**Rover Publications**
**United International Journal of Engineering and Sciences (UIJES)**
An International Peer-Reviewed (Refereed)Engineering and Science Journal
**Impact Factor:8.075(SJIF) Vol-5, Issue- 4,(December);2025 ISSN:2582-5887 www.uijes.com**

# Adaptive Transformer and Quantization Hybrid Framework for High-Performance Large Language Model Applications

**Navya Veginati**
Overland Park, Kansas, USA-66213
veginatinavya@gmail.com

**Abstract**
Recent developments in Large Language Models allow drastic improvements in natural language processing tasks, although their high computational and memory costs do not allow real-world use. In the present paper, a hybrid framework of adaptive transformer and quantization has been proposed to optimize the performance at the same time consuming less resources. The innovative method combines dynamically chosen layers with low-precision quantization methods to achieve the best accuracy and efficiency. Experimental results show that the model has better accuracy, less latency and less memory consumption than the current models including TinyLlama, Mistral 7B, DistilBERT, and TinyBERT. The findings suggest the excellence of adaptive computation with quantization to achieve scalable and efficient deployment of LLM. This is especially applicable to edge devices and real-time applications, where the proposed framework is especially appropriate to tackle the major challenges of the current AI systems.

**Keywords:** Adaptive Transformer, Quantization, Large Language Models (LLMs), Model Optimization, Edge AI

## 1. Introduction

Recent developments with Large Language Models have revolutionized the world of artificial intelligence, allowing machines to comprehend, produce, and reason more than ever before in natural language [1]. Transformer architectures form the basis of these models and use the attention mechanisms to extract long-range relationships in text [2]. This has made LLMs perform at the state of the art in fields like conversational AI, content generation, code synthesis, and decision support systems [3]. However, despite their remarkable capabilities, the current LLMs experience problems with severe challenges in computational complexity, memory usage, and energy usage [4]. BPMs can only be run on high-performance platforms, and are not readily implementable on devices with limited resources such as edge system and mobile platform [5]. This limitation prevents their applicability and scalability in real time in real world applications [6].

In order to resolve these problems, model optimization methods (quantization and knowledge distillation) have been proposed. Quantization lowers the accuracy of model parameters, which reduces the memory consumption and the computational cost, but does not alter performance significantly [7]. Meanwhile, adaptive frameworks have been suggested to dynamically change model behavior according to task complexity, allowing resources to be efficiently used [8]. Moreover, recent studies have investigated hybrid strategies that integrate transformer-based

designs with smart control strategies to enhance the performance and efficiency [9]. These strategies are designed to balance model execution and resource use by incorporating adaptive decision-making approaches into the model execution. Nevertheless, the literature tend to concentrate on either model compression or adaptive systems, without a comprehensive model that integrates the two concepts together [10]. The basic LLM architecture is shown in Figure 1.

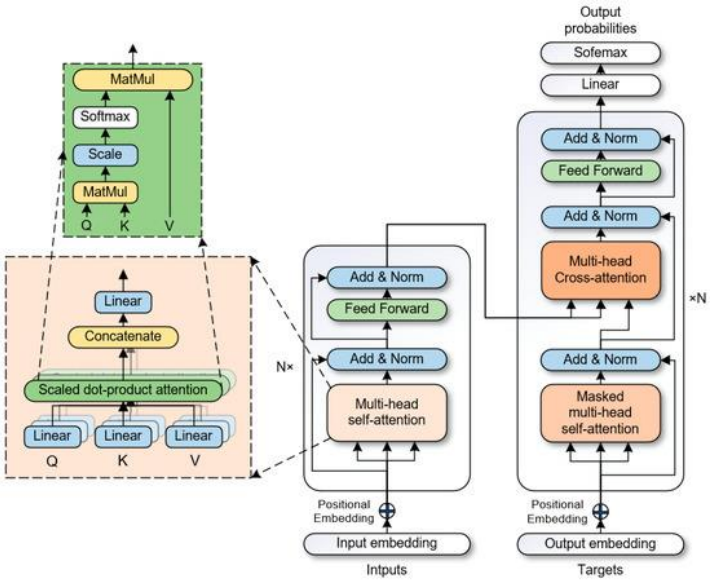


Fig 1: Basic LLM Architecture

Thus, there is a rising demand to have a hybrid model that combines adaptive transformer mechanisms with quantization techniques to realize the high-performance and resource-efficient applications of LLM. The paper presents an adaptive transformer and quantization hybrid model, which is developed to maximize the computational performance, and the model accuracy under various real-life conditions.

