计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(6): 1530−1546, 2025

DAQ: 基于分治策略的自适应 Vision Transformer 低位宽量化方法

吕倩茹　许金伟　姜晶菲　李东升
（并行与分布式计算全国重点实验室（国防科技大学）　长沙　410073）
（lvqianru11@nudt.edu.cn）

DAQ: Divide-and-Conquer Strategy Based Adaptive Low-Bit Quantization Method for Vision Transformer

Lü Qianru, Xu Jinwei, Jiang Jingfei, and Li Dongsheng
（*National Key Laboratory of Parallel and Distributed Computing* (*National University of Defense Technology*), *Changsha* 410073）

Abstract　Vision Transformers (ViTs) have demonstrated remarkable success in computer vision tasks, but their complex architecture and computational demands hinder deployment on edge devices. While post-training quantization (PTQ) is widely adopted for model compression, existing PTQ methods exhibit severe performance degradation in 4-bit ultra-low-bitwidth scenarios. This work systematically addresses two fundamental limitations: 1) spatial mismatch between quantization-sensitive layers (e.g., Softmax) and compute-intensive layers (e.g., linear projections), where quantizing Softmax causes 80% accuracy loss despite contributing merely 8% computational load; 2) non-Gaussian activation distributions with hidden Gaussian-like clustering properties (97% values less than three times z-score). We propose DAQ (divide-and-conquer and adaptive quantization), a hardware-friendly PTQ method. DAQ adpots z-score-driven dynamic partitioning algorithm to separate data into normal-range and abnormal-range groups and quantizes the two groups with connected parameter. DAQ further explores hardware accelerated kernel such as tensor core to speed up quantization ViT models. Experimental results demonstrate that DAQ achieves a maximum improvement of 4.37% in ImageNet Top-1 accuracy under 4-bit quantization. In object detection tasks, its average error margin remains below 0.4% compared with the baseline and achieves a maximum improvement of 8.2%, even surpassing the full-precision model by 0.1% in specific cases, thereby realizing near-lossless low-bit-width quantization. Through hardware implementation optimization, DAQ achieves 43%~86% computational acceleration without significantly increasing computational overhead. This approach establishes a synergistic algorithm-hardware co-optimized quantization deployment paradigm for resource-constrained scenarios, effectively balancing model efficiency and precision retention.

Key words　vision Transformer (ViT); post-training quantization (PTQ); outlier; low-bit quantization; z-score; uniform correlation quantization

摘　要　视觉 Transformer（Vision Transformer, ViT）模型在计算机视觉领域的多项任务中取得显著效果. 但 ViT 的复杂结构和计算开销限制了其在边缘计算设备中的部署. 训练后量化（post-training quantization,

收稿日期: 2025-03-01; **修回日期**: 2025-04-10
基金项目: 国家自然科学基金项目(62025208, 62421002, 62472435, 62172430); 湖南省科技创新计划项目(2022RC3065); PDL 实验室基金项目(2023-JKWPDL-02)
This work was supported by the National Natural Science Foundation of China (62025208, 62421002, 62472435, 62172430), the Science and Technology Innovation Program of Hunan Province (2022RC3065) , and the National Key Laboratory of Parallel and Distributed Computing Foundation (2023-JKWPDL-02).
通信作者: 许金伟(xujinwei13@nudt.edu.cn)

PTQ）技术被广泛应用于 ViT 模型轻量化中以解决实际部署难题，但现有 PTQ 方法在低位宽量化中的性能损失较大. 针对低比特量化场景，ViT 的量化敏感层（如 Softmax）与计算密集层（如线性变换）存在显著空间错位，且非高斯分布的激活值中隐含 97% 的类高斯聚集特性. 由此，基于标准分数 z-score 方法提出分治自适应量化（divide-and-conquer and adaptive quantization, DAQ）方法，通过量化敏感度-计算-存储开销联合分析与硬件协同设计，实现精度与效率的联合优化. DAQ 构建动态分治量化机制，通过动态感知的 z-score 方法实现正常值/离群值双域分割，均匀关联量化 2 个值域. 在 4-bit 量化下，DAQ 方法在分类任务上的 Top-1 精度最大提升 4.37 个百分点，目标检测任务最大精度提升达 8.2 个百分点，与基线模型相比误差平均低于 0.4 个百分点，超过最佳全精度模型 0.1 个百分点，接近实现无损的低位宽量化. 另一方面，DAQ 在硬件兼容设上适配 Tensor Core 的 INT4/INT8 内核，以量化定点计算来减轻线性计算压力. 实验表明，DAQ 硬件适配后对线性计算部分有 43%~86% 的加速效果，为资源受限场景提供了算法-硬件协同优化的量化部署范式.

关键词 视觉 Transformer（ViT）；训练后量化（PTQ）；离群值；低比特量化；z-score；均匀关联量化

中图法分类号 TP311.13；TP309

DOI: 10.7544/issn1000-1239.202550145 **CSTR:** 32373.14.issn1000-1239.202550145

自 Vaswani 等人[1]提出 Transformer 架构以来，其核心组件——自注意力（self-attention）机制通过动态长程依赖建模能力，成功突破了传统卷积神经网络（CNN）在感受野限制上的桎梏. 这一突破催生了视觉 Transformer（vision Transformer, ViT）模型[2]及其变体，并在计算机视觉领域引发范式变革. DeiT[3], Swin Transformer[4]等模型在图像分类领域超越 ResNet 系列[5], DETR[6]重构目标检测范式，MaskFormer[7]推动图像语义分割问题上的精度突破，ViT-GAN[8]在图像合成领任务中展现细粒度控制能力，DiT[9]增强了图像生成任务中的广度. 然而，其出色的性能背后是高昂的计算开销和内存占用. ViT-L 模型参数量达 307×10^6，单张 224×224 图像推理需 190.7 GFLOPs[10]. 这与其在资源受限的边缘设备[11]上的部署需求形成显著矛盾.

为缓解 ViT 的硬件部署压力，模型量化（model quantization）是模型压缩的核心技术之一. 模型量化将 32 位浮点精度（FP32）权重或激活值映射至低位宽（如 8/4-bit）整数空间表示，有效降低模型大小. 同时利用低位宽定点运算代替高精度浮点运算以加速模型推理性能. 训练后量化（post-training quantization, PTQ）[10-15]是一类高效实用的模型量化方法. PTQ 方法基于统计先验的轻量化校准，仅需少量无标签校准数据甚至不需要校准数据即可完成模型量化，其免训练特性能够适配边缘设备实时部署需求.

多种 PTQ 方法被用于 ViT 量化[10-15]. 针对 ViT 中的特定层的输出激活，采用复杂高阶函数[14]、Hessian 矩阵指导的度量方法[12]等进行量化指导和补偿，在 8/6-bit 量化中取得了成效. 然而，将现有 PTQ 方法应用于低比特（如 4-bit）量化中时仍然面临不可忽视的挑战：

1）低位宽表示引起模型性能衰减. ViT 中激活值的高维特征空间存在显著长尾分布现象，将大量数据聚合到少量点位带来的的量化损失无法通过反量化补偿.

2）量化策略与计算特性的错位匹配. ViT 架构，如 ViT-Base，线性层 FFN, QKV Gen, Proj 等占据了 80% 以上的计算开销，其计算密度是 LayerNorm（LN）的 98 倍、Softmax 的 294 倍，GELU 的 45 倍[16]. 然而，现有研究更多地强调非线性层的特殊分布和激活量化，忽视了 ViT 自身的计算特性和资源利用特性，缺乏量化算法与计算加速相结合的全局性视角.

3）硬件适配性缺失. 现有 PTQ 量化算法与底层硬件架构的协同设计存在显著差距. 尽管多数方法宣称硬件友好性，但出于模型精度考虑，仍采用计算前反量化、计算时全精度的计算范式，缺乏对量化定点计算的直接利用.

针对上述 3 方面的挑战，本文从“量化-计算联合优化”出发，针对 ViT 低位宽量化场景展开系统性研究. 通过分析量化敏感度并借助 z-score 方法，揭示了低比特量化误差与计算效率间的耦合作用机制，并发现其与异常激活特征及概率分布特性的内在关联规律：

1）量化-计算的非一致性. 量化导致性能衰减强的部分并不意味着其也是高计算/存储开销部分. 这表明即使对该部分设计了精细的量化算法，量化带来的计算收益和存储收益都可能相当有限.

2)高量化敏感度数据均表现出显著离群值(outlier)现象,同时,实验中还观察到离群值的普遍存在.这表明低位宽量化的关键在于高动态范围的离群值表征与低位宽约束间的本质冲突,而非单纯源于数据的长尾分布或高斯分布本身.

3)数值分布中存在的普适约束性.尽管 ViT 激活分布中呈现明显的非高斯特性(长尾分布),其数值范围仍然保持了高度集中性,即至少 97% 的数据呈现出聚集现象.

基于上述发现,本文提出面向 ViT 的 PTQ 自适应分治量化(divide-and-conquer and adaptive quantization, DAQ)方法.DAQ 在保持硬件兼容性的前提下实现 4-bit 量化的性能突破,主要贡献有 3 点:

1)针对量化-计算/存储开销联合分析,揭示 ViT 系列模型中的 2 个关键规律:一是计算密集层与量化敏感层存在显著空间错位(如 Softmax 层量化可带来 80% 的性能下降却仅占 8% 的计算负载);二是借助 z-score 方法发现非高斯分布中隐藏数值集中特性(97% 激活值分布仍呈类高斯分布的聚集性),统一数据定量分析标准,突破传统高斯假设局限.

2) 基于贡献 1)提出了针对 ViT 的 4-bit PTQ 量化方法 DAQ.DAQ 支持融合分治策略与自适应参数调节:分治策略基于数据聚集性,利用动态感知的 z-score 方法将数据划分为正常值与离群值,自适应参数调节划分范围以获得更高量化性能.基于量化-计算瓶颈分析的结论,DAQ 以层归一化(layers normalization, LN)和 GELU 输出激活量化出发,形成了普适性的量化方法.DAQ 在不同模型、不同量化对象的实验中都表现出 SOTA 性能,获得了最大达 4.37% 的精度提升,目标检测任务上 4-bit 量化模型的平均精度损失低至 0.4%.

3)DAQ 充分考虑量化-计算协同设计,提出计算友好的均匀关联量化方法,同时依靠加速器硬件架构如 GPU Tensor Core 等加速内核,实现低位宽定点计算代替高位宽浮点运算,实验表明,DAQ 未引起明显性能降低,对线性计算有 43%~86% 的加速效果,与 SOTA 算法相比较,端到端性能最大提升 41.8%.

1 相关工作

PTQ 作为深度学习模型压缩领域的主流范式,因其无需微调的特性而具备显著的操作便捷性.缺乏训练阶段的误差补偿机制也导致 PTQ 方法在低位宽(如 4-bit)动态激活值量化场景中面临严峻的精度保持挑战.针对 ViT 架构的激活量化研究,现有方法可根据量化对象划分为以下 3 类:Softmax 输出激活(post-softmax activation)量化、GELU 输出激活(post-GELU activation)量化和 LN 输出激活(post-LN activation)量化.

