https://crad.ict.ac.cn
63(6): 1574−1583, 2026

基于低比特共享的整型量化方法

李林浩[1] 曾宣迪[1] 唐忆滨[2,3] 刘博生[1] 武继刚[1]

[1](广东工业大学计算机学院 广州 510006)

[2](武汉数字工程研究所 武汉 430074)

[3](图形处理芯片及智能计算系统技术创新中心 武汉 430074)

(lilinhao0830@163.com)

Integer Quantization Method Based on Low Bit Sharing

Li Linhao[1], Zeng Xuandi[1], Tang Yibin[2,3], Liu Bosheng[1], and Wu Jigang[1]

[1](*School of Computer Science and Technology*, *Guangdong University of Technology*, *Guangzhou* 510006)

[2](*Wuhan Digital Engineering Institute*, *Wuhan* 430074)

[3](*Innovation Center for GPU and Intelligent Computing System Technology*, *Wuhan* 430074)

Abstract With the rapid advancement of artificial intelligence technologies, large language models (LLMs) have emerged as foundational components in modern intelligent systems. However, the ever-increasing model size—ranging from billions to hundreds of billions of parameters—poses significant challenges in terms of memory footprint and computational overhead, particularly on latency-sensitive and resource-constrained platforms. While low-bit integer quantization (e.g., INT8) has demonstrated effectiveness in reducing memory consumption and improving inference efficiency, it suffers from notable quantization limitations when applied to large-scale models, including quantization-induced accuracy degradation and inefficient storage utilization. To address these issues, we propose a numerical quantization framework, termed low bit sharing (LBS). Built upon conventional integer quantization, LBS provides a structured high-low bit decomposition scheme that preserves only the most significant bits for each parameter and shares the less significant bits within a tensor group. The shared solution can reduce storage requirements while upholding the numerical representational capability. Furthermore, to mitigate accuracy degradation caused by the quantization of high-impact weights, we develop a salient value aware quantization strategy. By employing a Top-*K* selection algorithm, we isolate the most influential weights—typically residing in the tail of the parameter distribution—and assign them dedicated scaling factors. This targeted treatment effectively suppresses error accumulation during quantization and improves robustness in downstream performance. Experimental results across several state-of-the-art LLMs show that LBS achieves substantial storage reduction while preserving high model accuracy, making it a practical and scalable solution for deployment of large-scale models.

Key words low bit sharing (LBS); large language models (LLMs); integer (INT) quantization; model accuracy; numerical representation

摘 要 随着人工智能技术的快速发展,大语言模型(large language models, LLMs)已成为现代智能系统的关键组成部分。然而,模型规模的持续增长带来了显著的存储与计算开销,严重制约了其在资源受限

收稿日期: 2025-09-02; **修回日期**: 2026-01-12

基金项目: 国家自然科学基金项目(62302102); 广东省基础与应用基础研究基金项目(2023A1515012844, 2025A04J5208)

This work was supported by the National Natural Science Foundation of China (62302102) and the Guangdong Basic and Applied Basic Research Foundation (2023A1515012844, 2025A04J5208).

通信作者: 刘博生(liubosheng@gdut.edu.cn)

设备中的部署效率。尽管低比特整数量化技术（如INT8）在降低存储需求与提升推理效率方面取得了一定成效，但在面对超大模型时仍存在精度下降与存储冗余的问题。为解决上述问题，提出一种数值量化方法——低比特共享（low bit sharing, LBS）。该方法在传统整型量化基础上引入结构化的高低位分离机制，仅保留权重的高有效位，并在张量组内共享低比特信息，从而在保持数值表达能力的同时显著地降低存储开销。此外，针对整数量化中显著权重易受量化影响的问题，进一步提出显著值感知量化策略，基于Top-K选择算法识别具有高影响力的关键权重值，并通过引入缩放因子缓解其在量化过程中的误差累积，增强模型精度鲁棒性。实验结果表明，所提出的LBS方法在多个主流大语言模型上展现出优越的存储性能与精度保持能力，为大语言模型的部署提供了一种可行的解决方案。

关键词 低比特共享；大语言模型；整型量化；模型精度；数值表示

中图法分类号 TP181

DOI: 10.7544/issn1000-1239.202550648 **CSTR:** 32373.14.issn1000-1239.202550648

随着人工智能技术的持续演进，大语言模型（large language models, LLMs）已成为构建现代智能系统的基础模块，广泛赋能文本生成、对话系统、跨语言处理等多种自然语言处理任务，例如应用于文本生成[1-5]、多轮对话[6-7]及辅助机器学习任务[8-9]等，展现出强大的精度性能与泛化能力。然而，大语言模型通常包含数十亿至数百亿级别的参数规模，导致其在实际部署过程中面临高内存占用、吞吐率受限等关键挑战，尤其在延迟敏感及资源受限的终端设备中更为突出。传统的浮点数表示虽然具备高精度优势，但其在计算与存储方面的高开销显著增加了部署成本，严重制约了大模型在边缘设备中的应用。

针对上述问题，低比特整型量化技术[10-12]被广泛认为是提升部署效率、降低资源开销的有效途径之一。其通过将模型参数从浮点格式（如FP32）转换为低比特整型格式（如INT8及以下），在显著减少访存带宽与存储空间的同时，仍可在一定程度上维持模型精度并保持推理效率。近期研究逐步聚焦于如何在保证精度的前提下进一步减少低比特模型的存储开销。SmoothQuant[13]通过引入平滑变换策略，缓解了异常值在量化过程中的误差传播问题，使得浮点张量可以更为稳定地映射至INT8整数域，有效提升了精度鲁棒性。然而，尽管INT8量化已取得阶段性进展，其存储成本在大规模模型中依然不可忽视。为进一步减少模型规模并优化精度性能，Tender[14]提出了基于张量通道维度的分解量化方法。通过将模型参数划分为多个子张量并隔离异常值区域，有效提升了量化灵活性，在减少存储负担的同时有效维持关键数值精度。

