Quantization Methods for Energy Efficient LLM Deployments

Tomislav Šubić[1,2]
[1]University of Trieste, Italy
[2]Arctur d.o.o., Nova Gorica, Slovenia

tomislav.subic@arctur.si

Abstract: The deployment of large language models (LLMs) in production environments faces significant challenges due to computational and energy requirements during inference. This paper presents a comprehensive empirical analysis of quantization methods applied to the Qwen3 model family, ranging from 0.6B to 32B parameters. We evaluate six quantization approaches: GPTQ 4-bit, GPTQ 8-bit, AWQ, FP8 W8A8 and INT8 W8A8, and the original FP16 baseline across six established benchmarks (MMLU, HumanEval, TruthfulQA, MetaBench, GSM8K, ARC Challenge). Our analysis examines the relationship between model size, quantization method, accuracy preservation, energy consumption, and inference performance across various context lengths. We demonstrate that larger Qwen3 models exhibit increased resilience to quantization-induced accuracy degradation, while aggressive quantization methods provide substantial energy savings with acceptable trade-offs in model performance. These findings provide crucial insights for optimizing LLM deployments in resource-constrained environments.

Keywords: Large language models, Quantization, Energy efficiency, Model compression, Inference optimization

1. Introduction

The rapid advancement of large language models has transformed natural language processing capabilities, with models achieving remarkable performance across diverse tasks. However, the deployment of these models in production environments presents significant challenges, particularly regarding computational requirements and energy consumption during inference. The Qwen3 model family, ranging from 0.6B to 32B parameters, represents a comprehensive suite of models that enables systematic analysis of quantization effects across different scales.

Quantization has emerged as a critical technique for model compression, enabling the deployment of large models on resource-constrained hardware while maintaining acceptable performance levels. However, the relationship between model size, quantization method, and resulting performance characteristics remains poorly understood. This work addresses this gap through a systematic empirical analysis of quantization methods applied to the Qwen3 model family.

Our primary motivation is reducing energy consumption in inference deployments while maintaining model quality. As LLMs become increasingly prevalent in production applications, understanding the trade-offs between model compression, accuracy preservation, and energy efficiency becomes essential for sustainable AI deployment.

2. Related Work

2.1 Large Language Models Quantization

Quantization is a model compression technique where weights and activations are represented using reduced numerical precision, typically from 32 or 16-bit floating-point to lower bit-widths like 8-bit integer. This technique significantly reduces memory requirements and computational complexity while potentially maintaining model performance. Contemporary quantization methods have shifted towards post-training approaches, that avoid the complexity of quantization-aware training and are more accessible to individuals and smaller organizations. Some methods use calibration datasets to optimize scale factors, while simple quantization techniques typically round weights and/or activations to the nearest number (Dettmers et al., 2023; “Zhu et al. - 2024 - A Survey on Model Compression for Large Language Models.pdf,” no date) (Zhu et al., 2024) .

GPTQ (Generative Post-Training Quantization) employs approximate second-order information to achieve accurate quantization of large models. The method can quantize models with 175 billion parameters in approximately four GPU hours, reducing bit-width per weight with minimal accuracy degradation. Recent improvements to GPTQ include enhanced bounds and theoretical guarantees (Frantar et al., 2023).

AWQ (Activation-aware Weight Quantization) identifies and protects salient weights based on activation distribution patterns. By preserving only 1% of the most important weights at higher precision, AWQ achieves significant compression while maintaining performance. The method's effectiveness stems from the observation

that activation distributions are highly non-uniform, with a small fraction of weights contributing disproportionately to model outputs. The method demonstrates particular effectiveness for edge device deployment(Lin et al., 2024).

SmoothQuant is a post-training quantization technique that enables accurate 8-bit weight and 8-bit activation inference by smoothing activation outliers that typically degrade quantization performance(Xiao et al., 2024). The method applies a mathematically equivalent transformation using scaling factors to redistribute quantization difficulty from activations to weights. The key innovation lies in its ability to smooth the distribution of both weights and activations without requiring retraining (Xiao et al., 2024).

Round-to-Nearest (RTN) quantization represents the simplest post-training approach, using deterministic rounding without optimization. While less sophisticated than GPTQ or AWQ, RTN offers computational simplicity and hardware compatibility, making it attractive for resource-constrained deployments where implementation complexity is a concern (Nagel et al., 2021; Kuzmin et al., 2023). The following two techniques use RTN quantization utilising different numerical precision formats.

