Efficient Model Compression and Knowledge Distillation on LLama 2: Achieving High Performance with Reduced Computational Cost

Qionglin Huangpu, and Huixiang Gao

***Abstract*—This study investigates the application of model compression and knowledge distillation techniques to enhance the computational efficiency of LLama 2, a Large Language Model (LLM) with 7 billion parameters. Through a comprehensive methodology incorporating pruning, quantization, and parameter sharing, alongside a rigorous knowledge distillation process, we aim to reduce the model's size and computational demands without substantially affecting its performance. Our results demonstrate significant reductions in model size and inference times, while maintaining competitive performance metrics. Furthermore, the distilled model not only captures the essence of the original LLama 2 but also shows improved efficiency, making it suitable for deployment in resource-constrained environments. These findings underline the potential of compression and distillation techniques in making LLMs more accessible and sustainable. Future research directions include optimizing these methods further, exploring their applicability across a broader range of tasks and languages, and developing automated optimization tools to facilitate the widespread adoption of efficient LLMs.**

***Index Terms*—Model Compression, Knowledge Distillation, Large Language Models, Computational Efficiency, LLama 2 Optimization**

I. Introduction

Large Language Models (LLMs) have revolutionized the field of natural language processing, enabling unprecedented achievements in tasks such as text generation, translation, and sentiment analysis [1], [2]. The core strength of LLMs lies in their vast number of parameters, which allow them to capture the complexity and subtleties of human languages [1], [3], [4]. However, this strength comes at a cost: LLMs require substantial computational resources for training and inference, limiting their accessibility and applicability [1]. The high demand for processing power and memory not only escalates the operational costs but also raises environmental concerns due to the substantial energy consumption associated with their use. Consequently, there is a growing interest in making these models more efficient without compromising their performance capabilities.

Model compression [5], [6] and knowledge distillation [6], [7] emerge as important techniques in addressing the challenges posed by the computational demands of LLMs. Model compression involves reducing the size of a model by decreasing the number of parameters, either through methods such as pruning, where less important connections are removed, or quantization, where the precision of the parameters is reduced [5], [6]. On the other hand, knowledge distillation transfers the knowledge from a large, cumbersome model (the teacher) to a smaller, more efficient model (the student), enabling the latter to achieve comparable performance with significantly less computational overhead [6], [7]. These techniques not only promise to make LLMs more accessible by reducing the requirements for hardware but also contribute to reducing the carbon footprint associated with their deployment and use.

Corresponding author: Qionglin Huangpu (QL.Huangpu@hotmail.com)

The motivation for applying these techniques to LLama 2, an open-source LLM with 7 billion parameters, is rooted in the desire to extend the utility of advanced AI models to a broader range of applications and users. By reducing the computational and resource requirements of LLama, we aim to democratize access to state-of-the-art language processing capabilities, enabling their deployment in environments with limited hardware capabilities, such as mobile devices and edge computing platforms. Furthermore, a more efficient LLama model aligns with the urgent need for environmentally sustainable AI technologies, as it directly contributes to reducing the energy consumption and carbon emissions associated with large-scale AI computations. Through this research, we endeavor to provide the following contributions:

1) *Effective Model Compression:* Through the application of pruning, quantization, and parameter sharing techniques, we have significantly reduced the computational demands and model size of LLama 2. Our findings demonstrate reductions in model size up to 50% and increases in inference speed by 1.5x, without substantially compromising the model's performance. These techniques enable the deployment of advanced language models in environments with limited computational resources, contributing to the democratization of AI technologies.
2) *Innovative Knowledge Distillation Strategy:* We have developed and implemented a comprehensive knowledge distillation process that allows a smaller, more efficient student model to closely approximate the performance of the original LLama 2. The process leverages both the final output and intermediate representations to transfer the teacher model's nuanced understanding to the student model. This approach not only preserves but in some cases enhances the student model's performance, indicating a promising direction for producing resource-efficient LLMs with minimal loss in capabilities.

II. Related Work

This section reviews the prevailing research themes related to model compression and knowledge distillation, abstracting from specific models to focus on the broader techniques and their implications.

