Enhancing Large Language Models: A Survey on Quantization, Model Compression, and Neural Network Optimization

Abstract

This survey examines advanced techniques in artificial intelligence and machine learning aimed at enhancing the efficiency of Large Language Models (LLMs). Focusing on quantization, model compression, and neural network optimization, the paper explores strategies to reduce computational and memory demands while maintaining or improving model accuracy. Quantization methods, including vector quantization and mixed precision, are highlighted for their role in minimizing memory usage and accelerating inference times. Model compression techniques such as pruning and knowledge distillation are discussed for their ability to streamline model architectures without significant accuracy loss, facilitating deployment in resource-constrained environments. The integration of hybrid compression strategies, combining pruning, quantization, and knowledge distillation, demonstrates potential for superior efficiency and performance. Neural network optimization, through parameter-efficient tuning and architecture search, further enhances LLMs' adaptability and efficiency. The survey underscores the importance of developing comprehensive evaluation frameworks to address challenges such as benchmark leakage and data contamination. By leveraging these advanced methodologies, LLMs can achieve a balance between reduced computational complexity and maintained accuracy, broadening their applicability across diverse domains. The ongoing refinement of these techniques is crucial for advancing LLM efficiency, ensuring their continued relevance in the rapidly evolving field of AI.

1 Introduction

1.1 Significance of Large Language Models (LLMs)

Large Language Models (LLMs) have emerged as pivotal tools in artificial intelligence, significantly advancing natural language processing (NLP) and generative AI across various applications [1]. Their capacity to utilize pre-trained knowledge enables them to execute diverse tasks based on natural language prompts, thereby overcoming traditional AI limitations and enhancing adaptability [2].

LLMs are particularly effective in multilingual contexts, addressing challenges in training models for low-resource languages, which is vital during crises requiring machine translation. They also enhance reasoning capabilities in AI, providing frameworks for complex cognitive tasks. The integration of LLMs into multimodal applications allows for the processing of textual, visual, and auditory data, as evidenced by innovations like memory-augmented models for long-term video understanding, end-to-end speech summarization systems, and training paradigms that align LLMs with brain-like visual encoding processes [3, 4, 5, 6].

Moreover, LLMs play a crucial role in decision-making for large-scale multi-objective optimization by automating and enhancing these processes through Generative AI and Evolutionary Algorithms. In the medical field, the application of LLMs for tasks such as medical diagnosis necessitates

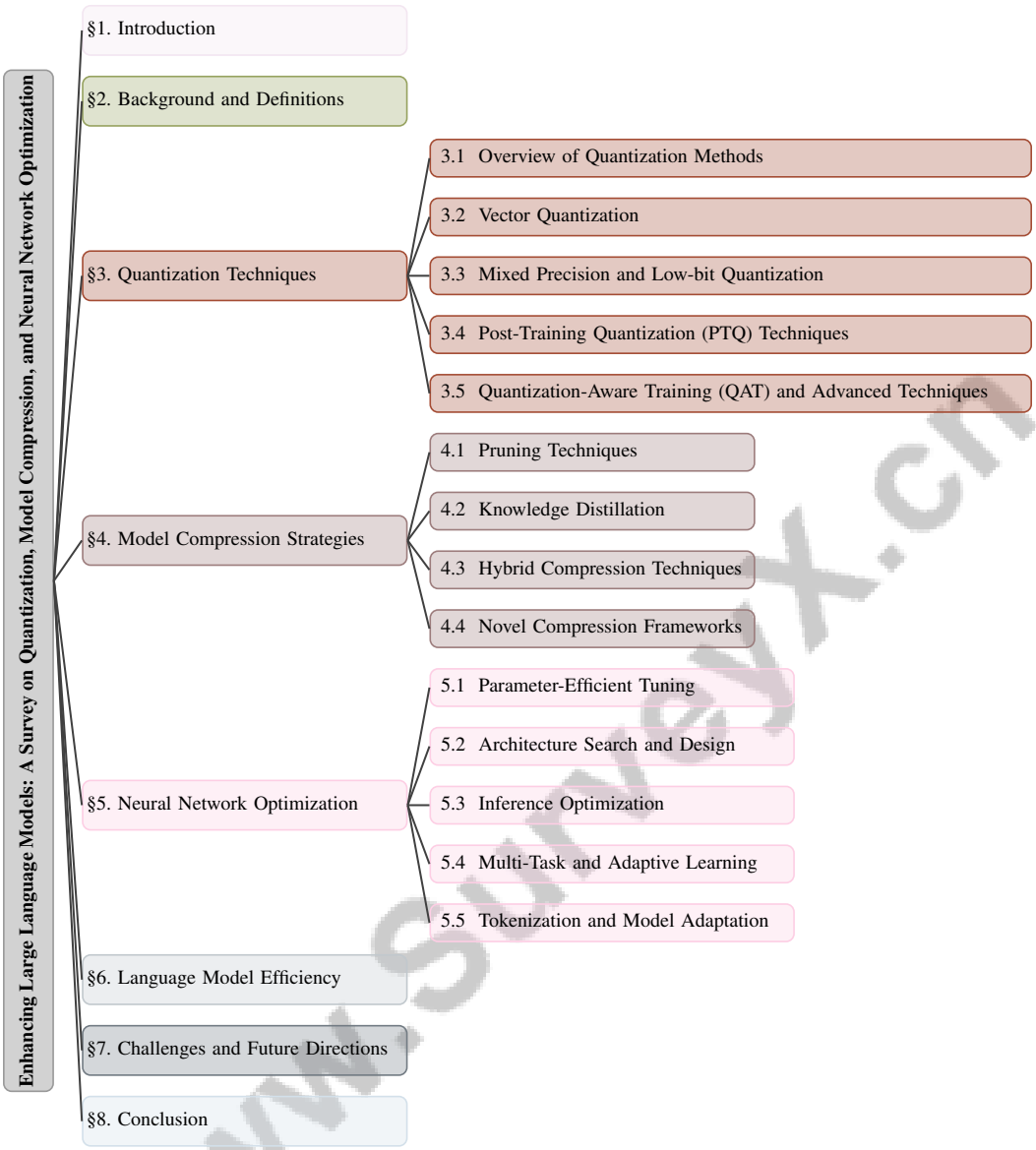


Figure 1: chapter structure

standardized evaluation criteria to accurately assess their capabilities and ensure reliability. The lack of a comprehensive evaluation framework complicates the assessment of risks associated with medical LLMs, as existing methods often depend on labor-intensive human evaluations. Recent advancements, including the LLM-specific Mini-CEX evaluation criterion and automated tools like patient simulators, aim to improve the accuracy and trustworthiness of LLM applications in clinical settings [7, 8, 9].

Despite the prevalence of proprietary models, the open-source community is making strides in transparency and accessibility, although these models often struggle with reasoning and coding tasks compared to closed-source alternatives. In specialized sectors, LLMs enhance operational efficiency and decision-making by transforming expert knowledge into quantifiable features, improving predictive analytics, and optimizing agricultural machinery management. Their integration of human insights with advanced machine learning techniques preserves critical domain expertise while scaling its impact across applications, leading to improved risk assessment accuracy and contextually relevant outputs. Additionally, LLMs facilitate document understanding through innovative distillation methods, advancing knowledge-driven analytics in fields reliant on expert insight [10, 11, 9, 12].

AI-generated, for reference only.

LLMs are at the forefront of machine learning innovation, addressing complex challenges across various domains. They enhance predictive analytics by converting expert insights into quantifiable features, thus improving decision-making accuracy. In agriculture, LLMs are utilized in intelligent machinery management, showcasing significant advancements in operational efficiency through advanced prompt engineering techniques. Furthermore, methodologies like AHAM leverage LLMs for literature mining, aiding researchers in navigating the expanding body of scientific literature, thus underscoring LLMs' essential role in shaping the future of artificial intelligence [12, 11, 9, 13].

1.2 Challenges of LLMs

Large Language Models (LLMs) encounter significant computational and memory challenges that impede their efficient deployment, particularly on resource-constrained devices like smartphones [1]. The unpredictable nature of resource demands in LLM services, where context lengths can vary, exacerbates inefficiencies in resource management [14]. The extensive computational resources and energy consumption required for real-time deployment of deep neural networks (DNNs) are core issues, particularly in embedded environments [15].

Training LLMs with long sequences further complicates these challenges, necessitating vast memory resources and often resulting in out-of-memory errors [16]. High communication overhead in existing parallelism strategies hampers training speed and model performance [17]. Additionally, significant latency and computational overhead during the pre-filling stage of long-context LLMs can be substantial, taking up to 30 minutes for 1 million tokens on a single A100 GPU due to inefficient attention computations [18].

The rapid complexity growth of LLMs has led to a proportional increase in computational costs, necessitating efficient processing and compression strategies [19]. However, the trade-off between reducing computational costs through quantization and maintaining acceptable accuracy levels remains a significant challenge, as quantization can introduce noise that degrades performance. The inefficiency of current quantization methods, particularly those relying on hand-crafted parameters, leads to performance degradation in low-bit quantization scenarios.

Moreover, efficiently compressing large language models while maintaining performance is a notable challenge, with structured pruning methods offering potential solutions by reducing model size without substantial accuracy loss [20]. LLMs face inherent limitations in storing and retrieving factual information, resulting in hallucination and inconsistent reasoning in generated outputs [21]. The issue of benchmark leakage, where training data overlaps with evaluation data, results in misleadingly high performance assessments of LLMs, complicating their evaluation [22]. These challenges highlight the urgent need for innovative solutions to enhance the applicability and efficiency of LLMs in modern AI applications.

