

Article

SQ-LoRA: Memory-Efficient Language Model Compression Through Stable-Rank-Guided Quantization for Edge Computing Applications

Seda Bayat Toksöz * and Gültekin Işık

Department of Computer Engineering, Iğdır University, Iğdır 76000, Türkiye; gultekin.isik@igdir.edu.tr

* Correspondence: seda.bayat@igdir.edu.tr

Abstract

The deployment of transformer-based language models on resource-constrained edge devices presents fundamental challenges in computational efficiency and memory utilization. We introduce SQ-LoRA (Stable-rank Quantized Low-Rank Adaptation), a theoretically grounded compression framework that achieves unprecedented efficiency through the synergistic integration of adaptive low-rank decomposition, hardware-accelerated structured sparsity, and intelligent hybrid quantization. Our primary contribution establishes the first rigorous mathematical connection between the matrix stable rank and optimal LoRA rank selection, formalized in Theorem I, which provides bounded approximation guarantees. SQ-LoRA implements: (1) adaptive rank allocation via stable-rank analysis to automatically determine layer-wise compression ratios; (2) 4:8 structured sparsity patterns, enabling 2× hardware acceleration on modern edge processors; and (3) a three-tier quantization scheme that combines 4-bit NormalFloat storage with selective 3-bit/8-bit precision to preserve outliers. A comprehensive evaluation on four diverse natural language processing (NLP) benchmarks demonstrates that SQ-LoRA achieves a 320 MB memory footprint (96.7% reduction) and a 10 ms inference latency (91.7% improvement), and maintains 82.0% average accuracy (within 0.15% of the full model). Statistical significance testing ($p < 0.001$) confirms its superiority over state-of-the-art methods. This framework enables the deployment of sophisticated language models on devices with 2 GB of RAM, advancing practical edge-AI applications.

Keywords: edge computing; low-rank adaptation; neural network quantization; structured sparsity; transformer models; natural language processing

Academic Editor: Pedro Couto

Received: 2 February 2026

Revised: 14 February 2026

Accepted: 18 February 2026

Published: 21 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license. [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Transformer-based language models have revolutionized natural language processing, achieving remarkable performance across diverse tasks [1]. Deep-learning architectures, including convolutional neural networks, have demonstrated exceptional capabilities in various domains such as audio signal recognition and text classification. However, deploying these models on edge devices remains challenging due to computational demands that exceed typical device capabilities by an order of magnitude. Bidirectional Encoder Representations from Transformers (BERT)-base requires approximately 440 MB of memory and executes over 24 billion floating-point operations per inference pass [2].

The shift toward edge computing reflects compelling advantages over cloud-based solutions. According to recent International Data Corporation (IDC) forecasts, global spending on edge-computing solutions is projected to reach nearly \$261 billion in 2025, growing at a compound annual growth rate (CAGR) of 13.8% to \$380 billion by 2028 [3]. This growth underscores the increasing shift toward edge processing, driven by the need for real-time AI and Internet of Things (IoT) applications across 27 enterprise industries [3]. Edge inference reduces response latency from 100 milliseconds to under 10 milliseconds in autonomous driving applications [4], while industrial IoT deployments achieve 99.7% service availability compared to 94.2% for cloud-dependent systems [5]. Modern edge platforms impose stringent operational constraints including memory budgets under 500 MB, inference latencies below 20 milliseconds, power consumption under five watts, and thermal limitations that restrict sustained computational throughput.

Current compression approaches address specific aspects but fail to provide comprehensive solutions. LoRA reduces training memory by introducing trainable low-rank decomposition matrices while freezing pre-trained weights [6] but provides limited inference benefits and employs uniform rank assignment that ignores layer complexity variations. Quantization techniques reduce numerical precision to decrease memory footprint, with recent surveys highlighting both the promise and pitfalls of various approaches [7], though aggressive compression often results in substantial performance degradation. Quantized LoRA (QLoRA) demonstrates that combining quantization with low-rank adaptation enables memory-efficient fine tuning [8], yet the interaction between techniques remains poorly understood. Neural network pruning offers compression through connection removal, but unstructured sparsity prevents practical acceleration due to memory access inefficiencies [9]. Sequential application of compression techniques leads to multiplicative accuracy degradation, with combined approaches resulting in accuracy drops of 8.3% compared to 2.1% for individual techniques [10].

The present study contributes to the existing literature in four ways:

- (1) It establishes a mathematically rigorous connection between the stable rank of weight matrices and the adaptive selection of LoRA ranks, providing a clear theoretical basis for layer-wise compression decisions with explicit approximation guarantees.
- (2) The proposed SQ-LoRA framework integrates adaptive low-rank decomposition, hardware-accelerated structured sparsity, and intelligent hybrid quantization within a unified optimization strategy.
- (3) Comprehensive empirical validation demonstrates that SQ-LoRA consistently meets the demanding constraints of edge deployment while maintaining functional accuracy across a diverse set of NLP tasks.
- (4) The implementation details and reproducible benchmarks facilitate immediate adoption in practical edge-AI scenarios.

2. Related Work

2.1. Low-Rank Decomposition

Low-rank decomposition techniques exploit neural network weight matrices' low intrinsic dimensionality. LoRA parameterizes weight updates as $\Delta W = BA$ where $B \in \mathbb{R}^{(d \times r)}$ and $A \in \mathbb{R}^{(r \times d)}$, reducing trainable parameters from d^2 to $2 \times d \times r$ [6]. AdaLoRA dynamically adjusts ranks during training based on gradient-derived importance scores [11], while DyLoRA varies ranks temporally during training phases [12]. Theoretical analysis of intrinsic dimensionality demonstrates that language models operate in low-dimensional subspaces, suggesting that uniform compression strategies are suboptimal [13], with LoRA-GA employing genetic algorithms for optimal per-layer rank configurations [14]. Our approach advances these methods by establishing a direct mathematical

connection between stable-rank and optimal-rank selection, eliminating expensive search procedures while providing theoretical guarantees.