INT8 W8A8 represents a straightforward quantization approach that converts both weights and activations to 8-bit integer representation using simple rounding. While less sophisticated than other methods, RTN offers computational simplicity and hardware compatibility.

FP8 W8A8 quantization utilizes 8-bit floating-point representation, offering a balance between compression and accuracy preservation. This approach has gained adoption in modern hardware accelerators designed for machine learning workloads (van Baalen et al., 2023).

Memory bandwidth is often the biggest source of latency in single-prompt inference, as for the inference process to complete, we need to pass the whole model from VRAM to the GPU processors, and for each generated token (Gholami et al., 2024). On the NVIDIA L40S with its 864 GB/s memory bandwidth and 48 GB of GDDR6 VRRAM quantization dramatically reduces this transfer overhead. At FP16 precision a 14 billion–parameter model like Qwen3-14B occupies about 29.5 GB and requires roughly 34 ms per token just to stream its weights. Converting the model to INT8 cuts its size to around 15.7 GB, halving the transfer time to approximately 18 ms per token. Pushing further to a highly optimized 4-bit format reduces the footprint to about 9 GB and brings per-token transfer times down to just over 10 ms. Because the L40S can accommodate even full-precision models entirely in its large VRAM pool, quantization yields a clear net performance gain

It is important to understand that 4-bit quantized models perform inference dequantizing and then computing. The weights are stored in compressed 4-bit format but are dequantized to FP16/BF16 for actual mathematical operations. GPUs don't have native capabilities for INT4 computations, but they do for INT8 and recently also for FP8. This approach achieves significant memory savings and bandwidth improvements while maintaining computational accuracy, though it does not eliminate the need for higher-precision arithmetic during inference, and often comes with an computational and energy overhead. However, we find that the the decrease in memory-transfer latency outweighs any additional compute from dequantization.

(Dettmers and Zettlemoyer, 2023) show that larger models are more resilient to quantization accuracy degradation and some even suggest there are scaling laws to confirm it. However, they focused their evaluations on Facebook OPT model family, which is now outdated and is considered to be undertrained. OPT-175B was trained on 300 billion tokens, yet the compute-optimal training for a 175 billion-parameter model would require roughly 1.4 trillion tokens according to the Chinchilla scaling laws (Hoffmann et al., 2022), indicating that OPT models were undertrained. If a model is undertrained, a lot of weights are irrelevant, which has an effect on quantization accuracy and resillience.

2.2 Energy Efficiency in Neural Networks

While most research focuses on the energy costs of training AI models, inference, the deployment phase where models are used for actual tasks, accounts for up to 70-90% of the total energy consumption across a model's lifecycle (Wu et al., 2019; Desislavov, Martínez-Plumed and Hernández-Orallo, 2023) . This critical insight necessitates a fundamental reorientation of sustainability efforts in AI development, prioritizing inference optimization over training efficiency.

Energy consumption in neural network inference depends on multiple factors including model size, quantization strategy, and hardware implementation. Recent work has established that quantization can achieve substantial energy savings, with studies reporting up to 37% energy reduction without accuracy loss in Convolutional Neural Networks (Klhufek et al., 2024).

The relationship between model parameters and energy consumption follows predictable patterns, with larger models requiring proportionally more energy per inference operation. However, quantization can significantly alter this relationship, enabling energy-efficient deployment of larger models, but the magnitude of these effects is not clearly understood. Aggressive quantization methods provide substantial energy savings while maintaining acceptable accuracy levels, particularly for larger models.

3. Methodology

3.1 Experimental Setup

We conducted comprehensive experiments across the Qwen3 model family, evaluating six model sizes: 0.6B, 1.7B, 4B, 8B, 14B, and 32B parameters. Qwen3 is one of the few models which has the weights available at various sizes, ideal for scaling analysis. All the experiments were performed on NVIDIA L40s GPUs; most models on a single GPU, with few larger few models (32B) using two GPUs. These GPUs have 48GB GDDR6 vRAM and native FP8 support. Each model was quantized using six different methods:

- FP16 (Baseline): Original 16-bit floating-point precision
- GPTQ 4-bit: 4-bit quantization using GPTQ algorithm, activations in FP16
- GPTQ 8-bit: 8-bit quantization using GPTQ algorithm, activations in FP16
- AWQ: Activation-aware weight quantization, weights in INT8, activations in FP16
- FP8 W8A8: 8-bit floating-point quantization, weights and activations rounding to nearest
- INT8 W8A8: 8-bit weight and activation round-to-nearest quantization

All models were quantized locally using the llm-compressor library (Red Hat AI and vLLM Project, 2024). We calibrated both AWQ and GPTQ methods using 256 samples randomly selected from the Pile validation dataset (mit-han-lab/pile-val-backup), applying default calibration parameters.

