max(r - TN, 0)$。如图 2.9所示，实验结果表明，时序估计器和上下文预测器均提供了有效的过滤性能。通过在 PacketGame 中结合这两个模块，实现了最佳且接近最优的性能。例如，在目标准确度为 90% 的情况下，PacketGame 在不同任务上实现了 51.8%、56.5%、57.7% 和 53.9% 的过滤率。最优过滤率为 60%，PacketGame 非常接近这一最优性能，展示了其在准确过滤冗余数据包方面的有效性。

在线。在线验证中，实验关注处理并发流，同时调整解码预算。上下文预测器使用每个数据集中随机采样的 80% 数据进行训练。在目标推理准确度为 90% 的情况下，实验报告了在同时处理 1000 个流时实现的解码预算节省程度。表 2.3显示了 PacketGame 实现的显著解码预算节省，从 52.0% 到 79.3%，同时仍保持 90%



(a) 人数统计，B=248　　**(b) 异常检测，B=207**

(c) 超分辨率，B=238　　**(d) 火灾检测，B=480**

图 2.10　在相同解码预算下随时间变化的在线推理准确度

以上的准确度。值得注意的是，提出的时序和上下文模块的组合是必要的，带来的进一步节省分别为 7.1%、7.5%、0.4% 和 1.5%。此外，实验报告了在相同解码预算（875 FPS）下，同时保持 90% 精度目标的最大并发水平。PacketGame 在所有四个任务上有效提高了并发水平，实现了 2.1× 到 4.8× 的并发。

时序估计器和上下文预测器对每个任务的贡献有所不同。虽然时序估计器对于人物计数任务的效果较差，但在提升超分辨率任务性能方面起主导作用。这种差异可以归因于超分辨率任务中视频的稳定时序模式。此外，实验分析了随时间推移的推理精度。图 2.10显示了不同时间段的精度，解码预算用 B 表示，平均精度在图例中表示。预算被设置为确保 PacketGame 的平均精度超过 90% 的最小值。实验观察到人物计数和动作检测任务的直观曲线，考虑到这些任务与人类活动之间的相关性，选择必要的数据包在白天（片段 16-20）相对于夜晚（片段 4-8）更具挑战性。相反，超分辨率和火灾检测视频的时序模式是随机模拟的，导致随时间推移精度相对稳定。

离线和在线验证全面验证了 PacketGame ，展示了其在准确过滤冗余数据包、显著节省解码预算以及在各种任务和操作条件下提高并发水平方面的有效性。

开销。PacketGame 作为视频推理流水线中的插件，在表 2.4中报告了其计算开销。实验考虑了三个指标：与设备无关的浮点运算数（FLOPs，由 TensorFlow

表 2.4　PacketGame 在边缘服务器和移动手机上的开销

模型	FLOPs	每帧延迟（边缘/移动）	每帧能耗（移动）
MobileNetV1	1137M	4/116 毫秒	410 毫焦
InFi (image)	351M	0.8/16 毫秒	15 毫焦
Reducto (area)	N/A	0.9/20 毫秒	22 毫焦
PacketGame	5K	7/154 微秒	<1 毫焦

分析器 API *float_operation* 进行分析），每帧的延迟和每帧的能耗（在移动手机上测试）。实验结果显示，与轻量级模型（MobileNetV1）、边缘服务器帧过滤器（InFi[63]）和摄像头帧过滤器（Reducto[58]）相比，PacketGame 的计算成本在数量级上小得多。PacketGame 具有 5K FLOPs，仅为 MobileNetV1（1137M）的 0.004%。对于延迟，PacketGame 每帧的成本为 7 微秒，比 MobileNetV1（每帧 4 毫秒）快 570 倍。尽管 PacketGame 不是为摄像头部署设计的，但在移动手机上运行的成本仅为 154 微秒，能耗小于 1 毫焦。作为参考，InFi 和 Reducto 在相同的移动设备上每帧分别消耗 15 和 22 毫焦。因此，原则上，在摄像头上执行数据包门控也是可行的。

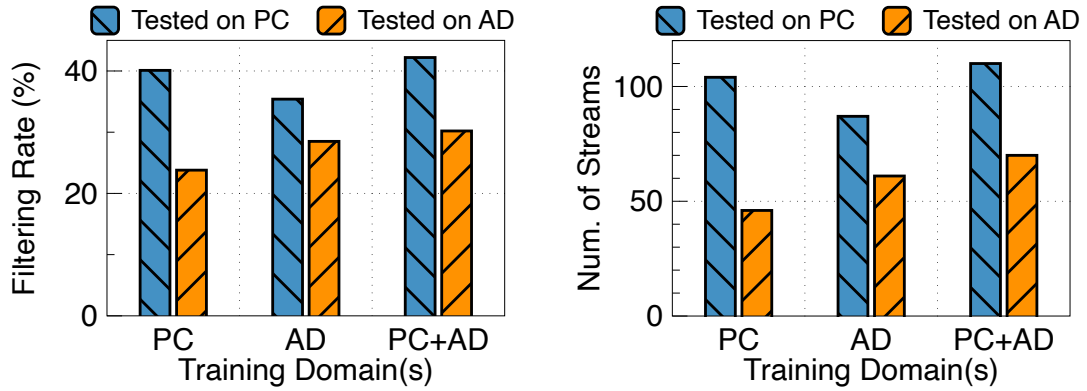
2.5.3　微基准测试

本小节的实验探讨了 PacketGame 设计中涉及的变量的影响。

多任务扩展。为了增强上下文预测模块的功能，本工作已经扩展了 PacketGame 的设计以支持多任务数据包过滤。对于这个扩展，实验考虑了两个推理任务，即人物计数和动作检测，在 Campus1K 数据集上，将它们视为独立的领域。如图 2.11所示，实验观察到直接利用在其他领域上训练的上下文预测器会导致性能下降。具体而言，人物计数的过滤率降低了 16.3%，动作检测的过滤率降低了 6.9%，同时人物计数的并发流量减少了 58 个，动作检测减少了 26 个。然而，当使用多任务扩展预测器时，该预测器利用了跨任务共享的表示[101]，实现了改善的性能。多任务扩展预测器展示了人物计数任务的过滤率提高了 2.1%，动作检测任务提高了 1.7%，人物计数任务的并发流量增加了 6 个，动作检测任务则增加了 9 个。

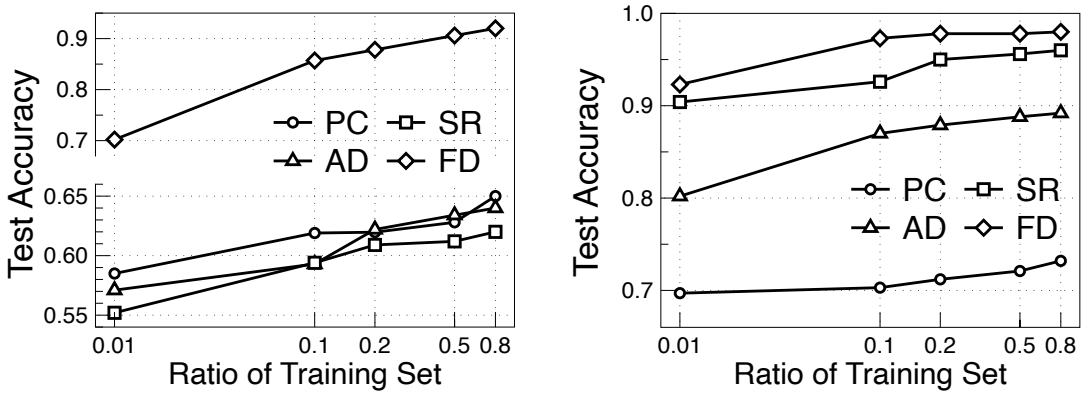
这种性能改善可以归因于在多个任务上共享学习的好处。通过同时在多个任务上训练上下文预测器，模型可以利用在领域之间共享的有用表示。这种共享表示学习增强了模型的整体学习能力，并促使人物计数和动作检测任务的更好性能。

对训练样本大小的敏感性。训练样本的大小对构建 PacketGame 的效率有显著影响。为了验证这种敏感性，实验随机采样了不同比例（0.01、0.1、0.2、0.5、0.8）的数据，并在相同的测试集上验证了分类精度。如图 2.12所示，测试精度随



(a) 离线过滤率　　(b) 在线并发度

图 2.11　上下文预测器的多任务扩展效果

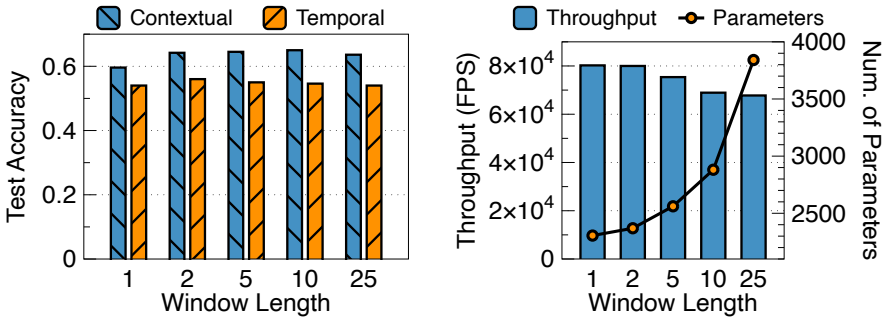


(a) 上下文预测器　　(b) PacketGame

图 2.12　不同大小的训练样本对准确度的影响

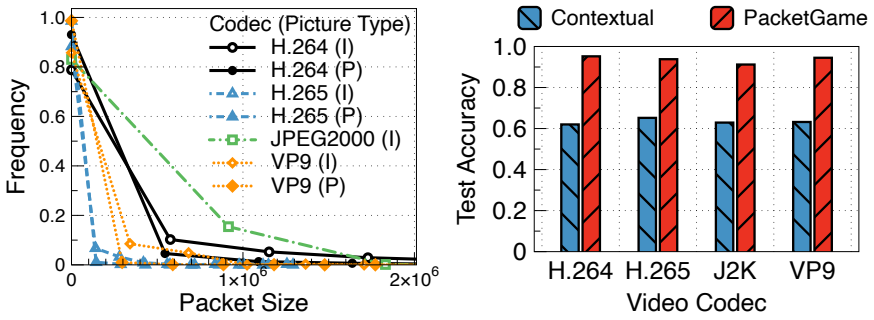
着训练样本大小的增加而一致提高，表明训练样本大小与本文提出的上下文预测器的有效性之间存在正相关关系。除了仅使用 1% 的样本进行训练的极端情况外，上下文预测器（不考虑时间视图）和完整的 PacketGame 模型都展示了它们从可用数据中进行有效学习的能力。这些发现强调了为了有效地训练 PacketGame，需要相对充足的数据量，因为这使得模型能够学习多样的模式，并对未见的测试数据进行良好的泛化。

窗口长度参数的影响。 PacketGame 中的窗口长度参数在决定数据包过滤性能和计算效率方面起着至关重要的作用。为了研究其影响，实验在人数统计任务上使用不同的窗口长度进行了测试。结果如图 2.13所示，显示出随着窗口长度的增加，上下文和时间模块的性能最初提高，然后开始下降。同时，随着窗口长度的增加，计算吞吐量减少。因此，精度和效率之间存在权衡。实验确定窗口长度为 5 在精度和效率之间取得了良好的平衡，成为默认选择。然而，值得注意的是，最佳窗口长度可能会根据具体的应用和要求而变化。可以进行进一步的探索和微调窗口长度参数，以满足不同场景和性能目标的要求。



(a) 学习性能　　**(b) 计算效率**

图 2.13　不同窗口长度的影响



(a) 数据包大小直方图　　**(b) 学习性能**

图 2.14　不同视频编解码器的影响

视频编解码器。由数据包过滤处理的输入流是由视频编码器生成的。为了研究不同视频编解码器对 PacketGame 的影响，实验使用三种不同的编解码器[84]（H.265、JPEG2000 和 VP9）对 YT-UGC 数据集中的原始 H.264 视频进行重新编码。图 2.14a展示了不同编解码器之间特征（图片类型和数据包大小）分布的明显差异。值得注意的是，由于 JPEG2000 编解码器只产生具有独立帧的流，PacketGame 的上下文预测器对于这个特定编解码器移除了独立帧的视图。实验结果表明，PacketGame 在所有测试的编解码器上都表现出鲁棒的性能（91.2-95.2% 的测试准确度），展示了其在实践中对各种常用视频编解码器的多功能性和适应性。

对于训练大小的敏感性、窗口长度参数的影响以及在不同视频编解码器上的性能的测试，为了在实际应用中灵活使用 PacketGame 提供了指导。通过理解和利用这些因素，开发者可以优化 PacketGame 的性能以满足其特定的要求和视频分析环境。

极端情况。为了研究 PacketGame 的能力和局限性，进行了考虑系统边界的两个极端情况的实验。

(1) 极低比特率。在这种情境下，实验使用极低的比特率（例如 100K）对 1080p 视频进行重新编码。实验观察到，在如此低的比特率下，大多数任务的数

表 2.5　PacketGame 与基线方法在人数统计任务上的对比

方法	过滤率	视频流路数
原始推理任务	0%	1
TRT	0%	30
TRT+Grace	0%	30
TRT+Reducto	78.4%	162
TRT+InFi	85.1%	35
PacketGame	79.3%	5
TRT+PacketGame	79.3%	169

据包大小信息变得难以区分。因此，在 PacketGame 的上下文预测器中，数据包的两个视图不再提供有意义的信息，而往往生成接近随机的猜测。然而，这种限制对于依赖推理结果的时序估计器没有影响。因此，即使在极低比特率的情况下，PacketGame 仍然可以通过利用时序估计器进行准确的选择而有效地运行。

(2) 极大图片组。在实时流应用中，常常会遇到图片组长度异常大的情况，比如 300。实验结果表明，在这些情况下，独立数据包的视图变得不太有效，因为在如此长的图片组中，观察不经常发生变化。然而，实验发现 PacketGame 利用的另外两个视图，即上下文预测器和时序估计器，不受极大图片组长度的影响。因此，PacketGame 的整体性能在不同的图片组设置下保持鲁棒和稳定。

这些极端情况揭示了 PacketGame 在具有挑战性的情景中的适应性。虽然极低比特率可能会限制某些数据包视图的有效性，但数据包序列的时间相关性仍然提供有价值的信息。同样，在极大图片组长度的情况下，尽管独立数据包的视图可能不太具信息量，但其他视图仍然保持有效。这些发现突显了 PacketGame 在不同条件下的鲁棒性，并且表明本文将元数据和反馈结合起来的设计对于处理实际环境中常遇到的极端情况非常必要。

2.5.4　与视频推理优化方法的比较

为了展示本文提出的数据包过滤方法的独特性和有效性，实验将其与四种视频推理优化方法进行比较：Grace[72]、Reducto[58]、InFi[63]和 TensorRT (TRT)[54]。PacketGame 与这些方法互补：数据包过滤的优化空间与帧过滤[58,63] 重叠，并且与视频压缩[72] 和模型加速[54] 正交。尽管这些方法各自有助于改善视频推理的不同方面，但 PacketGame 的重点是提高端到端的并发性能。例如，TensorRT (TRT) 可以将推理速度从 27.7 FPS 提高到 753.9 FPS，从而实现 30 倍的并发性提升。另一方面，Grace 减少了解码成本，但不涉及帧过滤。因此，它的并发级别仍然受到推理速度的限制，即 TRT+Grace 也仅支持 30 个并发流。Reducto 虽然将并发流的数量从 30（仅使用 TRT）提高到 162（TRT+Reducto），但需要修改摄像头并且不支持离线视频。InFi 可以降低推理成本，但其并发瓶颈转移到解码模

块，结果仅增加了 5 个并发流。相比之下，本文提出的数据包过滤方法专门设计用于通过在解码之前选择要处理的数据包，降低解码器和推理模型的成本，从而提高并发级别。如表 2.5所示，PacketGame 在改善并发视频流数量方面优于这些现有方法。与原始方法相比，PacketGame 实现了 5 倍的并发性。当与 TRT 结合使用时，PacketGame 支持 169 个并发流，而无需对视频源进行任何修改。这突显了 PacketGame 在实现高并发性和可伸缩性方面的有效性，同时保持与现有视频推理方法的兼容性。

讨论。（1）安全审计：本文提出的方法展示了从解析的数据包元数据到是否执行推理模型的映射的可行性。虽然这一进展在效率和资源利用方面带来了显著的好处，但也引入了潜在的安全风险。具体而言，获得流元数据访问权限的攻击者可能会窃取涉及某些位置发生异常事件的隐私敏感推理结果。为了缓解这一安全风险，需要在进行视频推理时，优先保护 RGB 帧和数据包级元数据的机密性。通过确保视觉内容和相关元数据的机密性，可以防止未经授权的访问，并减轻恶意实体对敏感推理结果的潜在利用。

（2）模态扩展：除了支持视频数据包外，本文的设计具有扩展支持其他类型的数据包序列的潜力。最近的工作[63]探讨了将帧过滤泛化为输入过滤的可能性，并提出了一个全面的框架，使其能够过滤各种数据模态，包括音频、运动传感器信号和无线信号。这种模态扩展代表了 PacketGame 未来研究的一个可能方向。通过扩大支持的模态范围，可以创建一个更多功能、适应性更强的系统，满足更广泛的多媒体应用和场景。以统一的方式过滤和处理各种类型的数据流的能力为增强推理效率开辟了新机会。

2.6　小结

本章首先确定了端到端并发的瓶颈受到视频解码的限制，并给出了一个定量条件。为了提高视频分析的端到端并发度，本章提出了一种新的方法：视频包门控，这种方法补充了现有优化方法在解码效率方面的不足。本章开发了一个名为 PacketGame 的框架，用于多流视频包门控，它利用轻量级的时序估计器和上下文预测器来自适应地表示视频包，并使用一个组合优化器进行跨视频流的资源协调。本章从理论上证明了 PacketGame 的组合算法具有最优近似比且整体性能具有在线遗憾上界，并在一个包含 1108 个摄像头的真实系统以及公共视频上进行了四个推理任务的验证。实验结果表明，与原始推理方法相比，PacketGame 可以显著解码成本，实现更高的数据源并发性。

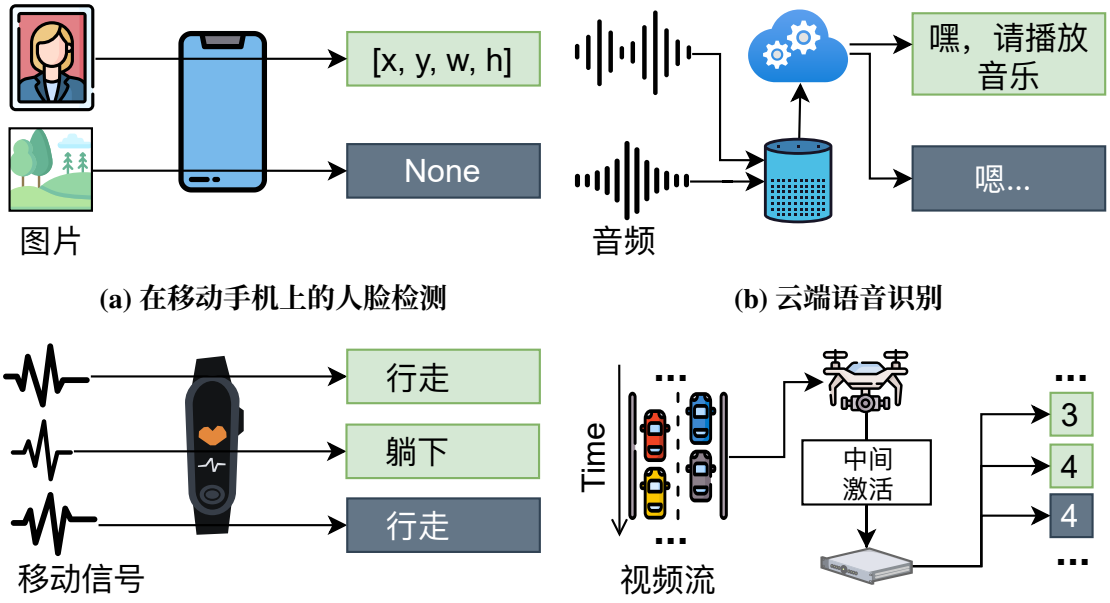
第 3 章　端边协同的冗余输入过滤

随着移动设备的计算能力增强和对实时传感器数据分析的需求不断增长，移动中心（Mobile-Centric）人工智能成为一个趋势[8,111-113]。例如，计算机视觉模型在设备上进行推理为用户带来了越来越丰富的实时增强现实应用[114]。再比如，通过端设备和边缘计算的配合，可以实时分析无人机拍摄的视频[115]。对于人工智能应用，尤其是对于资源有限的移动设备和延迟敏感的任务，模型推理的资源效率至关重要。然而，许多具有最先进精度的模型[116-118]在进行高吞吐量推理时的计算仍然过于密集，即使它们被卸载到边缘或云服务器上延迟依旧很高[119]。

为了实现资源高效推理，一个直接的方法是通过加速和压缩技术消除深度模型本身的冗余[120-126]。本项工作遵循另一系列方法[57-63]，过滤输入数据中的冗余。图 3.1显示了移动中心人工智能应用中输入冗余的四个例子。本文将这一系列方法称为输入过滤（Input Filtering），并将其分类为跳过和重用两类：(1) 跳过（SKIP）方法[57,62] 的目标是过滤掉会导致无用推理结果的输入数据，例如，对于人脸检测器，没有脸部的图像（图 3.1a）和对于语音识别器，没有有效命令的音频（图 3.1b）。现有工作[57] 训练了一个二元分类器称为 FilterForward，根据分类置信度设置阈值来过滤输入图像。(2) 重用（REUSE）方法[59-60] 过滤掉那些推理结果可以重复使用先前推理结果的输入，例如，相同动作的运动信号（图 3.1c）和具有相同车辆数的视频帧（图 3.1d）。现有方法[59]提出 FoggyCache，维护先前输入的特征嵌入和推理结果的缓存，对于新到达的数据，则在缓存中搜索可重用的结果。输入过滤通常作为资源有限的移动系统进行推理的必要预处理模块。此外，与模型优化相比，输入过滤在精度和效率之间提供了更灵活的权衡，例如，FilterForward 可以调整跳过的阈值，FoggyCache 可以调整重用缓存大小。尽管先前的工作为一系列应用设计了有效的输入过滤器，但仍存在两个重要且具有挑战性的问题尚未解答：

（1）输入过滤的理论可过滤性分析：并非所有推理任务都可以通过使用输入过滤来进行优化。有时，为了达到所需的精度，跳过/重用过滤器的代价可能比原始推理更高。因此，对于输入过滤，建立推理任务是否能够有效过滤的条件是至关重要的。先前的研究从应用导向的角度研究了输入过滤问题。这些工作从冗余观察开始，提出了定制的输入过滤解决方案，但没有进一步分析推理任务与输入过滤器之间的关系。由于没有理论指导和解释，尽管为特定任务提供了准确且轻量的输入过滤器，当前为其他任务设计输入过滤器仍然非常繁琐。

（2）鲁棒的特征表示可辨识性：实现有效过滤的一个的关键在于得到具有可



(a) 在移动手机上的人脸检测　(b) 云端语音识别

(c) 在智能手环上的人体动作识别　(d) 使用无人机和边缘计算进行车辆计数

图 3.1　移动人工智能应用中的输入冗余

辨识的特征表示，因为它直接决定了做出跳过决策和找到可重用结果的精度。最近的工作[58]表明，对于不同的任务，低级特征的可辨识性是不同的，例如，对于计数任务区域特征效果更好，而对于检测任务边缘特征效果更好。大多数现有的过滤方法利用手工设计的特征或预训练的神经网络作为特征嵌入[58-60]，并隐含地假设这些特征对目标任务足够有可辨识性。然而，移动应用通常在输入内容和推理任务上具有很高的多样性。对预训练或手工设计的特征的依赖导致对这些任务，特征表示的可辨识性不能得到保证。实验证明，对于动作分类任务，使用预训练特征的跳过方法[57]以及使用手工设计特征的重用方法[59]都不能有效工作。理想的特征表示应该以一种与任务无关且可学习的方式获得，而不是逐案定制。

为了回答这些问题，本文首先提供了输入过滤问题和有效过滤条件的形式化描述。然后，本文从理论上定义了“可过滤性”并通过比较推理模型及其输入过滤器的假设族复杂性[64-65]，对两种最常见的推理任务（分类和回归）的可过滤性进行了分析。本文并非为特定任务设计解决方案，而是提出了第一个端到端可学习的框架，统一了跳过和重用方法[57-59]。端到端的可学习性以一种与任务无关的方式提供了具有鲁棒可辨识性的特征嵌入，从而显著拓宽了适用范围。基于统一框架，本文设计了一个名为 InFi 的输入过滤系统，支持跳过和重用功能。除了图像、音频和视频输入外，InFi 在支持文本、传感器信号和特征图输入方面补充了现有技术。先前的方法通常是为特定部署设计的，例如推理卸载[58-59]。InFi 灵活支持移动系统中的常见部署，包括端上的推理、卸载和模型切分[127]。对包括 8 种输入模态、14 个推理任务和 3 种移动中心化部署类型的任务进行的全面验证表明，InFi 具有更广泛的适用性，并在精度和效率方面优于基线方法。

对于移动平台上的视频分析应用，InFi 相对于原始的车辆计数任务，可以实现高达 8.5 倍的吞吐量提升，并节省 95% 的带宽，同时保持超过 90% 的精度。

3.1　问题定义

本节形式化输入过滤问题，并提供了“有效”输入过滤器的条件。

输入过滤问题需要确定对于给定的推理模型，哪些输入是冗余的并且应该被过滤掉。首先，输入过滤问题的定义基于其目标推理模型。设 $\mathcal{X}, \mathcal{Y}$ 分别表示目标模型的输入空间和标签空间。定义 $c : \mathcal{X} \to \mathcal{Y}$，称为目标概念[128]，它为每个输入提供真实标签。然后，训练目标模型是为了在假设族（Hypothesis Family）[128] $\mathcal{H}$ 中寻找一个函数 h，使用一组训练样本 $S = \{(x_i, y_i)\}_{i=1}^{m}$，其中 $(x_1, ..., x_m)$，样本从 $\mathcal{X}$ 中独立采样，具有相同的分布 D，且 $y_i = c(x_i)$。使用上述符号，本节通过 $(\mathcal{X}, \mathcal{Y}, c, \mathcal{H}, D, S)$ 定义目标推理模型 h 的学习问题。图 3.2 中显示了已训练模型 h 的原始推理工作流程，它从 $\mathcal{X}$ 中获取输入并返回推理结果 $y \in \mathcal{Y}$。

接下来，给定已训练的推理模型 h，可以定义其冗余度量函数为：

定义 3.1 (冗余度量)　模型 h 的冗余度量 $f_h : \mathcal{Y} \to \mathcal{Z}$ 是一个函数，仅接受 h 的输出作为输入，并返回指示推理计算是否冗余的值。

这样的度量在实践中很常见。例如，基于人脸检测器的输出，返回没有检测到人脸的推理计算是冗余的（可以跳过），便可以设置得分 $z = 0$；否则，$z = 1$。形式上，$y \mapsto 1(|y| > 0)$，其中 y 是检测到的人脸的输出集，$1(\cdot)$ 是指示函数。对于重用情况，如果对于新数据的动作分类器的推理结果与之前缓存的相同，那么计算是冗余的，可以定义 $f_h(y) = 1(y \notin Y_{cached})$。注意，此冗余度量的定义不依赖于真实标签，而只依赖于推理模型的输出结果。图 3.2 中显示了冗余度量的工作方式。

给定推理任务 h 和冗余度量 f_h，如图 3.2 所示，学习输入过滤器被定义为在假设族 $\mathcal{G}$ 中寻找一个函数 g，训练样本为 $S' = \{(x_i, z_i)\}_{i=1}^{n}$，其中 $(x_1, ..., x_n)$ 是独立采样的, 具有分布 D', 且 $z_i = f_h(h(x_i))$。这个学习问题用 $(\mathcal{X}, \mathcal{Z}, f_h \circ h, \mathcal{G}, D', S')$ 表示，即 g 的目标概念是 f_h 和 h 的复合函数。

带有输入过滤的推理流程。一旦训练好了一个输入过滤器 g，推理工作流程如图 3.2 所示。输入过滤器 g 成为推理任务的入口，预测每个输入 x 的冗余得分 z。如果不是冗余的，推理模型 h 将直接在输入上执行。

在定义输入过滤器之后，本节给出了一个推理任务中“有效的”输入过滤器需要满足的条件。输入过滤器旨在平衡资源和精度：过滤更多的输入可以节省更多的资源，但也带来了更高的错误推理风险。

推理精度。通过使用输入过滤器，对于输入 x 的推理结果 y 通过执行 $h(x)$

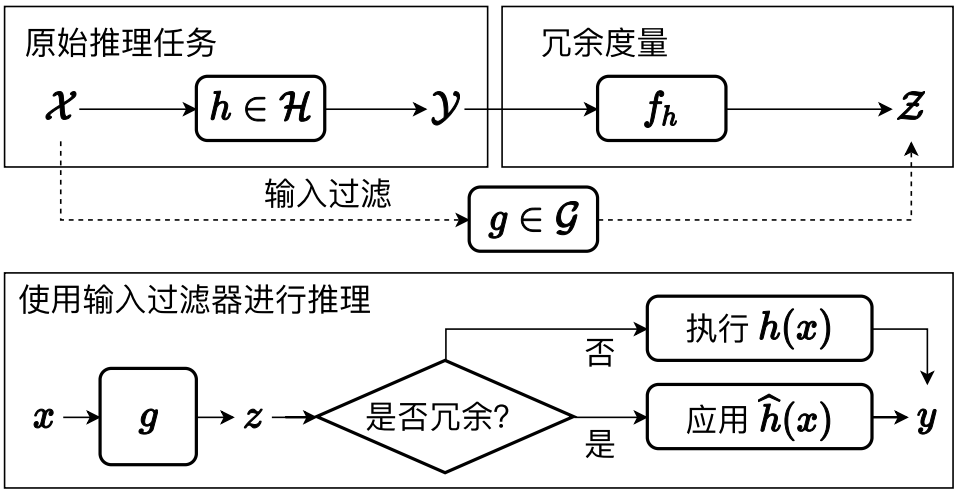


图 3.2　推理任务的输入过滤概览

或应用 $\hat{h}(x)$ 返回结果。遵循先前的工作[57-59]，结果 y 的正确性是指其与由 $h(x)$ 得到的精确推理结果的一致性。输入过滤器的推理精度 Acc 被定义为使用过滤器得到的推理结果正确的比率。

过滤率。过滤率，用 r 表示，定义为被过滤的输入的比率（即，通过应用 $\hat{h}$ 得到的结果的比率），这也是先前工作中考虑的一个重要性能指标[57-59]。

整体成本。带有输入过滤器的推理任务的开销需要考虑 g，h 和 $\hat{h}$。设 $C(\cdot)$ 表示某个函数的执行成本。对于计算成本（例如运行时），每个输入的平均成本从 $C(h)$ 变为 $C(g)+(1-r)C(h)+rC(\hat{h})$。通信成本（例如带宽）则取决于推理任务的部署方式。在端上推理不涉及通信，而边缘卸载[57-58] 和模型分割[127] 部署的整体带宽成本则变为原始成本乘以 $(1-r)<1$。

基于上述度量标准，如果满足如下两个条件，本文定义一个输入过滤器为“有效”：（1）足够准确：$Acc > T_{Acc}$，其中 T_{Acc} 是可接受推理精度的阈值。（2）减少开销：具有输入过滤器的整体成本更低。如果目标是减少计算成本，需要 $(C(g)+(1-r)C(h)+rC(\hat{h}))/C(h)<1$，即 $r > C(g)/(C(h)-C(\hat{h}))$；如果目标是减少通信成本，则只需要 $r>0$。

3.2　可过滤性分析

并不是所有的推理任务都能通过使用输入过滤技术进行优化。给定一个推理任务，是否存在有效的输入过滤器呢？为了回答这个问题，基于对输入过滤问题的定义，本节首先定义了推理任务的可过滤性。然后本节分析了在跳过设置中三种典型的推理情况的可过滤性。

3.2.1　可过滤性的定义

给定推理模型的学习问题 $(\mathcal{X}, \mathcal{Y}, c, \mathcal{H}, D, S)$ 和其输入过滤器的学习问题 $(\mathcal{X}, \mathcal{Z}, f_h \circ h, \mathcal{G}, D', S')$，为了简化分析，本文作出以下假设：(1) $D = D'$，即训练样本遵循相同的分布；(2) $S' = \{(x_i, z_i)\}_{x_i \in S}$，即两个学习问题在其训练样本中共享相同的输入，但它们在不同的标签下进行监督训练。推理模型 h 受到 $y_i = c(x_i)$ 的监督，而输入过滤器 g 受到 $z_i = (f_h \circ h)(x_i)$ 的监督。对于可过滤性的直观想法是，如果一个推理任务是可过滤的，那么其输入过滤器的学习问题的复杂性应该低于其推理模型的学习问题。形式上，本文将可过滤性定义如下：

定义 3.2 (可过滤性)　设 $Complex(\cdot)$ 表示假设族的复杂性度量。如果 $Complex(\mathcal{G}) \leqslant Complex(\mathcal{H})$，其中 $h \in \mathcal{H}$ 且 $(f_h \circ h) \in \mathcal{G}$，则称推理任务是可过滤的。

由于假设族不能仅基于输入和输出空间确定，本文将输入过滤器的目标概念 $f_h \circ h$ 的假设族作为 $\mathcal{G}$。

现在，可以通过利用计算学习理论（Computational Learning Theory）[64] 来表征给定推理模型的输入过滤器的理论可达精度和开销。已经证明，假设族越复杂，泛化误差的边界就越差。另一方面，神经网络的假设复杂性与参数数量呈正相关。例如，设 W, L 表示深度神经网络中的权重数量和层数，其 VC-维度[129]（假设复杂性的一种度量）是 $O(WL\log(W)$[130]。在相同的层结构下，参数越多，神经网络的推理开销就越高。泛化误差边界和参数数量对应于有效性条件，即前文的精度和效率指标。因此，如果一个推理任务是可过滤的，其输入过滤器具有较低的假设复杂性，因此有很大机会得到一个具有足够高精度和比推理模型更低开销的有效过滤器。接下来，本节将在不同情况下分析推理任务 h 及其输入过滤器 g 的假设族复杂性。

3.2.2　低置信度分类作为冗余

考虑一个推理任务，其中推理模型是一个返回分类置信度的二元分类器 h，而冗余度测量将置信度低于阈值 t 的分类结果视为冗余，即 $f_h(y) = \text{sign}(y > t)$。基于置信度的分类非常常见，例如用户验证。本文采用经验 Rademacher 复杂性[65]，用 $\widehat{\mathfrak{R}}_S(\cdot)$ 表示，作为复杂性测量，该复杂性测量得出以下泛化界[64]：

定理 3.1 (Rademacher 复杂性界)　设 $\mathcal{H}$ 为取值为 $\{-1, +1\}$ 的假设集合。对于任意 $\delta > 0$，以至少 $1 - \delta$ 的概率，对所有 $h \in \mathcal{H}$，以下关系成立：

$$R(h) \leqslant \widehat{R}(h) + \widehat{\mathfrak{R}}_S\mathcal{H}) + 3\sqrt{\frac{\log(2/\delta)}{2m}}, \tag{3.1}$$

其中 $R(h)$ 和 $\widehat{R}(h)$ 分别表示经验和泛化误差，m 是训练样本数。

该定理表明，假设集合的经验 Rademacher 复杂性越高，其泛化误差的界限

就越差。基于分类置信度的冗余度测量创建了两个平行于 $h=0$ 的超平面：它们之间的点被视为冗余，而它们之外的点被视为非冗余。因此，输入过滤器目标概念的假设集合的形式为：$\mathcal{G}=\{\text{sign}(h(x)(h(x)+b))\}$，其中 $h\in\mathcal{H}$，$b\in\mathbb{R}$。本文证明了以下引理，表明讨论的推理任务是**不可过滤的**。