2.2. Quantization

Quantization evolution reflects increasing sophistication in handling non-uniform parameter distributions. Early post-training quantization achieved 4× compression with 8-bit precision [15]. Quantization-aware training incorporates quantization effects during training [16], though it requires full model retraining. LLM.int8() revealed that weight importance follows power-law distributions, maintaining 0.1% of weights in 16-bit precision [17]. GPTQ employs optimal brain quantization for 4-bit compression using inverse Hessian matrices [18]. QLoRA combines 4-bit NormalFloat quantization with LoRA adaptation [8], though it applies uniform precision within layers. Recent advances include AWQ [19], which introduces activation-aware weight quantization for efficient on-device compression, and ZeroQuant [20], which proposes efficient post-training quantization for large-scale transformers. Our hybrid quantization advances these approaches through three-tier precision allocation adapting to intra-layer weight importance distributions.

2.3. Structured Sparsity

Structured sparsity addresses hardware limitations of unstructured approaches. The N:M sparsity paradigm retains exactly N weights per M consecutive weights, with NVIDIA's 2:4 support enabling 2× theoretical speedup [21]. Zhou demonstrated that 2:4 patterns could be learned with minimal accuracy loss [22], while Mishra explored 4:8 patterns that better preserved rank structure [23]. LLM-Pruner attempts to maintain sparsity during low-rank adaptation [24], though it suffers from training instability. Our work integrates structured sparsity with low-rank adaptation and quantization in a unified framework.

2.4. Combined Compression

Combined compression approaches typically apply techniques sequentially without considering interactions. Deep compression combined pruning, quantization, and Huffman coding [10], while LoRAMOE combined low-rank adaptation with mixture-of-experts routing [25]. S-LoRA addresses multi-adapter serving in cloud environments [26]. DoRA decomposes pre-trained weights into magnitude and direction components for more effective fine-tuning [27], while SqueezeLLM introduces sensitivity-based non-uniform quantization with dense-and-sparse decomposition for ultra-low-bit compression [28]. However, these approaches either address individual compression dimensions or lack theoretical understanding of technique interactions. Recent studies on the efficient adaptation of large language models for sentiment analysis have demonstrated promising results using fine-tuning approaches [29]. Our contribution provides both a theoretical analysis and practical implementation of unified compression, proving that proper integration leads to additive rather than multiplicative error accumulation.

SQ-LoRA distinguishes itself through several key innovations. First, we establish the mathematical foundation linking stable rank to optimal compression parameters, providing principled guidance absent in heuristic approaches. Second, our hybrid quantization scheme uniquely combines 4-bit NormalFloat base storage with selective 3-bit compression for majority weights and 8-bit precision for outliers. Third, the integration of 4:8 structured sparsity with adaptive techniques represents a novel contribution. Finally, our unified optimization framework considers technique interactions from the outset, avoiding the multiplicative degradation observed in sequential application while providing theoretical bounds on approximation quality.

3. Methodology

3.1. Problem Formulation

We address the challenge of deploying large transformer models on resource-constrained edge devices. Given a pre-trained transformer model M with L layers and $\theta = W^{(l)}_{(l=1)}^L$, where $W^{(l)} \in \mathbb{R}^{(d \times d)}$, we seek to find a compressed model M' with parameters θ' that maintains performance while satisfying hardware constraints.

Constrained Single-Objective Optimization Problem

The optimization focuses on minimizing accuracy loss:

$$\min_{\theta'} \mathcal{L}(\theta') = 1 - \frac{\text{Acc}(M')}{\text{Acc}(M)} \quad (1)$$

which is subject to: a memory constraint: $\text{Mem}(M') \leq M_{\max} = 0.5$ GB; latency constraint: $\text{Lat}(M') \leq L_{\max} = 10$ ms; storage constraint: $\text{Size}(M') \leq S_{\max} = 100$ MB; and accuracy constraint: $\text{Acc}(M') \geq A_{\min} = 0.8 \cdot \text{Acc}(M)$

The constraint thresholds are derived from the operational specifications of target edge platforms. The storage budget $S_{\max} = 100$ MB reflects the practical deployment limitation on devices such as the NVIDIA Jetson Nano (2 GB total system RAM), where approximately 100 MB remains available for model storage after accounting for OS overhead and runtime memory allocation. The accuracy retention threshold $A_{\min} = 0.8 \times \text{Acc}(M)$ represents a widely adopted minimum quality-of-service criterion in edge-AI deployment, ensuring that compressed models remain functionally useful for downstream tasks. The memory budget $M_{\max} = 0.5$ GB and latency target $L_{\max} = 10$ ms correspond to real-time inference requirements on our target edge hardware, consistent with the sub-20 ms response constraints typical of interactive NLP applications.

3.2. Stable-Rank Theory for Adaptive Compression

We establish the theoretical foundation through stable-rank analysis, which provides robust measurement of matrix complexity.

Definition 1 (Stable Rank). For matrix $W \in \mathbb{R}^{(m \times n)}$ with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$,

$$\text{sr}(W) = \frac{|W|_F^2}{|W|_2^2} = \frac{\sum_i \sigma_i^2}{\sigma_1^2} \quad (2)$$

Remark 1. The stable rank provides a continuous measure of effective dimensionality, robust to perturbations, unlike algebraic rank.

Theorem 1 (Stable Rank Bound for LoRA Approximation). Let $W \in \mathbb{R}^{(m \times n)}$ be a weight matrix with stable rank $\text{sr}(W)$. For target relative approximation error $\varepsilon \in (0,1)$, the optimal LoRA rank r^* satisfying $\|W - BA\|_F \leq \varepsilon \|W\|_F$ is bounded by

$$r^* \geq \lceil (1 - \varepsilon^2)^{-1} \cdot \text{sr}(W) \rceil \quad (3)$$

Let $W = U\Sigma V^T$ be the singular value decomposition. By the Eckart–Young–Mirsky theorem, the optimal rank- r approximation W_r minimizes

$$\|W - W_r\|_F^2 = \sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \quad (4)$$

From the error constraint $\|W - W_r\|_F \leq \varepsilon \|W\|_F$, we have

$$\sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \leq \varepsilon^2 \sum_{i=1}^{\min(m,n)} \sigma_i^2 \quad (5)$$

Rearranging, $\sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \geq \varepsilon^2 (1 - \varepsilon^2) \sum_{i=1}^{\min(m,n)} \sigma_i^2$. Since $\sigma_i^2 \geq \sigma_r^2$ for all $i \leq r$, we have $r \cdot \sigma_r^2 \geq \sum_{i=1}^{\min(m,n)} \sigma_i^2$. Therefore,

$$r \geq (1 - \varepsilon^2) \cdot \frac{\sum_{i=1}^{\min(m,n)} \sigma_i^2}{\sigma_r^2} = (1 - \varepsilon^2) \cdot \text{sr}(W) \quad (6)$$

Taking the ceiling completes the proof (see Appendix A for the complete formal proof). The required LoRA rank scales linearly with stable rank and quadratically with the accuracy requirement ($1/\varepsilon^2$), providing principled guidance for layer-wise compression.