本文提出一种新颖的数值量化框架——低比特共享（low bit sharing, LBS）。该方法在整型量化基础上引入结构化共享机制，在提升存储效率的同时保持较高模型精度。本文通过分析权重张量指数分布，如图1所示，发现其小值众多但贡献很小，大指数权重虽少却影响显著。基于此，将大语言模型中的权重数值划分为显著值与普通值以实现分别量化。具体地，通过Top-K算法筛选出显著权重，将其与普通值分离并单独量化，以提升普通值量化分辨率。对于显著值，设计缩放因子缓解其量化过程的误差，以提升量化的数值表达能力。对于普通值，LBS的核心思想是将整型表示划分为高位与低位2部分。这源于随着量化位宽的增加，低有效位对整体数值精度的边际影响逐步减弱。基于此，LBS保留每个元素的高位部分，并在张量组内共享低位信息。此共享机制在几乎不损失数值表达能力的前提下，显著减少了存储需求。

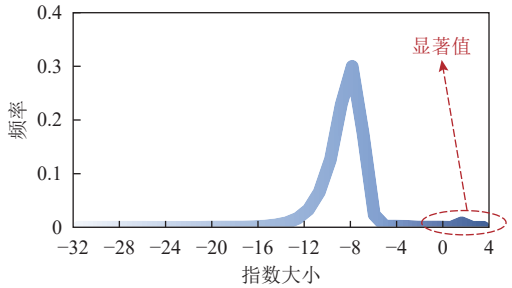


Fig. 1 The original floating-point weight exponential distribution

图1 原始浮点权重指数分布

本文的主要贡献如下：

1）提出了一种新颖的LBS量化方法，支持在张量组内共享低位权重数值，实现减少存储的同时保持优异的精度性能。

2）提出了一种显著值感知量化策略，基于Top-K

筛选机制，精准控制高权重值的量化误差，有效解决整型量化中因显著值量化导致的精度损失问题。

1 相关工作

大语言模型的量化方法包括针对权重的量化[15-17]和针对激活的量化[18-20]。针对权重的量化在推理过程中可以进行离线操作，实现将权重先量化好之后，再加载到大语言模型中进行运算。针对激活的量化为在线操作，这是由于每一层的激活值是通过前一层的输入激活和权重计算得来。在推理过程中，由于将生成的激活进行量化需要在计算中同步处理，本文主要聚焦权重量化。

在已有量化技术不断进步的基础上，近年来的研究焦点已转向对数值表示方法本身的创新，涵盖低精度浮点格式(FP)[20-23]与整数量化技术[24-25]。相比传统整数量化方法，低精度浮点格式在低比特宽度下引入指数与尾数的灵活位宽配置，实现在动态范围表达与数值精度之间的更优动态平衡。其中，FP8-LM[26]通过在表示能力与实现复杂度间的深度协同优化，在内存占用、算力效率及分布式训练通信开销等方面，均显著优于BF16，展现出其在大规模模型部署中的工程潜力。

在整数量化方面，ANT[27]提出面向低比特大语言模型量化的自适应数值表示机制，通过融合浮点与整数格式的优势，并针对不同张量分布动态选择最适配的数值类型，在兼顾硬件友好的前提下，显著提升了量化的灵活性与精度性能。OMNIQUANT[28]在极低比特量化框架下引入LWC与LET机制，用以增强对异常值量化的鲁棒性，从而在保持后训练量化(PTQ)效率的基础上，实现了模型推理速度与内存占用的显著优化。SmoothQuant[13]通过引入平滑变换策略，实现在超大规模语言模型中的无损量化。在降低推理成本的同时维持了高精度表现，成为当前大模型部署中的关键量化技术之一。Tender方法[14]基于张量分解与以2次幂为基础的缩放策略，实现了对大语言模型推理过程的有效量化。在无需引入混合精度训练或依赖定制化硬件支持的前提下，该方法可完成隐式再量化操作，从而在保持计算通用性的同时，有效提升模型的推理性能与预测精度。

近期国内外研究中，一类量化方法聚焦于优化数值映射。例如，MergeQuant[18]方法通过逐通道校准等技术，将每个浮点参数映射为独立的整型数值，以减少数值层面的信息损失。另一类研究则从数据分布特性出发。DAQ[29]通过z-score将激活值划分为2个子集并应用关联量化。QuaRot[30]则运用数学旋转重塑数据分布以消除离群点，并将该操作融合到权重中。Systematic outliers[31]的研究则指出，这些离群点是自注意力机制[32]中Softmax操作的固有结果，在模型中可作为上下文感知的缩放因子。而本文提出的LBS框架则在比特层面进行存储优化。该方法通过在权重组内共享高位比特，仅为每个权重独立存储低位比特，从而利用整数表示中的存储冗余。

2 整数量化基础

2.1 INT量化

INT量化作为一种深度学习模型优化技术，通过将模型中的计算数值从高精度浮点数(如FP32)转换为低精度整数(如8b的INT8或4b的INT4)，在保持模型性能和准确率的前提下减少存储空间，提升模型在硬件设备上的运行效率。

对于浮点数，将集合定义为S，其表示为：

$$S=\{x_1,\cdots,x_i,\cdots,x_N\}\,, \tag{1}$$

其中N表示浮点数的总数。将原始浮点数量化为整数的过程如下：

$$x_{\mathrm{q}}=\frac{x}{s},\quad s=\frac{x_{\max}}{2^{b-1}}\,, \tag{2}$$

其中x_{q}表示量化后的整数，x表示浮点数，s为量化因子，b为INT量化后的位宽。其权重在INT量化的过程会存在误差，其误差定义为：

$$Err(Q(W))=s\cdot Round\left(\frac{W}{s}\right)\,, \tag{3}$$

其中W表示浮点权重，$Round$表示随机舍入误差，Q为量化函数。

2.2 数值表示

图2展示了LBS方法与传统方法之间的对比。在传统整型方法中，每个元素通过L位位宽进行编码。而LBS机制在此基础上引入组内比特重构策略：每组采用统一结构，由1位符号位(sign bit)、$L-m-1$位私有比特位(private bit)以及m位共享比特位(shared bit)组成，实现跨元素的比特资源共享，从而在保持精度的同时显著降低整体存储开销。