引理 3.2　设 $\mathcal{H}$ 是一个取值为 $\{-1,+1\}$ 的二元分类器集合。对于 $\mathcal{G}=\{\text{sign}(h(h+b))\}$，其中 $h\in\mathcal{H}, b\in\mathbb{R}$：

$$\widehat{\mathfrak{R}}_S(\mathcal{G})\geqslant\widehat{\mathfrak{R}}_S(\mathcal{H}). \tag{3.2}$$

证明　根据定义，

$$\widehat{\mathfrak{R}}_S(\mathcal{H})=E_\sigma[\sup_{h\in\mathcal{H}}(\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i))]$$

和

$$\widehat{\mathfrak{R}}_S(\mathcal{G})=E_\sigma[\sup_{h\in\mathcal{H},b\in\mathbb{R}}(\frac{1}{m}\sum_{i=1}^{m}\sigma_i\text{sign}(h(x_i)(h(x_i)+b))],$$

其中 Rademacher 变量 $\sigma_i\in\{-1,+1\}$。固定 $b=2$，得到

$$\begin{aligned}\widehat{\mathfrak{R}}_S(\mathcal{G})&\geqslant E_\sigma[\sup_{h\in\mathcal{H},x_i\in S}(\frac{1}{m}\sum_{i=1}^{m}\sigma_i\text{sign}(h(x_i)(h(x_i)+2))]\\&=E_\sigma[\sup_{h\in\mathcal{H},x_i\in S}(\frac{1}{m}\sum_{i=1}^{m}\sigma_i\text{sign}(h(x_i))]=\widehat{\mathfrak{R}}_S(\mathcal{H}),\end{aligned}$$

其中使用了 $\text{sign}(h(x_i)+2)\equiv 1$ 的恒等式。 ∎

多类分类器可以被视为一组置信度评分函数，每个类别对应一个函数。上述引理同样适用于推导出使用这种基于置信度的冗余度测量的多类分类器也是**不可过滤**的结论。

3.2.3　类别子集作为冗余

考虑推理模型 h 作为一个多类单标签分类器，$\mathcal{Y}=\{y_1,...,y_l\}$。那么其假设集合 $\mathcal{H}$ 的形式为：$\mathcal{H}=\{\max(h_1,...,h_l):h_i\in\mathcal{H}_i, i\in[1,l]\}$，其中 h_i 返回第 i 类的概率。冗余度测量检查预测的类别是否属于特定子集，即 $f_h(y)=1(y\in\mathcal{Y}')$，其中 $\mathcal{Y}'\subseteq\mathcal{Y}$。实际应用中常常会只选择一部分标签使用。例如，在将预训练的通用目标检测器[131] 部署到用于交通监控的无人机时，应用只关心车辆和行人的标签，而将其他标签如动物和树木视为冗余。使用类别子集为基础的冗余度测量，输入过滤器目标概念的假设集合的形式为：$\mathcal{G}=\{\max(h_i):y_i\in Y'\}$。本文证明了以下引理，表明所讨论的推理任务是**可过滤**的：

引理 3.3　设 $\mathcal{H}_1, \cdots, \mathcal{H}_l$ 是 $\mathbb{R}^{\mathcal{X}}$ 中的 l 个假设集合，$l \geqslant 1$，且 $\mathcal{H} = \{\max(h_1, \cdots, h_l) : h_i \in \mathcal{H}_i, i = 1, \cdots, l\}$。对于 $\mathcal{G} = \{\max(h_i) : i \in J\}$，其中 $J \subseteq \{1, ..., l\}$：

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \leqslant \widehat{\mathfrak{R}}_S(\mathcal{H}). \tag{3.3}$$

证明　对于任意的 $j = 1, ..., l$：

$$\begin{aligned}\widehat{\mathfrak{R}}_S(\mathcal{H}) &= \frac{1}{m} \underset{\sigma}{E}[\sup_{x_i \in S} \sigma_i \max_{h_k \in \mathcal{H}_k} (h_k(x_i))] \\ &\geqslant \frac{1}{m} \underset{\sigma}{E}[\sup_{x_i \in S} \sigma_i \max_{j \in J} (h_j(x_i))] = \widehat{\mathfrak{R}}_S(\mathcal{G}).\end{aligned}$$

∎

其中的相等关系仅在所有 $x_i \in S$ 的情况下，输出概率最大的函数位于选定的子集中时成立。这意味着在没有推理精度损失的情况下，数据中的理论可过滤比率为 0。除非在这种极端情况下，一般可以认为学习输入过滤器的复杂度严格地更低。

3.2.4　回归上界作为冗余

考虑一个有界回归模型 h，其输出由 $M \in \mathbb{R}$ 限制，即对于所有 $x \in X$，都有 $|h(x) - c(x)| \leqslant M$（这里 c 是目标概念）。冗余度量检查返回的值是否大于阈值，即 $f_h(y) = 1(y > T)$。例如，人脸验证通常要求检测到的脸部坐标在指定范围内。然后，学习输入过滤器的目标概念变成了学习一个输出受到 T 限制的回归模型，其中 $T < M$。本文采用经验 Rademacher 复杂性，并有以下定理[64]：

定理 3.4　设 $p \geqslant 1$，$\mathcal{H} = \{x \mapsto |h(x) - c(x)|^p : h \in H\}$。假设对于所有 $x \in X$ 和 $h \in H$ 都有 $|h(x) - c(x)| \leqslant M$。则以下不等式成立：$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leqslant pM^{p-1}\widehat{\mathfrak{R}}_S(H)$。

由于 $M > T$，该定理表明 $\widehat{\mathfrak{R}}_S(\mathcal{G})$ 的上界比 $\widehat{\mathfrak{R}}_S(\mathcal{H})$ 的上界更紧。因此，可以确信所讨论的有界回归推理任务是**可过滤的**。

讨论（1）其他推理任务。分类和回归是最常见的推理任务，讨论的三种冗余度量被广泛采用[57,74,132]。然而，有一些推理任务很难衡量假设的复杂性，比如强化学习[133] 和结构化学习[134]。此外，它们的冗余度量通常定义不清晰。本文的问题形式化和分析方法是通用的，基于此可以在未来的工作中分析其他任务的可过滤性。（2）重用方法。对于重用方法，无法确定输入过滤器目标概念的假设集合。在这里，本文只给出一个必要条件：推理结果是离散的或可以离散化。例如，分类和计数模型返回离散输出。但检测模型的连续定位坐标不能直接重用，除非将高 IoU 的检测结果重用视为正确，这也相当于对输出进行离散化。

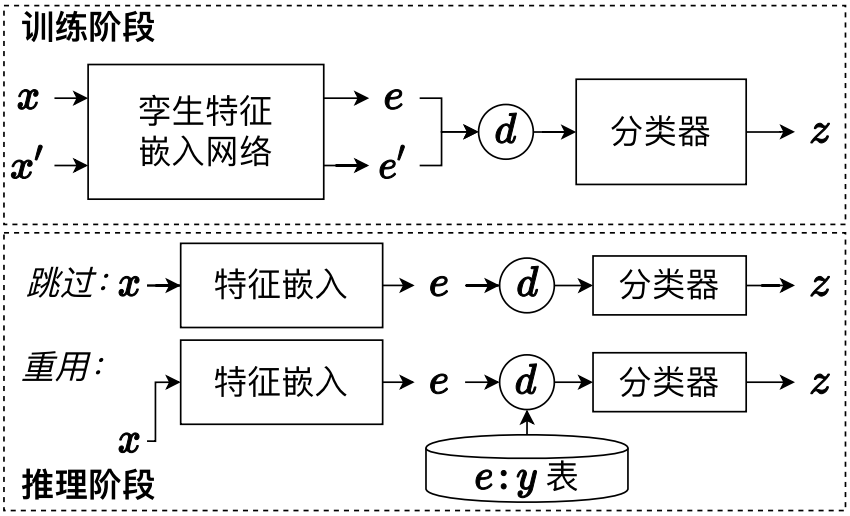


图 3.3　统一的端到端可学习的跳过和重用输入过滤框架

3.3　冗余输入过滤框架

本节首先提出了一个新颖的输入过滤框架，统一了跳过和重用方法。然后本节介绍了关键设计思想、端到端可学习性以及它带来的优势。

3.3.1　统一跳过和重用方法

本文基于以下思想统一了跳过和重用方法:

跳过等同于重用 $h(\vec{0})$ 输出为 *NONE* 的结果。

假设有一个全零输入 $\vec{0}$，显然其推理结果可以被解释为 NONE。然后，给定一个新输入 x，如果它在特征空间中与 $\vec{0}$ 相似，可以重用已缓存的 NONE 结果，即跳过推理计算。重用的关键是测量当前输入与先前缓存的输入之间的语义相似性。然而，直接基于原始输入准确地测量语义相似性是困难的。如图 3.3 所示，本文的框架首先计算每个原始输入的特征嵌入。取一对输入 x, x'，对它们对应的嵌入 e, e' 应用一个差异函数 d，并将结果输入给一个分类器，该分类器预测单个标量 z。在这个框架下，对于跳过，将 x' 固定为全零输入 $\vec{0}$，然后该过程退化为一个二元分类任务，以 x 作为输入并返回预测 z。通过这种方式，该框架统一了跳过和重用方法，只是对值 z 的解释上有所不同。对于重用，本文将 z 解释为两个输入之间的距离。对于跳过，本文将 z 解释为输入 x 不冗余的概率。

对于推理阶段，如图 3.3所示，跳过和重用过滤器只在差异函数 d 的输入上有所不同。(1) **跳过：**使用跳过方法进行推理与提供二元分类器相同。本文可以在预测的冗余得分 z 上设置阈值，以确定是否跳过推理。(2) **重用：**使用重用过滤器进行推理需要维护一个键值表，其中键是特征嵌入，其值是相应的推理结果。对于新到达的输入 x，训练好的特征嵌入网络返回其嵌入 e，并通过差异函数 d 和训练好的分类器计算 e 与缓存键之间的距离 z。然后，本文可以利用分类

算法，例如 K 近邻，获取可重用的缓存结果。

下面介绍本文的框架如何涵盖三种最先进的输入过滤方法[57-59]，这些方法将在后续的验证实验中用于对比。

子实例 1： FilterForward（FF）[57] 是一种用于图像输入的跳过方法。FF 使用预训练的 MobileNet 的中间输出作为特征嵌入。然后，它训练一个由卷积块组成的“微分类器”，对过滤进行二元决策。

子实例 2： FoggyCache（FC）[59] 是一种用于图像和音频输入的重用方法。FC 使用低级特征（图像使用 SIFT，音频使用 MFCC）并应用局部敏感哈希（Local Sensitive Hashing）进行嵌入。然后，FC 使用 L2 范数作为差异函数，并应用 K 近邻从先前缓存的结果中获取可重用的推理结果。

子实例 3： Reducto[58] 是一种用于视频输入的跳过方法的变种。它测量连续帧之间低级特征（像素、边缘、角、面积）的差异。如果两帧足够相似，Reducto 跳过当前帧并返回最近的推理结果。形式上，设 x 为当前帧，x' 为前一帧。Reducto 定义 $d(e,e') = (e-e')/e'$，其中 e, e' 是 x, x' 的低级特征。它使用一个阈值函数作为分类器，即 $1(d(e,e') > T)$。

3.3.2　端到端可学习性

为了获得对应用中多样的数据模态和推理任务具有鲁棒可辨识性的特征，本文框架的一个关键设计原则是端到端可学习性。端到端学习系统将复杂的处理组件转化为深度神经网络中的一致连接，并通过应用基于梯度的反向传播算法来优化整个网络[135]。端到端学习模型在包括自动驾驶[136] 和语音识别[96] 在内的各种任务上已经展示出最先进的性能。正如前面提到的，本文统一框架的主要组成部分之一是衡量两个输入之间的语义相似性。为了使框架端到端可学，本文采用了度量学习（Metric Learning）范式，其目标是在两个对象上学习一个任务特定的距离函数。度量学习范式将现有方法中使用的固定差异函数 d（例如欧几里得距离和 L2 范数）转化为一个端到端可学习的网络。基于度量学习范式，本文采用孪生网络（Siamese Network）结构[137] 用于特征嵌入，以支持两个输入和各种输入模态。孪生网络在处理两个不同输入时使用相同的权重来计算可比较的输出向量，并已成功应用于人脸验证[138]、行人跟踪[139] 等领域。可以通过将不同的神经网络结构应用于不同模态特征的孪生网络中，以端到端的方式学习，而不是定制手工设计或预训练特征模块。实验结果表明，端到端学习的特征对于人工智能应用中的各种推理任务具有鲁棒的可辨识性。

3.4　InFi 设计

基于提出的输入过滤框架，本节介绍了 InFi（INput FIlter）的具体设计，它支持跳过（SKIP）和重用（REUSE）功能，分别命名为 InFi-Skip 和 InFi-Reuse。InFi 的设计包括四个关键组件：特征嵌入，分类器，训练机制和推理算法。本节还讨论了在移动、边缘和云设备上在人工智能应用中多样化的 InFi 部署。

3.4.1　全模态特征网络

InFi 支持在移动应用中使用的六种典型输入模态进行过滤推理工作：文本、图像、视频、音频、传感器信号和特征图。本文提出了一系列模态特定的特征网络作为学习特征嵌入的模块。在设计这些特征网络时，本文主要考虑在移动设备上的资源效率。

文本模态（g_{text}）。文本被表示为一个整数序列，其中每个整数表示一个词项的索引。本文采用词嵌入层将序列映射到一个固定长度的向量，通过一个变换矩阵，并使用带有 Sigmoid 激活的全连接层来学习文本特征。

图像模态（g_{image}）。本文使用深度可分离卷积（Separable Convolution）[140]，表示为 $SepConv$，来学习视觉特征。$SepConv$ 是传统卷积的高效变体，它对每个特征通道单独执行深度空间卷积，并在所有输出通道上执行点卷积。然后，本文构建残差卷积块[141] $ConvRes$ 如下：

$$
\begin{aligned}
&ConvStep(x) = LN(SepConv(ReLU(x))),\\
&c_1(x) = ConvStep(x), c_2(x) = ConvStep(c_1(x)),\\
&ConvRes(x) = MaxPool2D(c_2(x)) + ConvStep(x),
\end{aligned}
$$

其中 LN 表示层归一化，$MaxPool2D$ 表示二维最大池化层。最后，本文通过两个残差块、一个全局最大池化层和一个 Sigmoid 激活的全连接层构建图像特征网络。

视频模态（g_{video}）。对于视频模态，需要表示空间和时间特征。给定一窗口的帧，本文为每一帧堆叠一个残差块，然后拼接它们的结果特征图。除了第一个残差块外，视频特征网络执行与图像特征网络相同的操作。

音频模态（g_{audio}）。本文考虑以一维波形或二维频谱图的形式输入音频，并使用与图像特征网络相同的结构从音频中学习特征。

传感器信号和特征图模态（g_{vec}）。运动传感器广泛用于移动设备，并在许多智能应用中发挥关键作用，例如，增强现实（Augmented Reality）中的陀螺仪[142]和活动分析中的加速器[143]。特征图是深度模型的中间输出，需要在涉及模型切分的任务中传输[127]。本文将这两种类型的输入视为具有固定形状的向量，并使用两个全连接层从向量中学习特征嵌入。

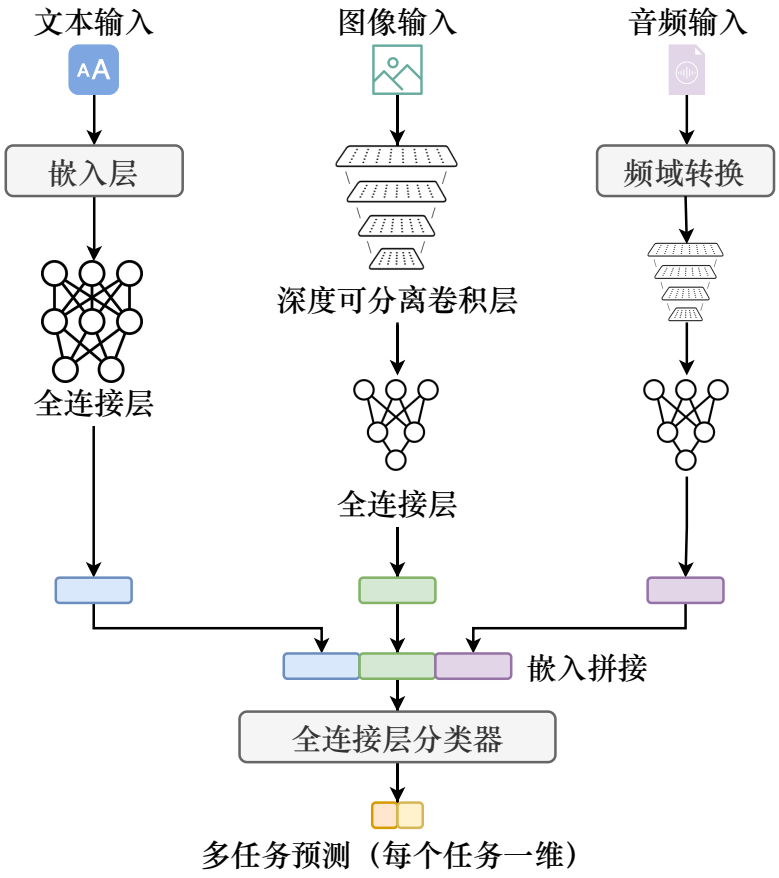


图 3.4　输入过滤的多模态和多任务扩展

对输入模态的灵活支持。本文的设计为移动中心的应用提供了对多样化输入模态的灵活支持。本文可以轻松地将先进的机器学习研究中的特定模态神经网络集成到框架中，作为特征网络块，以实现端到端的特征嵌入学习。

任务无关分类器。每个特征网络 $g_{modality}$，其中 $modality$ 属于 {text, image, video, audio, vec}，以 x 作为输入并输出嵌入 emb。本文在特征网络的最后一个全连接层之后添加了一个 Dropout 层以减少过拟合。按照孪生网络的现有设计[137]，本文使用绝对差异作为函数 d。设 emb_1, emb_2 表示两个输入 x_1, x_2 的嵌入输出。分类器定义为 $g_{cls} = \sigma(\sum_j w_j |emb_1^{(j)} - emb_2^{(j)}| + b)$，其中 $emb^{(j)}$ 表示嵌入向量中的第 j 个元素，σ 是 Sigmoid 函数。总之，输入过滤函数 $g : \mathcal{X} \to \mathcal{Z}$ 可以定义为 $g(x) = (g_{cls} \circ g_{modality})(x)$。通过适当的实现，输入数据的模态可以在不手动设置的情况下被自动检测。

3.4.2　多任务扩展

上述设计是针对单任务的，然而，在实际应用中同时运行多个人工智能模型是很常见的。本节将展示 InFi 的设计可以灵活地扩展到多模态和多任务推理任务。

多模态单任务。多模态学习旨在学习给定多个具有不同模态的输入的模型，这在自动驾驶等领域引起了越来越多的关注[136]。本文设计的模态特定特征网络和任务无关分类器天然地支持多模态扩展：对于每个模态 $mod \in \{\text{text, image, video, audio, vec}\}$，本文构建相应的特征网络 g_{mod} 来学习其嵌入，然后拼接生成的嵌入并将其发送到分类器 g_{cls}。

单模态多任务。在同一输入上部署多个模型来进行不同的任务分析是很常见的，例如，在同一视频流上检测车辆并分类交通状况。对于输入过滤，只需扩展分类器 g_{cls} 中最后一个全连接层的长度，每个任务增加一个维度。已有关于多任务学习的工作[101]证明了跨任务表示提高了学习性能。形式上，给定 t 个任务，已被证明所需的样本复杂度[144] 如下：

$$Complex(g_{mod}) + t \cdot Complex(g_{cls}). \tag{3.4}$$

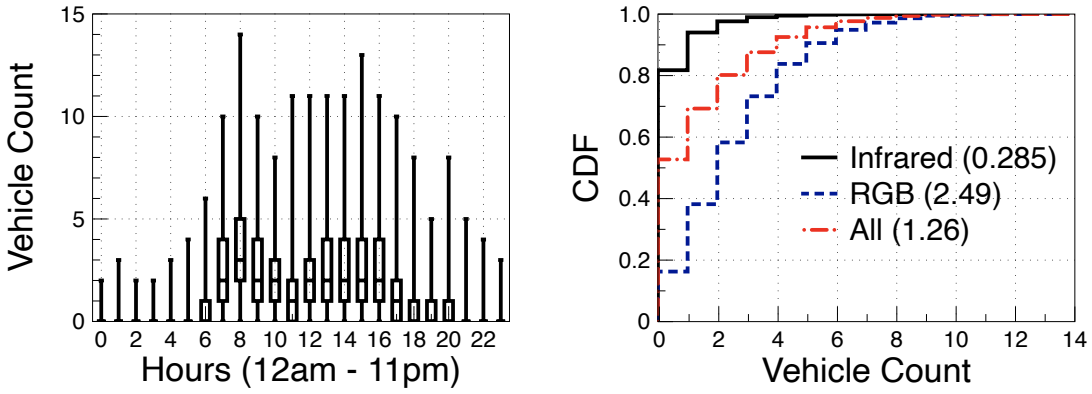
也就是说，与分别为每个任务学习过滤器相比，可以节省 $(t-1) \cdot Complex(g_{mod})$ 的样本复杂度。而本文的实验结果（图 3.14）也表明跨任务表示对于输入过滤是有益的。

多模态多任务。考虑一个需要为多模态和多任务过滤输入的一般情况，本文可以将上述两个扩展组合起来，如图 3.4 所示：对于每个输入模态，本文构建相应的特征网络并拼接生成的嵌入；而对于每个任务，本文构建一个多维分类器，每个任务增加一个维度，该分类器以连接生成的嵌入作为输入。与为不同推理任务部署独立的 InFi 的方式相比，本文提出的扩展节省计算并利用跨任务和跨模态表示的潜在优势。

3.4.3　训练与推理

InFi-Skip 和 InFi-Reuse 共享相同的模型架构,但具有不同格式的训练数据。(1) 训练 InFi-Skip 过滤器使用与训练二元分类器相同的范例。因此，它的训练样本为 $(x_i, f_h(h(x_i)))_{i=1}^n$，并使用二元交叉熵损失函数。在实践中，可以使用 h 的原始训练集或在服务 h 过程中收集的数据。由于 f_h 仅依赖于推理结果，监督标签可以自动收集。(2) InFi-Reuse 过滤器使用对比损失（Contrastive Loss）[145] 进行训练。给定一组输入及其离散推理结果，冗余度度量被定义为一对输入之间的距离度量。形式上，一个训练样本包含一对输入及其距离标签 $(x_i, x_j, 1(y_i \neq y_j))$。可以使用标准的反向传播算法，端到端地优化所有可训练的参数。

在线主动更新。与基准数据集不同，实际世界输入的分布，例如监控摄像头捕获的视频流，要窄得多，并且在线发生变化[146]。在基于视频的车辆计数应用中（详见第 3.5.1节的详细设置），本文探讨了随着时间推移而发生的分布偏移。如图 3.5a所示，车辆计数随时间变化。早上和晚上有两个明显的计数峰值，而夜



(a) 车辆计数分布　　**(b) 红外和 RGB 帧上的车辆计数的 CDF**

图 3.5　在线视频流中的数据分布变化

间结果保持在较低的数值上稳定。实验中注意到在光照条件改变时，拍摄的帧在红外（Infra-Red / IR）和 RGB 图像之间切换。RGB-IR 技术为摄像头提供了昼夜视觉能力，并得到了流行的商业传感器的支持。实验将所有帧分为红外和 RGB 子集，并在图 3.5b中绘制它们在检测到的车辆数量上的分布。实验结果上可以看到分布上存在明显的差异。因此，一个仅选择初始样本（例如，第一个小时内的帧）进行训练输入过滤器的离线训练策略会迅速导致性能下降。为了克服离线训练的适应性差的问题，本文采用了最低置信度[147]策略，以主动选择样本进行实时参数更新。现有工作[148]证明了主动学习的样本复杂性渐近小于被动学习。具体而言，本文设定一个周期长度和一个采样比例 $\beta\%$。然后本文在一个周期内对所有输入执行输入过滤器，并选择具有最低置信度（$|g(x)-0.5|$）的 $\beta\%$ 样本。对于 InFi-Reuse，本文将 $1-\theta$ 视为置信度分数。本文的实验结果（图 3.15）表明，在相同的训练样本数量预算下，这种主动策略明显优于离线策略。

在训练完 InFi 过滤器之后，本文使用算法 3.1将其整合到原始推理任务中。

InFi-Skip. 本文为 InFi-Skip 设置了一个冗余阈值，以确定是否跳过当前输入。如果跳过输入，InFi-Skip 将返回一个 NONE 结果，其解释取决于特定应用中的冗余度量。例如，在人脸检测中，NONE 表示未检测到人脸，在车辆计数应用中表示零车辆，而在语音识别中表示无意义的语音等。

InFi-Reuse. 为了重用先前的推理结果，InFi 需要维护一个缓存，其条目是输入嵌入和其推理结果的键值对。与之前的重用方法[59]相似，本文采用 K-近邻（K-Nearest Neighbors）算法来重用缓存结果。但是可能存在一个新输入与任何已缓存条目都不相似的情况，即缓存未命中。本文采用同质性 K 近邻（H-KNN）算法[59]来处理这个问题，该算法计算找到的 K 个最近邻的同质性分数 θ，并在同质性分数上设置阈值 θ_T 来检测缓存未命中。然后，本文可以使用诸如最少使用（Least Frequently Used）等策略通过算法 3.1中的 `replace` 替换缓存条目。与

算法 3.1　使用 InFi 过滤器的推理

输入: 输入源 *src*, 冗余阈值 T, 缓存大小 s, KNN 参数 K, 同质性阈值 θ_T

```
def InFiSkip(src):
    while x ← read(src) do
        if g(x) > T then
            y ← inference(x);
        else
            y ← None;
        end
    end
def InFiReuse(src):
    初始化空的 cache;
    while x ← read(src) do
        if Len(cache) < s then
            y ← inference(x);
            cache[g_modality(x)] ← y;
        else
            y, θ ← HKNN(cache, g_modality(x), g_cls, K);
            if θ < θ_T then
                y ← inference(x);
                replace (cache, {g_modality(x) : y});
            end
        end
    end
```

通常使用欧几里得距离的原始 K 近邻不同，欧氏距离是非参数的，本文将距离测量设置为训练好的 g_{cls}。本文用 HKNN$(cache, emb, g_{cls}, K)$ 表示 H-KNN 函数，该函数使用 g_{cls} 计算嵌入之间的距离，返回 $cache.keys$ 中 emb 的 K 个最近邻的大多数推理结果，并计算 θ。本文专注于利用端到端的可学习性，其他优化机会，如缓存预热等，不在本研究的范围内。

与现有为特定部署定制的过滤工作（例如推理卸载[57-59]）不同，InFi 支持多样的基于移动设备的部署方式：(1) 端上推理：推理模型和输入过滤器都部署在同一设备上；(2) 卸载推理：输入过滤器部署在一个设备上，推理模型部署在另一设备上。(3) 模型切分 (Model Partition)[127]：推理模型被切分到两个设备上，输入过滤器与第一部分一起部署在端侧。模型切分是一种有前途的方法，可以共同利用移动和边缘设备的计算资源[149-150]，并更好地保护移动数据的隐私[151]。对于模型切分部署，过滤器的输入是特征图，因此现有的过滤方法[57-59]都不适用。由于支持特征图模态, InFi 是第一个可以应用于模型切分任务的输入过滤器。在模型切分场景下，InFi 对端侧模型输出的中间特征图进行冗余过滤，只将预测为必要的特征图传给后续的边侧服务器完成推理计算，能够同时节省传输和运算效率。请注意，InFi 不仅限于单个移动和边缘节点的系统。例如，对每个服务器训练一个过滤器，或将一个过滤器的二元分类器更改为多类别分类器（每个服务器一个类），InFi-Skip 可以灵活地支持多租户环境中的应用[57]。

表 3.1　数据集和推理任务总结

数据集	模态	推理任务
Hollywood2	视频片段	动作分类 (AC)
	图像	人脸检测 (FD) 姿态估计 (PE) 性别分类 (GC)
	音频	语音识别 (SR)
	文本	命名实体识别 (NER) 情感分类 (SC)
ESC-10	音频	异常检测 (AD)
UCI HAR	运动信号	活动识别 (HAR)
MoCap	运动信号	用户识别 (UI)
DeepSig	无线电信号	调制识别 (MR)
WiFiHAR	WiFi CSI	活动识别 (WAR)
City Traffic	视频流	车辆计数 (VC)
	特征图	车辆计数 (VC-MP)

3.5　验证实验

3.5.1　实验设置

实现。本工作使用 Python 实现了 InFi。实验使用 TensorFlow 2.4 构建所有特征网络和分类器。学习率设置为 0.001，批大小为 32，训练周期数为 20。在文本特征网络中，嵌入层的输出维度为 32。对于图像、视频和音频特征网络，实验在两个残差块中使用 32 和 64 个卷积核。在向量特征网络的第一个全连接层中使用 128 个单元。所有特征网络的最后一个全连接层具有 200 个单元和 0.5 的 Dropout 概率。

数据集和推理模型。为了验证 InFi 的广泛适用性，实验选择了 14 个推理任务，涵盖 8 种输入模态和三种部署方式（见表 3.1）。使用了七个数据集：(1) 实验重新处理了视频数据集 Hollywood2[152]，创建了四种不同的输入模态：视频片段，图像，音频和文本。在原始视频片段上部署了动作分类模型[153]。从视频片段中采样图像，并部署了人脸检测[154]，姿态估计[5]和性别分类[154]模型。从每个视频片段中提取音频，并部署了语音识别模型[96]。文本是由图像描述模型[155]在采样图像上生成的，实验部署了命名实体识别模型（spaCy[156]）和情感分类模型[157]。(2) 实验使用 ESC-10 数据集[158]进行音频异常检测，并部署了基于 Transformer 的模型[159]。(3) 实验使用 UCI HAR 数据集[143]进行基于运动信号的人体活动识别，并部署了一个基于 LSTM 的模型。(4) 实验使用 MoCap 数据集[160]训练了基于运动信号的用户识别模型（12 个用户），使用了基于 LSTM 的架构，并将其部署为推理任务。(5) 实验使用 DeepSig 数据集[161]，并部署了一个基于 ResNet

表 3.2　在 90% 推理精度下的跳过方法过滤率

方法	FD	PE	GC	AC	VC	AD	WAR
FF	0	14.5	0.0	0.0	48.0	/	/
Reducto	/	/	/	/	48.6	/	/
InFi-Skip	36.1	18.9	33.1	56.0	66.5	75.4	11.6
最优	64.8	34.4	71.8	91.2	77.7	86.8	31.1

方法	SR	NER	HAR	UI	SC	VC-MP	MR
InFi-Skip	44.1	26.8	91.2	72.4	22.5	70.7	40.9
最优	59.9	34.4	91.8	79.8	63.8	77.7	59.9

表 3.3　在 90% 推理精度下的重用方法过滤率

方法	GC	AC	HAR	SC	VC-MP	VC
FC	66.1%	13.2%	/	/	/	59.4%
InFi-Reuse	**98.8%**	**32.1%**	**98.3%**	**43.4%**	**95.0%**	**91.1%**

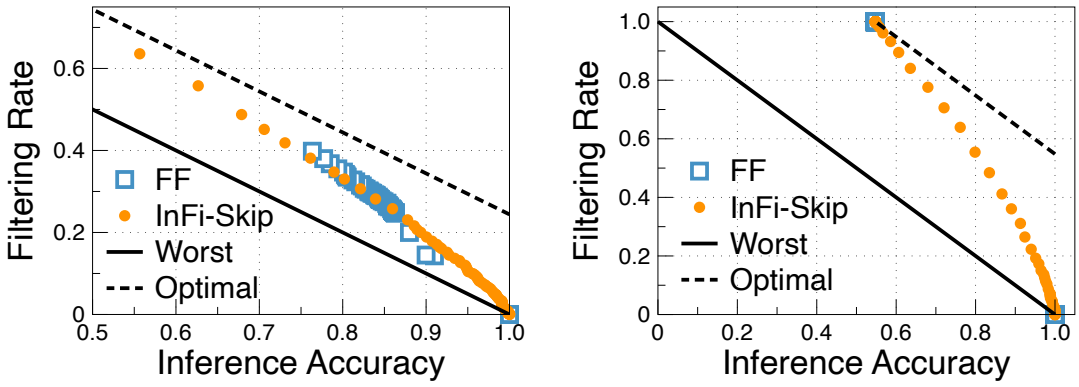
的模型，用于无线电信号的调制识别。(6) 实验使用 WiFiHAR 数据集[162]进行活动识别，并部署了一个基于 LSTM 的模型。(7) 实验从一个真实的城市规模视频分析平台中收集了一个名为 City Traffic 的视频数据集。实验从 10 个交叉口的 10 个摄像头中收集了 48 小时的视频（1 FPS），并使用在 TensorFlow 2.0 中重新实现的 YOLOv3 来统计视频帧中的车辆数量。所有部署的推理模型都加载了公开发布的预训练权重。实验按 1:1 拆分每个数据集用于训练和测试（Hollywood2 和 UCI HAR 是随机拆分的，而 City Traffic 是按每个摄像头的时间拆分的）。

设备和部署。实验使用一台搭载一颗 NVIDIA 2080Ti GPU 的边缘服务器和三个移动平台：(1) NVIDIA JETSON TX2，(2) 小米 Mi 5，和 (3) 华为 WATCH。所有与设备无关的度量标准都在边缘进行测试。对于车辆计数，实验测试了三种部署方式：在端上，卸载，以及模型切分。

基线。实验采用了三个基线方法：FilterForward (FF)[57]，Reducto[58]，和 FoggyCache (FC)[59]。对于没有现有方法的任务，实验测试了一种名为低级特征（Low-level）的方法，该方法首先为输入计算低级特征嵌入（对于音频是 MFCC，对于文本是词袋（Bag-of-Words）模型，对于动作信号和特征图是原始数据）。然后，低级特征方法对于跳过和重用两种情况都使用 K 最近邻投票（K=10）。实验还部署了 YOLOv3-tiny[120] 模型进行车辆计数，以及一个轻量级的姿态估计模型[121]以比较输入过滤和模型压缩技术。

3.5.2　推理精度和过滤率

首先，实验在十个推理任务上测试两个设备独立的度量标准（推理精度和过滤率）。实验在 FF、Reducto 和 InFi-Skip 中调整置信阈值，以及在 FC 和 InFi-Reuse 中调整缓存输入的比率，从 0 到 1，间隔为 0.01。



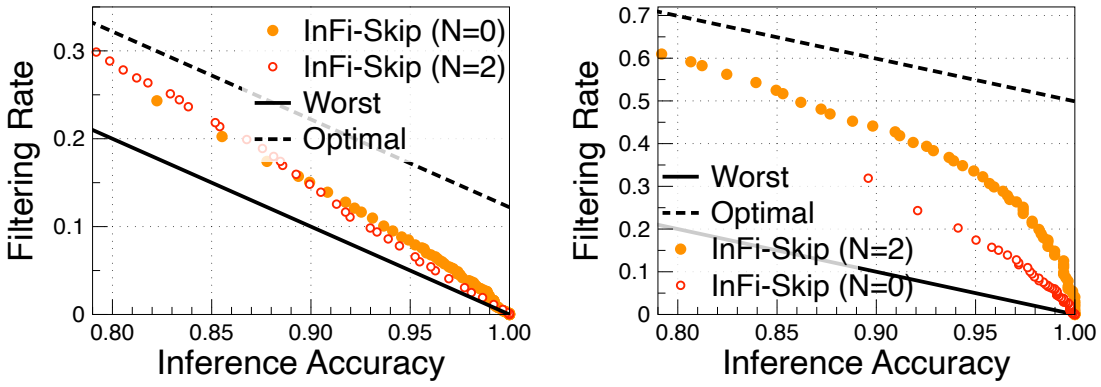
(a) 姿态估计　　**(b) 人脸检测**

图 3.6　InFi-Skip 在视觉检测任务上的过滤效果

冗余度测量。(1) 跳过：对于 FD（PE），没有检测到面部（人物关键点）的输出是冗余的。对于 GC（SC），分类置信度低于阈值 CONF（0.9）的输出是冗余的。对于 AC，不在子类 Sub 中的输出是冗余的。对于 SR，识别单词数低于阈值 N 的输出是冗余的。对于 NER，没有实体标签“PERSON”的输出是冗余的。对于 HAR，不是“LAYING”的输出是冗余的。对于 UI，不属于前 6 个用户的输出是冗余的。对于 AD，不在 {“Cry, Sneeze, Firing”}（异常事件）中的输出是冗余的。对于 MR，实验随机选择一半的无线调制类型作为冗余的。对于 WAR，带有“NO PERSON”的输出是冗余的。对于 VC 和 VC-MP，计数为零的输出是冗余的。(2) 重用：实验结果表明，缓存未命中很少发生，因此同质性阈值设置为 0.5。实验将命中缓存的输入视为冗余的。对于 VC（-MP），由于每个摄像头有 86K 张图像，固定的缓存比率可能导致 K 近邻算法效率低下。实验将缓存大小固定为 1000，并在每 5000 帧重新初始化缓存。对于其他推理任务，实验根据缓存比率设置固定的缓存大小。