针对 Softmax 输出激活,FQ-ViT[10] 采用非均匀的 log2 量化方法,为其中的高频小值分配更多更精细的量化点以获得更小的量化误差.PTQ4ViT[11] 引入的孪生均匀量化(twin uniform quantization)方法将 Softmax 输出激活分成 2 个独立的量化区间以实现更为细致的量化过程,其由 2 个唯一但不独立的缩放因子控制,2 个缩放因子之间具有 2^k 倍的关联性;APQ-ViT[13] 利用 Softmax 函数的马修效应(Matthew effect),实现了一种非对称线性量化方法以改善量化误差的分布.RepQ-ViT[14], RepQuant[15] 先采用 $\log\sqrt{2}$ 量化以更好地拟合具有幂律分布的注意力分数,随后对缩放因子重参数化以使其基数为 2,从而在推理中实现以快捷移位计算代替反量化.TSPTQ-ViT[17] 利用非正态分布值中的位稀疏性为 Softmax 输出激活的不同区域分配不同的比例因子以获得更小量化误差.

对于 GELU 输出激活,PTQ4ViT[11] 仍然采用了与 Softmax 输出激活相一致但将区间划分改为由符号决定的孪生均匀量化方法.TSPTQ-ViT[17] 也采用了类似的双区域策略,并针对 GELU 输出激活值中的符号位适应性处理.

对 LN 输出激活,FQ-ViT[10] 提出 PTF(power-of-two factor)量化方法.作为一种逐通道量化方法的变形,PTF 按通道尺度分配量化调节因子 α,缩放因子取 $2^{\alpha}s$ 而非 s.RepQ-ViT[14] 针对 LN 输出采用逐通道量化后再调整量化参数缩放因子 s 和零点偏移 z,同时修改了 LN 对应的权重参数.RepQuant[15] 扩展了逐通道双边阶段策略以实现更为准确的量化,最大限度减少量化空间内的偏差.TSPTQ-ViT[17] 则利用 K-means 算法筛选离群值后,选择一组线性正常值基于 Hessian-Guide 方法求解缩放因子,该过程可与权重的缩放因子联合优化以获得更佳性能.

当前基于 PTQ 的 ViT 量化研究存在 3 个方面的局限:1)低位宽(4-bit)量化性能下降严重;2)量化算法泛化性不足,现有方法普遍针对特定分布和层量化,过分精细地针对性量化设计而导致方法扩展性和适应性不佳;3)硬件部署低效.尽管多数方法宣称硬件友好性,但量化/反量化开销实际上降低了计算效率,同时量化方法也未充分利用新一代 AI 加速器

特性等，缺乏直接利用量化计算加速的相关实践.

2 预备知识

2.1 ViT

ViT[2] 是基于 Transformer 编码器结构的图像分类模型. ViT 将原本 CNN 中的卷积结构用 Transformer 替换，实现了 Transformer 架构从自然语言处理（natural language processing, NLP）领域到计算机视觉（computer vision, CV）领域的迁移. 为了进一步提高性能，ViT 的变体如 DeiT[3] 和 Swin [4] 相继被提出，并在图像分类[1,18-19]、目标检测[6,20-21]、语义分割[22-23]、图文生成[24]等计算机视觉任务中取得了巨大的成功.

图 1 展示了 ViT 模型的组成结构. ViT 由图像块嵌入（patch embedding）和 L 个 Transformer 编码器模块（block）构成. 具体而言，ViT 首先将输入大小为 H(高度)×W(宽度)×C(通道数)的图像分割为 N 个大小为 $P\times P$ 的非重叠图像块（patches），经过线性嵌入将每个图像块展平为一维向量后线性投影到 d 维嵌入空间中，得到图像输入张量 $\boldsymbol{X}\in\mathbb{R}^{N\times d}$ 并参与到一系列 Transformer 编码器模块的计算中. 每个编码器模块由一个多头自注意（multi self-attention, MSA）子模块和一个前馈神经网络（feed-forward neural network, FFN）子模块组成，每个子模块后添加残差连接和 LN.

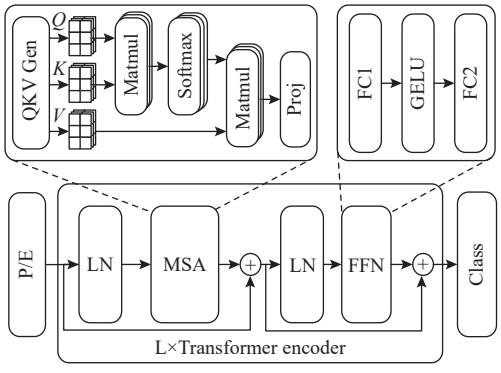


Fig. 1 Illustration of ViT structure

图 1 ViT 结构示意图

编码器模块的计算过程可以用公式统一描述为：

$$\boldsymbol{Y}_l = MSA(LN(\boldsymbol{X}_{l-1})) + \boldsymbol{X}_{l-1}, \tag{1}$$

$$\boldsymbol{X}_l = FFN(LN(\boldsymbol{Y}_l)) + \boldsymbol{Y}_l, \tag{2}$$

其中 $l=1,2,\cdots,L$ 表示编码器模块编号索引. 如图 1 所示，MSA 子模块通过多头自注意力机制学习不同图像块间的关联性：

$$\left[\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i\right] = QKVGen(\boldsymbol{X}') = \boldsymbol{X}'\boldsymbol{W}^{QKV} + \boldsymbol{b}^{QKV}x, \tag{3}$$

$$Attn_i = Softmax\left(\boldsymbol{Q}_i\boldsymbol{K}_i^{\mathrm{T}}/\sqrt{d_h}\right)\boldsymbol{V}_i, \tag{4}$$

$$MSA(\boldsymbol{X}') = Proj(Attn_1,\cdots,Attn_h) = Attn\boldsymbol{W}^p + \boldsymbol{b}^p, \tag{5}$$

其中 $i=1,\cdots,h$，h 表示自注意力机制中的多头数，$\boldsymbol{W}^{QKV}\in\mathbb{R}^{d\times 3d_h}$，$\boldsymbol{b}^{QKV}\in\mathbb{R}^{3d_h}$，$\boldsymbol{W}^p\in\mathbb{R}^{hd_h\times d}$，$\boldsymbol{b}^p\in\mathbb{R}^d$. QKV Gen 基于输入激活 $\boldsymbol{X}'$ 计算 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 矩阵，Proj 投影层通过线性计算得到 MSA 的输出结果.

FFN 模块将特征投影到更高维度来学习特征的不同表达. 设 FFN 的输入为 $\boldsymbol{Y}'$，FFN 的计算过程为：

$$FFN(\boldsymbol{Y}') = GELU\left(\boldsymbol{Y}'\boldsymbol{W}^{\mathrm{FC1}} + \boldsymbol{b}^{\mathrm{FC1}}\right)\boldsymbol{W}^{\mathrm{FC2}} + \boldsymbol{b}^{\mathrm{FC2}}. \tag{6}$$

Transformer 结构的二次复杂性来源于自注意力机制中的 Softmax 过程. $\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}$ 的计算复杂度为 $O(N^2d)$，$\boldsymbol{Q},\boldsymbol{K}\in\mathbb{R}^{N\times d}$，因而自注意力机制的计算复杂度随着输入维度 N 的增长而二次增加. 在大语言模型（large language model, LLM）中，输入长度和上下文长度一般远超过隐藏维度 d，大量计算开销会集中于自注意力机制. 但 ViT 与 LLM 有一个显著区别：当输入图像分辨率不变时，ViT 划分输入图像为子图时采用的划分数量 N 一般也固定不变. 因此，ViT 模型的计算瓶颈与 LLM 模型的计算瓶颈并不完全一致.

2.2 量 化

模型量化是模型压缩领域的关键技术之一. 其核心目标是将高位宽浮点数表示的权重（weights）和/或激活值（activations）用较低位宽定点数表示，从而显著减少内存占用和带宽需求. 同时，通过用低位定点运算代替高位浮点运算，有望提升计算吞吐率并降低能耗，加速模型推理过程.

2.2.1 量化对象

需要量化的目标张量可以分为权重量化和权重-激活量化.

权重量化仅量化模型静态权重，减少模型权重存储开销，量化过程通常不需要或者只需要小部分校准数据集参与. 但仅量化权重通常无法获得计算性能的提升，反而由于需要在推理过程中保持计算精度而增加反量化开销.

权重-激活量化则是同时量化静态权重和动态激活. 激活值由模型推理过程动态产生，因而需要校正数据集来辅助预测激活值的范围并设计相应的量化方法. 权重-激活量化能带来存储和计算 2 个维度上的优势.

DAQ 主要针对激活量化，但可以和其他权重量化方法结合，实现权值-激活量化.

2.2.2 量化粒度

根据共享量化参数的范围，量化粒度由粗到细

可以分为逐层量化、逐组量化和逐通道(channel)/Token量化3类.

逐层量化以一个层的输出激活值或整层的权重为单位,特征张量共用一组缩放因子(scale factor)s和零点(zero point)偏移z.

逐组量化是对目标张量进行分组,以组(group)为单位,组内共享缩放因子s和零点偏移z,组间的量化参数可以不同.当组等于1时,逐组量化与逐层量化等价;当以通道或者Token为分组单位时,逐组量化就是逐通道/Token量化.

当量化粒度越细,量化参数包含的补偿信息越多,量化误差越小,但量化过程越复杂,量化和量化后相关的计算效率越低.因此逐层量化因参数简单而硬件友好度最高,本文中DAQ采用粗粒度逐层量化.

2.2.3 均匀量化和非均匀量化

根据量化方法中的取值域是否等间隔,量化可以分为均匀量化(uniform quantization)和非均匀量化(non-uniform quantization).

均匀量化因其数学形式简洁且硬件友好而成为工业界应用最广泛的量化方案.其核心思想是将实数域映射到等间隔离散量化电平上,数学形式定义为:

$$Q(x;b,s,z)=clip\left(\left\lceil\frac{x-z}{s}\right\rfloor,0,2^b-1\right),\tag{7}$$

其中$x\in\mathbb{R}$为待量化数据,$b\in\mathbb{Z}^+$表示量化位宽,$z\in\mathbb{R}$为零点偏移,$s\in\mathbb{R}^+$为缩放因子.$\lceil\cdot\rfloor$表示近似取整的数学函数,可以是四舍五入、向上/向下取整等.*clip*函数为截断函数.

$$clip(x;a,b)=\begin{cases}a, & \text{if } x<a,\\ x, & \text{if } a\leqslant x\leqslant b,\\ b, & \text{if } x>b.\end{cases}\tag{8}$$

均匀量化的反量化过程为:

$$DeQ(\hat{x};s,z)=(\hat{x}+z)s.$$

均匀量化参数计算高效,仅需确定z与s即可完成全域映射,时间复杂度为$O(N)$.同时等间隔特性也使利用量化数据进行计算更加简便.但另一方面,单一均匀量化对长尾分布数据的适应性较差,易导致尾部区域的量化误差累积.

