arXiv:2301.01010v1 [cs.NI] 3 Jan 2023

移动边缘计算辅助增强现实系统中基于视频的人工智能推理任务联合优化

潘广进、张恒、徐书功*IEEE*会士、张顺清*IEEE*高级会员和陈晓静

摘要——**人工智能（AI）算法的高计算复杂度和能耗阻碍了其在增强现实（AR）系统中的应用。然而，移动边缘计算（MEC）使解决这一问题成为可能。本文研究在移动边缘计算系统中完成基于视频的人工智能推理任务的场景。我们构建了一个混合整数非线性规划问题（MINLP），以降低推理时延和能耗，并提高识别准确率。通过推导和实验，我们给出了推理复杂度模型和准确率模型的简化表达式。随后，采用交替优化对该问题进行迭代求解。具体而言，假设卸载决策已知后，该问题可分解为两个子问题：为本地完成推理任务的设备集分配资源，以及为卸载任务的设备集分配资源。针对卸载决策优化问题，我们提出了一种信道感知启发式算法。为进一步降低复杂度，我们提出了一种基于交替方向乘子法（ADMM）的分布式算法。该ADMM算法的计算复杂度较低，且随设备数量线性增长。数值实验表明，所提算法有效。本文还分析了时延、能耗与准确率之间的权衡关系。**

索引词—**移动增强现实、边缘智能、移动边缘计算、资源分配**。

一、引言

RECENTLY，网络、云计算、边缘计算、人工智能及其他技术的发展，激发了人们对元宇宙 [1]的无限想象。增强现实（AR）技术在帮助用户实现现实世界与虚拟世界交互方面发挥着至关重要的作用。与此同时，人工智能（AI）凭借其学习和推理能力，在自动语音识别（ASR） [2]，、自然语言处理（NLP） [3],计算机视觉（CV） [4], 等。在人工智能技术的助力下，增强现实能够实现更深入的场景理解和更具沉浸感的交互。

然而，人工智能算法，尤其是深度神经网络（DNN），通常具有很高的计算复杂度。对于计算能力和能耗受限的移动设备而言，要及时、可靠地完成深度神经网络推理颇具挑战。在 [5],的实验中发现，即使借助移动 GPU 加速，典型的单帧图像处理人工智能推理任务仍需约 600 毫秒。此外，在普通设备上持续执行上述推理任务，续航时间最多只有 2.5 小时。这些问题 [6]导致目前只有少数增强现实应用采用深度学习 [7], [8]。为了缩短深度神经网络的推理时间，一种方法是对神经网络进行网络剪枝 [7]。然而，若剪枝过多通道，可能会破坏模型；而且，微调也未必能恢复令人满意的准确率。

边缘人工智能 [9]–[11] 是解决这些问题的另一种方法。将移动边缘计算（移动边缘计算）与人工智能技术相结合，近年来已成为支持计算密集型任务的一种有前景的范式。边缘人工智能将人工智能模型的推理和训练过程转移到靠近数据源的网络边缘。因此，它能够缓解网络流量负载、时延和隐私问题。

A. 相关工作

许多现有研究利用移动边缘计算强大的计算能力，通过卸载来降低时延 [12],、能耗 [13], 或同时降低时延和能耗 [14]–[16] 。例如， [12] 在无人机辅助移动边缘计算系统中，通过优化用户的传输功率，构建了一个旨在最大限度降低增强型移动宽带和海量机器类通信用户处理时延的优化问题。 [13]提出了一种智能定价机制，用于协调多层设备的计算卸载，并降低能耗。 [14]采用斯塔克尔伯格博弈方法，优化任务分配系数、计算资源分配系数和传输功率，以最大限度降低基于非正交多址接入的移动边缘计算系统的能耗和时延。

对于边缘人工智能推理，现有研究已取得一些进展。文献 [17] 提出了一种框架，通过联合优化推理任务选择和下行协同波束成形，来最小化无线网络中的通信功耗。同样，文献 [18] 提出了一种智能反射面辅助边缘推理系统，并设计了一种任务选择策略

G. Pan、H. Zhang、S. Xu、S. Zhang和X. Chen任职于上海大学先进通信与数据科学研究院，地址：中国上海市200444。电子邮箱：{guangjin_pan, hengzhang, shugong, Shunqing, jodiechen}@shu.edu.cn。

本工作的部分内容已被 Globecom 2022 接收。本工作部分得到以下项目资助：国家自然科学基金委员会项目（项目编号：61871262、62071284 和 61901251），国家重点研发计划项目（项目编号：2017YFE0121400、2019YFE0196600 和 2022YFB2902000），上海市科学技术委员会创新计划（项目编号：20JC1416400 和 21ZR1422400），浦东新区科技发展基金，广东省重点领域研发计划项目（项目编号：2020B0101130012），佛山市科技创新团队项目（项目编号：FS0AA-KJ919-4402-0060），以及上海大学先进通信与数据科学研究院（SICS）的研究经费。通讯作者为徐书功。

以最小化上行链路和下行链路传输及计算的能耗。文献 [19] 分析并建模了人工智能驱动的时间关键型服务的传输错误概率、推理准确率和超时概率。文献 [20] 使用串联排队模型分析多个深度神经网络分区中深度学习任务的排队时延和处理时延。文献 [21] 联合优化服务部署、计算资源分配和无线电资源分配，以最小化用户的总时延和能耗。文献 [8] 将模型剪枝与深度神经网络分区相结合，使端到端时延降低了 4.81 倍。文献 [22] 设计了 Edgent 框架，可联合优化深度神经网络分区和深度神经网络规模调整，在满足应用时延要求的同时最大限度地提高推理准确率。这些研究通过实验测量推理时间 [8], [22]，或假设推理任务的计算复杂度与输入数据大小成正比，但未给出推导和证明 [20],[21]。然而，这些计算复杂度模型不够严谨，或无法推广至不同的神经网络模型。

关于准确率模型， [23] 中的作者设计了一种名为 FACT 的边缘网络编排算法，通过优化增强现实用户的边缘服务器分配和视频帧分辨率选择，提升基于边缘的增强现实系统的性能。然而， [23] 通过拟合特定任务的准确率曲线来构建准确率模型，该模型缺乏通用性。 [24]在本地压缩图像分辨率，并在边缘服务器上执行推理任务，旨在满足时延和能耗约束的同时最大化学习准确率。 [24] 提出使用抽象的非递减函数来描述准确率与输入图像大小之间的关系，但这种方法无法用于有区别地分析各种人工智能推理任务。当不同任务和模型联合部署时，需要进行联合优化。通用性不足的准确率模型或过于抽象的模型都可能对联合优化产生不利影响。因此，需要一种通用的准确率模型来衡量各种人工智能任务。

在上述研究中，大多数研究只考虑优化时延、能耗和准确率中的一项或两项性能指标。文献 [24]的作者联合考虑了图像识别场景中的时延、能耗和准确率。然而，该研究旨在满足时延、能耗和准确率约束的条件下最大化计算能力，且深度神经网络模型仅部署在边缘服务器上。文献 [6], [23], [25], 考虑了视频分析场景，但没有联合考虑时延、能耗和准确率。

B. 贡献与组织结构

本文考虑一个多用户移动边缘计算系统，并假设每台设备都执行基于视频的深度神经网络推理任务。每台设备可以是增强现实眼镜、移动机器人等。为了增强增强现实对场景的理解能力，我们需要利用时域信息来提升感知能力。因此，我们考虑基于视频的应用场景。对于基于视频的人工智能推理任务，有两种模式，例如逐帧识别模式（每次识别的输入为一帧）和多帧识别

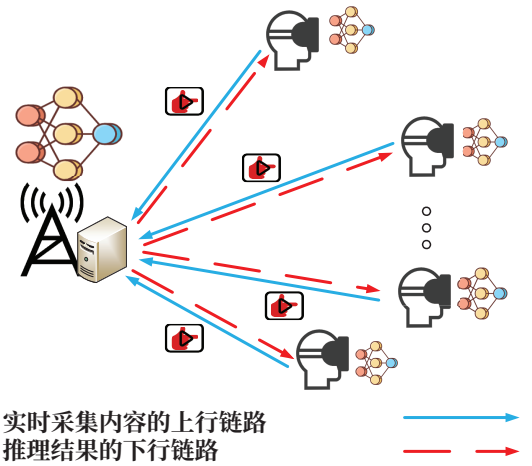


图 1. 多用户移动边缘计算系统模型。推理任务可在本地或边缘服务器上执行。当任务卸载到边缘服务器时，上行链路传输实时采集的内容，下行链路传输推理结果。

模式（每次识别的输入为多帧）。逐帧推理模式用于处理时间相关性较弱的任务，例如人脸识别和目标跟踪，并已在 [6], [23], [25]中得到研究。本文重点关注多帧识别任务，例如手势识别和动作识别任务。由于在空间维度进行采样会带来额外的计算开销 [24], ，我们只在时域进行采样。每次推理时，设备都会从历史帧中选取最近的若干帧进行传输或推理。

如图 1 所示，移动设备可以通过无线网络将采集的视频传输到边缘服务器。边缘服务器执行推理任务，并将结果发送回移动设备。然而，当边缘服务器的通信与计算资源不足时，设备可以在本地执行推理任务。我们将该问题建模为一个多目标优化问题，以优化时延、能耗和推理准确率。本文的主要贡献总结如下：

- 多维目标优化。高准确率、低时延和低能耗对于增强现实应用至关重要，必须进行联合优化。为探究时延、能耗与准确率之间的权衡关系，我们将基于视频的卸载问题表述为一个混合整数非线性规划问题（MINLP），旨在降低服务时延和能耗，并提高识别准确率。

- 通用计算复杂度与准确率模型。为了衡量具有不同架构和不同输入尺寸的神经网络模型的计算复杂度，我们引入乘累加运算次数（MACs）。我们通过实验阐明影响深度神经网络推理时延的主要因素，并表明乘累加运算次数可以作为衡量深度神经网络推理任务计算复杂度的有效指标。我们还提出了一种通用模型，用于表示推理准确率与输入帧数量之间的关系。该模型适用于不同的视频——

识别任务和不同的深度神经网络架构。我们给出了推理复杂度和准确率的简洁表达式，以简化优化问题。

•信道感知调度方案。为了解决优化问题，我们将原始问题分解。首先，假设卸载决策已知，分别求解在本地完成推理的设备集和将任务卸载到边缘服务器的设备集对应的资源分配问题。对于边缘深度神经网络推理，我们提出两种基于搜索和几何规划（GP）的算法来解决该问题。然后，为了获得最优卸载策略，我们提出一种信道感知启发式算法。通过交替优化迭代求解原始问题。

• 基于*ADMM*的分布式资源分配方案。为避免启发式算法的高复杂度，我们提出一种基于交替方向乘子法（ADMM）的算法。该算法将原始问题分解为可并行处理且易于求解的子问题。因此，基于ADMM的算法的总计算复杂度比启发式算法更具可扩展性，尤其是在设备数量较多时。

本文其余部分安排如下。第二节介绍系统模型，包括时延、能耗和准确率模型。第三节构建联合优化问题，并将原始问题转化为更易处理的问题。第四节提出一种信道感知启发式算法，以求解所提出的问题。第五节针对所提出的问题提出另一种基于交替方向乘子法的分布式资源分配算法，并分析该求解算法的计算复杂度。第六节给出数值结果与分析。最后，第七节总结全文。