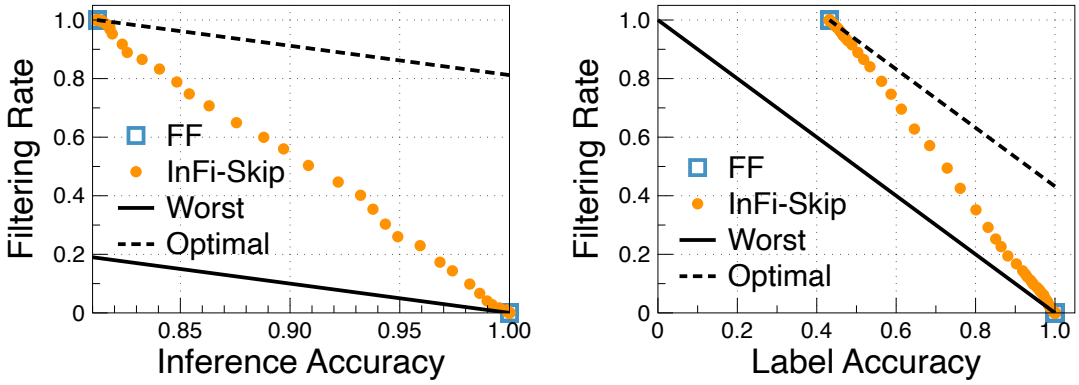
结果概览。表 3.2 和表 3.3 总结了跳过和重用方法的结果。按照相关工作[58]，实验报告在 90% 推理精度下的过滤率。最优结果通过计算 (1-0.9)+r_N 得到，其中 r_N 表示测试数据集中冗余输入的比例。结果显示，InFi-Skip 在所有 10 个任务上的性能均优于 FF 和 Reducto，具有更高的过滤率和更广泛的适用性。类似地，InFi-Reuse 在所有 6 个适用的任务上明显优于 FC。InFi-Skip 可以过滤 18.9%-91.2% 的输入，而 InFi-Reuse 可以过滤 32.1%-98.8% 的输入，同时保持 90% 以上的推理精度。对于所有任务，低级特征方法无法在未过滤任何输入（即 0.0% 的过滤率）的情况下达到 90% 的推理精度，本文在表格中省略了这些结果。

特征可辨识性。通过在 FD、PE、GC 和 AC 任务上比较 FF 和 InFi，实验验证了 InFi 端到端学习特征的可辨识性。如图 3.6 所示，FF 在姿态估计任务上有效，但在人脸检测任务上无效。最差情况（Worst）通过 $r = 1 - Acc$ 计算。原因可能



(a) SR (N=0)　　**(b) SR (N=2)**

图 3.7　InFi-Skip 在语音识别任务上的过滤效果

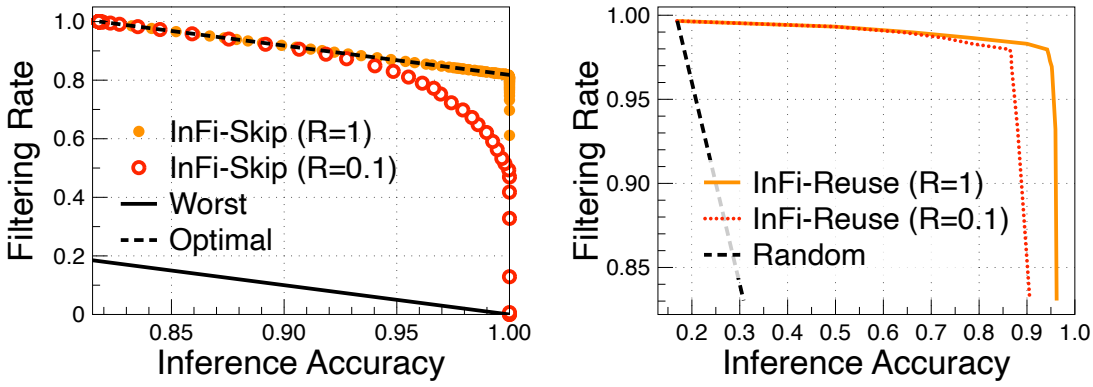


(a) AC (2 个类别)　　**(b) AC (8 个类别)**

图 3.8　InFi-Skip 在动作分类任务上的过滤效果

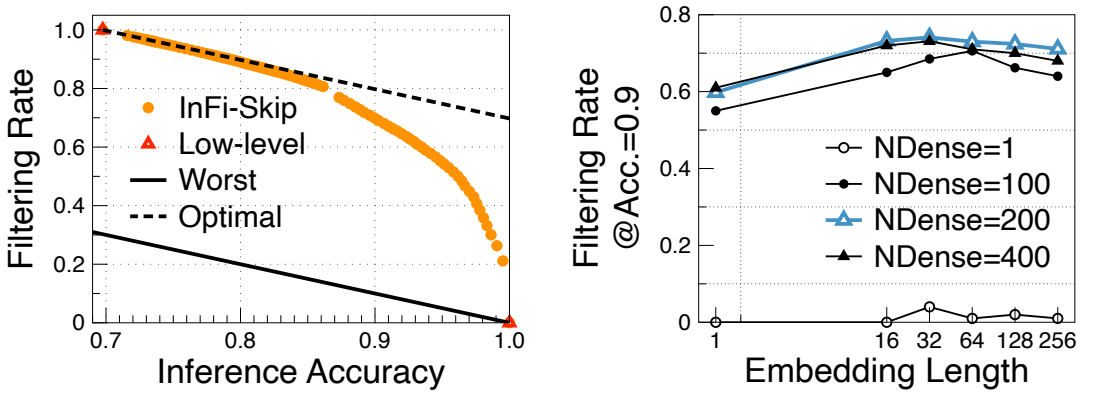
是 ImageNet 数据集中存在一个 “person” 标签，因此 FF 中的预训练特征嵌入对于判断是否存在人体姿态是具有可辨识性的。然而，在其他任务上（如 FD、GC 和 AC），预训练特征不具有可辨识性，FF 只能提供两种极端的过滤策略：要么过滤所有输入，要么不过滤任何输入，这在实践中是无用的。相反，InFi-Skip 学习了具有鲁棒可辨识性的特征嵌入，并在所有四个任务上表现良好。在 90% 以上的推理精度下，InFi-Skip 可以分别过滤 18.9% 和 36.1% 的 PE 和 FD 任务的输入。

可迁移性。一个有趣的问题是，训练得到的过滤器在冗余度测量更松或更紧的任务上是否可迁移？实验将识别单词的最小数量 N 设置为 0 和 2，并训练两个 InFi-Skip 过滤器。然后实验在两个具有不同 N 值的测试集上测试这两个过滤器。如图 3.7 所示，InFi-Skip（N=2）在 N=0 的情况下的性能接近 InFi-Skip（N=0），然而，在 N=2 的情况下，InFi-Skip（N=0）的性能明显较差。一个直观的解释是，具有更松冗余度测量的学得特征覆盖了具有更紧冗余度测量的特征，反之则不成立。



(a) HAR（L=LAYING）　　(b) HAR 重用

图 3.9　InFi 在人物动作识别推理任务上的过滤效果



(a) NDense=200，EmbLen=128　　(b) 模型复杂度

图 3.10　InFi-Skip 在用户验证任务上的过滤效果

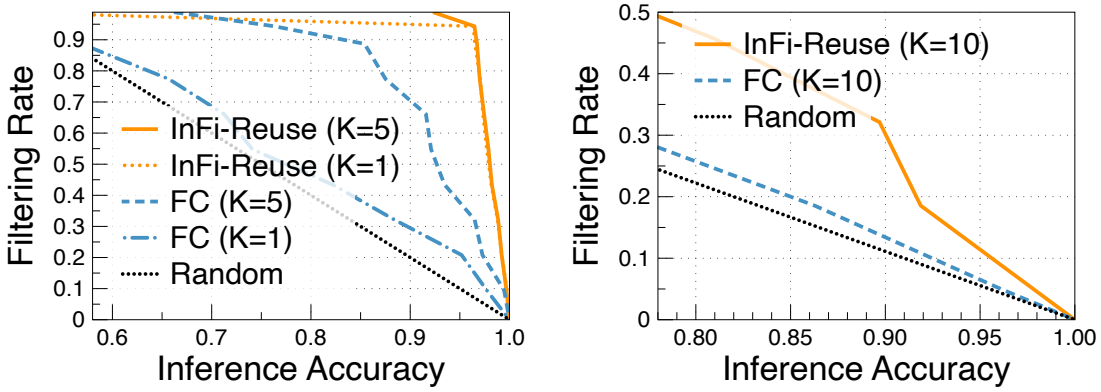
对类别子集大小的敏感性。对于类别子集的冗余度测量，实验设置不同的子集大小以测试其敏感性。如图 3.8 所示，对于动作分类任务，设置较小的类别子集会带来更多的冗余样本。而 InFi-Skip 可以鲁棒地在两种情况下提供平滑的精度-效率权衡曲线，明显优于 FF（只提供两个极端点）。

对训练规模的敏感性。实验进一步将训练集划分为不同大小的集合。如图3.9所示，仅使用训练集中的 10% 样本，InFi 仍然可以在 HAR 任务上实现接近最优的性能。记 R 为用于训练的训练样本比例。当达到 95% 以上的推理准确率时，InFi-Skip（R=1）过滤了 86.4% 的输入，而 InFi-Skip（R=0.1）仍然过滤了 81.1%。对于高精度的重用，训练规模的影响相对较大。当过滤 90% 的输入时，InFi-Reuse（R=1）可以达到 95.9% 的推理准确率，而 InFi-Reuse（R=0.1）的精度降至 88.1%。

对模型复杂度的敏感性。为了探索输入过滤器的复杂性与性能之间的关系，实验使用不同长度的嵌入（1、16、32、64、128、256）和分类器中的全连接单元数量（1、100、200、400）为 UI 任务训练 InFi-Skip 过滤器。实验通过在达到 90%

推理准确率时的最大过滤率来衡量性能。如图3.10b所示，除了极端情况（例如，仅一个全连接或嵌入单元），过滤性能相对鲁棒。

对 K 近邻中 K 参数的敏感性。 K 近邻中的参数 K 影响分类精度。实验将 K 从 1 变化到 20，测试重用过滤器的性能。如图3.11所示，在 GC 任务上，InFi-Reuse 对不同的 K 参数表现鲁棒，而 FC 遭受严重的性能下降。例如，以 90% 推理准确率为例，FC（K=5）可以过滤 68.4% 的输入，而 FC（K=1）只能过滤 27.3%，略高于随机猜测（20%）。相反，InFi-Reuse（K=1,5）都可以在 95% 以上的推理准确率下实现 94.3% 的过滤率。对于 AC 任务，结果表明手工制作的特征 SIFT 不具有可辨识性，所有测试的 K 参数导致与随机标签相似的性能。InFi-Reuse 可以学习到与动作相关的有区分性的特征，它可以过滤 18.6% 的输入并保持 90% 以上的推理准确率（K=10）。



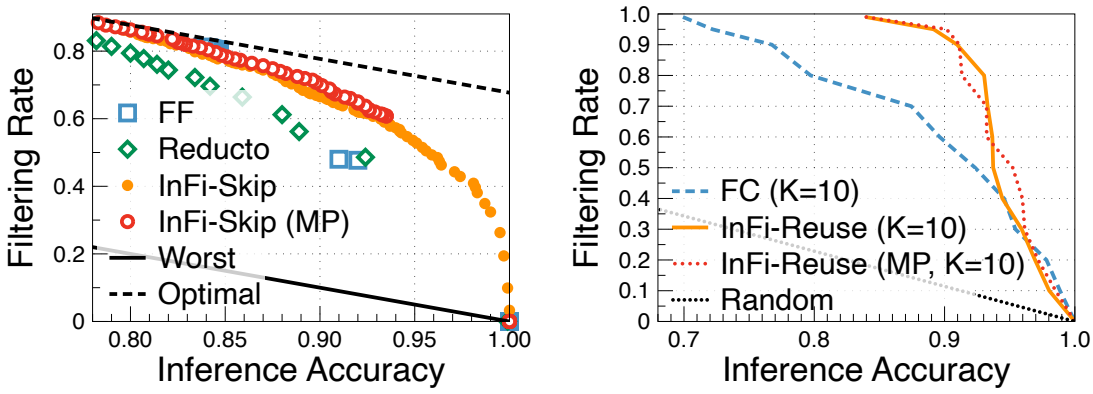
(a) 性别分类　　**(b) 动作分类**

图 3.11　InFi-Reuse 和基线方法在视觉分类任务上的比较

VC(-MP) 任务的比较。与其他数据集不同，视频帧是按时间顺序而不是随机到达的。对于 VC-MP，实验将 YOLOv3 模型分为移动端（前 39 层）和边缘端（其余层）。如图 3.12所示，InFi 优于 FF、Reducto 和 FC，同时也是唯一适用于 VC-MP 任务的方法。InFi-Skip 在超过 90% 的推理精度下实现了 66.5% 的过滤率，而 FF 和 Reducto 分别实现了 48.0% 和 48.6%；当 K=10 时，InFi-Reuse 过滤的输入比 FC 多 31.7%。结果表明端到端学习的特征优于手工设计和预训练的特征。

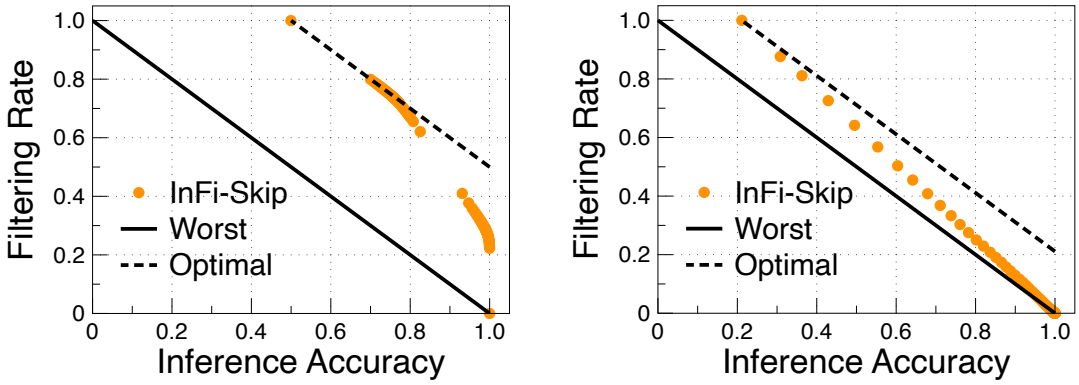
数据模态。对于移动应用程序，无线电信号和 WiFi CSI（信道侧信息）是常见模态。图 3.13显示了在两个移动特色任务上应用 InFi 的实验结果：无线调制识别和基于 WiFi CSI 的动作识别。请注意，现有方法都不能应用于这两个任务。在 90% 的目标精度下，InFi-Skip 为 MR 和 WAR 任务分别节省了 40.9% 和 11.6% 的计算。

多任务扩展。第 3.4.2节介绍了如何将 InFi 扩展到多任务任务。实验使用 Hollywood2 数据集和相应的推理任务来验证 InFi-Skip 的多任务扩展。首先，实



(a) 跳过方法　　(b) 重用方法

图 3.12　InFi 和基线方法在车辆计数任务上的比较。

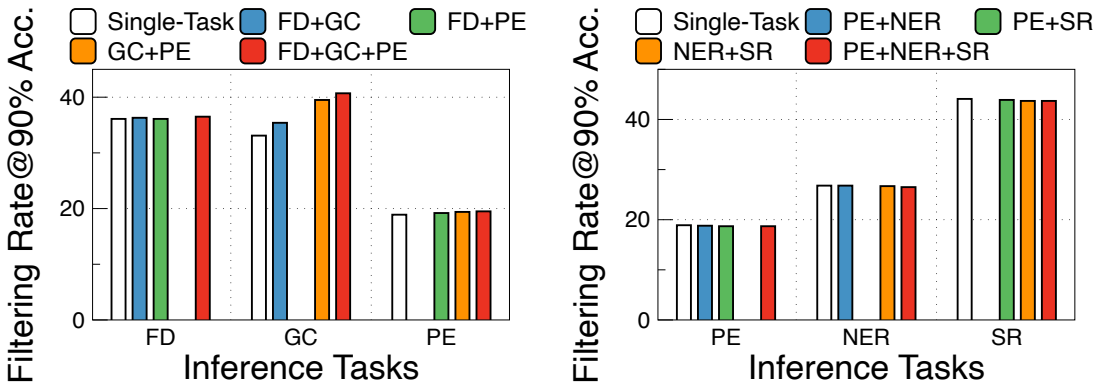


(a) 调制识别　　(b) WiFi 动作识别

图 3.13　InFi-Skip 在调制识别和 WiFi 动作识别任务上的过滤效果

验选择图像模态上的三个推理任务: FD、GC 和 PE。实验使用单一任务、两个任务和三个任务构建 InFi-Skip 过滤器, 并在每个任务上验证它们的性能。如图 3.14a所示，多任务过滤器在所有三个任务上优于单一任务过滤器，在达到 90% 推理精度时，过滤率提高了最多 7.6%（对于 GC 任务）。接下来，实验选择不同模态上的三个推理任务：图像上的 PE、文本上的 NER 和音频上的 SR。实验使用单一模态、两个模态和三个模态构建 InFi-Skip 过滤器并对其进行验证。如图 3.14b所示，将这些多模态任务融合成一个过滤器会导致过滤率略微降低。但请注意，由于在不同任务之间共享参数，整体效率得到了提高。

在线主动更新。为了验证 InFi 在线适应的主动策略，实验选择 VC 任务并比较三种训练方法：(1) 离线训练（Offline）：选择一天的前 10% 帧进行训练；(2) 周期更新（Periodic）：选择每小时的前 10% 帧进行训练和更新；(3) 主动更新（Active）：详见第 3.4.3节。为了公平比较，实验对三种方法设置相同的阈值 (0.5)。如图 3.15所示，本文提出的主动策略显著提高了 InFi-Skip 的在线适应性。离线策略在输入分布发生变化时性能严重下降，主要是因为帧从红外到 RGB 图像的



(a) 单模态多任务　　**(b) 多模态多任务**

图 3.14　单任务和多任务 InFi-Skip 的比较

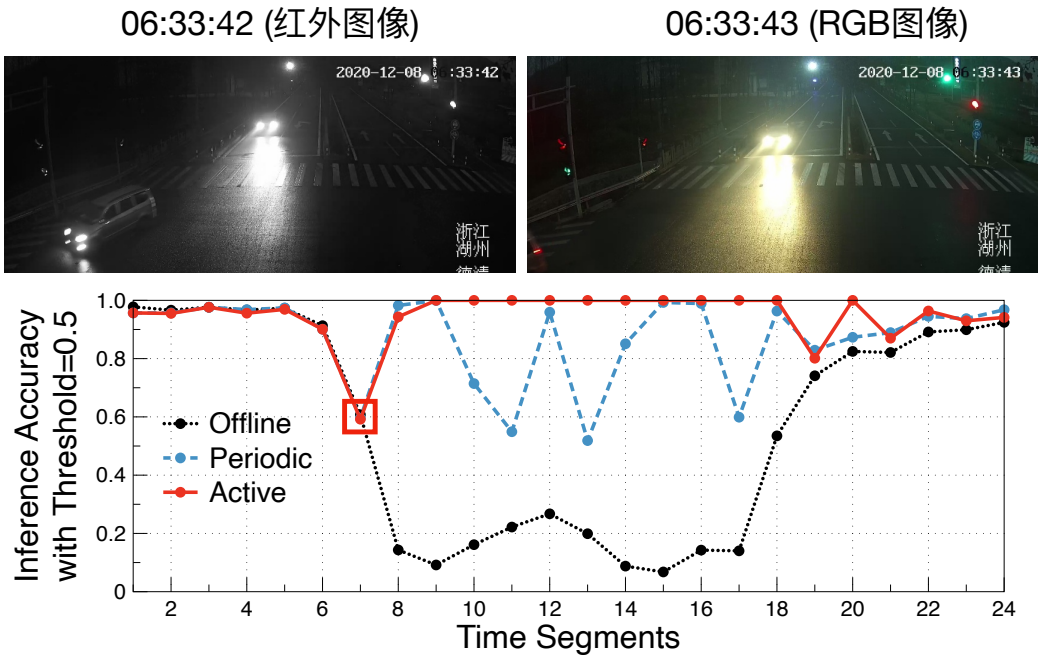


图 3.15　InFi-Skip 在车辆计数任务上的主动更新效果

变化。平均而言，离线策略仅实现了 56.4% 的推理精度。周期性更新策略在一定程度上缓解了这个问题，将平均精度提高到了 87.0%，但仍然受到性能波动的影响。主动策略的性能仅在第 7 个时间段下降，因为在那之前它没有看到任何 RGB 图像。主动策略有效地选择信息丰富的样本以适应新的分布，并鲁棒地执行准确的过滤。平均而言，本文提出的主动策略实现了 94.8% 的推理精度，比离线策略高 38.4%。

3.5.3　可过滤性

第 3.2节比较了推理模型和过滤模型的假设复杂性。令 “Conf.>T” 表示低置信度分类情况（见 3.2.2节），“Class Subset” 表示冗余类别子集情况（见 3.2.3节），

以及“Reg.>T”表示有界回归情况（见 3.2.4节）。GC 和 SC 属于“Conf.>T”情况，其中 T 为 0.9。AC，NER 和 HAR 属于“Class Subset”情况，其中 AC 选择 2 个动作标签，NER 选择“PERSON”标签，HAR 选择“LAYING”标签。FD，PE 和 VC(MP) 属于“Reg.>T”情况，其中 T 为 0。SR 是一个序列到序列模型，不能完全适用于这三种情况。实验计算了过滤率相对于 90% 推理准确率的最佳过滤率的比率，以比较不同情况的过滤性能。从实际的角度来看，实验验证了带有和不带有 InFi-Skip 过滤器的总体吞吐量。如图 3.16所示，在“Conf.>T”情况下（前文证明了过滤器的复杂性不低于推理模型），实现的过滤率明显较低（中值为 0.41），而在其他情况下（前文证明了过滤器倾向于更简单），实现的比率明显较高（中值为 0.71/0.78）。另一方面，在可过滤的情况下，InFi-Skip 过滤器对总体吞吐量的改善要比不可过滤的情况更显著。在不可过滤的情况下，GC 和 SC，InFi 实现了大约 1.3 倍的吞吐量提升，而在可过滤的情况下，它可以将吞吐量提高至 5.92 倍，并分别达到回归和子集类情况的中值为 1.8 和 2.25。这些结果显示了本文证明的可过滤性在实际应用中的指导意义。

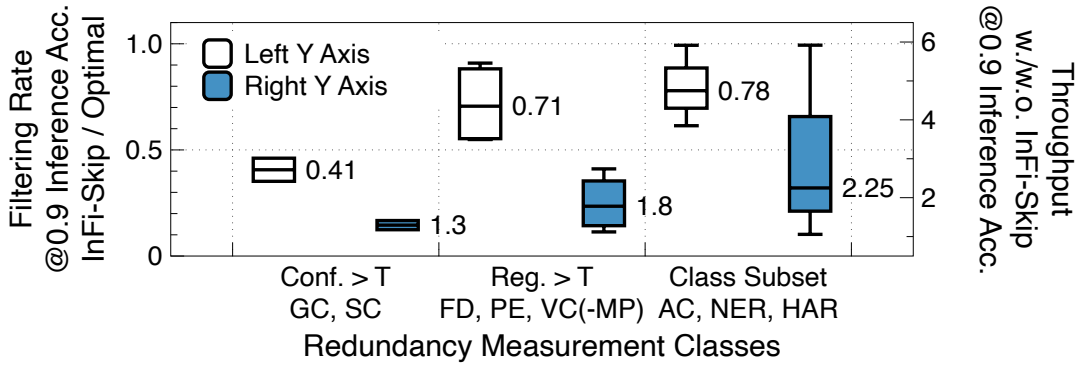


图 3.16　可过滤和不可过滤情况的比较

计算与资源效率。正如前文中讨论的那样，一个“有效”的过滤器应该既准确又轻量。上述结果显示，InFi 可以在保持准确推理的同时过滤大量输入。在训练阶段，InFi（图像模态）每批次（批大小为 32）需要大约 710 毫秒，并且需要 5337 兆字节的 GPU 内存，这是大多数商用 GPU 都可以满足的。InFi 对其他输入模态需要的资源要少得多，例如，InFi（向量）每批次只需要 3 毫秒，并且只需要 435 兆字节的内存。实验在移动平台上测试推理阶段的延迟和能耗。为了公平比较，实验选择了 TFLite 优化的 MobileNetV1，这是移动设备上最高效的卷积神经网络之一。如图 3.17所示，在三个移动平台上，InFi 与图像特征网络的成本仅为 MobileNetV1 运行时间的 12-25%。InFi 的平均能耗为每帧 14.4/79.7 毫焦，远低于手机/智能手表上 MobileNetV1（每帧 410.4/803.8 毫焦）。实验使用 MindSpore 实现 InFi，结果表明 InFi 的低能耗和低延迟执行不依赖于实现框架。

设备上的在线更新。基于 Chaquopy 库，实验在手机（小米 Mi 5）和智能手

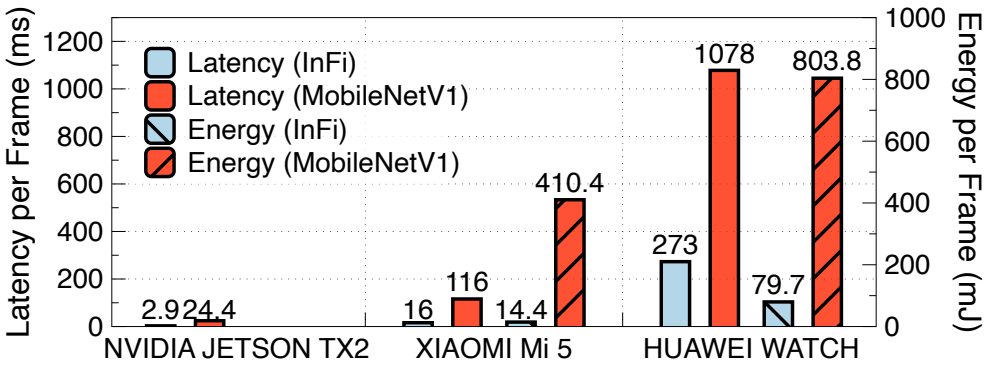


图 3.17　InFi 在移动平台上的延迟和能耗开销

表（华为 WATCH）上测试了在设备上训练的开销。实验随机生成了形状为（224，224，3）的图像，并将批大小设置为 16。对于 InFi（图像模态），实验表明在手机和手表上在线更新权重每批次分别需要大约 20 秒和 50 秒。而对于 NVIDIA JETSON TX2，以相同配置训练输入过滤器每批次大约需要 1 秒。

3.5.4　不同移动部署方式

现在，实验通过三种部署方式验证在实际系统中推理任务的整体性能。

车辆计数。首先，考虑车辆计数任务：(1) 端上：InFi（图像）和 YOLOv3 模型在 TX2 上；(2) 卸载：InFi（图像）在 TX2 上，YOLOv3 模型在边缘设备上；(3) 模型切分：YOLOv3 的前 39 层（10 个卷积块）和 InFi（特征图）在 TX2 上，YOLOv3 的其余部分在边缘服务器上。YOLOv3 模型在 TX2 和边缘上的平均吞吐量分别为 3.2 FPS 和 22.0 FPS。对于模型切分部署，边缘侧模型提供 24.5 FPS。实验报告了使用 InFi-Skip 和 InFi-Reuse 的平均吞吐量和带宽节省，在推理准确率超过 90% 的情况下，见表 3.4。为了公平比较，实验测试了 YOLOv3-tiny[120] 模型的吞吐量，这是 YOLOv3 的一个压缩版本。YOLOv3-tiny 的推理准确率仅为 67.9%，不符合 90% 的目标。分解开销，InFi 的推理每帧大约耗时 3 毫秒，K 近邻的平均延迟每帧为 6 毫秒，其中 K=10，缓存大小 =1000。在实现超过 90% 的推理准确率的情况下，InFi-Skip 将端上/卸载/模型切分部署吞吐量提高至分别为 9.3/55.2/39.0 FPS。显然，在车辆计数任务中，InFi-Reuse 有更多的过滤机会。InFi-Reuse 将这三个部署的吞吐量提高至 27.2/77.2/46.0 FPS。除了不涉及跨设备数据传输的设备上部署外，InFi-Skip / InFi-Reuse 还为卸载和模型切分任务节省了 66.5% / 91.1% 和 70.7% / 95.0% 的带宽。与 YOLOv3-tiny 交换了显著且固定的精度损失不同，InFi 在推理精度和开销之间提供了灵活的权衡。

姿势估计。其次，验证姿势估计任务：1）端上：InFi（图像）和 TX2 上的 OpenPose 模型；2）卸载：InFi（图像）在 TX2 上，OpenPose 模型在边缘设备上；3）模型切分：OpenPose 的前 39 层（10 个卷积块）和 InFi（特征图）在 TX2 上，

表 3.4　车辆计数任务的吞吐量（FPS）/ 带宽节省（%）

任务	YOLOv3	InFi-Skip	InFi-Reuse	YOLOv3-tiny
推理准确率 (%)	100	90.3	90.5	67.9
端上	3.2/-	9.3/-	27.2/-	20.4/-
卸载	22.0/-	55.2/66.5	77.2/91.1	225.3/-
模型切分	24.5/-	39.0/70.7	46.0/95.0	230.4/-

表 3.5　姿势估计任务的吞吐量（FPS）/ 带宽节省（%）

任务	OpenPose	InFi-Skip	OpenPose-light
推理准确率 (%)	100	90.1	76.5
端上	15.4/-	18.0/-	28.1/-
卸载	27.7/-	31.5/18.9	98.5/-
模型切分	29.2/-	33.1/20.2	102.4/-

表 3.6　两个自然语言处理任务的吞吐量（QPS）/ 带宽节省（%）

任务	NER	NER+InFi-Skip	SC	SC+InFi-Skip
推理准确率 (%)	100	90.2	100	90.0
端上	24.3/-	33.2/-	27.9/-	36.0/-
卸载	133.2/-	181.9/26.8	60.2/-	77.7/22.5
模型切分	N/A	N/A	62.5/-	82.0/24.1

OpenPose 的其余部分在边缘服务器上。此外, 实验还测试了 OpenPose-light[121] 模型的吞吐量，这是 OpenPose 的轻量级版本。实验结果如表 3.5所示。与车辆计数任务类似，尽管轻量级模型的吞吐量显著提高，但其不能达到目标的 90% 推理准确率。InFi-Skip 可以灵活平衡推理精度和吞吐量。例如，对于设备上的部署，使用 InFi-Skip 后，吞吐量提高至 1.17 倍，推理精度保持在 90% 以上。

自然语言处理任务。第三，实验测试了两个不同部署方式的自然语言处理任务, NER 和 SC。请注意, 由于黑盒 API, NER 任务不适用于模型切分。如表 3.6所示, InFi-Skip 有效地提高了吞吐量, 并分别为两个任务节省了 22.5% 和 26.8% 的卸载通信。

3.6　小结

本章首先对端边协同模型推理中涉及的输入过滤问题进行了形式化，并给出了过滤器的有效性条件。基于推理任务和输入过滤器的假设族之间的复杂性比较，本章对推理任务的“可过滤性”进行了定义和分析，旨在指导和解释输入过滤技术的应用。本章提出了名为 InFi 的端到端可学习的输入过滤框架, 统一了跳过和重用方法。由于端到端可学习性，本章提出的框架具有鲁棒辨别力的特征嵌入，支持各种输入模态和推理部署方式。全面的验证实验证实了的理论结果, 并显示 InFi 具有更广泛的适用性，在精度和效率方面优于基线方法。

第 4 章　端云协同的安全推理协议

基于自注意机制（Self-Attention）的 Transformer 模型的推理服务是最前沿的人工智能应用[12,163-164]。云计算能力，如自动扩展[9]，满足了为 Transformer 提供服务的需求，特别是具有数十亿参数的大型模型。因此，像 OpenAI 这样的头部组织选择将其基于 Transformer 的服务进行全云部署[7]。然而，在注重隐私的领域，将原始数据发送到云端是不可行的，例如三星在敏感代码泄漏后全面禁止公司内部使用 ChatGPT[15]。

模型切分不够安全。模型切分推理[165-167]在设备和云端之间有策略地分布神经网络层。设备将中间激活发送到云端以继续推理。模型切分推理避免了原始数据传输，同时保持了效率[168-169]。然而，随着研究揭示从中间激活中反向工程出敏感信息的潜力，基于模型切分的安全性仍然堪忧[170-171]。

安全两方协议产生了巨大的开销。最近的研究通过同态加密（Homomorphic Encryption）和安全两方计算（Secure Two-Party Computation）实现了安全的 Transformer 推理[40-42]。然而，这些协议产生了巨大的计算和设备与云端通信开销，特别是在具有非线性复杂层（如 LayerNorm 和 ReLU 激活层）的情况下。例如，CipherGPT 协议下使用 GPT2 模型生成单个词项花费了 25 分钟的处理时间和超过 90 GiB 的流量[42]。

第一性原理思维：三方威胁模型。为了克服效率障碍，本文使用第一性原理思维重新思考基本的两方假设：模型所有者（Model Owner）和数据所有者（Data Owner）。基于本工作开发两个实际的基于 Transformer 的服务，发现了一个共同的经验：**模型开发者 ≠ 模型服务器**。对于这两个服务，本工作使用收集的数据微调[66]开源参数[67-68]来开发 Transformer 模型。本工作具有足够的计算能力进行离线的模型开发，但缺乏为大规模用户提供长期服务的能力，即模型开发者需要依赖第三方云平台为开发的的模型提供服务。与在现实世界中的开发经验一致，本文提出了一个新的三方威胁模型。在这个模型中，将模型所有者分解为两个实体：模型开发者（Model Developer）和模型服务器（Model Server）。由于开发的模型是专有的，模型开发者必须保护他们的模型参数免受来自模型服务器的潜在攻击，因此本文假设它们不会共谋。

STIP 的设计思想。基于本文引入的三方威胁模型，开发了 STIP，即 Secure Transformer Inference Protocol 的缩写，具有两个主要的设计思想。首先，本文采用高效的特征空间置换（Feature-Space Permutation）进行安全且等价的 Transformer 推理。由于推理是在不受信任的服务器上执行的，模型参数和设备上的数据必须在上传到云端之前进行转换。基于特征空间的高效置换，本文设计了一种 Trans-

former 层的数据和参数转换方法。本文证明了使用提出的转换进行计算的数学等价性，从而确保没有精度损失。其次，本文设计了一种模型开发者和数据所有者之间的半对称（Semi-Symmetric）保护方案。这个设计思想源于神经网络的顺序结构。本文揭示了模型开发者只需要与数据所有者共享第一层和最后一层的相同置换，而中间层转换的信息则可以由模型开发者独自保留。类似的半对称保护方案在不同领域中得到了应用，如图像加密[172]和在线购物[173]。本文通过向量距离相关性（Vector Distance Correlation）[69]展示了 STIP 的隐私保护能力，并证明其对暴力和已知明文攻击（Known-Plaintext Attack）的抵抗性。本文实现了 STIP 并在实际系统上对具有多达 700 亿参数的各种 Transformer 模型进行了验证。实验结果展示了 STIP 的效率能够与未保护的全云推理相媲美，超过了最先进的安全两方协议[40-42]数百万倍。