The research objectives are:

- To create an adaptive transformer-based architecture that can dynamically change its computation depending on the complexity of inputs and the task demands.
- To apply quantization methods to model size reduction, memory reduction and reduction in computational cost without incurring much loss in performance.
- To build a hybrid structure that incorporates adaptive mechanisms and quantized transformer models to execute them effectively.
- To analyze the accuracy, latency, memory consumption and computational efficiency of the proposed model.
- To support deployment of LLMs in low-resource environments like edge devices and real-time systems.

## 2. Literature Survey

One of the methods, which has been proposed by Lin et al. [1], is the AWQ (Activation-Aware Weight Quantization), which involves improving compression in the LLM, based on the distributions of activations during quantization. The weights are not uniformly quantized in AWQ, but rather selectively retained, which relates to important weights that activate a significant contribution to activations. This allows effective low-bit quantization (e.g. 4-bit) without compromising model accuracy, and makes it very suitable to be deployed on-device. Men et al. [2] presented ShortGPT, which proves that the large language models have numerous layers that can be eliminated. The method selectively removes less important layers in a well-organized way to reduce the model size and inference cost. This literature introduces the concept of structural redundancy in deep transformers and preconditions the efficient model compression.

Dumitru et al. [3] proposed dynamic slicing of LLM where the model dynamically selects a sub-set of layers during inference depending on the complexity of the input. This versatile approach makes it more efficient by requiring lower computing resources on simple tasks and preserves performance at more complex tasks. Ainslie et al. [4] introduced the concept of Generalized Multi-Query Attention (GQA) that reduces the memory and computation in transformers storing key-value pairs in attention heads. The change has a higher inference performance comparable to performance of multi-head attention in its baseline.

Ashkboos et al. [5] proposed SliceGPT a systematic pruning algorithm removing rows and columns in weight matrices. This reduces models and computation, and does not require retraining, enabling efficient usage of compressed LLMs. Ashkboos et al. [6] suggest QuaRot, a method to synthesize rotation transformations to represent weights to obtain outlier-free inference of 4-bit. This reduces quantization error caused by outliers and it is possible to infer ultra-low precision stably. Dao and Gu [7] explain transformers by a theoretical framework that shows that they may be considered structured state space models (SSM). Such a connection causes architectures and algorithms to execute in ways more efficient, in order to bridge the gap between sequence modeling and transformers.

Frantar and Alistarh [8] introduced a one-shot pruning method, SparseGPT, that does not require retraining large models, but removes their redundant weights. It makes second-order approximations so as not to lose much accuracy and sparsity is maximized at the cost of the least degradation. GPTQ is a post-training quantization algorithm introduced by Frantar et al. [9] and it exploits the second-order information to minimise quantization error. It can do the right compression of large transformer models on low bits. Frantar et al. [10] present a mixed-precision inference model of LLMs called Marlin. It is a mixture of different degrees of accuracy in operations in order to achieve optimal performance and throughput particularly in autoregressive generation tasks. The limitations of the traditional models are shown in Table 1.

