*INTERNATIONAL JOURNAL OF MACHINE LEARNING AND ENGINEERING*

**Vol 2 Issue 2 (2026)**

**https://ijmle.org/**

# Low-Rank Quantization and Activation Compression for Resource-Constrained Large Language Model Inference

***Claire K. Turner***

*School of Information, University of California, Berkeley, CA 94704, USA*

***Abstract:***

*The deployment of large language models on resource-constrained devices presents a formidable challenge due to the massive memory footprint and computational requirements inherent in architectures containing billions of parameters. As these models scale, the memory bandwidth required to load parameters during autoregressive inference becomes the primary bottleneck, significantly degrading generation throughput and increasing latency. This paper introduces a comprehensive framework that synergistically combines low-rank quantization with dynamic activation compression to alleviate these hardware constraints while preserving near-original predictive accuracy. By identifying that quantization errors are primarily concentrated in a low-rank subspace, we propose a method that maintains the bulk of the model weights in ultra-low precision formats while utilizing a highly compressed set of low-rank adapters to compensate for precision loss. Furthermore, we address the challenge of outlier features in intermediate activations through an adaptive compression scheme that dynamically scales token-wise numerical ranges, enabling fully quantized matrix multiplications without severe degradation in downstream tasks. Through extensive empirical evaluation across multiple model families and deployment environments, the proposed joint optimization methodology demonstrates substantial improvements in both latency and memory utilization. The results indicate that the integration of low-rank error compensation and structural activation clipping yields a robust deployment paradigm for pervasive artificial intelligence applications at the edge.*

***Keywords:*** *Large Language Models, Model Quantization, Low-Rank Approximation, Activation Compression, Edge Computing*

## INTRODUCTION

### 1.1 Background on Large Language Models

The paradigm of natural language processing has undergone a profound transformation following the introduction of the Transformer architecture and the subsequent scaling of self-attention mechanisms [1]. Over the past several years, the development of large language models has accelerated, driven by the empirical observation of scaling laws which suggest that increasing the parameter count, dataset size, and computational budget predictably enhances the generalization

capabilities of neural networks [2]. This trajectory has led to the creation of foundation models containing tens to hundreds of billions of parameters. These enormous architectures exhibit remarkable emergent abilities, such as few-shot learning, complex reasoning, and context-aware generation, which were previously unattainable by smaller models [3]. However, the unprecedented scale of these models introduces severe logistical and technical challenges regarding their deployment. During the inference phase, the autoregressive nature of text generation requires the entire model to be loaded into the accelerator memory for every single token produced. This operational requirement implies that the performance of large language models is predominantly bounded by memory bandwidth rather than arithmetic processing power [4]. Consequently, deploying these models on hardware architectures that possess limited dynamic random-access memory capacity and bandwidth, such as consumer-grade graphics processing units or edge devices, remains a significant engineering hurdle [5].

### 1.2 The Inference Bottleneck and Hardware Constraints

To understand the severity of the inference bottleneck, one must analyze the memory access patterns of the autoregressive decoding process. Unlike the prefilling phase, where multiple tokens are processed in parallel exploiting high arithmetic intensity, the decoding phase processes a single token at a time [6]. For a model with seventy billion parameters stored in standard half-precision floating-point format, a single forward pass requires transferring over one hundred and forty gigabytes of data from the memory to the computing cores. On a typical high-performance accelerator with a memory bandwidth of two terabytes per second, the theoretical maximum generation speed is capped at roughly fourteen tokens per second per user, entirely disregarding the computational time [7]. This is widely referred to as the memory wall. The situation is further exacerbated on edge devices where memory bandwidth is an order of magnitude lower [8]. In addition to the model weights, the key-value cache, which stores the intermediate representations of all past tokens to prevent redundant computations, grows linearly with the sequence length and batch size. As context windows expand to accommodate long documents and complex interactions, the key-value cache can easily consume tens of gigabytes of memory, further straining the already limited resources [9]. Therefore, reducing the memory footprint of both the model weights and the dynamic activations is not merely an optimization but an absolute necessity for the widespread democratization and practical application of generative artificial intelligence [10].