4.1　背景

本节首先介绍端云协同和安全 Transformer 推理的相关工作，之后提出了一个三方威胁模型，并阐述了安全协议的范围、设计目标和独特性。

全云推理：高效但不安全。基于 Transformer 的推理服务已经成为最引人注目的人工智能应用，例如，ChatGPT 创下了用户增长最快的记录[12]。为了支撑 Transformer 推理服务，特别是具有数十亿参数的大型模型，云计算能力（如自动扩展[9]）是一种理想选择。工业界已经发布了几个云原生的用于 Transformer 推理的框架，例如 NVIDIA NeMo[13]和 Microsoft DeepSpeed[14]。因此，包括 OpenAI 在内的大多数组织选择为其基于 Transformer 的服务进行全云部署[7]。然而，将原始数据发送到云服务器在各种注重隐私的领域是不可行的，例如，三星在敏感代码泄露后正式禁止员工使用 ChatGPT[15]。

全端推理：安全但不可扩展。为了解决数据隐私问题，另一种选择是完全在端设备上部署 Transformer 模型。通过诸如权重量化[16]之类的模型压缩技术，具有数十亿参数的 Transformer 模型可以在端设备上运行推理[17]。然而，全端部署的可扩展性有限。首先，浮点操作（FLOPs）和内存占用随参数数量线性增长，而端设备上的计算资源增长远远慢于 Transformer 模型大小的增速[4,18]。其次，模型压缩不可避免地导致精度损失[19]，但在竞争激烈的大型 Transformer 市场中，即使轻微的精度降级也可能导致产品落后竞争对手[174]。

端云协同。由于云服务和端设备都无法独立满足 Transformer 模型的要求，因此端云协同思路自然而然地出现在最近的研究工作中[165-167]。通过在端设备和云服务器之间合理分配推理，协同推理避免了传输原始数据的同时保持了效率[168-169]。

4.1.1　安全两方推理

不传输原始数据仅仅是安全性的底线。针对深度神经网络的攻击研究表明，中间激活（例如，文本嵌入[171]）可以被反向工程从而揭示原始数据中的敏感信息[170]。因此模型推理迫切需要更严密的安全性保障。

用于 Transformer 推理的同态加密和安全两方计算。同态加密是一种密码学技术，允许在加密数据上执行计算而无需解密，而安全多方计算允许多个参与方共同计算其输入上的函数，同时保持这些输入的私密性。在模型推理的上下文中，通常考虑两方计算，这是安全多方计算的一种特殊情况，其中模型所有者（云服务器）和数据所有者（端设备）分别代表两方。最近的研究已经证明了组合使用同态加密和安全两方计算来实现安全 Transformer 推理[40-42]。然而，这些协议在处理非线性复杂层（例如 LayerNorm 和 ReLU 激活层）时会产生显著的计算和通信开销[42]。

4.1.2　与实际应用需求对齐：三方威胁模型

两方设置的简单性，其中一方代表端设备，另一方代表云，与同态加密和安全两方计算理论完美契合。然而，效率挑战也同样根植于同态加密和安全多方计算理论固有的计算难度[175]，这使得思考如下问题变得非常必要：**两方设置是否真正符合实际应用的需求?**

令人惊讶但幸运的是，答案是否定的。这个结论来自本工作开发两个真实世界基于 Transformer 的推理服务的经验。

服务 1：校园安全聊天机器人。本工作为校园安全开发了一个基于大型语言模型的聊天机器人。聊天机器人使用监控视频分析的数据库作为信息源。用户，包括学生和校园安全人员，可以向聊天机器人提问，比如“在某个时段是否发生了任何异常行为?”，并以自然语言获得回复。

服务 2：座舱助手。本工作部署了一个多模态 Transformer 模型，以增强座舱智能助手的功能。多模态 Transformer 以车内视频帧为输入，并生成自然语言的场景描述。场景描述可以帮助车内助手变得更加用户友好，例如根据面部表情推荐音乐。

共同经验：模型开发者并非模型服务器。对于这两项服务，本工作使用收集的数据微调[66]开源参数[67,176] 来进行 Transformer 模型的开发。本工作依托的校园实验室和合作公司的计算能力足以进行离线模型开发，但无法支持大规模的长期服务。模型开发者需要依赖第三方云计算平台来为开发的 Transformer 模型提供服务。实际上，这不仅仅是本工作的经验，对于其他模型开发公司也是如此。例如 OpenAI 使用 Microsoft Azure 云平台为数亿用户提供 ChatGPT 服务[6-7]。

三方威胁模型。为了与开发实际服务的经验保持一致，本文引入了一个三方

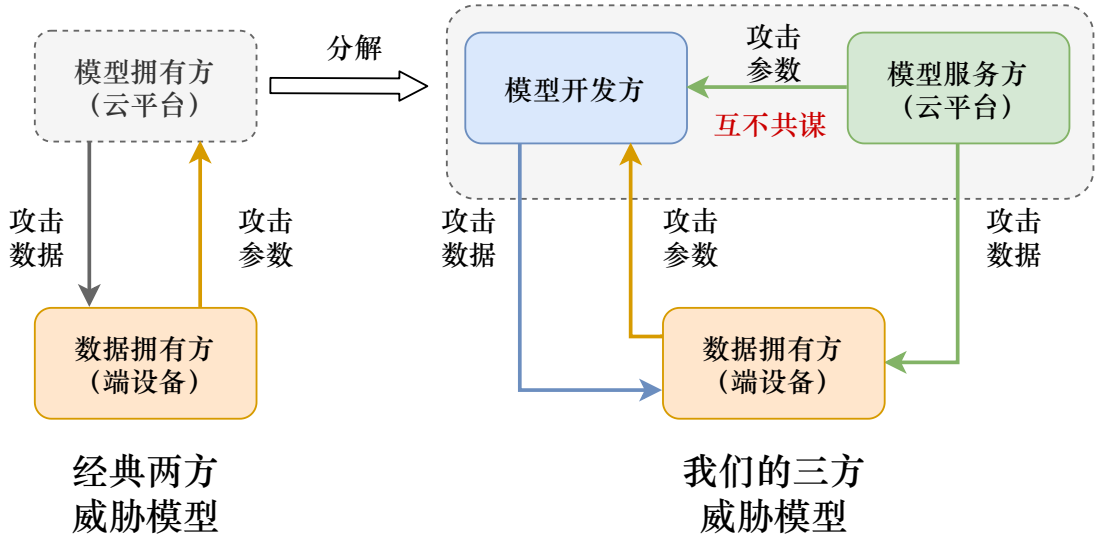


图 4.1　三方威胁模型

威胁模型，如图 4.1所示。从经典的两方设置出发，本文将模型所有者分为两个不同的实体：模型开发者和模型服务器。考虑到开发的模型是专有的，模型开发者必须保护他们的模型参数免受来自模型服务器的攻击[177]。本文假设模型开发者不会与云平台共谋。这两者之间的任何合作都会导致回归到经典的两方设置。模型所有者角色的这种分解不仅增强了实用性，而且放宽了对手假设。这种调整在克服两方协议固有的计算困难性方面被证明是关键的。

值得注意的是，本文的三方设置也有局限性。例如，它并不完全适用于那些可以开发自己模型的云提供商，比如 Google Gemini[163]。

研究范围。本工作的重点是在三方设置中实现 Transformer 模型推理的高效且安全的推理，即端到端的前向传播过程。Transformer 推理结果可以作为各种下游应用的基础，例如可以自动调用外部 API 的 AI 代理[178]。下游应用中的潜在安全风险和效率问题不在本工作的研究范围内。

设计目标。本文的协议有四个主要设计目标：

（1）数据和参数安全。首要目标是确保端设备上的数据和模型参数的安全。

（2）无精度损失。协议需要执行无精度损失的推理，即在 Transformer 模型中不应有任何计算的近似。

（3）支持生产环境。协议必须支持在生产环境中使用的推理框架，包括诸如 kv-cache 之类用于效率优化的技术[13-14]。

（4）对 Transformer 变体的灵活扩展。鉴于 Transformer 模型以各种新兴变体不断演变[67-68,179-181]，协议必须具备对 Transformer 变体的灵活扩展能力。这确保了协议在长期内能够持续可用，而无需进行逐案适应。

表 4.1 显示了本文提出的方法与现有的 Transformer 推理方法在上述四个设计目标上的比较。

表 4.1　STIP 与现有的 Transformer 推理方法的比较

方法	安全	精度无损	支持生产环境	测试的模型
全云	✗	✓	✓	所有
Iron[40]	✓	✗	✗	BERT 系列
THE-X[41]	✓	✗	✗	BERT-Tiny
CipherGPT[42]	✓	✗	✗	GPT-2
STIP	✓	✓	✓	GPT/LLaMA/ ViT/LLaVA/ BERT/Mixtral 系列

4.2　挑战

4.2.1　过高的密码学开销

数据和参数的转换是保护的关键。现有的基于同态加密和安全两方计算技术的协议在安全性方面具有极高的计算和通信开销。正如图 4.2所示，CipherGPT[42] 对于 GPT2[179] 的单次前向传播需要超过 25 分钟和 90 GiB 的流量。

Transformer 模型的成功在于其在自注意力和前馈模块中利用全局矩阵乘法，与递归架构相比，这使其在实现层面上高度可并行化[182]。Transformer 架构的这种归纳偏差不仅在实现层面上高效，还具有隐藏单元的置换对称性[183]和词项间置换不变性[184]等一些属性。鉴于这些属性，本文提出第一个设计思想：

▷ **在特征空间上进行高效的随机置换以实现等价的 *Transformer* 推理。**

由于推理是在不受信任的服务器上执行的，因此必须在上传到云之前对模型开发者的参数和端设备上的数据进行转换。本文基于特征空间中的随机置换设计了专门用于 Transformer 层的数据和参数转换。这种转换可以通过移动内存指针来高效实现，复杂度为 $O(d)$，其中 d 是特征维度。正如图 4.2所示，本文的协议，STIP，实现了比 CipherGPT 高几个数量级的效率，延迟接近全云部署。本文证明了使用提出的转换进行计算的数学等价性，从而确保不会丢失精度。

4.2.2　攻击面脆弱性

虽然基于随机置换的方案效率高且无损准确，但赋予其鲁棒的安全性面临着技术挑战。直接采用序列级置换方案[184]导致可能的置换数量为 $n!$，当输入词项数量 n 较小时，无法防范针对数据的暴力攻击（Brute-Force Attack）。选择在特征空间进行置换可以增强保护，但仍然使用单一置换矩阵 π 对已知明文攻击仍存在漏洞。原因在于，一旦云服务器获取了端设备上的已知明文数据对和转换后的数据，就可以轻松恢复置换矩阵，随后反向转换所有参数并暴露敏感信息。

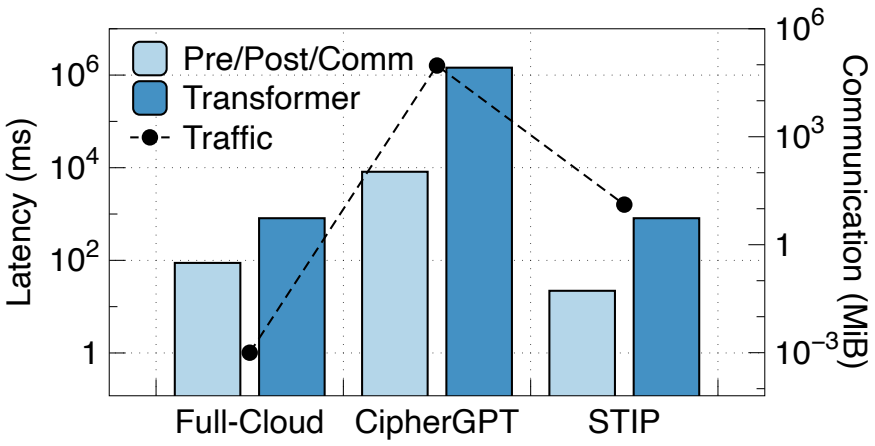


图 4.2　GPT2-124m 模型的延迟和通信开销

表 4.2　不同基于置换的方案在置换数量和对抗攻击方面的比较

保护方案	数据	参数	暴力攻击	已知明文攻击
序列维度置换	$n!$	1	✗	✗
特征维度置换（使用单个 π）	$d!$	$d!$	✓	✗
特征维度置换（使用集合 $\{\pi_1, ..., \pi_{3L}\}$）	$d!$	$(d!)^{3L}$	✓	✓

表 4.2总结了基于数据和参数的可能置换数量以及对暴力攻击和已知明文攻击的抵抗能力。

▷ 模型开发者和数据所有者之间使用半对称的置换矩阵集合。

这一设计思想源于神经网络的顺序结构。数据所有者只需要与模型开发者共享第一层和最后一层的相同置换。中间层的变换信息可以由模型开发者独占保留。因此，本文提出了一种使用矩阵集合 $\{\pi_1, ..., \pi_{3L}\}$ 的特征空间置换方案，其中 L 表示层数。类似的半对称保护方案已经在诸如图像加密[172]和在线购物[173]等领域得到探讨。本文基于距离相关性[69]分析了提出的方案的隐私保护能力，并证明了它对暴力攻击和已知明文攻击的抵抗力。

4.3　问题定义

为了形式化呈现数学等价的转换并分析理论安全性, 本节定义了 Transformer 推理和三方威胁模型。

4.3.1　Transformer 推理

首先使用原始的 Transformer 架构[176] 来介绍推理工作流程，不失一般性，高级的 Transformer 变体（GPT[185]，LLaMA[67]，ViT[176] 和 Mixtral[180]）将在第 4.5节中讨论。

端云模型切分。在 Transformer 模型中，嵌入操作是将离散输入（例如单词或图像）映射到连续向量的初始步骤[186]。为了将本文的协议与原始数据模态解耦，本文将端云协同推理的起点设置为嵌入，而不是输入。默认情况下，仅嵌入操作在设备上执行，而复杂的 Transformer 层和分类器部署在云端，如图 4.3所示。

Transformer 层前向传播。如图 4.3所示，云端 Transformer 模型由 L 个顺序 Transformer 层和一个分类器组成。用 F_θ 表示具有可训练参数 θ 的 Transformer 模型。本文将 $\{f_i : \mathbb{R}^{n\times d} \mapsto \mathbb{R}^{n\times d} | i \in [L]\}$ 定义为 Transformer 层，$f_c : \mathbb{R}^{n\times d} \mapsto \mathbb{R}^{n\times s}$ 为分类器，其中 n 是序列长度（例如标记数），d 是模型特征维度，s 是输出维度（例如词汇量大小）。本文使用 x_i 和 y_i 来表示第 i 个 Transformer 层的输入和输出，所有这些中间激活共享相同的形状 $\mathbb{R}^{n\times d}$。Transformer 层的前向传播，即 $f(x) = y$，计算如下[①]:

自注意力子块:

$$
\begin{aligned}
&Q = xW_q,\ \ K = xW_k,\ \ V = xW_v, && W_q, W_k, W_v \in \mathbb{R}^{d\times d},\\
&u = \mathrm{SoftMax}\left(\frac{QK^T}{\sqrt{k}} + M\right)VW_o, && M \in \mathbb{R}^{n\times n}, W_o \in \mathbb{R}^{d\times d},\\
&v = \mathrm{LayerNorm}(u + x; \gamma_1, \beta_1), && \gamma_1, \beta_1 \in \mathbb{R}^{d},
\end{aligned}
$$

前馈子块:

$$
\begin{aligned}
&z = \mathrm{ReLU}(vW_1)W_2, && W_1 \in \mathbb{R}^{d\times m}, W_2 \in \mathbb{R}^{m\times d},\\
&y = \mathrm{LayerNorm}(z + v; \gamma_2, \beta_2), && \gamma_2, \beta_2 \in \mathbb{R}^{d},
\end{aligned}
$$

其中 k 和 m 是依赖于模型架构超参数的常数，M 表示掩码矩阵。SoftMax、LayerNorm 和 ReLU 是常用的神经网络函数。在 L 层 Transformer 之后，分类器计算如下:

$$
o = \mathrm{SoftMax}\left(y_L W_c\right), \qquad W_c \in \mathbb{R}^{d\times s}.
$$

使用掩码。掩码矩阵是一个下三角矩阵，其中主对角线以上的元素被设置为负无穷，主对角线上及以下的元素被设置为零。在原始的 Transformer 工作[186]中，提出了两种 Transformer 层，编码器和解码器。掩码仅应用于解码器中的自注意力子块，以防止当前位置之后的位置被关注。这确保在生成输出序列中的每个词项时，模型只关注序列中在其之前的词项。掩码操作并不是微不足道的，因为它导致使用序列级置换构建等价计算变得不可行。

[①]为简化表达，此处使用 xW 代替实际实现中使用的 xW^T。

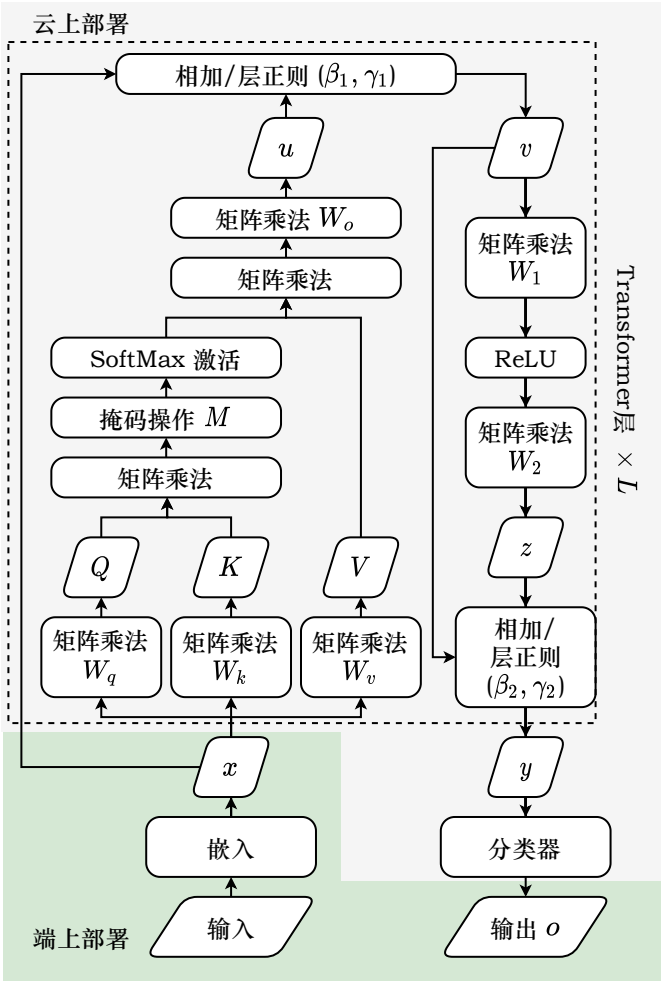


图 4.3　原始 Transformer 的推理工作流程

4.3.2　三方设置和威胁模型

对于提供 Transformer 模型，本文考虑三个参与方：

• 模型开发者（P_1）负责训练和拥有私有的 Transformer 模型参数，例如，OpenAI 开发的 GPT4。

• 模型服务器（P_2）具有计算硬件，例如，在 Azure 云平台上的 GPU 集群。

• 数据所有者（P_3）拥有隐私的输入和推理输出，例如，ChatGPT 用户的文本提示和响应。

推理协议应确保 P_1 和 P_2 不知道 P_3 的输入 x_1 和推理输出 o。P_1 的模型参数 θ 应对 P_2 和 P_3 都隐藏。参数 θ 包括注意权重（W_q, W_k, W_v, W_o），前馈权重 (W_1, W_2)，LayerNorm 权重 (γ, β) 和分类器权重 (W_c)。在本文考虑的 Transformer 推理场景中，P_3 的输入可以是文本提示[67]，图像[176]，以及多模态的多种组合[68]，而推理输出通常是最后分类头的概率向量[4]。

本文采用广泛使用的半诚实设置[40-42,187]，在该设置中，每个参与方将遵循协议规范，但尝试从观察到的协议消息中推断出其他敏感信息。

4.4　STIP 设计

本节首先介绍如何使用特征空间置换实现 Transformer 模型的等价推理。然后本节介绍核心安全推理协议，Secure Transformer Inference Protocol（STIP），并分析协议的安全性。

4.4.1　特征空间置换

置换操作由置换矩阵 π 定义，它是一个二值方阵，每行和每列都恰好有一个 1 的条目，所有其他条目都为 0。对 $x \in \mathbb{R}^{n\times d}$，$\pi_{n\times n}x$ 和 $x\pi_{d\times d}$ 分别代表执行序列级和特征级置换。

掩码使得序列级置换不等价。对于 Transformer 编码器（无掩码的自注意力），相关工作已经研究了序列级置换等变性属性，即 $f(\pi x) = \pi f(x)$[184]。然而，由于解码器中的掩码操作，对序列级置换的数据的注意力计算无法返回等价的输出。一个快速的解决办法是将一个经过置换的 $M' = \pi M\pi^T$ 发送到云计算平台。然而，由于 M 的值结构是已知的，云平台可以轻松推断出 π，这将导致保护丧失。

参数转换。相反，本文提出在特征空间中使用一组随机置换矩阵来转换参数。首先，为输入 x 生成 $\pi \in \{0,1\}^{d\times d}$。对于第 i 个 Transformer 层，使用另外三个矩阵 $\pi_{i,1}, \pi_{i,2}, \pi_{i,3}$ 来转换参数：

$$
\begin{aligned}
&W_q' = \pi^T W_q \pi_{i,1},\ \ W_k' = \pi^T W_k \pi_{i,1},\ \ W_v' = \pi^T W_v \pi_{i,2},\\
&W_o' = \pi_{i,2}^T W_o \pi,\ \ W_1' = \pi^T W_1 \pi_{i,3},\ \ W_2' = \pi_{i,3}^T W_2 \pi,\\
&\gamma_1' = \gamma_1\pi,\ \ \beta_1' = \beta_1\pi,\ \ \gamma_2' = \gamma_2\pi,\ \ \beta_2' = \beta_2\pi.
\end{aligned}
$$

对于分类器，需要生成一个置换矩阵 $\pi_c \in \{0,1\}^{s\times s}$。通过以下方式转换分类器参数：

$$
W_c' = \pi^T W_c \pi_c.
$$

图 4.4说明了参数转换过程。

计算等价性。设 $F_{\theta'}$ 表示具有转换参数的 Transformer 模型，本文证明了可以等价地恢复原始推理结果：

定理 4.1　$F_{\theta'}(x\pi)\pi_c^T = F_\theta(x)$。

证明　首先，由于非线性激活函数是逐元素计算的，这些函数都是置换等价的，即 ReLU($x\pi$)=ReLU(x)π 以及 SoftMax($x\pi$)=SoftMax(x)π.
之后证明：

LayerNorm($x\pi; \gamma\pi, \beta\pi$) =LayerNorm($x; \gamma, \beta$)$\pi$.

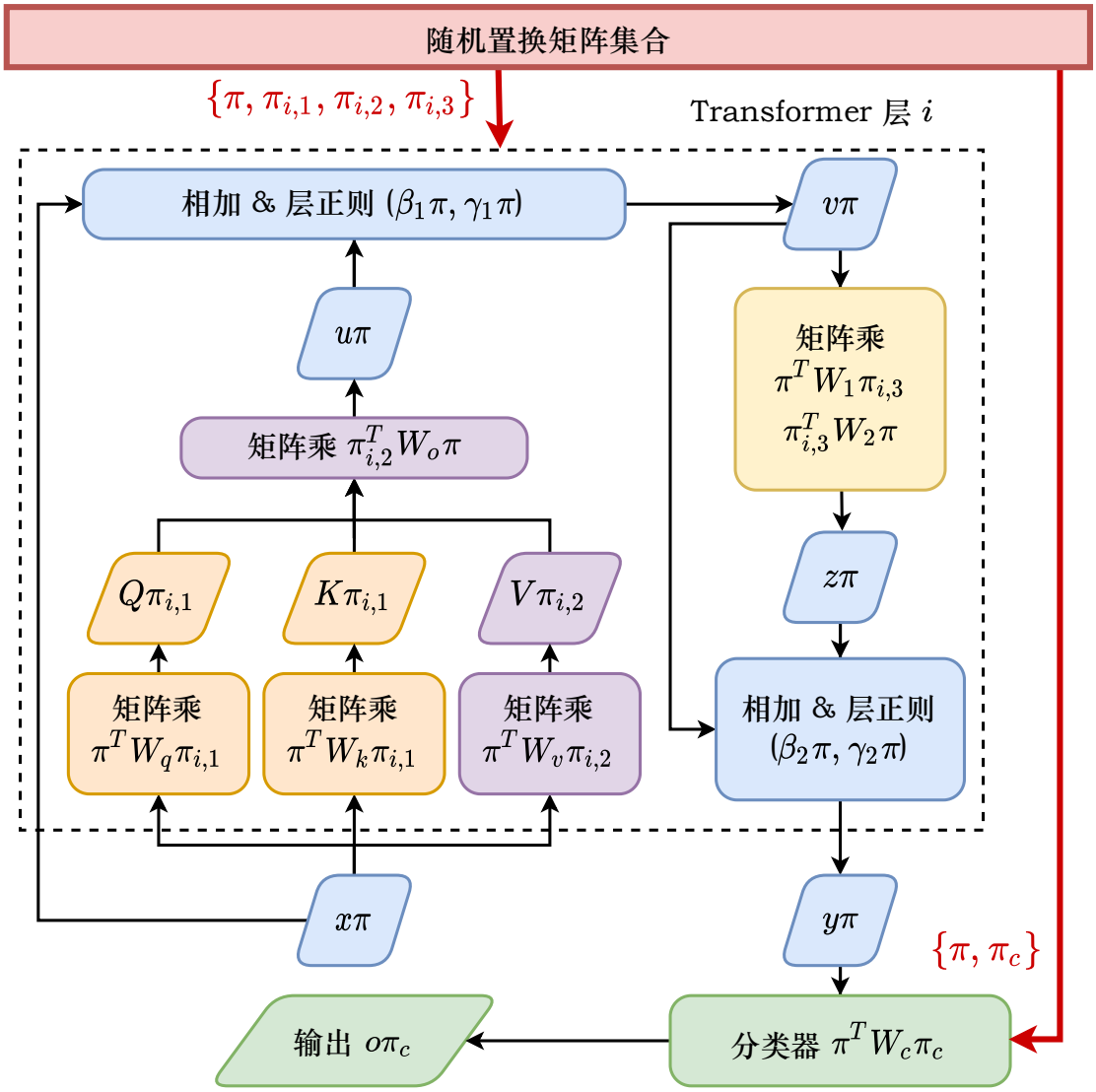


图 4.4　特征空间参数转换的示意图

以 $x \in \mathbb{R}^{n\times d}$ 为输入的 LayerNorm 函数定义为

$$\text{LayerNorm}(x;\gamma,\beta) = \gamma \circ \frac{x-\mu_x}{\sigma_x} + \beta, \qquad \gamma,\beta \in \mathbb{R}^d,$$

其中 $\circ$ 指 Hadamard（逐元素）乘积运算符。因为 μ_x 和 σ_x 是按行计算的，所以 $\mu_{x\pi} = \mu_x$ 且 $\sigma_{x\pi} = \sigma_x$。因此，

$$\begin{aligned}\text{LayerNorm}(x\pi;\gamma\pi,\beta\pi) &= \gamma\pi \circ \frac{x\pi-\mu_x}{\sigma_x} + \beta\pi \\ &= \left(\gamma \circ \frac{x-\mu_x}{\sigma_x} + \beta\right)\pi \\ &= \text{LayerNorm}(x;\gamma,\beta)\pi.\end{aligned}$$

由于 $\forall \pi, \pi\pi^T = I$:

$$
\begin{aligned}
Q' &= x\pi\pi^T W_q \pi_{i,1} = x W_q \pi_{i,1} = Q\pi_{i,1}, \\
K' &= x\pi\pi^T W_k \pi_{i,1} = x W_k \pi_{i,1} = K\pi_{i,1}, \\
V' &= x\pi\pi^T W_v \pi_{i,2} = x W_v \pi_{i,2} = V\pi_{i,2}, \\
u' &= \text{SoftMax}\left(\frac{Q'K'^T}{\sqrt{k}} + M\right) V' \pi_{i,2}^T W_o \pi \\
&= \text{SoftMax}\left(\frac{Q\pi_{i,1}\pi_{i,1}^T K^T}{\sqrt{k}} + M\right) V\pi_{i,2}\pi_{i,2}^T W_o \pi \\
&= \text{SoftMax}\left(\frac{QK^T}{\sqrt{k}} + M\right) V W_o \pi = u\pi, \\
v' &= \text{LayerNorm}(u' + x\pi; \gamma_1', \beta_1') \\
&= \text{LayerNorm}(u\pi + x\pi; \gamma_1\pi, \beta_1\pi) \\
&= \text{LayerNorm}((u + x)\pi; \gamma_1\pi, \beta_1\pi) = v\pi, \\
z' &= \text{ReLU}(v'\pi W_1')W_2' = \text{ReLU}(v\pi\pi^T W_1 \pi_{i,3})\pi_{i,3}^T W_2 \pi \\
&= \text{ReLU}(vW_1)W_2\pi = z\pi, \\
y' &= \text{LayerNorm}(z' + v'; \gamma_2', \beta_2') \\
&= \text{LayerNorm}(z\pi + v\pi; \gamma_2\pi, \beta_2\pi) \\
&= \text{LayerNorm}((z + v)\pi; \gamma_2\pi, \beta_2\pi) = y\pi, \\
o' &= \text{SoftMax}(y' W_c') = \text{SoftMax}(y\pi\pi^T W_c \pi_c) = o\pi_c.
\end{aligned}
$$

因此 $F_\theta'(x\pi)\pi_c^T = o'\pi_c^T = o\pi_c\pi_c^T = o = F_\theta(x)$. ∎

4.4.2　协议

基于提出的基于置换的 Transformer 模型变换，本文设计了 STIP 协议。图 4.5 展示了 STIP 流程。STIP 有两个阶段：初始化和推理。在初始化阶段，模型开发者 P_1 随机生成置换矩阵集合 $\Pi = \{\pi, \pi_c\} \cup \{\pi_{i,1}, \pi_{i,2}, \pi_{i,3} | i \in [L]\}$。$P_1$ 使用 Π 转换其拥有的训练模型 F_θ，得到变换后的 $F_{\theta'}$。然后，P_1 将变换后的模型 $F_{\theta'}$ 发送到云平台，并将输入和输出的置换矩阵（π 和 π_c）分发给其注册用户。此时，初始化阶段完成。在推理阶段，一旦用户想要使用推理服务，它在设备上运行嵌入操作以获得 x。然后，用户使用接收到的输入置换矩阵 π 对嵌入进行变换，通过超轻量级操作 $x\pi = x'$。然后用户将 x' 发送到云端。与正常的 Transformer 模型服务相比，云平台的工作负载没有变化。云端只是执行 $F_{\theta'}(x')$ 计算，并在置换特征空间中获得输出 o'。一旦用户从云端接收到返回的 o'，它只需通过 $o = o'\pi_c^T$

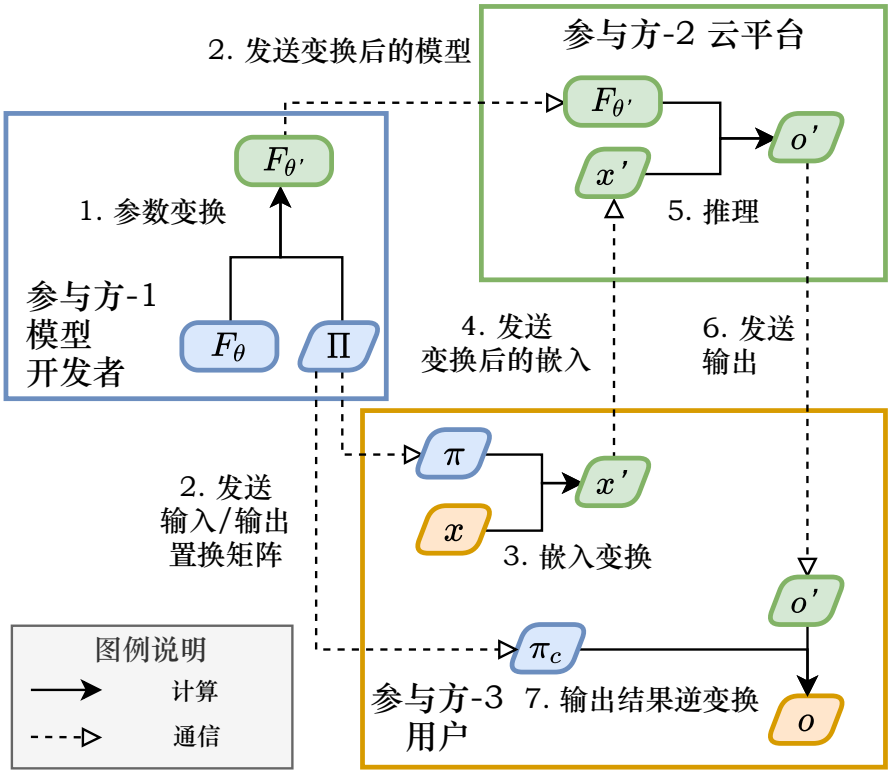


图 4.5　STIP 三方推理协议概述

反向变换输出，这只涉及内存移动操作，并且可以实现超高效。到目前为止，一轮推理完成。算法 4.1 形式化地展示了 STIP 协议。

多遍自回归生成。语言模型的服务基于自回归生成，在该生成方式中，模型按顺序预测词项，每次一个，每个词项都取决于给定的提示和先前生成的词项。STIP 可以通过要求用户将本地预测的词项与其初始提示结合，然后重新嵌入输入来支持自回归生成服务。与涉及单轮通信的全云推理相比，STIP 协议引入了 n 轮通信，其中 n 是最终生成的词项数量。然而，由于需要保持推理输出的机密性，这种额外的通信开销是不可避免的。实验证明，STIP 可以在有线连接下以 3 秒的延迟生成 100 个词项。考虑到全云推理的延迟超过 1 秒，STIP 在提供关键安全保障的同时在实际中仍是高效的。

生产环境支持。生产级别的 Transformer 推理框架，如 DeepSpeed[14] 和 HuggingFace[188]，融合了许多技术以提高效率。例如，它们使用 KV-cache 机制通过存储先前的注意力计算的中间结果来减轻冗余计算。STIP 保持了一种非侵入式方法，转换参数值同时保持底层 Transformer 体系结构不变。从云端的角度看，STIP 协议仅切换了一组不同的权重，而在推理计算过程中没有任何更改。因此，STIP 与生产级别的框架无缝对齐，本工作已经使用 HuggingFace[188] 库实现了 STIP 。

算法 4.1　STIP 推理协议

输入: Transformer 层数 L

初始化阶段:

$[P_1]$ $\Pi \leftarrow \{\pi, \pi_c\} \cup \{\pi_{i,1}, \pi_{i,2}, \pi_{i,3} | i \in [L]\}$;

$[P_1]$ $F_{\theta'} \leftarrow \texttt{ParaTrans}\,(F_\theta, \Pi)$;

$[P_1]$ $\texttt{Send}\,(F_{\theta'}, P_2)$ 和 $\texttt{Send}\,(\{\pi, \pi_c\}, P_3)$;

推理阶段:

$[P_3]$ $x' \leftarrow x\pi$ 和 $\texttt{Send}\,(x', P_2)$;

$[P_2]$ $o' \leftarrow F_{\theta'}(x')$ 和 $\texttt{Send}\,(o', P_3)$;

$[P_3]$ $o \leftarrow o'\pi_c^T$.

