A Survey of Optimization Techniques for Large Language Models

www.surveyx.cn

Abstract

This survey paper explores the optimization techniques for Large Language Models (LLMs), emphasizing their transformative impact on artificial intelligence and natural language processing. It highlights the Mixture of Experts (MoE) architecture as a pivotal advancement in enhancing scalability and performance, particularly in multilingual contexts, and addresses overfitting challenges in low-resource languages. The paper discusses parameter-efficient methods like Low-Rank Adaptation (LoRA) and Parameter-Efficient Fine-Tuning (PEFT), which optimize computational resources while maintaining high model performance. Continued pretraining and supervised fine-tuning (SFT) are identified as critical strategies for refining LLM capabilities. The survey underscores the importance of dynamic adaptation strategies, such as the APA framework, which enhance unlearning efficiency and recommendation performance. Future research directions include the development of robust multilingual datasets to address bias and toxicity mitigation, the enhancement of post-pretraining applications like the TG method, and the exploration of quantization techniques for scalability. The study concludes with a call for further exploration into applications in low-resource languages and human-involved evaluations for a deeper understanding of translation quality. These insights pave the way for more efficient and effective AI applications across various domains, ensuring the continued advancement of LLMs in diverse and dynamic environments.

1 Introduction

1.1 Significance of LLMs in AI

Large Language Models (LLMs) play a pivotal role in advancing artificial intelligence, particularly in natural language processing (NLP). Their ability to generate human-like text significantly enhances machine translation, improving both quality and fluency [1]. In Knowledge Base Question Answering (KBQA), LLMs effectively translate natural language queries into executable logical forms, bridging user inquiries with structured knowledge bases [2].

The integration of LLMs into multimodal applications underscores their significance, with recent advancements focusing on scalable models and diverse datasets to improve performance [3]. The Universal Approximation Theory (UAT) elucidates the theoretical foundations of LLMs, highlighting their capability to approximate complex functions in NLP tasks [4].

Beyond technical capabilities, LLMs can convert expert insights into quantifiable features, enhancing model performance and bridging the qualitative-quantitative divide in predictive modeling [5]. However, their effectiveness in specialized domains, such as scientific literature, is hindered by a lack of domain-specific knowledge, necessitating further adaptation and training [6].

Understanding LLM structure and capabilities is crucial for predicting their behavior, which is essential for effective deployment across various AI applications [7]. The transformative impact of