1.3 Introduction to Solutions

Quantization, model compression, and neural network optimization are key strategies to address the computational and memory challenges associated with Large Language Models (LLMs). Quantization, particularly low-bit quantization, reduces the precision of computations, thereby decreasing the complexity and resource demands of deep neural networks [19]. This approach facilitates the deployment of LLMs on resource-constrained devices by optimizing performance metrics such as token throughput and latency [14]. The MInference method enhances the pre-filling stage of long-context LLMs by dynamically building sparse attention indices, thereby improving efficiency in processing extensive sequences [18].

Model compression techniques, including token compression retrieval augmented models (TCRA-LLM), are essential for reducing the token size of LLM inputs through summarization and semantic compression, preserving model accuracy while enhancing efficiency. Knowledge distillation and parameter pruning refine LLMs by eliminating redundant parameters, resulting in more compact models that maintain performance [22]. Hybrid compression schemes, which adjust compression intensity based on the nature of communicated messages during training, have been proposed to improve training efficiency, especially in large-scale applications [14].

Neural network optimization encompasses strategies such as parameter-efficient tuning and architecture search, significantly enhancing LLM efficiency. Techniques like DistAttention, introduced by

Infinite-LLM, enable distributed attention computation and efficient management of KVCache across multiple instances, optimizing resource use during inference [14]. Additionally, leveraging weak LLMs, which are less resource-intensive, can provide automated feedback, representing a middle ground between human and AI feedback, thus optimizing resource utilization.

These innovative solutions collectively enhance the operational efficiency of Large Language Models (LLMs) by integrating advanced methodologies such as domain-specific adaptation, distillation techniques, and sophisticated query systems, thereby expanding their applicability across various sectors, including scientific research, document understanding, and agricultural management [12, 10, 13, 23, 9]. By adopting these advanced strategies, LLMs can achieve superior performance and efficiency, reinforcing their pivotal role in the advancement of artificial intelligence and machine learning.

1.4 Structure of the Survey

This survey is organized into several sections, each addressing key aspects of enhancing the efficiency of Large Language Models (LLMs) through advanced techniques. Section 2 provides foundational definitions and explanations of core concepts such as LLMs, quantization, vector quantization, model compression, and neural network optimization, elucidating their roles in improving language model efficiency. Section 3 delves into quantization techniques, exploring methods including vector quantization, mixed precision quantization, post-training quantization (PTQ), and quantization-aware training (QAT), discussing their benefits and limitations. In Section 4, model compression strategies are examined, highlighting techniques such as pruning, knowledge distillation, and hybrid compression approaches, and their impact on model performance and efficiency. Section 5 focuses on neural network optimization, discussing strategies like parameter-efficient tuning, architecture search, and inference optimization, as well as multi-task and adaptive learning. Section 6 analyzes how these techniques collectively contribute to language model efficiency, presenting examples of successful implementations. Section 7 identifies current challenges and potential future research directions in LLM efficiency, emphasizing the need for comprehensive evaluation frameworks. Finally, Section 8 concludes the survey by summarizing key points and reinforcing the importance of these strategies in advancing LLM efficiency.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Background and Definitions

Large Language Models (LLMs) mark a pivotal advancement in AI, characterized by their extensive parameter counts that facilitate diverse natural language processing tasks such as text generation, classification, and translation [2]. Their deployment, particularly on resource-constrained mobile platforms, necessitates optimization techniques to mitigate substantial computational and memory demands.

Quantization is a key optimization strategy for LLMs, reducing model weights and activation precision to lower memory usage and computational needs. Its effectiveness varies with model size; smaller LLMs are more sensitive to quantization, potentially impairing complex tasks like retrieval-augmented generation (RAG) that require long-context reasoning. Studies reveal quantization can cut resource consumption but may compromise model accuracy. Techniques like Low-Rank Compensation (LoRC) help restore quality during quantization, emphasizing strategic implementation to balance efficiency and capability [4, 24]. Vector quantization, which clusters similar data points, enhances storage efficiency and accelerates inference, crucial for optimizing LLM deployment across platforms.

Model compression methods, such as pruning and knowledge distillation, aim to reduce model size without sacrificing performance. Pruning removes redundant parameters, while knowledge distillation transfers knowledge from larger to smaller models, maintaining accuracy with a reduced computational footprint. These strategies are vital for deploying LLMs in resource-limited environments, enhancing performance and efficiency in applications like crisis communication and scientific literature mining [9, 13, 25].

Neural network optimization further enhances LLM efficiency through parameter-efficient tuning and architecture search, refining model architecture and training processes. These approaches involve domain-specific adaptations and advanced methodologies, such as the AHAM framework and

distillation techniques, to optimize metrics like speed and accuracy, particularly in unstructured data tasks and document understanding [11, 10, 23, 22, 9]. By systematically adjusting parameters and exploring innovative designs, these methods develop more efficient LLMs, broadening applicability across domains.

Integrating these optimization techniques is crucial for advancing AI and meeting the demand for efficient, high-performing language models. Hybrid optimization strategies, exemplified by frameworks like LLaMEA-HPO, enhance LLM-driven algorithm generation. Comprehensive evaluation tools, including datasets annotated with human preferences, are essential for assessing and enhancing LLM performance across tasks [2]. These advancements highlight the evolution and application of LLMs across fields, reinforcing their role as universal embedders capable of managing tasks from semantic textual similarity to code search and classification. Continued development and integration of these techniques are vital to overcoming the computational and memory challenges of LLMs, ensuring their efficiency and broad applicability in modern AI applications.

3 Quantization Techniques

Category	Feature	Method
Overview of Quantization Methods	Memory Optimization	IE[16]
Vector Quantization	Discrete Representation	VQ-NN[26], AHAM[9]
Mixed Precision and Low-bit Quantization	Holistic Optimization Efficiency-Driven Precision Tailored Quantization Strategies	MQ[27] W4A4[28], LPQ[29], LP-LQ[30] PEG[31], BAC[32]
Post-Training Quantization (PTQ) Techniques	Efficiency and Size Reduction Learnable and Adaptive Techniques Error and Performance Optimization Channel and Layer Optimization	QTIP[33], LLMS[20] QN[34], OmniQuant[35] LQ-Nets[36] SQ[37]
Quantization-Aware Training (QAT) and Advanced Techniques	Data-Free and Sparse Techniques Quantization Optimization Strategies Diversity and Adaptability	MI[18], ZQ[38], NNQ[39] P4Q[40], TSQ[41] MoPs[42]

Table 1: This table provides a comprehensive summary of various quantization methods used in neural network optimization, particularly for Large Language Models (LLMs). It categorizes the methods into key areas such as vector quantization, mixed precision and low-bit quantization, post-training quantization techniques, and quantization-aware training, highlighting their features and associated research. The table serves as a valuable resource for understanding the advancements and applications of these techniques in enhancing model efficiency and deployment in resource-constrained environments.

In the realm of neural network optimization, quantization techniques play a pivotal role in enhancing the performance and efficiency of models, particularly Large Language Models (LLMs). By reducing the precision of model parameters, these techniques not only minimize memory usage but also decrease computational demands, which is essential for deploying advanced AI models in resource-constrained environments. As illustrated in **??**, the hierarchical structure and categorization of quantization techniques highlight key methods, applications, and advancements that contribute to the efficiency and deployment of LLMs. Table 1 offers an in-depth overview of the principal quantization methods employed to optimize neural networks, focusing on their application in improving the efficiency of Large Language Models (LLMs). Additionally, Table 4 offers a detailed comparison of different quantization methods employed to enhance the efficiency and performance of neural networks, with a focus on their application in large language models (LLMs). The subsequent subsection will delve into the various quantization methods employed in this context, providing a comprehensive overview of their mechanisms and applications.

3.1 Overview of Quantization Methods

Quantization techniques are instrumental in optimizing the efficiency of neural networks, particularly Large Language Models (LLMs), by reducing the precision of model parameters, thereby decreasing memory usage and computational demands. This process typically involves approximating model weights to lower precision formats, such as converting from 32-bit floating-point to 8-bit integer representations, significantly reducing model size and accelerating inference times. The importance of quantization is underscored by its ability to facilitate the deployment of advanced AI models in environments with constrained resources, where computational efficiency is paramount [43].

As illustrated in Figure 2, the hierarchical categorization of quantization methods highlights key areas such as efficiency optimization, advanced strategies, and deployment enhancements in neural networks. Advanced quantization strategies have been developed to optimize model performance across various tasks. For instance, the integration of external memory in neural networks enhances the retrieval capabilities of LLMs during inference, particularly for long-context tasks, thereby improving efficiency [16]. Additionally, frameworks like AHAM employ a combination of sentence transformers and generative models to adapt topic modeling techniques to specific scientific domains, highlighting the versatility of quantization in heterogeneous applications [44].

Quantization is further enhanced through techniques such as prompt engineering and few-shot learning, which have been shown to significantly improve the expressive capabilities of LLMs by optimizing the use of lower precision formats. The implementation of cost-based scheduling frameworks that intelligently distribute Large Language Model (LLM) tasks across diverse hardware platforms—considering both energy efficiency and performance demands—demonstrates the effective integration of quantization techniques in practical applications. This approach not only optimizes energy consumption but also enhances the operational efficiency of data centers, as evidenced by a hybrid model that achieved a 7.5% reduction in energy use compared to traditional methods by strategically selecting between energy-efficient processors and high-performance GPUs based on workload characteristics [45, 46].

These advancements highlight the essential role of quantization techniques in facilitating the practical deployment of sophisticated AI models, particularly in resource-constrained environments where optimizing memory usage and computational efficiency is critical. By addressing unique challenges such as high dynamic activation ranges and the need for effective long-context reasoning, quantization strategies—like per-embedding-group quantization and adaptive bit-width optimization—enable the successful application of transformer architectures and smaller large language models in real-world scenarios, ensuring minimal accuracy loss while achieving significant reductions in resource consumption [31, 4, 47, 48]. By leveraging lower precision formats, quantization not only reduces the computational burden but also enhances the generalization capabilities of LLMs across various tasks, often surpassing existing benchmarks.