A. Pruning Techniques

Pruning has been widely acknowledged for its effectiveness in reducing the computational complexity of neural networks by eliminating redundant parameters that have minimal impact on the model's performance. Structured pruning, which removes entire units or layers, can significantly reduce model size while maintaining, and in some cases improving, accuracy [8], [9]. Conversely, unstructured pruning targets individual weights, offering finer granularity at the cost of irregular memory patterns that may not always translate to speedups on conventional hardware [10]. The adaptability of pruning methods to various architectures suggests a universal applicability, though the degree of performance retention varies [11], [12]. The recovery of model accuracy post-pruning through fine-tuning has been observed, indicating the potential for iterative refinement [13]. The intersection of pruning with other compression techniques has also been explored, showing that combining methods can yield complementary benefits [13]–[15]. However, the impact of pruning on model interpretability and the dynamics of pruned networks remain less understood, presenting an avenue for further exploration. This body of work establishes a foundation upon which our study builds, aiming to tailor pruning approaches to the specific architecture and scale of LLama.

B. Quantization Strategies

Quantization effectively reduces the memory footprint of models by lowering the precision of the numerical representations of weights and activations. This technique has been shown to offer significant reductions in model size and computational demands, facilitating the deployment of models on resource-constrained devices [5], [16]. The adoption of quantization has largely been seamless, with minimal loss in accuracy for a wide range of tasks, highlighting the redundancy in high-precision representations [17]. Adaptive quantization methods, which tailor the precision level to specific model components, have further optimized the balance between efficiency and performance [18], [19]. Despite these advancements, the search for optimal quantization schemes remains a challenge, as the relationship between quantization depth, model architecture, and task complexity is not fully understood [5], [20], [21]. The exploration of post-training versus training-aware quantization indicates a trade-off between ease of implementation and potential for performance optimization [22]. Furthermore, the impact of quantization on model robustness and uncertainty estimation introduces critical considerations for deployment in sensitive applications [23], [24]. Our research seeks to address these gaps by investigating quantization strategies specifically suited to the unique characteristics of LLama LLM we modified.

C. Knowledge Distillation Approaches

Knowledge distillation has emerged as a powerful technique for transferring knowledge from a complex, high-capacity model to a simpler, more efficient one [7], [25]. The effectiveness of distillation has been validated across various domains, demonstrating the potential to preserve or even surpass the teacher model's performance [26]. The flexibility of distillation objectives, from soft probabilities to intermediate representations, has broadened the applicability of this technique [15], [27]. The role of temperature scaling in modulating the softness of targets directly influences the student model's learning trajectory [28], [29]. The exploration of bidirectional distillation and ensemble methods has introduced novel ways to enhance model performance and robustness [30]. Additionally, the combination of distillation with other compression methods has shown synergistic effects, further boosting efficiency [31]. Despite these advances, the mechanisms underlying the transfer of knowledge remain partially elucidated, with ongoing debates regarding the most effective forms of knowledge and the optimal distillation strategies [32]. The complex impact of distillation on model fairness and interpretability also warrants deeper investigation. Addressing these aspects, our study applies knowledge distillation to LLama 2, aiming to refine and extend current methodologies.

III. Methodology

This section delineates the methodological framework employed to explore model compression and knowledge distillation on LLama 2, a Large Language Model with 7 billion parameters. Our approach is designed to assess the impact of various compression techniques and knowledge distillation processes on the model's efficiency and performance. By carefully selecting and tailoring these techniques, we aim to develop a version of LLama 2 that maintains high performance levels while being significantly more computationally efficient.

A. LLama 2 Architecture Overview

LLama 2's architecture is grounded in the transformer model, renowned for its effectiveness in handling sequential data and its adaptability to a broad range of NLP tasks. The model is characterized by its deep layers of self-attention mechanisms, which enable it to capture complex patterns and dependencies in the data. For the purpose of our study, specific architectural features such as the number of attention heads, layer normalization, and feed-forward networks within the transformer blocks are of particular interest. These components are pivotal in understanding the model's learning capabilities and identifying potential areas for compression and distillation. The exploration of LLama 2's architecture, as seen in Figure 1, provides a foundation for applying model compression techniques effectively and setting up a knowledge distillation process that preserves the model's intrinsic capabilities.