While SmoothQuant was not tested as a standalong quantization method, we adopted its distribution smoothing techniques to mitigate activation outliers in FP8 W8A8 and INT8 W8A8 quantization methods.

3.2 Benchmark Selection

Our benchmark selection encompasses diverse evaluation dimensions essential for comprehensive model assessment. Each benchmark was chosen to evaluate specific aspects of model capability:

The ARC Challenge benchmark evaluates scientific reasoning through multiple-choice questions requiring complex inference. The MMLU (Massive Multitask Language Understanding) benchmark evaluates multitask language understanding across 57 subjects, providing a broad assessment of world knowledge and reasoning capabilities. TruthfulQA assesses the factual accuracy and truthfulness of model responses across 817 questions spanning 38 categories, addressing concerns about model reliability.

GSM8K focuses on mathematical reasoning capabilities through grade-school mathematical problems.

HumanEval specifically measures code generation abilities through 164 programming tasks, representing a crucial capability for many applications.

MetaBench provides a sparse benchmark for reasoning and knowledge evaluation in large language models. These benchmarks collectively provide a comprehensive evaluation framework for assessing model capabilities across different domains.

3.3 Performance Metrics

We collected comprehensive performance metrics divided into two categories: accuracy metrics, and efficiency metrics.

Accuracy metrics include benchmark scores across all evaluation tasks, enabling assessment of quantization impact on model capabilities. Benchmarks were done using the LM Evaluation Harness (Sutawika et al., 2024)

While perplexity is commonly used as an intrinsic metric for quantized models, it doesn't always correlate well with downstream task performance, and two models with similar perplexity scores can show significantly different capabilities on practical applications – that is why we focused on benchmark specific accuracy metrics.

For efficiency metrics, we measured tokens per second and energy per output token. The experiments include measurements across different context lengths (300, 500, 1000, 4000 and 8000 tokens) and output lenghts (100, 500, 1000, 4000 and 8000). The results were collected as an average across 100 prompts. Inference was

measured via an offline vLLM inference server. Energy was measured with the Zeus framework and measured only GPU energy usage - no specialized setup was implemented apart from including the measuring functions in the inference process.

4. Results and Analysis

4.1 Quantization Impact on Model Performance

Contrary to other research our analysis of Qwen3 does not reveal clear scaling patterns in quantization resilience across model sizes. For some benchmarks, data demonstrates that larger models consistently maintain higher accuracy retention rates when subjected to aggressive quantization methods. Retention refers to the fraction of a model's original (FP16) performance that remains after it has been quantized.

Table 1: Quantization resiliency across model sizes for Metabench benchmark

Model size	AWQ	GPTQ-4bit	GPTQ-8bit	INT8	FP8
0.6B	90.07%	91.25%	97.02%	100.94%	93.52%
1.7B	91.84%	95.83%	100.13%	95.01%	97.86%
4B	95.32%	96.73%	99.53%	97.90%	98.50%
8B	96.36%	97.08%	99.12%	97.43%	95.58%
14B	98.05%	100.15%	99.18%	98.05%	100.56%
32B	98.28%	97.43%	100.02%	99.99%	100.97%

For Metabench benchmark, AWQ, GPTQ-4bit and FP8 show scaling patterns where models are more resilient to quantization as they grow in size. Parameter redundancy at larger model sizes keeps the quantization error to a minimum. Interestingly, GPTQ-8bit and INT8 show better resiliency at smaller model sizes.