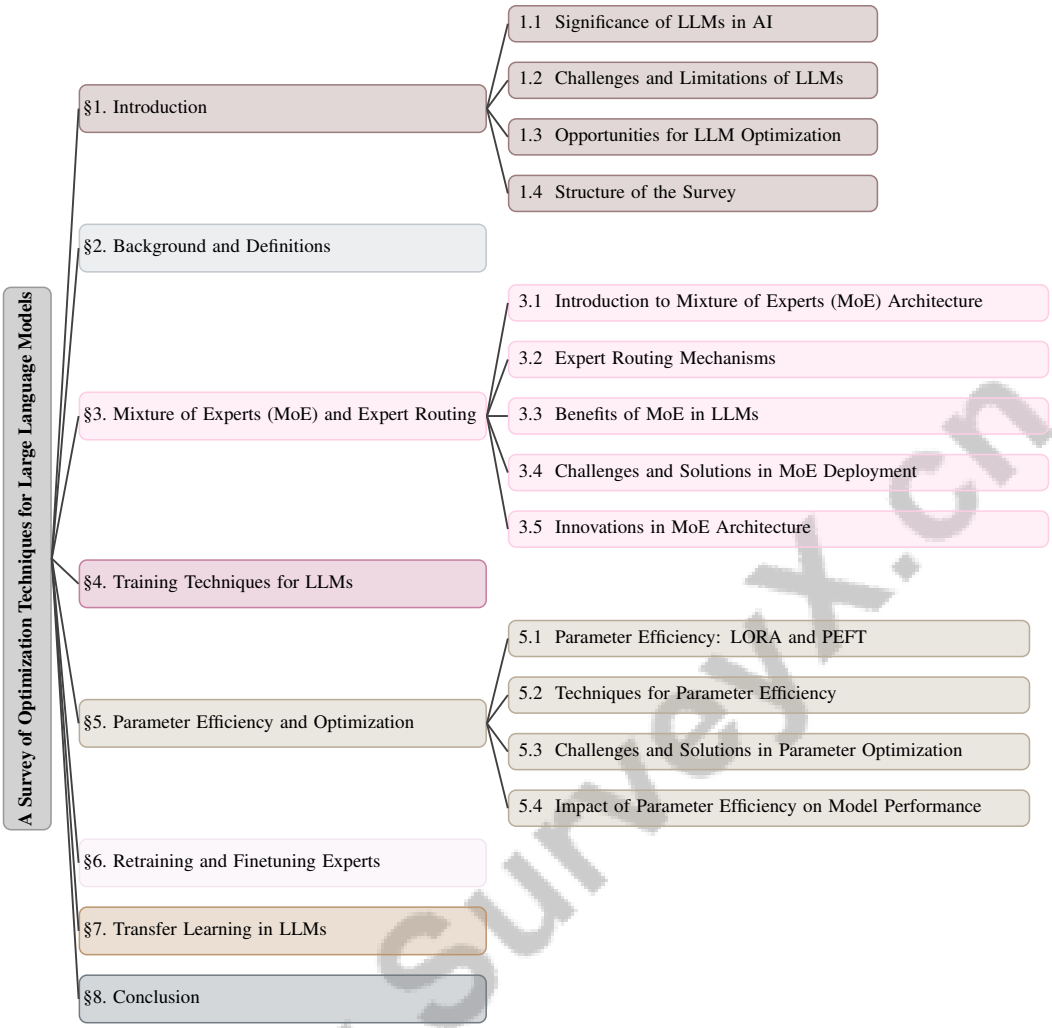


Figure 1: chapter structure

LLMs spans multiple domains, including e-commerce, healthcare, and multilingual applications, underscoring their critical role in driving AI advancements.

1.2 Challenges and Limitations of LLMs

LLMs face numerous challenges that demand advanced optimization strategies to improve scalability and efficiency. A primary challenge is the substantial computational and memory resources required for training and deployment, particularly for architectures like Mixture of Experts (MoE), which escalate pre-training costs and inference latency [8]. The rapid growth of LLMs has accentuated issues related to training efficiency and computational costs, highlighting the necessity for optimization techniques [9]. Additionally, the energy consumption associated with training large-scale models calls for more energy-efficient architectures [6].

LLMs often struggle to comprehend and extract information from scientific literature due to insufficient scientific knowledge, limiting their application in specialized fields [6]. In healthcare, the deployment of LLMs is constrained by the need for extensive domain-specific knowledge and concerns about the proprietary nature of existing models [10]. Current benchmarks fail to offer explanatory or predictive insights into LLM performance, complicating efforts to understand their capabilities [7].

The adaptation of LLMs to new domains is hindered by inefficiencies in scaling transformer-based models, with current methods not optimizing model depth effectively, leading to redundant layer rep-

resentations [11]. This inefficiency is exacerbated by limitations in fine-tuning methods, constrained by GPU memory and the inability to manage multiple tasks simultaneously [12]. Catastrophic forgetting during language adaptation is a significant challenge, particularly for smaller LLMs, which struggle to maintain performance across less-represented languages [13].

Moreover, LLMs encounter difficulties in executing complex computations and providing accurate information, often resulting in errors known as hallucinations [14]. The inefficiency of LLM inference under small-batch conditions, primarily due to memory access limitations, necessitates further optimization [8]. The inability to selectively remove knowledge from LLMs, particularly in light of data privacy regulations like the 'Right to be Forgotten,' poses significant challenges, as existing methods lack the required flexibility and efficiency for rapid adaptation without comprehensive retraining [15].

In multilingual contexts, zero-shot cross-lingual knowledge transfer remains a challenge, with multilingual pretrained language models (mPLMs) often generating text in incorrect languages, necessitating optimization [16]. Current long context LLMs can process inputs of up to 100,000 tokens but struggle to generate outputs exceeding 2,000 words, indicating limitations in handling extensive textual information [6]. Additionally, effectively utilizing heterogeneous data from multiple sources to construct specialized prompts for LLMs that can adapt to diverse tasks without extensive retraining remains a challenge [17]. These challenges underscore the urgent need for optimization techniques to enhance LLM performance and efficiency across various domains.

1.3 Opportunities for LLM Optimization

Optimizing Large Language Models (LLMs) presents numerous opportunities to enhance performance and efficiency, particularly through advanced fine-tuning and model compression techniques. The integration of Knowledge Distillation (KD) with Mixture of Experts (MoE) architecture exemplifies a promising approach, facilitating the development of specialized multilingual models that leverage both techniques for improved efficiency and adaptability [8]. Model-agnostic self-decompression methods, such as Tree Generation (TG), offer solutions to mitigate catastrophic forgetting during fine-tuning by generating synthetic data to preserve embedded knowledge within LLMs [4].

Parameter-efficient fine-tuning methods provide a cost-effective alternative to full model training, especially for sparse MoE models, allowing specialization without incurring high training costs, thus optimizing resource usage [14]. Simple depth pruning techniques can effectively compress LLMs while maintaining or enhancing performance, highlighting the potential for efficient model compression [17].