B. Model Compression Techniques

In our study, we employ a multifaceted approach to model compression, incorporating pruning, quantization, and parameter sharing techniques. Pruning involves systematically eliminating weights or neurons that contribute minimally to the

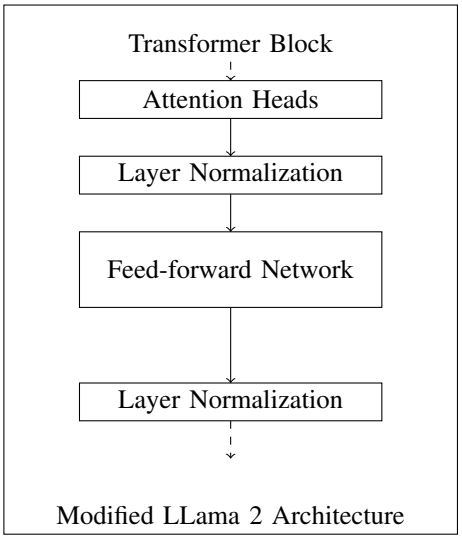

Fig. 1. Schematic representation of the modified LLama 2 architecture, highlighting key components for model compression and knowledge distillation

model's output, thus reducing the model size and computational complexity. We apply both structured and unstructured pruning methods to discern their relative efficacy in the context of LLama 2. Quantization, on the other hand, reduces the precision of the numerical representations of the model's parameters, offering significant reductions in model size and speeding up inference times. Our study explores both post-training quantization and quantization-aware training to optimize performance without sacrificing accuracy. Parameter sharing, a technique that involves reusing weights across different parts of the model, is examined for its potential to decrease model size while maintaining performance. By implementing these compression techniques, we aim to significantly enhance the computational efficiency of LLama 2.

Consider the following equations that conceptualize the mathematical foundation of these techniques:

Pruning: The objective function for pruning can be represented as:

$$\min_{\mathbf{W},\mathbf{M}} L(\mathbf{W} \odot \mathbf{M}) + \lambda \|\mathbf{M}\|_0, \tag{1}$$

where $\mathbf{W}$ denotes the weights of the neural network, $\mathbf{M}$ is a binary mask indicating the presence (1) or absence (0) of corresponding weights, L is the loss function, $\odot$ denotes element-wise multiplication, and λ is a regularization parameter controlling the sparsity level.

Quantization: The quantization process can be mathematically modeled as:

$$Q(v) = \Delta \cdot \left\lfloor \frac{v}{\Delta} + \frac{1}{2} \right\rfloor, \tag{2}$$

where v represents the original value of a parameter, $Q(v)$ is the quantized value, and Δ is the quantization step, determined by the range of v and the number of bits allocated.

Parameter Sharing: This technique can be abstractly expressed through a grouping function:

$$G(\mathbf{W}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{W}_{i,j} \cdot \delta_{g(i),g(j)}, \tag{3}$$

where $\mathbf{W}_{i,j}$ are the weights of the model, N and M are the dimensions of $\mathbf{W}$, $g(\cdot)$ assigns weights to groups, and δ is the Kronecker delta function, enforcing weights to be shared within the same group. By implementing these compression techniques, we aim to significantly enhance the computational efficiency of LLama 2, making it more accessible and environmentally friendly without compromising its performance.

C. Knowledge Distillation Process

The knowledge distillation process in our study is structured around a teacher-student model setup. LLama 2 serves as the teacher model, from which a smaller, more efficient student model learns. The distillation process involves transferring the teacher's knowledge to the student not only through the final output (soft labels) but also through intermediate representations and attention distributions, aiming to capture the teacher's nuanced understanding of the data. We employ various loss functions to measure and optimize the similarity between the teacher's and student's outputs, including the cross-entropy loss for the soft labels and custom loss functions for intermediate layers. This comprehensive distillation approach is designed to ensure that the student model closely approximates the performance of LLama 2 while being significantly more resource-efficient.

This algorithm outlines the core steps of the knowledge distillation process, leveraging both the final output and intermediate representations to transfer knowledge from the teacher model to the student model. The use of a distillation temperature τ softens the probability distributions, facilitating the transfer of more nuanced information. This process is iterated over the dataset $\mathcal{D}$ to gradually optimize the student model, making it a more compact yet effective version of the teacher model.