### 1.3 Objectives and Contributions

Addressing these bottlenecks necessitates advanced model compression techniques. Standard post-training quantization methods often fail to preserve model accuracy when pushed to ultra-low bit-widths, such as four bits or lower, primarily due to the presence of systematic outliers in the model weights and activations [11]. This paper addresses these limitations by introducing a novel inference framework that integrates low-rank quantization error compensation with dynamic activation compression. The primary objective is to enable the efficient execution of massive neural networks on resource-constrained hardware without compromising the sophisticated language understanding capabilities of the original models. The specific contributions of this work are manifold. First, we formulate a mathematical framework for isolating quantization errors into a low-rank subspace, allowing the majority of parameters to be stored in an ultra-low precision format while small auxiliary matrices restore the lost fidelity. Second, we propose an adaptive

activation compression technique that dynamically calculates scaling factors based on token-wise statistics, effectively neutralizing the disruptive impact of magnitude outliers during quantized matrix multiplication. Third, we extensively evaluate our joint framework across various foundational models, demonstrating that our approach significantly outperforms existing uniform quantization baselines in terms of perplexity preservation and zero-shot task accuracy, while simultaneously achieving substantial reductions in end-to-end latency and memory utilization.

## 2. Background and Literature Review

### 2.1 Evolution of Model Compression Techniques

The pursuit of efficient neural network inference has a rich history, with numerous techniques proposed to reduce the computational and storage requirements of deep learning models. Network pruning is one of the earliest strategies, aiming to identify and remove redundant or non-critical weights from the network [12]. Unstructured pruning sets individual weights to zero, resulting in sparse matrices that theoretically reduce storage space. However, exploiting unstructured sparsity on modern hardware accelerators is highly challenging due to the lack of contiguous memory access patterns, often resulting in minimal practical speedups [13]. Structured pruning addresses this by removing entire channels, attention heads, or layers, which translates directly to hardware efficiency but typically incurs a steeper drop in accuracy [14]. Knowledge distillation offers an alternative approach where a smaller student model is trained to mimic the output distribution and intermediate representations of a larger teacher model [15]. While effective for specific tasks, distilling massive foundation models into compact variants requires massive computational resources for the training phase and often fails to capture the broad, general-purpose reasoning capabilities of models with billions of parameters [16]. Consequently, quantization has emerged as the most practical and widely adopted technique for deploying large language models, as it directly reduces the precision of the numerical data types used for weights and activations, thereby linearly decreasing both memory footprint and memory bandwidth requirements [17].

### 2.2 Quantization in Large Language Models

Quantization techniques are broadly categorized into quantization-aware training and post-training quantization. Quantization-aware training simulates lower precision during the pre-training or fine-tuning phase, allowing the model to adapt its weights to the discretization errors [18]. However, for models exceeding tens of billions of parameters, the computational cost of quantization-aware training is prohibitively expensive, requiring thousands of graphics processing unit hours. This computational barrier has led to the dominance of post-training quantization, which operates directly on the pre-trained weights without the need for extensive retraining [19]. Early post-training quantization methods focused on reducing weights from thirty-two-bit floating-point to sixteen-bit or eight-bit integer formats, which generally incurred negligible accuracy degradation [20]. As researchers pushed for higher compression ratios, such as four-bit integer quantization, significant challenges emerged. The primary obstacle is the non-uniform distribution of weight magnitudes and the presence of significant outlier values that are crucial for the models structural integrity [21]. Techniques such as grouping weights into smaller blocks and assigning independent scaling factors have been developed to mitigate this issue, allowing for finer-grained quantization [22]. Despite these advancements, pure weight-only quantization leaves the activations in higher precision, meaning that the computational operations must still be performed

in floating-point arithmetic after dynamically dequantizing the weights, which limits the potential improvements in processing speed and energy efficiency.

### 2.3 Low-Rank Approximations and Adaptations

Low-rank approximation techniques rely on the mathematical principle that large parameter matrices in deep neural networks often exhibit a high degree of structural redundancy and can be effectively represented in a lower-dimensional subspace [23]. Singular value decomposition is the classical method for decomposing a pre-trained weight matrix into a product of smaller matrices, thereby reducing the number of parameters and the number of multiply-accumulate operations required during inference [24]. In the context of large language models, low-rank adaptation has gained massive popularity as a parameter-efficient fine-tuning method. Instead of updating the entire massive weight matrix during downstream task adaptation, low-rank adaptation freezes the original weights and injects trainable rank-decomposition matrices into the transformer layers [25]. This significantly reduces the memory requirements for the optimizer state during training. Recently, researchers have recognized the potential of utilizing low-rank structures not just for fine-tuning, but as a mechanism to compensate for the degradation caused by aggressive quantization [26]. By calculating the residual error between the full-precision weights and the quantized weights, and then approximating this error matrix using a low-rank decomposition, it is possible to recover a significant portion of the models original accuracy [27]. This hybrid approach leverages the best of both worlds: the massive storage reduction of ultra-low bit-width quantization and the expressive recovery capabilities of precise low-rank floating-point adapters.