In multilingual and low-resource language processing, robust benchmarks for evaluating in-context learning abilities, particularly in translation tasks, emphasize the need for optimized LLMs capable of addressing diverse linguistic challenges [10]. Additionally, developing effective bias and toxicity mitigation methods applicable across languages is critical for ensuring ethical and inclusive LLM performance [16].

Enhancing fine-tuning frameworks that utilize diverse feedback sources presents further optimization opportunities. These frameworks can significantly improve LLM adaptability and effectiveness across various applications by simultaneously enhancing multiple performance metrics [6]. Optimizing summarization techniques for user behavior can also improve recommendation performance, showcasing the potential for LLMs in personalized AI applications [7].

New benchmarks tailored for evaluating ultra-long generation and long-text understanding offer avenues for optimizing LLMs. By addressing previous benchmark limitations and providing diverse user writing instructions, these frameworks can develop LLMs capable of handling extensive textual information, expanding their applicability [18]. Frameworks like ModelGPT, which leverage LLMs to generate customized models based on user data or task descriptions, enhance AI accessibility and adaptability [11].

The use of local LLMs in structured data extraction tasks presents significant optimization opportunities in medical data applications, improving the models' ability to tackle domain-specific challenges [6]. Integrating modality-specific encoders with sparse MoE architecture, as proposed in Uni-MoE, enables efficient processing of diverse modalities, including audio, video, text, and images, broadening LLM application scope [8].

Collectively, these opportunities highlight the vast potential for optimizing LLMs, paving the way for more efficient AI applications across various domains. The exploration of innovative techniques, such as GRIFFIN, which adaptively selects feedforward experts based on input sequences without extensive retraining, illustrates ongoing advancements in this field [14]. Additionally, frameworks like PORTLLM facilitate seamless transfer of domain-specific knowledge across evolving models, addressing model personalization and adaptability challenges [4].

1.4 Structure of the Survey

This survey is structured to provide a comprehensive overview of optimization techniques for Large Language Models (LLMs), systematically addressing critical aspects such as the integration of optimization algorithms with LLM architectures, advancements in fine-tuning methodologies, and practical deployment strategies. By exploring the synergy between LLMs and optimization techniques, the survey highlights innovative approaches to enhance model performance, manage computational challenges, and optimize decision-making processes in dynamic environments, serving as a valuable resource for researchers and practitioners aiming to advance their understanding and application of LLMs in real-world scenarios [19, 20, 21].

The survey begins with an **Introduction** that highlights the significance of LLMs in artificial intelligence, emphasizing their transformative impact and the pressing need for optimization strategies. This is followed by a discussion on the **Challenges and Limitations of LLMs**, outlining obstacles faced in effectively scaling and deploying these models.

Subsequently, the survey explores **Opportunities for LLM Optimization**, presenting innovative approaches and techniques to enhance model performance and efficiency. The **Background and Definitions** section provides foundational knowledge on LLM architectures and key terminologies, setting the stage for deeper exploration.

The core sections delve into specific optimization strategies, starting with **Mixture of Experts (MoE) and Expert Routing**, examining the role of expert routing mechanisms and their impact on model efficiency. This is followed by an analysis of **Training Techniques for LLMs**, covering methods like Continued Pretraining and Supervised Fine-Tuning (SFT) crucial for refining model capabilities.

The survey further investigates **Parameter Efficiency and Optimization**, discussing techniques such as LORA and PEFT aimed at improving parameter efficiency, alongside associated challenges and solutions. The section on **Retraining and Finetuning Experts** explores strategies for adapting LLMs to new tasks or domains with minimal computational cost.

Finally, the survey addresses the role of **Transfer Learning in LLMs**, highlighting strategies to mitigate forgetting and enhance adaptability across multilingual and domain-specific contexts. The concluding section synthesizes key findings and discusses future research and development directions in optimizing LLMs, ensuring a thorough understanding of the current landscape and future potential of LLM optimization.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Overview of Large Language Models (LLMs)

Large Language Models (LLMs) have transformed natural language processing (NLP) through transformer-based architectures that produce contextually coherent human-like text. These models are categorized into encoder-only, decoder-only, and encoder-decoder types. Encoder-only models, such as BERT, excel in tasks requiring deep text understanding, like classification and named entity recognition. Decoder-only models, exemplified by GPT, are tailored for generative tasks, including text completion and dialogue generation. Encoder-decoder models, like T5, are adept at tasks necessitating both comprehension and generation, such as machine translation and summarization [7].

The Mixture of Experts (MoE) framework significantly advances LLM architectures by enhancing scalability and efficiency, routing inputs to specialized sub-models or experts. This modular approach, facilitated by an Expert Router, is crucial for processing domain-specific datasets [22]. LLMs also show adaptability in specialized tasks, particularly in commonsense reasoning and arithmetic [17].