非均匀量化作为处理非对称数据分布的关键技术之一,对具有显著长尾特征的数据建模更为精确.研究者提出了多种非线性量化函数如指数量化[10]、对数量化[16]等.非均匀量化的难点在于计算复杂度高以及难以找到合适的非均匀量化映射函数.

2.2.4 量化与模型训练

基于量化过程与模型训练的耦合程度,现有方法可分为量化感知训练(quantization-aware training, QAT)和训练后量化(post-training quantization, PTQ).

QAT通过将量化噪声注入前向传播过程,联合优化模型参数与量化器超参数(如缩放因子s、零点偏移z).尽管QAT能通过端到端微调获得更高的量化模型性能,但其重训练过程计算开销高昂.另一方面,QAT依赖大规模数据进行微调,微调过程本身也可能导致模型无法收敛,也不适用于小样本数据集或样本获取困难的场景.

PTQ则提供了一种无需训练的方法(或用于校准目的的最小训练成本),以实现快速有效的量化.同时,PTQ可直接生成符合INT8/FP16等工业标准的量化模型,硬件兼容性更高.低计算成本、易部署的特性使得PTQ方法在资源受限的边缘计算场景中展现出更优的工程可行性.

本文研究专注于PTQ低位宽(4-bit)的量化实现.

3 模型分析

ViT低位宽量化的核心挑战在于动态激活值的高效量化.为了充分利用ViTs中的数据特性,本节将分析ViT激活量化瓶颈及量化计算效率,挖掘低位宽量化引起性能衰减的原因,探索量化有效目标,指导低位宽量化方法的设计.

3.1 低位宽量化瓶颈分析

为了定位ViT模型低位宽量化的瓶颈所在,本节将基于ViT-S架构开展分层量化敏感性分析,采用Min-Max均匀量化算法[16],针对不同功能层LN、Softmax、FFN、GELU等的输出激活值实施6/5/4比特低位宽量化,评估单一模块量化对模型性能的影响(ImageNet数据集上Top-1指标变化),量化当前模块的输出等价于量化下一模块的输入.测试结果如图2所示.为便于分析,图2中的模块量化精度顺序由高到低排序,而非按照模块中计算顺序LN、QKV Gen、Softmax、Proj、FC1、GELU、FC2进行排序.

图2表明,不同层对低位宽量化误差敏感度不同,从而导致模型实际性能下降有所区别.MSA模块中的QKV Gen和Proj线性层的输出激活,随着量化位宽精度的下降而引起的模型精度衰减并不明显;但当量化位宽低于某个阈值,比如5-bit时,LN和FC1的量化误差传递给下一层后都出现了较为明显的精度下降,下降25%~35%;而GELU、Softmax和FC2层输出激活量化与模型的性能损失呈强正相关性,即随着量化位宽逐渐减小,模型精度迅速衰减.当采用

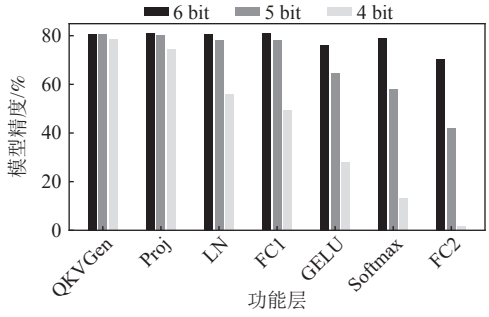


Fig. 2 Model accuracy under various quantization widths in ViT-S

图 2 ViT-S 不同量化位宽下的模型精度

4-bit 量化 FC2 输出时，量化误差造成模型性能崩溃.

为了进一步研究部分层输出激活明显受量化位宽影响的原因，图 3 分析了其输出激活的分布状态. 其中，LN2(第 2 个 LN 的输出)，FC1 和 FC2 的输出激活都服从或近似服从正态分布，但明显存在高动态范围的离群值；非线性层 GELU 和 Softmax 的输出呈现显著的长尾特性，其离群值亦导致特征张量的动态范围呈指数级扩展.

因此，实验表明量化性能下降的核心矛盾在于异常值表征与低位宽约束间的本质冲突，而非单纯源于数据的长尾分布或高斯分布本身.

图 4 为将离群值研究扩展到其他 ViT 系列模型中的结果，统计了 ViT 及其变体模型推理过程中所有激活张量的极大值与极小值. 图 4 显示离群值现象在 ViT 模型中保持跨模型的一致性和显著性. 这与 LLM 中离群值仅在模型规模较大(≥7 B 参数)时才观察到的现象[25] 不同，ViT 模型即使在微型架构(如 ViT-Tiny，5×10^6 的参数量)也表现出显著的激活离群值现象. 而且，Llama 7B[26] 中的离群值变化幅度(9.7)[15] 远低于 ViT($10^2\sim10^3$).

跨模型尺度的离群值泛在现象进一步表明，传统均匀量化方法下，低位宽(4-bit 及以下)的有限表征能力将导致显著的离群值信息损失，进而损害模型性能.

为量化分析 ViT 中的离群值，本文利用统计学方法中的标准分数 z-score 来评估数据的偏离程度以判定离群值. z-score 的计算方法为：

$$Z(x)=\frac{x-\mu}{\sigma}, \tag{9}$$

其中，μ，σ分别为原始数据的均值和标准差. 以 z-score 的绝对值作为离群值的判断指标且参考概率统计学意义上的高 z-score，图 5 展示了 ViT-S/DeiT-B/Swin-S 推理过程中产生的每个层激活的高 z-score(>3 或>6)的占比情况，离群值占比最高约 3%.

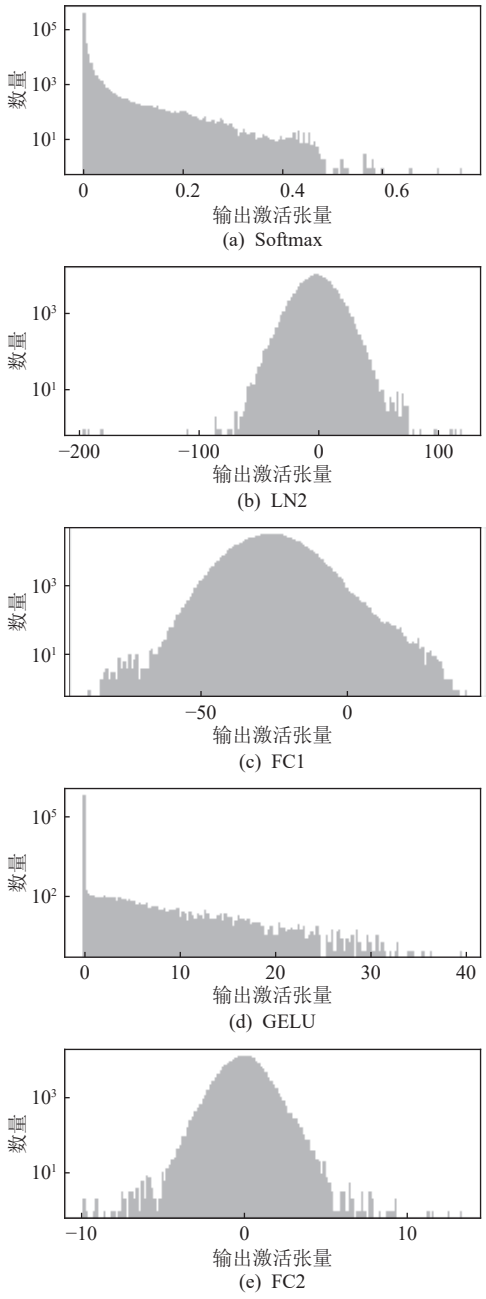


Fig. 3 Data distribution of partial activations in ViT-S, Block 8

图 3 ViT-S, Block 8 部分激活值数据分布

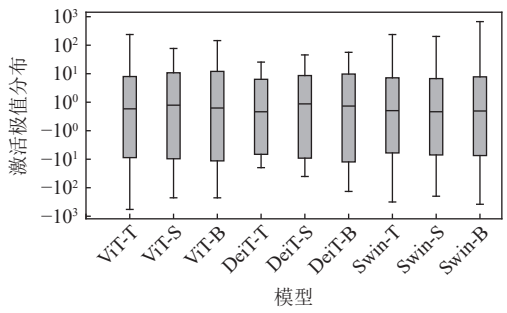


Fig. 4 Extreme value distribution of activation tensor in ViTs reasoning

图 4 ViTs 推理时激活张量的极值分布

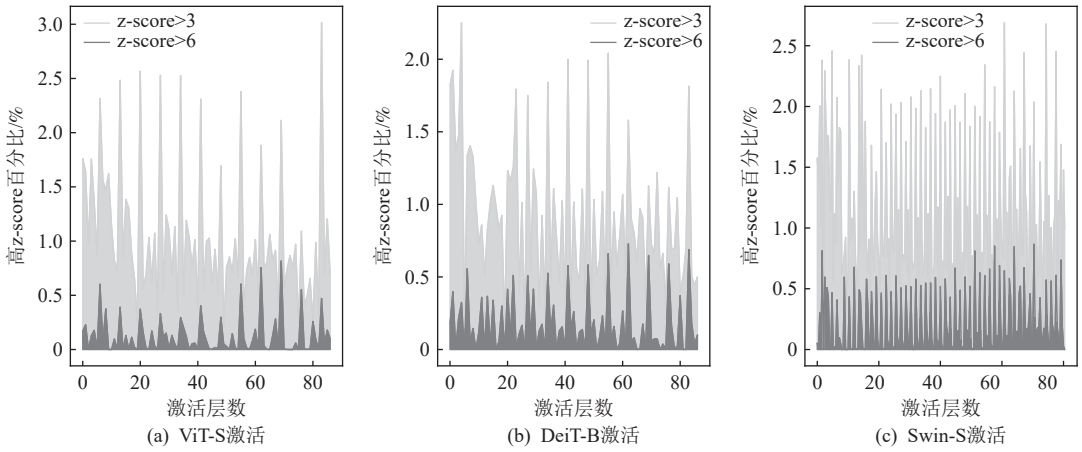


Fig. 5 Proportion of high z-score in ViT-S/DeiT-B/Swin-S activations

图 5 ViT-S/DeiT-B/Swin-S 激活值中高 z-score 占比

图 5 也从另一方面揭示了 ViTs 中数值分布的集中性规律，即至少 97% 以上的激活值在标准化空间内呈现紧密聚集特性，而且这种聚集性并不因激活数值分布的不同而改变.