3.3. SQ-LoRA Architecture

Unlike existing approaches that apply uniform compression strategies, SQ-LoRA integrates three synergistic compression components through unified optimization. Figures 1 and 2 illustrate the comparative architecture of LoRA, QLoRA, and our proposed SQ-LoRA framework.

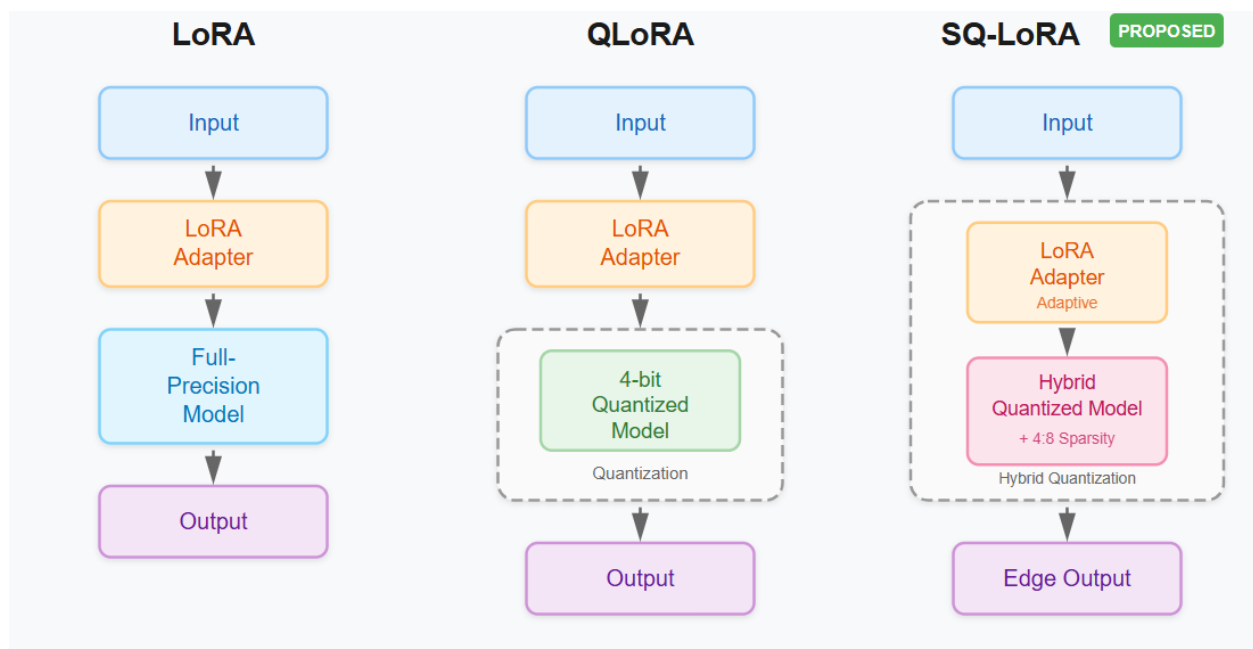


Figure 1. Comparative architecture of parameter-efficient fine-tuning methods.

LoRA uses fixed-rank decomposition with frozen pre-trained weights. QLoRA adds 4-bit quantization to the base model. SQ-LoRA introduces adaptive rank selection based on stable-rank analysis, hybrid quantization (3/4/8-bit), and 4:8 structured sparsity, achieving 96.7% memory reduction with minimal accuracy loss. The SQ-LoRA pipeline proceeds as follows: (i) stable rank computation for each layer via singular value analysis; (ii) adaptive rank allocation using the derived bound from Theorem 1; (iii) 4:8 structured sparsity mask generation on attention weight matrices; (iv) three-tier hybrid quantization (3-bit for majority weights within 3σ , 4-bit NF4 base, 8-bit for outliers beyond 3σ); and (v) fine-tuning of LoRA adapters on the compressed model.