现有整型量化方法多采用固定比特宽度(如FP32转INT4)以实现模型轻量化，然而此类统一比特分配策略未能充分考虑张量内部显著值的分布特性，易在高显著值区域引入较大量化误差，导致模型精度变差。本文提出的LBS机制，通过自适应识别

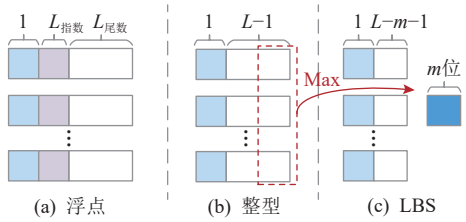


Fig. 2 Comparison of floating-point, INT and LBS

图 2 浮点、INT 和 LBS 的对比

张量中的显著值分布，并动态调整比特分配策略，能在有效保持关键信息表达精度的同时，实现更优的精度性能提升。

3 基于 LBS 的整型量化方法

3.1 LBS 工作流

图 3 展示了所提出的 LBS 机制的整体工作流，主要包括显著值放大与 LBS 策略两大模块。如图 3 所示，原始浮点权重具有显著的幅值差异。为了兼顾精度与量化效率，本文首先利用 Top-K 算法识别出显著值，并对其进行独立高精度量化与放大处理，以避免因粗粒度量化导致的信息丢失。对于其余普通值，本文采用 LBS 方法，将其划分为若干组，并在组内采用共享编码方式以提升存储效率。基于上述划分，本方法对 2 类数据分别采取差异化量化策略。显著值采用放大保护机制，以增强其精度表征；普通值则采用 LBS 机制以减少存储开销。最终的量化结果表达为：

$$S' = S_{\text{Amplify-value}} \cup S_{\text{Shared-value}}, \tag{4}$$

其中 S'为将原始浮点量化后的集合。$S_{\text{Amplify-value}}$ 表示经放大处理后的显著值集合(具体见式(7))，$S_{\text{Shared-value}}$ 表示经 LBS 处理的普通值集合(见式(10))。

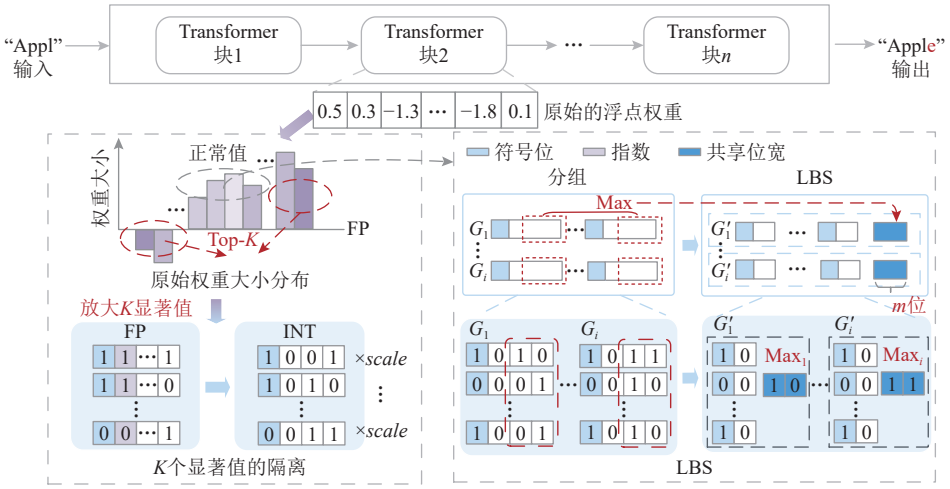


Fig. 3 Workflow of LBS

图 3 低比特共享的工作流程

3.2 Top-K 显著值放大

显著值通常具备较大的数值幅度，直接参与量化会大幅扩大整体数据的表示范围，进而导致非显著值的量化步长增大及精度损失加剧。因此，本文将显著值集合与普通值集合进行分离。显著值的隔离与单独处理对于提升整体量化精度至关重要。显著值放大的目的在于在有限的比特位宽量化中，针对超出数值表示范围的显著值数值，通过引入放大因子，有效减少量化带来的误差损失。

针对原始浮点集合 S，首先执行指数统计，并在去除重复项后，从中选取指数幅值不同的前 k 个最大元素，过程如式(5)所示。

$$\begin{cases} e_i = \lfloor \text{lb}(|x_i|) \rfloor, \\ D = TopK(Unique(Sort(e_i)), k), \end{cases} \tag{5}$$

其中函数 $Sort()$用于将指数值按降序排列，函数 $TopK()$为选取 k 个不相同的最大指数，D 为 k 个指数不同的元素组成的集合。

为标识显著值在原集合中的位置，引入掩码函数 $Mask(x_i)$，定义为：

$$Mask(x_i) = \begin{cases} 1, e_i \in D, \\ 0, e_i \notin D, \end{cases} \quad 1 \leqslant i \leqslant N。 \tag{6}$$

据此，原始集合 S 包括 2 个子集：显著值集合($S_{\text{Critical-value}}$)和普通值集合($S_{\text{Common-value}}$)，分别对应掩码值为 1 和 0 的元素。

针对显著值的量化，其处理流程形式化表达为：

$$\begin{cases} S_{\text{Critical-value}} \xrightarrow{\text{INT量化+放大}} S_{\text{Amplify-value}}, \\ S_{\text{Amplify-value}} = scale \times INT(x_i), \ x_i \in S_{\text{Critical-value}}, \end{cases} \tag{7}$$