LLMs demonstrate versatility across domains. In e-commerce, they outperform traditional models in classification, generation, summarization, and named entity recognition [10]. Their performance in continual pre-training strategies underscores adaptability in dynamic data environments [4]. Additionally, LLMs automate systematic review processes by efficiently managing large datasets [18].

In multilingual contexts, LLMs address language inconsistencies and improve performance across languages, with techniques enhancing reasoning capabilities in non-English languages to bridge performance gaps in low-resource languages [16]. In specialized fields like medical data processing, they ensure terminological and semantic accuracy, addressing domain-specific challenges [7].

LLMs are integral to frameworks such as Retrieval Augmented Generation (RAG), excelling in complex data retrieval and matching tasks. Their ability to process long texts with context windows exceeding 100k tokens highlights their potential in managing extensive textual information [15]. The architecture and diverse applications of LLMs underscore their transformative impact on NLP and AI, establishing them as essential tools in advancing AI capabilities across various domains. Continuous innovations in LLM architectures and applications drive AI evolution, enhancing efficiency and accuracy in tackling complex tasks [14].

2.2 Key Definitions

Optimizing Large Language Models (LLMs) involves understanding several key terms essential to the methodologies employed. The Mixture of Experts (MoE) architecture enhances model efficiency by routing inputs dynamically to specialized sub-models or experts, optimizing computational resources and performance via an Expert Router [23].

Continued Pretraining involves further training a pre-existing model on additional domain-specific data to enhance task performance, improving knowledge accumulation and adaptation [24]. Supervised Fine-Tuning (SFT) focuses on refining the model with labeled data, enhancing its capabilities in specific tasks, such as recommendation tasks, including rating prediction and explanation generation [25].

Techniques like LORA (Low-Rank Adaptation) and PEFT (Parameter-Efficient Fine-Tuning) improve parameter efficiency in LLMs. LORA reduces trainable parameters by adapting a low-rank subset, while PEFT fine-tunes a fraction of the model's parameters, lowering computational costs and enhancing adaptability [23].

Transfer Learning, a strategy where a model designed for one task serves as a foundation for another, is particularly beneficial in multilingual contexts, mitigating knowledge transfer limitations and enhancing adaptability to diverse linguistic and domain-specific challenges [24]. These definitions are foundational for understanding LLM optimization techniques, facilitating their deployment across various applications and domains.

In recent years, the Mixture of Experts (MoE) framework has emerged as a pivotal architecture in the development of large language models (LLMs). This approach not only enhances model performance but also addresses the challenges of scalability and resource allocation. As illustrated in Figure 2, the hierarchical structure of MoE and its expert routing mechanisms are depicted, emphasizing the core components that contribute to its efficiency and flexibility. The figure encapsulates the benefits of MoE frameworks, detailing how expert routing can significantly influence model performance while also highlighting the ongoing innovations aimed at overcoming deployment challenges. This visual representation serves to reinforce the discussion on the transformative potential of MoE architectures in the realm of artificial intelligence.

3 Mixture of Experts (MoE) and Expert Routing

3.1 Introduction to Mixture of Experts (MoE) Architecture

The Mixture of Experts (MoE) architecture represents a significant advancement in large language models (LLMs), enhancing computational efficiency by distributing tasks across specialized sub-models or experts. This is achieved through sparse activation, where only a subset of experts is engaged for any input, reducing computational load and optimizing resource allocation. The Expert Router is pivotal in this process, directing inputs to the most suitable experts based on domain expertise, thereby enhancing model performance and flexibility [26]. Recent developments in MoE