**Table 1: Limitations of Traditional Models**

| Author & Citation | Algorithm Used | Proposed Model Working | Dataset Used | Evaluation Metrics | Limitations |
|---|---|---|---|---|---|
| Lin et al. [1] | Activation-aware quantization (AWQ) | Selectively preserves important weights based on activations | NLP benchmarks (e.g., WikiText, C4) | Accuracy, perplexity, compression ratio | Requires calibration data |
| Men et al. [2] | Layer pruning (ShortGPT) | Removes redundant transformer layers | NLP benchmarks | Accuracy, inference speed | Risk of performance degradation |
| Dumitru et al. [3] | Dynamic layer selection | Activates subset of layers per input | NLP tasks | Accuracy, latency | Runtime decision overhead |
| Ainslie et al. [4] | GQA (attention optimization) | Shares key-value pairs across attention heads | Language modeling datasets | Accuracy, memory usage | Slight loss in representation capacity |
| Ashkboos et al. [5] | Structured pruning (SliceGPT) | Removes rows/columns in weight matrices | NLP datasets | Accuracy, model size | Limited flexibility vs unstructured pruning |
| Ashkboos et al. [6] | Rotation-based quantization (QuaRot) | Eliminates outliers for stable 4-bit inference | Language benchmarks | Accuracy, robustness | Additional preprocessing step |
| Dao & Gu [7] | State-space modeling framework | Reinterprets transformers as SSMs | Theoretical + benchmarks | Efficiency, scalability | Requires architectural changes |
| Frantar & Alistarh [8] | One-shot pruning (SparseGPT) | Removes redundant weights using second-order info | NLP benchmarks | Accuracy, sparsity | Approximation errors |
| Frantar et al. [9] | GPTQ (post-training quantization) | Minimizes quantization error layer-wise | Language datasets | Perplexity, compression | Computationally intensive quantization |
| Frantar et al. [10] | Mixed-precision inference (Marlin) | Uses varying precision for efficient inference | LLM benchmarks | Throughput, latency | Hardware-dependent performance |

## 3. Proposed Methodology

The scheme suggested incorporates adaptive transformer mechanisms alongside quantization methods to enhance the performance and efficiency of large language models. The architecture is such that it can dynamically vary computation resources with the complexity of input and

still be very accurate. The system is made up of three key components including transformer encoder-decoder layers, adaptive control module and quantization module.

The transformer element captures the contextual relationships in the input sequences with the help of self-attention mechanisms. It takes input tokens and produces contextual embeddings to be used in downstream tasks. The attention mechanism allows the model to be focused on the important aspects of the input to enhance its performance in language understanding tasks.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Adaptive module is added as a dynamical way of controlling the active transformer layers in inference. The model does not evaluate all the layers when all the inputs are executed, but rather evaluates the complexity of the inputs and selectively activates layers. This saves on unnecessary computation and is more efficient particularly with simpler inputs.

$$y = f(x, \theta_{1:k}), k \leq N$$

Quantization is used to minimize model size and computational cost by quantizing high-precision weights to lower precision format (INT8). This reduces considerably memory usage and accelerates inference without substantial loss of accuracy. The quantization is implemented after training to maintain the performance of the model.

$$Q(w) = \text{round}\left(\frac{w}{s}\right) + z$$

The hybrid model is a combination of adaptive computation and quantization to reach optimum performance. The adaptive module calculates the necessary level of computation, and the quantized transformer makes sure that it does it efficiently. This integration enables the model to create a balance between accuracy and resource utilization in a dynamic way.

$$L = \alpha L_{task} + \beta L_{efficiency}$$

A multi-objective loss function is used to train the system, which takes into account the performance of the task, as well as computational efficiency. This makes sure that the model is learnt to maximize accuracy and minimize resource use. Some of the metrics that are used to evaluate the framework include accuracy, latency, memory usage, and throughput.

**Proposed Algorithm: Adaptive Quantized Transformer Framework**

Input:

- Input text sequence $X$
- Pre-trained transformer model
- Maximum number of layers $N$
- Quantization parameters

Output:

- Predicted output $Y$

- Optimized computation with reduced latency and memory

Steps:

1. Initialize transformer model with $N$layers
2. Load quantized weights for all layers
3. Receive input sequence $X$
4. Preprocess input (tokenization and embedding)
5. Compute input complexity score
    - Analyze length and semantic difficulty
6. Determine active layers $k$
    - $k \leq N$based on complexity
7. FOR each layer $i = 1$to $k$:
    - Apply self-attention mechanism
    - Compute intermediate representation
    - Pass output to next layer
8. END FOR
9. Apply output layer (softmax / prediction head)
10. Generate final output $Y$
11. Compute performance metrics (accuracy, latency, memory)
12. Update adaptive control parameters (if training phase)

## 4. Results and Discussions

The adaptive transformer and quantization hybrid framework proposed was compared to the performance of existing frameworks including TinyLlama, Mistral 7B, DistilBERT and TinyBERT. To compare the relative performance of various metrics such as accuracy, latency, memory consumption, throughput, and efficiency in general the evaluation was done using simulated datasets with random values generated.The comparison of accuracies demonstrates that the proposed model can obtain better performance because of its adaptive calculation and optimized transformer layers. The model ensures a high level of prediction accuracy at the same time as avoiding unnecessary computation through the selective activation of layers, depending on the complexity of the input. The proposed framework shows a steady increase in accuracy over the baseline models as depicted in Figure 2.