II. 系统模型

本节介绍一种单小区移动边缘计算系统，并建立时延、能耗和准确率模型。如图1所示，我们考虑一个包含一个基站（BS）和 N 个移动设备的多用户移动边缘计算系统，记为集合 $\mathcal{N} = \{1, 2, \dots N\}$。每台设备都配备一个摄像头，并需要完成深度神经网络推理任务。由于设备计算资源有限，深度神经网络推理任务可以
部署在本地或边缘服务器上。当推理任务在本地执行时，有限的计算资源会导致更长的计算时延和更高的功耗。然而，当推理任务在边缘服务器上执行时，会产生额外的无线传输时延。此外，准确率也是深度神经网络推理任务中一个非常重要的优化目标。

A. 卸载框架

本文仅考虑二元卸载方法。二元卸载要求深度神经网络推理任务必须完全在设备端或移动边缘计算服务器上执行。深度神经网络计算卸载系统的概览如图所示

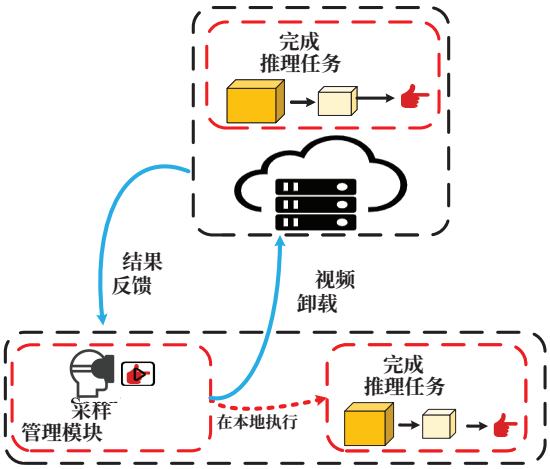


图 2. 视频采样与计算卸载系统概览。视频采样管理模块可以控制所采集视频的采样率，并确定用于人工智能推理的视频帧数量。设备可以根据无线信道信息和计算能力，将视频传输到边缘服务器，或在本地执行推理任务。

2。首先，设备在时间维度上对实时采集的视频进行采样，以获得包含一定数量帧的短视频。其次，执行深度神经网络推理任务。这些推理任务可以在设备本地执行，也可以在边缘服务器上执行。因此，每台设备的视频采样管理模块都需要选择合适的视频采样率（即需要输入多少帧），并决定是否将任务卸载至移动边缘计算服务器。分别用 D_n、 E_n 和 ϕ_n 表示设备 n的总时延、能耗和识别准确率。设备 n 的总时延和能耗可表示为：

$$D_n = (1 - x_n)D_n^{md} + x_n(D_n^t + D_n^e), \tag{1}$$

$$E_n = (1 - x_n)E_n^{md} + x_n E_n^t, \tag{2}$$

其中， x_n 表示推理任务是在本地还是边缘服务器上执行。 D_n^t 是上行链路的传输时延，D_n^{md}是本地推理时延， D_n^e 是边缘服务器完成推理所需的时延。 E_n^t 和 E_n^{md}分别是传输能耗和计算能耗。由于计算结果的数据大小很小，因此可以合理地忽略下载计算结果所需的时延和能耗。

B. 推理时延与能耗模型

推理时延取决于深度神经网络模型的架构、设备或服务器的算力，以及模型的输入。本节首先介绍深度神经网络模型计算复杂度的度量方法，然后给出推理时延和能耗的表达式。

不同的人工智能识别任务可能需要不同的人工智能模型架构，包括 Resnet-18、Resnet-34、Resnet-50、VGG-16 等经典人工智能模型 [26], [27]。为了更合理地优化人工智能推理任务，需要采用一种通用方法来评估不同人工智能模型的计算

复杂度。本文使用乘加运算次数（MACs） [28] 来衡量人工智能推理任务的计算复杂度。各层（如全连接（FC）层、卷积层等）的 MACs 计算方法见 [28]。以三维卷积神经网络（3DCNN）为例， n^{th} 设备的第 l^{th} 层的计算复杂度（以 MACs 衡量）可表示为：

$$c_{n,l} = \quad o_l o_{l+1} \prod_{j=0}^{2} K_l^j, \prod_{j=0}^{2} M_{n,l+1}^j, \tag{3}$$

其中， o_l 表示输入通道数， o_{l+1} 表示输出通道数， $\prod_{j=0}^{2} K_l^j$ 表示卷积核的尺寸， $\prod_{j=0}^{2} M_{n,\ l+1}^j$ 表示输出特征图的尺寸。 $j = 0$ 表示时间维度（帧数）， $j = 1,\ 2$ 表示空间维度（单帧的像素数）。请注意， o_l、 o_{l+1}和 $\prod_{j=0}^{2} K_l^j$ 均由神经网络架构决定，而$\prod_{j=0}^{2} M_{n,\ l+1}^j$ 取决于输入尺寸。输出特征尺寸与输入尺寸之间的关系可表示为：

$$M_{n,l+1}^j = \frac{M_{n,l}^j - K_l^j + 2d_l}{r_l} + 1, \tag{4}$$

其中， r_l 表示步幅， d_l 表示填充大小。

如上所述，深度神经网络模型的计算复杂度由层数、深度神经网络模型的架构以及输入和输出尺寸决定。本文主要关注输入视频帧数 M_n 对识别准确率以及通信与计算资源分配的影响。输入更多帧 M_n 会使推理结果更加准确，但通信与计算开销也会更大。 n^{th} 设备的任务计算复杂度可表示为 $C(M_n)$。

然后给出推理时延和能耗的表达式。设 f^{max} 和 f_n^{max}（单位为 CPU周期/秒）分别表示边缘服务器和移动设备 n的总计算资源。设 f_n^e 和 f_n^{md}（单位为 CPU周期/秒）分别表示边缘服务器和设备分配给设备 n 的计算资源。因此，计算资源满足 $\sum_{n\in\mathcal{N}} f_n^e \leq f^{max}$和 $f_n^{md} \leq f_n^{max}$。设备 n和 MEC 的计算时延可分别表示为：

$$D_n^{md} = \frac{\rho C(M_n)}{f_n^{md}}, \tag{5}$$

$$D_n^e = \frac{\rho C(M_n)}{f_n^e}, \tag{6}$$

其中， ρ （周期/MAC）表示完成一次乘法与加法所需的 CPU周期数，该数值取决于 CPU型号。

对于能耗，设 κ 为由相应设备 [24]，决定的系数，则设备 n 的计算能耗可表示为：

$$E_n^{md} = \kappa\rho C(M_n) {f_n^{md}}^2. \tag{7}$$

C. 传输时延和能耗模型

我们采用时分多址（TDMA）方法进行信道接入。具体而言，每个无线帧划分为 N 个时隙用于传输，每台设备只能在自己的时隙内传输。我们假设每个无线帧的长度为 ΔT，该长度足够短（例如，长期演进或新空口系统 [24] 中的 10 ms），且时隙长度为 $\Delta T t_n$。

设 h_n 和 p_n 分别表示设备 n的信道增益和传输功率。根据 [21], , 设备 n的可达数据速率可表示为：

$$R_n = B_w log_2\left(1 + \frac{p_n h_n}{B_w N_0}\right), \tag{8}$$

其中， B_w 和 N_0 分别表示带宽和加性高斯白噪声（AWGN）的方差。

令 d 表示单个视频帧的数据大小。由于我们只想分析时间维度信息（输入帧数 M_n）对识别准确率的影响，因此 d 为常数。每个无线帧中可传输的数据大小为 $\Delta T R_n t_n$。因此，每次传输需要$\lceil \frac{M_n d}{\Delta T R_n t_n} \rceil$ 个无线帧，其中$\lceil \cdot \rceil$表示向上取整函数。考虑到无线帧的时长远短于传输时延，卸载到移动边缘计算的传输时延可表示为：

$$D_n^t = \lceil \frac{M_n d}{\Delta T R_n t_n} \rceil \Delta T \approx \frac{M_n d}{R_n t_n}, \tag{9}$$

其中， t_n 是设备 n 的传输时间占比。此外，根据 [24], , 每台设备传输其视频所消耗的能耗可表示为：

$$E_n^t = \frac{M_n d}{R_n} p_n. \tag{10}$$

D. 推理任务准确率模型

如上所述，我们主要关注输入视频帧数 M_n 对识别准确率的影响。我们假设不同设备输入视频的质量相同。对于特定任务和深度神经网络模型，准确率仅由输入帧数决定。因此，设备 n 的准确率可表示为 $\phi_n = \Phi(M_n)$。根据 [29], , 更多帧将带来更高的推理准确率；随着输入帧数不断增加，性能增益会逐渐降低。一些先前的研究也表明，帧率与准确率之间的关系可表示为凹函数 [23]。因此，我们将 $\Phi(M_n)$ 定义为单调非递减函数，用于描述准确率与输入帧数之间的关系。

III. 问题构建

本节将构建优化问题，以降低系统时延和设备能耗，并提高准确率。我们分析了求解该问题的难点。为简化问题，我们对该问题进行了合理转换。

A. 原始问题建模

根据上述分析，结合式 (1)、(2)、(5)-(7)、(9) 和 (10)，n^{th} 设备的时延和能耗可表示为：

$$D_n = (1-x_n)\frac{\rho C(M_n)}{f_n^{md}} + x_n(\frac{\rho C(M_n)}{f_n^e} + \frac{M_n d}{R_n t_n}), \quad (11)$$

$$E_n = (1-x_n)\kappa\rho C(M_n) f_n^{md2} + x_n(\frac{M_n d}{R_n} p_n). \quad (12)$$

基于前文所述的系统模型，我们的目标是降低端到端时延和能耗，并提高识别准确率。每个设备都遵循二元卸载策略。总成本（时延、能耗和准确率）的数学优化问题可表示为：

问题 $\mathcal{P}1$ (原始问题)：

$$\underset{\{M_n,t_n,f_n^{md},f_n^e,x_n\}}{\text{minimize}} \quad \sum_{n\in\mathcal{N}} \Big(\beta_1 D_n + \beta_2 E_n - \beta_3 \Phi(M_n)\Big), \quad (13)$$

$$\text{subject to} \quad \Phi(M_n) \geq \alpha_n, \ \forall n \in \mathcal{N}, \quad (13a)$$

$$M_n \leq M_n^{max}, \ M_n \in \mathbb{Z}, \quad (13b)$$

$$\sum_{n\in\mathcal{N}} x_n t_n \leq 1, \quad (13c)$$

$$\sum_{n\in\mathcal{N}} x_n f_n^e \leq f^{max}, \quad (13d)$$

$$t_n, f_n^e \geq 0, \ \forall n \in \mathcal{N}, \quad (13e)$$

$$0 \leq f_n^{md} \leq f_n^{max}, \forall n \in \mathcal{N}, \quad (13f)$$

$$x_n \in \{0,1\}, \forall n \in \mathcal{N}, \quad (13g)$$

其中，α_n 表示识别准确率要求，β_1、β_2、β_3 为权重因子。(13a)表示每个设备的识别准确率要求。(13b)表示输入视频的帧数限制，$\mathbb{Z}$ 是整数集，M_n^{max}是输入视频的最大帧数。(13c)和(13d)分别表示通信和计算资源限制。(13f)限制了每个设备的计算资源。