其中 *scale* 为放大因子，用以提升关键数值的表示能力，确保其在低比特约束下的信息表达不被淹没。

对于被放大的显著值总数，标记为 K，具体为：

$$K = Count(S_{\text{Amplify-value}}), \tag{8}$$

其中函数 $Count()$ 为统计个数。这里如果存在数值不同但具有相同的指数值，会被选中，根据式(5)，$K \geqslant k$。

3.3 LBS

1)分组

分组是实现 LBS 的核心策略，其目的在于平衡比特利用效率与模型精度这一核心矛盾。若无分组，全局共享将因巨大的量化误差导致精度崩溃。为此，LBS 采用基于权重“局部相似性”的分组方法，通过实现“局部共享”来控制量化误差。这一策略中关键超参数组大小(g)的选择，以 4.4 节进行的实验探索为依据。

对于普通值集合 $S_{\text{Comman-value}}$，在执行整型量化后，对齐进行等规模分组，形式表达为：

$$\begin{cases} S_{\text{Comman-value}} \xrightarrow{\text{分组}} \{G_1, \cdots, G_i, \cdots, G_t\}, \\ G_i = \{X_{11}, \cdots, X_{ij}, \cdots, X_{ig}\}, t = \left\lceil \dfrac{N-K}{g} \right\rceil, \end{cases} \tag{9}$$

其中 G_i 表示第 i 个分组，g 为每组包含的元素个数，t 为总的分组数。

2)LBS 编码

在每个分组 G_i 内，本文采用 LBS 策略对量化后的整数值进行编码，得到共享表示集合 $S_{\text{Shared-value}}$。

$$\begin{aligned} &S_{\text{Shared-value}} = \{G'_1, \cdots, G'_i, \cdots, G'_t\}, \\ &G'_i = \{X'_{11}, \cdots, X'_{ij}, \cdots, X'_{ig}\}。 \end{aligned} \tag{10}$$

具体而言，在 LBS 框架下，每个量化整数由 2 部分构成：私有位 $X_{s_{ij}}$ 和共享位 X_{pi}，其编码结构定义为

$$\begin{cases} X'_{ij} = \{\text{符号位} \mid X_{s_{ij}} \mid X_{pi}\}, \\ X_{s_{ij}} = X_{ij}[L-1, m], \\ X_{pi} = \max(Low_m(X_{ij})),\ X_{ij} \in G_i, \end{cases} \tag{11}$$

其中{}表示不同的数据位组合，[ ]表示取其中对应的比特宽度，$X_{s_{ij}}$ 表示私有比特对应的比特数，X_{pi} 表示第 i 组中所有整数的低 m 位的最大值，作为组内共享部分进行复用。

3.4 误差分析

1)显著值的放大误差分析

在显著值的处理方面，先前工作如 AWQ[17] 主要关注显著值应用缩放因子(*scale*)以提升表示能力，试图在增强显著性与误差诱导之间取得平衡。与之相比，本文方法首先基于掩码机制将显著值与普通值解耦，并对显著值独立执行量化与放大操作。

根据式(3)及相关误差定义，权重显著值量化并放大后的误差表达为：

$$\delta_{\text{Amplify}} = scale \cdot Err(Q(W)), \tag{12}$$

其中 $Err(Q(W))$ 为量化过程中由舍入产生的误差，其理论分布近似为 [0, 0.5] 的均匀分布，因此，平均误差为 0.25。显著值的放大因子 *scale* 将此误差按比例放大为 δ_{Amplify}，但由于本文采用的是适当放大策略，因此整体误差可控。此外，在有限比特下，显著值的放大可避免其被低精度量化所“淹没”，从而保留关键权重信息并增强其对模型行为的贡献，有效提升整体表达能力与推理性能。

2)LBS 误差分析

针对所提出的 LBS 机制，本节对其引入的量化误差进行分析。根据式(11)，LBS 中每个量化值 X'_{ij} 由私有位与共享位组成，其中误差主要源于分组中共享比特的近似表示。

具体而言，量化误差可定义为组共享值 X_{pi} 与组内真实值 X_{ij} 的低 m 位之间的差值。形式化表达为：

$$\delta_{ij} = \left| X_{pi} - Low_m(X_{ij}) \right|。 \tag{13}$$

假设分组内低 m 位比特在 $[0, 2^m-1]$ 范围内均匀分布，则该误差的数学期望均值 E 可表示为

$$E(\delta_{ij}) = \frac{2^m - 1}{2}。 \tag{14}$$

4 实验结果

4.1 实验设置

1)模型与数据集

本文在多个主流大语言模型上对所提出的 LBS 方法进行了系统评估，包括 OPT[33] 和 LLAMA[34] 系列。评估涵盖 8 个具有代表性的数据集，包括 ARC_challenge[35]，HellaSwag[36]，QNLI[37]，PIQA[38]，Wiki[39]，lambada_openai[40]，lambada_standard[40]，MuTual[41]。这里采用 3 类评价指标，包括精度性能、困惑度和召回率。其中数据集 ARC_challenge，HellaSwag，QNLI，PIQA 用于评估精度性能；Wiki，lambada_openai，lambada_standard 用于评估困惑度指标；数据集 MuTual 用于评估召回率。在实现方面，本文基于 PyTorch[42] 和 HuggingFace[43] 框架完成了对模型中所有线性层的权重量化实现，并统一采用无训练的后量化流程，以保证评估过程的公平与重现性。

2)对比方法设置

在实验中，本文将 LBS 与当前代表性的方法进

行了对比，包括 SmoothQuant[13]，Tender[14]，FP16。其中，SmoothQuant 通过通道缩放环节激活分布的长尾问题，将量化难点转移至权重，从而支持大模型在 8b 位宽下的有效量化。Tender 基于张量分解与 2 的幂次缩放，在无需混合精度的条件下实现加速推理，并隐式完成再量化操作，从而提升性能和准确性。FP16 作为高精度量化基线，用于参考精度性能上限。