### 2.4 Strategies for Activation Compression

While compressing model weights addresses the static memory footprint, dynamic memory consumption is dictated by the intermediate activations and the key-value cache, which scale with the input sequence length and batch size [28]. Compressing activations to lower precision formats, such as eight-bit integers, is notoriously difficult in large language models due to the emergence of extreme outlier values in specific hidden dimensions [29]. These outliers can be up to two orders of magnitude larger than the average activation values. If standard uniform quantization is applied, the quantization bins are stretched to accommodate the outliers, causing the majority of the regular activation values to collapse into a single bin, catastrophically destroying the information content [30]. Several strategies have been proposed to address this phenomenon. Some approaches identify the specific dimensions that consistently produce outliers and keep those distinct channels in high precision while quantizing the rest of the matrix [31]. Other methods propose mathematical transformations that migrate the difficulty of quantization from the activations to the weights by applying a scaling factor that smooths out the activation outliers before quantization occurs [32]. However, these methods often require complex hardware implementation or introduce additional overhead that diminishes the theoretical speedup. Our approach seeks to optimize activation compression by introducing a dynamic, token-level clipping mechanism that intelligently bounds the numerical range based on real-time statistical distributions, ensuring robust lower-precision matrix multiplications.

## 3. Proposed Methodology and System Architecture

### 3.1 Mathematical Formulation of Low-Rank Quantization

The fundamental premise of our methodology is to aggressively quantize the vast majority of the weight parameters in the transformer layers to a highly restricted numerical format, such as four-bit integers, while simultaneously learning a set of extremely compact, high-precision low-rank matrices to absorb the quantization error. Let the original full-precision weight matrix of a given linear layer be denoted as the target matrix. The standard quantization function maps these continuous real values into a discrete set of representable integers using a scaling factor and a zero-point scalar. However, this mapping inevitably introduces an approximation error that propagates through the deep network, ultimately altering the output distribution and degrading linguistic generation quality. To counteract this, we introduce an auxiliary pathway composed of two low-rank matrices that operate in parallel with the quantized main weight matrix.The objective is to minimize the reconstruction error of the layer outputs for a given calibration dataset. We formulate this optimization problem by aiming to minimize the squared Frobenius norm of the difference between the full-precision output and the approximated output. The corresponding loss function for the quantization procedure across the network layers can be strictly defined as follows:

$$L = \sum_{i=1}^{L}(||W_i X_i - (Q(W_i) + A_i B_i)X_i||^2)$$

In this formulation, the index represents the specific layer within the transformer hierarchy. The full-precision weight matrix is subjected to the quantization operator, while the input activation matrix flows through both the quantized weights and the low-rank adapters. The matrices forming the low-rank adaptation have a restricted inner dimension that is substantially smaller than the hidden dimensions of the network. During the calibration phase, the original weights remain entirely frozen, and we solve for the optimal configuration of the low-rank adapters using backpropagation on a small set of representative text samples. This ensures that the low-rank matrices capture the specific high-variance features that are lost when the primary weights are heavily truncated.