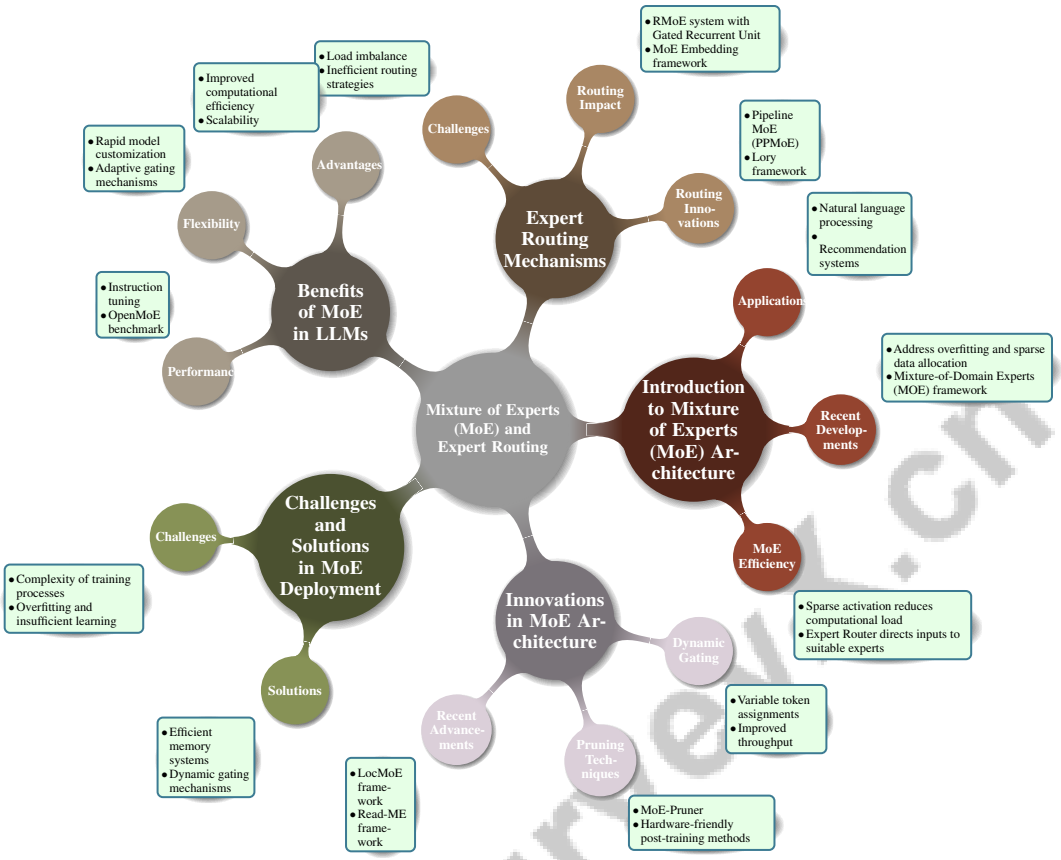
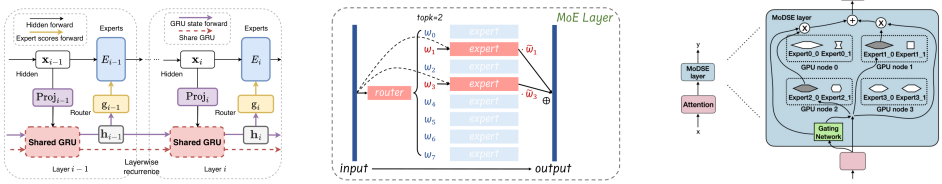


Figure 2: This figure illustrates the hierarchical structure of Mixture of Experts (MoE) and Expert Routing, showcasing the architecture's core components, expert routing mechanisms, benefits in large language models (LLMs), deployment challenges, and recent innovations. It highlights the efficiency and flexibility of MoE frameworks, the impact of expert routing on model performance, and the ongoing advancements addressing challenges in scalability and resource allocation.

architectures address challenges such as overfitting and efficient management of sparse data allocation, crucial for maintaining performance as the number of experts grows [9]. The Mixture-of-Domain Experts (MOE) framework exemplifies this potential by dynamically selecting experts for each token based on domain-specific knowledge [26]. Furthermore, methods like Mixture of Prompts (MoPs) optimize prompt tuning, leveraging data from central servers and local clients to minimize training interference across heterogeneous tasks [17].

The versatility of the MoE framework is evident in its applications in natural language processing and recommendation systems, integrating qualitative insights into quantifiable features for predictive modeling. Recent surveys categorize MoE advancements into algorithmic, systemic, and application perspectives, providing a comprehensive lens for ongoing research and development [9].

As illustrated in Figure 3, the MoE architecture enhances model efficiency and scalability by leveraging specialized components known as experts. The first example depicts a deep learning model integrating experts with recurrent neural networks (GRUs), where experts process input data to generate scores that update the GRUs, improving sequential data handling. The second example features an MoE layer configured with a top-k setting, orchestrating multiple experts via a router to selectively combine outputs, optimizing decision-making through a dot product operation. Lastly, the MoDSE framework demonstrates a novel strategy for distributed learning, employing a MoDSE layer alongside a gating network to efficiently manage expert contributions, ensuring scalable and effective learning across distributed systems. These examples collectively highlight the MoE architecture's versatility and potential in advancing deep learning models [27, 28, 29].



(a) Deep Learning Model with Experts and Recurrent Neural Networks[27]

(b) MoE Layer with Topk=2 and Router[28]

(c) MoDSE: A Novel Framework for Efficient and Scalable Distributed Learning[29]

Figure 3: Examples of Introduction to Mixture of Experts (MoE) Architecture

3.2 Expert Routing Mechanisms

Expert routing mechanisms are crucial for the efficiency and performance of MoE architectures, determining the allocation of computational resources among various experts. The Expert Router, central to this process, uses sophisticated algorithms like k-means clustering to classify and route inputs to the most appropriate expert, optimizing processing efficiency [30]. Recent innovations have led to advanced architectures like Pipeline MoE (PPMoE), which combines expert parallelism with tensor parallelism to minimize communication overhead by replacing traditional all-to-all communication with tensor index slicing and inner-node all-reduce [31]. The Lory framework enhances expert utilization through causal segment routing and similarity-based data batching, promoting specialization and refining the routing process [32].