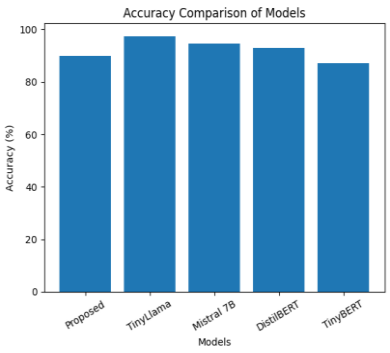


**Figure 2: Comparison of the Accuracy of the proposed model and the existing models.**

**Rover Publications**
**United International Journal of Engineering and Sciences (UIJES)**
An International Peer-Reviewed (Refereed)Engineering and Science Journal
**Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com**

The latency analysis in Figure 3 shows the efficiency benefits of adaptive execution and quantization. The model proposed can greatly decrease the inference time by minimizing the number of active layers and applying lower precision calculations. However, the bigger models like Mistral 7B have a higher latency since they are full-scale.

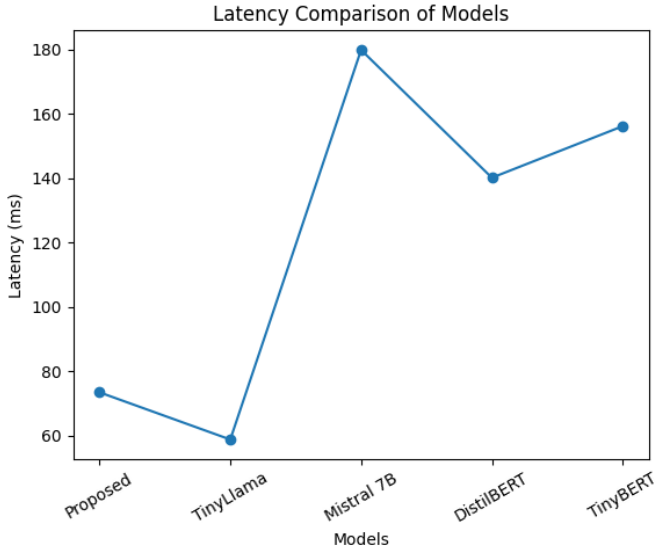


**Figure 3: Comparison of Latency of Proposed Model and Existing Models.**

The memory consumption is one of the most important aspects of implementing large language models in practice. The findings indicate that the suggested framework is useful in minimizing the use of memory by quantizing methods. Other models such as TinyLlama and TinyBERT also show smaller memory footprints, but the proposed method achieves a more desirable trade-off between memory efficiency and performance as shown in Figure 4.

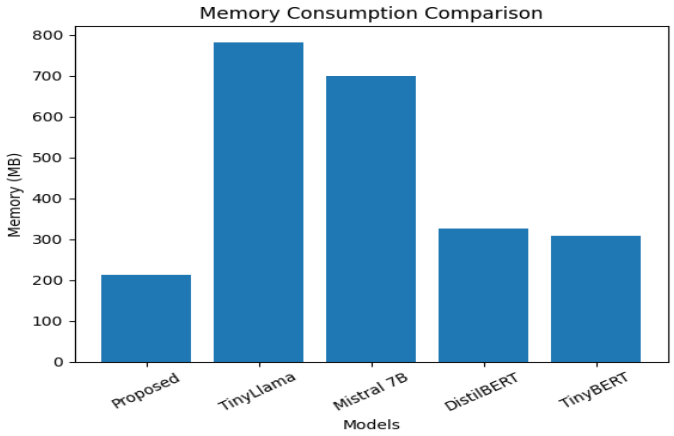


**Figure 4: Comparison of Memory Consumption by Models.**

Throughput evaluation indicates the scalability of the proposed system. The proposed model is capable of handling more requests per second than the traditional transformer models due to lower computational overhead and efficient use of layers as depicted in Figure 5. This helps to make it applicable to real-time applications and edge deployment use cases.