4.4.3　安全性分析

现在展示 STIP 如何保护模型参数和用户数据免受各种攻击，并使用距离相关性量化隐私泄漏风险的边界。

随机置换抵抗穷举攻击。首先，将 P_1 视为试图访问用户数据 x, o 的攻击者。由于 $x\pi$ 和 $o\pi_c$ 的不可访问性，P_1 无法使用 π, π_c 恢复 x, o。接下来，将 P_2 视为攻击者，针对模型参数 θ 和用户数据 x, o。鉴于 P_2 拥有变换后的参数和 $x\pi$，正确猜测 $\pi_{d\times d}$ 的概率是 $1/(d!)$，其中 d 在实际应用中通常大于 512，例如在 LLaMA2-7b 中 $d = 4096$[67]。这使得成功攻击的可能性微乎其微。值得注意的是，基于置换的保护方案通常在保持元素集（例如英语词汇）方面存在弱点[187]。幸运的是，STIP 通过将置换应用于中间激活而不是原始数据，避免了这种弱点。第三，将 P_3 视为针对模型参数 θ 的攻击者。由于 P_3 无法访问 θ'，即使拥有 π, π_c，也无法恢复 θ。由于 STIP 需要在设备上部署嵌入模型，因此嵌入的权重对 P_3 是可见的。然而，嵌入模块本身无法执行有价值的任务，因此不是敏感的（例如，OpenAI 已经发布了其嵌入模块[189]）。

半对称方案抵御已知明文攻击。已知明文攻击是一种密码攻击，对手同时拥有密文（加密数据）和相应的明文（未加密数据）。已知明文攻击的目标是揭示用于加密数据的密钥或算法。在本文的背景下，如果模型开发者参数的明文泄漏了，就没有必要继续攻击保护方案。因此，已知明文攻击的考虑重点完全放在用户数据上。假设 P_2 知道 x 和 $x\pi$，它可以通过 d 次列匹配来恢复 π，除非有两列或更多完全相同的列。因此，如果云端的参数完全依赖 π 进行保护，它们都有泄漏的风险。这强调了本文采用半对称保护方案的基本原理，其中层参数使用两个矩阵进行置换。一个完全归模型开发者所有，而另一个与用户共享。STIP 中的这种设计使模型参数对于已知明文攻击具有抵抗力。对于特定用户，揭示 π 将导致所有后续的嵌入都暴露给 P_2。为了解决这个漏洞，可以实施定期更改置换矩阵集合的策略（在极端情况下，使用一次性变换），这是抵抗已知明文攻击常用的做法[190-191]。

社交工程攻击。考虑在云平台欺骗性地假装是用户并获取了嵌入模型以及

$\boldsymbol{\pi}, \boldsymbol{\pi}_c$ 的情况下，它可能会潜在地揭示其他共享相同置换矩阵的用户的数据。为了抵御这种风险，模型开发者可以部署多个模型实例，每个实例都使用不同的转换。然后，用户可以被随机分配到共享一个模型实例（在极端情况下，每个用户可能有一个专用的模型实例），有效地降低通过这种社交工程攻击泄漏数据的风险。值得注意的是，由于上面讨论的已知明文攻击等原因，参数对于这种攻击仍然是有抵抗力的。

距离相关性边界。上文分析了 STIP 如何确保原始数据值不能被恢复。一个重要的方面是研究原始数据和置换后数据之间的相关性程度。为了量化隐私泄漏的风险，本文使用距离相关性[69]，这是两个随机向量之间依赖性的统计度量。让 Corr 表示距离相关性。基于现有定理[187]，已经证明:

$$\mathop{\mathbb{E}}_{\pi_{d\times d}, A\in\mathbb{R}^{d\times d}}[\mathrm{Corr}(x, xA\pi)] \leqslant \mathop{\mathbb{E}}_{B\in\mathbb{R}^{d\times 1}}[\mathrm{Corr}(x, xB)].$$

简单来说，这意味着通过对中间激活应用随机置换导致的隐私泄漏程度至少不大于使用一维降维导致的泄漏。而一维降维已经在实际中证明具有实用的隐私保护能力[192-193]。

模型切分考虑。默认情况下, STIP 仅在用户设备上部署嵌入模块, 如图 4.3所示。如果嵌入模块不放在端设备，那么设备需要将词项化的 one-hot 向量（一个矩阵 $\in \mathbb{Z}^{n\times s}$，其中 s 表示词汇量大小）传输到云端。尽管该矩阵可以被随机置换，但向量的内在 one-hot 性质使得云端容易恢复置换，从而使其不安全。另一方面，将更多的层分布到设备上也不是一个明智的选择。这主要是因为将额外的参数暴露给终端设备不仅会影响效率（正如图 4.9的实验结果所示），而且也无法增强模型对云设备攻击的理论抵抗力。

4.5　STIP 对 Transformer 变体的支持

本节讨论 STIP 如何支持各种 Transformer 变体，包括语言模型，多模态模型和混合专家模型。接着，本节建立了 STIP 的广义规则，并阐明其应用范围。

Pre-LayerNorm 结构。GPT 的第一个版本[194]直接采用了原始的 Transformer 解码器。GPT-2[179]引入了 Pre-LayerNorm，将层归一化移至自注意力和前馈子块的输入，形式上为:

$$\begin{aligned} v &= \mathrm{Attn}(\mathrm{LayerNorm}(x)) + x, \\ y &= \mathrm{ReLU}(\mathrm{LayerNorm}(v)W_1)W_2 + v, \end{aligned}$$

其中 Attn 表示自注意子块。从定理 4.1的证明中，可以证明使用相同的参数转换，这个定理对于 Pre-LayerNorm 结构仍然成立。

证明　从定理 4.1的证明中, 可以看到置换等价性质对于自注意子块成立, 即 $\mathrm{Attn}(x\pi) = \mathrm{Attn}(x)\pi$。因此

$$\begin{aligned}
v' &= \mathrm{Attn}(\mathrm{LayerNorm}'(x\pi)) + x\pi \\
&= \mathrm{Attn}(\mathrm{LayerNorm}(x)\pi) + x\pi \\
&= \mathrm{Attn}(\mathrm{LayerNorm}(x))\pi + x\pi \\
&= (\mathrm{Attn}(\mathrm{LayerNorm}(x)) + x)\pi = v\pi, \\
y' &= \mathrm{ReLU}(\mathrm{LayerNorm}'(v')W_1')W_2' + v' \\
&= \mathrm{ReLU}(\mathrm{LayerNorm}'(v\pi)\pi^T W_1 \pi_{i,3})\pi_{i,3}^T W_2 \pi + v\pi \\
&= (\mathrm{ReLU}(\mathrm{LayerNorm}(v)W_1)W_2 + v)\pi = y\pi,
\end{aligned}$$

其中 $\mathrm{LayerNorm}'$ 表示具有变换参数的层归一化。
因此，$F_\theta'(x\pi)\pi_c^T = F_\theta(x)$ 仍然成立。 ∎

RMSNorm 层。 LLaMA 系列[67]使用 RMSNorm[195]替代 LayerNorm。为了支持 RMSNorm 运算符，STIP 将其权重 γ 通过 $\gamma\pi$ 进行转换，然后可以证明

$$\mathrm{RMSNorm}(x\pi;\gamma\pi) = \mathrm{RMSNorm}(x;\gamma)\pi.$$

证明　RMSNorm 函数对 $x \in \mathbb{R}^{n\times d}$ 定义如下:

$$\mathrm{RMSNorm}(x;\gamma) = \gamma \circ \frac{x}{\sqrt{\frac{1}{n}\sum_i x_i^2}}, \qquad \gamma \in \mathbb{R}^d,$$

其中 $\circ$ 表示 Hadamard（元素级）乘法运算符。由于 $\sum_i x_i^2$ 是按行计算的，有 $\sum_i (x\pi)_i^2 = \sum_i x_i^2$。因此,

$$\begin{aligned}
\mathrm{RMSNorm}(x\pi;\gamma\pi) &= \gamma\pi \circ \frac{x\pi}{\sqrt{\frac{1}{n}\sum_i (x\pi)_i^2}} \\
&= \left(\gamma \circ \frac{x}{\sqrt{\frac{1}{n}\sum_i x_i^2}}\right)\pi \\
&= \mathrm{RMSNorm}(x;\gamma)\pi.
\end{aligned}$$

∎

GeLU 层。GPT 使用 GeLU 替代前馈子块中的 ReLU。与 ReLU 类似，GeLU 涉及无可学习参数的逐元素计算，因此有 $\mathrm{GeLU}(x\pi) = \mathrm{GeLU}(x)\pi$，定理 4.1成立。

SwiGLU 前馈。 LLaMA[67]在前馈层中使用 SwiGLU[196]代替 ReLU。令 $\mathrm{FFN}_{\mathrm{SwiGLU}}$ 表示使用 SwiGLU 的前馈层，其定义为：

$$\mathrm{FFN}_{\mathrm{SwiGLU}}(x) = \left(xW_1\mathrm{Sigmoid}(xW_1)xW_3\right)W_2,$$
$$W_1, W_3 \in \mathbb{R}^{d\times m}, W_2 \in \mathbb{R}^{m\times d}.$$

通过以下方式转换参数：

$$W_1' = \pi^T W_1,\ \ W_3' = \pi^T W_3\pi_{i,3},\ \ W_2' = \pi_{i,3}^T W_2\pi,$$

其中 $\mathrm{FFN}'_{\mathrm{SwiGLU}}$ 表示转换后的前馈子块。证明了 $\mathrm{FFN}'_{\mathrm{SwiGLU}}(x\pi)$ =$\mathrm{FFN}_{\mathrm{SwiGLU}}(x)\pi$。

证明　根据定义，

$$\begin{aligned}
&\mathrm{FFN}'_{\mathrm{SwiGLU}}(x\pi) = (x\pi W_1'\mathrm{Sigmoid}(x\pi W_1')x\pi W_3)W_2' \\
&= (x\pi\pi^T W_1\mathrm{Sigmoid}(x\pi\pi^T W_1)x\pi\pi^T W_3\pi_{i,3})\pi_{i,3}^T W_2\pi \\
&= (xW_1\mathrm{Sigmoid}(xW_1)xW_3)W_2\pi \\
&= \mathrm{FFN}_{\mathrm{SwiGLU}}(x)\pi.
\end{aligned}$$

∎

稀疏注意力。 GPT-3[197]在 Transformer 层中采用了稀疏注意力模式[198]。类似地，Longformer[181]被提出以提高对长上下文的内存效率。这些注意力的修改等同于修改掩码 M 矩阵。根据证明，定理 4.1对于任何 M 矩阵都成立，无需调整参数转换。

视觉模型。 ViT[176]将图像分成不重叠的补丁，并将每个补丁线性嵌入以创建词项嵌入的序列。这些词项嵌入作为 Transformer 模型的输入。由于 STIP 不依赖于原始数据的预处理，因此可以无缝支持 ViT。LLaVA[68]同时接受文本和图像作为输入。它使用视觉变换器嵌入图像，并随后通过线性投影 $x_v W$ 连接它们与文本输入的嵌入，其中 x_v 表示视觉嵌入。要将 LLaVA 与 STIP 集成，只需要通过 $\pi_v^T W \pi_t$ 转换投影权重 W，其中 π_v 和 π_t 分别表示用于视觉和文本变换器特征的置换矩阵。

混合专家模型。 Mixtral[180]通过在并行构建多个前馈子块（称为专家）的基础上，辅以一个路由器（或门控层）将混合专家集成到 Transformer 中。该路由器通过 $g(x) = xW_g$ 确定专家的权重，其中 $W_g \in \mathbb{R}^{d\times e}$，$e$ 表示专家的数量。为了支持混合专家模型，对 W_g 的简单变换就足够了，通过 $\pi^T W_g$ 实现。

应用范围。 对于具有可学习参数的层，STIP 仅要求这些参数涉及全局矩阵乘法（例如，线性、自注意和前馈）或词项级别的聚合（例如，LayerNorm）。举一些反例，STIP 不能扩展到卷积和循环层。对于没有可学习参数的层，STIP 要

表 4.3　测试平台和 Transformer 模型概述

测试平台	数据模态	Transformer 模型
校园安全 聊天机器人 (CHAT)	文本	GPT2/LLaMA2 系列
车辆座舱 场景理解 (CABIN)	图像	ViT/LLaVA 系列
模拟平台 (SIMU)	文本	BERT/Mixtral 系列

求它们满足 $f(x\pi) = f(x)\pi$ 属性，即按列的置换等变性。例如，ReLU、GeLU、SoftMax 和 Sigmoid 激活层。

4.6　验证实验

实验使用真实系统和公共数据集在各种 Transformer 推理任务上验证 STIP 。实验的主要发现如下:

• STIP 在模型参数和用户数据方面展示了实际的安全性且没有精度损失。

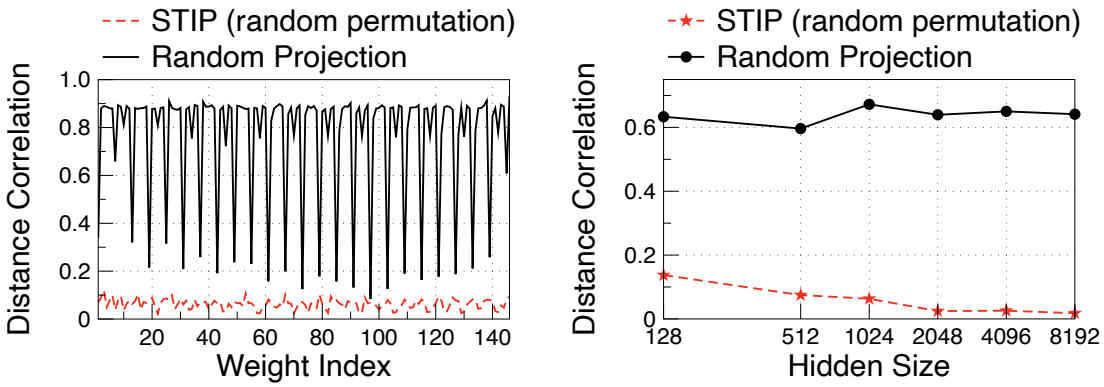
• STIP 实现了与未受保护的全云推理相媲美的吞吐量水平，比现有的安全两方协议[40-42]表现出数百万倍的性能。

• STIP 在验证各种微基准测试中表现出高效性，包括通信开销、内存占用和模型拆分的影响。

4.6.1　实验设置

实现。本工作使用 PyTorch 和 HuggingFace[188]库实现了 STIP 。现代深度学习框架，包括 PyTorch，采用行主内存布局。为了与内存布局保持一致，PyTorch 在线性层中执行矩阵乘法，形式为 xW^T 而不是 xW。因此，本工作对先前介绍的参数转换进行了转置以进行实现。对于置换操作，选择生成一个随机的置换向量 π_v 而不是一个矩阵 π_m。然后，该向量用于索引行或列，如 $W[:, \pi_v]$，这实现了与 $W\pi_m$ 相同的效果。基于索引的方法在计算上比矩阵乘法更高效。

测试平台和 Transformer 模型。实验在验证中使用了三个测试平台和六个代表性的 Transformer 模型，见表 4.3。(1) 校园安全聊天机器人（CHAT）。为了支持校园安全的自然语言问答，本工作开发了一个基于大型语言模型的聊天机器人。实验选择预训练的 LLaMA2-7b[67]实现此服务。为了扩大验证规模，实验还部署了 GPT2-124m/355m/774m/1.5b[179]和 LLaMA2-13b/70b 模型（连接符后的数字表示参数量）。(2) 车辆座舱场景理解（CABIN）。本工作使用 LLaVA-13b[68]实现了座舱场景理解功能。LLaVA 模型接收车内视频帧和预设提示作为输入生成场景描述。实验还部署了 ViT-86m/307m/632m 模型[176]进行实验。(3) 模拟器 (SIMU)。为了进一步验证 STIP 对 BERT 系列[199]和 Mixtral[180]模型的性能，实验构建了



(a) GPT2 模型参数　　**(b) 特征嵌入**

图 4.6　基于距离相关性的隐私泄漏量化

一个模拟测试平台，并使用随机生成的数据进行测试。

基线。为了进行比较，实验考虑了四种方法：（1）全云。使用原始参数在云端部署 Transformer 模型，设备将原始数据（明文）发送到云端进行推理。（2-4）Iron[40]，THE-X[41]和 CipherGPT[42]。它们提出了用于提供 BERT 系列和 GPT-2 模型的安全两方协议。

设备。对于所有情况，实验使用一台配备 4 个 NVIDIA A100 GPU 的服务器作为模型服务器。在 CHAT 测试平台中，实验使用一台配备 8 核 Intel Core i7 CPU 的 PC 作为用户设备。在 CABIN 测试平台中，实验使用一块 NVIDIA Orin 开发板作为用户设备。对于 SIMU，实验使用一台配备 4 核 Intel Core i5 CPU 的 MacBook Pro 笔记本电脑作为用户设备。

4.6.2　安全性和精度保证

首先，通过实验证明先前证明的安全性和计算等价性。

距离相关性。前文的隐私分析采用距离相关性[69] 作为验证隐私泄漏的指标。作为基线，实验在参数和嵌入上都使用随机线性投影，称为随机投影。在图 4.6a中，实验展示了 GPT2-1.5b 模型的原始参数和转换后参数之间的距离相关性。值得注意的是，与随机投影相比，STIP 表现出显著较低的距离相关性。平均而言，随机投影的距离相关性为 0.76，而 STIP 实现了显著较低的值 0.062。为了验证设备上数据的安全性，实验对嵌入应用了不同隐藏大小（从 128 到 8192）的变换。图 4.6b展示了生成的距离相关性。在随机投影的情况下，转换后的数据平均保持 0.6 以上的相关性。相反，STIP 的距离相关性随着隐藏大小的增加而减小，范围从 0.14 到 0.017。这展示了 STIP 在降低与转换数据相关的隐私泄漏方面的有效性。实验结果证实了基于置换的转换数据和参数的低隐私泄漏性，为前文在第 4.4.3节中的分析提供了验证。

表 4.4　未经授权使用云端转换模型生成的无意义语句

实际生成	云端生成
提示词：I'm a language model,	
but what I do in that role is to change everything in our lives.	examines Blazers consolationtechorate applicationkiJanuary PLANkikiorate Blazers consolation Beyondki

表 4.5　STIP 实现了精度无损的 Transformer 推理

模型	GPT-2	LLaMA2	ViT
参数	124m/355m/774m/1.5b	7b/13b/70b	86m/307m/632m
绝对差值之和	0.021/0.033/0.0478/0.051	0.009/0.012/0.012	3e-4/3e-4/3e-4
分类准确率	100%	100%	100%

模型	BERT	LLaVA	Mixtral
参数	4m/41m/110m/336m	13b	47b
绝对差值之和	5e-3/8e-3/9e-3/9e-3	0.016	0.008
分类准确率	100%	100%	100%

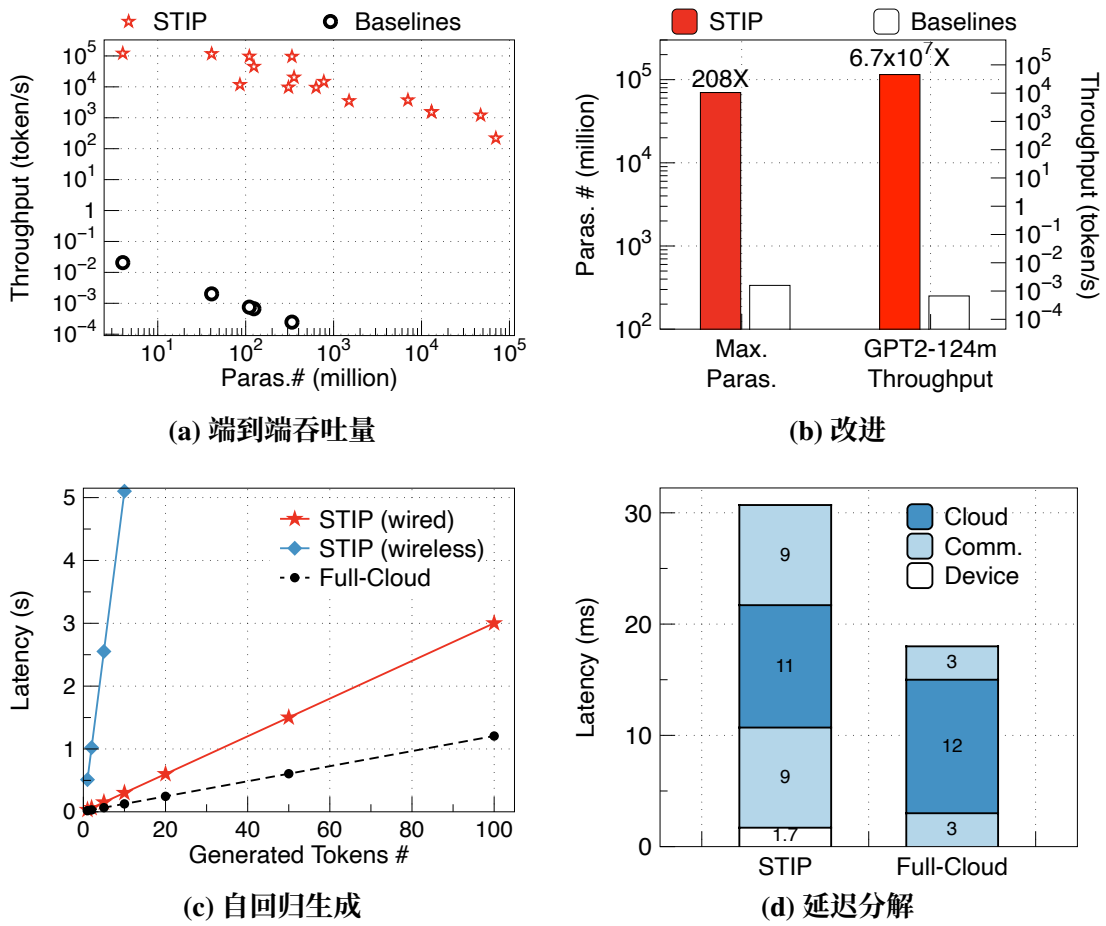
云端参数不可用性。除了量化相关性之外，实验还通过使用转换后的参数生成词项来验证云端参数的实际不可用性。实验使用相同的提示，并将它们提供给使用原始和 STIP 转换后参数的 GPT2-1.5b 模型。表 4.4展示了结果。使用提示"I'm a language model,"，通过云端转换后的参数生成的词项完全没有意义，突显了云上部署参数的实际不可用性。这一观察结果强调了 STIP 在保护参数免受未经授权使用方面的有效性。

精度无损失。 STIP 的一个关键优势在于其计算等价性，确保使用 STIP 为 Transformer 模型提供服务不会损失精度。实验通过检查两个指标来验证这一点：预测的绝对差异之和和前 1 个词项分类精度。实验对所选的所有六个模型系列进行了测试，参数范围从 400 万到 700 亿，每个模型使用 10000 个样本。如表 4.5所示，STIP 在所有模型上均实现了 100% 的精度。值得注意的是，轻微的非零绝对差异是由于固有的浮点运算误差引起的，而不是由 STIP 引入的精度损失。

4.6.3　推理效率

接下来，实验验证了 STIP 的推理效率。

端到端吞吐量与参数规模的可扩展性。实验进行了测试，验证了使用 STIP 为 Transformer 模型提供端到端吞吐量。批处理大小设置为 100，每个样本的词项数设置为 100。如图 4.7（a）所示，与基线相比，STIP 展示了数量级的更高吞吐量。请注意，由于安全协议[40-42]缺乏开源代码，基线的吞吐量是从其论文中报告的结果推断出来的。此外，实验进行了全云推理测试，但结果接近 STIP，导致



(a) 端到端吞吐量　(b) 改进

(c) 自回归生成　(d) 延迟分解

图 4.7　STIP 实现高效 Transformer 推理

标记重叠，因此为了清晰起见被省略。为了提高最大参数数量，STIP 达到了 700 亿，而基线为 3.36 亿，增加了 208 倍。对于 GPT2-124m 的吞吐量，CipherGPT 报告为 6.7e-4 词项/秒，而 STIP 实现了 45,366 词项/秒，显示出了 670 万倍的改进。图 4.7（b）总结了参数规模和吞吐量的改进。

自回归生成。除了单轮前向传播测试，实验还对使用 STIP 进行自回归生成进行了测试，考虑了 STIP 通信的有线和无线网络连接。有线连接的平均通信延迟约为 10 毫秒，而无线连接的延迟约为 250 毫秒。使用批处理大小为 1 和 128 个输入提示，图 4.7（c）展示了 LLaMA2-7b 模型的结果。所有服务方法的延迟随生成的词项数量呈线性增加。全云（Full-Cloud）、STIP 有线和 STIP 无线的结果线的斜率分别约为 12、30 和 510。如第 4.4.2节中所讨论的，为了确保输出隐私保护，每生成的词项的通信成本是不可避免的。考虑到 STIP 相对于未受保护的全云推理引入的实际安全性，略高的延迟（例如，100 个词项多 2 秒）被认为是可以接受的。

延迟分解。为了深入了解 STIP 引入的开销，实验对延迟分解进行了分析，并将其与全云推理进行了比较。如图 4.7d所示，验证显示 STIP 在设备上引入了额

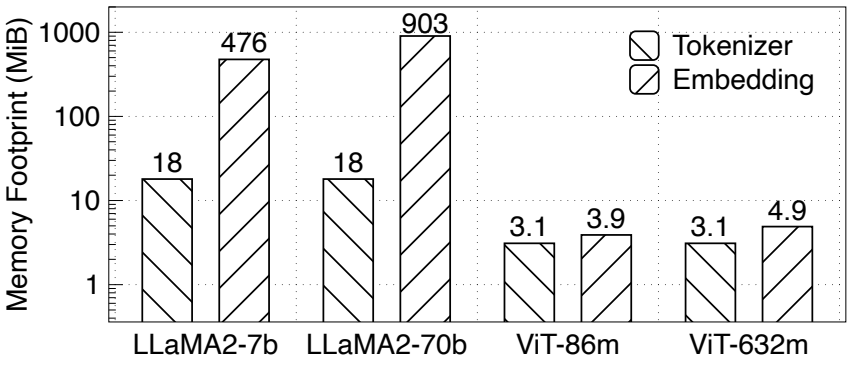


图 4.8　STIP 协议下设备上的内存使用情况

外的 1.7 毫秒延迟，同时将云端延迟从 12 毫秒降低到 11 毫秒。导致 STIP 性能较全云推理较慢的一个关键因素是通信阶段。这是由于需要传输中间嵌入，即一个 批大小 $\times n \times d$ 结构的张量，通常超过在全云服务中传输的明文单词的大小。虽然之前的工作[200]已经研究了在模型拆分场景中压缩中间激活并提高通信效率的技术，但值得注意的是，本文的工作对无损精度提出了严格的要求，使得这些压缩技术超出了当前的设计范围。将这些压缩方法与 STIP 集成是未来研究的有前途的方向。

4.6.4　微基准测试

端云通信流量。STIP 引起的通信流量受三个因素的影响：输入词项数量、隐藏大小和输出词汇表大小。举例说明，考虑 GPT2-124m 模型，单轮推理操作导致输入嵌入和输出激活分别产生 5.8 MiB 和 7.5 MiB 的流量。如图 4.2所示，与 CipherGPT（95,151 MiB）相比，STIP 引起的通信流量明显较低。这种大幅减少的流量突显了 STIP 在适度成本下实现安全性的能力。

设备内存占用。鉴于可能用于基于 Transformer 的服务的各种设备，实验验证了设备内存占用。图 4.8中的结果展示了内存需求。对于标记器组件，LLaMA2 和 ViT 模型的内存占用分别为 18 MiB 和 3.1 MiB。对于嵌入部分，内存分配取决于隐藏大小参数。采用大隐藏大小（8192）的 LLaMA2-70b 产生了 903 MiB 的内存成本。相比之下，ViT 模型展示了更为适度的内存需求，范围从 3.9 MiB 到 4.9 MiB。这表明，即使对于具有大隐藏大小的模型，STIP 在设备上的内存需求仍然适用于终端设备。

模型切分的影响。实验改变了设备上的 Transformer 层数，从 0 增加到 20，并分析了对推理延迟的影响。如图 4.9所示，端到端推理的延迟随着设备上层数的增加而成比例地上升。这种延迟增加归因于设备相对于云的计算能力较低。正如在第 4.4.3节中讨论的，将更多层部署到设备上不仅导致更高的延迟，还会向用户暴露更多参数，从而引入隐私风险。考虑到这些因素，只在设备上部署嵌入模

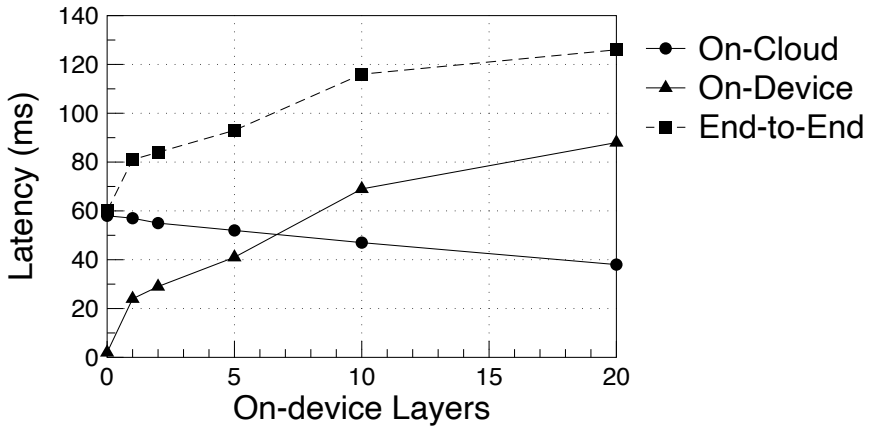


图 4.9　STIP 协议在不同模型切分下的延迟

块是最佳选择。这种配置可以最小化延迟，同时减轻向用户暴露更多层次可能带来的潜在隐私风险。

讨论（1）开发内部模型的云服务商。STIP 的一个局限性源于其三方设置。尽管这种设置适用于许多应用，如第 4.1.2节中说明的那样，但在云服务商也参与模型开发时可能不太适用。考虑到云服务商（例如 Google）在其云平台上维护大型 Transformer 模型[163-164]的情景。尽管可以通过在同一公司的不同部门之间实施访问控制来避免共谋，但对于外部用户仍然存在一定程度的怀疑。困难在于建立一个值得信赖的环境，特别是当外部用户无法保证云服务商内部模型开发和服务过程中利益分离的情况。

（2）支持训练的扩展。STIP 旨在解决前向传播的问题，模型训练则引入了反向梯度传播的复杂性。STIP 的基本原理有望被扩展以支持隐私保护的训练。然而，这一扩展涉及研究与梯度相关的通信开销以及在梯度传播过程中引入的额外隐私泄漏风险[201]。未来需要进行进一步的探索，以充分实现 STIP 在隐私保护 Transformer 训练方面的潜力。

4.7　小结

本章首先定位出针对 Transformer 模型安全推理协议的两方设置（模型所有者和数据所有者）中存在的固有效率瓶颈，以及这些瓶颈与实际应用的不一致之处。与传统的同态加密和安全两方计算框架不同，本章提出了一个新的三方威胁模型，将模型所有者分解为两个不同的实体：模型开发者和模型服务器。基于这一模型，本章提出了第一个用于三方 Transformer 推理的安全协议 STIP，并证明了其具有隐私泄漏的理论界限和精度无损的保证。本章实现了 STIP 并在实际的端云协同推理系统中验证了各种 Transformer 模型（其中最大的具有多达 700 亿参数），覆盖了文本、图像以及文本-图像多模态输入。实验结果表明，STIP 的效

率可与未保护的全云推理相媲美，且实现了精度完全无损的推理结果。

第 5 章　边云协同的自适应模型部署

多模型推理工作负载日益普遍，例如智能音箱助手[202]、智能城市[1]、基于无人机的视频监控[115]、多模态自动驾驶[203]等。在多模型部署阶段，除了考虑模型的精度之外，模型推理开销也可能成为影响服务质量的瓶颈，特别是对于延迟敏感的任务和资源有限的设备。

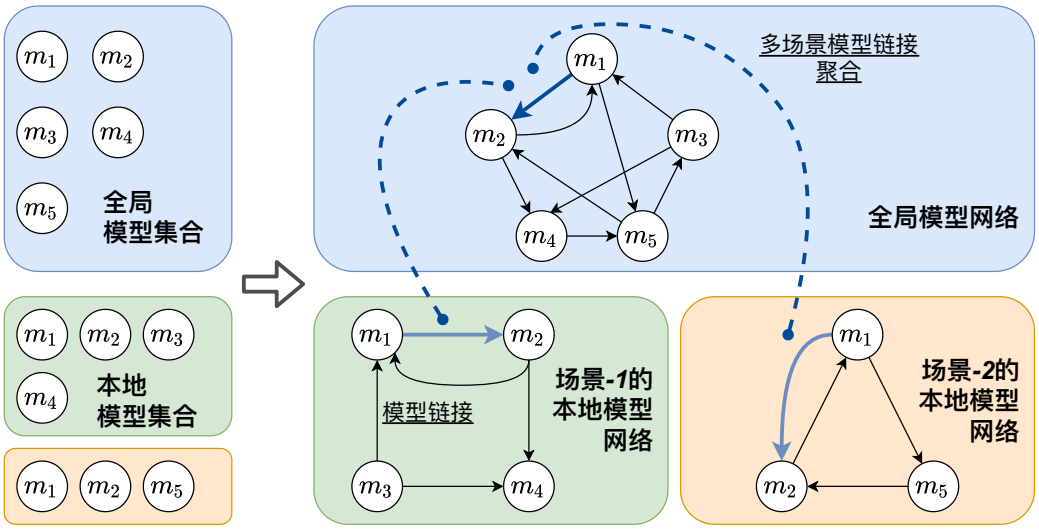
为了实现成本高效的推理，现有的工作从各种角度探索了权衡资源和性能的方法：多任务学习（Multi-Task Learning）[73,101,204-205] 可以通过在不同任务之间共享神经元来减少计算开销；模型压缩（Model Compression）[206-209] 技术试图通过消除与推理精度无关的参数和连接来减小模型体积；推理重用（Inference Reuse）[210-211] 方法旨在避免相同或相似的计算；数据源过滤（Source Filtering）[58] 方法试图仅向后端模型传递必要的输入数据。自适应配置（Adaptive Configuration）[212] 和多模型调度（Multi-Model Scheduling）[74]技术使推理工作负载能够适应输入内容的动态变化。本章将这些方法统一总结为对下面这个问题的回答：

在不完全执行模型的情况下，如何尽可能获得准确的推理结果？

从这个角度来看，多任务学习和模型压缩通过修剪原始模型生成相同推理任务的轻量版本模型。推理重用和数据源过滤技术通过分析输入之间的相关性将先前的推理结果重新用作预测结果。自适应配置通过分析输入的动态分布，通过执行轻量模型来代替复杂模型得到推理结果。多模型调度则利用执行过的模型的输出作为提示信息，将不必要执行的模型推理结果预测为空。

本章从一个新的角度来解决多模型部署问题：**链接黑盒模型**。对于机器学习模型，存在一个现象，即使模型在输入模态、学习任务、架构等方面有所不同，它们也可以相互共享知识。其原因在于模型容易出现过拟合[70]，不同模型的输出具有语义相关性[71]。这一特点使得有机会将原本独立存在的多个模型，通过知识层级的“模型链接”（Model Link），组织成一个在语义上相互关联的网络结构，本文称之为“模型网络”（Network of Models）。图 5.1展示了本文提出的模型链接方法将现有的多模型独立推理改变为协同推理，其中 m_1-m_5 表示模型，箭头指示本文提出的模型链接，蓝色虚线表示跨领域模型链接的聚合过程。要实现这一愿景，仍需要解决以下三个主要挑战：

(1) **如何在黑盒且异构的模型之间建立知识级别的链接？**在实际应用中，部署的模型可能具有不同的架构和输入模态，它们可能由不同的编程语言和机器学习框架开发。另一方面，理想的模型链接对原始推理系统应该是非侵入式的，并尽可能少地需要模型信息和代码修改。模型的异构性以及黑盒访问权限使得设计模型之间的通用知识级链接模型成为一项具有挑战性的任务。



(a) 独立推理　　**(b) 使用MLink进行协同推理**

图 5.1　对比独立推理和基于模型链接的协同推理

(2) **如何在动态数据分布中实时适应模型链接?** 由于输入的动态性，模型链接面临与传统机器学习相似的适应性挑战，即领域适应（Domain Adaptation）[20]。领域漂移（Domain Shift）通常由两个因素引起，即实时数据内容的在线动态性和应用场景的差异。例如，在不同摄像机拍摄的视频流上运行计算机视觉模型或在不同时间拍摄的视频上运行模型都面临分布漂移[146]。