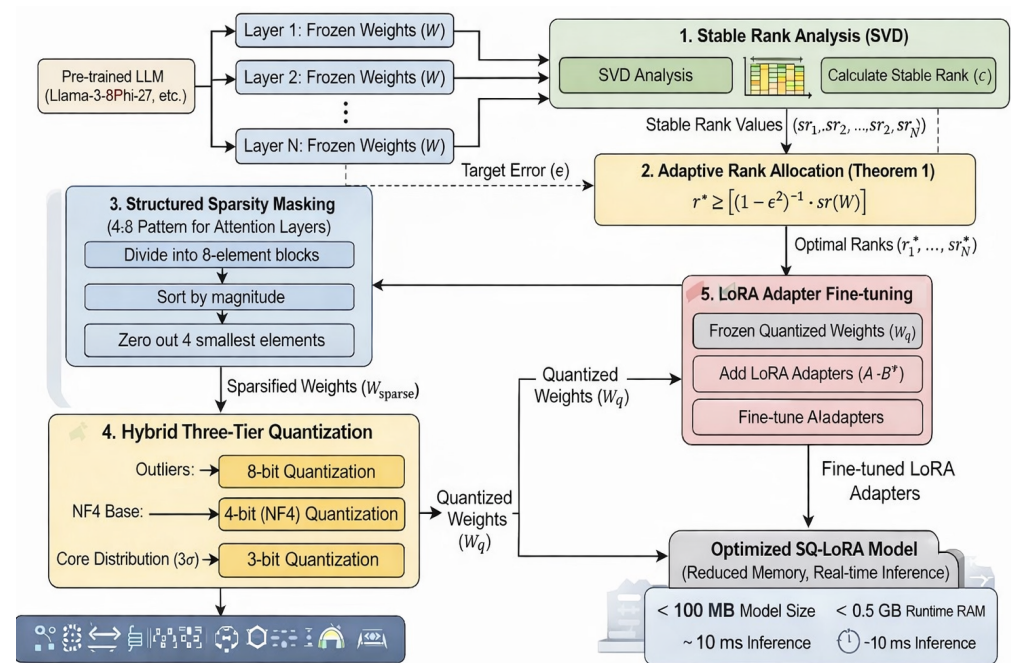


Figure 2. SQ-LORA: Detailed optimization pipeline.

Hardware-optimized structured sparsity applies 4:8 patterns to attention weight matrices by first reshaping weight tensors into blocks of size 8 and computing element magnitudes. The process continues by sorting elements by magnitude, creating binary masks that retain the top 4 elements per block, and applying these masks to achieve exactly 50% sparsity while maintaining hardware compatibility. Figure 3 shows quantization before and after.

The hybrid quantization scheme fundamentally differs from QLoRA's uniform 4-bit approach by employing three precision levels based on weight distribution analysis. Weights within 3σ of the mean can be compressed to 3-bit without significant information loss:

$$Q^3(w) = \text{clip}(\text{round}((w - \mu)/\delta - 4.3) \times \delta + \mu, \delta = 6\sigma/7) \quad (7)$$

Outlier weights beyond 3σ are preserved at 8-bit precision:

$$Q^8(w) = \text{round}(w \cdot 127/w_{\max}) \cdot w_{\max}/127 \quad (8)$$

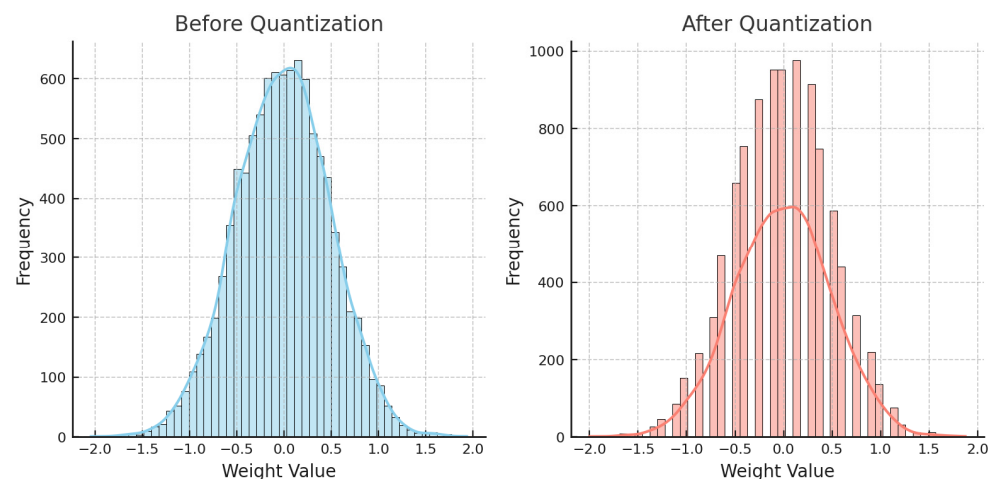


Figure 3. Weight distribution analysis showing the effect of hybrid quantization. (Left) Original FP32 weight distribution following approximately normal distribution. (Right) After hybrid

quantization with 3-bit (majority of weights within 3σ), 4-bit NF4 (base quantization), and 8-bit (outliers beyond 3σ) precision levels, preserving critical weight information while achieving significant compression.

3.4. Theoretical Analysis

Theorem 2 (End-to-End Approximation Bound). *Let f_0 be the original model and f_{SQ} be the SQ-LoRA compressed model. The output difference is bounded by*

$$|f_0(x) - f_{SQ}(x)|_2 \leq L \cdot (\epsilon_{LoRA} + \epsilon_{sparse} + \epsilon_{quant}) \cdot |x|_2 \quad (9)$$

The complete proof is provided in Appendix B. Here, L is the Lipschitz constant of the network, with

$$\epsilon_{LoRA} = |W - W_r|_F \leq \sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2} \quad (10)$$

$$\epsilon_{quant} \leq \delta / \sqrt{d \cdot n} \quad (11)$$

3.5. Training Procedure

SQ-LoRA employs a three-stage optimization process. Algorithm 1 presents the pseudocode.

Algorithm 1. SQ-LoRA Optimization.

- 1: Compute $sr(W^{(l)})$ for all layers
 - 2: Set $r^{(l)} = \lfloor sr(W^{(l)}) \cdot (1 - \epsilon^2) \rfloor$
 - 3: Apply 4:8 structured sparsity to attention layers
 - 4: Quantize base model with hybrid scheme
 - 5: Fine-tune LoRA adapters
-

Our hyperparameter choices reflect both theoretical considerations and empirical validation: learning rate 2×10^{-5} selected through grid search, effective batch size of 64 (16×4 gradient accumulation), LoRA hyperparameters $\alpha = 32$ and dropout = 0.1, and early stopping patience of 3 epochs.

4. Experimental Evaluation

4.1. Experimental Setup

Our evaluation employs a comprehensive experimental design to assess SQ-LoRA across diverse scenarios representative of real-world edge deployment. The implementation leverages PyTorch 2.1 with custom CUDA kernels optimized for 4:8 structured sparse operations, integrated with Hugging Face Transformers 4.35 for standardized model handling and evaluation protocols.