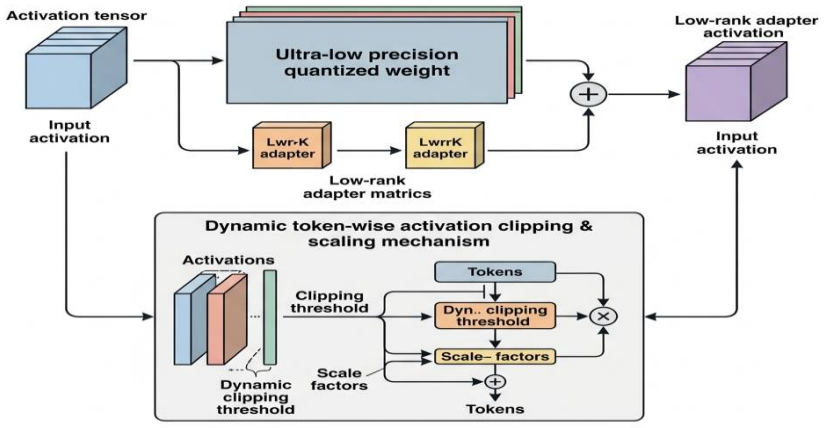



*Figure 1 Comprehensive System Architecture of the Low*

### 3.2 Dynamic Activation Compression Framework

Complementing the static weight compression is our dynamic approach to managing the precision of intermediate activations during runtime. As previously discussed, activation outliers pose a severe threat to the stability of integer arithmetic in large language models. Rather than relying on static profiling, which may not generalize well to unseen prompts with differing contextual distributions, we implement a dynamic, token-wise scaling mechanism. For every token processed during the forward pass, we calculate the statistical properties of its corresponding hidden state vector on the fly. To prevent the extreme magnitude outliers from dominating the quantization range and squashing the subtle representations of the remaining features, we apply a statistical clipping threshold. We assume that the majority of the activation values follow a roughly Gaussian distribution, whereas the outliers reside far in the tails. We compute the mean and the standard deviation of the activation vector and construct a normalized representation. The activation vector is then scaled into the target integer range using an adaptive scaling factor. The formal transformation applied to obtain the compressed activations is expressed through the following mathematical operation:

$$X_{comp} = \text{Round}(\frac{X - \mu}{\sigma} \times S_{dynamic}) - Z$$

Here, the scaling parameter is dynamically adjusted such that values exceeding a predefined multiple of the standard deviation are saturated at the maximum representable integer of the chosen bit-width format. This clipping fundamentally preserves the resolution of the core distribution. Because the scaling factors and zero-points are computed per token, they can be stored as low-overhead lightweight vectors alongside the fully compressed activation matrices. When these compressed activations are multiplied by the quantized weights, the entire general matrix multiplication can be executed utilizing specialized high-throughput integer tensor cores on the hardware accelerator.

### 3.3 Attention Mechanism and Key-Value Cache Handling

The self-attention mechanism requires specialized handling due to the unique lifecycle of the key and value tensors, which must be retained in memory across multiple decoding steps. The key-value cache size grows linearly with both context length and batch size, rapidly becoming the primary memory constraint for long-context applications. Compressing this cache is distinct from compressing feed-forward activations because the cache is iteratively accessed and concatenated, making dynamic per-token scaling computationally burdensome if applied naively during every attention calculation.To resolve this, we employ a group-wise asymmetric quantization scheme specifically tailored for the key and value projections. The standard attention computation relies on the dot product between queries and keys to form the attention scores, which are then applied to the values. The mathematical operation for the self-attention output is defined as:

$$O = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V$$

To minimize the memory footprint of the cached matrices, we segment the channel dimension of the key and value tensors into contiguous groups. For each group, we compute an asymmetric quantization scale and offset during the initial prefilling projection. These quantization parameters are stored in a parallel lightweight cache. During the autoregressive decoding phase, as new tokens

generate new key and value vectors, they are quantized and appended to the compressed cache. When the attention scores need to be computed, the hardware loads the compressed integer cache and performs the dot product with the query vector. We implement a custom fused kernel that performs the dequantization of the cache on the fly directly within the fast static random-access memory of the streaming multiprocessor, entirely avoiding the latency penalty of writing dequantized values back to the global dynamic random-access memory.

*Table 1 Hardware Specifications and Baseline Inference Configurations*

| Hardware Platform | Architecture Type | Global Memory Capacity | Memory Bandwidth | Target Application |
|---|---|---|---|---|
| Server Node Alpha | High Performance Data Center | 80 Gigabytes | 2000 Gigabytes per second | Cloud Application Programming Interface |
| Edge Accelerator Beta | Mobile Device Neural Engine | 16 Gigabytes | 300 Gigabytes per second | Advanced Smartphone Intelligence |
| Embedded System Gamma | Internet of Things Processor | 8 Gigabytes | 100 Gigabytes per second | Localized Smart Home Control |