The RMoE system introduces a Gated Recurrent Unit (GRU) to establish dependencies between routing decisions across layers, enhancing expert selection diversity and coherence [27]. The MoE Embedding (M O EE) framework integrates routing weights with LLM hidden states, demonstrating efficient routing's impact on embedding quality without additional training [33]. The LLaMA-MoE method emphasizes expert routing's role in improving model performance through expert construction and continual pre-training [34]. Adaptive routing strategies, such as those proposed by Zeng, optimize resource allocation by allowing tokens to choose between true and null experts based on specific needs, enhancing model adaptability [35].

The DeepSpeed MoE system exemplifies efficient training of large-scale MoE models by employing various forms of parallelism and offloading memory to CPUs, showcasing routing mechanisms' critical role in managing expansive architectures [36]. Observations by Li et al. have identified inefficiencies in current routing strategies due to poor load balancing, prompting the development of methods to enhance routing efficiency [37]. The MoEfication concept converts dense feed-forward layers of LLMs into a mixture-of-experts structure, allowing for visualization of expert activation frequencies and insights into expert distribution in multilingual contexts [38].

As illustrated in Figure 4, this figure categorizes expert routing mechanisms in MoE architectures, highlighting advanced architectures, routing strategies, and embedding performance methods. The MoE and Expert Routing concepts enhance machine learning model efficiency by dynamically selecting specialized subsets of components, or "experts," for processing input data. The "Top-4 Routing in CartesianMaeF: A Comparative Study" presents a comparative analysis of routing strategies within the CartesianMaeF model, emphasizing the effectiveness of Conventional Top-2 Routing for directing outputs to the most relevant neurons. The "Routing Token Adaptation for Image-Text Fusion" showcases a neural network architecture that integrates a routing token adaptation module, facilitating the fusion of image and text data through a combination of fully connected and multi-head attention layers. Lastly, the heatmap visualization provides insights into expert distribution across different layers of a neural network, highlighting task-specific expert allocation. Together, these examples underscore the diverse methodologies employed in expert routing, each tailored to leverage MoE framework strengths for specific applications [39, 40, 41].

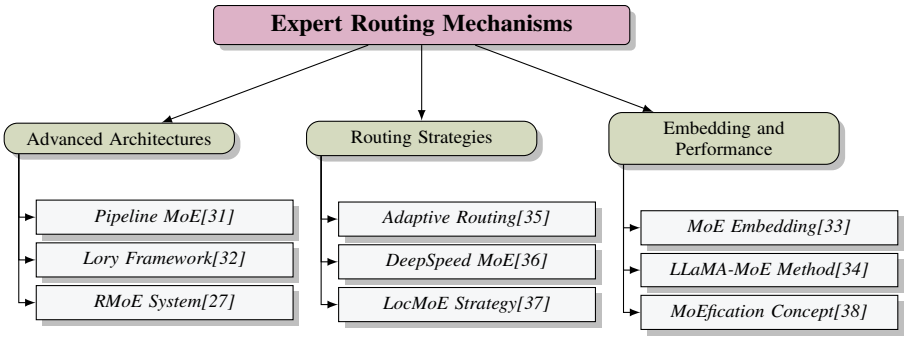


Figure 4: This figure illustrates the categorization of expert routing mechanisms in MoE architectures, highlighting advanced architectures, routing strategies, and embedding performance methods.

3.3 Benefits of MoE in LLMs

The MoE architecture significantly enhances large language models (LLMs) through improved computational efficiency and scalability. By activating only a subset of experts for any task, MoE models can achieve larger sizes without increased training costs, making them effective for complex tasks [42]. This sparse activation reduces computational demands and facilitates the integration of new experts into existing architectures without extensive retraining, as demonstrated by the MoExtend framework [43]. MoE architectures offer flexibility and cost-effectiveness, enabling rapid model customization for specific tasks without extensive retraining, as highlighted by Lee [26]. The incorporation of adaptive gating mechanisms, as identified by Cai, enhances performance by dynamically selecting the most appropriate experts for each task, optimizing resource utilization [44]. Yun's dual-metric approach, incorporating both validation loss and cost per token as constraints for budget allocation, demonstrates MoE models' potential for large-scale applications [45].

The OpenMoE benchmark supports MoE model flexibility by providing a comprehensive analysis of routing mechanisms and performance, yielding valuable insights into adaptability and efficiency [46]. The MoEC framework addresses overfitting and enhances training sample diversity, improving robustness and generalization capabilities [47]. Instruction tuning significantly enhances MoE model performance on specific tasks compared to dense models, as observed in benchmarks developed by Shen [9]. These advantages position MoE architectures as transformative in advancing LLM capabilities, enabling them to tackle increasingly complex challenges effectively.