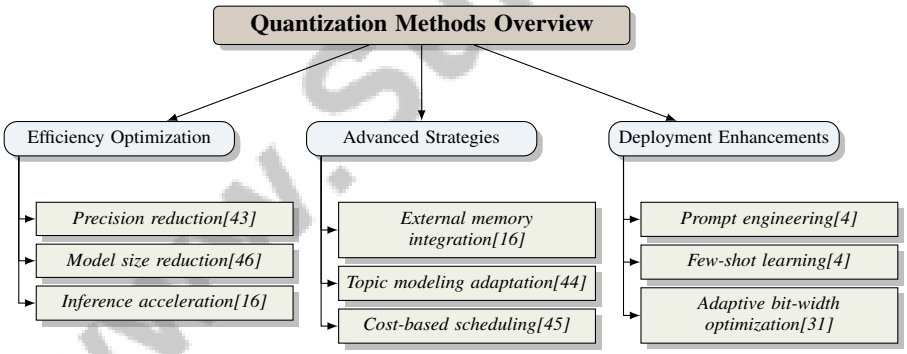


Figure 2: This figure illustrates the hierarchical categorization of quantization methods, highlighting efficiency optimization, advanced strategies, and deployment enhancements in neural networks.

3.2 Vector Quantization

Method Name	Data Discretization	Efficiency Optimization	Adaptability
VQVAE[49]	Discrete Latent Space	Improved Tts Performance	Different Domains Applications
VQ-NN[26]	Weighted K-means	Memory Savings	Generalizes Well
AHAM[9]	Corpus Vectorization	-	Domain Adaptation Techniques

Table 2: Comparison of Vector Quantization Methods in Terms of Data Discretization, Efficiency Optimization, and Adaptability. This table outlines the distinct approaches of VQVAE, VQ-NN, and AHAM methods, highlighting their unique strategies in data discretization, efficiency optimization, and adaptability across different domains.

Vector quantization is a pivotal technique in the optimization of neural networks, particularly in enhancing the efficiency of Large Language Models (LLMs). This method involves the discretization

of continuous data into a finite set of vectors, significantly reducing the storage requirements and computational complexity of models. The primary innovation of vector quantization lies in its adoption of a discrete latent space, which contrasts with traditional continuous methods, thereby enhancing performance in various applications such as voice cloning tasks [49]. Table 2 provides a comprehensive comparison of various vector quantization methodologies, detailing their approaches to data discretization, efficiency optimization, and adaptability in enhancing the performance of Large Language Models.

The application of vector quantization in LLMs is particularly advantageous due to its ability to minimize reconstruction errors, not just for model weights but also for network outputs. By leveraging weighted k-means clustering, vector quantization achieves better accuracy and efficiency, which is crucial for maintaining the performance of LLMs while reducing their resource demands [26]. This approach is essential for deploying LLMs in environments with limited computational resources, ensuring that they remain effective and efficient.

Moreover, the effectiveness of vector quantization is further demonstrated in methods like AHAM, which leverage LLMs for semantic understanding and adapt topic modeling techniques to specific domains [9]. This adaptability is critical for enhancing the inference efficiency of LLMs, allowing them to perform optimally across diverse applications and domains [50].

Despite its advantages, vector quantization presents challenges, particularly in balancing the trade-off between model compression and accuracy. The discrete nature of vector quantization can introduce quantization noise, which may affect the precision of LLM outputs. Therefore, ongoing research focuses on refining these techniques to minimize such drawbacks while maximizing the benefits of reduced model size and improved computational efficiency [51].

3.3 Mixed Precision and Low-bit Quantization

Mixed precision and low-bit quantization techniques are integral to enhancing the computational efficiency of Large Language Models (LLMs) by reducing the precision of arithmetic operations, thus decreasing memory usage and computational overhead. Mixed precision quantization involves employing varying precision levels across different network components, optimizing performance without significantly compromising model accuracy. This approach is exemplified by MobileQuant, a post-training quantization method that simultaneously optimizes weight transformation and activation range parameters, facilitating efficient deployment of LLMs on mobile devices [27]. Furthermore, the introduction of per-embedding-group quantization and quantization-aware training offers additional strategies to address the challenges associated with mixed precision quantization [31].

Low-bit quantization, such as W4A4 quantization, reduces both weights and activations to 4 bits, leveraging hardware capabilities to enhance inference speed and reduce memory usage [28]. Techniques like LPQ employ linear approximation to minimize the mean squared error of quantized tensors, effectively maintaining model accuracy while reducing bit-width [29]. The LP-LQ method further refines this approach by optimizing low-bit precision through Minimum Mean Squared Error (MMSE) optimization, ensuring minimal accuracy loss [30].

Innovations in this domain include Bound Aware Clipping (BAC), which customizes quantization schemes based on local network properties to improve performance, particularly in recognition tasks [32]. Additionally, the Two-Step Quantization (TSQ) framework separates code learning and transformation function learning, facilitating more effective training of low-bit neural networks [41]. These methods collectively enhance the efficiency and applicability of LLMs, making them suitable for deployment in resource-constrained environments. By leveraging these advanced quantization strategies, LLMs can achieve a balance between reduced computational complexity and maintained accuracy, broadening their applicability across various domains.

3.4 Post-Training Quantization (PTQ) Techniques

Post-training quantization (PTQ) techniques are pivotal in optimizing large language models (LLMs) for deployment by significantly reducing model size and computational demands without the need for extensive retraining. These techniques convert models from high-precision to lower bit-width representations, thereby conserving memory while maintaining performance. SmoothQuant exemplifies such methods by facilitating accurate 8-bit quantization of weights and activations, utilizing

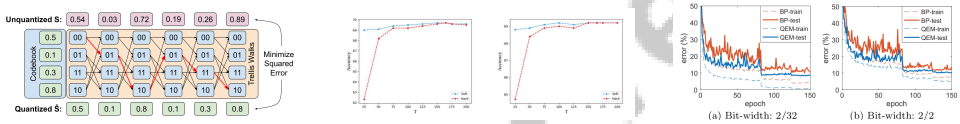
per-channel scaling transformations to manage quantization difficulty effectively [37]. This approach ensures minimal impact on model accuracy while enhancing deployment efficiency.

ZeroQuant offers a comprehensive PTQ framework aimed at compressing large Transformer models to maintain accuracy and efficiency [38]. OmniQuant further refines PTQ by optimizing quantization parameters for both weights and activations, underscoring the potential of PTQ methods to achieve significant memory savings without degrading model accuracy [35]. These techniques highlight the capability of PTQ to balance memory efficiency with performance retention.

Advanced PTQ methods also integrate architectural modifications to enhance model efficiency. The LLM Surgeon framework, for example, employs advanced curvature approximations to guide weight pruning, effectively minimizing performance loss [20]. This synergy between pruning and quantization exemplifies a refined approach to model size reduction and efficiency.

Furthermore, techniques like BOA use attention-aware Hessian matrices to address inter-layer dependencies, optimizing quantization processes to minimize performance degradation [19]. The integration of these advanced techniques within PTQ frameworks exemplifies a comprehensive strategy for model compression and efficiency enhancement.

The benchmark provided by ZeroQuantV2 aims to evaluate and compare different PTQ methods, optimizing model size reduction while minimizing accuracy loss in LLMs [24]. This benchmark underscores the critical role of PTQ techniques in enhancing the deployment of LLMs in resource-constrained environments, ensuring their effectiveness across diverse applications. By leveraging these sophisticated quantization strategies, LLMs can achieve a balance between reduced computational complexity and maintained accuracy, broadening their applicability across various domains.



(a) Trellis Walks for Quantized and Unquantized State Representations[33]

(b) Effect of Temperature on the Accuracy of a Machine Learning Model[34]

(c) Comparison of BP and QEM training and testing errors for different bit-widths[36]

Figure 3: Examples of Post-Training Quantization (PTQ) Techniques

As shown in Figure 3, In the realm of machine learning and neural network optimization, quantization techniques play a pivotal role in enhancing computational efficiency and reducing model size without significantly compromising accuracy. Among these techniques, Post-Training Quantization (PTQ) stands out as a practical approach, allowing models to be quantized after the initial training phase. The provided figures illustrate various aspects of PTQ techniques, offering insights into their application and impact. The first image, "Trellis Walks for Quantized and Unquantized State Representations," visually represents the transitions between states in a Markov process, highlighting the differences in state representation before and after quantization. The second image explores the "Effect of Temperature on the Accuracy of a Machine Learning Model," demonstrating how varying temperatures can influence model accuracy under different quantization settings, labeled as 'Soft' and 'Hard.' Lastly, the third image provides a comparative analysis of training and testing errors between two training methods, BP and QEM, across different bit-widths, underscoring the trade-offs in precision and computational demand inherent in quantization. Together, these examples encapsulate the nuanced considerations involved in employing PTQ techniques, emphasizing their significance in optimizing machine learning models for real-world applications. [**?**]tseng2024qtip,yang2019quantization,zhang2018lq)

3.5 Quantization-Aware Training (QAT) and Advanced Techniques

Quantization-Aware Training (QAT) is a crucial methodology in enhancing the efficiency and accuracy of neural networks, particularly Large Language Models (LLMs), by incorporating the effects of quantization during the training phase. This approach enables models to adapt to the quantization-induced noise, ensuring robust performance post-quantization. The Quantization Hypothesis posits that many prediction problems can be decomposed into an enumerable set of quanta that models must learn to minimize loss, providing a theoretical foundation for QAT [52].