原始问题 $\mathcal{P}1$ 中的优化变量包括输入视频帧数 M_n、传输时间占比 t_n、本地计算资源 f_n^{md}、边缘计算资源分配 f_n^e以及卸载决策 x_n。此外，式(13)中的第一项旨在降低计算和传输的总时延，第二项旨在降低设备的能耗，最后一项则旨在增加输入视频帧数并提高识别准确率，这是因为函数 $\Phi(M_n)$是单调非递减函数。

问题 $\mathcal{P}1$ 是一个非凸混合整数非线性规划问题，难以求解。首先，复杂度函数 $C(M_n)$ 是离散的，并且取决于深度神经网络的架构和输入视频的大小。随着输入帧数 M_n增加，计算复杂度也会增加。根据式（4），由于受到深度神经网络层结构的影响（例如三维卷积神经网络的步幅和填充大小），这种增长并不规则。因此，无法直接使用 $C(M_n)$ 进行优化。其次，如上所述，准确度函数 $\Phi(M_n)$ 是非递减的。然而，我们无法给出 $\Phi(M_n)$ 的确定性表达式，因此无法对其进行优化。此外，M_n 和 x_n 都是整数，这使得该问题难以求解。

B. 问题转换

为使问题 $\mathcal{P}1$ 更易处理，我们对问题进行转换。首先，给出计算复杂度函数 $C(M_n)$ 的近似表达式。根据式（3）和（4），三维卷积神经网络各层的计算复杂度与输入数据的大小成正比。对于其他类型的层，例如全连接层 [28]，也可以得出类似结论。基于上述结论并结合第六节A中的实验，为简洁地表示计算复杂度模型，$C(M_n)$ 可写为：

$$C(M_n) = m_{c,0} M_n + m_{c,1}, \quad (14)$$

其中 $m_{c,\,0} \geq 0$ 和 $m_{c,\,1}$ 为常数，且取决于网络模型。

其次，我们提出一个通用模型，用于表示准确率与输入视频帧数量之间的关系。考虑到函数 $\Phi(M_n)$ 单调不减，且随着输入帧数量增加，准确率增益会降低，结合第六节A中的实验，我们将函数 $\Phi(M_n)$ 建模为：

$$\Phi(M_n) = -\frac{m_{a,0}}{M_n + m_{a,1}} + m_{a,2}, \quad (15)$$

其中 $m_{a,0} \geq 0$, $m_{a,2} \geq 0$ 和 $m_{a,1} > -1$ 为常数，取决于推理任务的目标以及深度神经网络模型的架构。

最后，我们放宽变量 M_n 的取值范围。考虑到 $\Phi(M_n)$ 是一个单调非递减函数，并且取决于识别任务和网络架构，为了不失一般性，定义 M_n^{min} =argmin$_{M_n}$ $\Phi(M_n)$ $\Phi(M_n) \geq \alpha_n$, $M_n \in \mathbb{Z}$。我们也可以将 M_n 放宽为实轴上的一个闭连通子集，并将 (13a)、(13b) 写为 $M_n \in \left[M_n^{min}, M_n^{max}\right]$。那么，$[M_n]$ 可视为输入视频帧的数量，其中 $[\cdot]$ 表示取整。我们定义两个设备集合，即 $\mathcal{N}_0 = \{n \mid x_n = 0, n \in \mathcal{N}\}$ 和 $\mathcal{N}_1 = \{n \mid x_n = 1, n \in \mathcal{N}\}$。$\mathcal{F}_{0,n}$ 和 $\mathcal{F}_{1,n}$ 分别是集合 $\mathcal{N}_0$ 和 $\mathcal{N}_1$ 中设备 n 的成本函数。问题 $\mathcal{P}1$ 可重写为：

问题 $\mathcal{P}2$（转换后的问题）：

$$\underset{\{M_n,t_n,f_n^{md},f_n^e,x_n\}}{\text{minimize}} \quad \sum_{n\in\mathcal{N}_0} (1-x_n)\mathcal{F}_{0,n}(M_n, f_n^{md}) + \sum_{n\in\mathcal{N}_1} x_n \mathcal{F}_{1,n}(M_n, f_n^e, t_n), \quad (16)$$

$$\text{subject to} \quad M_n \in \left[M_n^{min}, M_n^{max}\right], \quad (16a)$$

$$(13c) - (13g),$$

其中

$$\mathcal{F}_{0,n}(M_n, f_n^{md}) = \beta_1 \frac{\rho C(M_n)}{f_n^{md}} + \beta_2 \kappa \rho C(M_n) f_n^{md2} - \beta_3 \Phi(M_n), \quad (17)$$

$$\mathcal{F}_{1,n}(M_n, f_n^e, t_n) = \beta_1 \frac{\rho C(M_n)}{f_n^e} + \beta_1 \frac{M_n d}{R_n t_n}$$

$$+\beta_2\frac{M_n dp_n}{R_n}-\beta_3\Phi(M_n). \tag{18}$$

IV. 优化问题求解

在本节中，我们将问题 $\mathcal{P}2$ 分解，并提出一种信道感知启发式算法来求解该问题。首先，假设卸载决策（即 $\{x_n\}$）已给定，我们分别针对集合 $\mathcal{N}_0$ 和 $\mathcal{N}_1$ 求解优化问题。其次，我们提出一种信道感知启发式算法来优化卸载决策 $\{x_n\}$。

A. $\mathcal{N}_0$的优化问题求解

对于集合 $\mathcal{N}_0$，即设备在本地执行推理任务时，优化问题变为:

问题 $\mathcal{P}_{\mathcal{N}_0}$ （ $\mathcal{N}_0$的问题）：

$$\underset{\{M_n,f_n^{md}\}}{\text{minimize}} \quad \mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}} \triangleq \sum_{n\in\mathcal{N}_0}\mathcal{F}_{0,n}(M_n,f_n^{md}), \tag{19}$$

满足约束条件 (13f)，(16a)。

$\mathcal{P}_{\mathcal{N}_0}$ 中的优化变量是输入视频帧数 M_n 和本地计算资源 f_n^{md}。令 $\{M_n^*,f_n^{md*}\}$表示 $\mathcal{P}_{\mathcal{N}_0}$的最优解。我们可以用闭式表达式推导出$\mathcal{P}_{\mathcal{N}_0}$ 的最优解。

定理*1*： $\mathcal{P}_{\mathcal{N}_0}$ 的最优解为:

$$f_n^{md*}=\min\{\sqrt[3]{(\frac{\beta_1}{2\beta_2\kappa})},f_n^{max}\}, \tag{20}$$

$$M_n^*=\min\{\max\{\sqrt{\frac{\beta_3 m_{a,0}}{\frac{\beta_1\rho m_{c,0}}{f_n^{md}}+\beta_2\kappa\rho m_{c,0}f_n^{md2}}}-m_{a,1},M_n^{min}\},M_n^{max}\}. \tag{21}$$

证明：请参见附录A。

根据定理*1*可知，最优本地CPU周期频率 f_n^{md}由权重因子 β_1、β_2以及CPU能耗系数 κ决定，并受相应上界 f_n^{max}的限制。更具体地说， f_n^{md}与 $\beta_1^{\frac{1}{3}}$成正比，与$\beta_2^{\frac{1}{3}}$和 $\kappa^{\frac{1}{3}}$成反比。至于输入视频帧数，当$\sqrt[3]{(\frac{\beta_1}{2\beta_2\kappa})}\le f_n^{max}$时，结合式（20）和（21），可得:

$$M_n^*=\min\{\max\{3^{-\frac{1}{2}}2^{\frac{1}{3}}\rho^{-\frac{1}{2}}\kappa^{-\frac{1}{6}}m_{c,0}^{-\frac{1}{2}}\beta_1^{-\frac{1}{3}}\beta_2^{-\frac{1}{6}}\beta_3^{\frac{1}{2}}m_{a,0}^{\frac{1}{2}}-m_{a,1},M_n^{min}\},M_n^{max}\}. \tag{22}$$

对应于每个设备的优化结果仅与该设备自身的参数有关，与其他设备的参数无关。

B. $\mathcal{N}_1$的优化问题求解

然后，我们求解 $\mathcal{N}_1$的优化问题。问题 $\mathcal{P}2$ 可写为:

问题 $\mathcal{P}_{\mathcal{N}_1}$ （问题 $\mathcal{N}_1$）：

$$\underset{\{M_n,f_n^e,t_n\}}{\text{minimize}} \quad \sum_{n\in\mathcal{N}_1}\mathcal{F}_{1,n}(M_n,f_n^e,t_n), \tag{23}$$

$$\text{subject to} \quad \text{(13c), (13d), (13e), (16a).}$$

算法1： 算法1：用于求解 $\mathcal{P}_{\mathcal{N}_1}$的基于搜索的算法

输入： 卸载策略 $\mathcal{N}_1$、信道增益 $\{h_n\}$以及其他系统参数。
输出： $\{M_n^\star,f_n^{e\star},t_n^\star\}$
将成本函数的结果初始化为 $\mathcal{F}^\star_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}$ to a 足够大的值;
使用（8）计算可达数据速率 $\{R_n\}$；
对于每个 $\{M_n\}\in\mathcal{M}$，**执行**
  计算 $\mathcal{F}_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}$ 使用 (27);
  若 $\mathcal{F}_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}<\mathcal{F}^\star_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}$，**则**
    $\{M_n^\star\}=\{M_n\}$; $\mathcal{F}^\star_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}=\mathcal{F}_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}$;
使用 (25) 和 (26) 计算 $\{f_n^{e\star}\}$ 和 $\{t_n^\star\}$；
返回 $\{M_n^\star\}$、$\{f_n^{e\star}\}$和 $\{t_n^\star\}$。

该问题 $\mathcal{P}_{\mathcal{N}_1}$ 中的优化变量包括输入视频帧数 M_n、边缘计算资源 f_n^e以及传输时间占比 t_n。令 $\{M_n^*,f_n^{e*},t_n^*\}$表示 $\mathcal{P}_{\mathcal{N}_1}$的最优解。我们可以利用拉格朗日乘子法求得 $\mathcal{P}_{\mathcal{N}_1}$ 的最优解。部分拉格朗日函数可写为:

$$\mathcal{L}_{\mathcal{P}_{\mathcal{N}_1}}=\sum_{n\in\mathcal{N}_1}\left(\frac{\beta_1\rho C(M_n)}{f_n^e}+\frac{\beta_1 M_n d}{R_n t_n}+\frac{\beta_2 M_n dp_n}{R_n}-\beta_3\Phi(M_n)\right)+\mu_0(\sum_{n\in\mathcal{N}_1}t_n-1)+\mu_1(\sum_{n\in\mathcal{N}_1}f_n^e-f^{max}), \tag{24}$$

首先，根据 (24)，假设给定 M_n^*，我们可以基于 Karush-Kuhn-Tucker (KKT) 条件求解问题 $\mathcal{P}_{\mathcal{N}_1}$。我们可以得到相对于 M_n 的 f_n^{e*}和 t_n^* 的函数表达式，如下述定理所示。