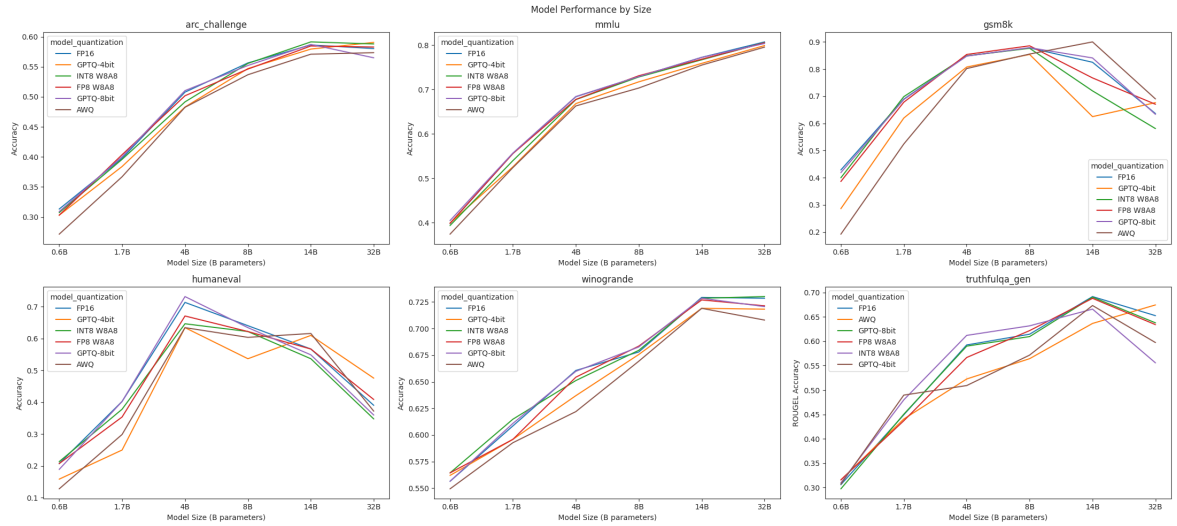

Figure 1: Model performance by size for 6 benchmarks - we evaluate zero-shot accuracy on ARC, MMLU, GSM8, HunamEval, WinoGrande and TruthfulQA

The mathematical reasoning tasks (GSM8K) show particular sensitivity to quantization effects. Smaller models experience pronounced degradation, while larger models – especially Qwen3-14B, have large discrepancies between quantization methods, AWQ even performing significantly better than the FP16 baseline.

Code generation capabilities (HumanEval) demonstrate similar scaling performance. The evaluation suggests that complex coding and mathematical abilities in large language models degrades even without quantization – possibly due to overthinking. The GPTQ-8bit and AWQ quantization methods even performing better than the FP16 baseline.

In language understanding tasks (MMLU, ARC Challenge and WinoGrande), quantization error and scaling seem to be as expected – with consistent errors which reduce and stabilise as we scale the model. Larger models maintain strong reasoning capabilities even under aggressive quantization.

Factual accuracy (TruthfulQA) shows consistent degradation patterns across model sizes, suggesting that truthfulness preservation is more dependent on quantization method than model scale.

4.2 Energy Efficiency Characteristics

Some quantization methods deliver substantial energy savings across all model sizes. Our measurements reveal consistent patterns in energy reduction effectiveness:

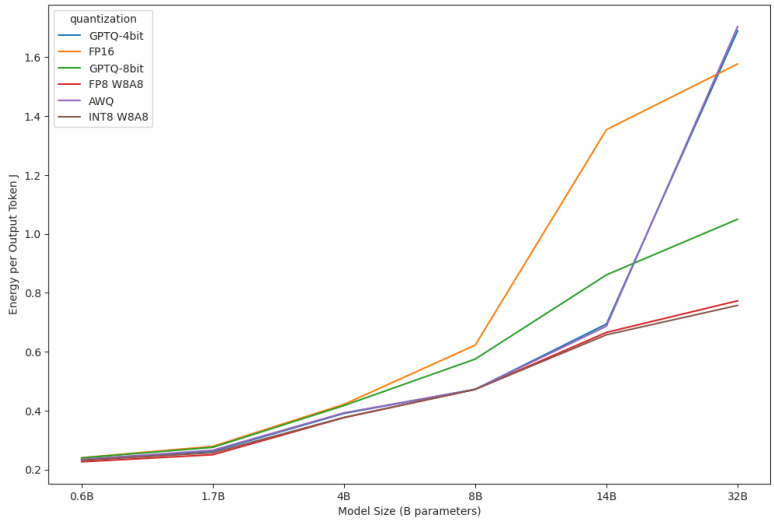

Figure 2: Energy use per output token on across model sizes using different quantization methods for context length 4000 tokens and output length 1000 tokens