3.4 Challenges and Solutions in MoE Deployment

Deploying MoE architectures in LLMs presents several challenges impacting scalability and efficiency. A significant issue is the complexity of traditional training processes, which are resource-intensive and hinder rapid model capability augmentation [26]. This complexity is compounded by fixed pruning decisions made during initial stages, which do not account for changes in weight importance during fine-tuning, leading to suboptimal performance [12]. As the number of experts increases, the likelihood of each expert receiving diverse training samples decreases, resulting in insufficient learning and exacerbating overfitting [47]. This is particularly problematic in low-resource language tasks, where MoE models often underperform compared to high-resource tasks. Additionally, MoE models exhibit inefficiency during inference, leading to slower performance than dense models, compounded by high latency when migrating activated experts between CPU and GPU memory.

Current benchmarks inadequately address the significant communication overhead associated with MoE architectures, inflating performance metrics when evaluated using traditional methods. Existing comparisons often rely on FLOPs or activated parameters as complexity measures, failing to capture the true computational costs incurred by the sparse layers' communication-intensive nature. This oversight can lead to an underestimation of the training budget required for MoE models. Recent studies advocate for adopting step time as a more accurate metric and implementing optimization techniques, such as dynamic gating and expert buffering, to enhance training efficiency and throughput. Without addressing these factors, the performance advantages of MoE over dense models may be overstated in conventional evaluations [48, 49, 31]. Moreover, existing routing policies contribute to load imbalance and increased training times, as they do not dynamically adjust expert gating based

on real-time performance, further highlighted by suboptimal GPU utilization during limited batch sizes, as current width pruning techniques fail to enhance inference speeds effectively.

To illustrate these complexities and potential solutions, Figure 5 presents a figure that illustrates the hierarchical structure of challenges and solutions in the deployment of Mixture of Experts (MoE) architectures. This figure highlights key deployment issues, performance metrics, and potential solutions for enhancing scalability and efficiency in large language models. To address the challenges depicted, potential solutions include developing more efficient memory systems to enhance memory augmentation methods and improving dynamic gating mechanisms for real-time adjustments based on expert performance. Refining pruning techniques to maintain model performance while reducing computational demands could significantly improve MoE model practicality. These innovative solutions, combined with ongoing research and technological advancements, hold significant potential for addressing current challenges in deploying MoE architectures, maximizing their effectiveness in LLM applications across various domains, including scientific literature understanding and efficient fine-tuning methodologies [6, 50, 20, 21].

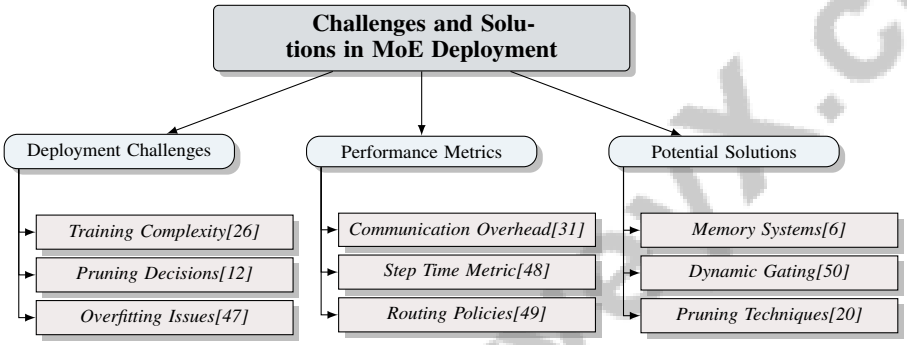


Figure 5: This figure illustrates the hierarchical structure of challenges and solutions in the deployment of Mixture of Experts (MoE) architectures, highlighting key deployment issues, performance metrics, and potential solutions for enhancing scalability and efficiency in large language models.

3.5 Innovations in MoE Architecture

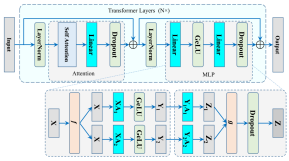
Recent advancements in MoE architecture have significantly enhanced the efficiency and scalability of LLMs. Notable innovations include the LocMoE framework, which integrates load balancing with locality in routing, effectively addressing communication overhead and improving load distribution, optimizing MoE model deployment [37]. This approach is crucial for maintaining performance while minimizing resource consumption in large-scale applications. The Read-ME framework represents a significant advancement by transforming pre-trained dense LLMs into smaller, more efficient MoE models, circumventing the high costs of training models from scratch, thus providing a cost-effective solution for deploying large-scale models without sacrificing performance [51]. Additionally, the RMoE architecture facilitates cross-layer information sharing, enhancing expert utilization and model performance by improving expert selection coherence and diversity [27].