AI-generated, for reference only.

Method Name	Methodology Focus	Computational Efficiency	Model Adaptability
ZQ[38]	Post-training Quantization	Reducing Computational Burden	Maintain Performance
NNQ[39]	Quantization-aware Training	Reduce Power Latency	Quantization-induced Noise
MI[18]	Sparse Attention Indices	Reducing Computational Requirements	Maintain Performance
TSQ[41]	Decoupling Quantization	Sparse Quantization	Stable Representations
MoPs[42]	Optimize Prompt Tuning	Without Additional Costs	Improving Performance Adaptability
P4Q[40]	Low-bit Quantization	Reduced Computational Costs	Maintain Performance

Table 3: This table provides a comparative analysis of various quantization techniques used in neural network training, focusing on their methodology, computational efficiency, and model adaptability. The methods highlighted include post-training quantization, quantization-aware training, and low-bit quantization, each contributing to reducing computational costs while maintaining model performance. The table aims to elucidate the strengths of these techniques in optimizing large language models for deployment in resource-constrained environments.

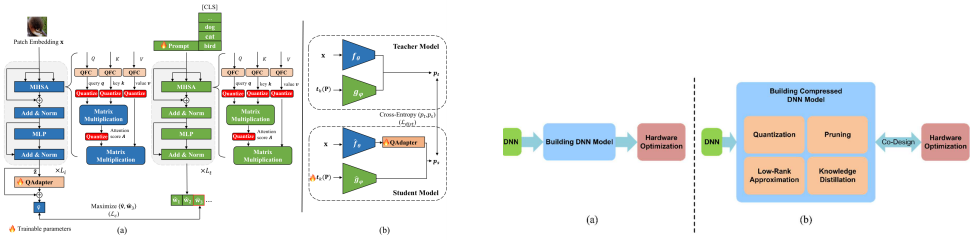
ZeroQuant exemplifies the advancements in QAT by performing layer-wise quantization without requiring access to the original training data, significantly reducing computational burdens [38]. This innovation facilitates the deployment of LLMs in resource-constrained environments by optimizing computational efficiency while maintaining accuracy. Moreover, the introduction of Low-Rank Compensation (LoRC) in ZeroQuantV2 enhances model quality recovery during quantization, representing a significant advancement over previous techniques [24].

Advanced techniques in QAT, such as those proposed by Nagel et al., focus on developing state-of-the-art algorithms to mitigate the negative effects of quantization noise, allowing for low-bit weights and activations [39]. The dynamic sparse attention calculation method introduced in MInference further exemplifies this by identifying optimal sparse patterns for each attention head, significantly reducing computational requirements while maintaining or improving accuracy [18].

The BOA method employs an attention-aware Hessian-based strategy to iteratively quantize weights, compensating for quantization errors and enhancing model efficiency [19]. Additionally, the two-step process proposed by Wang et al. addresses the coupling issue in quantization by first learning codes and then learning transformation functions based on those codes, optimizing the quantization process [41].

Furthermore, the integration of MoPs allows for effective handling of task and data heterogeneity, improving performance across varying scenarios without additional computational costs [42]. These advanced techniques collectively enhance the applicability and efficiency of LLMs, making them more suitable for deployment in diverse and resource-constrained environments. By integrating these sophisticated quantization strategies, LLMs can achieve a balance between reduced computational complexity and maintained accuracy, broadening their applicability across various domains.

Table 3 presents a detailed comparison of advanced quantization techniques, illustrating their methodologies, computational efficiencies, and adaptability in enhancing the performance of large language models.



(a) Quantitative Adaptation for Efficient and Accurate Image Classification[40]

(b) Building Compressed DNN Models[53]

Figure 4: Examples of Quantization-Aware Training (QAT) and Advanced Techniques

As shown in Figure 4, In the realm of deep learning, quantization techniques have emerged as pivotal tools for enhancing both the efficiency and effectiveness of neural networks. The concept of Quantization-Aware Training (QAT) and other advanced techniques are exemplified through illustrative flowcharts, as depicted in the accompanying figures. The first example, "Quantitative Adaptation

for Efficient and Accurate Image Classification," outlines a systematic approach to quantizing input image patch embeddings. This process, involving operations like Multi-Head Self-Attention (MHSA), Add Norm, and Multi-Layer Perceptron (MLP), culminates in the generation of a teacher model aimed at improving classification accuracy while maintaining computational efficiency. Meanwhile, the second example, "Building Compressed DNN Models," demonstrates a comprehensive method for constructing compressed deep neural network models. This flowchart highlights the initial development of a DNN model, followed by hardware optimization to tailor the model's performance to specific hardware capabilities. Together, these examples underscore the significance of integrating quantization techniques into the neural network training pipeline to achieve optimal performance and resource utilization. [**?**]sun2024p4qlearningpromptquantization,rokh2023comprehensive)

Feature	Vector Quantization	Mixed Precision and Low-bit Quantization	Post-Training Quantization (PTQ) Techniques
Precision Level	Discrete Vectors	Varying Precision	Lower Bit-width
Deployment Efficiency	Reduced Complexity	Memory Reduction	Memory Savings
Adaptability	Domain-specific	Hardware-optimized	Minimal Accuracy Loss

Table 4: This table provides a comparative analysis of various quantization methods, focusing on their precision levels, deployment efficiency, and adaptability. The methods examined include vector quantization, mixed precision and low-bit quantization, and post-training quantization (PTQ) techniques, highlighting their distinct features and contributions to optimizing neural networks, particularly in the context of large language models (LLMs).

4 Model Compression Strategies

To meet the increasing demands for efficient deployment of Large Language Models (LLMs), exploring various model compression strategies is vital for optimizing performance and reducing computational requirements. Among these strategies, pruning techniques are foundational, systematically eliminating redundant parameters to streamline model architecture while preserving accuracy. This subsection examines the intricacies of pruning techniques and their significant role in enhancing the efficiency and adaptability of LLMs across diverse operational contexts.

4.1 Pruning Techniques

Pruning techniques are essential for optimizing LLMs by systematically removing redundant parameters, which reduces model size and computational load while maintaining performance. These methods are particularly crucial for deploying LLMs in resource-constrained environments, enhancing system performance and efficiency without compromising core decoding mechanisms. This is especially important for applications like crisis communication for low-resource languages, where rapid and accurate machine translation is needed, and in document analytics, where efficient management of unstructured data can lead to substantial cost savings and improved query accuracy [23, 9, 13, 25].

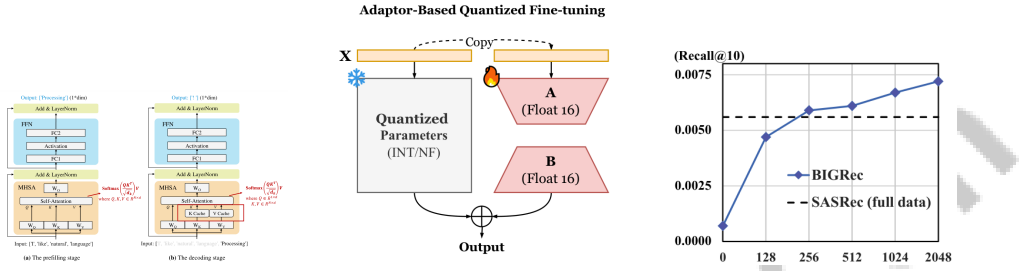
Structured pruning methodologies, such as LLM-Pruner, selectively eliminate non-critical structures within LLMs. By utilizing gradient information, LLM-Pruner achieves effective compression with minimal accuracy loss, ensuring that only the most vital components of the model are retained [54]. This approach is instrumental in maintaining model performance while reducing computational and memory requirements.

Integrating pruning with other optimization techniques further enhances model compactness and efficiency. For instance, the combination of parameter pruning with parameter quantization, as proposed by Choukroun et al., demonstrates significant reductions in model size and computational demands while maintaining accuracy, making deployment on resource-constrained devices feasible [30].

Innovative strategies like LQ-LoRA decompose each pretrained matrix into a high-precision low-rank component and a memory-efficient quantized component, focusing on updating only the low-rank component during finetuning to ensure efficiency and adaptability without sacrificing accuracy [55].

Advanced gating functions, as suggested by Dun et al., enhance the adaptability and efficiency of pruning strategies. These functions can be pre-trained using unlabeled instruction data with domain or task labels, improving their initial performance and adaptability to specific deployment scenarios [42].

Practical applications of these advanced pruning techniques have shown significant reductions in model size and computational needs, facilitating effective deployment of LLMs in various environments without substantial performance loss. By implementing these techniques, LLMs can reduce computational demands while maintaining accuracy, thereby enhancing their versatility in fields such as literature mining, recommendation systems, and predictive analytics. Notable approaches like the AHAM methodology for scientific text analysis and the LLM-Pruner for structural compression demonstrate how LLMs can retain multi-task capabilities with reduced resource requirements [11, 56, 54, 13, 9].



(a) Transformer-based Model for Language Understanding[50]

(b) Adaptor-Based Quantized Fine-tuning[57]

(c) Recall@10: Comparison of BIGRec and SASRec (full data)[56]

Figure 5: Examples of Pruning Techniques

As illustrated in Figure 5, pruning techniques are pivotal in optimizing machine learning models without significantly compromising performance. The figures depict distinct approaches within this domain: the first image showcases a transformer-based model for language understanding, emphasizing its intricate architecture; the second image presents Adaptor-Based Quantized Fine-tuning, refining model performance by adjusting quantized parameters; and the third image compares the Recall@10 performance between two recommendation algorithms, BIGRec and SASRec, highlighting their effectiveness across varying dataset sizes. Together, these examples underscore the diverse methodologies employed in pruning techniques to achieve model compression, facilitating more efficient and scalable machine learning solutions [50, 57, 56].