(3) **如何高效选择模型子集进行部署?** 在完成模型之间的链接构建之后，现实应用常常需要在某个成本预算（例如边缘设备上可分配的 GPU 内存或推理服务最大可容忍延迟）下选择要部署的模型子集。高效的模型选择对于成本性能权衡至关重要。经分析，模型子集选择问题理论上是 NP 困难的，由于其理论难度，高效地完成选择是一件具有挑战的任务。

针对多模型部署存在的诸多技术挑战，本章首先形式化了模型链接任务，并提出了支持链接异构黑盒模型的模型链接设计，称为 MLink。接着，本章提出了模型链接的适应和聚合方法，优化了模型链接在面对在线动态和跨领域分布偏移问题时的表现。本章开发了一种基于模型链接的算法，用于在成本预算下部署多模型推理。实验使用一个包含七个不同模型的多模态数据集（涵盖了五类学习任务和三种输入模态）上验证了 MLink，结果表明本章提出的模型链接可以有效地在异构黑盒模型之间建立。实验还在两个真实的视频分析系统上进行了验证，一个用于智能建筑，另一个用于城市交通监控，包括六个视觉模型和来自 58 台摄像机的 3264 小时视频。实验结果显示，提出的在线自适应训练方法有效地提高了性能，比原始的离线训练更为优越。而提出的聚合方法实现了比原始模型高 7.9%的平均准确度。在 GPU 内存预算下，MLink 优于多种基线方法（多任务学习[73]，基于深度强化学习的调度器[74]和帧过滤[58]），可以节省 66.7%的推理计算同时保持 94%的输出准确度。

5.1　问题定义

本节定义了模型链接任务以及在给定成本下的多模型部署问题。

模型链接。给定一组黑盒模型 $F = \{f_i\}_{i=1}^{k}$，其中 $f_i : X_i \to Y_i$ 是将输入映射到推理结果的函数。模型可能是高度异构的，即具有不同的输入模态、学习任务、架构等。对于这样的一组模型，本文仅假设输入空间 $\{X_i\}_{i=1}^{k}$ 相同或对齐，下面对此假设的合理性进行说明：不同模型共享相同输入空间的情况很常见，例如基于多任务学习的机器人[73,101]和多媒体（包含视频、音频、文本等）语义广告[213]。对于多模态场景，例如多模态事件检测[214]和视觉语音合成[215]，输入空间通常是对齐的，即不同模态数据从不同角度反应着相同事件。在实践中，通过时间上的同步可以轻松地将许多应用的输入对齐，例如对齐视频帧和对应的音频。此外，针对特定场景，可以采用多视图视频的空间对齐[216]和音频-视觉语义对齐[217]等方法。本文将模型链接定义为一个函数 $g_{ij} : Y_i \to Y_j$，表征一个从源模型 f_i 的输出空间到目标模型 f_j 的映射。这样的话，复合函数 $g_{ij} \circ f_i : X_i \to Y_j$ 便可以实现预测 f_j 的推理计算。相应地，本文称 g_{ji} 将 f_i 的知识链接到 $g_{ji} \circ f_j$。

多源模型链接集成。当模型数量 $k \geqslant 3$ 时，对于同一个目标模型 f_j，可能存在来自不同源的多个模型链接。设 $A \subseteq F$ 表示源模型的集合。那么对于所有 $f_i \in A$, $g_{ij} \circ f_i$ 执行预测 f_j 推理输出的任务。接下来的问题是，如何根据多个源模型的模型链接来确定对于目标模型最终的预测？从集成学习（Ensemble Learning）的角度来看，$\{g_{ij} \circ f_i\}_{f_i \in A}$ 构成了一个多专家模型（Mixture of Experts）[218]，因此具有使用多任务和多模态表示进行更好预测的潜力[101,215]。本文定义 $h_{A,j}$ 为从 A 到 f_j 的集成模型链接。因此，$h_{A,j}$ 的输入是由 g_{ij} 给出的预测集合，其中 $f_i \in A$。请注意，如果 A 只有一个元素 f_i，那么 $h_{A,j} = g_{ij} \circ f_i$。

在成本预算下的多模型推理。在多模型部署场景下，如上定义的模型链接可以用于实现多模型推理任务的资源性能权衡。设 $c(\cdot)$ 表示运行函数的成本，例如 GPU 内存或推理时间。对于资源有限的设备（例如智能手表和手机）和对延迟敏感的任务（例如实时视频分析和语音助手），一般都会对总成本有一定的约束。本文将 B 定义为成本预算，优化目标是在该预算下最大化推理精度。设 $p(h_{A,j})$ 表示集成模型链接的性能度量，这取决于目标模型的任务。例如，性能度量可以是分类任务的分类精度和物体检测任务的边界框 IoU。不失一般性地，本文假设 p 的范围被归一化为 $[0, 1]$。与之前为了优化推理效率的工作一样[58,74]，本文考虑的性能度量是获取的推理结果和准确推理输出之间的一致性，而不是与真实标签（Groundtruth）的一致性。基于上述定义，在给定的成本预算下的多模型推理问题可以形式化为：

$$\begin{aligned} &\max_{A\subseteq F}\overbrace{(\frac{1}{|F|}(\underbrace{\sum_{f_i\in A}1}_{\text{激活的}}+\underbrace{\sum_{f_j\in F\setminus A}p(h_{A,j})}_{\text{预测的}}))}^{\text{平均输出精度}} \\ &s.t.\quad \underbrace{\sum_{f_i\in A}c(f_i)}_{\text{准确推理开销}}+\underbrace{\sum_{f_j\in F\setminus A}c(h_{A,j})}_{\text{模型链接开销}}\leqslant B. \end{aligned} \tag{5.1}$$

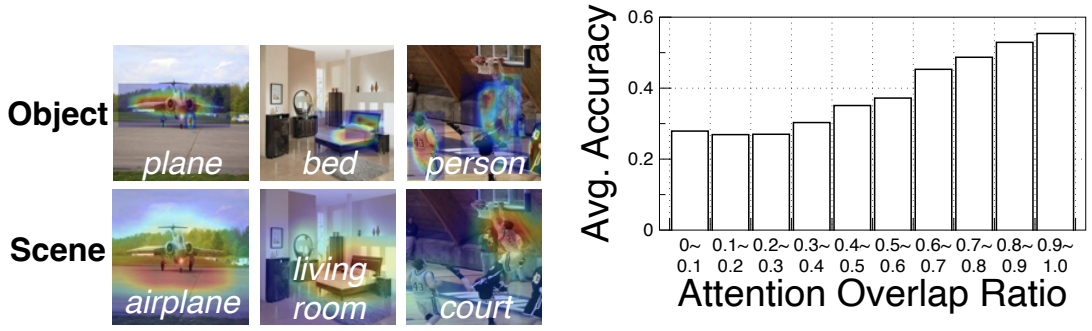
在成本预算 B 下，优化问题的目标是通过选择要执行的激活的模型子集 A 来最大化所有模型 F 的平均性能。为了方便描述，本文将目标函数描述为输出精度。激活的模型进行准确推理，因此它们的性能分数都是 1。未激活的模型只参与构建模型链接，在推理阶段不会被执行；相反，它们的推理结果通过集成模型链接由激活的模型进行预测得到。激活模型的成本是执行准确推理，而预测模型的成本来自运行模型链接。因此，理想的模型链接应既准确又轻量，以减少成本同时保持多模型推理的精度。

5.2　MLink 设计

本节讨论链接黑盒模型的动机，并介绍模型链接的理论分析、架构设计、集成和训练方法。

在为不同任务训练模型时，它们学到的理想表示应该是相互独立和解耦的[219]，即每个模型只学到与其目标任务相符的语义。然而，由于数据和模型的复杂性不匹配，机器学习过程容易发生过学习（Over-Learning）[70]，这意味着在学到的表示中编码了不需要的语义。此外，不同任务的输出之间存在语义相关性，不同模型可能关注相同的内容，例如图像中的相同区域。例如，在图 5.2a中, 基于 G-CAM[220], 绘制了 YOLO-V3[221] 目标检测器和 ResNet50[222] 场景分类器在相同图像上的注意力（Attention）热图，它们的注意区域有很大的重叠。本文在注意力热图的重叠比例与模型链接性能之间进行了实验。例如，从场景分类模型到目标检测模型，如图 5.2b所示，模型链接的精度明显与重叠比例 $((Map_{source}\wedge Map_{target})/Map_{target})$ 呈正相关。在一定程度上，这表明模型链接学到的相关性与语义注意力相似。过学习的特征和输出之间的潜在语义相关性使得从相同或对齐的输入空间到不同输出空间的映射是可迁移的[71]。

一个关键的设计原则是本文只使用源模型的黑盒输出进行模型链接。现有研究表明，通过微调（Finetuning）最后几层[223]，神经网络的中间表示可以用于预测其他不同的任务。然而，在实际应用中，通常需要面对只提供黑盒推理 API



(a) 模型的注意力热图　　**(b) MLink 精度与注意力重叠的关系**

图 5.2　模型间语义相关性的可视化和量化

的模型，例如通过容器（Container）的形式。与中间表示相比，下游的黑盒输出在一般学习任务的表示能力上确实较弱。但最近的工作[74]表明，在给定相同（或对齐）的输入的情况下，执行模型的输出对于调度未执行的模型非常有效。其见解是，相同输入的多个任务之间的黑盒输出的相关性比中间特征更为明确，甚至更强。实验结果也表明，使用相同数量的训练数据，黑盒模型链接实现了比知识蒸馏（Knowledge Distillation）方法更高的精度（见图 5.5）和比多任务学习方法更高的精度（见表 5.5）。考虑到更好的实用性和令人满意的精度，本文选择黑盒输出而不是中间表示来链接模型。

样本复杂性分析。设 $f \in \mathcal{F}$ 表示任务特定的参数，h 表示跨任务共享的参数。已经证明，当 h 的训练数据丰富时，为了在新任务上实现有界的预测误差，只需要 $C(\mathcal{F})$ 的样本复杂性[224]，其中 $C(\cdot)$ 是假设族的复杂性。学习从源模型 $f_i \in \mathcal{F}_i$ 到目标 $f_j \in \mathcal{F}_j$ 的模型链接 $g_{ij} \in \mathcal{G}$ 构成一个复合学习模型 $g_{ij} \circ f_i$。模型链接的轻量设计使得 $C(\mathcal{G}) < C(\mathcal{F}_j)$ 成立。因此，应用上述结果，与从头学习目标模型的 $C(\mathcal{F}_j)$ 复杂性相比，模型链接可以显著降低样本复杂性至 $C(\mathcal{G})$。这个结果也得到了实验证实：有效的模型链接可以通过非常小的训练样本量（例如 1%）学到（见图 5.5）。

5.2.1　模型链接架构

本文提出的模型链接在黑盒模型的输出空间之间进行映射，因此输出格式决定了架构。本文将输出格式分类为固定长度向量（表示为 Vec，Vector）和可变长度序列（表示为 Seq，Sequence），这两种类型的输出可以涵盖大多数模型。分别以这两种类型的输出作为源模型和目标模型，本文基于类似学习任务的最佳实践提出了四种模型链接架构。

Vec-to-Vec：从向量输出源映射到向量输出目标的模型链接。本文使用 ReLU 激活的多层感知机（MLP）作为 vec-to-vec 模型链接。

Seq-to-Vec：从序列输出源映射到向量输出目标的模型链接。本文首先使用

嵌入（Embedding）层，该层通过矩阵乘法将序列转换为固定大小的嵌入。然后，使用一个 LSTM[225]层，后面是一个 MLP 来生成向量输出。

Vec-to-Seq：从向量输出源映射到序列输出目标的模型链接。本文采用编码器-解码器（Encoder-Decoder）框架，其中 MLP 作为编码器，解码器包括一个嵌入层，一个 LSTM 层，一个注意力层[226]和一个全连接层，按照正向顺序排列。

Seq-to-Seq：从序列输出源映射到序列输出目标的模型链接。本文采用序列到序列（Seqence-to-Sequence）框架[227]，其中嵌入层后面是一个 LSTM 层作为编码器，解码器与 vec-to-seq 模型链接中的解码器相同。

输出激活函数由目标模型的学习任务确定，例如 Softmax 用于单标签分类，Sigmoid 用于多标签分类和序列预测，而线性激活适用于回归和定位任务。在本文的实现中，隐藏单元的默认数量是输出维度的两倍，这在经验上实现了效果和效率之间的良好平衡。

多源链接集成。多源模型链接的集成有可能提高预测性能[228]，因为跨任务和跨模态的表示能力可能是有益的。对于目标模型 f_j，给定源集合 A，本文通过可训练的权重将 g_{ij} 的输出与 $f_i \in A$ 相乘。然后，根据 f_j 的学习任务激活加权预测。$h_{A,j}$ 的学习权重可以用于集成来自任何源子集的模型链接，即 $h_{A',j}, A' \subset A$。

5.2.2　模型链接训练

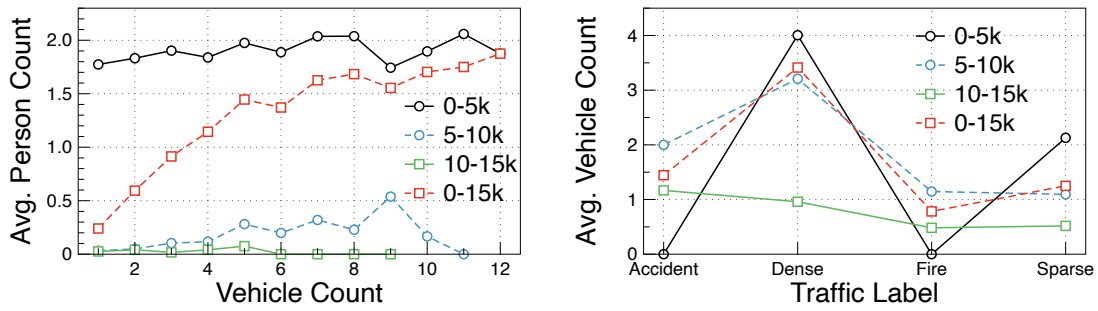
经典的知识蒸馏[206]表明，对于训练学生模型，软标签（Soft Label，即包含各类分类置信度的标签，而非单一最高置信度类别的标签）监督更好，因为教师模型的输出通过不同类别之间的关系扩充了硬标签（Hard Label）空间。实验结果表明，这种经验在本文提出的模型链接设置中仍然有效。为了训练模型链接和集成模型，本工作收集 k 个模型在相同或对齐输入上的 n 个推理结果 $\{\{y_i^j\}_{j=1}^k\}_{i=1}^n$。将 f_i, f_j 分别作为源和目标，训练模型链接 g_{ij} 的目标是：

$$\min \sum_{l=1}^{n} \mathcal{L}_j(g_{ij}(y_i^l), y_j^l), \tag{5.2}$$

其中损失函数 $\mathcal{L}_j$ 依赖于目标模型 f_j 的学习任务。将 A, f_j 分别作为源集合和目标，训练集成模型 $h_{A,j}$ 的目标是：

$$\min \sum_{i=l}^{n} \mathcal{L}_j(h_{A,j}(\{y_i^l\}_{f_i \in A}), y_j^l). \tag{5.3}$$

模型链接和集成模型均通过梯度下降（Gradient Decent）方法进行优化。请注意，如果 A 只有一个元素 f_i，则集成模型简单地作为一个恒等层进行拟合，$h_{A,j} = g_{ij} \circ f_i$。



(a) 车辆计数与人员计数结果分布　　**(b) 交通状况分类与车辆计数结果分布**

图 5.3　不同时间段之间的分布偏移现象

5.3　模型链接自适应和聚合

本节介绍用于模型链接在线自适应训练的设计。之后讨论如何利用模型链接进行领域自适应，并提出了一种聚合跨领域模型链接的方法。

5.3.1　在线自适应训练

通常，机器学习算法侧重于输入的普遍分布。相反，实际输入（例如，摄像机拍摄的视频流）具有更狭窄的分布，并且会实时变化[146,229]。在一个视频分析系统中（详见第 5.5.1节的详细设置），本文部署了三个模型（车辆计数、人员计数和交通状况分类），分析了模型推理结果随时间变化的分布偏移。如图 5.3a所示，在相同的车辆计数下，四个时间段之间的人物计数结果存在很大差异。注意，图例中的 x-yk 表示从第 x 千帧到第 y 千帧的视频片段。类似的差异也存在于交通状况分类和车辆计数模型之间（参见图 5.3b）。因此，使用初始输出来训练模型链接的基本方法在在线服务时会导致性能严重下降。而收集具有足够普遍分布的样本后再进行训练则会引发冷启动问题。因此，本文研究了用于模型链接的在线训练方法。

周期性更新。适应输入动态性的一种简单方法是定期收集样本来更新模型链接，例如，每十分钟选择 50 帧运行所有机器学习模型，并使用额外得到的这 50 个样本增量式更新训练模型链接。本文的实验证明，这种方法是有效的，但缺乏适应性：在不需要更新的时期会浪费过多计算资源来收集数据。因此，本文提出了以下主动选择样本进行模型链接更新的自适应方法。

自适应更新。具体而言，本文使用不确定性阈值（Uncertainty Threshold）方法[230] 来决定要标注哪些数据（即运行源模型和目标模型以获得匹配推理结果的一对）。本文利用适用于分类和回归模型的经典熵不确定性度量（置信度由输出方差近似）[231]。除了基于不确定性的主动策略，本文采用了基于损失预测（Loss Prediction）的方法[232]。其关键思想是添加一个神经网络，以中间激活作为输入并预测样本损失。原则上，任何主动采样（Active Sampling）方法都适用

算法 5.1　跨领域聚合

输入: 有 K 个边节点，由索引 k 标记
云端执行:
初始化全局模型链接权重 $\Theta_{G,0}$;
for 每轮 $t = 1, 2, ...$ **do**
  for 每个边节点 k 并行执行 **do**
    $\Delta\Theta_{L,k} \leftarrow \text{LocalUpdate}(k)$;
  end
  $\Theta_{G,t+1} \leftarrow \Theta_{G,t} + \frac{1}{K}\sum_k \Delta\Theta_{L,k}$;
end
边节点执行 LocalUpdate(k):
在本地数据集上计算梯度向量 $\Delta\Theta_{L,k}$;
将 $\Delta\Theta_{L,k}$ 上传至云端;

于模型链接。实验结果表明，在相同的标注成本下，在线自适应方法可以进一步改善周期性方法，因为它能够在更必要的时候进行采样。

5.3.2　领域自适应

机器学习长期以来的一个问题是，在公共数据集上训练的模型在目标应用数据上性能不佳，即领域偏移问题。对于领域自适应，现有工作已经全面研究了无监督[233]、半监督[234] 和有监督[223] 的各种方法。在有真实监督的情况下，本文提出的模型链接还可以用于将部署的黑盒模型适应到目标领域。

链接到抽象模型节点。为了实现领域自适应功能，在本文的模型链接形式化下，需要首先抽象出一个代表真实标签生成器（通常是人类标注者）的理想模型节点。接下来，可以构建并训练从部署模型（源领域）到抽象模型（目标领域）的模型链接。然后，可以执行原始模型和训练后的模型链接进行服务。由于本文设计模型链接的架构是简单的，一小部分训练样本就足以获得有效的模型链接。这种方法能够起作用的关键在于：与通常关注理解一般分布的通用机器学习不同，实际应用中的服务模型只需适应一个更狭窄的分布。

跨领域聚合。考虑一个常见情况，云服务器持有机器学习模型并将其部署到许多边缘设备，用于分析本地数据流。那么对于本文提出的模型链接，一个有趣的问题是：如何聚合在这些边缘设备上本地训练的模型链接？训练模型链接避免了大多数隐私问题，因为它只需要机器学习模型的推理输出，而不需要原始传感数据。因此，可以通过要求边缘设备将本地推理结果发送到云服务器来训练全局模型链接。然而，尽管推理结果通常比原始数据更不敏感，但在某些高度敏感隐私的场景中，例如医院中的医学图像[235]，不允许传输本地分析结果。遵循联邦学习（Federated Learning）的思想[236]，该思想旨在使用分布式和私有数据训练全局模型，本文提出通过在训练期间聚合相应的本地模型链接的梯度来训练全局模型链接。具体而言，设 g^G 是具有参数 Θ_G 的全局模型链接，该链接在本地

领域（例如，边缘设备）之间共享。初始阶段，云服务器将 g^G 发送到每个边缘设备。在每一轮中，每个边缘设备训练本地模型链接 g^L，并将使用本地训练样本计算的梯度发送到云端。然后，服务器通过对收集到的梯度进行平均来聚合本地变化，并更新 Θ_G。算法 5.1 显示了在边缘和云端执行的详细步骤。除了更好的初始化外，全局链接还可以通过使用集成方法聚合本地和全局链接，从而进一步提高领域适应性[237]。

5.4　多模型协同推理

本节提出了一种基于模型链接的算法，以在成本预算下部署多模型推理。设 $\mathcal{F}(A)$ 表示平均输出精度，即

$$\mathcal{F}(A) = \frac{1}{|F|}(\sum_{f_i \in A} 1 + \sum_{f_j \in F \setminus A} p(A, f_j)). \tag{5.4}$$

然后定义激活一个模型 f_i 的增益为 $\Delta(A, f_i) = \mathcal{F}(A \cup \{f_i\}) - \mathcal{F}(A)$。假设将一个模型链接源添加到集成模型中不会降低性能：$p(A \cup \{f_i\}, f_j) \geqslant p(A, f_j)$，这在经验上通常是成立的[238]。因此 $\Delta(A, f_i) \geqslant 0$，即目标函数是非减的。至于子模性（Submodular），给定 $A_1 \subset A_2 \subset F, f_i \notin A_2$，定义 $A_1' = A_1 \cup \{f_i\}, A_2' = A_2 \cup \{f_i\}$。然后有：

$$\begin{aligned}
\Delta(A_2, f_i) - \Delta(A_1, f_i) = \frac{1}{|F|}\{&(p(A_1, j) - p(A_2, j)) \\
&+ \sum_{f_j \in A_2 \setminus A_1, j \neq i} (p(A_1, f_j) - p(A_1', f_j)) \\
&+ \sum_{f_j \in F \setminus A_2, j \neq i} [\underbrace{(p(A_2', f_j) - p(A_2, f_j))}_{A_2 \text{中} f_j \text{的增益}} \\
&- \underbrace{(p(A_1', f_j) - p(A_1, f_j))}_{A_1 \text{中} f_j \text{的增益}}]\}.
\end{aligned}$$

显然，如果将 f_j 添加到 A_2, A_1 中的边际增益递减，那么 $\Delta(A_2, f_i) - \Delta(A_1, f_i) \leqslant 0$，即目标函数是子模的。但这个性质并不总是成立。实验证明，存在两种典型情况：(1) 主导情况，即集成模型的性能大致等于最佳性能的模型链接源。设 $f_{i^*} = argmax_{f_i \in A} p(g_{ij})$ 表示性能最佳的源模型链接。实验观察到 $p(h_{A,f_j}) \approx p(g_{i^*j})$，即最佳源主导了整体性能。(2) 互助情况，即多源模型链接集成优于任何单一源。$\forall f_i \in A, p(h_{A,f_j}) > p(g_{ij})$，模型链接源相互协助。在这种情况下，如果 f_j 在 $A_2 \setminus A_1$ 中的模型协作更好，则 f_j 对 A_2 的增益可能大于其对 $A_1, A_1 \subset A_2$ 的增益。因此，目标函数并不恒具有子模性。

算法 5.2　多模型协同推理

输入: 模型集合 F, 成本预算 B
输出: 推理结果 y_i

对于每个 $f_i, f_j \in F, i \neq j$，训练模型链接 g_{ij};
对于每个 $f_j \in F$，训练集成模型 $h_{A_j,j}$，其中 $A_j = F \setminus \{f_j\}$;
for 每个周期 **do**
    根据公式 5.7为每个 $f_i \in F$ 估算激活概率 $\mathcal{P}_i$;
    贪心地选择 $A \leftarrow A \cup \{argmax_{f_i \in F \setminus A}(\mathcal{P}_i)\}$ 直到达到成本预算 B;
    输入 x 到达;
    for $f_i \in F$ **do**
        if $f_i \in A$ **then**
            $y_i \leftarrow f_i(x)$;
        else
            $y_i \leftarrow h_{A,i}(\{y_j\}_{f_j \in A})$;
        end
    end
end

激活概率。求解方程 (5.1) 是 NP-困难的，现有的具有 $(1 - e^{-1})$ 最优近似比的算法[239] 需要部分枚举，并需要 $O(n^5)$ 次计算目标函数。注意，这一优化问题求解不是一次性的过程，需要在线执行以适应推理系统的动态性。因此，本文设计了一种称为“激活概率”的启发式度量，其计算仅依赖于模型链接的性能，而不是集成模型的性能。给定一个模型 f_i，激活概率考虑三个因素：(1) 从 f_i 到所有其他模型的模型链接的平均性能，表示为:

$$\mathcal{P}_i^1 = \frac{\sum_{j \neq i} p(g_{ij})}{|F| - 1}; \tag{5.5}$$

(2) 从所有其他模型到目标为 f_i 的模型链接的平均性能，表示为:

$$\mathcal{P}_i^2 = \frac{\sum_{j \neq i} p(g_{ji})}{|F| - 1}; \tag{5.6}$$

(3) f_i 的成本，即 $c(f_i)$。然后，激活概率设计如下:

$$\mathcal{P}_i = \frac{1 + \mathcal{P}_i^1 - \mathcal{P}_i^2}{wc(f_i)}, \tag{5.7}$$

其中权重 w 可以通过以下归一化确定。通过将范围规范化为 0 到 1，有 $(1 + 1 - 0)/(w \min_i(c(f_i)) = 1$，即 $w = 2/\min_i(c(f_i))$。这个激活概率可以被视为一个与选择模型时目标函数的增益正相关的系数。

周期性重新选择。由于内容动态性，激活模型的最优子集可能随时间变化。但是适应这样的动态性会带来在内存中加载和卸载模型的额外开销。因此，本文提出定期重新选择激活的模型。在每个周期开始时，使用一小部分数据（例如 1%）对模型链接的预测性能进行分析。然后，更新模型的激活概率，并重新选

表 5.1　Hollywood2 数据集上使用的模型总结

任务类别	模型	输入模态	输出格式	指标
单标签分类	性别分类（Gender）[240]	音频	2-D 软标签	精度
多标签分类	动作分类（Action）[241]	视频	12-D 软标签	mAP
定位	人脸检测（Face）[242] 人物检测（Person）[221]	图像 图像	4-D 边界框	IoU
回归	年龄预测（Age）[243]	图像	1-D 标量	MAE
序列生成	图像描述（Caption）[155] 语音识别（Speech）[244]	图像 音频	可变长度文本	WER

择在当前周期加载的模型。通过合理设置周期长度和用于分析的数据比例，可以将加载/卸载模型的开销摊销到可以忽略不计。

算法 5.2 显示了如何利用 MLink 实现多模型协同推理。初始阶段，训练成对的模型链接和集成模型。在每个周期中，首先通过对模型链接的数据进行分析来计算激活概率。然后 MLink 在成本预算下基于激活概率贪心地选择要在当前周期加载的模型。在服务阶段，激活的模型进行精确推理，而其他模型的输出将由激活源的模型链接集成进行预测。

5.5　验证实验

5.5.1　实验设置

实现。本文基于 TensorFlow 2.0[106] 在 Python 中实现了 MLink 设计，作为推理系统的可插拔中间件。本文在使用 TensorFlow[106]、PyTorch[245] 和 MindSpore[107] 实现的程序上进行了集成测试，只需修改几十行代码，这显示了 MLink 的易用性。

多模态数据集和模型。实验使用了 Hollywood2 视频数据集[246]。为了获得多模态模型的对齐输入，实验选择了每个视频的第 30 帧并提取了音频数据。实验部署了七个预训练的模型，涵盖了单标签和多标签分类、目标定位、回归和序列生成等五类学习任务。它们具有不同的模型架构、输入模态和输出格式。为了验证模型链接的性能，实验使用了任务特定的度量标准，包括精度、平均精度（mAP）、边界框的交并比（IoU）、平均绝对误差（MAE）和词错误率（WER）。表 5.1 总结了这些模型的信息。

智慧建筑和城市交通监控系统。除了公开数据集，本文还在两个真实世界的视频分析系统上验证了 MLink。(1) 智慧建筑。为了支持包括自动空调和照明控制、异常事件监控等应用，实验部署了三个模型：基于 OpenPose[5] 的人数计数、基于 ResNet50[222] 的动作分类[247]，以及基于 YOLOV3[221] 的物体计数。本

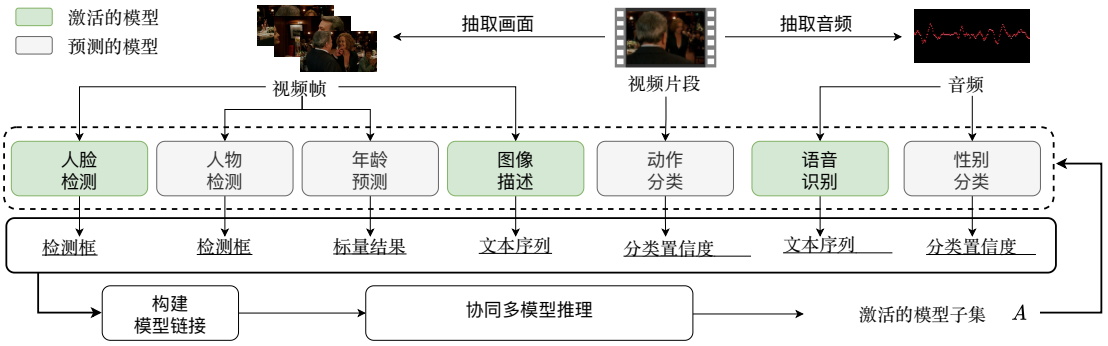


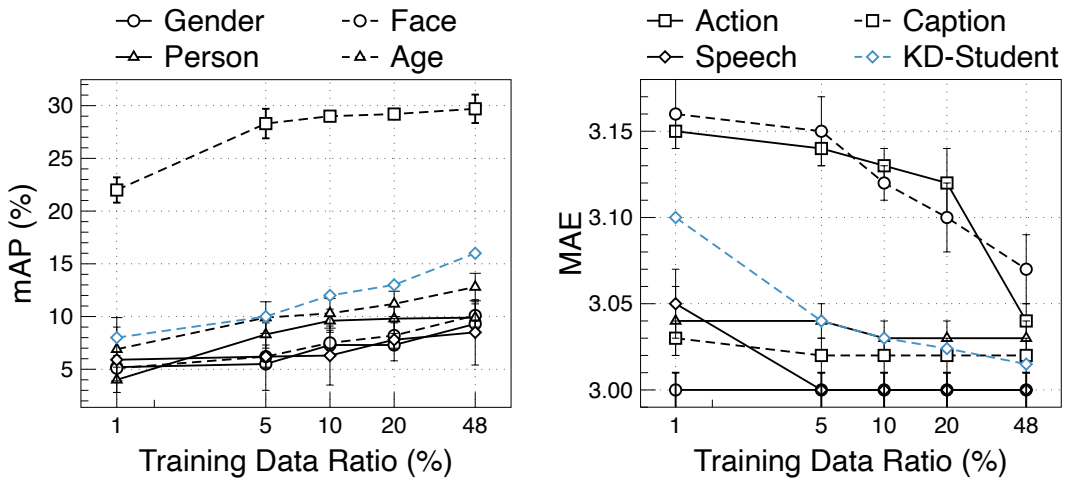
图 5.4　Hollywood2 数据集上的多任务推理工作流

工作从所有 58 个摄像头收集了两天（一个工作日和一个周末）的视频帧（以每分钟 1 帧的频率）。本工作使用一台搭载 NVIDIA 2080Ti GPU 的边缘服务器进行测试。(2) 交通监控。在一个拥有超过 20,000 个摄像头的城市规模视频分析平台上，为了进行交通监控，部署了三个模型：基于 OpenPose[5] 的人数计数、基于 ResNet50[222] 的交通状况分类[248]，以及基于 YOLOV3[221] 的车辆计数。本工作选择了 10 个路口的摄像头，并收集了两天（一个工作日和一个周末）的视频帧（以每秒 1 帧的频率）。本工作使用了五台服务器，每台搭载四个 NVIDIA T4 GPU。

基线方法。在实验中，本工作使用独立推理和三种强大的权衡资源性能的现有方法作为基线。(1) Standalone: 独立运行各个模型。(2) MTL：采用了一个多任务学习架构[73]，其中包含一个全局特征提取器，所有任务共享，并且有任务特定的输出分支。使用 ResNet50[222] 实现特征提取器，使用全连接层进行任务特定输出。使用在 ImageNet[249] 上预训练的 ResNet50 特征提取器的权重初始化，并在智慧建筑/交通监控测试平台上为人数计数、动作/交通分类和物体/车辆计数任务连接三个输出分支。MTL 模型在相应模型的准确推理结果的监督下进行训练。(3) Reducto[58]：一种视频帧过滤方法。对于每个模型，Reducto 首先计算相邻帧的特征差异。如果特征差异低于阈值，它会过滤掉当前帧并重用最新的推理输出。实验中测试了 Reducto 中提出的四种低级特征类型，并选择了性能最好的一种来报告结果。(4) DRLS[74]：一种基于深度强化学习的多模型调度方法。DRLS 训练一个深度强化学习代理，根据已执行模型输出的观察，预测在给定数据上应该执行的下一个模型。

5.5.2　黑盒模型链接

对训练数据大小的敏感性。 Hollywood2 数据集中的原始训练集和测试集包含分别包含 823 个视频片段（约 48%）和 884 个视频片段。为了测试模型链接在不同大小的训练数据下的性能，实验进一步随机抽样了四个训练数据子集，其比例分别为总数据集的 1%，5%，10%，20%。实验使用 RMSprop[250] 优化器和相同



(a) 目标：动作分类　**(b) 目标：年龄预测**

(c) 目标：人物检测　**(d) 目标：性别分类**

图 5.5　不同源模型在四个目标上的模型链接性能

的超参数（学习速率 0.01，训练 100 个 epochs，批大小 32）训练成对的模型链接。作为比较，在一些目标模型（动作、年龄、性别）上，实验采用知识蒸馏[206]方法，其中学生模型有两个卷积层。重复实验三次，并报告性能的平均值和标准差。如图 5.5所示，其中 KD-Student 图例指通过知识蒸馏在目标模型上训练的学生模型，在使用所有训练数据的情况下，*Caption*-to-*Action* 模型链接可以达到 31.7% 的 mAP。而两个检测模型之间的模型链接，*Face*-to-*Person* 和 *Person*-to-*Face* 模型链接，分别达到 59% 和 32% 的 IoU。即使在非常有限的训练样本（1%）下，一些模型链接也能够实现高性能。*Face*-to-*Gender* 的模型链接实现了 92.1% 的准确率。对于 *Gender*-to-*Age* 模型链接，实现了 3.0 的 MAE。与通过知识蒸馏在目标模型上训练的学生模型相比，模型链接实现了更高的预测性能，尤其是当训练数据量较小时。但是对于语音识别和视频字幕模型，实验表明，无法有效地构建针对它们的模型链接。

模型链接集成。对于一个目标模型，已经从不同的源模型构建了多个模型链接。然后，实验使用与模型链接相同的优化器和相同的超参数，使用所有源训

表 5.2　人物检测模型链接的 IoU 分数和 Pearson 相关性

源模型	动作分类	年龄预测	人脸检测	性别分类
IoU (%)	39.4 (±0.1)	38.9 (±0.1)	**58.5** (±1.3)	39.0 (±0.1)
Pearson 相关性	0.123	0.042	**0.244**	-0.053

表 5.3　模型链接集成中的主导和相互帮助案例

目标模型 \ 源模型	动作	年龄	描述	人脸	性别	人物	语音	集成
动作 mAP(%)	-	12.8	**29.7**	10.1	9.3	9.9	8.5	30.8
面部 IoU(%)	11	11.2	0	-	10.3	**31.9**	0	32.2
人物 IoU(%)	39.4	38.9	0	**58.5**	39.0	-	0	59.2
年龄 MAE	3.04	-	3.02	3.07	3.0	3.03	3.0	2.98
性别准确率 (%)	92	92.1	92	92.1	-	92	92	92.3