定理*2*： f_n^{e*}和 t_n^* 关于 M_n^* 的函数表达式如下:

$$f_n^{e*}=\frac{f^{max}\sqrt{C(M_n^*)}}{\sum_{i\in\mathcal{N}_1}\sqrt{C(M_i^*)}}, \tag{25}$$

$$t_n^*=\frac{\sqrt{\frac{M_n^*}{R_n}}}{\sum_{i\in\mathcal{N}_1}\sqrt{\frac{M_i^*}{R_i}}}. \tag{26}$$

证明：请参阅附录B。

结合式（23）、（25）和（26），问题 $\mathcal{P}_{\mathcal{N}_1}$ 可改写为仅包含变量M_n 的优化函数，如下:

问题 $\widetilde{\mathcal{P}_{\mathcal{N}_1}}$ （M_n 关于 $\mathcal{N}_1$ ）的优化问题:

$$\underset{\{M_n\}}{\text{minimize}} \quad \mathcal{F}_{\widetilde{\mathcal{P}_{\mathcal{N}_1}}}\triangleq\frac{\beta_1\rho}{f^{max}}(\sum_{n\in\mathcal{N}_1}\sqrt{C(M_n)})^2+\beta_1 d(\sum_{n\in\mathcal{N}_1}\sqrt{\frac{M_n}{R_n}})^2+\beta_2 dp_n(\sum_{n\in\mathcal{N}_1}\frac{M_n}{R_n})-\sum_{n\in\mathcal{N}_1}\beta_3\Phi(M_n), \tag{27}$$

算法2： 用于求解的基于几何规划的算法
求解 $\mathcal{P}_{\mathcal{N}_1}$

输入： 卸载策略 $\mathcal{N}_1$，信道增益 $\{h_n\}$，以及其他系统参数。
输出： $\{M_n^\star, f_n^{e\star}, t_n^\star\}$
使用（8）计算可达数据速率 $\{R_n\}$；
使用 CVX工具求解（29），得到 $\{\hat{M_n^\star}\}$;
$\{M_n^\star\} = \{[e^{\hat{M_n^\star}}]\}$;
使用（25）和（26）计算 $\{f_n^{e\star}\}$ 和 $\{t_n^\star\}$；
返回 $\{M_n^\star\}$、$\{f_n^{e\star}\}$和 $\{t_n^\star\}$。

满足式（16a）。

令 $\mathcal{M}_n^{opt} = \{M_n \mid M_n^{min} \le M_n \le M_n^{max}, M_n \in \mathbb{Z}\}$ 表示设备 n的可选视频帧数。通过搜索 $\{M_n\} \in \mathcal{M}$（其中 $\mathcal{M} = \{\{M_i\} \mid M_i \in \mathcal{M}_i^{opt}, i \in \mathcal{N}_1\}$）可获得最优解。基于搜索的算法的详细内容见算法1。

考虑到给定 M_n 时，问题$\mathcal{P}_{\mathcal{N}_1}$ 是凸的，算法1是全局最优的。然而，随着设备数量增多，基于搜索的算法的计算复杂度会变得非常高，甚至不可接受。本文还提出了一种基于几何规划的次优算法来求解问题 $\mathcal{P}_{\mathcal{N}_1}$。首先，我们放宽问题 $\mathcal{P}_{\mathcal{N}1}$的目标函数。引入函数 $\widehat{\Phi}(M_n) = -^{m_a}_{M_n{}^0} + m_{a,\,2}$，则 $\mathcal{P}_{\mathcal{N}_1}$可改写为:

问题$\mathcal{P}_{GP_{\mathcal{N}_1}}$（$\mathcal{N}_1$的基于几何规划的问题）：

$$\underset{\{M_n, f_n^e, t_n\}}{\text{minimize}} \quad \sum_{n\in\mathcal{N}_1}\left(\beta_1\frac{\rho C(M_n)}{f_n^e} + \beta_1\frac{M_n d}{R_n t_n} + \beta_2\frac{M_n d p_n}{R_n} - \beta_3\widehat{\Phi}(M_n)\right), \tag{28}$$

$$\text{subject to} \quad \text{(13c), (13d), (13e), (16a).}$$

这是一个非凸几何规划问题。受 [30], 启发，可以通过变量替换以及变换目标函数和约束，将该几何规划问题转化为凸问题。因此，引入变量 $M_n = \ln M_n$，$f_n^e = \ln f_n^e$，$\hat{t}_n = \ln t_n$，该问题可写为:

问题 $\widetilde{\mathcal{P}_{GP_{\mathcal{N}_1}}}$（$\mathcal{N}_1$）：转换后的基于几何规划的问题

$$\underset{\{\hat{M_n}, \hat{t_n}, \hat{f_n^e}\}}{\text{minimize}} \sum_{n\in\mathcal{N}_1}\left(\beta_1\rho m_{c,0}e^{\hat{M_n}-\hat{f_n^e}} + \beta_1\rho m_{c,1}e^{-\hat{f_n^e}} + \frac{\beta_1 d e^{\hat{M_n}-\hat{t_n}}}{R_n} + \frac{\beta_2 d p_n e^{\hat{M_n}}}{R_n} + \beta_3 m_{a,0}e^{-\hat{M_n}}\right), \tag{29}$$

$$\text{subject to} \quad \hat{M_n} \in \left[\ln M_n^{min}, \ln M_n^{max}\right], \forall n \in \mathcal{N}_1, \tag{29a}$$

$$\sum_{n\in\mathcal{N}_1} x_n e^{\hat{t_n}} \le 1, \tag{29b}$$

$$\sum_{n\in\mathcal{N}_1} x_n e^{\hat{f_n^e}} \le f^{max}, \tag{29c}$$

这是一个严格凸问题，可使用 CVX工具求解 [31]。考虑到 M_n 是整数，CVX优化的结果需要进行后处理。基于几何规划的算法的详细内容见算法2。

算法3： 用于优化卸载策略的信道感知启发式算法 $\{x_n\}$

输入： 与问题 $\mathcal{P}$1对应的参数。
输出： 卸载策略 $\mathcal{N}_0$ 和 $\mathcal{N}_1$。
使用以下方法计算集合 $\mathcal{N}$ 的成本函数 $\{\mathcal{F}_{0,n}\}$ (20) 和 (21);
设置 $\mathcal{N}_0 = \emptyset$、$\mathcal{N}_1 = \mathcal{N}$;
计算与以下内容对应的成本函数 $\{\mathcal{F}_{1,n}\}$ 使用算法1或算法2计算集合 $\mathcal{N}_1$；
设置 $Flag = 1$;
当 $Flag == 1$ **时执行**
  $k = \text{argmin}_n h_n, n \in \mathcal{N}_1$;
  $\mathcal{N}_0^* = \mathcal{N}_0 \cup \{k\}$, $\mathcal{N}_1^* = \mathcal{N}_1 - \{k\}$;
  计算成本函数 $\{\mathcal{F}_1^*$ $_{,n}\}$ 相应的对于集合 $\mathcal{N}_1^*$，使用算法1或算法2;
  如果 $\sum_{n\in\mathcal{N}_0}\mathcal{F}_{0,n} + \sum_{n\in\mathcal{N}_1}\mathcal{F}_{1,n} > {}_*\mathcal{F}_{0,\,n} + \sum_{n\in\mathcal{N}_1^*}\mathcal{F}_{1,n}^*$ **then**
    $\mathcal{F}_{1,n} = \mathcal{F}_{1,n}^* \; \forall n \in \mathcal{N}_1^*$;
    $\mathcal{N}_0 = \mathcal{N}_0^*$; $\mathcal{N}_1 = \mathcal{N}_1^*$;
  else
    $Flag = 0$;
返回 $\mathcal{N}_0$ 和 $\mathcal{N}_1$。

C. 卸载策略优化 $\{x_n\}$

考虑到当设备数量 N增大时，基于搜索的卸载策略算法的复杂度会变高。本节提出一种信道感知启发式算法，以优化卸载决策 $\{x_n\}$。受定理*1* 和定理*2*的启发，当在本地执行推理时，成本函数 $\mathcal{F}_{0,\,n}$ 以及优化变量 f_n^{md}、M_n 仅取决于设备自身的参数。然而，对于边缘集合 $\mathcal{N}_1$，成本函数与集合 $\mathcal{N}_1$中的设备数量及其参数有关。信道感知启发式算法见算法3。首先，计算每台设备的任务均在本地执行时集合 $\mathcal{N}_0$ 的成本函数 $\{\mathcal{F}_{0,n}\}$。其次，假设所有设备都卸载到边缘服务器进行推理，并且 $|\mathcal{N}_1| = N$。在每次迭代中，计算 $\mathcal{N}_1$ 中每台设备对应的成本函数 $\{\mathcal{F}_{1,n}\}$。我们选取集合 $\mathcal{N}_1$中信道增益最小的设备 k。尝试将设备 k 从集合 $\mathcal{N}_1$ 移至集合 $\mathcal{N}_0$，并计算新集合的成本。如果新集合的总成本降低，则继续下一次迭代。否则，将设备 k 移回集合$\mathcal{N}_1$。

V. 基于ADMM的方法进行联合优化

当 UE 数量增加时，信道感知启发式算法的复杂度会变高。本节提出一种基于ADMM的算法。该基于ADMM的算法可以将 $\mathcal{P}2$ 分解为 N 并行子问题。每个用户只需解决一个子问题，从而降低每台设备的平均复杂度。

A. 基于*ADMM*的问题转换

为了使原始问题易于处理，我们联合考虑问题 $\mathcal{P}_2$ 和问题 $\mathcal{P}_{GP_{\mathcal{N}_1}}$，并将该问题转化为一个基于几何规划的问题，

问题 $\mathcal{P}_3$ (转换后的基于几何规划的问题):

$$\begin{aligned}\underset{\left\{\hat{M}_n,\hat{t_n},\hat{f_n^{md}},\hat{f_n^e},x_n\right\}}{\text{minimize}} \quad & \sum_{n\in\mathcal{N}}\Big[(1-x_n)\hat{\mathcal{F}_{0,n}}(\hat{M}_n,\hat{f_n^{md}}) \\ & + x_n\hat{\mathcal{F}_{1,n}}(\hat{M}_n,\hat{f_n^e},\hat{t_n})\Big], \qquad (30)\\ \text{subject to} \quad & \hat{f_n^{md}} \le \ln f_n^{max}, \forall n\in\mathcal{N}, \qquad (30a)\\ & (13g),(29a)-(29c),\end{aligned}$$