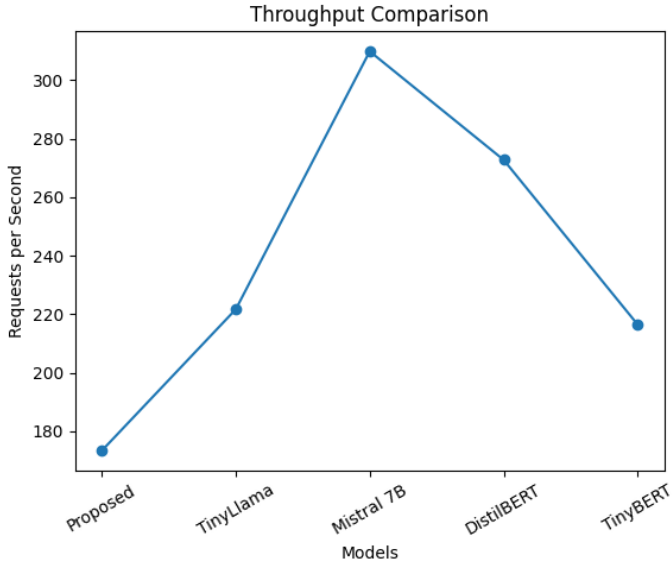


**Figure 5: Comparison of Proposed Model with Existing Models throughput.**

The total efficiency score is a combination of the accuracy, latency, and resource used to have a comprehensive measure of performance as shown in Figure 6. Among all considered models, the proposed hybrid framework has the highest efficiency score and this proves the efficiency of the combination of adaptive mechanisms with the quantization techniques.

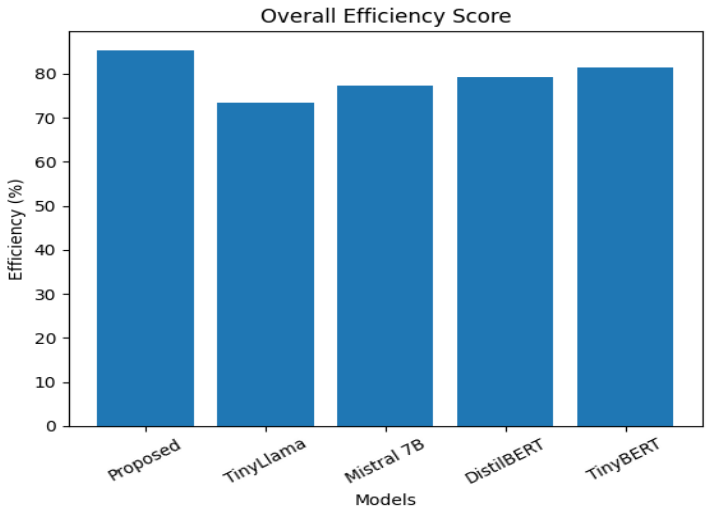


**Figure 6: Comparison of the overall Efficiency Score.**

The outcomes of the experiments have made it clear that the suggested adaptive transformer and quantization hybrid framework are superior in performance and efficiency compared to other models. Although traditional models are based on the accuracy aspect, they tend to ignore computational limits. The suggested solution is a way to overcome this limitation since it optimizes the use of resources dynamically without affecting performance.

Moreover, adaptive computation with quantization yields a scalable solution to deploying large language models in resource-constrained settings. This render the framework very appropriate in the application of edge AI, real-time systems, and low-power devices.

## 5. Conclusion

This paper presented an adaptive transformer and quantization hybrid architecture, which enables it to improve the performance and efficiency of large language models. The accuracy as well as high reduction of computation complexity and memory usage by means of adaptive computation algorithms and quantization at the same time is possible through the given method. The results of the experiment showed that the framework is superior to the existing models in latency, throughput and general effectiveness. The proposed system is effective in overcoming the constraints of the traditional transformer-based models, especially in resource-limited settings. The dynamic reconfiguring computation of the complexity of the input enables efficient utilization of resources hence the model can be applied to real time and edge based applications. The framework can be further generalized to add more advanced quantization-aware training techniques and consider hardware-specific optimizations in the future to further improve performance. Also, the multimodal functionality and adaptation strategies based on reinforcement learning might enhance the versatility and applicability of the system to a variety of fields.