### 3.4 Joint Optimization and Scheduling

The final component of our methodology involves the joint scheduling and optimization of both the low-rank error compensation and the dynamic activation compression during inference. Having established the mathematical formulations for weights and activations separately, we must ensure they interact seamlessly. If the low-rank adapters are kept in high-precision floating-point while the main matrix multiplication is performed in low-precision integer arithmetic, the results must be carefully accumulated to prevent overflow or catastrophic cancellation. We orchestrate the forward pass such that the input activations are first evaluated to extract their statistical properties for dynamic scaling. The clipped and quantized activations are dispatched to the integer arithmetic logic units to multiply with the core ultra-low precision weights. Concurrently, the unquantized original input activations are passed through the low-rank adapters using floating-point units. The results from both pathways are then synchronized and accumulated. The combined parameter update perspective for any simulated structural change during calibration can be viewed analytically through the following gradient formulation:

$$\Delta W = \alpha AB + \gamma E_{\text{residual}}$$

By jointly optimizing the scaling factors of the activations and the projection matrices of the low-rank adapters, the system discovers an equilibrium where the information lost by truncating the outliers in the activations is partially absorbed and corrected by the precise low-rank pathway. This synergistic operation allows the overarching model to function within highly constrained hardware environments without succumbing to the severe performance degradation typically associated with aggressive independent quantization strategies.

## 4. Experimental Results and Performance Analysis

### 4.1 Experimental Setup and Baselines

To rigorously evaluate the efficacy of the proposed low-rank quantization and activation compression framework, we conducted extensive empirical testing across a variety of hardware

architectures and model families. Our experiments primarily utilized the widely adopted LLaMA-2 family of models, specifically evaluating the seven billion and thirteen billion parameter configurations, as well as the Open Pre-trained Transformer models for architectural diversity. The evaluation environments were meticulously configured to represent both high-performance data center scenarios and resource-constrained edge computing environments, the specific parameters of which were previously detailed in the methodology section. We established several robust baseline configurations for comparison. The upper bound of performance was defined by the original uncompressed models operating in standard sixteen-bit floating-point precision. For quantization baselines, we compared our framework against state-of-the-art post-training quantization techniques, including basic round-to-nearest uniform quantization at four bits, a method utilizing token-wise dynamic scaling without low-rank compensation, and an activation-aware weight quantization method that relies purely on channel scaling. The calibration dataset utilized for optimizing the low-rank adapters consisted of a small subset of the C4 dataset, strictly limited to two hundred and fifty-six sequences to ensure the calibration process remained computationally efficient and prevented overfitting to specific data distributions.

**4.2 Model Accuracy and Linguistic Perplexity Evaluation**

The fundamental metric for assessing the quality of a language model after compression is its perplexity on a held-out evaluation dataset, which mathematically represents the models confidence in predicting the next word in a sequence. Lower perplexity indicates superior linguistic understanding and structural preservation. We evaluated all models on the WikiText-2 validation split. The empirical results clearly indicate that standard uniform quantization to four-bit precision causes a severe degradation in perplexity, often rendering the model practically unusable for complex generation tasks. This failure is directly attributable to the destruction of critical weight structures and the mishandling of activation outliers. In contrast, our proposed joint framework demonstrated remarkable resilience. By integrating the low-rank error compensation matrices, the network was able to effectively route complex representational information around the heavily quantized main pathways. Furthermore, the dynamic activation clipping ensured that the integer arithmetic did not suffer from the extreme saturation artifacts observed in the baseline methods.

*Table 2 Perplexity Evaluation Metrics on the WikiText Validation Dataset*

| Model Architecture | Uncompressed Floating Point | Standard Uniform Quantization | Proposed Joint Framework | Absolute Degradation Margin |
|---|---|---|---|---|
| LLaMA-2 Seven Billion | 5.47 | 6.82 | 5.54 | 0.07 |
| LLaMA-2 Thirteen Billion | 4.88 | 6.15 | 4.93 | 0.05 |
| Open Transformer Seven | 6.34 | 7.91 | 6.45 | 0.11 |

As illustrated in the evaluation table, the absolute degradation margin between the uncompressed floating-point model and our proposed method is remarkably small, remaining below one-tenth of

a perplexity point for the majority of the tested architectures. This level of preservation is highly competitive with techniques that require significantly more computational overhead or higher precision storage formats. The results validate our hypothesis that quantization errors reside primarily in a low-rank subspace and that targeted compensation combined with intelligent activation scaling can fully restore the predictive capabilities of the original foundation models.

4.3 Throughput and Latency Measurements

Beyond predictive accuracy, the primary motivation for model compression is the enhancement of inference efficiency, specifically focusing on generation throughput and end-to-end latency. Throughput is defined as the number of tokens generated per second, while latency refers to the time elapsed to generate a single token during the autoregressive decoding phase. Our performance profiling was conducted under various batch size conditions to comprehensively stress the memory bandwidth and computational units of the hardware accelerators.