练集训练了集成模型。表 5.3显示了五个目标模型的结果，表中列标题为源模型，行标题为目标模型，主导源的性能以粗体显示，模型链接和集成模型都使用了所有训练样本（48% 比例）进行训练。实验表明，模型链接集成的性能优于每个单一的源模型。实验结果中可以看到有两种典型情况：主导和互助。对于动作、人脸、人物目标模型，描述、人物、人脸源模型分别主导了集成模型链接的性能。但对于年龄和性别目标模型，源模型相互协助，并通过集成实现性能提升。

相关性量化。实验计算了在训练集上不同模型推理输出之间的 Pearson 相关系数。对于单标签和多标签分类模型，实验使用具有最高置信度的索引作为标签。对于定位模型，实验检查边界框是否为空，并将 0 或 1 分配为标签。实验使用回归标量作为标签，并跳过两个序列生成模型。表 5.2显示了针对人物检测模型的模型链接的结果，可以看到模型链接性能与 Pearson 相关系数之间存在正相关关系。

5.5.3　在线 MLink 训练

实验在交通监控应用中验证了提出的模型链接在线训练方法。对于从车辆计数源模型到人数计数目标模型的模型链接，图 5.6显示了不同训练方法在一个摄像头上每小时的片段级别精度，其中视频图像对应于四个红色框，指示着分布发生显著改变的时间点。实验将训练样本的比例设为 1%。离线初始化（Offline Init）方法使用前 1% 的样本来训练模型链接，并在后续数据中不进行更新。实验结果显示, 由于训练样本分布有限, 离线初始化方法在 26 个片段上的精度低于 1%, 平均精度为 6.3%。本文提出的在线周期更新（Online Periodic）方法显著提高了平均精度至 70.2%，在线基于不确定度更新（Online Uncertainty-Based）的方法则带来了额外的 3.3% 提升。本文的基于损失预测的方法（Online Loss Prediction-Based）优于其他方法，实现了 74.7% 的平均精度。

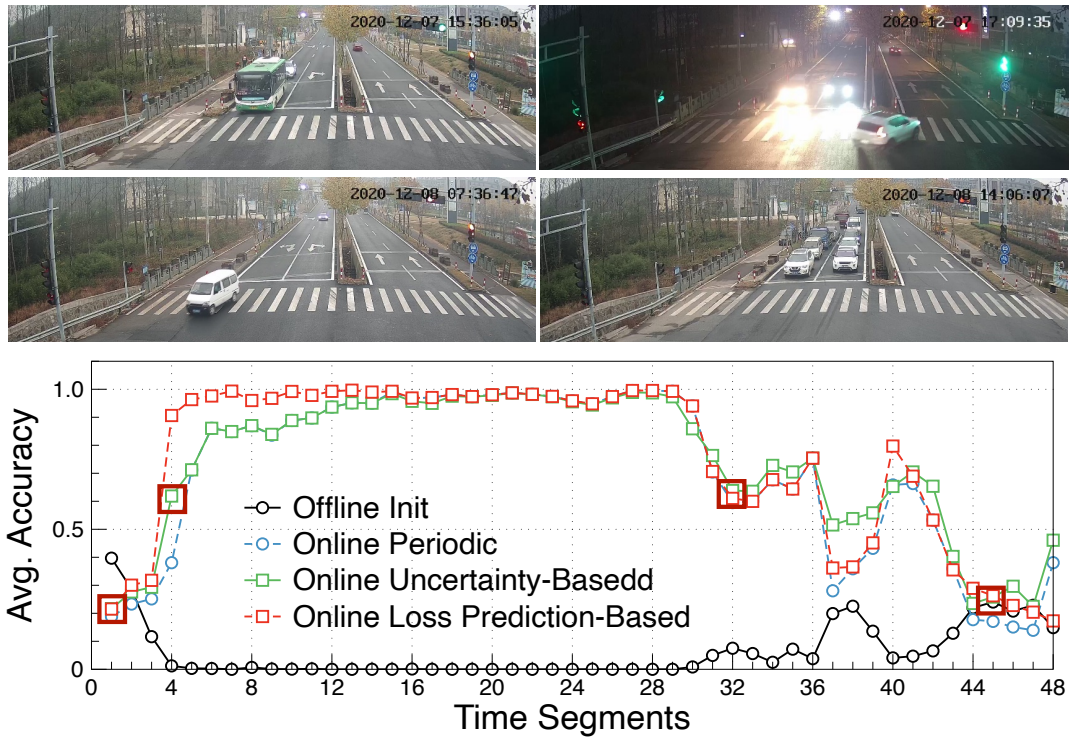


图 5.6　车辆计数源模型到人物计数目标模型的 MLink 在线训练效果

5.5.4　MLink 领域自适应与跨域聚合

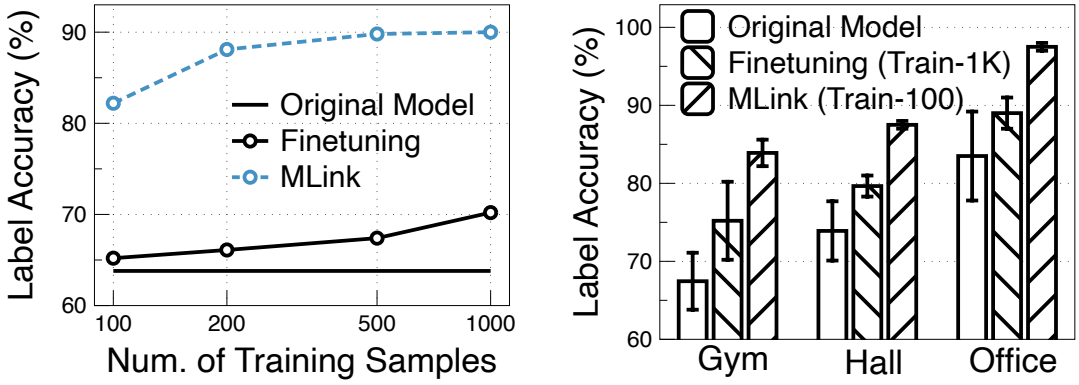
在公共数据集上进行领域自适应。实验使用 Office-Home 数据集[251] 验证模型链接的领域自适应性能。该数据集包含四个领域的图像：艺术 (A)、剪贴艺术 (C)、产品 (P) 和现实世界 (R)。训练和测试集数据包含分别包含 7728 和 7860 张图像。作为基线，实验分别使用每个领域中的所有训练样本训练 ResNet50[222] 分类器。在同一领域上进行测试（方法标记为“相同领域”），这些模型分别达到 87.9%，89.9%，96.7% 和 95.0% 的准确率。当将这些模型应用于不同的领域（方法标记为“偏移领域”）时，由于领域偏移，它们遭受了显著的性能降低，平均准确率严重下降从 92.3% 降至 59.5%。实验使用目标领域训练拆分中的 10% 随机抽样数据，从源领域的 ResNet50 训练模型链接。为了比较，测试了两种领域自适应的最新方法：无监督 SDAT[233] 方法和半监督 CLDA[234] 方法。详细的对比结果显示在表 5.4 中。在无监督 / 半监督方式使用所有目标域图像时，SDAT / CLDA 将平均准确率提高到 72.2% / 75.5%。仅使用目标域训练样本的 10%，并将 ResNet50 视为黑盒，模型链接实现了 72.6% 的平均精度。结果表明，模型链接能够有效减轻领域偏移的影响，提高黑盒模型的自适应性。

在实际应用中的领域自适应。如第 5.5.1 节介绍的，为智慧建筑应用开发的三个模型最初是通过公共数据集进行训练的。其中，当应用于建筑中拍摄的视频时，动作分类器的性能下降最为严重。实验将这 58 个摄像头分为三个真实场景：运动馆（Gym，用 G 代指），大厅（Hall，用 H 代指），和办公室（Office，用 O

表 5.4　Office-Home 数据集上的领域自适应性能比较

方法	A->C	A->P	A->R	C->A	C->P	C->R
相同领域		87.9			89.9	
偏移领域	46.9	65.2	71.8	49.0	60.9	63.5
SDAT[233]	58.2	77.1	82.2	66.3	77.6	76.8
CLDA[234]	63.4	81.4	81.3	70.5	80.9	80.3
MLink	69.2	85.3	80.3	59.8	77.1	71.4

方法	P->A	P->C	P->R	R->A	R->C	R->P	平均
相同领域		96.7			95.0		92.3
偏移领域	53.6	44.1	73.3	61.9	47.9	76.1	59.5
SDAT[233]	63.3	57.0	82.2	74.9	64.7	86.0	72.2
CLDA[234]	72.4	63.9	82.2	76.7	66.0	87.6	75.5
MLink	62.9	64.2	80.1	69.7	66.1	86.1	72.6

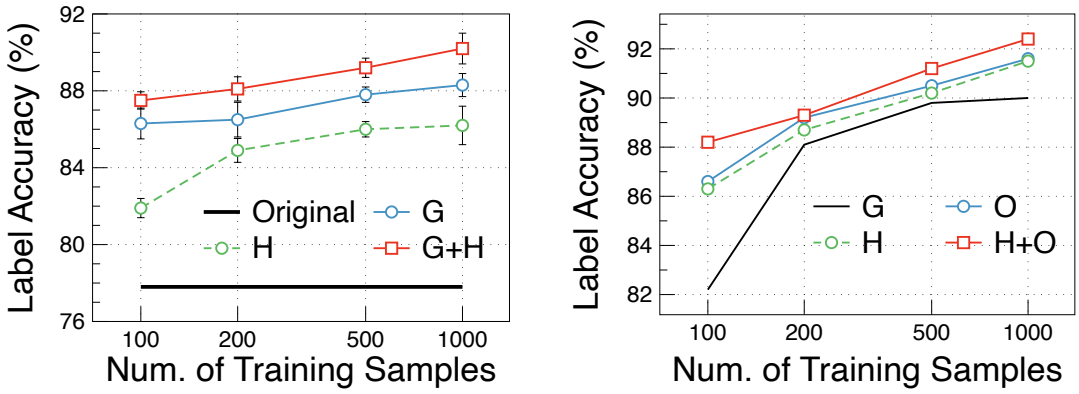


(a) Gym 场景　　**(b) 智慧建筑整体**

图 5.7　智慧建筑场景中 MLink 的领域自适应性能

代指），分别有 11、25 和 22 个摄像头。实验收集了每个场景中的 2000 张图像，并手动标记了人物动作标签。标记的数据被分为训练和测试子集，每个子集包含 1000 张图像。预先训练的动作分类器在运动馆、大厅和办公室场景中的测试准确率分别仅为 67.5%，73.9% 和 83.5%。使用预先训练的动作分类器的输出和真实标签，实验使用不同数量的样本（100、200、500、1000）训练模型链接。作为比较，实验采用分类器微调方法[252]，冻结预训练模型中的特征提取器的参数并重新训练分类器。图 5.7a 绘制了在运动馆领域上进行的精度测试，其中微调的分类器达到了 66.1% 的精度，而模型链接在仅使用 200 个训练样本时以 88.1% 的精度显著优于它。如图 5.7b 所示，使用 90% 更少的训练样本，模型链接仍然将标签精度提高了最多 20.7%，并且在预先训练模型的平均精度改进上至少优于微调的分类器 7.85%。原因在于微调在高维特征空间中工作，这是如此复杂，以至于有限的训练样本无法调整它以适应目标领域。

跨领域聚合。实验使用三个智慧建筑场景来验证通过聚合跨领域本地模型



(a) 在办公室领域上的模型链接聚合　　**(b) 在运动馆领域上的本地-全局聚合**

图 5.8　在智慧建筑场景中 MLink 聚合性能

链接而训练的全局模型链接。以一个场景作为目标域，将其他两个作为源域，实验将在源域上训练的局部模型链接聚合成全局模型链接。对于动作分类，图 5.8a 绘制了在办公室场景中使用不同数量的样本进行训练时，局部和全局模型链接的标签精度。直接将在运动馆（或大厅）场景中训练的局部模型链接应用于办公室场景可以将原始模型的精度从 77.8% 提高到 86.2%（或 88.3%）。通过应用从运动馆和大厅聚合的全局模型链接（表示为 G+H），标签精度达到 90.2%。平均而言，其他两个领域的全局模型链接为目标领域的精度带来了 7.85% 的提升，而无需使用目标领域中的任何样本。

本地-全局聚合。实验验证了在智慧建筑系统中聚合本地和全局模型链接的效果。对于动作分类模型，图 5.8b 显示了在运动馆领域上进行的精度测试，从实验结果中可以看到聚合局部模型链接（标记为 G）和其他领域（标记为 O，H，O+H）将精度提高了高达 6%，比仅使用运动馆领域的数据更准确。对于所有三个领域的整体结果，平均而言，聚合跨领域模型链接将精度额外提高了 1.1%。

跨任务聚合。然后，实验从另外两个模型，物体计数（Object）和人员计数（Person）模型，训练了到动作分类器（Action）的模型链接。并测试了从跨任务模型获取的这些链接聚合的影响。图 5.9 绘制了在智慧建筑应用中使用不同数量的训练样本训练的带有聚合权重的模型链接的标签精度。从多个来源聚合的模型链接（由多个以 + 组合的源进行标记）明显优于单一来源的模型链接。与动作模型链接相比，聚合来自三个跨任务模型的链接（标记为 Action+Object+Person）可以将精度提高多达 4.1%。

5.5.5　视频分析任务

实验在智慧建筑系统的 58 个摄像头的 48 小时视频和城市交通监控平台的 10 个摄像头的 48 小时视频上测试了 MLink。实验利用前 10% 的数据来训练模

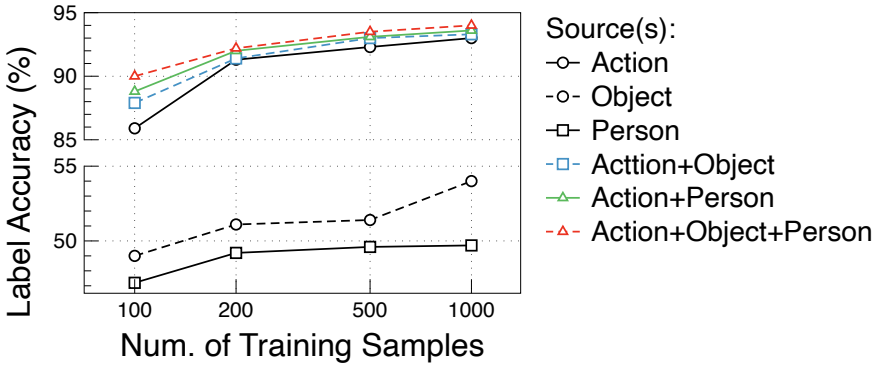
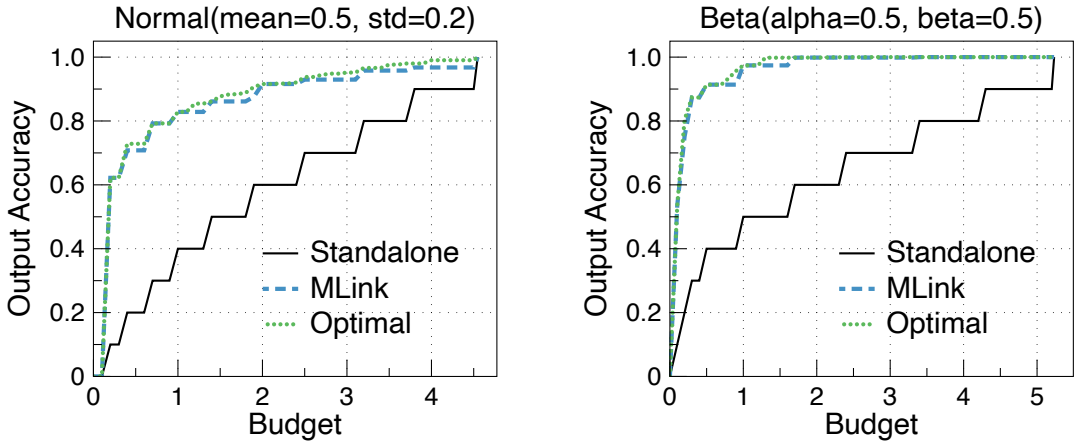


图 5.9　在智慧建筑应用中以动作分类为目标任务的跨任务聚合效果



(a) 正态分布　　**(b) Beta 分布**

图 5.10　模拟 MLink 调度 10 个模型的效果

型链接和集成模型。实验将周期长度设置为一小时，并使用初始 1% 的数据来对激活概率进行分析。对于计数模型，输出精度是通过检查预测数量的绝对误差是否小于 0.5 来计算的。每个模型的时间成本是通过训练数据进行离线分析得出的平均推理时间。在智慧建筑系统中，动作/人物/物体模型每帧耗时 30/44/60 毫秒。在交通监控系统中，交通/人物/车辆模型每帧耗时 55/66/70 毫秒。每个模型的 GPU 内存成本是峰值使用情况：人数计数为 4.6 GB，动作/交通分类为 1.5 GB，物体/车辆计数为 3.7 GB。实验将预算 B 设置为分配给模型的最大 GPU 内存，以验证 MLink 如何提高多模型推理的资源效率。实验中平等对待每个模型的输出精度，并报告它们的平均输出精度。在 GPU 内存预算下，基线独立运行（Standalone）简单地选择具有最小平均时间成本的模型。重复了三次调度实验，并在表5.5中报告了结果。由于标准差较小（< 0.1），为简单起见而没有呈现它们。在两个系统中，MLink 在输出精度上优于其他替代方案。与独立运行相比，在智慧建筑系统中，MLink 节省了 66.7% 的推理执行开销，同时保持了 94.1% 的输出精度。

可扩展性。本文进行了模拟实验来测试当模型数量较大时 MLink 的调度性

表 5.5　视频分析系统上 MLink 与基线方法的比较

方法	智慧建筑（5/9GB 内存）		智慧城市（5/9GB 内存）	
	精度（%）	时间（毫秒）	精度（%）	时间（毫秒）
Standalone	33.3/66.7	30/74	33.3/66.7	55/121
MTL	53.3	32.8	61.3	32.5
DRLS	45.7/81.3	58.7/107	39.5/77.6	102/188
Reducto	91.8/96.9	45.7/89	84.1/95.3	64/127
MLink	**94.1/97.9**	39.3/84	**94/97.4**	62/125

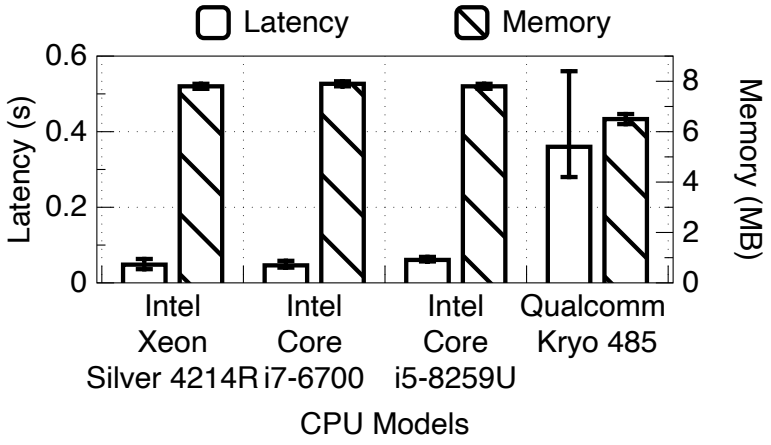


图 5.11　MLink 在服务器和手机上的延迟和内存开销

能。图5.10显示了 Standalone、MLink 和最优调度结果在两个不同分布模拟情况下的比较。为了模拟模型链接的正常和相对极端的情况，实验分别使用正态分布（均值 0.5，标准差 0.2）和 Beta 分布（alpha 和 beta 均为 0.5）生成了 10 个模型之间的模型链接性能和它们的成本。实验将集成增益固定为 0.02。最优调度和 Standalone 调度都是通过暴力枚举找到的。实验结果表明，MLink 实现了接近最优的调度结果，并显著优于独立运行的基线方法。

MLink 的开销。模型链接的训练和推理操作具有很好的并发性和效率。实验测试了并发模型链接训练和推理，输入和输出长度随机设置为 1 到 100。每个进程的平均训练时间随着超过 20 个并发训练进程而减少到不到一秒。并给定 100 个并发处理器和一百万个样本进行推理，整体延迟仅约为一分钟。实验将 MLink 部署在四种不同的设备上（云服务器、边缘服务器、笔记本电脑和手机），并测试其延迟和内存占用。如图 5.11所示，MLink 仅引入可忽略的额外开销。实验还测试了聚合模型链接的通信开销。对于 3 个客户端的情况，总体通信成本为 88*3（服务器广播）+ 88*3（客户端更新）= 每个样本 528 字节。实验将批量大小设置为 32，这相当于每个训练步骤的通信成本为 16.5 KB。

5.6　小结

本章首先揭示了模型节点孤立存在的现状，以及该现状对于模型部署带来的技术挑战，进而提出利用语义关联性构建模型网络的思想。紧接着，本章对模型链接任务进行了形式化，提出了支持异构黑盒机器学习模型的模型链接设计。针对跨边缘场景中模型链接的适应性和聚合性，本章提出了涵盖在线动态和跨领域分布偏移两个方面的方法。为优化边云多模型推理的部署任务，本章开发了一种名为 MLink 的基于模型链接的算法，在有限成本预算下进行优化。在一个包含七个不同机器学习模型的多模态数据集上验证了 MLink 的设计，涵盖了五类学习任务和三种输入模态。结果表明，本章提出的模型链接有效地在异构黑盒模型之间建立了联系。本章还在两个真实的视频分析系统上验证了 MLink，一个用于智能建筑，另一个用于城市交通监控，包括六个视觉模型和来自 58 台摄像机的 3264 小时视频。实验结果显示，相较于原始的离线训练，提出的在线自适应训练方法有效地提高了 MLink 的性能，并且提出的聚合方法实现了比原始模型更高的平均准确度。在 GPU 内存预算下，MLink 可以显著节省推理计算，同时保持较高输出准确度。MLink 的技术局限性和未来的工作总结如下：(1) 当源模型和目标模型之间的语义相关性较低时，模型链接的输出精度较差。(2) 当加入的模型数量非常大时，两两模型链接将变得不切实际。因此，在未来将研究如何智能地选择模型来建立模型链接，而非完整地两两模型直接构建链接。

第 6 章　总结和展望

本文关注智能模型推理任务从部署到运行时全周期中所遇到的技术挑战，以“异构设备协同推理”为核心思想，针对推理过程的资源效率和安全性进行了研究和探索。本章对全文的工作进行总结，并展望未来可能的研究方向。

6.1　论文总结

本文针对智能模型推理任务从部署到运行时全周期中所遇到的四个技术问题，即数据源并发度受限、通信计算开销过高、数据安全风险严重、模型部署效率低下，以“异构设备协同推理”为核心思想，探索了（1）多端协同的并发包门控，（2）端边协同的输入过滤，（3）端云协同的安全推理协议，（4）边云协同的自适应模型部署四个方向。具体贡献总结如下：

• **定位视频推理并发瓶颈，设计多端协同并发包门控方法**。第2章首先描述了如何在实际系统中识别视频推理流程中被忽视的并发度瓶颈。为了提高视频分析的端到端并发度，本文提出了一种新的方法：视频包门控，这种方法补充了现有优化方法在解码效率方面的不足。本文引入了一个名为 PacketGame 的框架，用于多流视频包门控，它利用轻量级的时序估计器和上下文预测器来自适应地表示视频包。此外，PacketGame 使用一个组合优化器进行跨视频流的资源协调，理论分析证明了该组合算法具有最优近似比。本文证明了 PacketGame 的整体性能具有在线遗憾上界，并在一个包含 1108 个摄像头的真实系统以及公共视频上进行了四个推理任务的验证。实验结果表明，与原始推理方法相比，PacketGame 可以节省 52.0-79.3% 的解码成本，并实现 2.1-4.8 倍的并发性。与四种最先进的互补方法[54,58,63,72]的比较表明，在端到端并发性和广泛适用性方面，PacketGame 具有显著的优势。相关代码已在https://github.com/yuanmu97/PacketGame开源。

• **建立可过滤性理论框架，设计支持全模态输入过滤方法**。第3章首先对输入过滤问题进行了形式化，并给出了过滤器的有效性条件。基于推理任务和输入过滤器的假设族之间的复杂性比较，本文对推理任务的“可过滤性”进行了定义和分析，旨在指导和解释输入过滤技术的应用。本文提出了一个端到端可学习的输入过滤框架，统一了跳过和重用方法。由于端到端可学习性，提出的框架具有鲁棒辨别力的特征嵌入，支持各种输入模态和推理部署方式。本文设计并实现了名为 InFi 的输入过滤系统。对包括 8 种输入模态、14 个推理任务进行全面验证表明，InFi 具有更广泛的适用性，并在准确性和效率方面优于基线方法。针对移动平台上的视频分析应用，相对于原始的车辆计数任务，InFi 实现了高达 8.5 倍的

吞吐量提升，并节省了 95% 的带宽，同时保持超过 90% 的准确性。相关代码已在https://github.com/yuanmu97/infi开源。

• 提出创新的三方威胁模型，设计精度无损的高效安全推理协议。第4章首先定位出针对 Transformer 模型安全推理协议的两方设置（模型所有者和数据所有者）中存在的固有效率瓶颈，以及这些瓶颈与实际应用的不一致之处。与传统的同态加密和安全两方计算框架不同，本文提出了一个新的三方威胁模型，将模型所有者分解为两个不同的实体：模型开发者和模型服务器。基于这一模型，本文提出了第一个用于三方 Transformer 推理的安全协议 STIP，并证明了其具有隐私泄漏的理论界限和精度无损的保证。本文实现了 STIP 并在实际系统中验证了各种 Transformer 模型（其中最大的具有多达 700 亿参数），覆盖了文本、图像以及文本-图像多模态输入。实验结果表明，STIP 的效率可与未保护的全云推理相媲美，超过了最先进的安全两方协议[40-42]数百万倍。相关代码已在https://github.com/yuanmu97/secure-transformer-inference开源。

• 构建黑盒模型语义链接，将孤立模型节点整合为互联模型网络。第5章首先揭示了模型节点孤立存在的现状，提出利用语义关联性构建模型网络的思想。紧接着，本文对模型链接任务进行了形式化，提出了支持异构黑盒机器学习模型的模型链接设计。针对模型链接的适应性和聚合性，本文提出了涵盖在线动态和跨领域分布偏移两个方面的方法。为优化多模型推理的部署任务，本文开发了一种名为 MLink 的基于模型链接的算法，在有限成本预算下进行优化。在一个包含七个不同机器学习模型的多模态数据集上验证了 MLink 的设计，涵盖了五类学习任务和三种输入模态。结果表明，本文提出的模型链接有效地在异构黑盒模型之间建立了联系。本文还在两个真实的视频分析系统上验证了 MLink，一个用于智能建筑，另一个用于城市交通监控，包括六个视觉模型和来自 58 台摄像机的 3264 小时视频。实验结果显示，相较于原始的离线训练，提出的在线自适应训练方法有效地提高了 MLink 的性能，并且提出的聚合方法实现了比原始模型高 7.9% 的平均准确度。在 GPU 内存预算下，MLink 显著优于多个基线方法，可以节省 66.7% 的推理计算，同时保持 94% 的输出准确度。相关代码已在https://github.com/yuanmu97/MLink开源。

6.2　未来展望

（1）数据传感器可扩展性。在智能物联网环境中，传感器可扩展性指推理系统能够有效地增加接入的传感器数量的能力。随着物联网的快速发展，连接设备和传感器的数量不断增加，导致产生的数据量呈指数级增长，这给数据处理和存储带来了巨大压力，以及集成异构传感器的技术挑战。为了有效地监测、分析和

应用这些数据，系统必须具备支持更多传感器的能力。本文的第2章探讨了一种策略，即通过筛选解码前的视频数据包来提高视频分析系统对摄像头数量的扩展性。然而，物联网应用范围广泛，涵盖智能城市、工业自动化、健康监测和农业等领域。不同的应用场景对传感器类型和数量的需求各不相同，因此在未来需要研究在多样的传感器和分析任务下的可扩展性问题。例如，针对音视频、无线信号等传感器数据，可以利用其天然的时序关联性来进行自适应数据过滤。同时，通过利用推理任务在特定物联网数据领域中的分布特性，相较于广泛的训练数据领域，可以对数据进行更大幅度的压缩。

（2）异构协同的神经网络拆分推理方法。在异构协同的神经网络拆分推理方法中，面临着诸多挑战和机遇。一个重要的问题是端侧设备的算力异构性，这意味着不同移动设备的内存和 CPU 性能存在显著差异，导致静态的端云协同部署方案无法适用于所有端设备。举例来说，针对广告推荐模型的端云部署，如果在算力较弱的设备上执行，可能会导致过高的推理延迟，严重影响用户体验和服务质量。因此，未来的研究可以探索协同推理在应对端侧算力异构性方面的自适应性。具体来说，这需要考虑多种且动态变化的资源优化目标约束，以优化大型神经网络模型在端设备和云端之间的分配和执行方案。例如可以根据端设备的算力情况，将大型神经网络模型拆分成适合端侧和云端处理的子模型。这需要综合考虑模型层级、计算复杂度、内存需求等因素，以最大程度地利用端侧设备的资源，同时保证推理的效率和准确性。一方面，未来需要开发能够动态调整和优化推理部署策略的算法和技术，以应对端侧设备算力变化和不确定性。这可能涉及实时监测端设备的性能、网络负载和用户需求，从而实现最佳的模型拆分和分配方案。另一方面，在端云协同推理中，可能存在多种优化目标，如推理延迟、功耗、带宽利用率等。研究应该探索如何在这些目标之间进行权衡和优化，制定综合考虑各种因素的策略。

（3）异构协同的参数个性化方法。异构协同的参数个性化方法涉及解决端侧数据异构性带来的挑战，以实现更精准的端侧智能服务。为了训练模型并提供个性化服务，需要收集用户相关的多样化数据，包括个人配置、设备使用行为等信息。部署整个模型在端侧并基于用户数据进行训练可能会导致巨大的计算开销和能耗，同时也存在隐私泄露的风险。而将这些数据直接传输到云端会带来严重的隐私问题。未来的研究可以探索权衡隐私保护和资源效率的个性化训练方案。一种可能的方法是在云端部署经过预训练的通用模型参数，以减少端侧计算负担和隐私风险。同时，在端侧根据设备的算力情况自适应地部署和微调个性化参数层，以实现个性化的智能服务。这种参数个性化方法旨在充分利用云端和端侧的优势，通过在不同环境中部署适当的模型部分，实现既保护用户隐私又提供高效的智能服务。

参 考 文 献

[1] DUAN L, LOU Y, WANG S, et al. Ai-oriented large-scale video management for smart city: Technologies, standards, and beyond[J]. IEEE MultiMedia, 2018, 26(2): 8-20.

[2] BARTHÉLEMY J, VERSTAEVEL N, FOREHEAD H, et al. Edge-computing video analytics for real-time traffic monitoring in a smart city[J]. Sensors, 2019, 19(9): 2048.

[3] OPENAI. Chatgpt[EB/OL]. 2022. https://openai.com/blog/chatgpt.

[4] ZHAO W X, ZHOU K, LI J, et al. A survey of large language models[A]. 2023.

[5] Cao Z, Hidalgo Martinez G, Simon T, et al. Openpose: Realtime multi-person 2d pose estimation using part affinity fields[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[6] TUNG L. Microsoft: We're bringing chatgpt to the azure cloud-computing service[EB/OL]. 2023. https://www.zdnet.com/article/microsoft-were-bringing-chatgpt-to-the-azure-openai-cloud-computing-service/.

[7] AZURE. Chatgpt is now available in azure openai service[EB/OL]. 2023. https://azure.microsoft.com/en-us/blog/chatgpt-is-now-available-in-azure-openai-service/.

[8] MAZUMDER A N, MENG J, RASHID H A, et al. A survey on the optimization of neural network accelerators for micro-ai on-device inference[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2021, 11(4): 532-547.

[9] ROMERO F, LI Q, YADWADKAR N J, et al. Infaas: Automated model-less inference serving [C]//2021 USENIX Annual Technical Conference (USENIX ATC 21). 2021: 397-411.

[10] KILOVIEW. Dc230 video/ip camera decoder[EB/OL]. 2022. https://www.kiloview.com/en/decoder/h264-dc230.

[11] NVIDIA. Nvidia video codec sdk[EB/OL]. 2022. https://developer.nvidia.com/nvidia-video-codec-sdk.

[12] REUTERS. Chatgpt sets record for fastest-growing user base - analyst note[EB/OL]. 2023. https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/.

[13] KUCHAIEV O, LI J, NGUYEN H, et al. Nemo: a toolkit for building ai applications using neural modules[A]. 2019.

[14] AMINABADI R Y, RAJBHANDARI S, AWAN A A, et al. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale[C]//SC22: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2022: 1-15.

[15] FORBES. Samsung bans chatgpt among employees after sensitive code leak[EB/OL]. 2023. https://www.forbes.com/sites/siladityaray/2023/05/02/samsung-bans-chatgpt-and-other-chatbots-for-employees-after-sensitive-code-leak/.

[16] DETTMERS T, LEWIS M, BELKADA Y, et al. Llm. int8 (): 8-bit matrix multiplication for transformers at scale[A]. 2022.

[17] CARREIRA S, MARQUES T, RIBEIRO J, et al. Revolutionizing mobile interaction: Enabling a 3 billion parameter gpt llm on mobile[A]. 2023.

[18] DURMAZ INCEL O, BURSA S . On-device deep learning for mobile and wearable sensing applications: A review[J/OL]. IEEE Sensors Journal, 2023, 23(6): 5501-5512. DOI: 10.1109/JSEN.2023.3240854.

[19] XIAO G, LIN J, SEZNEC M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models[C]//International Conference on Machine Learning. PMLR, 2023: 38087-38099.

[20] WANG M, DENG W. Deep visual domain adaptation: A survey[J]. Neurocomputing, 2018, 312: 135-153.

[21] CHENG Y, WANG D, ZHOU P, et al. A survey of model compression and acceleration for deep neural networks[A]. 2017.

[22] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848-6856.

[23] LIN J, CHEN W M, LIN Y, et al. Mcunet: Tiny deep learning on iot devices[J]. Advances in Neural Information Processing Systems, 2020, 33: 11711-11722.

[24] BLALOCK D, GONZALEZ ORTIZ J J, FRANKLE J, et al. What is the state of neural network pruning?[J]. Proceedings of machine learning and systems, 2020, 2: 129-146.

[25] SUN P, ZHANG R, JIANG Y, et al. Sparse r-cnn: End-to-end object detection with learnable proposals[C]//Proceedings of the IEEE/CVF CVPR 2021 Conference. 2021: 14454-14463.

[26] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 11264-11272.

[27] NAGEL M, FOURNARAKIS M, AMJAD R A, et al. A white paper on neural network quantization[A]. 2021.

[28] WENG O. Neural network quantization for efficient inference: A survey[A]. 2021.

[29] JIN Q, YANG L, LIAO Z. Adabits: Neural network quantization with adaptive bit-widths [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2146-2156.

[30] VEPAKOMMA P, GUPTA O, SWEDISH T, et al. Split learning for health: Distributed deep learning without sharing raw patient data[A]. 2018.

[31] KIM J, PARK S, JUNG S, et al. Spatio-temporal split learning[C]//2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). IEEE, 2021: 11-12.

[32] CHEN J, ZHANG A. Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022: 87-96.

[33] AYAD A, RENNER M, SCHMEINK A. Improving the communication and computation efficiency of split learning for iot applications[C]//2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021: 01-06.

[34] XIAO G, LIN J, HAN S. Offsite-tuning: Transfer learning without full model[A]. 2023.

[35] KANG Y, HAUSWALD J, GAO C, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge[J]. ACM SIGARCH Computer Architecture News, 2017, 45(1): 615-629.

[36] LI E, ZENG L, ZHOU Z, et al. Edge ai: On-demand accelerating deep neural network inference via edge computing[J]. IEEE Transactions on Wireless Communications, 2019, 19(1): 447-457.

[37] BANITALEBI-DEHKORDI A, VEDULA N, PEI J, et al. Auto-split: A general framework of collaborative edge-cloud ai[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 2543-2553.

[38] DU W, ATALLAH M J. Secure multi-party computation problems and their applications: a review and open problems[C]//Proceedings of the 2001 workshop on New security paradigms. 2001: 13-22.