其中，$\hat{}\ M_n = \ln\ M_n$、$\hat{f_n^{md}} = \ln\ f_n^{md}$、$\hat{f_n^e} = \ln\ f_n^e$和$\hat{}\ t_n = \ln\ t_n$。$\hat{\mathcal{F}}_{0,n}(\hat{}M_n,\hat{f_n^{md}})$和$\hat{\mathcal{F}}_{1,n}(\hat{}M_n,\hat{f_n^e},\hat{t_n})$由下式给出，

$$\begin{aligned}\hat{\mathcal{F}_{0,n}}(\hat{M}_n,\hat{f_n^{md}}) = \ & \beta_1\rho m_{c,0}e^{\hat{M}_n-\hat{f_n^{md}}} + \beta_1\rho m_{c,1}e^{-\hat{f_n^{md}}} \\ & + \beta_2\kappa m_{c,0}e^{\hat{M}_n+2\hat{f_n^{md}}} \\ & + \beta_2\kappa m_{c,1}e^{2\hat{f_n^{md}}} + \beta_3 m_{a,0}e^{-\hat{M}_n}, \qquad (31)\\ \hat{\mathcal{F}_{1,n}}(\hat{M}_n,\hat{f_n^e},\hat{t_n}) = \ & \beta_1\rho m_{c,0}e^{\hat{M}_n-\hat{f_n^e}} + \beta_1\rho m_{c,1}e^{-\hat{f_n^e}} \\ & + \frac{\beta_1 d e^{\hat{M}_n-\hat{t_n}}}{R_n} + \frac{\beta_2 d p_n e^{\hat{M}_n}}{R_n} \\ & + \beta_3 m_{a,0}e^{-\hat{M}_n}, \qquad (32)\end{aligned}$$

优化变量 $\{\hat{t_n},\hat{f_n^e}\}$ 在约束(29b)和(29c)中与各设备相互耦合。为分解问题 $\mathcal{P}_3$，我们引入局部变量 $\{y_n\}$ 和 $\{z_n\}$。然后，基于ADMM的问题可写为:

问题 $\mathcal{P}_4$ (基于*ADMM*的问题):

$$\begin{aligned}\underset{\left\{\hat{M}_n,\hat{t_n},\hat{f_n^{md}},\hat{f_n^e},x_n,y_n,z_n\right\}}{\text{minimize}} \quad & \sum_{n\in\mathcal{N}}\hat{\mathcal{F}}_n(x_n,\hat{M}_n,\hat{f_n^{md}},y_n,z_n) \\ & + g(\hat{f_n^e},\hat{t_n}), \qquad (33)\\ \text{subject to} \quad & y_n=\hat{f_n^e}, z_n=\hat{t_n}, \qquad (33a)\\ & (13g),(29a),(30a),\end{aligned}$$

其中，

$$\begin{aligned}\hat{\mathcal{F}}_n(x_n,\hat{M}_n,\hat{f_n^{md}},y_n,z_n) = \ & (1-x_n)\hat{\mathcal{F}_{0,n}}(\hat{M}_n,\hat{f_n^{md}}) \\ & + x_n\hat{\mathcal{F}_{1,n}}(\hat{M}_n,x_n,y_n), \qquad (34)\end{aligned}$$

$$g(\hat{f_n^e},\hat{t_n}) = \begin{cases}0, & \text{if}(\hat{f_n^e},\ \hat{t_n})\in\mathcal{G},\\ +\infty & ,\text{otherwise},\end{cases} \qquad (35)$$

and,

$$\mathcal{G} = \left\{(\hat{f_n^e},\ \hat{t_n})|\ \sum_{n\in\mathcal{N}_1}x_n e^{\hat{t_n}} \le 1, \sum_{n\in\mathcal{N}_1}x_n e^{\hat{f_n^e}} \le f^{max}\right\}. \qquad (36)$$

B. 基于*ADMM*的问题求解

问题 $\mathcal{P}_4$ 可以使用 ADMM 算法有效求解。我们可以将该问题 $\mathcal{P}_4$ 的部分增广拉格朗日函数写为:

$$\begin{aligned}\mathcal{L}_4(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\theta}) = & \sum_{n\in\mathcal{N}}\hat{\mathcal{F}}_n(x_n,\hat{M}_n,\hat{f_n^{md}},y_n,z_n) + g(\hat{f_n^e},\hat{t_n}) \\ & + \sum_{n\in\mathcal{N}}\theta_n^f(y_n-\hat{f_n^e}) + \sum_{n\in\mathcal{N}}\theta_n^t(z_n-\hat{t_n}) \\ & + \sum_{n\in\mathcal{N}}\frac{s}{2}(y_n-\hat{f_n^e})^2 + \sum_{n\in\mathcal{N}}\frac{s}{2}(z_n-\hat{t_n})^2, \qquad (37)\end{aligned}$$

其中 $\boldsymbol{u} = \{x_n,\hat{}M_n,\ \hat{f_n^{md}},\ y_n,\ z_n\}$、$\boldsymbol{v} = \{\hat{f_n^e},\ \hat{t_n}\}$、$\boldsymbol{\theta} = \{\theta_n^f,\theta_n^t\}$，且 s 为固定步长。因此，对偶函数为:

$$\begin{aligned}p(\boldsymbol{\theta}) = & \underset{\boldsymbol{u},\boldsymbol{v}}{\text{minimize}}\ \mathcal{L}_4(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\theta}) \qquad (38)\\ & \text{subject to } (13g),(29a),(30a),\end{aligned}$$

对偶问题可表示为，

$$\underset{\boldsymbol{\theta}}{\text{maximize}}\ p(\boldsymbol{\theta}), \qquad (39)$$

问题 (38) 可通过迭代更新 $\boldsymbol{u}$、$\boldsymbol{v}$ 和 $\boldsymbol{\theta}$ [32] 求解。令 $\{\boldsymbol{u}^i,\boldsymbol{v}^i,\boldsymbol{\theta}^i\}$ 表示第 i^{th} 次迭代中的取值。在第 i^{th} 次迭代中，变量的更新策略如下，

1) 步骤 *1*：局部变量更新。在此步骤中，我们首先更新局部变量 $\boldsymbol{u}$。给定变量 $\boldsymbol{v}^i$ 和 $\boldsymbol{\theta}^i$，我们通过以下方式最小化 $\mathcal{L}_4(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\theta})$，

$$\boldsymbol{u}^{i+1} = \underset{\boldsymbol{u}}{\text{argminimize}}\ \mathcal{L}_4(\boldsymbol{u},\boldsymbol{v}^i,\boldsymbol{\theta}^i). \qquad (40)$$

问题 (39) 可分解为 N 个并行子问题。对于每个子问题，我们考虑 $x_n=0$ 和 $x_n=1$ 两种情况，并将问题表示为，

$$\begin{cases}\underset{\{\hat{M}_n,\hat{f_n^{md}},y_n,z_n\}}{\text{minimize}}\ \hat{\mathcal{F}_{0,n}}(\hat{M}_n,\hat{f_n^{md}}) = \theta_n^f y_n + \sum_{n\in\mathcal{N}}\frac{s}{2}(y_n-\hat{f_n^e})^2 \\ \qquad +\theta_n^t z_n + \sum_{n\in\mathcal{N}}\frac{s}{2}(z_n-\hat{t_n})^2, \qquad \text{if } x_n=0,\\ \underset{\{\hat{M}_n,y_n,z_n\}}{\text{minimize}}\ \hat{\mathcal{F}_{1,n}}(\hat{M}_n,y_n,z_n) = \theta_n^f y_n + \sum_{n\in\mathcal{N}}\frac{s}{2}(y_n-\hat{f_n^e})^2 \\ \qquad +\theta_n^t z_n + \sum_{n\in\mathcal{N}}\frac{s}{2}(z_n-\hat{t_n})^2, \qquad \text{if } x_n=1.\end{cases} \qquad (41)$$

这两个问题都是严格凸问题，可以使用 CVX工具 [31]求解。因此，我们可以计算 $x_n=0$ 和 $x_n=1$ 的目标值，并选择较小者作为最终结果。求解 N个并行子问题后，(40) 的最优解为 $\boldsymbol{u}^{i+1} = \{(x_n)^{i+1},\ (\hat{}M_n)^{i+1},\ (\hat{}f_n^{md})^{i+1},\ (y_n)^{i+1},\ (z_n)^{i+1}\}$。

2）步骤 *2*：全局变量更新。在第二步中，我们更新全局变量 $\boldsymbol{v}$。根据 (35) 中 $g(\boldsymbol{v})$ 的定义，在最优解处必须满足 $\boldsymbol{v}^{i+1}\in\mathcal{G}$。因此，该子问题可以等价地改写为:

$$\begin{aligned}\boldsymbol{v}^{i+1} = & \underset{\{\hat{f_n^e},\hat{t_n}\}}{\text{argminimize}} \sum_{n\in\mathcal{N}}(\theta_n^f)^i(-\hat{f_n^e}) + \sum_{n\in\mathcal{N}}(\theta_n^t)^i(-\hat{t_n}) \\ & + \sum_{n\in\mathcal{N}}\frac{s}{2}(y_n^{i+1}-\hat{f_n^e})^2 + \sum_{n\in\mathcal{N}}\frac{s}{2}(z_n^{i+1}-\hat{t_n})^2, \qquad (42)\\ & \text{subject to,}\quad (29b),(29c).\end{aligned}$$

该问题也可以通过 CVX 工具 [31]求解。我们提出了一种低复杂度方案来求解该子问题。

算法4： 算法4：基于ADMM的算法

输入： 问题 $\mathcal{P}1$ 对应的参数。

输出： $\{x_n, M_n, f_n^{md}, f_n^e, t_n\}$

初始化 $i = 0, \{\boldsymbol{u}^i, \boldsymbol{v}^i, \boldsymbol{\theta}^i\} = 0, s = 0.5,$ $\mu_f^\star = \mu_t^\star = 10^6, \delta = 10^{-4}$;

重复

  对于每个 $n \in \mathcal{N}$ **执行**

    通过求解（41）更新 $\boldsymbol{u}^{i+1}$，并选择较小的结果;

  对于每个 $n \in \mathcal{N}$ **执行**

    使用（43）更新全局变量 $\boldsymbol{v}^{i+1}$，并（44）;

  对每个 $n \in \mathcal{N}$ **执行**

    使用（45）和（46）更新乘子 $\boldsymbol{\theta}^{i+1}$;

  $i = i + 1$;

直到 $|\mathcal{F}^i - \mathcal{F}^{i+1}| < \delta$;

$M_n = e^{\hat{}M_n}$, $f_n^{md} = e^{\hat{}f_n^{md}}$, $f_n^e = e^{\hat{}f_n^e}$, $t_n = e^{\hat{}t_n}$;

返回 $\{x_n, M_n, f_n^{md}, f_n^e, t_n\}$。

考虑约束条件 (29b) 和 (29c)，令 μ_f 和 μ_t表示拉格朗日乘子。该子问题的闭式最优解可表示为:

$$(\hat{f_n^e})^{i+1} = y_n{}^{i+1} + \frac{(\theta_n^f)^i - \mu_f}{s}, \tag{43}$$

$$(\hat{t_n})^{i+1} = z_n{}^{i+1} + \frac{(\theta_n^t)^i - \mu_t}{s}, \tag{44}$$

其中 μ_f 可通过二分搜索法在 (0，$\mu_f^\star$) 上求得，直到满足 $\sum_{n\in\mathcal{N}_1} x_n e^{f_n^e} \le f^{max}$。$\mu_f^\star$是一个足够大的值。这是因为当 $\mu_f \ge 0$ 时，$(\hat{f_n^e})^{i+1}$ 单调不增。类似地，μ_t 可通过二分搜索法在 (0，$\mu_t^\star$) 上求得，其中 $\mu_t^\star$是一个足够大的值，直到满足 $\sum_{n\in\mathcal{N}_1} x_n e^{t_n} \le 1$。