ViT 量化中的离群值低位宽表征以及数据聚集性启发本文建立离群值驱动的分段量化，即基于激活值呈现的“集中正常值+稀疏离群值”，DAQ 采用分治量化策略，对正常值和离群值用均匀量化但关联的量化参数，既避免全局量化的模型性能下降，又能保持整体计算效率.

3.2 量化-计算分析

量化的另一个潜在优势是利用低位宽定点计算替代高位宽浮点运算从而获得计算加速. 量化计算密集运算层可借助低位宽整型计算降低运算强度.

本文研究在 Nvidia RTX 3090[27] 硬件平台上，从 ImageNet-1K 测试集[28] 中随机选取 100 张图片在 ViT-S 模型上进行前向推理，测试各计算层时延和平均参数开销占比及 4-bit 量化模型精度损失百分比，结果如表 1 所示.

表 1 从计算、存储、量化精度 3 方面研究了量化-计算开销错位的情形. 实验结果表明，仅关注数据的特殊分布而设计的量化方法在量化效能上可能是次优甚至低效的. 量化方法需要与计算实践相结合，才能有效获得存储\计算性能上的提升.

如表 1 所示，尽管 Softmax 激活输出呈现量化位宽高度敏感性(模型精度衰减达 84%)，但其关联层 Proj 的计算平均开销和存储开销分别仅占 9% 和 8%. 这说明即使采用细粒度量化 Softmax 输出激活以保持模型精度，但对模型压缩和推理加速方面的贡献收益仍然十分有限.

相较而言，LN 与 GELU 的激活输出作为 QKV 生成器和全连接层(FC1/FC2)的核心输入，其在存储占用和计算加速双维度均呈现显著优化潜力(二者计算延时和存储占比分别高达 87% 和 75%). 其他如 QKV Gen 和 Proj 输出激活的量化对模型性能和计算能效的提升并不显著，模型精度保持也较好.

Table 1 Proportion Analysis of Accuracy Calculation/Storage/Reduction of ViT-S layer

表 1 ViT-S 层精度计算/存储/下降占比分析

层	计算平均时延占比/%	参数+激活存储占比/%	量化精度下降/%
LN	6	5	32
QKV Gen	21	17	4
Softmax	2	8	84
Proj	9	8	9
FC1	27	22	40
GELU	4	10	65
FC2	**31**	**30**	**98**

注：黑体数值表示最大值，下划线数值表示最小值.

根据表 1 的综合分析，本文将针对 LN 和 GELU 的输出激活作为关键量化对象，充分考虑量化方法本身的复杂性和量化后数据的硬件友好性，提出了基于 z-score 方法的关联量化方法 DAQ.

值得说明的是，虽然本文研究的量化算法关注 LN 和 GELU 的输出激活，但 DAQ 本质上提供了一种抽象的处理模型以应对 ViT 量化中的不同数据分布. 因此，DAQ 不仅对除 LN 和 GELU 之外的层的激活量化仍然适用，也支持 ViT 中的呈典型高斯分布的权重量化，无疑增加了 DAQ 方法的统一性和普适性.

4 DAQ: 基于 z-score 的自适应分治量化方法

DAQ 采用了分治-量化的思路，以 z-score 方法作

为分治基准，结合硬件友好的均匀关联量化算法. 图 6 描述了 DAQ 量化流程. 虽然 DAQ 从量化能效最大化的角度出发选择 LN 和 GELU 输出激活作为关键量化对象，但 DAQ 方法对其他计算模块（如图 6 左侧 ViT 模块中部分填充的层）同样适用.

DAQ 经由 z-score 将输入数据划分为正常值集 X_1 与异常值集 X_2：

$$X_2=\{x_i\,|\,|Z(x_i)|>\tau\},\ \tau\in\mathbb{R}^+. \tag{10}$$

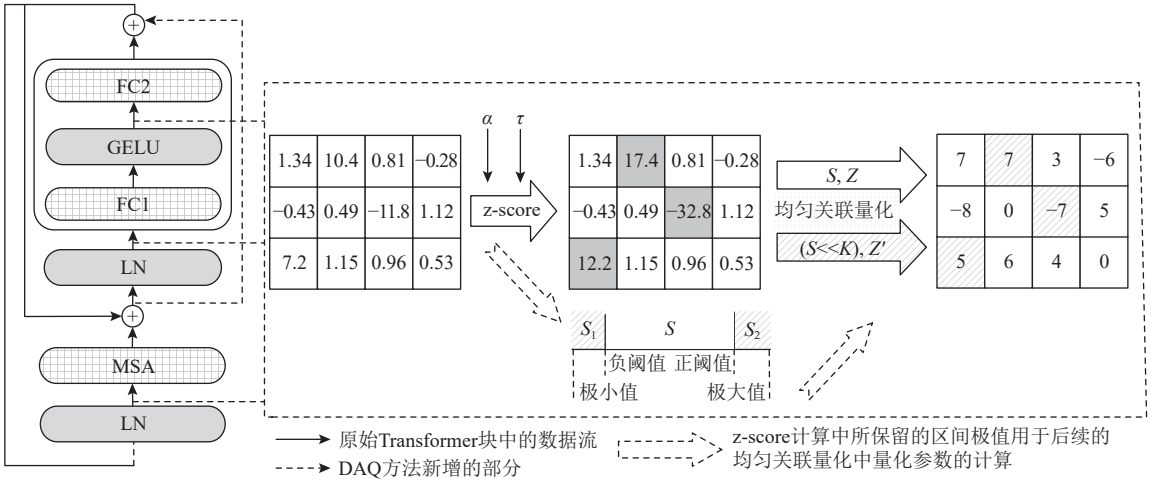


Fig. 6 Illustration of DAQ quantization process

图 6 DAQ 量化流程示意图

ViT 激活中隐含的跨分布的数据集中性使 z-score 方法突破了传统方法对先验分布假设的依赖，实现了跨分布的离群值检测方法. 尽管 z-score 方法简洁易行、计算高效（时间复杂度 $O(n)$），但其在实际应用中存在 3 个问题：

1）统计量估计的冗余存储/计算开销. z-score 方法中隐式使用了数据的均值 μ 和标准差 σ. 求取数据全局标准差涉及到数据的二次访问和串行计算开销，加重了资源受限的设备的存储和访存压力.

2）单一静态全局阈值不匹配高动态变化激活值域. 当 ViTs 中的激活张量呈现出显著非高斯特性时，特别是长尾分布场景下，单一静态全局阈值难以适应其特征空间的动态统计特性.

3）数据规整性破坏引起的存储效率降低. 对正常/离群值集采用混合精度量化可能导致特征张量在内存中的非对齐存储，降低访存效率.

针对问题 1），4.1 节提出了利用数据极值估计标准差的方法，离线求解估测系数 α，简化标准差的计算过程，减少数据二次访问. 针对问题 2），4.2 节提出动态分布阈值 τ 感知算法：通过离线校正阶段的代表性样本分布特征，对具有不同分布特征的数据动态自适应调节阈值 τ. 同时，问题 1）和问题 2）中的参数可以通过联合优化进一步提升算法性能和效率. 4.3 节提出的均匀关联量化用于解决问题 3）. 该量化方法保持数据存储的地址对齐特性，实现数据的高效存取.

为了便于后续行文表达，表 2 对 5 个常用的重要变量进行说明.

Table 2 Meanings of Variables

表 2 变量含义

变量名称	含义
$\boldsymbol{X}$	目标特征张量
$\boldsymbol{X}_1$	目标特征张量中的正常值集
$\boldsymbol{X}_2$	目标特征张量中的离群值集
$\hat{\boldsymbol{X}}$	$\boldsymbol{X}$ 的量化表示
$\tilde{\boldsymbol{X}}$	$\boldsymbol{X}$ 的反量化结果

4.1 标准差估计方法

为了减轻计算统计量均值 μ 和标准差 σ 带来的计算/访存增加，同时避免引入高昂的除法代价[26]，DAQ 提出利用极差估计标准差的自适应方法. 这个方法的有效性来源于 z-score 方法允许标准差存在可接受的精度误差，如 $10^{-2}\sim10^{-3}$，而不影响筛选结果.

通过均值 μ、P 个极值和估测系数 α 可以近似计算标准差 σ：

$$\sigma\approx\sqrt{\left(\sum_{i=1}^{P}(M_i-\mu)^2+\alpha L\right)/L}. \tag{11}$$

下面对式（10）的理论有效性进行推导. 为了便于说明，设待量化 $\boldsymbol{X}=\boldsymbol{X}_1\cup\boldsymbol{X}_2$. 如 3.1 节中观察到的数

值集中分布特性, 正常值$\boldsymbol{X}_1$占比高达 97% 以上, 离群值$\boldsymbol{X}_2$仅占少量.

待量化$\boldsymbol{X}$的总体标准差σ满足:

$$L\sigma^2=\sum_{i=1}^{L}(x_i-\mu)^2=\sum_{x\in\boldsymbol{X}_1}(x-\mu)^2+\sum_{x\in\boldsymbol{X}_2}(x-\mu)^2, \tag{12}$$

其中L为待量化$\boldsymbol{X}$的数据总量.

当$x\in\boldsymbol{X}_2$当时, 不妨设$x=x_1+x_2, x_2\in\boldsymbol{X}_2, x_1\in\boldsymbol{X}_1\cup\{0,\mu\}$. 且我们将$\boldsymbol{X}_2$视作服从正态分布, 式(12)可以表示为:

$$\begin{aligned}L\sigma^2=&\sum_{x\in\boldsymbol{X}_1}(x-\mu)^2+\sum_{x_1\in\boldsymbol{X}_2}((x_1-\mu)+(x_2-\mu))^2\approx\\&\sum_{x\in\boldsymbol{X}}(x-\mu)^2+\sum_{x_1\in\boldsymbol{X}_2}(x_1-\mu)^2=L\hat{\sigma}^2+\sum_{x_1\in\boldsymbol{X}_2}(x_1-\mu)^2.\end{aligned} \tag{13}$$

进一步地, 用极值表示关于$\boldsymbol{X}_2$的和. 假设取绝对值最大的P个数据, 那么有:

$$\sum_{x_1\in\boldsymbol{X}_2}(x_1-\mu)^2=\sum_{i=1}^{P}(topk(\boldsymbol{X}_2)[i]-\mu)^2+\delta, x_1\in\boldsymbol{X}_2. \tag{14}$$

δ表示用P个极值表示后的值与原式的值之差. 将δ记为$\delta=\gamma L$, 那么可以得到最后的计算结果:

$$L\sigma^2=\left(\hat{\sigma}^2+\gamma\right)L+\sum_{i=1}^{P}(topk(\boldsymbol{X}_2)[i]-\mu)^2. \tag{15}$$

至此, 我们得到了用P个极值近似估计标准差的方法. 将$\alpha=(\hat{\sigma}^2+\gamma)$视作变量, 通过自适应算法求解$\alpha$, 就能得到不同特征张量的标准差估计. 以最小化估计标准差与实际标准差之间的平方和作为优化目标, 自适应求解估测系数α, 具体算法如算法 1 所示.