4.2 Knowledge Distillation

Knowledge distillation is a crucial model compression technique that transfers knowledge from a large, complex "teacher" model to a more compact and efficient "student" model. This process is particularly significant for LLMs, characterized by high parameter counts and substantial computational requirements, presenting challenges for effective deployment in resource-constrained environments. Researchers are actively developing optimization techniques across data, model, and system levels to enhance LLM inference efficiency, addressing issues like large model sizes and complex attention mechanisms [50, 58, 9]. By distilling knowledge from a teacher model, typically pre-trained on vast datasets, it is possible to achieve comparable performance in tasks such as paraphrase generation while significantly reducing size and computational requirements.

The D2LLM framework exemplifies the potential of knowledge distillation in enhancing LLM efficiency and performance. By decomposing complex tasks into simpler sub-tasks, the student model learns with improved accuracy and reduced resource consumption, broadening the applicability of LLMs across various reasoning tasks, particularly in document understanding, literature mining, and predictive analytics. This methodology integrates expert domain knowledge into quantifiable features and utilizes reinforcement learning for fine-tuning, addressing computational challenges to facilitate deployment of sophisticated language models in real-world applications [11, 9, 10, 59]. The incorporation of labeling and curriculum learning strategies within the distillation process further enhances knowledge transfer effectiveness, contributing to the development of high-performance, resource-efficient models.

The practical applications of knowledge distillation are extensive, especially in resource-constrained scenarios like mobile or edge deployments. By ensuring distilled models maintain high-quality outputs, this technique offers a cost-effective solution for deploying LLMs in environments with

limited computational resources. Future research should explore the synergistic integration of post-training quantization (PTQ) with other lightweight compression techniques and fine-grained quantization schemes to enhance efficiency and performance, particularly where memory constraints impact inference throughput. Recent advancements, such as trellis coded quantization (TCQ) and low-rank compensation methods, highlight the potential for improved quantization quality and model accuracy, underscoring the need for comprehensive studies assessing various quantization strategies across different model architectures and bit precisions [60, 24, 33]. By refining knowledge transfer processes, these techniques ensure distilled models can perform comparably to larger counterparts while offering significant advantages in computational efficiency and cost-effectiveness.

4.3 Hybrid Compression Techniques

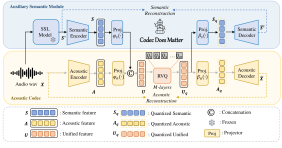
Hybrid compression techniques represent an advanced approach to optimizing LLMs by integrating multiple compression strategies to achieve superior efficiency and performance. These techniques leverage advanced methods such as pruning, quantization, and knowledge distillation, effectively harnessing their unique advantages to significantly decrease model size and computational requirements. This approach not only preserves but can also enhance model accuracy, facilitating the deployment of sophisticated language models in resource-constrained environments, as shown in recent studies on document understanding and deep neural network optimization [15, 4, 48, 10].

The integration of pruning and quantization, exemplified by the LQ-LoRA framework, demonstrates the potential of hybrid approaches. LQ-LoRA decomposes pretrained matrices into a high-precision low-rank component and a memory-efficient quantized component, updating only the low-rank component during finetuning to retain high accuracy while significantly reducing memory usage and computational costs [55].

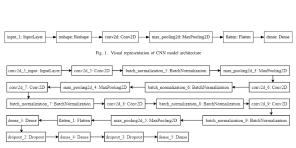
Additionally, combining knowledge distillation with quantization-aware training (QAT) offers another effective hybrid strategy. By distilling knowledge from a larger teacher model into a smaller student model while implementing Efficient Quantization-Aware Training (EfficientQAT), it is possible to significantly enhance the student model's performance and efficiency. EfficientQAT optimizes the quantization process through block-wise training and end-to-end training of quantization parameters, enabling the smaller model to maintain high accuracy with reduced memory requirements, thus making advanced language models more accessible for practical applications [61, 10, 62]. This dual approach ensures effective operation of the student model in resource-constrained environments while maintaining high accuracy.

Hybrid compression techniques also benefit from the synergy between structured pruning and dynamic sparse attention calculations, as seen in methods like MInference. By dynamically optimizing attention patterns and integrating pruning strategies, these techniques reduce computational overhead while enhancing model inference speed and accuracy [18].

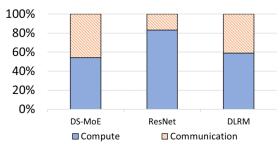
Developing hybrid compression frameworks is crucial for addressing the growing demand for efficient deployment of LLMs across various applications. These frameworks can enhance performance and reduce resource consumption, enabling scalable and sustainable use in real-world environments [23, 10, 63, 13]. By combining multiple compression methods, these frameworks achieve a balance between reduced computational complexity and maintained accuracy, broadening LLM applicability across domains. Ongoing refinement and integration of hybrid compression techniques will play a pivotal role in advancing LLM efficiency and performance, ensuring their continued relevance in the rapidly evolving field of artificial intelligence.



(a) Audio-visual semantic reconstruction with auxiliary semantic module[64]



(b) CNN Model Architecture Visualization[65]



(c) Comparison of Compute and Communication in Deep Learning Models[17]

Figure 6: Examples of Hybrid Compression Techniques

As depicted in Figure 6, hybrid compression techniques are pivotal in enhancing deep learning model efficiency. The figure presents three distinct examples of these techniques: the first example illustrates audio-visual semantic reconstruction with an auxiliary semantic module, detailing the transformation of audio signals into unified feature representations; the second example visualizes a CNN architecture, emphasizing the sequential layering of operations; and the third example compares resource distribution in deep learning models, revealing the balance between compute and communication resources. Collectively, these examples encapsulate diverse strategies employed in hybrid compression techniques to optimize model performance and resource utilization [64, 65, 17].

4.4 Novel Compression Frameworks

Innovative compression frameworks are essential for overcoming challenges associated with deploying LLMs by introducing advanced methodologies that significantly improve model efficiency and performance. A pivotal advancement in this domain is the LLM-Pruner, which employs a task-agnostic approach to rapidly compress models while maintaining performance, eliminating the need for task-specific fine-tuning [54]. This approach is particularly advantageous for rapid deployment across diverse tasks.

The theoretical framework proposed by Michaud et al. offers valuable insights into neural scaling, providing better predictions of model performance as they scale [52]. This framework is instrumental in guiding the development of scalable compression techniques that maintain model performance across various sizes and complexities.

The Two-Step Quantization (TSQ) framework, introduced by Wang et al., significantly improves accuracy and stability during training by decoupling weight and activation quantization [41]. This method also incorporates sparse quantization, further enhancing model compression by reducing redundancy in model parameters, thereby optimizing computational efficiency.

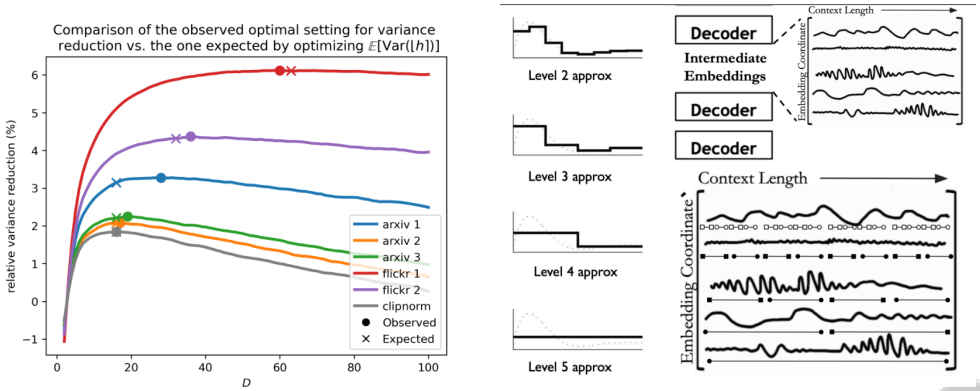
The ALC3 method achieves oracle performance with human feedback on significantly fewer examples than the noisy examples in the dataset, exemplifying the potential of active learning strategies in model optimization [66]. This approach highlights the importance of integrating human feedback in the compression process to enhance model performance and efficiency.

These innovative frameworks signify substantial progress in model compression, addressing critical challenges and enhancing the deployment of LLMs across diverse applications. For instance, the AHAM methodology improves scientific literature mining by adapting the BERTopic framework with the help of domain experts, while advancements in weight quantization, such as the CVXQ method, enable efficient model deployment on resource-constrained devices. Furthermore, a comprehensive survey of LLM inference serving highlights system-level enhancements that optimize performance and efficiency, broadening the practical applicability of LLMs in real-world settings [67, 13, 9]. By leveraging these advanced methodologies, LLMs can achieve a balance between reduced computational complexity and maintained accuracy, enhancing their deployment in diverse applications.

As illustrated in Figure 7, novel compression frameworks have emerged as pivotal tools for enhancing neural network efficiency. The figure presents two distinct examples: the first subfigure compares variance reduction strategies within hidden layers, showcasing observed optimal settings against expectations derived from optimizing expected variance, thus providing insights into structural adjustments for optimal performance. The second subfigure explores the intricacies of decoding speech signals, visually representing how context length and embedding coordinates influence the decoding process. This underscores the importance of context in accurately interpreting and compressing audio data. Together, these examples reflect ongoing advancements in creating more efficient and effective neural network architectures [68, 69].

5 Neural Network Optimization

Optimizing neural networks is vital for improving the performance and efficiency of Large Language Models (LLMs), especially in resource-constrained settings. This section examines strategies such as parameter-efficient tuning and architecture design, which aim to enhance model performance while minimizing computational overhead.