4.2 精度实验结果

表 1 展示了 LBS 在 OPT-2.7B, OPT-6.7B, OPT-13B 模型中的性能表现。LBS(INT6)在关键任务中展现出显著优势：在 HellaSwag 任务中，LBS 在 OPT-6.7B 模型上达到 48.00% 的精度，优于 FP16 基线(47.75%)、SmoothQuant(INT6)(47.91%)以及 Tender(INT6)(47.58%)；在 QNLI 任务中，LBS(INT8)在 OPT-13B 模型上实现 57.00% 的精度，比基线(55.66%)高 1.34 个百分点，同时在 OPT-6.7B 模型中以 52.00% 的精度，高于 Tender(50.00%)达 2 个百分点；在 PIQA 任务中，配置为 INT6 的 LBS 在 OPT-13B 模型上达到 75.25% 的精度，高于基线(75.00%)、SmoothQuant(67.91%)以及 Tender(75.00%)。上述提升的精度性能主要源于 LBS 的显著值隔离策略来提升量化能力，从而在量化中保留语义表达与决策影响，并且 LBS 未显著降低精度性能，有效兼顾数值表达能力与模型推理性能。

表 2 系统展示了 LBS 在 LLAMA 模型上的实验结果。整体来看，LBS 的性能在多数任务和配置下与 Tender 方法相近，但在特定任务场景中具有更强的适应性和表现力。例如，在 LLAMA-2.7B 模型的 INT6 位宽配置下，LBS 在 QNLI 任务中达到 50.75% 的精度，优于 SmoothQuant(48.75%)和 Tender(50.25%)。此外，在 LLAMA-2.13B 模型的 INT8 配置下，LBS 在 ARC 推理任务中实现了 52.21% 的精度，优于所有对比方法，这是由于 LBS 能够通过充分利用显著值和非显著值以提升精度性能。

Table 1 Accuracy Comparison of INT8/INT6 (g=8, m=2) Post-Training Quantization on OPT Model

表 1 OPT 模型的 INT8/INT6（g=8, m=2）后训练量化精度对比 %

模型	数据集	基线	SmoothQuant		Tender		LBS（本文）	
		FP16	INT8	INT6	INT8	INT6	INT8	INT6
OPT-2.7B	ARC_challenge	26.79	27.02	26.02	26.70	27.21	27.64	26.64
	HellaSwag	43.91	44.25	44.00	44.16	44.25	45.00	44.75
	QNLI	51.75	51.58	52.83	51.25	50.83	51.33	51.50
	PIQA	73.83	73.91	72.33	73.16	74.08	74.50	73.00
OPT-6.7B	ARC_challenge	30.54	30.54	29.77	30.88	30.80	31.48	31.48
	HellaSwag	47.75	48.00	47.91	47.75	47.58	48.41	48.00
	QNLI	50.08	49.41	50.66	50.00	50.83	52.00	49.83
	PIQA	75.91	75.75	73.16	75.58	75.00	76.75	75.75
OPT-13B	ARC_challenge	32.84	32.84	25.68	32.76	32.84	34.21	34.21
	HellaSwag	49.08	48.16	40.58	48.83	48.66	48.08	47.83
	QNLI	55.66	56.00	50.91	56.66	54.91	57.00	54.58
	PIQA	75.00	74.91	67.91	75.41	75.00	74.91	75.25

Table 2 Accuracy Comparison of LLAMA Model (g=8, m=2) Under Different Quantization Methods

表 2 不同量化方法下 LLAMA 模型（g=8, m=2）的精度对比 %

量化位宽	方法	LLAMA-2.7B				LLAMA-2.13B			
		ARC_challenge	HellaSwag	QNLI	PIQA	ARC_challenge	HellaSwag	QNLI	PIQA
FP16	基线	43.43	50.08	49.83	77.50	48.46	51.75	48.75	78.16
INT8	SmoothQuant	43.51	50.66	49.16	77.83	48.46	52.25	48.83	78.08
	Tender	43.17	50.50	50.16	77.75	48.46	51.83	48.75	78.41
	LBS（本文）	43.17	50.62	51.00	77.41	52.21	52.08	48.66	78.16
INT6	SmoothQuant	43.08	49.41	48.75	76.50	45.05	50.58	48.66	76.85
	Tender	43.17	50.16	50.25	77.08	47.69	51.83	48.83	78.08
	LBS（本文）	42.57	49.41	50.75	77.50	50.00	50.41	48.66	77.08

表 3 进一步展示了在更低量化位宽(W4A4 配置)下, LBS 在 Wiki, lambada_openai(OpenAi), lambada_standard(Standard)数据集上的困惑度表现。实验结果表明, LBS 在多数模型和数据集中表现较好。例如, 在困惑度评比中, LBS 在低比特(INT4)下达到了和 FP16 相同水平的性能, 如 OPT-13B 模型的 OpenAi 任务中, 其困惑度为 4.15(略优于 FP16 的 4.38)。这主要源于 2 个方面: 首先, LBS 通过显著值隔离策略, 有效提升了普通值的量化分辨率, 从而改善整体的数值表达能力; 另一方面, 显著值作为模型中的关键特征, 经过放大后可避免在低比特量化过程中被削弱, 从而更准确地保留其语义表达与决策影响, 显著增强模型在推理任务中的表现。

Table 3 Perplexity Comparison of INT4 (*g*=2, *m*=1) of OPT Model After Trained on Different Datasets

表 3 OPT 模型的 INT4 (*g*=2, *m*=1) 在不同数据集训练后的困惑度对比

模型	方法	数据集		
		Wiki	OpenAi	Standard
OPT-2.7B	FP16	15.25	6.33	8.14
	SmoothQuant	2E+4	2.9E+6	3.3E+7
	Tender	19.00	7.72	13.04
	LBS (本文)	18.76	8.97	12.15
OPT-6.7B	FP16	12.99	4.84	5.50
	SmoothQuant	-	-	-
	Tender	15.22	7.72	10.07
	LBS (本文)	14.24	5.25	6.59
OPT-13B	FP16	11.31	4.38	4.98
	SmoothQuant	-	-	-
	Tender	12.91	5.42	5.77
	LBS (本文)	12.47	4.15	5.38