算法 1. 自适应求解估测系数α.

输入: 校正集C, 估测系数α, 极值个数P, 校正集数量c;

输出: 估测系数α.

① 初始化$\alpha=1$;

② for i in $[0:c:1]$ then

③ 计算$C[i]$的均值μ和标准差σ;

④ $\boldsymbol{P}=topk(abs(C[i]),P)$, $\boldsymbol{P}\in\mathbb{R}^P$;

⑤ $L=length(C[i])$;

⑥ $\sigma'=\sqrt{((\boldsymbol{P}-\mu)^2+\alpha L)/L}$;

⑦ $\alpha'=\arg\min\limits_{\alpha}\|\sigma-\sigma'\|_2^2$;

⑧ $\alpha=(\alpha\times i+\alpha')/(i+1)$;

⑨ end for

⑩ return α.

标准差估计算法减少了平方运算, 但其更重要的意义在于避免了二次逐元素遍历数据, 极值、均值求取可以同时完成. 经实验, 离线求解α仅需要极少量数据即可完成(4~12 个样本).

另一方面, 算法中的超参数——极值个数P的取值并不是一个决定性因素. P的取值变化所引起的计算误差会自适应补偿到估测系数α中. 在实际应用中, 我们实验性地采用了$P=8,16,32$等较小数值, 均能达到估计标准差与实际标准差的误差在10^{-3}之内(与校正集数据比较).

4.2 动态分布阈值τ的感知算法

传统 z-score 方法采用全局静态阈值策略, 假设所有张量数据服从单一分布特性. 这一假设违背了 ViT 权重以及激活值的分布异质性. 如图 3 所示, ViT 模型中不同层的激活值呈现显著分布差异. 基于此, DAQ 提出动态分布阈值τ感知算法以推广 z-score 方法到所有权重和激活中.

阈值τ优化目标为使反量化后的张量$\tilde{\boldsymbol{X}}$与原始张量$\boldsymbol{X}$之间误差最小, 如式(16)所示.

$$\arg\min_{\tau}\|\boldsymbol{X}-\tilde{\boldsymbol{X}}\|_2^2. \tag{16}$$

算法 2. 阈值τ自适应算法.

输入: 校正集C, 估测系数α, 极值个数P, 校正集数量c;

输出: 阈值τ.

① for i in $[0:c:1]$ then

② 算法 1 计算均值μ、标准差σ和极值列表p;

③ $\boldsymbol{X}_2=C[i]>(u+\tau\sigma)|C[i]<(u-\tau\sigma)$;

④ $\boldsymbol{X}_1=C[i]\setminus\boldsymbol{X}_2$;

⑤ $\tilde{\boldsymbol{X}}_1=Dequant(NormQuant(\boldsymbol{X}_1))$;

⑥ $\tilde{\boldsymbol{X}}_2=Dequant(OutlierQuant(\boldsymbol{X}_2))$;

⑦ $\tau'=\arg\min\limits_{\tau}\left\|(\tilde{\boldsymbol{X}}_1+\tilde{\boldsymbol{X}}_2)-(\boldsymbol{X}_1+\boldsymbol{X}_2)\right\|_2^2$;

⑧ $\tau=(\tau\times i+\tau')/(i+1)$;

⑨ end for

⑩ return τ.

动态分布阈值τ感知算法有 2 方面的核心优势: 1)算法普适性. 阈值τ将正常值与离群值区域分割, 两者既可独立量化也可联合量化, 适配任意量化方法; 2)参数轻量化. 仅需离线求解阈值τ, 测试中通过滑动平均法(4~12 个样本)快速收敛, 避免复杂优化计算.

另外, 算法 1 和算法 2 可以联合优化, 同时求解估测系数α和阈值τ. 自此, z-score 计算所涉及到的参数都已求解完成.

z-score 分域结果为与输入张量同样维度的离群值位图. 为了简化计算, 筛选离群值时将求得的每个

元素的 z-score 值与阈值τ做比较转换为与阈值τ代表的数值逐元素比较，减少了逐元素计算 z-score，具体算法如算法 3.

算法 3. z-score 方法筛选离群值.

输入：张量 $\boldsymbol{X}$，估测系数α，极值个数 P，阈值τ；

输出：离群值位图 $\boldsymbol{M}$，极值列表 p.

① 算法 1 计算均值μ、标准差σ和极值列表p；

② $up=(u+\tau\sigma)$，$down=(u-\tau\sigma)$；

③ for i in $[0:L:1]$ then

④ if $\boldsymbol{X}[i]>up$，$\boldsymbol{X}[i]<down$ then

⑤ $M[i]=1$；

⑥ else

⑦ $M[i]=0$；

⑧ end if

⑨ end for

⑩ $p=p+[up,down]$；

⑪ return M，p.

4.3 均匀关联量化

基于 4.2 节的 z-score 检测离群值后，原始数据被划分为正常值$\boldsymbol{X}_1=\{x||Z(x)|\leqslant\tau\}$与离群值$\boldsymbol{X}_2=\{x||Z(x)|>\tau\}$. 该过程同时记录全局极值$x_{\max}$与$x_{\min}$、正常值集极值$\mu+\tau\sigma$和$\mu-\tau\sigma$. 这正是均匀关联量化所需要的关键边界值. 全局极值是正负离群值集中的最大值或最小值，而正常值集极值亦是正负离群值集的临界值. 这些边界值为后续量化参数计算提供统计基准.

针对$\boldsymbol{X}_1$与$\boldsymbol{X}_2$的数值特性，受到分治思想的启发，DAQ 提出的均匀关联量化方法采用双域协同优化策略，其流程包含 2 个阶段：对 z-score 分域后的各域独立设置量化参数，再通过参数调整提升计算效率.

如图 7 所示，空白框图描述了各域的均匀量化步骤. 根据区间极值列表 p 和目标量化位宽 b，分别计算正常值集、正离群值、负离群值集 3 个区间的缩放因子 s，s_+，s_-以及零点偏移 z，z_+，z_-后量化 $\boldsymbol{X}$.

图 7 中阴影框图则为提升计算效率而设计. 多尺度量化参数阻碍了直接利用量化后数据参与计算. 为保持线性计算的高效性，需要对缩放因子和零点偏移做正常值集和离群值集间的关联调整.

对缩放因子引入归一化缩放系数$k_\pm$，采用幂次缩放对齐策略求解相应的正/负离群值的归一化因子：

$$s_\pm 2^{\left\lceil \mathrm{lb}\frac{s_\pm}{s}\right\rfloor}=2^{k_\pm}\cdot s. \tag{17}$$

通过将离群值集的缩放系数调整为正常值集的 2 的幂次方倍，缩放因子尺度不同带来的影响可以通过简单的移位操作进行消除.

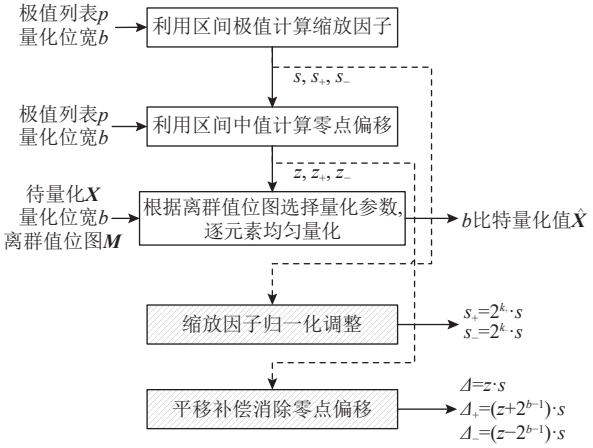


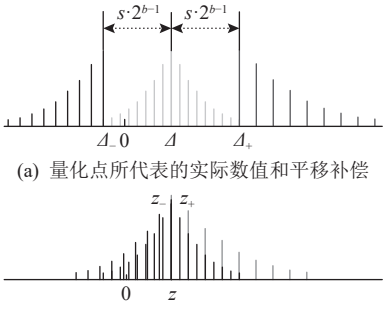
Fig. 7 Flow chart of uniform correlation quantization

图 7 均匀关联量化流程图

另一方面，零点偏移而引起的乘法交叉项可能减弱低位宽量化数据计算带来的增益. 关于零点偏移如何影响计算将在 5.1 节中详细阐述. 本节只阐释零点偏移如何产生以及如何利用平移补偿消除零点偏移.

数据平移并不改变正常值集和异常值集的区分和极值区间长度. 记Δ，Δ_+，$\Delta_-\in\mathbb{R}$分别为正常值集、正离群值集、负离群值集的补偿偏移.

图 8 中横轴表示待量化 $\boldsymbol{X}$ 的数值范围，左、右两侧黑色间隔分别表示量化后的负、正离群值，中间浅色刻度表示量化后的正常值. 横轴“0”为真实 0 值. 零点偏移刻画的是量化值“0”与真实值“0”之间的偏移量. 原始数据除以缩放因子后得到的量化数值分布如图 8(a)所示. 量化数值需减去零点偏移以满足量化位宽 b 的表示范围，如图 8(b)所示.



(a) 量化点所代表的实际数值和平移补偿

(b) 借助零点偏移产生的量化值

Fig. 8 Compensation offset and zero-point offset

图 8 补偿偏移和零点偏移

离群值集共享量化的2^b个数值，故两者量化值恰好以量化后零点为分界点. 图 8(a)表示正负离群值的零点偏移到正常值的零点是相同且固定的，距离为2^{b-1}. 故有：

$$z_+ = z + 2^{b-1},\ z_- = z - 2^{b-1}. \tag{18}$$

而正常值集的零点偏移 z 由均值和缩放因子计算得到:

$$z = \left\lceil \frac{\mu}{s} \right\rfloor. \tag{19}$$

因此, 可以通过 z 计算所有的补偿偏移. 当 $z=0$ 时, 在缩放因子 s 的尺度下, $\Delta = 0$, $\Delta_+ = -2^{b-1}$, $\Delta_- = +2^{b-1}$. 当 $z \neq 0$ 时, 说明数据整体偏离真实值“0”, 平移补偿则将数据整体平移回真实值“0”. 同样在缩放因子 s 的尺度下, 有 $\Delta = z$, $\Delta_+ = -z - 2^{b-1}$, $\Delta_- = -z + 2^{b-1}$.

为了减少对正负离群值的符号表达, 均匀关联量化中所有量化值均采用带符号的量化表达. 以 DAQ 中 4-bit 量化为例, 其符号值域如表 3 所示.

Table 3 4-bit Symmetric Uniform Quantization

表 3 4-bit 对称均匀量化

数值类型	值域	量化参数	
正常值	$0, \pm1, \pm2, \pm3, \pm4, \pm5, \pm6, \pm7, -8$	s	z
离群值*	$0, \pm1, \pm2, \pm3, \pm4, \pm5, \pm6, \pm7, -8$	s'	z'
正离群值	$0, 1, 2, 3, 4, 5, 6, 7, 8$	s_1	z_1
负离群值	$-1, -2, -3, -4, -5, -6, -7, -8$	s_2	z_2

注: “*”表示当仅存单侧正/负离群值时, 不加以符号区分.