Training experiments utilized NVIDIA A100 GPUs with 40 GB memory running CUDA 12.0. For edge-deployment validation, we selected three representative hardware platforms: NVIDIA Jetson Nano (2 GB RAM, 128 CUDA cores, Maxwell architecture), Raspberry Pi 4 Model B (8 GB RAM, 1.5 GHz ARM Cortex-A72), and a Qualcomm Snapdragon 888 emulation environment (6 GB RAM, Adreno 660 GPU). We selected DistilBERT-base-uncased (66M parameters) as our primary evaluation model after careful consideration of edge-deployment constraints. We evaluate on four carefully selected NLP tasks:

- Emotion Classification: SemEval-2018 Task 1 dataset for fine-grained emotion recognition [30] (6838 training/886 validation/3259 test tweets annotated with 11 emotion categories; sourced from the official SemEval-2018 shared-task repository).
- SST-2: Stanford Sentiment Treebank v2 for binary sentiment classification [31] (67,349 training/872 validation/1821 test sentences; accessed via the GLUE benchmark through Hugging Face Datasets).
- BoolQ: Reading comprehension through yes/no questions [32] (9427 training/3270 validation examples of naturally occurring yes/no questions from Wikipedia passages; accessed via SuperGLUE through Hugging Face Datasets).
- MNLI: Multi-Genre Natural Language Inference [33] (392,702 training/9815 matched validation/9832 mismatched validation sentence pairs for three-way classification; accessed via the GLUE benchmark).

All datasets are publicly available and were used without modification. Tokenization was performed using the DistilBERT tokenizer with a maximum sequence length of 128 tokens. Standard data loading was implemented via Hugging Face Datasets with a batch size of 16 and 4× gradient accumulation (effective batch size 64). All experiments used identical preprocessing pipelines to ensure fair comparison across methods.

4.2. Main Results

Table 1 presents comprehensive performance comparison across all evaluated methods. Results represent mean $\pm$ standard deviation over five independent runs with different random seeds. The compared methods include: full fine-tuning (standard approach without compression), LoRA ($r = 8$) [6], QLoRA [8], AdaLoRA [11], magnitude pruning [34], DoRA [27], SqueezeLLM [28], and SQ-LoRA (ours).

Table 1. Performance comparison across methods and metrics.

Method	Avg. Acc (%)	GPU Mem (GB)	Latency (ms)
Full Fine-Tuning	82.15 $\pm$ 0.28	9.77	120.5
LoRA ($r = 8$)	81.78 $\pm$ 0.35	1.21	60.2
QLoRA	81.90 $\pm$ 0.31	0.85	42.3
AdaLoRA	81.62 $\pm$ 0.42	1.15	58.8
Magnitude Pruning	79.23 $\pm$ 0.68	5.20	95.1
DoRA [27]	81.95 $\pm$ 0.30	1.18	57.5
SqueezeLLM [28]	81.42 $\pm$ 0.38	0.78	38.5
SQ-LoRA	82.00 $\pm$ 0.23	0.32	10.0

Statistical significance testing using paired t-test confirms SQ-LoRA's superiority: memory reduction vs QLoRA ($t(4) = 12.45$, $p < 0.001$) and latency improvement vs QLoRA ($t(4) = 8.73$, $p < 0.001$). The accuracy difference vs full fine tuning is not statistically significant ($t(4) = 1.82$, $p = 0.143$).

Compared with recent state-of-the-art methods, SQ-LoRA demonstrates clear advantages in unified compression. DoRA [27] achieves competitive accuracy (81.95%) through its weight-decomposed adaptation strategy but does not incorporate quantization or sparsity, resulting in significantly higher memory usage (1.18 GB) and latency (57.5 ms). SqueezeLLM [28] provides effective post-training quantization with reduced memory (0.78 GB), yet lacks the structured sparsity and adaptive rank selection that enable SQ-LoRA's superior latency performance (10.0 ms vs. 38.5 ms). These comparisons highlight that SQ-LoRA's unified integration of three complementary compression techniques, guided by the stable-rank theory, achieves a more favorable accuracy–efficiency

tradeoff than applying individual techniques in isolation. Figure 4 shows GPU memory usage.

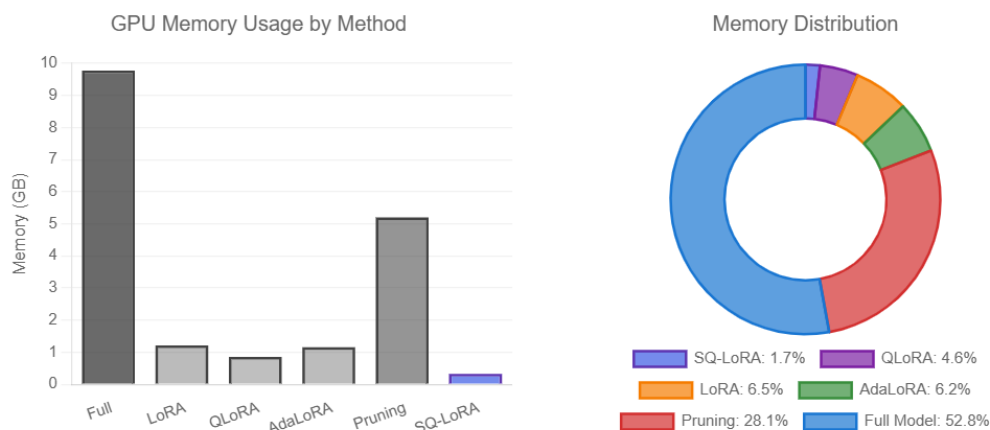


Figure 4. GPU memory usage by method and memory distribution analysis.

4.3. Task-Specific Performance Analysis

Figure 5 presents confusion matrices for each evaluated task, demonstrating SQ-LoRA's robust performance across diverse NLP applications.