注: "-"表示困惑度数值远超合理范围。

为评估 LBS 在实际应用中的有效性, 本文在工业应用代表性基准 Wiki 数据集上进行了分析。实验结果如表 3 所示, 在所有测试模型规模上, LBS 于 INT4 位宽设定下获得的困惑度均低于 SmoothQuant 和 Tender。具体到 OPT-13B 模型, LBS 的困惑度为 12.47, 与未量化的 FP16 基线(11.31)差距为 1.16。此结果表明, LBS 能够在低比特(INT4)下, 将模型在通用文本任务上的性能损失控制在较低水平, 为大语言模型在资源受限环境下的部署提供了一种可行的技术路径。

表 4 展示了在 One-shot 任务中, MuTual 数据集上 LLAMA 系列模型的召回率(Recall, R)。实验结果表明针对 LLAMA2-7B 模型, LBS 方法保持与 Tender 相同的 *R*@1 和 *R*@2 值(24.16% 和 43.50%)的基础上, MRR(mean reciprocal rank)仅下降 0.16 个百分点, 保持较为优异的整体性能表现。例如, 在 LLAMA-2-13B 模型中, LBS 方法的 MRR 达到 74.09%, 高于其他对比方法。在该方法下, LBS 的 *R*@1 与 Tender 保持一致(24.16%), *R*@2 则为 42.16%, 仅比 Tender 略低 0.17 个百分点。相较于 SmoothQuant 方法, LBS 在多个指标上均取得更高性能, 体现其在大模型场景中具备的鲁棒性与适应性。更高的召回率是由于 LBS 通过分层量化处理显著值与普通值, 从而在低比特量化下实现模型推理性能的稳定提升。

Table 4 Recall of LLAMA Models on MuTual Dataset Under One-shot Learning Scenario (W8A8, *g*=8, *m*=2)

表 4 One-shot 学习场景下 (W8A8, *g*=8, *m*=2) LLAMA 模型在 MuTual 数据集上的召回率 %

模型	方法	*R*@1	*R*@2	MRR
LLAMA2-7B	FP16	24.16	44.00	72.84
	SmoothQuant	22.57	42.09	71.61
	Tender	24.16	43.50	72.91
	LBS (本文)	24.16	43.50	72.75
LLAMA2-13B	FP16	24.16	42.66	73.22
	SmoothQuant	22.50	40.29	73.22
	Tender	24.16	42.33	73.81
	LBS (本文)	24.16	42.16	74.09

表 5 进一步对比了不同量化方法的访存效率(memory access efficiency)。其中共享位宽 *m* 为 2, 所有的结果正则化到 FP16。相比于 FP16, LBS 方法在该配置下实现了最高 2.56 倍的访存效率。这源于共享比特所带来的访存规模下降。其通过组内多数据

Table 5 Memory Access Efficiency of Different Quantization Methods Under W8A8 Configuration

表 5 不同量化方法在 W8A8 配置下的访存效率

方法	位宽配置	访存效率提升倍数
FP16	-	1
FP8	-	2
SmoothQuant	W8A8	2
Tender	W8A8	2
LBS (本文)	W8A8	2.56

注: "-"表示无需量化位宽配置。

复用低比特量化值的权重共享策略，有效降低存储传输量与访存开销，相较 Tender 和 SmoothQuant 在访存效率上同样具有优势。

4.3 消融实验结果

为验证算法框架内显著值保护机制(O_1:Top-K)与比特共享策略(O_2:Only LBS)的独立贡献与协同效应，本文开展了消融实验。如表 6 所示，单独应用比特共享(INT6+O_2)会导致模型性能严重下降(如 HellaSwag 任务从 47.58% 降至 29.08%)，而单独的显著值保护(INT6+O_1)则能有效提升性能至 50.00%。当二者结合(O_1+O_2)时，性能恢复至 48.00% 的高水平。这有力地证明，显著值保护弥补了比特共享在处理离群值时的不足，二者形成了互补的协同机制，是实现高精度量化的根本保障。

4.4 组设计空间探索

在 LBS 方法中，组大小(g)是一个对模型精度有关键影响的超参数。本文通过实验探索了其最优配置，结果如图 4 所示。从图中可以清晰地看到，不同量化位宽对分组大小的敏感度存在显著差异。INT8 与 INT6 的曲线均在组大小 g=8 处达到一个明确的性能峰值。然而，INT4 的曲线则呈现出截然不同的趋势：其性能在 g=2 时表现最佳，随后总体下降。这种差异源于不同位宽对量化误差的敏感度：INT4 的位宽极小，对因分组引入的误差高度敏感，因此需要更小的分组来保证精度。基于这一观察，本文采用了一种自适应的分组策略：为 INT8 和 INT6 量化设置 g=8，而为更敏感的 INT4 量化设置 g=2。这种针对性的配置，旨在最大化每种量化位宽下的模型性能。

Table 6 Ablation Accuracy of OPT-6.7B Model Under Different Combinations of Quantization Strategies

表 6 不同量化策略组合下 OPT-6.7B 模型的消融精度 %

数据集	方法							
	INT8	INT8+O_1	INT8+ O_2	O_1+ O_2（8b）	INT6	INT6+ O_1	INT6+ O_2	O_1+ O_2（6b）
ARC_challenge	30.88	32.00	30.00	31.48	30.80	32.00	19.70	31.48
HellaSwag	49.00	52.00	47.83	48.41	47.58	50.00	29.08	48.00
QNLI	50.00	52.00	48.83	52.00	50.83	52.01	45.85	49.83
PIQA	77.00	80.00	75.50	76.75	75.00	80.00	54.91	75.75