The 32B model with GPTQ 4-bit quantization consumes 1.69 J per output token compared to 1.58 J for the FP16 baseline, while the FP8 quantized model consumes only 0.77 J, representing a improvement of more than 2x in energy efficiency. INT8 and FP8 quantized models perform consistently better than other methods, and their energy efficiency improves on larger model sizes. This is mostly due to the native hardware support of INT8 and FP8 on modern GPUs – which reduces not only the memory footprint, but also the amount of computations necessary.

The 32B model sizes quantized to FP16, GPTQ-4bit, GPTQ-8bit and AWQ had to be run on 2 GPUs. Although we ran experiments with various context lengths and output lengths, the result were consistent to the ones shown in Figure 2. This improvement enables deployment of larger models within similar energy budgets as smaller unquantized models – where a 32B FP8 models energy requirements can be compared to a 8B unquantized models energy requirements.

4.3 Throughput Performance Analysis

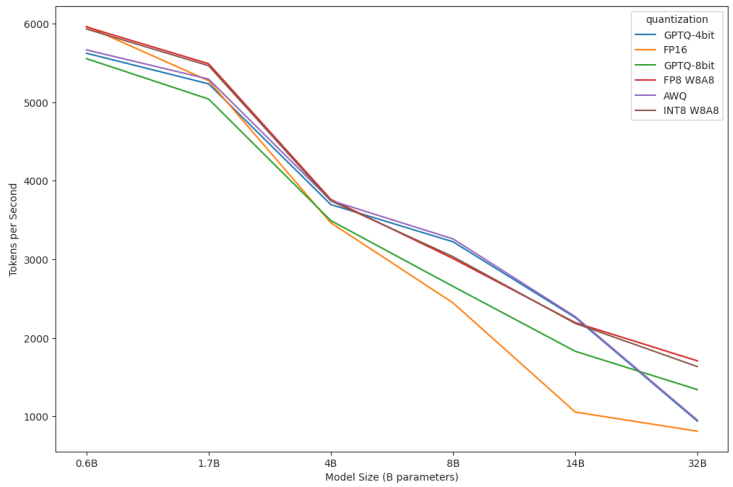

Figure 3: Tokens per second throughput across model sizes using different quantization methods for context length 4000 tokens and output length 1000 tokens

Inference throughput improvements scale predictably with quantization. Smaller model sizes perform well on modern GPUs in any format, while larger models show better throughput performance with quantization across all quantization methods.

FP8 and INT8 quantization provide approximately 2x throughput improvement on larger models compared to FP16 baseline across all context length. GPTQ-4bit and AWQ show almost identical throughput performance as expected, sine both of them use 4-bit representation of weights with 16-bit activations.

5. Practical Implications

The quantization tolerance of larger models fundamentally challenges conventional deployment strategies. Our findings suggest that deploying a quantized large model often provides better performance than deploying a smaller unquantized model within similar resource constraints.

For production environments requiring maximum efficiency, INT8 or FP8 W8A8 RTN quantization offers substantial benefits, particularly when combined with larger models that demonstrate quantization resilience. GPTQ-8bit quantization emerges as the preferred method for applications requiring high accuracy preservation with moderate efficiency gains. GPTQ methods offer excellent compression ratios with good accuracy preservation, making them suitable for memory-constrained deployments. The GPTQ-4-bit variant has better accuracy across benchmarks than AWQ, while being on par with energy-efficiency and throughput.

FP8 and INT8 W8A8 provide maximum throughput improvements and maximum energy efficiency with simple implementation requirements, making it attractive for high-volume inference scenarios where slight accuracy degradation is acceptable.

The ability to deploy larger, more capable models within similar energy budgets as smaller unquantized models represents a paradigm shift in deployment strategy optimization. This finding suggests that quantization enables "scaling up" rather than "scaling down" for energy-constrained scenarios.

6. Future Research Directions

The results suggest that advanced quantization methods lag behind simple INT8 and FP8 quantization in energy efficiency and performance, while providing small accuracy improvements in some cases. Adaptive quantization strategies that leverage natively implemented number formats – instead of using mixed precision, are a promising research direction. Hardware-aware quantization that considers specific accelerator architectures could further optimize the energy-performance trade-offs identified in this work.