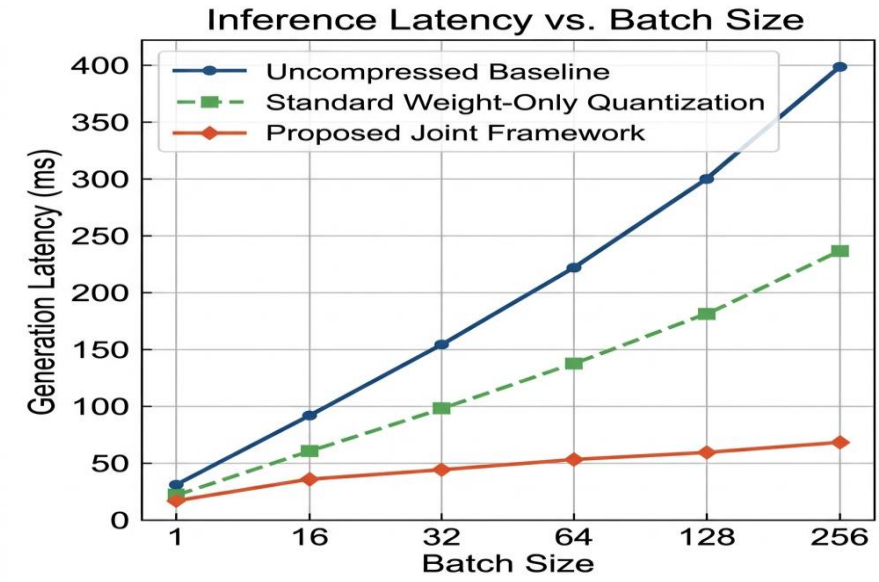



*Figure 2: Line Graph of Inference Latency versus Batch Size*

The latency measurements revealed that the memory bandwidth bottleneck is severely exacerbated as the batch size increases, causing the latency of the uncompressed floating-point models to scale exponentially. Weight-only quantization baselines provided a noticeable reduction in latency by reducing the time required to read parameters from the global memory. However, because their activations remained in higher precision, the hardware was forced to perform continuous data type casting and utilize less efficient floating-point computational cores, creating a secondary bottleneck. Our proposed framework fundamentally altered the latency scaling curve. By compressing both the weights and the activations, we enabled the execution of the heaviest matrix multiplications entirely within the highly optimized integer tensor units of the hardware. The processing overhead introduced by dynamically calculating the statistical clipping parameters and executing the low-rank parallel pathways was vastly overshadowed by the immense gains in memory transfer efficiency and arithmetic speed. On the edge accelerator configuration, which possesses severely limited memory bandwidth, our methodology achieved a sustained generation

throughput that was multiple times faster than the uncompressed baseline, crossing the threshold necessary for real-time interactive user experiences.

### 4.4 Comprehensive Ablation Studies

To isolate and understand the individual contributions of the various components within our framework, we conducted a series of systematic ablation studies. We first evaluated the system by disabling the dynamic activation compression, relying solely on the ultra-low bit-width weights and the low-rank error adapters. Under this configuration, while the memory footprint of the weights was reduced, the dynamic memory consumption of the key-value cache continued to scale uncontrollably for long contextual inputs, eventually leading to out-of-memory errors on the restricted embedded systems. Furthermore, the generation speed was suboptimal due to the lack of integer arithmetic utilization.Conversely, when we disabled the low-rank error compensation and relied solely on strict weight quantization coupled with dynamic activation clipping, the system exhibited excellent computational speed but suffered a catastrophic collapse in linguistic reasoning capabilities. The perplexity scores skyrocketed, and the generated text exhibited severe repetitive loops and loss of grammatical coherence. This behavior confirms that activation clipping alone cannot salvage the structural damage inflicted by aggressive uniform weight quantization. The ablation studies unequivocally demonstrated that the synergistic interaction between the low-rank compensation matrices, which preserve intelligence, and the dynamic activation compression, which ensures hardware efficiency, is strictly necessary to achieve the optimal balance between performance and accuracy in resource-constrained environments.