5 线性计算

本节在第 4 节的基础上讨论基于现有硬件加速架构, 充分利用 DAQ 量化后的低位宽定点计算来降低计算开销.

5.1 量化乘法

本节讨论 2 个均匀量化后的数据相乘可能的情形. 其计算结果可以表示为:

$$val = (x + z_x)\, s_x \cdot (y + z_y) s_y. \tag{20}$$

根据零点偏移 z 的取值情况, 式(20)有 3 种情形:

1) $z_x = 0,\ z_y = 0$

当原始数据的均值 μ 趋于 0 时, 量化乘法运算满足完美缩放条件:

$$val = (xy)(s_x s_y). \tag{21}$$

此时量化乘法与浮点乘法结果等效且不需要额外的计算补偿, 因此计算效率提升最大.

2) $z_x \neq 0, z_y = 0$ 或者 $z_x = 0, z_y \neq 0$

以 $z_x = z, z_y = 0$ 为例, 有:

$$val = xys_x s_y + zys_y. \tag{22}$$

式(22)中 z 为常量, 故增加的偏移部分 zys_y 为可提前计算的固定值. 式(22)中的 2 个求和项之间也不存在数据依赖, 量化乘法和常量偏移补偿可并行计算, 即“先计算(乘法), 后补偿(加法)”.

3) $z_x \neq 0, z_y \neq 0$

双零点偏移若直接用量化数据计算, 会因较多交叉项的计算导致整体计算效率降低. 此时将 INT4 量化值动态扩展到 INT8 以消除双零点偏移后再计算, 即“先补偿, 后计算”.

$$x' = x + z_x,\ y' = y + z_y,\ val = x'y' s_x s_y. \tag{23}$$

虽然情况 3)为了消除零点偏移产生的交叉项将数据扩展到 INT8 表示, 只是为了将零点偏移提前补偿回 INT4 量化中, 并不意味着此时的量化位宽增长了 1 倍.

5.2 GPU Tensor Core 计算加速

在 NVIDIA Ampere[27] 架构图形处理器(graphics processing unit, GPU)中, 第三代 Tensor Core 实现了对 INT4/INT8 整型矩阵乘累加(matrix multiply-accumulate, MMA)运算的硬件级支持. 以 NVIDIA GeForce RTX 3090(基于 GA102 GPU)[27] 为例, 其包含 82 个流式多处理器(streaming multiprocessor, SM), 每个 SM 集成 4 个 Tensor Core 模块, 提供总计 328 个 Tensor Core, 支持混合精度计算模式(INT 输入, FP16/FP32 累加器).

量化后的 INT 数据有望通过 Tensor Core 实现硬件支持的 INT4/INT8 计算加速. 模型权重参数具有高斯近零分布特性, 对称均匀量化时零点偏移 z 自然满足 $z=0$. DAQ 对激活值虽采用均匀量化方法, 但原始数据分布却不一定基于原点对称. 这就导致计算时需要考虑零点偏移的情形.

若正常值集零点偏移 $z=0$, 可参照第 5.1 节中的情形 2)并结合式(15)的参数归一化, 可以利用 Tensor Core 所支持的 INT4 计算单元直接对量化后数据进行计算, 通过 CUDA Core 并行计算少量离群值量化中产生的固定偏移补偿 $(2^{b-1} \cdot s)$. Tensor Core 计算过程为:

$$val = xy2^{k_x}2^{k_y}s_x s_y;\ k_x,\ k_y \in \{0, k_+, k_-\}. \tag{24}$$

如果正常值集零点偏移 $z \neq 0$, 则属于 5.1 节中的情形 3). CUDA Core 计算补偿项的开销超过了 INT4 计算加速. 此时应先消除零点偏移, 将量化 INT4 数据扩展为 INT8(加上零点偏移后), 再采用 INT8 的 Tensor Core 进行计算. DAQ 基于 GPU 所设计的线性计算加速结构如图 9 所示. Tensor Core 负责整型量化数据的矩阵计算, CUDA Core 则根据量化情形计算偏移补偿、缩放因子移位调节等.

Tensor Core 中 INT4/INT8 矩阵乘加运算对输入矩阵的分块大小有固定限制. PTX 指令形如: *mma.sync.*

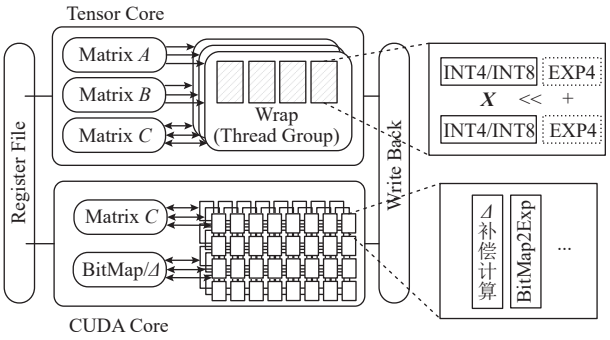


Fig. 9 Accelerated architecture of DAQ based on GPU

图 9 DAQ 基于 GPU 的加速结构

aligned.*m*8*n*8*k*16.*row*.*col*.*s*32.*s*8.*s*8.*s*32. *m*8*n*8*k*16 指示矩阵计算中的分块大小，*s*8/*s*32 指示输入和累加器数据精度，*row*/*col* 指示输入数据分别以行/列主序存储. Tensor Core 支持的 INT4 和 INT8 矩阵分块如表 4 所示. 其中，ViT 激活的通道维度与权重维度都能被分块大小所整除(32/64)，但激活的 Token 维度(197)需要填充以适应 Tensor Core 所支持的分块大小.

Table 4 Computing Patch of Tensor Core and Linear Computational Dimensions of ViT

表 4 Tensor Core 计算分块和 ViT 线性计算维度

数据格式	Tensor Core 矩阵分块 M，N，K	ViT 线性计算（部分） M，N，K
INT8	16，16，16	197，192，384
	32，8，16	197，1 536，384
	8，32，16	197，3 072，768
INT4	8，8，32	197，2 304，768

6 实 验

6.1 实验设置

本文实验部分将在 ViT[2]，DeiT[3] 和 Swin[4] 及其变体上开展. 图像分类任务所选用的实验数据集为 ImageNet-1K[28] 测试集，目标检测任务则使用 COCO 2017 数据集[29]. 图像分类任务的所有全精度模型来自 Timm 库. 目标检测任务的全精度模型来自 MMDetection[30]. 为了与之前的研究方法保持一致，DAQ 在校正过程中随机选取 32 个样本作为校正集，COCO 数据集上选择 1 个样本作为校正数据集，同时采用与文献 [14] 相同的离线量化方案的权重，Softmax 输出激活保持原始精度. 需要说明的是，DAQ 可以使用更少的样本(如 8 个或 12 个)完成校正.虽然 DAQ 认为量化 LN 和 GELU 的输出最有利于量化-计算-存储性能的提升，但这不代表 DAQ 仅适用于 LN 和 GELU 层. 事实上，DAQ 方法对其他层的激活输出仍具备普适性. 在实验过程中先将 DAQ 应用于 LN 和 GELU 以得到模型量化结果，随后给出 DAQ 对其他层的量化结果.

6.2 ImageNet 数据集上的量化模型性能

DAQ 与现有研究在图像识别任务上的量化模型精度比较结果见表 5. 表 5 中每项表示量化模型图像识别任务的 Top-1 精度，精度越高表示识别率越高，模型性能越好. 量化层表示各个研究工作所主要针对的量化目标.

FQ-ViT 在 4-bit 量化时模型崩溃，这可能是由于低位宽计算过程中引入的误差过大. 相比之下，PTQ4ViT，APQ-ViT 的量化模型达到了全精度模型一半的精度；RepQ-ViT 和 RepQuant 明显提升了量化模型精度. 但 DAQ 在大多数模型上的量化效果仍然高于目前最优的 RepQ 系列方法，单个模型最大提升高达 4.37 个百分点(DeiT-T).

DAQ 主要关注的是 4-bit 低位宽量化，但其在 6-bit 量化模型上仍保持了高准确率. 与其他方法相比，DAQ 在超过半数的量化模型达到最好精度. 而且，各个量化模型与全精度模型的平均准确率误差仅有 0.4%，甚至在 DeiT-B 模型上超过全精度模型 0.14 个百分点.

在不同量化位宽下，DAQ 无论是在 G/N 或 S/N 上都展现出了相似的高准确率，表明了 DAQ 方法的普适性.

6.3 COCO 数据集上量化模型性能

DAQ 和现有 SOTA 量化算法在目标检测任务上的结果如表 6 所示. 实验所采用的数据集为 COCO 2017，Mask R-CNN 分别用 Swin-T 和 Swin-S 作为骨干网络. 表 6 中指标 AP^{box}(average precision of bounding box)和 AP^{mask}(average precision of mask)分别表示目标检测任务中检测框定位准确率和实例分割任务中像素级轮廓匹配度，指标越高表示模型性能越好.

表 6 结果显示，DAQ 的 4-bit 量化在目标检测任务上全面超越了之前的研究工作，各项指标提升 8.2~2.5 个百分点，与全精度模型的最大误差为 0.6 个百分点，在低位宽 4-bit 量化上实现了近乎无损的量化模型.

以 Swin-T 作为 Mask R-CNN 的主干网络进行目标检测时，4-bit 量化模型的 AP^{box} 提升高达 8.2 个百分点，6-bit 量化模型在实例分割上的性能指标 AP^{mask} 甚至超过全精度模型 0.1 个百分点.

另一方面，DAQ 将量化目标设为 S/N 或者 G/N

Table 5 Top-1 Accuracy of the Quantization Models on ImageNet Dataset of the Image Classification Task

表 5 量化模型在图像分类任务的 ImageNet 数据集上的 Top-1 准确率

方法	量化目标	位宽	准确率						
			ViT-S/%	ViT-B/%	DeiT-T/%	DeiT-S/%	DeiT-B/%	Swin-S/%	Swin-B/%
全精度模型		32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT[10]	S/N	4/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT[11]	S/G	4/4	42.75	30.96	36.96	34.08	64.39	76.09	74.02
APQ-ViT[13]	S	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
RepQ-ViT[14]	S/N	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
RepQuant[15]	S/N	4/4	73.28	77.84	64.44	75.21	78.46	**81.52**	**82.80**
DAQ（本文）	G/N	4/4	**75.06**	**80.36**	**67.15**	**75.46**	**78.52**	**80.43**	81.45
DAQ（本文）	S/N	4/4	**73.29**	**82.00**	**68.81**	**75.45**	**80.08**	79.95	**82.30**
FQ-ViT[10]	S/N	6/6	4.26	0.10	58.66	45.51	64.63	66.50	52.09
PTQ4ViT[11]	S/G	6/6	78.63	81.65	69.68	76.28	80.25	82.38	84.01
APQ-ViT[13]	S	6/6	79.10	82.21	70.49	77.76	80.42	82.67	84.18
RepQ-ViT[14]	S/N	6/6	80.43	83.62	70.76	78.90	81.27	82.79	**84.57**
RepQuant[15]	S/N	6/6	80.51	83.75	70.89	79.06	81.41	**82.93**	**84.86**
DAQ（本文）	G/N	6/6	**80.81**	**84.09**	**71.69**	**79.56**	**81.63**	82.31	83.98
DAQ（本文）	S/N	6/6	**80.86**	**83.85**	**71.74**	**79.61**	**81.94**	**82.89**	84.34

注：量化目标 S，G，N 分别代表 Softmax，GELU，LN 层输出激活；黑体数值表示最优值.