[39] ACAR A, AKSU H, ULUAGAC A S, et al. A survey on homomorphic encryption schemes: Theory and implementation[J]. ACM Computing Surveys (Csur), 2018, 51(4): 1-35.

[40] HAO M, LI H, CHEN H, et al. Iron: Private inference on transformers[J]. Advances in Neural Information Processing Systems, 2022, 35: 15718-15731.

[41] CHEN T, BAO H, HUANG S, et al. The-x: Privacy-preserving transformer inference with homomorphic encryption[A]. 2022.

[42] HOU X, LIU J, LI J, et al. Ciphergpt: Secure two-party gpt inference[J]. Cryptology ePrint Archive, 2023.

[43] SABT M, ACHEMLAL M, BOUABDALLAH A. Trusted execution environment: What it is, and what it is not[C]//2015 IEEE Trustcom/BigDataSE/Ispa: Vol. 1. IEEE, 2015: 57-64.

[44] PINTO S, SANTOS N. Demystifying arm trustzone: A comprehensive survey[J]. ACM

computing surveys (CSUR), 2019, 51(6): 1-36.

[45] COSTAN V, DEVADAS S. Intel sgx explained[J]. Cryptology ePrint Archive, 2016.

[46] AMD. Amd sev[EB/OL]. 2024. https://www.amd.com/en/developer/sev.html.

[47] ZHU J, HOU R, WANG X, et al. Enabling rack-scale confidential computing using heterogeneous trusted execution environment[C]//2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020: 1450-1465.

[48] LEE D, KOHLBRENNER D, SHINDE S, et al. Keystone: An open framework for architecting trusted execution environments[C]//Proceedings of the Fifteenth European Conference on Computer Systems. 2020: 1-16.

[49] MO F, SHAMSABADI A S, KATEVAS K, et al. Darknetz: towards model privacy at the edge using trusted execution environments[C]//Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services. 2020: 161-174.

[50] ZHANG X, LI F, ZHANG Z, et al. Enabling execution assurance of federated learning at untrusted participants[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020: 1877-1886.

[51] MO F, HADDADI H, KATEVAS K, et al. Ppfl: privacy-preserving federated learning with trusted execution environments[C]//Proceedings of the 19th annual international conference on mobile systems, applications, and services. 2021: 94-108.

[52] GE Z, LIU S, WANG F, et al. Yolox: Exceeding yolo series in 2021[A]. 2021.

[53] BAIDU. Paddledetection, object detection and instance segmentation toolkit based on paddlepaddle[EB/OL]. 2019. https://github.com/PaddlePaddle/PaddleDetection.

[54] NVIDIA. Tensorrt[EB/OL]. 2022. https://developer.nvidia.com/tensorrt.

[55] ZHOU D, LI L, GU Q. Neural contextual bandits with ucb-based exploration[C]//Proceedings of the 37th ICML Conference. JMLR, 2020: 11492-11502.

[56] CHEN W, WANG L, ZHAO H, et al. Combinatorial semi-bandit in the non-stationary environment[C]//Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence: Vol. 161. PMLR, 2021: 865-875.

[57] CANEL C, KIM T, ZHOU G, et al. Scaling video analytics on constrained edge nodes[C]// TALWALKAR A, SMITH V, ZAHARIA M. Proceedings of Machine Learning and Systems: Vol. 1. 2019: 406-417.

[58] LI Y, PADMANABHAN A, ZHAO P, et al. Reducto: On-camera filtering for resource-efficient real-time video analytics[C]//Proceedings of the ACM SIGCOMM 2020 Conference. 2020: 359-376.

[59] GUO P, HU B, LI R, et al. Foggycache: Cross-device approximate computation reuse[C]// Proceedings of the 24th Annual International Conference on Mobile Computing and Net-

working. 2018: 19-34.

[60] GUO P, HU W. Potluck: Cross-application approximate deduplication for computation-intensive mobile applications[C]//Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. 2018: 271-284.

[61] CHEN T Y H, RAVINDRANATH L, DENG S, et al. Glimpse: Continuous, real-time object recognition on mobile devices[C]//Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. 2015: 155-168.

[62] KANG D, EMMONS J, ABUZAID F, et al. Noscope: Optimizing neural network queries over video at scale[J]. Proceedings of the VLDB Endowment, 2017, 10(11).

[63] YUAN M, ZHANG L, HE F, et al. Infi: End-to-end learnable input filter for resource-efficient mobile-centric inference[C/OL]//MobiCom '22: Proceedings of the 28th Annual International Conference on Mobile Computing And Networking. New York, NY, USA: Association for Computing Machinery, 2022: 228–241. https://doi.org/10.1145/3495243.3517016.

[64] MOHRI M, ROSTAMIZADEH A, TALWALKAR A. Foundations of machine learning[M]. MIT press, 2018.

[65] KEARNS M J, VAZIRANI U V, VAZIRANI U. An introduction to computational learning theory[M]. MIT press, 1994.

[66] HU E J, SHEN Y, WALLIS P, et al. Lora: Low-rank adaptation of large language models [J/OL]. CoRR, 2021, abs/2106.09685. https://arxiv.org/abs/2106.09685.

[67] TOUVRON H, LAVRIL T, IZACARD G, et al. Llama: Open and efficient foundation language models[A]. 2023.

[68] LIU H, LI C, WU Q, et al. Visual instruction tuning[M]. arXiv:2304.08485, 2023.

[69] SZÉKELY G J, RIZZO M L, BAKIROV N K. Measuring and testing dependence by correlation of distances[Z]. 2007.

[70] SONG C, SHMATIKOV V. Overlearning reveals sensitive attributes[C]//International Conference on Learning Representations. 2020.

[71] TAN C, SUN F, KONG T, et al. A survey on deep transfer learning[C]//International conference on artificial neural networks. Springer, 2018: 270-279.

[72] XIE X, KIM K H. Source compression with bounded dnn perception loss for iot edge computer vision[C]//Proceedings of the ACM MobiCom 2019 Conference. 2019: 1-16.

[73] CRAWSHAW M. Multi-task learning with deep neural networks: A survey[A]. 2020.

[74] YUAN M, ZHANG L, LI X Y, et al. Comprehensive and efficient data labeling via adaptive model scheduling[C]//2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1858-1861.

[75] WANG J, FENG Z, CHEN Z, et al. Bandwidth-efficient live video analytics for drones via edge computing[C]//Proceedings of 2018 IEEE/ACM Symposium on Edge Computing (SEC). 2018: 159-173.

[76] YEO H, LIM H, KIM J, et al. Neuroscaler: Neural video enhancement at scale[C]// Proceedings of the ACM SIGCOMM 2022 Conference. ACM, 2022: 795–811.

[77] YUAN M, ZHANG L, WU Z, et al. High-quality activity-level video advertising[C]// Proceedings of the IEEE/ACM IWQoS 2020 Conference. 2020: 1-10.

[78] ZHANG H, ANANTHANARAYANAN G, BODIK P, et al. Live video analytics at scale with approximation and {Delay-Tolerance}[C]//Proceedings of the 14th USENIX NSDI Conference. 2017: 377-392.

[79] DEAN B. Twitch statistics[EB/OL]. 2022. https://backlinko.com/twitch-users.

[80] DONG C, LOY C C, TANG X. Accelerating the super-resolution convolutional neural network[C]//Proceedings of the ECCV 2016 Conference. Springer, 2016: 391-407.

[81] HSIEH K, ANANTHANARAYANAN G, BODIK P, et al. Focus: Querying large video datasets with low latency and low cost[C]//13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). 2018: 269-286.

[82] JIANG J, ANANTHANARAYANAN G, BODIK P, et al. Chameleon: scalable adaptation of video analytics[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. 2018: 253-266.

[83] CANEL C, KIM T, ZHOU G, et al. Scaling video analytics on constrained edge nodes[J]. Proceedings of Machine Learning and Systems, 2019, 1: 406-417.

[84] BELLARD F. Ffmpeg[EB/OL]. 2022. https://ffmpeg.org/ffmpeg.html.

[85] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. International Conference on Learning Representations (ICLR), 2016.

[86] BABOESCU F, VARGHESE G. Scalable packet classification[J]. ACM SIGCOMM Computer Communication Review, 2001, 31(4): 199-210.

[87] QI Y, XU L, YANG B, et al. Packet classification algorithms: From theory to practice[C]// Proceedings of the IEEE INFOCOM 2009 Conference. IEEE, 2009: 648-656.

[88] ERMAN J, ARLITT M, MAHANTI A. Traffic classification using clustering algorithms[C]// Proceedings of the 2006 SIGCOMM workshop on Mining network data. 2006: 281-286.

[89] NGUYEN T T, ARMITAGE G. A survey of techniques for internet traffic classification using machine learning[J]. IEEE communications surveys & tutorials, 2008, 10(4): 56-76.

[90] CORBILLON X, BOYRIVENT F, DE WILLIENCOURT G A, et al. Efficient lightweight video packet filtering for large-scale video data delivery[C]//Proceedings of the IEEE ICMEW

2016 Conference. 2016: 1-6.

[91] YAHIA M B, LOUEDEC Y L, SIMON G, et al. Http/2-based frame discarding for low-latency adaptive video streaming[J]. ACM Trans. Multimedia Comput. Commun. Appl., 2019, 15(1).

[92] LIU P, QI B, BANERJEE S. Edgeeye: An edge service framework for real-time intelligent video analytics[C]//Proceedings of the 1st international workshop on edge systems, analytics and networking. 2018: 1-6.

[93] BUILDOTS. Ai that scans a construction site can spot when things are falling behind [EB/OL]. 2020. https://buildots.com/.

[94] HAYES A. Youtube stats[EB/OL]. 2022. https://www.wyzowl.com/youtube-stats/.

[95] SPOLAOR N, LEE H D, TAKAKI W S R, et al. A systematic review on content-based video retrieval[J]. Engineering Applications of Artificial Intelligence, 2020, 90: 103557.

[96] AMODEI D, ANANTHANARAYANAN S, ANUBHAI R, et al. Deep speech 2: End-to-end speech recognition in english and mandarin[C]//Proceedings of the ICML 2016 Conference. PMLR, 2016: 173-182.

[97] LI J, LI Z, LU R, et al. Livenet: A low-latency video transport network for large-scale live streaming[C]//Proceedings of the ACM SIGCOMM 2022 Conference. ACM, 2022: 812–825.

[98] CRANMER M, SANCHEZ GONZALEZ A, BATTAGLIA P, et al. Discovering symbolic models from deep learning with inductive biases[J]. Advances in Neural Information Processing Systems, 2020, 33: 17429-17442.

[99] YAN X, HU S, MAO Y, et al. Deep multi-view learning methods: a review[J]. Neurocomputing, 2021, 448: 106-129.

[100] ZHAO B, LU H, CHEN S, et al. Convolutional neural networks for time series classification [J]. Journal of Systems Engineering and Electronics, 2017, 28(1): 162-169.

[101] ZHANG Y, YANG Q. A survey on multi-task learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.

[102] QIN L, CHEN S, ZHU X. Contextual combinatorial bandit and its application on diversified online recommendation[C]//Proceedings of the 2014 SIAM International Conference on Data Mining. SIAM, 2014: 461-469.

[103] ARULKUMARAN K, DEISENROTH M P, BRUNDAGE M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[104] MAO H, ALIZADEH M, MENACHE I, et al. Resource management with deep reinforcement learning[C]//Proceedings of the 15th ACM workshop on hot topics in networks. 2016: 50-56.

[105] LUONG N C, HOANG D T, GONG S, et al. Applications of deep reinforcement learning in

communications and networking: A survey[J]. IEEE Communications Surveys & Tutorials, 2019, 21(4): 3133-3174.

[106] TENSORFLOW. Tensorflow[EB/OL]. 2021. https://github.com/tensorflow/tensorflow.

[107] MINDSPORE. Mindsopre[EB/OL]. 2021. https://github.com/mindspore-ai/mindspore.

[108] YOUTUBE. Ugc dataset[EB/OL]. 2022. https://media.withyoutube.com/.

[109] CARDINALE F. Image super-resolution[EB/OL]. 2018. https://github.com/idealo/image-super-resolution.

[110] JADON A, OMAMA M, VARSHNEY A, et al. Firenet: a specialized lightweight fire & smoke detection model for real-time iot applications[A]. 2019.

[111] LIU C, ZHANG L, LIU Z, et al. Lasagna: Towards deep hierarchical understanding and searching over mobile sensing data[C/OL]//MobiCom '16: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking. New York, NY, USA: Association for Computing Machinery, 2016: 334–347. DOI: 10.1145/2973750.2973752.

[112] LI E, ZENG L, ZHOU Z, et al. Edge ai: On-demand accelerating deep neural network inference via edge computing[J]. IEEE Transactions on Wireless Communications, 2019, 19(1): 447-457.

[113] WANG X, HAN Y, WANG C, et al. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning[J]. IEEE Network, 2019, 33(5): 156-165.

[114] CHATZOPOULOS D, BERMEJO C, HUANG Z, et al. Mobile augmented reality survey: From where we are to where we go[J]. IEEE Access, 2017, 5: 6917-6950.

[115] DILSHAD N, HWANG J, SONG J, et al. Applications and challenges in video surveillance via drone: A brief survey[C]//2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2020: 728-732.

[116] GHIASI G, CUI Y, SRINIVAS A, et al. Simple copy-paste is a strong data augmentation method for instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 2918-2928.

[117] BULAT A, KOSSAIFI J, TZIMIROPOULOS G, et al. Toward fast and accurate human pose estimation via soft-gated skip connections[C]//2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). IEEE, 2020: 8-15.

[118] JIANG H, HE P, CHEN W, et al. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 2177-2190.

[119] ZHANG W, HE Z, LIU L, et al. Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading[C]//Proceedings of the 27th Annual International Conference on Mobile Computing and Networking. 2021: 201-214.

[120] ADARSH P, RATHI P, KUMAR M. Yolo v3-tiny: Object detection and recognition using one stage improved model[C]//2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020: 687-694.

[121] OSOKIN D. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose[C]// ICPRAM 2019-Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods. 2019: 744-748.

[122] SUN Z, YU H, SONG X, et al. Mobilebert: a compact task-agnostic bert for resource-limited devices[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 2158-2170.

[123] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[124] TAN M, CHEN B, PANG R, et al. Mnasnet: Platform-aware neural architecture search for mobile[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 2820-2828.

[125] HAN S, SHEN H, PHILIPOSE M, et al. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints[C]//Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. 2016: 123-136.

[126] HE Y, LIN J, LIU Z, et al. Amc: Automl for model compression and acceleration on mobile devices[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 784-800.

[127] ZHOU L, WEN H, TEODORESCU R, et al. Distributing deep neural networks with containerized partitions at the edge[C]//2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19). 2019.

[128] VALIANT L. Probably approximately correct: Nature's algorithms for learning and prospering in a complex world[M]. Basic Books (AZ), 2013.

[129] VAPNIK V, CHERVONENKIS A Y. On the uniform convergence of relative frequencies of events to their probabilities[J]. Theory of Probability & Its Applications, 1971, 16(2): 264-280.

[130] HARVEY N, LIAW C, MEHRABIAN A. Nearly-tight vc-dimension bounds for piecewise linear neural networks[C]//Conference on learning theory. PMLR, 2017: 1064-1068.

[131] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]// European conference on computer vision. Springer, 2014: 740-755.

[132] YUAN M, ZHANG L, LI X Y, et al. Adaptive model scheduling for resource-efficient data

labeling[J/OL]. ACM Trans. Knowl. Discov. Data, 2022, 16(4). DOI: 10.1145/3494559.

[133] DUAN Y, JIN C, LI Z. Risk bounds and rademacher complexity in batch reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2021: 2892-2902.

[134] CORTES C, KUZNETSOV V, MOHRI M, et al. Structured prediction theory based on factor graph complexity[J]. Advances in Neural Information Processing Systems, 2016, 29: 2514-2522.

[135] GLASMACHERS T. Limits of end-to-end learning[C]//Asian Conference on Machine Learning. PMLR, 2017: 17-32.

[136] PRAKASH A, CHITTA K, GEIGER A. Multi-modal fusion transformer for end-to-end autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 7077-7087.

[137] KOCH G, ZEMEL R, SALAKHUTDINOV R, et al. Siamese neural networks for one-shot image recognition[C]//ICML deep learning workshop: Vol. 2. Lille, 2015.

[138] TAIGMAN Y, YANG M, RANZATO M, et al. Deepface: Closing the gap to human-level performance in face verification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 1701-1708.

[139] LEAL-TAIXÉ L, CANTON-FERRER C, SCHINDLER K. Learning by tracking: Siamese cnn for robust target association[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016: 33-40.

[140] CHOLLET F. Xception: Deep learning with depthwise separable convolutions[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[141] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[142] INDRAWAN I, BAYUPATI I, PUTRI D P S. Markerless augmented reality utilizing gyroscope to demonstrate the position of dewata nawa sanga.[J]. International Journal of Interactive Mobile Technologies, 2018, 12(1).

[143] ANGUITA D, GHIO A, ONETO L, et al. A public domain dataset for human activity recognition using smartphones.[C]//Esann: Vol. 3. 2013: 3.

[144] TRIPURANENI N, JORDAN M, JIN C. On the theory of transfer learning: The importance of task diversity[J]. Advances in Neural Information Processing Systems, 2020, 33: 7852-7862.

[145] HADSELL R, CHOPRA S, LECUN Y. Dimensionality reduction by learning an invariant mapping[C]//2006 IEEE Computer Society Conference on Computer Vision and Pattern

Recognition (CVPR'06): Vol. 2. IEEE, 2006: 1735-1742.

[146] MULLAPUDI R T, CHEN S, ZHANG K, et al. Online model distillation for efficient video inference[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3573-3582.

[147] LEWIS D D, CATLETT J. Heterogeneous uncertainty sampling for supervised learning[M]// Machine learning proceedings 1994. Elsevier, 1994: 148-156.

[148] BALCAN M F, HANNEKE S, VAUGHAN J W. The true sample complexity of active learning[J]. Machine learning, 2010, 80(2): 111-139.

[149] LIU J, ZHANG Q. Code-partitioning offloading schemes in mobile edge computing for augmented reality[J]. Ieee Access, 2019, 7: 11222-11236.

[150] TIAN X, ZHU J, XU T, et al. Mobility-included dnn partition offloading from mobile devices to edge clouds[J]. Sensors, 2021, 21(1): 229.

[151] OSIA S A, SHAMSABADI A S, SAJADMANESH S, et al. A hybrid deep learning architecture for privacy-preserving mobile analytics[J]. IEEE Internet of Things Journal, 2020, 7(5): 4505-4518.

[152] MARSZAŁEK M, LAPTEV I, SCHMID C. Actions in context[C]//IEEE Conference on Computer Vision & Pattern Recognition. 2009.

[153] TRAN A, CHEONG L F. Two-stream flow-guided convolutional attention networks for action recognition[C]//The IEEE International Conference on Computer Vision Workshop (ICCVW). 2017.

[154] SERENGIL S I, OZPINAR A. Lightface: A hybrid deep face recognition framework[C/OL]// 2020 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE, 2020: 23-27. DOI: 10.1109/ASYU50717.2020.9259802.

[155] WANG G. Image captioning[EB/OL]. 2021. github.com/DeepRNN/image_captioning.

[156] HONNIBAL M, MONTANI I, VAN LANDEGHEM S, et al. spacy: Industrial-strength natural language processing in python[M/OL]. Zenodo, Honolulu, HI, USA, 2020. DOI: 10.5281/zenodo.1212303.

[157] KIM Y. Convolutional neural networks for sentence classification[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1746-1751. DOI: 10.3115/v1/D14-1181.

[158] PICZAK K J. ESC: Dataset for Environmental Sound Classification[C/OL]//Proceedings of the 23rd Annual ACM Conference on Multimedia. Brisbane, Australia: ACM Press, 2015: 1015-1018. DOI: 10.1145/2733373.2806390.

[159] GONG Y, CHUNG Y A, GLASS J. Ast: Audio spectrogram transformer[C/OL]//Proc. In-

terspeech 2021. 2021: 571-575. DOI: 10.21437/Interspeech.2021-698.

[160] GARDNER A, KANNO J, DUNCAN C A, et al. Measuring distance between unordered sets of different sizes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 137-143.

[161] O’ SHEA T J, ROY T, CLANCY T C. Over-the-air deep learning based radio signal classification[J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(1): 168-179.

[162] ZHURAVCHAK A, KAPSHII O, POURNARAS E. Human activity recognition based on wi-fi csi data-a deep neural network approach[J]. Procedia Computer Science, 2022, 198: 59-66.

[163] DEEPMIND G. Gemini[EB/OL]. 2023. https://deepmind.google/technologies/gemini/.

[164] GOOGLE. Bard[EB/OL]. 2023. https://bard.google.com/chat.

[165] JIANG P, XIN K, LI C, et al. High-efficiency device-cloud collaborative transformer model [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 2203-2209.

[166] THAPA C, ARACHCHIGE P C M, CAMTEPE S, et al. Splitfed: When federated learning meets split learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 36. 2022: 8485-8493.

[167] ZENG L, CHEN X, ZHOU Z, et al. Coedge: Cooperative dnn inference with adaptive workload partitioning over heterogeneous edge devices[J]. IEEE/ACM Transactions on Networking, 2020, 29(2): 595-608.

[168] PHAM N D, ABUADBBA A, GAO Y, et al. Binarizing split learning for data privacy enhancement and computation reduction[J]. IEEE Transactions on Information Forensics and Security, 2023.

[169] PASQUINI D, ATENIESE G, BERNASCHI M. Unleashing the tiger: Inference attacks on split learning[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021: 2113-2129.

[170] ABUADBBA S, KIM K, KIM M, et al. Can we use split learning on 1d cnn models for privacy preserving training?[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 305-318.

[171] PAN X, ZHANG M, JI S, et al. Privacy risks of general-purpose language models[C]//2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020: 1314-1331.

[172] DI X, LI J, QI H, et al. A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems[J]. PloS one, 2017, 12(9): e0184586.

[173] FARES N, ASKAR S. A novel semi-symmetric encryption algorithm for internet applications [J]. Journal of University of Duhok, 2016, 19(1): 1-9.

[174] HUGGINGFACE. Open-llm-leaderboard[CP/OL]. 2023. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

[175] PRABHAKARAN M, ROSULEK M. Cryptographic complexity of multi-party computation problems: Classifications and separations[C]//WAGNER D. Advances in Cryptology – CRYPTO 2008. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008: 262-279.

[176] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. ICLR, 2021.

[177] ZHANG C, XIA J, YANG B, et al. Citadel: Protecting data privacy and model confidentiality for collaborative learning[C/OL]//SoCC '21: Proceedings of the ACM Symposium on Cloud Computing. New York, NY, USA: Association for Computing Machinery, 2021: 546–561. https://doi.org/10.1145/3472883.3486998.

[178] RUAN J, CHEN Y, ZHANG B, et al. Tptu: Large language model-based ai agents for task planning and tool usage[A]. 2023. arXiv: 2308.03427.

[179] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [J]. OpenAI blog, 2019, 1(8): 9.

[180] JIANG A Q, SABLAYROLLES A, ROUX A, et al. Mixtral of experts[A]. 2024. arXiv: 2401.04088.

[181] BELTAGY I, PETERS M E, COHAN A. Longformer: The long-document transformer[A]. 2020.

[182] RUSH A M. The annotated transformer[C]//Proceedings of workshop for NLP open source software (NLP-OSS). 2018: 52-60.

[183] AINSWORTH S K, HAYASE J, SRINIVASA S. Git re-basin: Merging models modulo permutation symmetries[A]. 2022.

[184] LEE J, LEE Y, KIM J, et al. Set transformer: A framework for attention-based permutation-invariant neural networks[C]//International conference on machine learning. PMLR, 2019: 3744-3753.

[185] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [Z]. 2019.

[186] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[187] ZHENG F, CHEN C, ZHENG X, et al. Towards secure and practical machine learning via secret sharing and random permutation[J]. Knowledge-Based Systems, 2022, 245: 108609.

[188] WOLF T, DEBUT L, SANH V, et al. Huggingface's transformers: State-of-the-art natural language processing[A]. 2020. arXiv: 1910.03771.

[189] OPENAI. tiktoken[EB/OL]. 2024. https://github.com/openai/tiktoken.

[190] ZHAO T, RAN Q, YUAN L, et al. Information verification cryptosystem using one-time keys based on double random phase encoding and public-key cryptography[J]. Optics and Lasers in Engineering, 2016, 83: 48-58.

[191] BIANCHI T, BIOGLIO V, MAGLI E. Analysis of one-time random projections for privacy preserving compressed sensing[J]. IEEE Transactions on Information Forensics and Security, 2015, 11(2): 313-327.

[192] OLIVEIRA S R, ZAIANE O R. Privacy-preserving clustering by object similarity-based representation and dimensionality reduction transformation[C]//Proceedings of the 2004 ICDM Workshop on Privacy and Security Aspects of Data Mining. 2004: 40-46.

[193] WANG Y, WANG Y X, SINGH A. A theoretical analysis of noisy sparse subspace clustering on dimensionality-reduced data[J]. IEEE Transactions on Information Theory, 2018, 65(2): 685-706.

[194] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[M]. OpenAI, 2018.

[195] ZHANG B, SENNRICH R. Root mean square layer normalization[J]. Advances in Neural Information Processing Systems, 2019, 32.

[196] SHAZEER N. Glu variants improve transformer[A]. 2020.

[197] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

[198] CHILD R, GRAY S, RADFORD A, et al. Generating long sequences with sparse transformers [A]. 2019.

[199] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[A]. 2018.

[200] YAO S, LI J, LIU D, et al. Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency[C]//Proceedings of the 18th conference on embedded networked sensor systems. 2020: 476-488.

[201] WANG J, GUO S, XIE X, et al. Protect privacy from gradient leakage attack in federated learning[C]//IEEE INFOCOM 2022-IEEE Conference on Computer Communications. IEEE, 2022: 580-589.

[202] BENTLEY F, LUVOGT C, SILVERMAN M, et al. Understanding the long-term use of smart speaker assistants[J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018, 2(3): 1-24.

[203] FENG D, HAASE-SCHÜTZ C, ROSENBAUM L, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(3): 1341-1360.

[204] HE X, ZHOU Z, THIELE L. Multi-task zipping via layer-wise neuron sharing[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018: 6019-6029.

[205] SANH V, WOLF T, RUDER S. A hierarchical multi-task approach for learning embeddings from semantic tasks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019: 6949-6956.

[206] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[A]. 2015.

[207] LIU S, LIN Y, ZHOU Z, et al. On-demand deep model compression for mobile devices: A usage-driven model selection framework[C]//Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services. 2018: 389-400.

[208] GOLDBLUM M, FOWL L, FEIZI S, et al. Adversarially robust distillation[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 34. 2020: 3996-4003.

[209] BAI H, WU J, KING I, et al. Few shot network compression via cross distillation[C]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 34. 2020: 3203-3210.

[210] GUO P, HU B, LI R, et al. Foggycache: Cross-device approximate computation reuse[C]// Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. 2018: 19-34.

[211] Ning L, Guan H, Shen X. Adaptive deep reuse: Accelerating cnn training on the fly[C]//2019 IEEE 35th International Conference on Data Engineering (ICDE). 2019: 1538-1549.

[212] JIANG J, ANANTHANARAYANAN G, BODIK P, et al. Chameleon: scalable adaptation of video analytics[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. 2018: 253-266.

[213] YUAN M, ZHANG L, WU Z, et al. High-quality activity-level video advertising[C]//2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS). IEEE, 2020: 1-10.

[214] ELHOSEINY M, LIU J, CHENG H, et al. Zero-shot event detection by multimodal distributional semantic embedding of videos[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[215] BALTRUŠAITIS T, AHUJA C, MORENCY L P. Multimodal machine learning: A survey and taxonomy[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41 (2): 423-443.

[216] BLACK J, ELLIS T, ROSIN P. Multi view image surveillance and tracking[C]//Workshop on Motion and Video Computing, 2002. Proceedings. IEEE, 2002: 169-174.

[217] WANG J, FANG Z, ZHAO H. Alignnet: A unifying approach to audio-visual alignment [C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.

2020: 3309-3317.

[218] YUKSEL S E, WILSON J N, GADER P D. Twenty years of mixture of experts[J]. IEEE transactions on neural networks and learning systems, 2012, 23(8): 1177-1193.

[219] HJELM R D, FEDOROV A, LAVOIE-MARCHILDON S, et al. Learning deep representations by mutual information estimation and maximization[C/OL]//International Conference on Learning Representations. 2019. https://openreview.net/forum?id=Bklr3j0cKX.

[220] SELVARAJU R R, COGSWELL M, DAS A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[J]. International Journal of Computer Vision, 2020, 128(2): 336-359.

[221] REDMON J, FARHADI A. Yolov3: An incremental improvement[A]. 2018.

[222] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[223] GUO Y, SHI H, KUMAR A, et al. Spottune: transfer learning through adaptive fine-tuning [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4805-4814.

[224] TRIPURANENI N, JORDAN M, JIN C. On the theory of transfer learning: The importance of task diversity[J]. Advances in Neural Information Processing Systems, 2020, 33.

[225] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[226] BAHDANAU D, CHO K H, BENGIO Y. Neural machine translation by jointly learning to align and translate[C]//3rd International Conference on Learning Representations, ICLR 2015. 2015.

[227] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks [C]//Advances in neural information processing systems. 2014: 3104-3112.

[228] SHEN Z, HE Z, XUE X. Meal: Multi-model ensemble via adversarial learning[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2019: 4886-4893.

[229] LI Z, YE J, SONG M, et al. Online knowledge distillation for efficient pose estimation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11740-11750.

[230] SETTLES B. Active learning literature survey[R]. University of Wisconsin-Madison Department of Computer Sciences, 2009.

[231] MACKAY D J. Information-based objective functions for active data selection[J]. Neural computation, 1992, 4(4): 590-604.

[232] YOO D, KWEON I S. Learning loss for active learning[C]//Proceedings of the IEEE/CVF

conference on computer vision and pattern recognition. 2019: 93-102.

[233] RANGWANI H, AITHAL S K, MISHRA M, et al. A closer look at smoothness in domain adversarial training[C]//International Conference on Machine Learning. PMLR, 2022: 18378-18399.

[234] SINGH A. Clda: Contrastive learning for semi-supervised domain adaptation[J]. Advances in Neural Information Processing Systems, 2021, 34: 5089-5101.

[235] NG D, LAN X, YAO M M S, et al. Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets[J]. Quantitative Imaging in Medicine and Surgery, 2021, 11(2): 852.

[236] YANG Q, LIU Y, CHEN T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

[237] PETERSON D, KANANI P, MARATHE V J. Private federated learning with domain adaptation[A]. 2019.

[238] ZHOU Z H. Ensemble methods: foundations and algorithms[M]. CRC press, 2012.

[239] SVIRIDENKO M. A note on maximizing a submodular set function subject to a knapsack constraint[J]. Operations Research Letters, 2004, 32(1): 41-43.

[240] KUMAR A. Pygender-voice[EB/OL]. 2021. https://github.com/abhijeet3922/PyGender-Voice.

[241] TRAN D, BOURDEV L, FERGUS R, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

[242] SERENGIL S I, OZPINAR A. Lightface: A hybrid deep face recognition framework[C/OL]// 2020 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE, 2020: 23-27. DOI: 10.1109/ASYU50717.2020.9259802.

[243] LEVI G, HASSNER T. Age and gender classification using convolutional neural networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2015: 34-42.

[244] MOZILLA. Deepspeech[EB/OL]. 2021. https://github.com/mozilla/DeepSpeech.

[245] PYTORCH. Pytorch[EB/OL]. 2021. https://github.com/pytorch/pytorch.

[246] MARSZALEK M, LAPTEV I, SCHMID C. Actions in context[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 2929-2936.

[247] OLAFENWA M. Action-net[EB/OL]. 2021. github.com/OlafenwaMoses/Action-Net.

[248] OLAFENWA M. Traffic-net[EB/OL]. 2021. github.com/OlafenwaMoses/Traffic-Net.

[249] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database [C]//2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009: 248-255.

[250] TIELEMAN T, HINTON G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude[J]. COURSERA: Neural networks for machine learning, 2012, 4(2): 26-31.

[251] VENKATESWARA H, EUSEBIO J, CHAKRABORTY S, et al. Deep hashing network for unsupervised domain adaptation[C]//(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[252] CHU B, MADHAVAN V, BEIJBOM O, et al. Best practices for fine-tuning visual classifiers to new domains[C]//European conference on computer vision. Springer, 2016: 435-442.

致　　谢

感恩我的父母，给予我无条件的爱与支持。

感谢我的导师，李向阳教授和张兰教授，对我一直以来的悉心指导和帮助，在您的肩膀上我才有机会望见科学的远峰。

永远怀念我的姥姥和姥爷。

在读期间发表的学术论文与取得的研究成果

已发表论文（第一作者）

1. **Mu Yuan**, Lan Zhang, Xuanke You and Xiang-Yang Li, "PacketGame: Multi-Stream Packet Gating for Concurrent Video Inference at Scale", ACM SIGCOMM 2023.
2. **Mu Yuan**, Lan Zhang, Zimu Zheng, Yi-Nan Zhang and Xiang-Yang Li, "MLink: Linking Black-Box Models From Multiple Domains for Collaborative Inference", IEEE TPAMI 2023.
3. **Mu Yuan**, Lan Zhang, Fengxiang He, Xueting Tong, Miao-Hui Song, Zhengyuan Xu and Xiang-Yang Li, "InFi: End-to-End Learning to Filter Input for Resource-Efficiency in Mobile-Centric Inference", IEEE TMC 2023.
4. **Mu Yuan**, Lan Zhang, and Xiang-Yang Li, "MLink: Linking Black-Box Models for Collaborative Multi-Model Inference", AAAI 2022 (**Oral**).
5. **Mu Yuan**, Lan Zhang, Fengxiang He, Xueting Tong and Xiang-Yang Li, "InFi: end-to-end learnable input filter for resource-efficient mobile-centric inference", ACM MobiCom 2022.
6. **Mu Yuan**, Lan Zhang, Xiang-Yang Li, Lin-Zhuo Yang and Hui Xiong, "Adaptive Model Scheduling for Resource-efficient Data Labeling", ACM TKDD 2022.
7. **Mu Yuan**, Lan Zhang, Zhengtao Wu and Daren Zheng, "High-quality Activity-Level Video Advertising," IEEE/ACM IWQoS 2020.
8. **Mu Yuan**, Lan Zhang, Xiang-Yang Li and Hui Xiong, "Comprehensive and Efficient Data Labeling via Adaptive Model Scheduling," IEEE ICDE 2020.

已发表论文（合作）

1. Junyang Wang, Lan Zhang, Junhao Wang, **Mu Yuan**, Yihang Cheng, Qian Xu and Bo Yu, "GraphProxy: Communication-Efficient Federated Graph Learning with Adaptive Proxy", IEEE INFOCOM 2024.
2. Miao-Hui Song, Lan Zhang, **Mu Yuan**, Zichong Li, Qi Song, Yijun Liu and Guidong Zheng, "CoTel: Ontology-Neural Co-Enhanced Text Labeling", ACM WWW 2023.
3. Lan Zhang, Daren Zheng, **Mu Yuan**, Feng Han, Zhengtao Wu, Mengjing Liu and Xiang-Yang Li, "MultiSense: Cross-labelling and Learning Human Activi-

ties Using Multimodal Sensing Data", ACM ToSN 2023.

4. Zichong Li, Lan Zhang, **Mu Yuan**, Miao-Hui Song and Qi Song, "Efficient Deep Ensemble Inference via Query Difficulty-dependent Task Scheduling", IEEE ICDE 2023.
5. Xuanke You, Lan Zhang, Haikuo Yu, **Mu Yuan** and Xiang-Yang Li, "KATN: Key activity detection via inexact supervised learning", ACM UbiComp 2021.

发明专利

1. 一种异构推理后端上的可扩展负载均衡方法及系统，张兰，李向阳，**袁牧**，宋淼荟，专利号 ZL202311373531.0
2. 一种并发视频包过滤方法、系统及存储介质，张兰，李向阳，宋淼荟，**袁牧**，专利号 ZL202310632105.8
3. 一种多源视频中复杂行为识别的方法，张兰，李向阳，**袁牧**，专利号 ZL201910228241.4