3) 步骤 *3*：乘子更新。在此步骤中，我们使用已获得的全局变量 $\boldsymbol{v}$ 和局部变量 $\boldsymbol{u}$ 来更新乘子 $\boldsymbol{\theta}$。更新方法如下:

$$(\theta_n^f)^{i+1} = (\theta_n^f)^i + s(y_n^{i+1} - (\hat{f_n^e})^{i+1}), \tag{45}$$

$$(\theta_n^t)^{i+1} = z_n{}^{i+1} + s(z_n^{i+1} - (\hat{t_n})^{i+1}), \tag{46}$$

重复上述三个步骤，直到成本函数不再下降。成本函数为 $\mathcal{F}^i = \sum_{n\in\mathcal{N}}[1 - (\hat{}, x_n^i \mathcal{F}_{0\,n} \; M_n{}^i \; f_n^{md\,i} + x_n^i \mathcal{F}_{1\,n} \; M_n{}^i \; f_n^{e\,i} \; t_n{}^i]$ $((\hat{}), (\hat{}))\hat{}, ((\hat{}), (\hat{}), (\hat{}))$。我们将 ADMM 算法的求解步骤总结为算法 4。

作为一种分布式迭代算法，基于 ADMM 的方案在设备与基站之间进行迭代，而非在本地迭代，从而能够在识别过程中进行在线优化。在每次迭代中，$\boldsymbol{u}^i$ 在本地计算后发送至移动边缘计算。移动边缘计算收到所有设备发送的 $\boldsymbol{u}^i$ 后，更新 $\boldsymbol{v}^i$ 和 $\boldsymbol{\theta}^i$，并将其发送至设备，以完成一次迭代。因此，ADMM 算法的迭代是一个在线收敛过程，能够适应信道的细微变化。

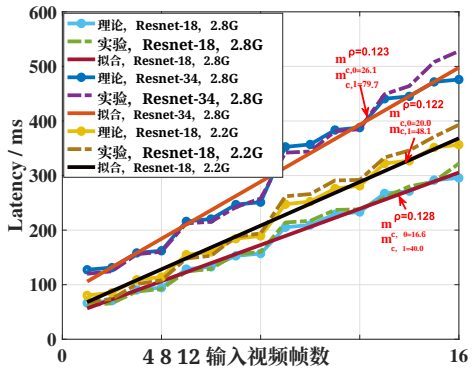


图3. 理论时延曲线、实验时延曲线以及与实验时延对应的拟合曲线。Resnet-18 和 Resnet-34 是两种经典的神经网络架构。CPU 的频率分别为 2.8G 和 2.2G。

C. 算法计算复杂度分析

本部分分析所提算法的计算复杂度。首先，求解问题 $\mathcal{P}_{\mathcal{N}_0}$ 的复杂度为 $O(|\mathcal{N}_0|)$。其次，如上所述，根据 [33]，采用内点法时，算法1的复杂度为 $O(\prod_{n\in\mathcal{N}_1}|\mathcal{M}_n^{opt}|)$，算法2的复杂度为 $O((3|\mathcal{N}_1|)^{3.5})$。当使用算法1求解$\mathcal{P}_{\mathcal{N}_1}$，并使用基于搜索的算法优化卸载策略时，计算复杂度为 $O(2^N \prod_{n\in\mathcal{N}}|\mathcal{M}_n^{opt}|)$。当使用算法1求解 $\mathcal{P}_{\mathcal{N}_1}$，并使用算法3优化卸载策略时，计算复杂度为 $O(N\prod_{n\in\mathcal{N}}|\mathcal{M}_n^{opt}|)$。此外，使用算法2求解 $\mathcal{P}_{\mathcal{N}_1}$ 并使用算法3优化卸载策略的计算复杂度为 $O(N^{4.5})$。对于基于 ADMM 的算法，由于每个步骤的复杂度为 $O(\mathcal{N})$，一次迭代的总体复杂度为 $O(\mathcal{N})$。

VI. 数值结果

本节通过仿真评估所提算法的性能。对于所有仿真结果，除非另有说明，下行带宽设为 $B_w = 5$ MHz，功率谱密度设为 $N_0 = -174$dBm/Hz [24]。根据 [17]，，路径损耗建模为 $PL = 128.1 + 37.6\log_{10}(D)$ dB，其中 D 是设备与基站之间以千米为单位的距离。设备随机分布在 [500m 500m]范围内。移动边缘计算服务器和设备的计算资源分别设为 1.8 GHz 和 22 GHz。识别准确率要求和输入视频帧的最大数量分别设为 $\alpha_n = 0.86$ 和 $M_n^{max} = 16$。系数 κ 由相应设备决定，本文根据 [24] 将其设为 10^{-28}。输入视频的大小为 $112*112*M_n$。此外，计算复杂度系数 ρ 设为 0.12 cycle/MAC，该数值通过第 VI-A 节中的多次实验获得。权重 β_1，β_2，β_3 分别设为 0.2、0.2、0.6。

A. 模型验证

首先，我们通过实验测量获得复杂度系数。计算复杂度系数的计算方法如下。首先，计算

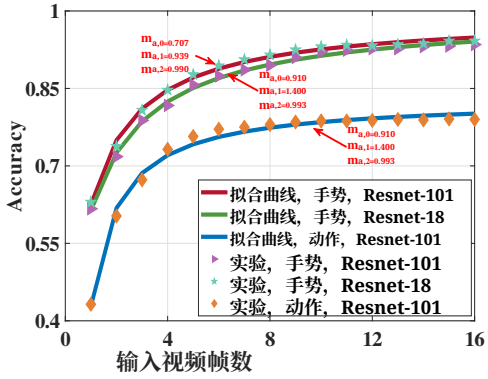


图4. 手势识别任务和动作识别任务的实验曲线与拟合曲线。

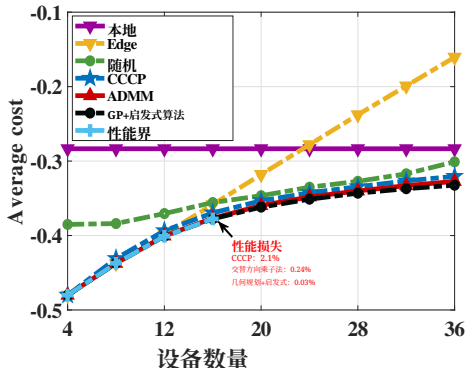


图5. 不同设备数量下所提方案和基准方案的平均成本。

输入视频帧数量不同时深度神经网络模型的乘加运算次数，记为 $\{C\}$。我们使用 Flops Counter 工具 [34] 计算乘加运算次数。其次，对输入视频帧数量不同的情况分别执行 100 次推理任务，并将平均推理时延记录为 $\{t\}$。最后，通过 $\rho = {}^{sum}(\{C\})\ sum(\{t\})$ 计算推理时延与乘加运算次数之间的系数。我们使用Intel(R) Xeon(R) E5-2630 CPU 进行测试。测试采用 Resnet-18 和 Resnet-34，并将 CPU 的最高频率限制为 2.8G 和 2.2G。图3展示了理论（基于乘加运算次数）时延曲线、实验时延曲线以及与实验时延对应的拟合曲线。从图3可以看出，理论时延与实验时延相近，这证明乘加运算次数可用于建模计算复杂度。我们还发现，线性拟合曲线能够近似表示计算复杂度：Resnet-18、2.8G 条件下的均方根误差（RMSE）为 9 ms，Resnet-34、2.8G 条件下为 17 ms，Resnet-18、2.2G 条件下为 11 ms。推理时延与输入视频帧的数量、深度神经网络模型的架构以及设备的能力有关。此外，三种条件下的计算复杂度系数分别为 0.128、0.122 和 0.123。因此，在后续实验中，我们设定为 $\rho = 0.12$ cycle/MAC。

我们选择手势识别和动作识别任务来验证准确率模型。我们使用 Jester数据集 [35]，，这是目前公开的最大手势数据集，用于测试手势识别任务。对于动作识别任务，我们使用 Kinetics-400数据集 [36]。我们选择 Resnet-18 和 Resnet-101 进行测试。如图 4 所示，在不同任务和不同网络模型下，准确率曲线均符合非递减函数的特性。此外，随着输入帧数量增加，准确率的性能增益会逐渐降低。这是因为随着输入帧数量增加，时域中引入的信息增益会减少。拟合曲线可以近似表示准确率与输入帧数量之间的关系。在使用 Resnet-101 模型的手势识别任务中，

采用 Resnet-18 模型的手势识别任务和采用 Resnet-101 模型的动作识别任务，其均方根误差分别为 0.0054、0.0048 和 0.0095。我们将

以 Resnet-18 和手势识别任务为例开展后续实验。

B. 平均成本的仿真结果

在本节中，我们比较所提方案与一些基准方案。我们进行了100次测试，并计算每台设备的平均成本以及每次测试的平均运行时间。我们比较以下方案。

*1)*搜索+搜索：我们使用基于搜索的算法求解 $\mathcal{P}_{\mathcal{N}_1}$，并使用启发式算法优化卸载策略。

2) 搜索+启发式：我们使用基于搜索的算法求解$\mathcal{P}_{\mathcal{N}_1}$，并使用基于搜索的算法优化卸载策略。

3) 几何规划+启发式：我们使用基于几何规划的算法求解$\mathcal{P}_{\mathcal{N}_1}$，并使用信道感知启发式算法优化卸载策略。

4) 交替方向乘子法：我们使用基于ADMM的算法求解原始问题。

*5)**CCCP** [37]*：我们使用凹凸过程（CCCP）算法决定是否将推理任务卸载到边缘服务器。然后，我们使用定理*1* 和基于几何规划的算法进行资源分配。

6) 随机：所有推理任务均随机在本地或边缘服务器上执行。我们使用定理*1* 和基于几何规划的算法进行资源分配。

7) 本地：所有推理任务均在本地执行。我们使用定理*1*进行本地资源分配。

*8)*边缘：所有推理任务均在边缘服务器上执行。我们使用基于几何规划的算法进行资源分配

在图5中，我们绘制了不同设备数量下各方案的平均成本。Search+Heuristic方案和Search+Search方案的性能相同，代表了性能界。当设备数量超过16时，由于计算复杂度无法接受，因此不再显示性能界。从图5可以看出，所提方案优于基准方案。与性能界相比，GP+Heuristic方案的性能略有下降，这是由于对准确度函数 $\Phi(M_n)$进行了松弛。ADMM方案的性能不如GP+Heuristic方案，但优于CCCP