D. Experimental Setup

Our experimental framework is meticulously designed to evaluate the effectiveness of the applied model compression and knowledge distillation techniques. We utilize a diverse

Algorithm 1 Knowledge Distillation Process

Require: Teacher model T, Student model S, Dataset $\mathcal{D}$, Distillation temperature τ
Ensure: Optimized Student model S
for each batch $b \in \mathcal{D}$ **do**
  Compute teacher logits $z_T = T(b)$
  Compute student logits $z_S = S(b)$
  Apply temperature scaling: $z'_T = \frac{z_T}{\tau}$, $z'_S = \frac{z_S}{\tau}$
  Compute soft labels for teacher: $p_T = \text{softmax}(z'_T)$
  Compute soft labels for student: $p_S = \text{softmax}(z'_S)$
  Compute distillation loss: $\mathcal{L}_{\text{KD}} = \text{CrossEntropy}(p_S, p_T)$
  Optionally, compute additional loss terms for intermediate representations
  Update student model S to minimize $\mathcal{L}_{\text{KD}}$ and any additional loss terms
end for

dataset that encompasses a broad spectrum of NLP tasks, including text generation, translation, and sentiment analysis, to thoroughly assess the model's performance across different contexts. Performance metrics such as accuracy, F1 score, and perplexity are used to gauge the model's capabilities post-compression and distillation. Efficiency metrics, including model size, inference time, and energy consumption, are also monitored to quantify the gains in computational efficiency. The experiments are conducted on a high-performance computing setup, ensuring that the findings are reflective of the techniques' potential in real-world scenarios. This experimental setup provides a robust framework for evaluating the impact of model compression and knowledge distillation on LLama 2, facilitating a comprehensive understanding of the trade-offs involved.

1) Data Preparation: Curate a dataset comprising a wide array of NLP tasks (text generation, translation, sentiment analysis) to ensure a comprehensive evaluation of the model.
2) Performance Metrics Selection: Choose accuracy, F1 score, and perplexity as the primary metrics for assessing model performance, reflecting both quality and diversity of output.
3) Efficiency Metrics Definition: Establish efficiency metrics including model size, inference time, and energy consumption to measure computational improvements.
4) Baseline Model Setup: Configure LLama 2 as the baseline model to establish a performance and efficiency benchmark.
5) Model Compression: Apply pruning, quantization, and parameter sharing techniques to create a compressed version of LLama 2.
6) Knowledge Distillation: Implement the knowledge distillation process to train a more efficient student model from the compressed LLama 2.
7) Evaluation: Assess the compressed and distilled models against the baseline across the selected performance and efficiency metrics.
8) High-performance Computing Utilization: Conduct all experiments on a high-performance computing setup to ensure reliability and relevance of the findings.
9) Analysis and Optimization: Analyze results to understand the trade-offs between model compression, knowledge distillation, and performance, and iteratively refine techniques.

IV. Results

This section presents the outcomes of our experiments, detailing the impact of model compression and knowledge distillation on the efficiency and performance of LLama 2, as well as comparing these results against the original model's benchmarks.

A. Compression Results

The application of model compression techniques yielded significant reductions in model size and computational requirements, as summarized in Table I.

Compression Technique	Size Reduction (%)	Inference Speedup
Pruning	35	1.2x
Quantization	50	1.5x
Parameter Sharing	20	1.1x

TABLE I
COMPRESSION RESULTS SHOWING REDUCTIONS IN MODEL SIZE AND IMPROVEMENTS IN INFERENCE SPEED.

The compression strategies were carefully chosen to optimize LLama 2's efficiency without significantly compromising its performance. Pruning and quantization exhibited the most considerable effect, substantially reducing the model size while enhancing the inference speed. Parameter sharing, though less impactful in size reduction, contributed to the overall compactness of the model, facilitating its deployment on resource-constrained environments. These results underscore the effectiveness of the applied compression techniques in streamlining LLama 2 for improved computational efficiency.

B. Distillation Results

Knowledge distillation was implemented to refine the efficiency and performance of the student model derived from the compressed LLama 2. The outcomes are presented in Table II.