(a) The image compares the observed optimal setting for variance reduction to the one expected by optimizing the expected variance of the hidden layer.[68]

(b) Decoding the Context of Speech Signals: A Visual Explanation[69]

Figure 7: Examples of Novel Compression Frameworks

5.1 Parameter-Efficient Tuning

Parameter-efficient tuning is crucial for enhancing LLMs' performance with minimal computational costs. The D2LLM framework, for instance, employs a bi-encoder structure and an Interaction Emulation Module to efficiently generate embeddings, highlighting the potential of innovative designs in boosting model efficiency [70]. Techniques like ECO allow targeted unlearning without extensive retraining, maintaining original weights while adapting input embeddings, thus enabling scalable adaptation across various LLM sizes [71].

LQ-LoRA combines low-rank updates with quantization to reduce memory requirements while preserving performance, facilitating deployment in resource-constrained environments [55]. Additionally, energy-efficient task assignment methods optimize resource usage by allocating tasks to suitable hardware based on energy efficiency [45]. The LLM-FEF method demonstrates LLMs' capability in qualitative data analysis, enhancing predictive features without extensive retraining [11].

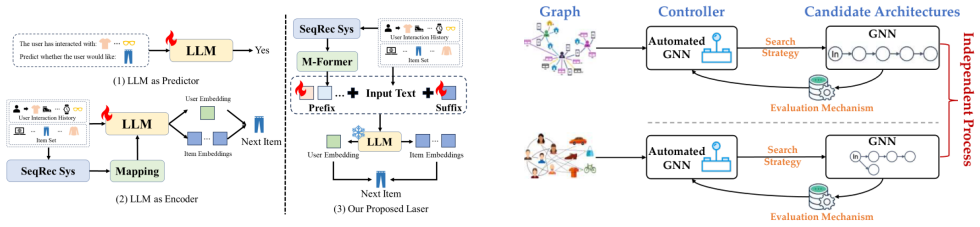
Parameter-efficient techniques, including few-shot fine-tuning and data pruning, enable rapid adaptation to new data, addressing large-scale data processing challenges. Domain-specific adaptations, such as AHAM, improve text analysis, while comprehensive surveys on efficient inference explore strategies to mitigate LLMs' computational demands, ensuring their applicability across various domains [50, 56, 9].

5.2 Architecture Search and Design

Innovative architecture search and design techniques are essential for improving LLM efficiency, particularly in resource-limited environments. Neural architecture search (NAS) automates the discovery of optimal model architectures, using reinforcement learning and evolutionary algorithms to explore architecture space and identify efficient models [38]. Modular design principles enhance architecture efficiency by enabling component reuse across models, ensuring consistent performance [24].

Incorporating attention mechanisms, such as sparse and dynamic attention, allows efficient handling of long sequences by focusing on relevant inputs, reducing unnecessary calculations [18]. These methodologies are crucial for improving LLM performance and efficiency, enabling better adaptation to specific tasks and effective evaluation of diverse data types in real-world applications [8, 10, 13, 63, 9].

As shown in Figure 8, architecture search and design are pivotal in enhancing machine learning models' performance and efficiency. The first example examines LLMs' roles as predictors and encoders in sequential recommendation systems, while the second presents an autonomous graph



(a) User Interaction Prediction with a Novel Approach: A Comparative Study of LLM as Predictor and Encoder in SeqRec Systems[72]

(b) Graph-based Autonomous GNN Architecture for Efficient Graph Classification[73]

Figure 8: Examples of Architecture Search and Design

neural network (GNN) architecture, emphasizing systematic optimization for graph-based models [72, 73].

5.3 Inference Optimization

Inference optimization is crucial for improving LLM efficiency by reducing latency and resource consumption. Key strategies include optimizing memory management to leverage LLMs' unique characteristics and key-value (KV) caches, significantly reducing context switching latency [74]. Dynamic sparse attention calculations identify optimal sparse patterns for attention heads, minimizing unnecessary computations and accelerating inference speed, as exemplified by the MInference method [18].

Cost-based scheduling frameworks optimize inference by distributing LLM tasks across heterogeneous hardware based on energy efficiency, reducing energy consumption and enhancing adaptability [1]. Quantization techniques, such as mixed precision and low-bit quantization, decrease arithmetic operation precision, reducing memory usage and computational overhead while maintaining accuracy [27].

Inference optimization strategies are essential for improving LLM performance, enabling faster, resource-efficient responses across applications. Recent advancements, like the LLMS framework for optimizing context switching and memory utilization, further enhance performance in diverse applications [4, 14, 13, 74].

5.4 Multi-Task and Adaptive Learning

Multi-task and adaptive learning enhance LLM adaptability and efficiency by enabling simultaneous learning across multiple tasks. This approach leverages shared representations and parameters, improving learning efficiency and reducing computational costs. Multi-task learning allows LLMs to utilize shared knowledge across tasks, enhancing effectiveness while minimizing reliance on extensive, task-specific training [75, 42, 10, 76, 9].

Adaptive learning dynamically adjusts parameters in response to varying input conditions and task requirements. The LOFIT framework exemplifies this adaptability by localizing modifications to specific attention heads, enhancing performance without extensive parameter updates [77]. Integrating these strategies significantly enhances LLM efficiency, robustness, and flexibility, enabling effective handling of diverse applications. Approaches like AHAM and Mixture of Prompts (MoPs) optimize performance across heterogeneous tasks, improving task accuracy and reducing errors [42, 9].

5.5 Tokenization and Model Adaptation

Tokenization strategies and model adaptation are critical for enhancing LLM efficiency. Tokenization breaks text into manageable units, optimizing input representation and minimizing computational costs. Recent advancements, including hybrid inference approaches and token compression techniques, significantly improve efficiency and response quality while reducing reliance on expensive cloud resources. Token compression can reduce input token size by up to 65

Subword tokenization techniques like Byte-Pair Encoding (BPE) and SentencePiece enhance LLM efficiency in processing diverse vocabularies and languages with varied morphological structures. These methods enable LLMs to perform comparably to existing solutions in tasks like mining process models and extracting insights from scientific literature [58, 13, 9, 10].

Model adaptation involves fine-tuning pre-trained LLMs for specific tasks or domains, enhancing performance and applicability. Transfer learning leverages knowledge from source tasks to improve performance on target tasks. Using pre-trained models like GPT and BERT as starting points for adaptation has become standard practice, allowing for tailored applications with minimal additional training [2].

Integrating sophisticated tokenization strategies and targeted model adaptation techniques is essential for enhancing LLM performance across various applications. These approaches optimize domain-specific adaptations for scientific literature mining, improve inference serving in production environments, and leverage hybrid inference approaches to balance computational costs while maintaining response quality [78, 10, 13, 79, 9].

6 Language Model Efficiency

Enhancing the efficiency of Large Language Models (LLMs) is essential for meeting the demands of specific application domains. This section explores efficiency techniques tailored for specialized applications, demonstrating how targeted strategies optimize LLMs for distinct computational and performance needs. By examining frameworks like DesiGNN and the implications of precision scaling laws, we gain insights into methodologies that enhance model efficiency across various fields.

6.1 Efficiency in Specialized Applications

Efficiency techniques are crucial for deploying LLMs in specialized domains with stringent computational and performance requirements. These techniques address challenges such as large model sizes, quadratic-complexity attention operations, and auto-regressive decoding, which can hinder LLM inference in resource-constrained settings. Recent advancements in data-level, model-level, and system-level optimizations improve inference efficiency, enabling LLMs to serve applications like literature mining and document analytics, where accurate processing of unstructured data is vital [23, 50, 9]. The DesiGNN framework exemplifies these techniques within Graph Neural Networks, reducing computational overhead and accelerating design cycles through efficient GNN architectures.

Precision in model scaling is emphasized by Kumar et al., who demonstrate that tailored precision settings significantly enhance model efficiency without compromising accuracy [80]. This highlights the necessity of precision optimization in specialized applications, where computational resources and performance requirements vary widely.

Implementation of efficiency techniques notably enhances performance in natural language processing, computer vision, and speech recognition. Methodologies like AHAM, employing domain-specific adaptations of topic modeling frameworks, and ZenDB, utilizing semantic structures for improved document analytics, exemplify this trend. Advancements in document understanding via distillation techniques optimize LLMs, while sequence-level knowledge distillation introduces new approaches for efficient paraphrase generation, contributing to the effectiveness of these technologies in complex tasks [81, 50, 10, 23, 9]. By customizing quantization strategies, model compression, and neural network optimization techniques to meet domain-specific requirements, researchers achieve substantial improvements in LLM efficiency, facilitating deployment across a spectrum of specialized applications.

The development and integration of efficiency techniques in specialized applications are essential for advancing LLM capabilities and ensuring their relevance in the rapidly evolving AI landscape. As demand for efficient and adaptable language models grows, incorporating advanced techniques becomes crucial for enhancing LLM performance and applicability in specialized domains. Recent advancements in LLM serving systems, detailed in comprehensive surveys, highlight system-level enhancements that boost performance without altering core decoding mechanisms, enabling real-world application deployment. Methodologies like AHAM illustrate the effectiveness of domain-specific adaptation in literature mining through generative models such as LLaMa2, enhancing topic modeling precision via expert-guided prompts. The synergy between LLMs and optimization

algorithms further paves the way for smarter decision-making in dynamic environments, underscoring the transformative impact of tailored LLM techniques across various fields [79, 9, 13].