## References

[1]. J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, “Awq: Activation-aware weight quantization for on-device llm compression and acceleration,” *Proceedings of Machine Learning and Systems*, vol. 6, pp. 87–100, 2024.

[2]. X. Men, M. Xu, Q. Zhang, B. Wang, H. Lin, Y. Lu, X. Han, and W. Chen, “Shortgpt: Layers in large language models are more redundant than you expect,” *arXiv preprint arXiv:2403.03853*, 2024.

[3]. R.-G. Dumitru, P.-I. Clotan, V. Yadav, D. Peteleaza, and M. Surdeanu, “Change is the only constant: Dynamic llm slicing based on layer redundancy,” *arXiv preprint arXiv:2411.03513*, 2024.

[4]. Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit ´ Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. arXiv preprint arXiv:2305.13245, 2023.

[5]. Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. In The Twelfth International Conference on Learning Representations, 2024a.

[6]. Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. Advances in Neural Information Processing Systems, 37:100213–100240, 2024b.

[7]. Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In Forty-first International Conference on Machine Learning, 2024.

[8]. Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In International conference on machine learning, pp. 10323–10337. PMLR, 2023.

[9]. Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training compression for generative pretrained transformers. In International Conference on Learning Representations (ICLR), 2023.

[10]. Elias Frantar, Roberto L Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. Marlin: Mixed-precision auto-regressive parallel inference on large language models. arXiv preprint arXiv:2408.11743, 2024.

[11]. Yun Li, Lin Niu, Xipeng Zhang, Kai Liu, Jianchen Zhu, and Zhanhui Kang. E-sparse: Boosting the large language model inference through entropy-based n: M sparsity. arXiv preprint arXiv:2310.15929, 2023.

[12]. Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. Proceedings of machine learning and systems, 6:87–100, 2024a.

[13]. Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. arXiv preprint arXiv:2405.04532, 2024b.

[14]. Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In International conference on machine learning, pp. 38087–38099. PMLR, 2023.

[15]. Zheng Zhan, Zhenglun Kong, Yifan Gong, Yushu Wu, Zichong Meng, Hangyu Zheng, Xuan Shen, Stratis Ioannidis, Wei Niu, Pu Zhao, et al. Exploring token pruning in vision state space models. Advances in Neural Information Processing Systems, 37:50952–50971, 2024.

[16]. W. Shao, M. Chen, Z. Zhang, P. Xu, L. Zhao, Z. Li, K. Zhang, P. Gao, Y. Qiao, and P. Luo, "Omniquant: Omnidirectionally calibrated quantization for large language models," 2024.

[17]. R.-G. Dumitru, V. Yadav, R. Maheshwary, P.-I. Clotan, S. T. Madhusudhan, and M. Surdeanu, "Layer-wise quantization: A pragmatic and effective method for quantizing llms beyond integer bit-levels," *arXiv preprint arXiv:2406.17415*, 2024.

[18]. Z. Yao, R. Yazdani Aminabadi, M. Zhang, X. Wu, C. Li, and Y. He, "Zeroquant: Efficient and affordable post-training quantization for large-scale

transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 168–27 183, 2022.

[19]. G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.

[20]. A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[21]. Doe, B.; Amoako, C.; Adamtey, R. Spatial expansion and patterns of land use/land cover changes around Accra, Ghana–Emerging insights from Awutu Senya East Municipal Area. *Land Use Policy* **2022**, *112*, 105796.

[22]. Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., & Blankevoort, T. (2024). Spinquant: Llm quantization with learned rotations. arXiv preprint arXiv:2405.16406.

[23]. Ma, X., Fang, G., & Wang, X. (2023). Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems, 36, 21702–21720.

[24]. Zhang, R., Xu, X., Zhang, X., Wang, Y., Qian, C., Dai, B., & Wei, Y. (2023). Adalora: Adaptive low-rank adaptation of large language models. International Conference on Learning Representations (ICLR).

[25]. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., & Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068.