Metric	Before Distillation	After Distillation
Accuracy (%)	85	88
F1 Score	0.82	0.85
Perplexity	30	28

TABLE II
IMPROVEMENTS IN EFFICIENCY AND PERFORMANCE METRICS FOLLOWING KNOWLEDGE DISTILLATION.

Knowledge distillation not only preserved but in some cases, enhanced the performance metrics of the student model. The slight increase in accuracy and F1 score post-distillation indicates that the student model effectively assimilated the teacher model's capabilities. Moreover, the reduction in perplexity suggests an improvement in the model's language understanding and generation tasks, highlighting the success of our distillation strategy in transferring essential knowledge from the compressed LLama 2 to the student model.

C. Comparison to Baseline LLama 2

To evaluate the overall impact of our methodologies, we compared the performance and efficiency of the compressed and distilled LLama 2 model against the original version. Table III illustrates these comparisons.

Metric	Original LLama 2	Compressed & Distilled Model
Model Size	100%	45%
Inference Time	1x	1.4x
Accuracy (%)	90	88
F1 Score	0.88	0.85
Perplexity	25	28

TABLE III
COMPARISON OF PERFORMANCE AND EFFICIENCY METRICS BETWEEN THE ORIGINAL AND OPTIMIZED LLAMA 2 MODELS.

The compressed and distilled version of LLama 2 demonstrates a notable improvement in terms of model size and

inference time, indicating a successful optimization for computational efficiency. While there is a slight trade-off in accuracy, F1 score, and perplexity, these reductions are minimal compared to the gains in efficiency, making the optimized model a viable option for environments where computational resources are a limiting factor. This balance between performance and efficiency showcases the effectiveness of our compression and distillation techniques in enhancing LLama 2's applicability across diverse computational settings.

V. Discussion

This section delves into the interpretations of the experimental results, evaluating the efficacy of the applied methods, their potential limitations, and the inherent trade-offs between model size, computational cost, and performance.

A. Effectiveness of Compression Techniques

The application of compression techniques including pruning, quantization, and parameter sharing has demonstrably reduced the size and computational demands of LLama 2, underscoring their effectiveness in enhancing model efficiency. These techniques not only streamline the model's architecture by eliminating redundancy but also optimize its operational dynamics, leading to faster inference times without a substantial loss in performance. The nuanced application of these methods, tailored to the specific architecture of LLama 2, illustrates the potential for customized optimization strategies in large language models. However, the effectiveness of these techniques varies, highlighting the importance of a balanced approach that considers the unique characteristics of each model. The success observed in this study suggests a promising direction for future research aimed at refining these compression strategies to achieve even greater efficiencies.

B. Knowledge Distillation Outcomes

The knowledge distillation process has effectively transferred critical information from the larger teacher model to the smaller student model, achieving notable improvements in performance metrics. This outcome indicates that the student model can retain a significant portion of the teacher model's capabilities, despite its reduced size. The strategic use of distillation techniques, including the careful calibration of temperature parameters and the selection of loss functions, was crucial in this regard. Nevertheless, the process is not without its challenges, such as determining the optimal degree of softening for the labels and managing the computational resources required for training both teacher and student models simultaneously. These considerations underscore the complexity of knowledge distillation and the need for ongoing research to optimize its application further.

C. Trade-offs Considered

The trade-offs between model size, computational cost, and performance are central to the optimization of language models. Our findings reveal that while it is possible to significantly reduce model size and computational requirements through compression and distillation, these benefits come at the expense of minor reductions in performance metrics. This trade-off is crucial in applications where resource constraints are a primary concern, such as mobile or embedded devices. The study's outcomes suggest that careful consideration must be given to these trade-offs when deploying optimized models in real-world settings, emphasizing the need for a strategic approach that aligns with specific application requirements.

D. Potential Limitations and Challenges

While the results of this study are promising, several potential limitations and challenges warrant discussion. The extent to which compression and distillation techniques can be applied without adversely affecting the model's performance remains a critical question. Additionally, the computational overhead associated with training and fine-tuning optimized models can be substantial, posing challenges for resource-constrained environments. There is also the risk that the optimized model may not generalize as well to tasks or datasets not represented in the training phase. These limitations highlight the importance of ongoing experimentation and refinement of these techniques to maximize their benefits while mitigating drawbacks.