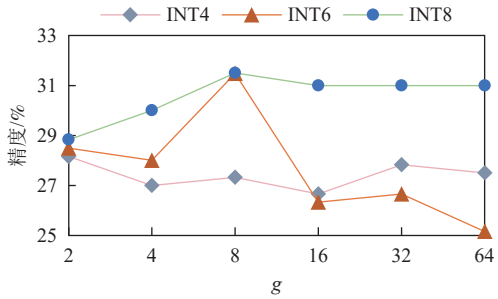


Fig. 4 Group size dependence analysis of quantization strategy on accuracy of OPT-6.7B model in ARC_challenge dataset

图 4 量化策略对 OPT-6.7B 模型在 ARC_challenge 数据集精度的组大小依赖性分析

5 结 论

本文提出了一种低比特共享(LBS)量化框架。该方法在传统整型量化机制基础上引入组比特复用与显著值感知策略，实现了对存储冗余的深度量化，同时有效抑制了因位宽降低所引发的精度损失。通过结构化的高低位分离与动态共享机制，LBS 不仅显著减少了内存访问成本，还结合 Top-K 驱动的显著权重识别与比例放大策略，增强了模型对关键权重的保持能力，提升了量化模型在复杂任务中的精度鲁棒性。实验证明，LBS 在多个大语言模型与典型下游任务中，在精度性能和访存效率上优于现有主流方法。

作者贡献声明：李林浩提出了本文的核心思想，负责算法设计、主要实验验证以及论文的撰写；曾宣迪参与了部分实验工作；唐忆滨和武继刚参与了论文的结果分析和讨论与修订；刘博生对整体研究工作提供了关键指导，并对论文进行了最终审阅和定稿。

参 考 文 献

[1] Karimi A, Rossi L, Prati A. Improving bert performance for aspect-basedsentiment analysis[J]. arXiv preprint, arXiv: 2010.11731, 2020

[2] Kirstein F, Khan M, Wahle J P, et al. You need to MIMIC to get FAME: Solving meeting transcript scarcity with multi-agent conversations[C]//Findings of the Association for Computational Linguistics: ACL 2025. Stroudsburg: Association for Computational

Linguistics, 2025: 11482-11525

[3] Samarinas C, Krubner A, Salemi A, et al. Beyond factual accuracy: Evaluating coverage of diverse factual information in long-form text generation[J]. arXiv preprint, arXiv: 2501.03545, 2025

[4] Bevilacqua M, Oketch K, Qin Ruiyang, et al. When automated assessment meets automated content generation: Examining text quality in the era of GPTs[J]. ACM Transactions on Information Systems, 2025, 43(2): 1−36

[5] Zhang Han, Meng Zixiang, Luo Meng, et al. Towards multimodal empathetic response generation: A rich text-speech-vision avatar-based benchmark[J]. arXiv preprint, arXiv: 2502.04976, 2025

[6] Wei Feng, Zhang Xu. Few-Shot contrastive learning-based multi-round dialogue intent classification method[J]. Expert Systems, 2025, 42(2): e13771

[7] Liu Zijie, Zhao Xinyu, Peng Jie, et al. Dialogue is better than monologue: Instructing medical LLMs via strategical conversations[J]. arXiv preprint, arXiv: 2501.17860, 2025

[8] Pang Jianhui, Ye Fanghua, Wang Longyue, et al. Salute the classic: Revisiting challenges of nachine translation in the age of large language models[J]. Transactions of the Association for Computational Linguistics, 2025, 13: 73−95

[9] Zhu Shaolin, Pan Leiyu, Jian Dong, et al. Overcoming language barriers via machine translation with sparse mixture-of-experts fusion of large language models[J]. Information Processing & Management, 2025, 62(3): 104078

[10] Kumar S. Evaluating full INT8 quantization and inference techniques for causal language model[D]. Enschede, Netherlands: University of Twente, 2025

[11] Fei Wen, Dai Wenrui, Zhang Liang, et al. Latent weight quantization for integerized training of deep neural networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025, 47(4): 2816−2832

[12] Nair P, Datta P, Dean J, et al. Matryoshka quantization[J]. arXiv preprint, arXiv: 2502.06786, 2025

[13] Xiao Guangxuan, Lin Ji, Seznec M, et al. SmoothQuant: Accurate and efficient post-training quantization for large language models[C]// Proc of the 23rd Int Conf on Machine Learning. New York: PMLR, 2023: 38087−38099

[14] Lee J, Lee W, Sim J. Tender: Accelerating large language models via tensor decomposition and runtime requantization[C]//Proc of the 51st Annual Int Symp on Computer Architecture (ISCA). Piscataway, NJ: IEEE, 2024: 1048−1062

[15] Yue Yuxuan, Yuan Zhihang, Duanmu Haojie, et al. Wkvquant: Quantizing weight and key/value cache for large language models gains more[J]. arXiv preprint, arXiv: 2402.12065, 2024

[16] Yu Hao, Zhou Yang, Chen Bohua, et al. Treasures in discarded weights for LLM quantization[C]//Proc of the 39th Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2025: 22218−22226

[17] Lin Ji, Tang Jiaming, Tang Haotian, et al. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration[C]//Proc of the 6th Machine Learning and Systems. New York: ACM, 2024: 12−17

[18] Wang Jinguang, Wang Jingyu, Sun Haifeng, et al. MergeQuant: Accurate 4-bit static quantization of large language models by channel-wise calibration[J]. arXiv preprint, arXiv: 2503.07654, 2025

[19] Hu Yuxuan, Chen Xiaodong, Li Cuiping, et al. QUAD: Quantization and parameter-efficient tuning of LLM with activation decomposition[J]. arXiv preprint, arXiv: 2503.19353, 2025

[20] Choi E, Song Sumin, Lim W, et al. Rotate, clip, and partition: Towards W2A4KV4 quantization by integrating rotation and learnable non-uniform quantizer[J]. arXiv preprint, arXiv: 2502.15779, 2025