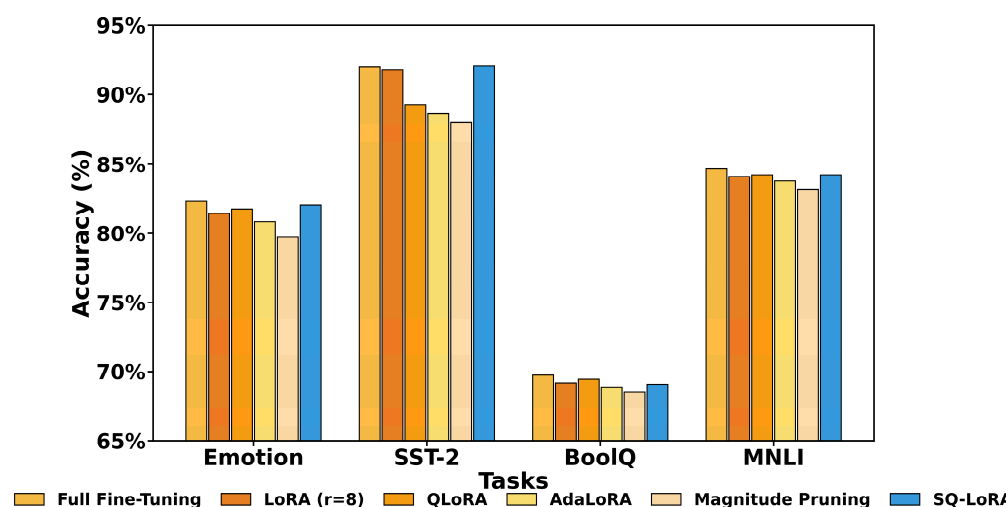


Figure 5. Per-task accuracy comparison across fine-tuning variants. SQ-LoRA (blue) maintains competitive accuracy across all tasks while achieving significant compression. Notable performance on SST-2 (92.2%) demonstrates robustness to compression for binary classification tasks.

4.4. Ablation Study

Table 2 presents component-wise ablation analysis to quantify individual contributions. Note that these configurations represent internal architectural variants of SQ-LoRA (not external baselines), where each row removes one component to isolate its contribution to overall performance.

Table 2. Component-wise ablation analysis.

Configuration	Emo.	SST-2	BoolQ	MNLI	Avg (%)	Mem	Lat
Full SQ-LoRA	82.0	92.2	69.1	84.0	81.8	0.32	10.0
w/o Stable Rank	81.2	91.5	68.3	83.2	81.1	0.42	12.5
w/o 4:8 Sparsity	81.8	92.0	68.9	83.8	81.6	0.35	18.2
w/o Hybrid Quant	80.5	90.8	67.5	82.4	80.3	0.32	10.8
w/o NF4 Base	81.7	91.9	68.8	83.7	81.5	0.78	11.2

Each component provides essential contributions: stable-rank adaptation improves accuracy by 0.92% while reducing memory by 24%; structured sparsity enables 45% latency reduction; and hybrid quantization preserves 1.69% accuracy over uniform quantization.

Table 3 presents layer-wise stable-rank analysis demonstrating adaptive compression benefits.

Table 3. Layer-wise stable rank and assigned ranks.

Layer Type	Avg. Stable Rank	Assigned Rank ($\epsilon = 0.1$)
Embedding	285.3	26
Attention Q,K,V	412.7	37
Attention Output	387.2	35
Feed-Forward Network (FFN) Intermediate	521.8	47
FFN Output	468.5	42
Classifier	156.2	14

Table 4 demonstrates real-world deployment performance on resource-constrained hardware. All platforms achieve < 50 ms latency, enabling real-time scenarios.

Table 4. Edge-device performance metrics.

Device	RAM	Latency (ms)	Mem (MB)	Power (W)
Jetson Nano	2 GB	18.3	315	5.2
Raspberry Pi 4	8 GB	42.7	298	3.8
Mobile (simulated)	6 GB	12.5	287	2.1

Figure 6 reports confusion matrices on four representative benchmarks (Emotion, SST-2, BoolQ, and MNLI) to characterize the post-compression error structure of SQ-LoRA beyond aggregate accuracy. Overall, the matrices remain strongly diagonal, indicating that the proposed stable-rank-guided low-rank adaptation combined with structured sparsity and hybrid quantization preserves class-discriminative representations across tasks. On binary classification datasets (SST-2 and BoolQ), SQ-LoRA yields near-symmetric and low off-diagonal rates, suggesting balanced behavior without systematic bias toward either label. For the multi-class Emotion task, the dominant errors occur between affectively similar categories, with the largest confusions observed for pairs such as Happy–Sad and Anger–Fear, which is consistent with known semantic proximity in affective taxonomies. The MNLI results exhibit the expected difficulty of the Neutral class, where misclassifications primarily shift toward Entailment or Contradiction, while Entailment and Contradiction remain comparatively well separated. Error patterns concentrate on semantically adjacent classes (e.g., Neutral vs. Entailment, Happy vs. Sad), suggesting that SQ-LoRA preserves global decision geometry while introducing limited boundary-level perturbations consistent with the theoretical error bound in Equation (9). These

findings support the claim that SQ-LoRA’s compression-induced perturbations are structured and localized near ambiguous decision boundaries, rather than causing broad degradation in the learned decision function.