Table 6 Performance of the Quantization Models on COCO Dataset in the Object Detection Task

表 6 量化模型在目标检测任务的 COCO 数据集上的性能

方法	量化目标	位宽	Mask R-CNN			
			Swin-T		Swin-S	
			AP^{box}/%	AP^{mask}/%	AP^{box}/%	AP^{mask}/%
全精度模型		32/32	46.0	41.6	48.5	43.3
PTQ4ViT[11]	S/G	4/4	6.9	7.0	26.7	26.6
APQ-ViT[13]	S	4/4	23.7	22.6	44.7	40.1
RepQ-ViT[14]	S/N	4/4	36.1	36.0	44.2	40.2
RepQuant[15]	S/N	4/4	37.2	36.8	44.5	40.5
DAQ（本文）	G/N	4/4	**45.4**	**41.3**	**47.9**	**43.0**
DAQ（本文）	S/N	4/4	**45.7**	**41.4**	**48.1**	**43.2**
PTQ4ViT[11]	S/G	6/6	5.8	6.8	6.5	6.6
APQ-ViT[13]	S	6/6	45.4	41.2	47.9	42.9
RepQ-ViT[14]	S/N	6/6	45.1	41.2	47.8	43.0
RepQuant[15]	S/N	6/6	45.3	41.3	48.1	**43.2**
DAQ（本文）	G/N	6/6	**46.0**	**41.6**	**48.2**	**43.2**
DAQ（本文）	S/N	6/6	**46.0**	**41.7**	**48.2**	43.1

注：量化目标 S，G，N 分别代表 Softmax，GELU，LN 层输出激活；黑体数值表示最优值.

上都得到了相当的 SOTA 性能. 2 个模型间的性能差异在+0.3 ~ −0.1个百分点. 实验结果充分证明了 DAQ 的普适性和有效性.

6.4 消融实验

为了说明 DAQ 分治策略和动态感知的 z-score 方法的有效性，在 ViT-S, DeiT-S, Swin-S 三个模型上展开消融实验. 实验采用 4-bit 量化位宽，分别对模型以 Min-Max 量化、DAQ 量化但采用传统阈值（DAQ w/o Adaptive）、DAQ 量化 3 种方法比较. 图 10 中的柱状图表示每类方法在单一类型层输出激活量化的模型准确率，点线表示对模型的 LN 和 GELU 层激活输出同时量化的准确率.

图 10 中，无论是单层量化还是联合量化，Min-Max 量化的量化效果最差，最低准确率仅有 6.45%. 这与第 3 节中的发现，均匀量化在低位宽中难以表征高动态范围的激活离群值，导致模型准确率大幅下降相一致. 当 DAQ 采取分治策略，对离群值和正常值加以区分后，模型的准确率得到显著提升，这一点在 ViT-S 和 DeiT-S 模型上得到充分表征，几乎所有量化结果都得到了 1 个数量级的性能提升. DAQ 采用动态感知的方法调整阈值后，模型准确率进一步得到提升，逼近量化前模型性能.

从图 10 还可以观察到一个非常有趣的现象，那就是 Swin 模型的抗量化误差能力比前 2 类模型要好. 这与 Swin 模型具有更多模型参数和更复杂的模型结构有关.

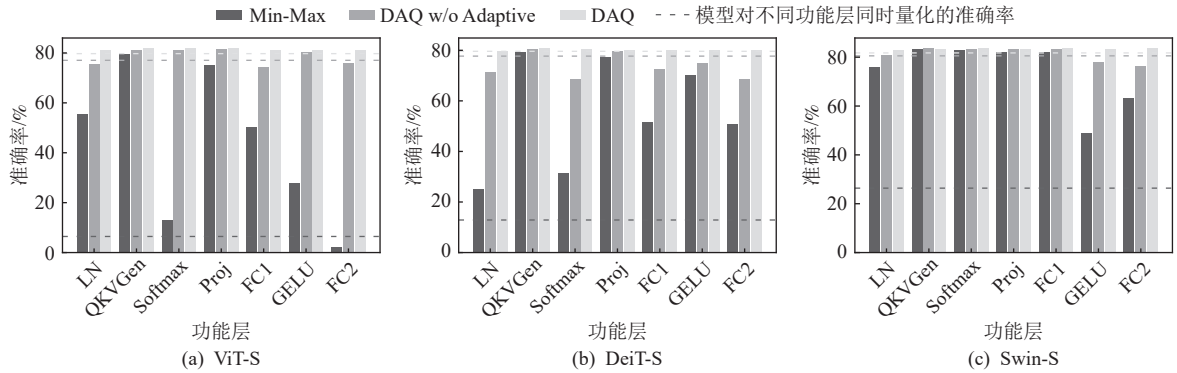


Fig. 10 Ablation study of DAQ

图 10 DAQ 消融实验

6.5 推理加速性能分析

ViT 的计算开销主要用在线性计算. 为了说明 DAQ 对硬件的适配, 表 7 展示了线性计算层的平均时延. 以 ImageNet-K1 数据集为输入, 在 NVIDIA RTX 3090 GPU 上随机推理 100 张图片后的平均时延. 以量化+FP32 计算作为基准, DAQ 表示结合 INT8/INT4 定点计算的归一化结果.

Table 7 Normalization of Average Latency in Linear Calculation

表 7 线性计算平均时延归一化

矩阵大小 (*M*/*N*/*K*)	量化+FP32	DAQ
197/192/192	1	0.57
197/384/192	1	0.51
197/1 536/384	1	0.30
197/2 304/768	1	0.23
197/3 072/768	1	0.20
197/3 072/1 024	1	0.19
197/4 096/1 024	1	0.16

基于 GPU Tensor Core 加速后, DAQ 方法能获得 43%~86% 的线性计算性能提升. 同时, 从表 7 我们还可以观察到, DAQ 的性能增益并未随着线性计算规模的增大而线性增加. 具体来说, 矩阵规模增大约 120 倍(从(197/192/192)到(197/4 096/1 024)), 性能却只增长了 3.5 倍(从 0.57 到 0.16).

另一方面, 表 8 以 DAQ 为基准, 比较了 DAQ 和 PTQ4ViT, RepQ-ViT 端到端的推理性能以及在 RTX 3090 上对 ImageNet-1K 的 val 数据集(共 50 000 张图片)推理的总耗时.

表 8 显示, 与 PTQ4ViT 相比, DAQ 最大性能提升高达 41.8%, 平均性能提升超过 28%; DAQ 相比 RepQ-ViT 的时间开销也降低了 36.3%~16.8%, 平均推理时延减少 27%. 与线性计算结果相一致, 当模型越大, 线性计算开销占比越高时, DAQ 的模型增益也有所放缓.

Table 8 Time Overhead Comparison of ImageNet-1K Test Set

表 8 ImageNet-1K 测试集时间开销比较 s

模型	DAQ	PTQ4ViT[11]	RepQ-ViT[14]
ViT-S	70.49	88.21(25.1%)	91.19(29.4%)
ViT-B	167.67	209.43(24.9%)	219.83(31.1%)
DeiT-T	32.30	45.81(41.8%)	44.03(36.3%)
DeiT-S	65.43	88.25(34.8%)	88.93(35.9%)
DeiT-B	167.02	209.48(25.4%)	214.60(28.5%)
Swin-T	104.57	130.26(24.5%)	121.36(16.1%)
Swin-S	168.70	205.2(21.6%)	197.03(16.8%)

注: 括号中数值表示当前方法相较于 DAQ 方法的时间增幅百分比, 数值越大, 表明推理延迟相对于 DAQ 越高.

因此, 量化性能受限于 2 个方面: 1)在线量化开销, 量化算法设计越精细, 增加的非规则计算越多, 算法引入的端到端延迟开销越高; 2)利用 Tensor Core 等计算核心对量化算法加速时, 矩阵分块形式、数据组织和传输都影响着最终加速效果.

6.6 离线校正效率分析

表 9 展示了不同方法之间的离线校正开销对比, 包括校正样本数和校正时间开销 2 个方面. 离线校正时间开销基于单卡 3090 GPU, 校正样本集由 ImageNet-1K 中随机抽取.

DAQ 在离线校正中采用网格搜索确定超参数, 单次查找计算复杂度由网格搜索的范围所决定. 因此 DAQ 离线校正的时间开销与样本数量呈正比. 但 DAQ 离线校正的显著优势在于仅小样本数据即可保证离线校正后模型的最终准确率.

从表 9 可以看出, 在同样的 32 个样本下, DAQ 的时间开销最短且能得到相对最佳模型准确率. 另一

Table 9 Efficiency Analysis of Off-Line Calibration

表 9 离线校正效率分析

模型	方法	Top-1/%	校正样本数	GPU 时间/min
DeiT-S	FP32	79.85		
	PTQ4ViT[11]	34.08	32	3.2
	RepQ-ViT[14]	69.03	32	1.3
	DAQ（本文）	**75.46**	32	**1.2**
	DAQ（本文）	75.12	12	0.5
	DAQ（本文）	75.08	4	**0.1**
Swin-S	FP32	83.23		
	PTQ4ViT[11]	76.09	32	7.7
	RepQ-ViT[14]	79.45	32	2.9
	DAQ（本文）	**80.43**	32	**2.5**
	DAQ（本文）	80.21	12	0.9
	DAQ（本文）	80.01	4	**0.3**

注：黑体数值表示最优值.

方面, DAQ 在样本量大幅减少到 12 或 4 时, 校正时间线性减少的同时仍然能保持模型准确率(降低小于 1%). 这也意味着 DAQ 在少样本条件下适应性仍然良好.

6.7 权重量化的适用性

DAQ 的重要特性之一是兼容具有不同分布的数据. 因此 DAQ 不仅适用于动态激活量化, 同样也适用于静态权重量化. 但 ViT 权重为静态数据且高度符合高斯分布, 关于权重量化有许多成熟的方法如 GPTQ[31], AdaRound[32] 等且能取得较好的量化效果. 因此权重量化并不作为 DAQ 的重点解决的问题. 另一方面, 为了与 SOTA 方法保持一致进行公平的比较, 6.2~6.6 节中的权重量化均采用与文献 [14] 相同的量化方法.