6.2 Mobile and Edge Deployment

Deploying efficient LLMs on mobile and edge devices presents unique challenges due to limited computational resources and memory capacity. Recent advancements in LLM inference serving systems focus on optimizing performance and efficiency while preserving core decoding mechanisms, addressing these constraints. The trend towards local deployment of lightweight LLMs, such as Gemini Nano and LLAMA2 7B, enhances user privacy and control over personal data. Evaluations of LLM performance on commercial mobile devices reveal critical metrics affecting user experience, including token throughput, latency, and battery consumption, alongside vital developer considerations like resource utilization and inference engine efficiency [43, 13]. Specialized techniques are necessary to optimize LLM performance while minimizing resource consumption. Advanced quantization methods, including mixed precision and low-bit quantization, significantly reduce computational demands, enabling deployment on resource-constrained devices without substantial accuracy loss.

Optimizing memory management is key for enhancing LLM efficiency on mobile and edge devices. By effectively utilizing LLM characteristics and their key-value (KV) caches, context switching latency is significantly reduced, improving overall responsiveness during inference [74]. This approach allows multiple applications to share a single LLM instance, maximizing resource utilization and preventing device overload.

Implementing cost-based scheduling frameworks that allocate LLM tasks across heterogeneous hardware based on energy efficiency and performance requirements further optimizes resource usage. This strategic approach intelligently selects between energy-efficient processors and high-performance GPUs based on specific task needs, optimizing resource usage [45, 82, 74].

Integrating advanced optimization techniques is crucial for effectively addressing the challenges of deploying LLMs on mobile and edge devices. These techniques enhance model performance and resource efficiency while accommodating mobile hardware constraints, such as memory limitations and processing power. By leveraging optimization algorithms, LLMs can maintain persistent states during execution, reduce context switching latency, and improve overall responsiveness [79, 83, 74]. Through specialized quantization methods, optimized memory management, and strategic task scheduling, LLMs achieve substantial improvements in efficiency and performance, broadening their applicability across a wide range of mobile and edge applications.

7 Challenges and Future Directions

7.1 Future Directions and Challenges

The advancement of Large Language Models (LLMs) introduces several challenges and opportunities, necessitating ongoing research to enhance their efficiency and applicability across diverse domains. A critical area for future exploration is developing comprehensive evaluation frameworks that mitigate data contamination risks and enhance the reliability of LLM assessments [22]. This involves refining benchmarks to encompass a wider array of tasks and evaluation criteria, reflecting the complexities of human language and reasoning [2].

Research should focus on optimizing quantization techniques, particularly activation quantization, which is crucial in specific contexts [19]. Investigating advanced methods like Bound Aware Clipping (BAC) and their applicability across various neural network architectures remains essential [32]. Additionally, integrating optimizations in transformation function learning and examining the impact of different sparsity levels on diverse network architectures warrant further investigation [41].

In model compression, there is an urgent need for more efficient algorithms adaptable to various domains and smaller, open-source LLMs [38]. Enhancing the scalability of inductive learning processes could be achieved by incorporating LLM outputs into the Inductive Logic Programming (ILP) framework, leveraging the strengths of both methodologies. This requires experimentation with larger datasets and alternative approaches for modeling latent embeddings in discrete latent spaces [49].

Addressing limitations in current models, such as mitigating hallucinations and improving performance on longer contexts, remains critical [21]. Developing faster optimizers, optimal compander designs, and extending methods like CVXQ to activation quantization are vital for advancing LLM efficiency [84]. Future research could also refine dynamic sparse attention patterns and their applicability across a broader range of LLM architectures and tasks [18].

On-device LLM challenges, including the effects of thermal states and resource constraints, require benchmarks that adequately consider these factors [1]. Enhancing scheduling algorithms and investigating resource management optimizations in LLM services are critical research areas [14].

Finally, refining gating functions and strategies for enhancing prompt training in complex multi-task environments is essential [42]. Future research will also focus on applying techniques like ALC3 to datasets from deployed AI systems and assessing their effectiveness with other LLMs [66]. By addressing these challenges and pursuing these research directions, the field of LLM efficiency can progress, ensuring that LLMs remain at the forefront of AI innovation and capable of addressing a wide array of applications with improved efficiency and effectiveness.

7.2 Developing Comprehensive Evaluation Frameworks

Benchmark	Size	Domain	Task Format	Metric
BL[22]	1,000,000	Question Answering	Evaluation Tasks	Accuracy, F1-score
MobileAIBench[83]	20,000	Trust	Safety	Question Answering Accuracy, F1 Score
LaMP[4]	2,487	Personalization	Classification	MAE, Rouge-L
SR-IBN[82]	180	Intent-Based Networking	Intent Classification	Accuracy
SLOVAK-BERT[85]	300	Banking	Intent Classification	In-scope Accuracy, Out-of-scope FPR
LLM-RB[86]	100,000	Natural Language Inference	Question Answering	Accuracy, ROUGE
DEMON[87]	477,720	Visual Instruction Following	Instruction-Response	ROUGE-L, Accuracy
LLMEval2[88]	2,553	Text Summarization	Question Answering	Accuracy, Kappa

Table 5: Table presents a comprehensive overview of various benchmarks used to evaluate Large Language Models (LLMs) across multiple domains. It details the size, domain, task format, and metrics employed for each benchmark, highlighting the diversity in evaluation approaches. This information is crucial for understanding the scope and focus of existing benchmarks in the context of LLM evaluation frameworks.

Developing comprehensive evaluation frameworks is essential for accurately assessing the effectiveness of efficiency techniques applied to Large Language Models (LLMs). These frameworks must address complexities like catastrophic forgetting and the balance between specialization and generalization, often overlooked in current studies [51]. By integrating diverse evaluation prompts and thorough data contamination analyses, these frameworks can minimize benchmark leakage risks, ensuring performance assessments genuinely reflect model capabilities [22]. Table 5 provides a detailed overview of representative benchmarks employed in the evaluation of Large Language Models (LLMs), illustrating the diversity in size, domain, task format, and evaluation metrics.

A crucial aspect of future research involves reconciling different model architectures in federated settings while enhancing privacy protections during inference [89]. This requires developing evaluation metrics that address the unique challenges posed by distributed and privacy-preserving model deployments.

Incorporating activation quantization techniques into evaluation frameworks is vital for capturing dependencies across multiple layers, as these techniques significantly influence model efficiency [19]. By recognizing these dependencies, evaluation frameworks can deliver a more holistic assessment of model performance, ensuring efficiency techniques do not compromise accuracy or generalization capabilities.

Establishing robust evaluation frameworks is fundamental for improving LLM efficiency, enabling more accurate performance assessments across diverse use cases, addressing existing benchmark limitations, and supporting the development of reliable LLM-as-a-Judge systems capable of providing consistent and scalable evaluations in various domains [8, 9, 86]. These frameworks must be adaptable to various model architectures and deployment scenarios, offering comprehensive insights into the effectiveness of efficiency techniques and guiding future research directions in LLM optimization.

8 Conclusion

This survey underscores the pivotal contributions of quantization, model compression, and neural network optimization in augmenting the efficiency of Large Language Models (LLMs). By employing techniques like vector quantization and mixed precision, LLMs can significantly reduce computational and memory demands, enabling their deployment in environments with limited resources while maintaining accuracy. Model compression strategies, such as pruning and knowledge distillation, further streamline model size and computational load, ensuring effective deployment without compromising performance. Neural network optimization, through advanced methods like parameter-efficient tuning and architecture search, plays a crucial role in refining LLMs to deliver high performance with reduced computational expense. Collectively, these advancements not only enhance the operational efficiency of LLMs but also expand their applicability across various domains. As AI and machine learning continue to evolve, these strategies will be instrumental in developing more efficient, reliable, and adaptable language models, thereby advancing the integration of AI technologies into diverse applications.

References

[1] Tolga Çöplü, Marc Loedi, Arto Bendiken, Mykhailo Makohin, Joshua J. Bouw, and Stephen Cobb. A performance evaluation of a quantized large language model on various smartphones, 2023.

[2] Timothy R. McIntosh, Teo Susnjak, Nalin Arachchilage, Tong Liu, Paul Watters, and Malka N. Halgamuge. Inadequacies of large language model benchmarks in the era of generative artificial intelligence, 2024.

[3] Hengchao Shang, Zongyao Li, Jiaxin Guo, Shaojun Li, Zhiqiang Rao, Yuanchang Luo, Daimeng Wei, and Hao Yang. An end-to-end speech summarization using large language model, 2024.

[4] Mert Yazan, Suzan Verberne, and Frederik Situmeang. The impact of quantization on retrieval-augmented generation: An analysis of small llms, 2024.

[5] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding, 2024.

[6] Shuxiao Ma, Linyuan Wang, Senbao Hou, and Bin Yan. Aligned with llm: a new multi-modal training paradigm for encoding fmri activity in visual cortex, 2024.

[7] Xiaoming Shi, Jie Xu, Jinru Ding, Jiali Pang, Sichen Liu, Shuqing Luo, Xingwei Peng, Lu Lu, Haihong Yang, Mingtao Hu, Tong Ruan, and Shaoting Zhang. Llm-mini-cex: Automatic evaluation of large language model for diagnostic conversation, 2023.

[8] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[9] Boshko Koloski, Nada Lavrač, Bojan Cestnik, Senja Pollak, Blaž Škrlj, and Andrej Kastrin. Aham: Adapt, help, ask, model – harvesting llms for literature mining, 2023.

[10] Marcel Lamott and Muhammad Armaghan Shakir. Leveraging distillation techniques for document understanding: A case study with flan-t5, 2024.

[11] Phoebe Jing, Yijing Gao, Yuanhang Zhang, and Xianlong Zeng. Translating expert intuition into quantifiable features: Encode investigator domain knowledge via llm for enhanced predictive analytics, 2024.

[12] Emily Johnson and Noah Wilson. Enhancing agricultural machinery management through advanced llm integration, 2024.

[13] Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Llm inference serving: Survey of recent advances and opportunities, 2024.