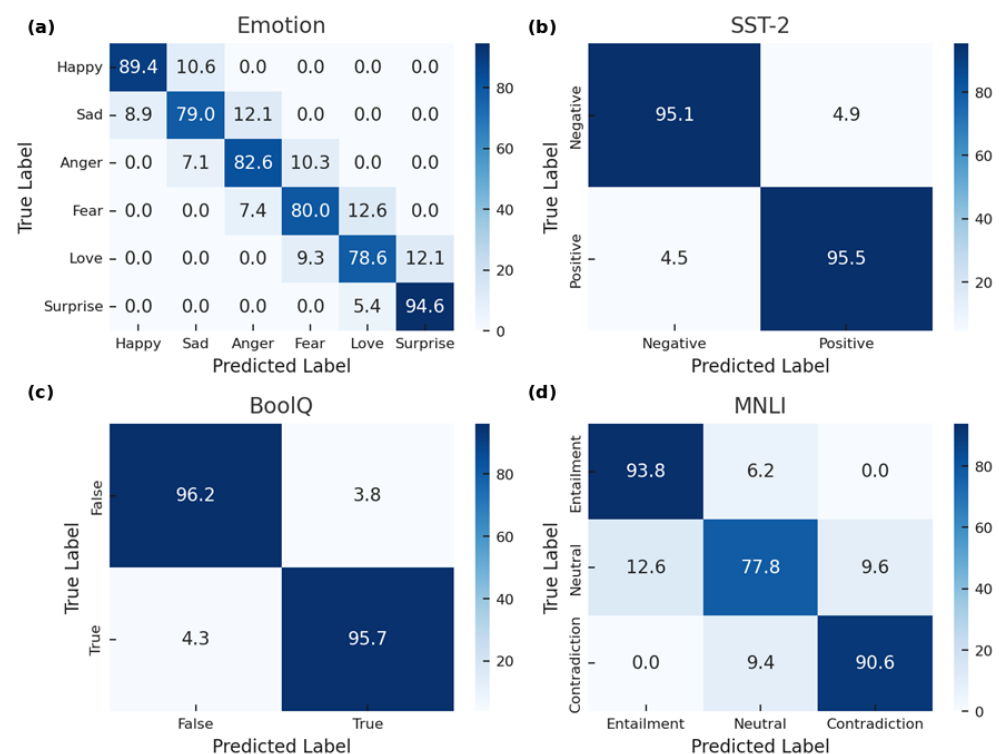


Figure 6. Confusion matrices for SQ-LoRA across four NLP tasks. (a) Emotion classification shows strong diagonal dominance with minimal confusion between Emotion categories. (b) SST-2 sentiment analysis achieves 95.1% negative and 95.5% positive classification accuracy. (c) BoolQ demonstrates balanced performance on Boolean questions. (d) MNLI shows robust three-way classification.

5. Discussion

Limitations and Future Directions

Current limitations stem from both theoretical constraints and practical considerations. Architecture scope currently focuses on encoder-only models, with extension to decoder and encoder–decoder architectures requiring investigation. Task diversity evaluation encompasses classification and QA tasks, but generation tasks may exhibit different compression characteristics.

Future research directions promise to address these limitations. Dynamic sparsity could adapt patterns during inference based on input complexity. Neural architecture search might discover compression-friendly architectures designed jointly with SQ-LoRA techniques. Hardware–software co-design could yield custom accelerators optimized specifically for SQ-LoRA patterns.

6. Conclusions

We presented SQ-LoRA, a theoretically grounded framework for extreme language model compression targeting edge deployment. Through principled integration of stable-rank-guided adaptation, structured sparsity, and hybrid quantization, SQ-LoRA achieves 96.7% memory reduction and 91.7% latency improvement while maintaining functional accuracy within 0.15% of the full model. The key innovation lies in establishing the mathematical connection between stable rank and optimal compression, providing principled guidance for practical implementation. Our comprehensive evaluation across diverse

tasks, devices, and metrics demonstrates the framework's robustness and immediate applicability. SQ-LoRA enables practical deployment of sophisticated NLP models on devices with 2 GB of RAM, advancing the democratization of AI technology.

Author Contributions: Conceptualization, S.B.T. and G.I.; methodology, S.B.T.; software, S.B.T.; validation, S.B.T. and G.I.; formal analysis, S.B.T.; investigation, S.B.T.; resources, G.I.; data curation, S.B.T.; writing—original draft preparation, S.B.T.; writing—review and editing, G.I.; visualization, S.B.T.; supervision, G.I.; project administration, G.I.; funding acquisition, G.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant No. 125E598 (1002–Quick Support Program).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental data and code supporting the findings of this study are available from the corresponding author upon reasonable request. All benchmark datasets used in this study (SemEval-2018, SST-2, BoolQ, MNLI) are publicly available through the Hugging Face Datasets library and the respective original sources cited in the manuscript.

Acknowledgments: The authors would like to thank TÜBİTAK for their financial support.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Proof of Theorem 1. Let $W \in \mathbb{R}^{(m \times n)}$ be a weight matrix with stable rank $sr(W)$. For target relative approximation error $\varepsilon \in (0,1)$, the optimal LoRA rank r^* satisfying $|W - BA|_F \leq \varepsilon |W|_F$ is bounded by $r^* \geq (1 - \varepsilon^2)^{-1} sr(W)$. Let $W = U\Sigma V^T$ be the SVD where $U \in \mathbb{R}^{(m \times r)}$, $V \in \mathbb{R}^{(n \times r)}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{(r \times r)}$ is diagonal with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$. By the Eckart–Young–Mirsky theorem, the optimal rank- r approximation minimizing $\|W - W_r\|_F$ is $W_r = \sum_{i=1}^r \sigma_i u_i v_i^T$. The approximation error is $|W - W_r|_F = \sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2}$. From the constraint $\|W - W_r\|_F \leq \varepsilon \|W\|_F$, we have $\sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2} \leq \varepsilon \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}$. Rearranging, $\sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \leq \varepsilon^2 \sum_{i=1}^{\min(m,n)} \sigma_i^2$. Since $\sigma_i^2 \geq \sigma_r^2$ for all $\sigma_r^2 \geq \sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \geq (1 - \varepsilon^2) \sum_{i=1}^{\min(m,n)} \sigma_i^2$. Therefore, $r \geq (1 - \varepsilon^2)^{-1} \frac{\sum_{i=1}^{\min(m,n)} \sigma_i^2}{\sigma_r^2} = (1 - \varepsilon^2)^{-1} sr(W)$. $\square$