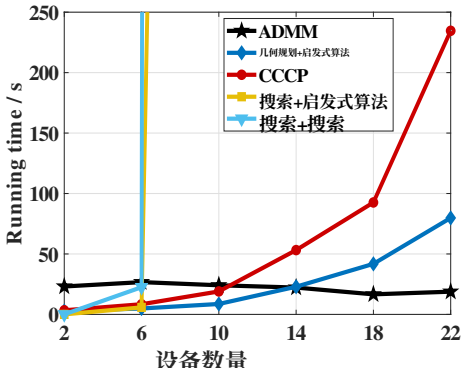


图 6. 所提算法在不同设备数量下的平均运行时间。

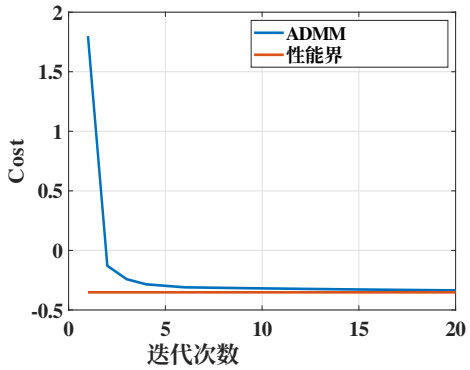


图 7. 成本函数与迭代次数的对应曲线。

方案。例如，当设备数量为16时，CCCP、ADMM和GP+Heuristic方案相较于性能界的性能损失分别为2.1%、0.24%和0.03%。此外，当设备数量少于8时，仅在边缘执行任务的方案的成本几乎等于所提GP+Heuristic方案的成本。这是因为设备数量较少时，所有设备都能从在边缘服务器上执行推理中获益。如果推理任务仅在本地执行，设备的平均成本将不会变化，因为各设备的本地资源互不影响。

在图 6 中，我们绘制了不同设备数量下各方案的平均运行时间。当设备数量超过 6 时，Search+Heuristic 和 Search+Search 场景的运行时间变得无法接受。GP+Heuristic 方案提高了求解效率。GP+Heuristic 的运行时间短于 CCCP 方案。然而，随着设备数量增加，求解复杂度仍不尽如人意。至于基于 ADMM 的方案，由于基于 ADMM 的算法是一种分布式算法，且更新全局变量的复杂度远低于更新局部变量的复杂度，因此我们只考虑每台设备的平均运行时间。随着设备数量增加，基于 ADMM 的方案的平均运行时间并未改善。值得注意的是，在基于 ADMM 的方案中，当 $|\mathcal{F}^i - \mathcal{F}^{i+1}| < \delta$ 时，迭代停止，其中 $\delta = 10^{-5}$。基于阈值的停止条件会导致不同情况下的迭代次数不同。设备数量不同，平均迭代次数也不同，因此运行时间也会有所差异。因此，18 台设备的平均运行时间短于 14 台和 22 台设备。

假设每执行一次推理任务，基于 ADMM 的方案就迭代一次，我们绘制了成本函数与迭代次数的对应曲线。如图 7 所示，基于 ADMM 的方案在完成 3–5 次迭代后即可收敛到可接受的性能。随着迭代次数增加，性能将逐渐接近最优性能。这表明 ADMM

算法可以通过在线迭代收敛。我们还测试了每台设备每次迭代的运行时间，平均约为 278 毫秒。

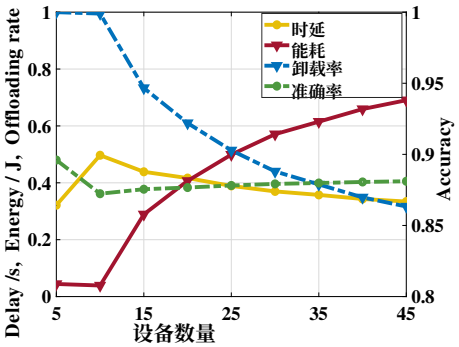
表 I 时延、边缘设备和本地设备的能耗与准确率

	本地设备	边缘设备
设备数量	12.3	12.7
平均时延	0.24 秒	0.52 秒
平均能耗	1.00 焦耳	0.025 焦耳
平均准确率	0.886	0.866

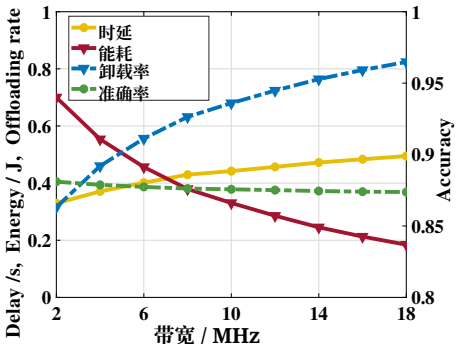
C. 时延、能耗和准确率的仿真结果

本节比较平均时延、能耗、准确率和卸载率（在边缘服务器上执行推理的设备所占比例）。我们考虑不同的设备数量、带宽、边缘计算资源以及权重 β_1、 β_2、 β_3。我们使用 GP+启发式方案进行测试。表 I 展示了在默认实验设置下，本地完成推理的设备与在边缘完成推理的设备之间的比较。平均而言，有 12.7 台设备选择卸载到边缘服务器执行推理。与边缘设备相比，本地设备的时延更低、准确率更高，但推理能耗也更大。

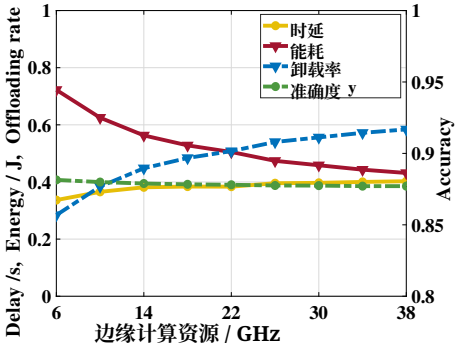
图 8 展示了设备数量、带宽和边缘计算资源不同时的平均时延、能耗、准确率和卸载率。图 8(a) 给出了设备数量不同时的结果。如图 8(a) 所示，当设备数量较少（少于 10 台）时，所有设备都会将任务卸载到边缘服务器（卸载率等于 1）。对于边缘设备，所有时延均来自传输时延和边缘推理时延，所有能耗均来自传输能耗。随着设备数量增加，更多设备共享通信资源和边缘服务器的计算资源，输入帧数量 M_n 随之减少。输入帧数量减少会导致准确率下降。随后，随着 M_n 减少，传输数据量随之减少，传输能耗也随之降低。同时，更多设备带来的竞争会导致时延增加。因此，当设备数量较少（少于 10 台）时，随着设备数量增加，平均时延上升，平均准确率和平均能耗下



(a) 不同设备数量



(b) 不同带宽



(c) 不同边缘计算资源

图 8. 不同设备数量、不同带宽和不同边缘计算资源下的平均时延、能耗、卸载率和准确率。

降。当设备数量超过 10 台时，平均能耗和准确率上升，而平均时延和卸载率逐渐下降。考虑不同的带宽和边缘计算资源，我们绘制了图 8(b) 和图 8(c)。在图 8(b) 和图 8(c) 中，随着带宽和边缘计算资源增加，设备会更倾向于将计算任务卸载到边缘，从而提高卸载率。根据表 I，当 β_1，β_2 和 β_3 固定时，边缘设备的能耗较低、准确率较低、时延较高。边缘设备越多，时延越高、功耗越低。同时，带宽增加时，由于边缘计算资源固定，视频帧数量会减少，以降低边缘计算开销，从而导致准确率下降。边缘计算资源增加时也可得出相同结论。因此，

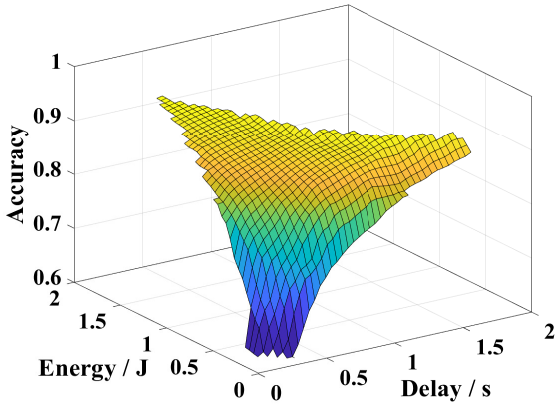


图 9. 时延、能耗与准确率之间的关系。

随着带宽和边缘计算资源的增加，更多边缘设备会导致时延增加、能耗降低且准确率下降。

我们设定输入帧的最小数量为 $M_n^{min} = 1$。我们采用不同的权重 β_1，β_2，β_3，研究平均时延、能耗和准确率之间的权衡关系。约束条件为 $\beta_1 + \beta_2 + \beta_3 = 1$。通过 GP+启发式方案得到权衡曲面的性能。图 9 显示，时延、能耗和准确率相互制约。当时延保持不变时，能耗越高，准确率也越高。换个角度来说，要提高准确率，就必须牺牲时延和能耗方面的性能。此外，在准确率相同的情况下，根据表 I，能耗越高，设备越倾向于在本地执行推理任务，时延也会降低。

VII. 结论

本文研究多用户移动边缘计算系统中基于视频的人工智能推理任务优化问题。我们构建了一个混合整数非线性规划问题，在计算和通信资源约束下，旨在最小化总时延和能耗，并提高总准确率。我们引入基于 MAC 的计算复杂度模型来描述计算时延，并提出一种简单的近似表达式以简化问题。我们还提出一种通用准确率模型，用于刻画识别准确率与输入帧数量之间的关系。随后，我们首先假设卸载决策已知，并将原始问题分解为两个子问题。第一个子问题是优化本地完成深度神经网络推理任务的设备资源。我们推导出该问题的闭式解。第二个子问题是优化将深度神经网络推理任务卸载至边缘服务器的设备资源。我们提出搜索算法和基于几何规划的算法来求解第二个子问题。针对卸载决策优化问题，我们提出了信道感知启发式算法。我们还提出一种基于交替方向乘子法的分布式算法。该算法以可接受的性能损失为代价，降低了计算复杂度。数值仿真和实验结果证明

了所提算法的有效性。我们还详细分析了不同设备数量、带宽和边缘计算资源条件下的时延、能耗和准确率。

附录A 定理1的证明

$\mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}$ 关于 f_n^{md}的偏导数为，

$$\frac{\partial \mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}}{\partial f_n^{md}} = -\beta_1 \frac{\rho C(M_n)}{f_n^{md2}} + 2\beta_2 \kappa \rho C(M_n) f_n^{md}, \tag{47}$$

令 $\frac{\partial \mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}}{\partial f_n^{md}} = 0$，可得，

$$f_n^{md} = \sqrt[3]{(\frac{\beta_1}{2\beta_2 \kappa})}, \tag{48}$$

因此，f_n^{md}在区间（$-\infty$，$\sqrt[3]{(\frac{\beta_1}{2\beta_2\kappa})}$）内单调递减，在区间（$\sqrt[3]{(\frac{\beta_1}{2\beta_2\kappa})}$ $+\infty$）内单调递增。考虑到 f_n^{md}的取值范围，最优解可表示为，

$$f_n^{md*} = \min\{\sqrt[3]{(\frac{\beta_1}{2\beta_2 \kappa})}, f_n^{max}\} \tag{49}$$

接下来分析 M_n。$\mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}$ 关于 M_n 的偏导数为，

$$\frac{\partial \mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}}{\partial M_n} = \frac{\beta_1 \rho m_{c,0}}{f_n^{md}} + \beta_2 \kappa \rho m_{c,0} f_n^{md2} - \frac{\beta_3 m_{a,0}}{(M_n + m_{a,1})^2}, \tag{50}$$

令 $\frac{\partial \mathcal{F}_{\mathcal{P}_{\mathcal{N}_0}}}{\partial M_n} = 0$，可得，

$$M_n = \sqrt{\frac{\beta_3 m_{a,0}}{\frac{\beta_1 \rho m_{c,0}}{f_n^{md}} + \beta_2 \kappa \rho m_{c,0} f_n^{md2}}} - m_{a,1}, \tag{51}$$