Long-term stability analysis of quantized models in production environments requires investigation. Understanding whether quantization effects accumulate or remain stable over extended deployment periods together with model shifts is crucial for production planning.

This study focuses exclusively on the Qwen3 model family, which may limit generalizability to other architectures. The scaling patterns observed may not transfer directly to models with different architectural choices or training procedures.

The evaluation was conducted under controlled conditions that may not reflect real-world deployment scenarios with variable workloads and diverse input distributions. Production environments may exhibit different performance characteristics.

7. Conclusion

This comprehensive analysis establishes fundamental findings in quantization effects across model sizes, providing crucial insights for efficient LLM deployment. Larger models demonstrate superior resilience to quantization in most cases. Additionally, we provided clear results on which quantization methods are the most energy efficient.

The substantial energy savings achieved through quantization offer significant benefits for sustainable AI deployment, while the throughput and evaluation results provide guidelines for quantization planning. These findings contribute to making large language models more accessible and environmentally sustainable through efficient inference optimization.

Our results demonstrate that quantization is not merely a compression technique but a fundamental enabler of scale-efficient deployment strategies. The ability to deploy larger, more capable models within similar resource constraints represents a significant advancement in LLM deployment optimization.

Ethics Declaration: No ethical approval was required for this research as it involved only computational experiments with publicly available datasets and did not involve human participants, animal subjects, or sensitive data.

AI Declaration: Artificial intelligence tools were used for proofreading and spellchecking to improve the clarity and accuracy of this paper, with all substantive content, analysis, and conclusions remaining the original work of the authors.

References

Desislavov, R., Martínez-Plumed, F. and Hernández-Orallo, J. (2023) "Compute and Energy Consumption Trends in Deep Learning Inference," Sustainable Computing: Informatics and Systems, 38, p. 100857. doi: 10.1016/j.suscom.2023.100857.

Dettmers, T. et al. (2023) "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv. doi: 10.48550/arXiv.2305.14314.

Dettmers, T. and Zettlemoyer, L. (2023) "The case for 4-bit precision: k-bit Inference Scaling Laws." arXiv. doi: 10.48550/arXiv.2212.09720.

Frantar, E. et al. (2023) "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers." arXiv. doi: 10.48550/arXiv.2210.17323.

Gholami, A. et al. (2024) "AI and Memory Wall." arXiv. doi: 10.48550/arXiv.2403.14123.

Hoffmann, J. et al. (2022) "Training Compute-Optimal Large Language Models." arXiv. doi: 10.48550/arXiv.2203.15556.

Klhufek, J. et al. (2024) "Exploring Quantization and Mapping Synergy in Hardware-Aware Deep Neural Network Accelerators," in 2024 27th International Symposium on Design & Diagnostics of Electronic Circuits & Systems (DDECS), pp. 1–6. doi: 10.1109/DDECS60919.2024.10508920.

Kuzmin, A. et al. (2023) "Pruning vs Quantization: Which is Better?" arXiv. doi: 10.48550/arXiv.2307.02973.

Lin, J. et al. (2024) "AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration." arXiv. doi: 10.48550/arXiv.2306.00978.

Nagel, M. et al. (2021) "A White Paper on Neural Network Quantization." arXiv. doi: 10.48550/arXiv.2106.08295.

Red Hat AI and vLLM Project (2024) LLM Compressor. Available at: https://github.com/vllm-project/llm-compressor (Accessed: August 31, 2025).

Sutawika, L. et al. (2024) EleutherAI/lm-evaluation-harness: v0.4.3. Zenodo. doi: 10.5281/zenodo.12608602.

van Baalen, M. et al. (2023) "FP8 versus INT8 for efficient deep learning inference." arXiv. doi: 10.48550/arXiv.2303.17951.

Wu, C.-J. et al. (2019) "Machine Learning at Facebook: Understanding Inference at the Edge," in 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 331–344. doi: 10.1109/HPCA.2019.00048.

Xiao, G. et al. (2024) “SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models.” arXiv. doi: 10.48550/arXiv.2211.10438.

Zhu, X. et al. (2024) "A Survey on Model Compression for Large Language Models." doi: 10.48550/arXiv.2308.07633.