[21] Fang Xiwei, Wang Yuhan, Chen Lei, et al. A reconfigurable floating-point division and square root architecture for high-precision softmax[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2025, 72(9): 4737−4750

[22] Mary T, Mikaitis M. Error analysis of matrix multiplication with narrow range floating-point arithmetic[J]. SIAM Journal on Scientific Computing, 2025, 47(4): B785−B800

[23] Mansour Y, Kaissar A, Ansari S. Review on recent matrix multiplication optimization using deep learning[C]//Proc of the Int Conf on Intelligent and Fuzzy Systems. Cham: Springer, 2024: 359−371

[24] Sakai Y, Tamiya Y. S-DFP: Shifted dynamic fixed point for quantized deep neural network training[J]. Neural Computing and Applications, 2025, 37(2): 535−542

[25] Rahman M. Estimating word lengths for fixed-point DSP implementations using polynomial chaos expansions[D]. Hamilton: McMaster University, 2023

[26] Peng Houwen, Wu Kan, Wei Yixuan, et al. FP8-LM: Training FP8 large language models[J]. arXiv preprint, arXiv: 2310.18313, 2023

[27] Guo Cong, Zhang Chen, Leng Jingwen, et al. ANT: Exploiting adaptive numerical data type for low-bit deep neural network quantization[C]//Proc of the 55th Int Symp on Microarchitecture (MICRO). Piscataway, NJ : IEEE, 2022: 1414−1433

[28] Shao Wenqi, Chen Mengzhao, Zhang Zhaoyang, et al. OMNIQUANT: Omnidirectionally calibrated quantization for large language models[C/OL]//Proc of the 12th Int Conf on Learning Representations (ICLR). Vienna: ICLR, 2024[2025-07-29]. https://iclr.cc/virtual/2024/poster/19323

[29] Lü Qianru, Xu Jinwei, Jiang Jingfei, et al. DAQ: Divide-and-conquer strategy based adaptive low-bit quantization method for vision transformer[J]. Journal of Computer Research and Development, 2025, 62(6): 1530−1546 (in Chinese)
(吕倩茹, 许金伟, 姜晶菲, 等. DAQ: 基于分治策略的自适应 Vision Transformer 低位宽量化方法[J]. 计算机研究与发展, 2025, 62(6): 1530−1546)

[30] Ashkboos S, Mohtashami A, Croci M, et al. QuaRot: Outlier-free 4-bit inference in rotated LLMs[C]//Proc of the 38th Int Conf on Neural Information Processing Systems(NeurIPS). New York: ACM, 2024: 100213−100240.

[31] An Yongqi, Zhao Xu, Yu Tao, et al. Systematic outliers in large language models[J]. arXiv preprint, arXiv: 2502.06415, 2025

[32] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proc of the 31st Conf on Neural Information Processing Systems (NIPS). Red Hook, NY: Curran Associates Inc, 2017: 5998−6008

[33] Zhang Susan, Roller S, Goyal N, et al. OPT: Open pre-trained transformer language models[J]. arXiv preprint, arXiv: 2205.01068, 2022

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

[35] Bhakthavatsalam S, Khashabi D, Khot T, et al. Think you have solved direct-answer question answering ? Try ARC-DA, the direct-answer AI2 reasoning challenge[J]. arXiv preprint, arXiv: 2102.03315, 2021

[36] Zellers R, Holtzman A, Bisk Y, et al. HellaSwag: Can a machine really finish your sentence?[J]. arXiv preprint, arXiv: 1905.07830, 2019

[37] Wang A, Singh A, Michael J, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding[J]. arXiv preprint, arXiv: 1804.07461, 2018

[38] Bisk Y, Zellers R, Bras L, et al. PIQA: Reasoning about physical commonsense in natural language[C]//Proc of the 34th Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2020: 7432−7439

[39] Lim D, Hohne F, Li X, et al. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods[C]//Proc of the 35th Neural Information Processing Systems(NIPS). New York: ACM, 2021: 20887−20902

[40] Paperno D, Kruszewski G, Lazaridou A, et al. The LAMBADA dataset: Word prediction requiring a broad discourse context[J]. arXiv preprint, arXiv: 1606.06031, 2016

[41] Cui Leyang, Wu Yu, Liu Shujie, et al. MuTual: A dataset for multi-turn dialogue reasoning[C]//Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2020: 1406–1416

[42] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[C]//Proc of the 33rd Int conf on Neural Information Processing System (NIPS). New York: ACM, 2019: 8026−8037

[43] Wolf T, Debut L, Sanh V, et al. HuggingFace's transformers: State-of-the-art natural language processing[J]. arXiv preprint, arXiv: 1910.03771, 2019



Li Linhao, born in 2001. Master candidate. His main research interest includes hardware-software co-design based on deep learning.

李林浩, 2001 年生。硕士研究生。主要研究方向为基于深度学习的软硬件协同设计。



Zeng Xuandi, born in 2003. Undergraduate. His main research interest includes hardware-software co-design based on deep learning.

曾宣迪, 2003 年生。本科生。主要研究方向为基于深度学习的软硬件协同设计。



Tang Yibin, born in 1991. PhD, professor. His main research interests include the hardware-software co-design based on deep learning and computer architecture.

唐忆滨, 1991 年生。博士，研究员。主要研究方向为基于深度学习的软硬件协同设计、计算机体系结构。



Liu Bosheng, born in 1988. PhD, associate professor. His main research interests include hardware-software co-optimization for LLMs and computer architecture.

刘博生, 1988 年生。博士、副教授。主要研究方向为大语言模型软硬件协同优化、计算机体系结构。



Wu Jigang, born in 1963. PhD, PhD supervisor. His main research interests include AI, high-performance computing, and data science. (asjgwucn@outlook.com)

武继刚, 1963 年生。博士、博士生导师。主要研究方向为人工智能、高性能计算、数据科学。