考虑到 M_n 的取值范围，最优解可表示为，

$$M_n^* = \min\{\max\{\sqrt{\frac{\beta_3 m_{a,0}}{\frac{\beta_1 \rho m_{c,0}}{f_n^{md}} + \beta_2 \kappa \rho m_{c,0} f_n^{md2}}} - m_{a,1}, M_n^{min}\}, M_n^{max}\} \tag{52}$$

至此，证明完成。

附录B 定理2的证明

根据KKT条件，我们可以得到以下充要条件，

$$\frac{\partial \mathcal{L}_{\mathcal{P}_{\mathcal{N}_1}}}{\partial f_n^{e*}} = -\frac{\beta_1 \rho C(M_n^*)}{f_n^{e*2}} + u_1^* = 0, \ f_n^{e*} > 0, \tag{53}$$

$$\frac{\partial \mathcal{L}_{\mathcal{P}_{\mathcal{N}_1}}}{\partial t_n^*} = -\frac{\beta_1 M_n^* d}{R_n t_n^{*2}} + u_0^* = 0, t_n^* > 0, \tag{54}$$

$$\mu_0^*(\sum_{n \in \mathcal{N}^*} t_n^* - 1) = 0, \tag{55}$$

$$\mu_1^*(\sum_{n \in \mathcal{N}^*} f_n^{e*} - f^{max}) = 0, \tag{56}$$

$$\mu_0^*, \mu_1^* \geq 0. \tag{57}$$

由于 $\frac{\beta_1 \rho C_{(M_n^*)}}{f_n^{e*2}}$和 $\frac{\beta_1 M_n^* d}{R_n t_n^{*2}}$ 均为正数，因此 μ_0^* 和 μ_1^* 也为正数。我们可以得到，

$$\sum_{n \in \mathcal{N}} f_n^{e*} - f^{max} = 0, \tag{58}$$

$$\sum_{n \in \mathcal{N}} t_n^* - 1 = 0, \tag{59}$$

$$f_n^{e*} = \sqrt{\frac{\beta_1 \rho C(M_n^*)}{R_n \mu_1^*}}, \tag{60}$$

$$t_n^* = \sqrt{\frac{\beta_1 M_n^* d}{R_n \mu_0^*}}. \tag{61}$$

结合式（58）和式（60），可得与 M_n^* 对应的 f_n^{e*} 的表达式，

$$f_n^{e*} = \frac{f^{max} \sqrt{C(M_n^*)}}{\sum_{i \in \mathcal{N}_1} \sqrt{C(M_i^*)}}. \tag{62}$$

同理，结合式（59）和式（61），可得与 M_n^* 对应的 t_n^* 的表达式，

$$t_n^* = \frac{\sqrt{\frac{M_n^*}{R_n}}}{\sum_{i \in \mathcal{N}_1} \sqrt{\frac{M_i^*}{R_i}}}, \tag{63}$$

至此，证明完成。

参考文献

[1] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, 和 M. Daneshmand, 《元宇宙综述：最新进展、技术、应用与挑战》, *arXiv*预印本 *arXiv:2111 09673*, 2021。[2] J. Li, L. Deng, Y. Gong, 和 R. Haeb-Umbach, 《抗噪自动语音识别概述》, *IEEE/ACM*音频、语音与语言处理汇刊, 第22卷, 第4期, 第745–777页, 2014年。[3] D. W. Otter, J. R. Medina, 和 J. K. Kalita, 《深度学习在自然语言处理中的应用综述》, *IEEE*神经网络与学习系统汇刊, 第32卷, 第2期, 第604–624页, 2021年。[4] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, 和 M. Shah, 《视觉跟踪：实验综述》, *IEEE*模式分析与机器智能汇刊, 第36卷, 第7期, 第1442–1468页, 2014年。[5] L. N. Huynh, R. K. Balan, 和 Y. Lee, 《Deepsense：一种面向商用移动设备的基于GPU的深度卷积神经网络框架》, 载于*ACM WearSys'16*会议论文集, 2016年, 第25–30页。[6] X. Ran, H. Chen, X. Zhu, Z. Liu, 和 J. Chen, 《Deepdecision：一种用于边缘视频分析的移动深度学习框架》, 载于*IEEE INFOCOM'18*会议论文集, 2018年, 第1421–1429页。[7] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, 和 C. Zhang, 《通过网络瘦身学习高效卷积网络》, 载于*IEEE ICCV'17*会议论文集, 2017年10月, 第2736–2744页。[8] W. Shi, Y. Hou, S. Zhou, Z. Niu, Y. Zhang, 和 L. Geng, 《通过两步剪枝改进深度学习的设备—边缘协同推理》, 载于*IEEE INFOCOM*研讨会*'19*会议论文集, 2019年, 第1–6页。[9] Y. Shi, K. Yang, T. Jiang, J. Zhang, 和 K. B. Letaief, 《通信高效的边缘人工智能：算法与系统》, *IEEE*通信调查与教程, 第22卷, 第4期, 第2167–2191页, 2020年。[10] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, 和 M. Chen, 《边缘内人工智能：通过联邦学习实现移动边缘计算、缓存与通信的智能化》, *IEEE*网络, 第33卷, 第5期, 第156–165页, 2019年。[11] K. B. Letaief, Y. Shi, J. Lu, 和 J. Lu, 《面向6G的边缘人工智能：愿景、使能技术与应用》, *IEEE*通信选题领域期刊, 第40卷, 第1期, 第5–36页, 2022年。

[12] S. R. Sabuj, D. K. P. Asiedu, K.-J. Lee, 和 H.-S. Jo, 《移动边缘计算中的时延优化：认知无人机辅助增强型移动宽带和海量机器类通信服务》，IEEE认知通信与网络汇刊，第8卷，第2期，第1019–1033页，2022年。[13] P. Wang, B. Di, L. Song, 和 N. R. Jennings, 《分布式异构移动边缘计算网络中的多层计算卸载》，IEEE认知通信与网络汇刊，第8卷，第2期，第1301–1315页，2022年。[14] K. Wang, Z. Ding, D. K. C. So, 和 G. K. Karagiannidis, 《采用非正交多址接入的移动边缘计算系统中能耗与时延的斯塔克尔伯格博弈》，IEEE通信汇刊，第69卷，第4期，第2191–2206页，2021年。[15] M. Qin, N. Cheng, Z. Jing, T. Yang, W. Xu, Q. Yang, 和 R. R. Rao, 《移动边缘计算增强型多无线接入技术网络中面向物联网任务部分卸载的服务导向能耗与时延权衡》，IEEE物联网期刊，第8卷，第3期，第1896–1907页，2021年。[16] L. Ale, N. Zhang, X. Fang, X. Chen, S. Wu, 和 L. Li, 《利用深度强化学习实现移动边缘计算中的时延感知与高能效计算卸载》，IEEE认知通信与网络汇刊，第7卷，第3期，第881–892页，2021年。[17] K. Yang, Y. Shi, W. Yu, 和 Z. Ding, 《用于边缘推理的高能效处理与鲁棒无线协作传输》，IEEE物联网期刊，第7卷，第10期，第9456–9470页，2020年。[18] S. Hua, Y. Zhou, K. Yang, Y. Shi, 和 K. Wang, 《面向绿色边缘推理的可重构智能表面》，IEEE绿色通信与网络汇刊，第5卷，第2期，第964–979页，2021年。[19] J. Liu 和 Q. Zhang, 《在移动边缘计算中通过非理想传输提高人工智能驱动的时间关键型服务的可靠性：一项实验研究》，IEEE物联网期刊，第7卷，第10期，第9357–9371页，2020年。[20] W. He, S. Guo, S. Guo, X. Qiu, 和 F. Qi, 《面向物联网中时延敏感型深度学习推理的联合深度神经网络分区部署与资源分配》，IEEE物联网期刊，第7卷，第10期，第9241–9254页，2020年。[21] Z. Lin, S. Bi, 和 Y.-J. A. Zhang, 《移动边缘智能系统中的人工智能服务部署与资源分配优化》，IEEE无线通信汇刊，第20卷，第11期，第7257–7271页，2021年。[22] E. Li, L. Zeng, Z. Zhou, 和 X. Chen, 《边缘人工智能：通过边缘计算按需加速深度神经网络推理》，IEEE无线通信汇刊，第19卷，第1期，第447–457页，2020年。[23] Q. Liu, S. Huang, J. Opadere, 和 T. Han, 《面向移动增强现实的边缘网络编排器》，载于IEEE INFOCOM’18会议论文集，2018年，第756–764页。[24] Y. He, J. Ren, G. Yu, 和 Y. Cai, 《利用卷积神经网络优化移动增强现实系统中的学习性能》，IEEE无线通信汇刊，第19卷，第8期，第5333–5344页，2020年。[25] Y. Zhao, Z. Yang, X. He, X. Cai, X. Miao, 和 Q. Ma, 《Trine：面向家庭应用的云—边缘—设备协同实时视频分析》，IEEE移动计算汇刊，第1–1页，2022年。[26] K. Hara, H. Kataoka, 和 Y. Satoh, 《时空三维卷积神经网络能否追溯2D CNN和ImageNet的历史？》，载于IEEE CVPR’18会议论文集，2018年，第6546–6555页。[27] K. Simonyan 和 A. Zisserman, 《用于大规模图像识别的超深卷积网络》，载于ICLR’15会议论文集，2015年5月，第1–14页。[28] M. Hollemans, 《我的模型有多快？》https://machinethink.net/blog/how-fast-is-my-model/，访问日期：2021年12月30日。[29] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, 和 M. Xiao, 《面向边缘实时视频分析的联合配置自适应与带宽分配》，载于IEEE INFOCOM’20会议论文集，2020年，第257–266页。[30] S. Boyd, S. P. Boyd, 和 L. Vandenberghe, 《凸优化》。剑桥大学出版社，2004年。[31] M. Grant 和 S. Boyd, 《CVX：用于规范凸规划的MATLAB软件，2.1版》，http://cvxr.com/cvx，2014年3月。[32] S. Bi 和 Y. J. Zhang, 《采用二元计算卸载的无线供能移动边缘计算中的计算速率最大化》，IEEE无线通信汇刊，第17卷，第6期，第4177–4190页，2018年。[33] J. Li, X. Li, Y. Bi, 和 J. Ma, 《共生无线电网络中利用可重构智能表面的高能效联合资源分配》，IEEE认知通信与网络汇刊，第1–1页，2022年。[34] V. Sovrasov, 《PyTorch框架中卷积网络的浮点运算次数计数器》，https://github.com/sovrasov/flops-counter.pytorch，访问日期：2021年12月30日。[35] J. Materzynska, G. Berger, I. Bax, 和 R. Memisevic, 《Jester数据集：一个大规模人体手势视频数据集》，载于IEEE ICCV Workshop’19会议论文集，2019年，第2874–2882页。[36] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, 和 A. Zisserman, “Kinetics 人体动作视频数据集”, *arXiv* 预印本 *arXiv:1705.06950*，2017年。[37] X. Chen, Y. Cai, L. Li, M. Zhao, B. Champagne, 和 L. Hanzo, “时延敏感型移动边缘计算的高能效资源分配”, *IEEE Veh. Technol.* 汇刊，第 69 卷，2262，2020年 2，2246 页码 –。