本节为了展示 DAQ 对静态权重量化的适用性, 基于 ImageNet-1K 数据集设计了对应的量化对比实验. 表 10 中, DAQ 权重量化仍采用与 RepQ-ViT 相同的方法[16], DAQ/W 表示模型权重与激活均采用 DAQ 量化.

表 10 显示, DAQ/W 同时应用在权重与激活上时, 比仅在激活上使用 DAQ 时模型准确率有所降低, 降幅最高达 9.00 个百分点, 但最低仅为 0.05 个百分点. 尽管如此, DAQ/W 的结果仍然全面超越采用文献 [16] 的 RepQ-ViT.

Table 10 Weight Quantization of DAQ Applied to ViTs

表 10 DAQ 应用于 ViTs 的权重量化 %

方法	ViT-S	ViT-B	DeiT-S	DeiT-B	Swin-S	Swin-B
FP32	81.39	84.54	79.85	81.80	83.23	85.27
RepQ-ViT[16]	65.05	68.48	69.03	75.61	79.45	78.32
DAQ	73.29	82.00	75.45	80.08	79.95	82.30
DAQ/W	70.32	73.00	75.14	79.80	79.90	79.64

7 结 论

本文针对视觉 Transformer(ViT)系列模型的低比特激活量化难题, 研究了 ViT 关键模块的量化误差与计算/存储开销的错位匹配, 观察到激活不依赖于数据分布的集中共性, 从而提出分治自适应量化方法 DAQ, 系统性解决了后训练量化中离群值表征、计算-精度失配及硬件协同适配等核心挑战. 理论分析与实验验证表明, DAQ 在绝大多数情况下达到最优水平, ImageNet 任务中 4-bit 量化的 Top-1 精度较现有 SOTA 方法最大提升 4.37 个百分点, 特定条件下量化模型精度超过全精度模型, 同时在目标识别任务上达到近似无损的低位宽量化. DAQ 设计的关联量化方法与 Tensor Core 适配计算流减少了反量化过程, 降低了量化引入的计算开销, 能够获得 43%~86% 的线性计算加速. DAQ 不仅揭示了非高斯分布下数值聚集性与量化误差传播的关联机制, 更为边缘设备部署高精度、低功耗视觉 Transformer 提供了理论指导与工程实践闭环方案.

8 讨 论

本文针对以下 3 个方面进行了讨论:

1)ViT 量化对性能的提升. 仅追求低比特量化并不总能带来性能上的提升. 量化虽然能减少访存和带宽需求, 但端侧部署还需要考虑计算实时性、能耗等方面. 同时, 现有硬件不能自适应地提供低位宽量化数据的高效计算支持, 而复杂精妙的量化方法如混合精度量化对硬件计算带来更高的挑战. 最后, PTQ 量化为了保持精度, 仍需要量化/反量化的额外计算, 增加了计算负载.

2)ViT 量化所产生的稀疏性. 原始数据中靠近均值的数值均匀量化后为 0, 因此量化数据的稀疏度会比原始数据要高. 仅依靠现有的计算架构如 GPU 无法很好地利用这样的非结构化稀疏. 因此, 量化算法想要切实带来计算方面的性能提升, 必须要考虑底层硬件架构的适配和设计.

3)ViT 量化对新兴视觉模型的借鉴意义. ViT 在计算机视觉领域中的广泛适用性催生了许多 ViT 的跨模态、跨网络结构融合的新兴视觉模型. 以 Sora[33] 为

代表的扩散模型从 DDPM[34] 转向 Diffusion Transformer (DiT)[33] 为例，基于 ViT 的 U-ViT 架构已展现出替代传统 U-Net 的潜力. 当以 ViT 作为新兴模型的骨干网络时，ViT 的二次复杂度、数据非高斯分布、存在离群性的特性是由模型本身所决定的，因此这些特性只会在表现强度上有所不同而不会完全消失. 成熟的量化方法在新兴模型中也能发挥相应的作用，但仍要结合 ViT 模型在新兴模型中结构的变形、融合和增改等引起的独特特性加以综合分析与考量. 比如 DETR 目标检测模型融合了 ViT 和 Decoder 结构，只针对 ViT 量化的方法在 DERT 上并不总能在保证精度的前提下获得相应的量化模型.

作者贡献声明：吕倩茹、许金伟负责方案整体设计并撰写论文；吕倩茹、许金伟、姜晶菲负责部分算法思路和实验方法；姜晶菲、李东升提出指导意见并修改论文.

参 考 文 献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998−6008

[2] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: Transformers for image recognition at scale[J]. arXiv preprint, arXiv: 2010.11929, 2020

[3] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention[J]. arXiv: 2012.12877

[4] Liu Ze, Lin Yutong, Cao Yue, et al. Swin Transformer: Hierarchical vision Transformer using shifted windows[C]//Proc of the IEEE/CVF Inte Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 10012−10022

[5] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C]//Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778

[6] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//Proc of European Conf on Computer Vision. Berlin: Springer 2020: 213−229

[7] Cheng B, Schwing A, Kirillov A. Per-pixel classification is not all you need for semantic segmentation[J]. Advances in Neural Information Processing Systems, 2021, 34: 17864−1787

[8] Gündüç Y. Vit-GAN: Image-to-image translation with vision Transformes and conditional GANS[J]. arXiv preprint, arXiv: 2110.09305, 2021

[9] Hatamizadeh A, Song J, Liu G, et al. Diffit: Diffusion vision transformers for image generation[C]//Proc of European Conf on Computer Vision. Berlin: Springer, 2024: 37−55

[10] Lin Yang, Zhang Tianyu, Sun Peiqin, et al. Fq-ViT: Fully quantized vision Transformer without retraining[J]. arXiv preprint, arXiv: 2111.13824, 2021

[11] Yuan Zhihang, Xue Chenhao, Chen Yiqi, et al. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization[C]//Proc of European Conf on Computer Vision. Berlin: Springer, 2022: 191−207

[12] Li Yanjing, Xu Sheng, Zhang Baochang, et al. Q-ViT: Accurate and fully quantized low-bit vision Transformer[J]. Advances in Neural Information Processing Systems, 2022, 35: 34451−34463

[13] Zhong Yunshan, Huang You, Hu Jiawei, et al. Towards accurate post-training quantization of vision transformers via error reduction[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025, 47(4): 2676−2692

[14] Li Zhikai, Xiao Junrui, Yang Lianwei, et al. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 17227−17236

[15] Li Zhikai, Liu Xuewen, Zhang Jing, et al. RepQuant: Towards accurate post-training quantization of large Transformer models via scale reparameterization[J]. arXiv preprint, arXiv: 2402.05628, 2024

[16] Du Dayou, Gong Gu, Chu Xiaowen. Model quantization and hardware acceleration for vision Transformers: A comprehensive survey[J]. arXiv preprint, arXiv: 2405. 00314, 2024

[17] Tai Yushan, Lin Mingguang, Wu A. TSPTQ-ViT: Two-scaled post-training quantization for vision Transformer[C]//Proc of 2023 IEEE Int Conf on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ: IEEE, 2023: 1−5

[18] Chen C F R, Fan Q, Panda R. Crossvit: Cross-attention multi-scale vision transformer for image classification[C]//Proc of the IEEE/CVF Int Conf on computer vision. Piscataway, NJ: IEEE, 2021: 357−366

[19] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84−90

[20] Zhu Xizhou, Su Weijie, Lu Lewei, et al. Deformable DETR: Deformable transformers for end-to-end object detection[J]. arXiv preprint, arXiv: 2010.04159, 2020

[21] Strudel R, Garcia R, Laptev I, et al. Segmenter: Transformer for semantic segmentation[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 7262−7272

[22] Zheng Sixiao, Lu Jiachen, Zhao Hengshuang, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 6881−6890

[23] Liu Yixin, Zhang Kai, Li Yuan, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models[J]. arXiv preprint, arXiv: 2402.17177, 2024

[24] Dettmers T, Lewis M, Belkada Y, et al. GPT3. Int8 (): 8-bit matrix multiplication for transformers at scale[J]. Advances in Neural Information Processing Systems, 2022, 35: 30318−30332

[25] Touvron H, Lavril T, Izacard G, et al. Llama: Open and efficient foundation language models[J]. arXiv preprint, arXiv: 2302.13971, 2023

[26] Chan T F, Golub G H, LeVeque R J. Algorithms for computing the sample variance: Analysis and recommendations[J]. The American Statistician, 1983, 37(3): 242−247

[27] Nvidia. NVIDIA GPU Ampere Architecture Whitepaper, https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf

[28] Deng Jia, Dong Wei, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]//Proc of 2009 IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2009: 248−255

[29] Lin Tsung-Yi, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context[C]//Proc of the 13th European Conf on Computer Vision (ECCV 2014). Berlin: Springer, 2014: 740−755

[30] Chen Kai, Wang Jiaqi, Pang Jiangmiao, et al. MMDetection: Open mmlab detection toolbox and benchmark[J]. arXiv preprint, arXiv: 1906.07155, 2019

[31] Frantar E, Ashkboos S, Hoefler T, et al. GPTQ: Accurate post-training quantization for generative pre-trained transformers[J]. arXiv preprint, arXiv: 2210.17323, 2022

[32] Nagel M, Amjad R A, Van Baalen M, et al. Up or down? adaptive rounding for post-training quantization[C]//Proc of Int Conf on Machine Learning. New York: ACM, 2020: 7197−7206

[33] Peebles W, Xie S. Scalable diffusion models with transformers[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 4195−4205

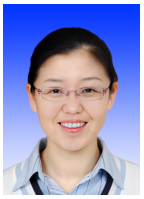
[34] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840−6851

Lü Qianru, born in 1993. PhD candidate. Her main research interests include computer architecture and artificial intelligence.

吕倩茹, 1993 年生. 博士研究生. 主要研究方向为计算机体系结构、人工智能.

Xu Jinwei, born in 1990. PhD, assistant research fellow. Member of CCF. His main research interests include artificial intelligence and reconfigurable computing.

许金伟, 1990 年生. 博士，助理研究员. CCF 会员. 主要研究方向为人工智能、可重构计算.



Jiang Jingfei, born in 1974. PhD, professor. Member of CCF. Her main research interests include reconfigurable computing, artificial intelligence, and computer architecture. (jingfeijiang@nudt.edu.cn)

姜晶菲, 1974 年生. 博士，研究员. CCF 会员. 主要研究方向为可重构计算、人工智能、计算机体系结构.

Li Dongsheng, born in 1978. PhD, professor. Member of CCF. His main research interests include parallel computing, artificial intelligence, computer architecture.(dsli@nudt.edu.cn)

李东升, 1978 年生. 博士，研究员. CCF 会员. 主要研究方向为并行计算、人工智能、计算机体系结构.