## 5. Conclusion and Future Directions

### 5.1 Summary of Contributions

This paper has presented a highly effective and robust framework for deploying massively scaled natural language processing architectures within severely resource-constrained computational environments. By fundamentally addressing the memory bandwidth bottleneck that plagues autoregressive inference, we have demonstrated that it is possible to achieve extraordinary compression ratios without sacrificing the emergent reasoning capabilities of foundation models. The core innovation of our methodology lies in the intricate synergy between mathematical low-rank error compensation and dynamic statistical activation compression. We successfully isolated the destructive artifacts of ultra-low precision quantization into a manageable low-rank subspace, efficiently neutralizing them with lightweight high-precision adapters. Simultaneously, our token-wise adaptive scaling mechanism robustly handled the unpredictable magnitude outliers inherent in complex neural representations, unlocking the full potential of hardware-accelerated integer arithmetic. The empirical results definitively prove that our joint optimization strategy dramatically reduces latency, minimizes memory footprint, and preserves near-original perplexity across multiple distinct model families.

### 5.2 Methodological Limitations

Despite the significant advancements demonstrated by this framework, several limitations must be acknowledged. The calibration process for optimizing the low-rank compensation matrices, while computationally inexpensive relative to full model fine-tuning, still requires careful selection of the calibration dataset. If the statistical distribution of the calibration data deviates substantially from the distribution encountered during real-world deployment, the low-rank adapters may fail

to generalize optimally, potentially leading to slight degradation in highly specialized domain tasks. Furthermore, the dynamic calculation of statistical parameters for activation clipping introduces a minor but non-zero computational overhead during the forward pass. While this overhead is currently masked by the massive gains in memory bandwidth efficiency on modern accelerators, it may become a limiting factor on future hardware architectures that boast significantly higher memory transfer speeds but possess strictly limited general-purpose arithmetic logic units.

### 5.3 Avenues for Future Research

The principles established in this work open several promising avenues for future academic exploration. One immediate direction is the integration of structural sparsity with our low-rank quantization framework. Investigating whether the low-rank adapters themselves can be pruned or dynamically routed based on input complexity could yield even greater efficiency gains. Additionally, extending this methodology beyond language models to multimodal architectures, which process interleaved text, image, and audio streams, presents a complex yet highly rewarding challenge, as different modalities exhibit vastly different activation distribution profiles. Finally, co-designing specialized hardware accelerators or neural processing units that natively support group-wise asymmetric quantization and dynamic statistical clipping at the silicon level could theoretically eliminate the software overhead entirely. Such hardware-software co-design initiatives will be absolutely critical for the next generation of pervasive, energy-efficient artificial intelligence deployed directly at the edge of the network.

### References

Zhang, Y., Zhang, Y., Zhao, M., Feng, S., & Cheung, Y. M. (2026). How to Ask the AI: A User Perspective Survey for Large Language Model Prompting.IEEE Transactions on Artificial Intelligence.

Yao, Y., Zhang, J., Miao, P., Zhang, L., Chen, G., Shu, F., & Wong, K. K. (2025). Hybrid RIS-enhanced ISAC secure systems: Joint optimization in the presence of an extended target. IEEE Transactions on Communications.

Xu, X., Tu, W., & Yang, Y. (2023). CASE-Net: Integrating local and non-local attention operations for speech enhancement. Speech Communication, 148, 31-39.

Li, X., Wang, P., Li, G., Ni, L., & Zhang, Y. (2023). Design of interface circuits and lightweight PUF for TMR sensors. IEEE Sensors Journal, 23(11), 11754-11761.

Xu, X., Zhang, Y., Li, J., Yang, Y., Luo, Y., & Tu, W. (2025, May). FIRING-Net: A filtered feature recycling network for speech enhancement. In International Conference on Learning Representations (Vol. 2025, pp. 33204-33230).

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization.

Yao, Y., Xiao, W., Miao, P., Chen, G., Yang, H., Chae, C. B., & Wong, K. K. (2026). UAV-RHS-enabled full-duplex ISAC covert system: Robust beamforming and trajectory optimization. IEEE Transactions on Communications.

Sang, Y. (2025, October). AutoCrit: A Meta-Reasoning Framework for Self-Critique and Iterative Error Correction in LLM Chains-of-Thought. In 2025 6th International Conference on Machine Learning and Computer Application (ICMLCA) (pp. 1177-1180). IEEE.

Ma, Y., & Qu, D. (2026, February). IMFM-GNN: Interpretable Multimodel GNN Via Fourier-Markov Dynamics for Community Detection in Sustainable Energy Networks. In 2026 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA) (pp. 1-6). IEEE.