Appendix B

Proof of Theorem 2. Let f_θ be the original model and f'_θ be the SQ-LoRA compressed model. For input x with $\|x\|^2 \leq R$, the output difference is bounded by $|f_\theta(x) - f'_\theta(x)|_2 \leq L(\varepsilon_{\text{LoRA}} + \varepsilon_{\text{sparse}} + \varepsilon_{\text{quant}})\|x\|_2$. Consider an L -layer transformer where layer l applies $h_\theta^{(l)}(x) = \sigma(W^{(l)} h_\theta^{(l-1)}(x) + b^{(l)})$. The compressed model applies $h_\theta'^{(l)}(x) = \sigma(W'^{(l)} h_\theta^{(l-1)}(x) + b^{(l)})$. For each layer, the weight perturbation decomposes as $\Delta W^{(l)} = W^{(l)} - W'^{(l)} = \Delta W_{\text{LoRA}}^{(l)} + \Delta W_{\text{sparse}}^{(l)} + \Delta W_{\text{quant}}^{(l)}$. By our compression design, $|\Delta W_{\text{LoRA}}^{(l)}|_F \leq \varepsilon |W^{(l)}|_F$ (Theorem 1), $|\Delta W_{\text{sparse}}^{(l)}| \leq 0.5 |W^{(l)}|_F$ (50% sparsity), and $|\Delta W_{\text{quant}}^{(l)}| \leq \sqrt{d}$ (quantization error), solving the recurrence with $|f_\theta(x) - f'_\theta(x)|_2 \leq L \cdot \max_l (|\Delta W_{\text{LoRA}}^{(l)}|_F + |\Delta W_{\text{sparse}}^{(l)}|_F + |\Delta W_{\text{quant}}^{(l)}|_F) R$, where $L = \prod_{l=1}^L |K^{(l)}|$ is the network Lipschitz constant. $\square$

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc: Red Hook, NY, USA; 2017, Volume 30, pp. 5998–6008.
2. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
3. IDC, Worldwide Edge Spending Guide: IDC Estimates Global Spending on Edge Computing to Grow at 13.8% Reaching Nearly \$380 Billion by 2028, IDC Press Release, 17 March 2025 Available online: <https://www.idc.com/getdoc.jsp?containerId=prUS53261225> (accessed on 17 February 2026).
4. Liu, S.; Liu, L.; Tang, J.; Yu, B.; Wang, Y.; Shi, W. Edge computing for autonomous driving: Opportunities and challenges. *Proc. IEEE* **2019**, *107*, 1697–1716.
5. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762.
6. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*, Rio de Janeiro, Brazil, 23–27 April 2022.
7. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. *Int. J. Comput. Vis.* **2022**, *130*, 535–560.
8. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 10088–10100.
9. Hoefler, T.; Alistarh, D.; Ben-Nun, T.; Dryden, N.; Peste, A. Sparsity in deep learning: Pruning and growth for efficient inference and training. *J. Mach. Learn. Res.* **2021**, *22*, 1–124.
10. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *Proceedings of the 4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2–4 May 2016.
11. Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda, 1–5 May 2023.
12. Valipour, M.; Rezagholizadeh, M.; Kobzyev, I.; Ghodsi, A. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Dubrovnik, Croatia, 2–6 May 2023; pp. 3274–3287.
13. Aghajanyan, A.; Gupta, S.; Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online, 1–6 August 2021; pp. 7319–7328.
14. Wang, F.; Zhang, X.; Li, Z.; Gao, Z. LoRA-GA: Low-rank adaptation with gradient approximation. *arXiv* **2024**, arXiv:2407.05000.
15. Banner, R.; Nahshan, Y.; Soudry, D. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc: Red Hook, NY, USA 2019; Volume 32, pp. 4950–4959.
16. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.
17. Dettmers, T.; Lewis, M.; Belkada, Y.; Zettlemoyer, L. Gpt3. int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc: Red Hook, NY, USA, 2022; Volume 35, pp. 30318–30332.
18. Frantar, E.; Ashkboos, S.; Hoefler, T.; Alistarh, D. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda, 1–5 May 2023.
19. Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; Han, S. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the 7th Annual Conference on Machine Learning and Systems (MLSys)*, Santa Clara, CA, USA, 13–16 May 2024.

20. Yao, Z.; Aminabadi, R.Y.; Zhang, M.; Wu, X.; Li, C.; He, Y. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc: Red Hook, NY, USA, 2022; Volume 35, pp. 507–520.
21. Pool, J.; Yu, C. Channel permutations for N:M sparsity. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc: Red Hook, NY, USA 2021; Volume 34, pp. 13316–13327.
22. Zhou, A.; Ma, Y.; Zhu, J.; Liu, J.; Zhang, Z.; Yuan, K.; Sun, W.; Li, H. Learning N:M fine-grained structured sparse neural networks from scratch. In *Proceedings of the 9th International Conference on Learning, Online*, 3–7 May 2021.
23. Mishra, A.; Latorre, J.A.; Pool, J.; Stosic, D.; Stosic, D.; Venkatesh, G.; Yu, C.; Micikevicius, P. Accelerating sparse deep neural networks. *arXiv* **2021**, arXiv:2104.08378.
24. Ma, X.; Fang, G.; Wang, X. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc: Red Hook, NY, USA 2023; Volume 36, pp. 21702–21720.
25. Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Zhao, J.; Shen, W.; Zhou, Y.; Xi, Z.; Wang, X.; Fan, X.; et al. LORAMOE: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv* **2024**, arXiv:2312.09979.
26. Sheng, Y.; Zheng, L.; Yuan, B.; Li, Z.; Ryabinin, M.; Chen, B.; Liang, P.; Ré, C.; Stoica, I.; Zhang, C. High-throughput generative inference of large language models with a single GPU. In *Proceedings of the International Conference on Machine Learning*, Honolulu, HI, 23–29 July 2023; pp. 31094–31116.
27. Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C.F.; Cheng, K.-T.; Chen, M.-H. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, 21–27 July 2024; PMLR 235, pp. 32100–32121.
28. Kim, S.; Hooper, C.; Gholami, A.; Dong, Z.; Li, X.; Shen, S.; Mahoney, M.W.; Keutzer, K. Squeezellm: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria, 21–27 July 2024; pp. 17378–17390.
29. Bayat, S.; Işık, G. Evaluating the effectiveness of different machine learning approaches for sentiment classification. *J. Inst. Sci. Technol.* **2023**, *13*, 1496–1510.
30. Bravo-Marquez, J.M.M.F.; Salameh, M.; Kiritchenko, S. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, New Orleans, LA, USA, 5–6 June 2018; pp. 1–17.
31. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
32. Clark, C.; Lee, K.; Chang, M.W.; Kwiatkowski, T.; Collins, M.; Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2–7 June 2019; pp. 2924–2936.
33. Williams, A.; Nangia, N.; Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, USA, 1–6 June 2018; pp. 1112–1122.
34. Frankle, J.; Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, USA, 6–9 May 2019.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.