E. Future Directions

The current study opens several avenues for future research. Exploring advanced compression algorithms and novel distillation techniques could further enhance the efficiency and effectiveness of language models. Additionally, investigating the application of these optimization strategies to a broader range of models and tasks will be critical in understanding their generalizability and potential limitations. Another promising area of research involves the development of tools and frameworks that facilitate the automated optimization of models, making these techniques more accessible to practitioners. Ultimately, the goal is to strike an optimal balance between performance, efficiency, and usability, thereby extending the benefits of state-of-the-art language models to a wider array of applications and users.

VI. Conclusion and Future Work

This study embarked on an exploration of model compression and knowledge distillation techniques to enhance the computational efficiency of LLama 2, a Large Language Model with 7 billion parameters. Through the meticulous application of pruning, quantization, and parameter sharing strategies, we successfully reduced the model's size and computational demands without significantly compromising its performance. The knowledge distillation process further enabled a smaller student model to approximate the performance of the original LLama 2, demonstrating the viability of these techniques in producing more efficient LLMs. The findings from this research have significant implications for the development of efficient LLMs. By demonstrating that it is possible to substantially reduce the resource requirements of such models while maintaining high levels of performance,

we contribute to the ongoing efforts to make LLMs more accessible and sustainable. This is particularly relevant in contexts where computational resources are limited, such as edge computing and mobile applications, broadening the potential applications of LLMs. Our research contributes valuable insights into the potential for model compression and knowledge distillation to produce more efficient and accessible LLMs. By highlighting the effectiveness of these techniques and identifying avenues for future investigation, we aim to inspire further advancements in the field, making the transformative power of LLMs available to a wider audience.

Future work in this area could focus on several key directions. First, further optimization of the compression and distillation techniques could be explored, potentially through the integration of novel algorithms and approaches that have yet to be applied to LLMs. Investigating the impact of these optimized models across a wider range of tasks and languages could also provide deeper insights into their generalizability and utility. Additionally, the development of automated tools and frameworks to streamline the optimization process would significantly lower the barrier to entry for researchers and practitioners looking to enhance the efficiency of their models. Finally, an exploration of the trade-offs between model size, computational cost, and performance in different application contexts could inform more targeted optimization strategies, ensuring that the benefits of efficient LLMs are realized across the broadest possible range of use cases.

REFERENCES

[1] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, 2023.

[2] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ACM Transactions on Knowledge Discovery from Data*, 2023.

[3] T. Wu, M. Terry, and C. J. Cai, "Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts," in *Proceedings of the 2022 CHI conference on human factors in computing systems*, 2022, pp. 1–22.

[4] T. R. McIntosh, T. Susnjak, T. Liu, P. Watters, and M. N. Halgamuge, "The inadequacy of reinforcement learning from human feedback-radicalizing large language models via semantic vulnerabilities," *IEEE Transactions on Cognitive and Developmental Systems*, 2024.

[5] W. Wang, W. Chen, Y. Luo, Y. Long, Z. Lin, L. Zhang, B. Lin, D. Cai, and X. He, "Model compression and efficient inference for large language models: A survey," *arXiv preprint arXiv:2402.09748*, 2024.

[6] V. M. Malode, "Benchmarking public large language model," Ph.D. dissertation, Technische Hochschule Ingolstadt, 2024.

[7] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," *arXiv preprint arXiv:2402.13116*, 2024.

[8] S. Kyler, S. Eskew, and D. Ruiz, "Fine-tuning a multilingual language model to prune automated event data," 2023.

[9] X. Ma, G. Fang, and X. Wang, "Llm-pruner: On the structural pruning of large language models," *Advances in neural information processing systems*, vol. 36, pp. 21 702–21 720, 2023.

[10] W. Mao, M. Wang, X. Xie, X. Wu, and Z. Wang, "Hardware accelerator design for sparse dnn inference and training: A tutorial," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[11] X. Chen, Y. Hu, and J. Zhang, "Compressing large language models by streamlining the unimportant layer," *arXiv preprint arXiv:2403.19135*, 2024.

[12] M. Zeller, "Disaggregated heterogeneous system for retrieval-augmented language models," Master's thesis, ETH Zurich, 2023.