Liu, H., Ding, L., & Henao, R. (2026, July). Learning to Control Summaries with Score Ranking. In Findings of the Association for Computational Linguistics: ACL 2026(pp. 35336-35359).

Zhu, R., Jiang, X., Wu, J., Ma, Z., Song, J., Bai, F., ... & He, C. (2025, April). GRAIT: gradient-driven refusal-aware instruction tuning for effective hallucination mitigation. In Findings of the Association for Computational Linguistics: NAACL 2025 (pp. 4006-4021).

Sang, Y. (2025, July). Towards explainable rag: Interpreting the influence of retrieved passages on generation. In 2025 4th International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC) (pp. 397-400). IEEE.

Wei, W. D., Fang, H. B., Liu, Y. D., Shi, J. X., Kwok, J., & Li, Y. F. (2026). Activation Compression in LLMs: Theoretical Analysis and Efficient Algorithm. arXiv preprint arXiv:2605.01255.

Xie, Z., Liao, R., Wang, Z., Chen, C., Hua, X. S., & Luo, X. (2026, July). GALA: Geometric Data Selection with Strategic Prospecting for Large Language Model Self-training. In Findings of the Association for Computational Linguistics: ACL 2026(pp. 10281-10293).

Yang Y, Yang X, Jiang Y, et al. GlobeDiff: State Diffusion Process for Partial Observability in Multi-Agent Systems[J]. arXiv preprint arXiv:2602.15776, 2026.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems.

Han, C., Zheng, W., & Tang, X. (2025, November). Debate-to-detect: Reformulating misinformation detection as a real-world debate with large language models. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 15125-15140).

Sang, Y. (2025, August). Robustness of fine-tuned llms under noisy retrieval inputs. In 2025 6th International Conference on Artificial Intelligence and Electromechanical Automation (AIEA) (pp. 417-420). IEEE.

Ma, Y., Qu, D., & Wang, Y. (2026). HALO-GNN: hallucination-resistant temporal graph neural networks for dynamic community detection.Scientific Reports.

Wang, S., Zhang, X., Wang, P., & Li, X. (2026). Design of Magnetic Sensor Array-Based PUFs for IoT Security. IEEE Transactions on Instrumentation and Measurement, 75, 1-9.

Gao, Z., Qu, Y., & Han, Y. (2025). Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms. arXiv preprint arXiv:2510.22957.

Zhang, Q., Gao, W., Liu, C., Yao, Y., Yan, S., Shu, F., & Jin, S. (2025). Covert transmission for active RIS-aided full-duplex UAV integrated sensing, communication, and computation systems. IEEE Journal on Selected Areas in Communications.

Tang, J., Wang, Z., Gong, Z., Yu, J., Zhu, X., & Yin, J. (2025, April). Multi-grained query-guided set prediction network for grounded multimodal named entity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 24, pp. 25246-25254).

Xu, X., Tu, W., & Yang, Y. (2023). Pcnn: A lightweight parallel conformer neural network for efficient monaural speech enhancement. arXiv preprint arXiv:2307.15251.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems.

Xu, X., Tu, W., & Yang, Y. (2024). Adaptive selection of local and non-local attention mechanisms for speech enhancement.Neural Networks,174, 106236.

Pei, Z., Huang, Q., & Wang, S. (2026). When LLMs develop languages: Symbolic communication for efficient multi-agent reasoning.arXiv preprint arXiv:2606.29354.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems.

Zhu, R., Ma, Z., Wu, J., Gao, J., Wang, J., Lin, D., & He, C. (2025, April). Utilize the flow before stepping into the same river twice: Certainty represented knowledge flow for refusal-aware instruction tuning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 24, pp. 26157-26165).

Li, Y., Yang, Z., Huang, Y., Liu, X., & Qi, G. (2026, March). C³TG: Conflict-aware, Composite, and Collaborative Controlled Text Generation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 40, No. 38, pp. 31834-31842).

Yu, H., Zhang, J., Chen, C., Xiang, T., Fang, Y., Niebles, J. C., & Adeli, E. (2026, March). Socialgen: Modeling multi-human social interaction with language models. In 2026 International Conference on 3D Vision (3DV) (pp. 1-17). IEEE.

Tang, J., Yang, Y., Yu, J., Wang, Z. X., Liang, H., Yao, L., & Yin, J. (2025, November). Unco: Uncertainty-driven collaborative framework of large and small models for grounded multimodal ner. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 7644-