[14] Bin Lin, Chen Zhang, Tao Peng, Hanyu Zhao, Wencong Xiao, Minmin Sun, Anmin Liu, Zhipeng Zhang, Lanbo Li, Xiafei Qiu, Shen Li, Zhigang Ji, Tao Xie, Yong Li, and Wei Lin. Infinite-llm: Efficient llm service for long context with distattention and distributed kvcache, 2024.

[15] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey, 2021.

[16] Qiaoling Chen, Diandian Gu, Guoteng Wang, Xun Chen, YingTong Xiong, Ting Huang, Qinghao Hu, Xin Jin, Yonggang Wen, Tianwei Zhang, and Peng Sun. Internevo: Efficient long-sequence large language model training via hybrid parallelism and redundant sharding, 2024.

[17] Lang Xu, Quentin Anthony, Qinghua Zhou, Nawras Alnaasan, Radha R. Gulhane, Aamir Shafi, Hari Subramoni, and Dhabaleswar K. Panda. Accelerating large language model training with hybrid gpu-based compression, 2024.

[18] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024.

[19] Junhan Kim, Ho young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. Boa: Attention-aware post-training quantization without backpropagation, 2025.

[20] Tycho FA van der Ouderaa, Markus Nagel, Mart Van Baalen, Yuki M Asano, and Tijmen Blankevoort. The llm surgeon. *arXiv preprint arXiv:2312.17244*, 2023.

[21] Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling, 2024.

[22] Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

[23] Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeigham, Aditya G. Parameswaran, and Eugene Wu. Towards accurate and efficient document analytics with large language models, 2024.

[24] Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation, 2023.

[25] Séamus Lankford and Andy Way. Leveraging llms for mt in crisis scenarios: a blueprint for low-resource languages, 2024.

[26] Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. And the bit goes down: Revisiting the quantization of neural networks. *arXiv preprint arXiv:1907.05686*, 2019.

[27] Fuwen Tan, Royson Lee, Łukasz Dudziak, Shell Xu Hu, Sourav Bhattacharya, Timothy Hospedales, Georgios Tzimiropoulos, and Brais Martinez. Mobilequant: Mobile-friendly quantization for on-device language models, 2024.

[28] Xiaoxia Wu, Cheng Li, Reza Yazdani Aminabadi, Zhewei Yao, and Yuxiong He. Understanding int4 quantization for transformer models: Latency speedup, composability, and failure cases, 2023.

[29] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference, 2019.

[30] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019.

[31] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. *arXiv preprint arXiv:2109.12948*, 2021.

[32] Andrea Fasoli, Chia-Yu Chen, Mauricio Serrano, Xiao Sun, Naigang Wang, Swagath Venkataramani, George Saon, Xiaodong Cui, Brian Kingsbury, Wei Zhang, Zoltán Tüske, and Kailash Gopalakrishnan. 4-bit quantization of lstm-based speech recognition models, 2021.

[33] Albert Tseng, Qingyao Sun, David Hou, and Christopher M De Sa. Qtip: Quantization with trellises and incoherence processing. *Advances in Neural Information Processing Systems*, 37:59597–59620, 2024.

[34] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7308–7316, 2019.

[35] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

[36] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 365–382, 2018.

[37] 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*, pages 38087–38099. PMLR, 2023.

[38] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

[39] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[40] Huixin Sun, Runqi Wang, Yanjing Li, Xianbin Cao, Xiaolong Jiang, Yao Hu, and Baochang Zhang. P4q: Learning to prompt for quantization in visual-language models, 2024.

[41] Peisong Wang, Qinghao Hu, Yifan Zhang, Chunjie Zhang, Yang Liu, and Jian Cheng. Two-step quantization for low-bit neural networks. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 4376–4384, 2018.

[42] Chen Dun, Mirian Hipolito Garcia, Guoqing Zheng, Ahmed Hassan Awadallah, Anastasios Kyrillidis, and Robert Sim. Sweeping heterogeneity with smart mops: Mixture of prompts for llm task adaptation, 2023.

[43] Jie Xiao, Qianyi Huang, Xu Chen, and Chen Tian. Large language model performance benchmarking on mobile platforms: A thorough evaluation, 2024.

[44] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey, 2024.

[45] Grant Wilkins, Srinivasan Keshav, and Richard Mortier. Hybrid heterogeneous clusters can lower the energy consumption of llm inference workloads, 2024.

[46] Yeonhong Park, Jake Hyun, SangLyul Cho, Bonggeun Sim, and Jae W. Lee. Any-precision llm: Low-cost deployment of multiple, different-sized llms, 2024.

[47] Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. Adaptive quantization for deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[48] Jan Achterhold, Jan Mathias Koehler, Anke Schmeink, and Tim Genewein. Variational network quantization. In *International conference on learning representations*, 2018.

[49] Hieu-Thi Luong and Junichi Yamagishi. Preliminary study on using vector quantization latent spaces for tts/vc systems with consistent performance, 2021.

[50] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. A survey on efficient inference for large language models, 2024.

[51] Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey, 2024.

[52] Eric Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *Advances in Neural Information Processing Systems*, 36:28699–28722, 2023.

[53] Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, 14(6):1–50, 2023.

[54] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[55] Han Guo, Philip Greengard, Eric P. Xing, and Yoon Kim. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning, 2024.

[56] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374, 2024.

[57] Tianyi Zhang, Junda Su, Aditya Desai, Oscar Wu, Zhaozhuo Xu, and Anshumali Shrivastava. Sketch to adapt: Fine-tunable sketches for efficient llm adaptation, 2025.

[58] Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse. Large language models can accomplish business process management tasks, 2023.

[59] Jonathan D Chang, Kiante Brantley, Rajkumar Ramamurthy, Dipendra Misra, and Wen Sun. Learning to generate better than your llm. *arXiv preprint arXiv:2306.11816*, 2023.

[60] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11):3245–3262, 2021.

[61] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models, 2024.

[62] Graziano A. Manduzio, Federico A. Galatolo, Mario G. C. A. Cimino, Enzo Pasquale Scilingo, and Lorenzo Cominelli. Improving small-scale large language models function calling for reasoning tasks, 2024.

[63] Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, et al. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.

[64] Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, Yike Guo, and Wei Xue. Codec does matter: Exploring the semantic shortcoming of codec for audio language model, 2024.

[65] Arhum Ishtiaq, Sara Mahmood, Maheen Anees, and Neha Mumtaz. Model compression, 2021.

[66] Karan Taneja and Ashok Goel. Can active label correction improve llm-based modular ai systems?, 2024.

[67] Sean I. Young. Foundations of large language model compression – part 1: Weight quantization, 2024.

[68] Sebastian Eliassen and Raghavendra Selvan. Activation compression of graph neural networks using block-wise quantization with improved variance minimization, 2024.

[69] Prateek Verma. Wavelet gpt: Wavelet inspired large language models, 2025.

[70] Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. D2llm: Decomposed and distilled large language models for semantic search, 2024.

[71] Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts, 2024.

[72] Xinyu Zhang, Linmei Hu, Luhao Zhang, Dandan Song, Heyan Huang, and Liqiang Nie. Laser: Parameter-efficient llm bi-tuning for sequential recommendation with collaborative information, 2024.

[73] Jialiang Wang, Shimin Di, Hanmo Liu, Zhili Wang, Jiachuan Wang, Lei Chen, and Xiaofang Zhou. Computation-friendly graph neural network design by accumulating knowledge on large language models, 2024.

[74] Wangsong Yin, Mengwei Xu, Yuanchun Li, and Xuanzhe Liu. Llm as a system service on mobile devices. *arXiv preprint arXiv:2403.11805*, 2024.

[75] Tuan Bui, Oanh Tran, Phuong Nguyen, Bao Ho, Long Nguyen, Thang Bui, and Tho Quan. Cross-data knowledge graph construction for llm-enabled educational question-answering system: A case study at hcmut, 2024.

[76] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.

[77] Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations. *Advances in Neural Information Processing Systems*, 37:9474–9506, 2024.

[78] Adarsh MS, Jithin VG, and Ditto PS. Efficient hybrid inference for llms: Reward-based token modelling with selective cloud assistance, 2024.

[79] Sen Huang, Kaixiang Yang, Sheng Qi, and Rui Wang. When large language model meets optimization, 2024.

[80] Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision, 2024.

[81] Lasal Jayawardena and Prasan Yapa. Parameter efficient diverse paraphrase generation using sequence-level knowledge distillation, 2024.

[82] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. Semantic routing for enhanced performance of llm-assisted intent-based 5g core network management and orchestration, 2024.

[83] Rithesh Murthy, Liangwei Yang, Juntao Tan, Tulika Manoj Awalgaonkar, Yilun Zhou, Shelby Heinecke, Sachin Desai, Jason Wu, Ran Xu, Sarah Tan, Jianguo Zhang, Zhiwei Liu, Shirley Kokane, Zuxin Liu, Ming Zhu, Huan Wang, Caiming Xiong, and Silvio Savarese. Mobileaibench: Benchmarking llms and lmms for on-device use cases, 2024.

[84] Joonhyung Lee, Jeongin Bae, Byeongwook Kim, Se Jung Kwon, and Dongsoo Lee. To fp8 and back again: Quantifying the effects of reducing precision on llm training stability, 2024.

[85] Bibiána Lajčinová, Patrik Valábek, and Michal Spišiak. Intent classification for bank chatbots through llm fine-tuning, 2024.

[86] Melissa Ailem, Katerina Marazopoulou, Charlotte Siska, and James Bono. Examining the robustness of llm evaluation to the distributional assumptions of benchmarks, 2024.

[87] Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions, 2024.

[88] Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*, 2023.

[89] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models, 2023.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.