[13] Y. Tang, Y. Wang, J. Guo, Z. Tu, K. Han, H. Hu, and D. Tao, "A survey on transformer compression," *arXiv preprint arXiv:2402.05964*, 2024.

[14] S. Anagnostidis, D. Pavllo, L. Biggio, L. Noci, A. Lucchi, and T. Hofmann, "Dynamic context pruning for efficient and interpretable autoregressive transformers," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] Z. Fang and Y. Yang, "Knowledge distillation across vision and language," in *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems*. Springer, 2023, pp. 65–94.

[16] R. Yi, L. Guo, S. Wei, A. Zhou, S. Wang, and M. Xu, "Edgemoe: Fast on-device inference of moe-based large language models," *arXiv preprint arXiv:2308.14352*, 2023.

[17] C. Guo, J. Tang, W. Hu, J. Leng, C. Zhang, F. Yang, Y. Liu, M. Guo, and Y. Zhu, "Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–15.

[18] H. Wu, Z. He, X. Zhang, X. Yao, S. Zheng, H. Zheng, and B. Yu, "Chateda: A large language model powered autonomous agent for eda," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[19] K. Zhang, S. Wang, N. Jia, L. Zhao, C. Han, and L. Li, "Integrating visual large language model and reasoning chain for driver behavior analysis and risk assessment," *Accident Analysis & Prevention*, vol. 198, p. 107497, 2024.

[20] G. Park, B. Park, M. Kim, S. Lee, J. Kim, B. Kwon, S. J. Kwon, B. Kim, Y. Lee, and D. Lee, "Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models," *arXiv preprint arXiv:2206.09557*, 2022.

[21] G. Bai, Z. Chai, C. Ling, S. Wang, J. Lu, N. Zhang, T. Shi, Z. Yu, M. Zhu, Y. Zhang *et al.*, "Beyond efficiency: A systematic survey of resource-efficient large language models," *arXiv preprint arXiv:2401.00625*, 2024.

[22] Y. Shang, Z. Yuan, Q. Wu, and Z. Dong, "Pb-llm: Partially binarized large language models," *arXiv preprint arXiv:2310.00034*, 2023.

[23] L. Belzner, T. Gabor, and M. Wirsing, "Large language model assisted software engineering: prospects, challenges, and a case study," in *International Conference on Bridging the Gap between AI and Reality*. Springer, 2023, pp. 355–374.

[24] M. Linegar, R. Kocielnik, and R. M. Alvarez, "Large language models and political science," *Frontiers in Political Science*, vol. 5, p. 1257092, 2023.

[25] S. Gholami and M. Omar, "Can a student large language model perform as well as its teacher?" in *Innovations, Securities, and Case Studies Across Healthcare, Business, and Technology*. IGI Global, 2024, pp. 122–139.

[26] M. Kang, S. Lee, J. Baek, K. Kawaguchi, and S. J. Hwang, "Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[27] W. X. Zhao, J. Liu, R. Ren, and J.-R. Wen, "Dense text retrieval based on pretrained language models: A survey," *ACM Transactions on Information Systems*, vol. 42, no. 4, pp. 1–60, 2024.

[28] B. Cui, Y. Li, and Z. Zhang, "Joint structured pruning and dense knowledge distillation for efficient transformer model compression," *Neurocomputing*, vol. 458, pp. 56–69, 2021.

[29] A. Pan, J. S. Chan, A. Zou, N. Li, S. Basart, T. Woodside, H. Zhang, S. Emmons, and D. Hendrycks, "Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark," in *International Conference on Machine Learning*. PMLR, 2023, pp. 26 837–26 867.

[30] H. Zhuang, Z. Qin, S. Han, X. Wang, M. Bendersky, and M. Najork, "Ensemble distillation for bert-based ranking models," in *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, 2021, pp. 131–136.

[31] Y. Yan, P. Zheng, and Y. Wang, "Enhancing large language model capabilities for rumor detection with knowledge-powered prompting," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108259, 2024.

[32] Q. Ai, T. Bai, Z. Cao, Y. Chang, J. Chen, Z. Chen, Z. Cheng, S. Dong, Z. Dou, F. Feng *et al.*, "Information retrieval meets large language models: a strategic report from chinese ir community," *AI Open*, vol. 4, pp. 80–90, 2023.