Innovative pruning techniques, such as those employed by the MoE-Pruner, focus on router weights, ensuring that pruning aligns with the model's routing dynamics. This approach preserves performance while reducing computational demands, distinguishing it from traditional methods that overlook these aspects [52]. Furthermore, hardware-friendly post-training methods for expert pruning and skipping have been developed to focus on structured sparsity, facilitating more efficient deployment of MoE models on existing hardware architectures [28]. The Post-MoE architecture selectively introduces sparse routing in later layers, improving generalization to low-resource languages without additional parameters [53]. This selective application enhances model performance across diverse linguistic contexts. Moreover, SwapMoE combines memory swapping with expert pruning, reducing memory usage and latency during inference while avoiding typical drawbacks associated with these techniques [54].

Dynamic gating mechanisms have also been introduced, allowing variable token assignments to experts, thereby reducing computational waste and improving throughput [49]. These mechanisms ensure that resources are allocated dynamically based on task requirements, enhancing the overall

efficiency of MoE architectures. The benchmark introduced by Du et al. employs a novel comparison methodology using step time as a metric, incorporating a 3D sharding strategy to optimize MoE training, significantly improving the evaluation process [48]. This approach provides a more comprehensive understanding of MoE performance, facilitating further advancements in model efficiency and scalability.

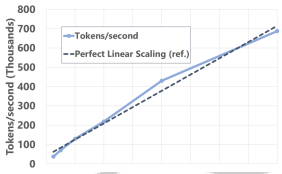
Future research directions should focus on developing more efficient training algorithms, improving load balancing techniques, and exploring new MoE applications in emerging fields [44]. These innovations represent significant strides in MoE architecture development, paving the way for more efficient and scalable LLMs.



(a) Transformer Layers[31]

# Layers	1
# Heads	4
Vocab size	32000
Embedding Dim.	512
Feature Dim.	512
MLP Intermediate Dim.	512
Activation Function	SwiGLU [32]
Positional Embedding	RoPE [33]
Normalization	RMSNorm [34]
# Params	18.0 M

(b) Model Parameters and Architectural Details[51]



(c) Performance Comparison of a Machine Learning Model with Perfect Linear Scaling[36]

Figure 6: Examples of Innovations in MoE Architecture

As shown in Figure 6, the examples delve into the intricacies of the MoE architecture and innovative expert routing approaches that enhance machine learning models' efficiency and scalability. The first image illustrates transformer layers' fundamental components, showcasing neural network elements such as self-attention mechanisms and GeLU activation functions integral to sequence modeling tasks. This visual representation emphasizes the layered structure enabling complex data processing. The second image provides a comprehensive breakdown of a neural network model's architecture, highlighting key parameters like embedding and feature dimensions crucial for understanding model capacity and performance. Lastly, the third image presents a performance comparison through a line graph, illustrating how a model achieves perfect linear scaling in processing tokens, emphasizing efficiency in handling large-scale data. Together, these images provide a holistic view of MoE architecture advancements, demonstrating how expert routing innovations can lead to more robust and scalable machine learning systems [31, 51, 36].

4 Training Techniques for LLMs

4.1 Training Techniques: Continued Pretraining and Supervised Fine-Tuning (SFT)

Method Name	Training Techniques	Optimization Strategies	Adaptability and Contextual Awareness
SL[6]	Hybrid Training Approach	-	Instruction-following Capabilities
MinorSFT[55]	Sft	Dynamic Coefficient	Better Adaptation
MoE-O[8]	Continued Pretraining	Mixture OF Experts	Adapt TO Tasks
MoELoRA[56]	Contrastive Learning	Mixture OF Experts	Dynamic Expert Selection
MOE[26]	Continued Pretraining	Mixture OF Experts	New Knowledge
PEFT[13]	Instruction Tuning	Parameter-efficient Fine-tuning	Language Adaptation
MoEC[47]	Cluster-level Dropout	Mixture OF Experts	Multilingual Contexts
LLM2LLM[57]	Iterative Data Augmentation	Targeted Approach	Multilingual And Multimodal
MoPs[17]	Prompt Instruction Tuning	Mixture OF Prompts	Dynamically Select Prompts
PSM[58]	Supervised Fine-tuning	Parameter-selection Merging	Multilingual And Multimodal
CGC-LoRA[59]	Multi-task Fine-tuning	Parameter Efficient Fine-tuning	Adapt TO Multiple

Table 1: Comparison of various training techniques, optimization strategies, and adaptability features across different methods for enhancing large language models (LLMs). This table outlines the specific training approaches, such as hybrid training and supervised fine-tuning, alongside optimization strategies including mixture of experts and dynamic coefficient adjustments. Additionally, it highlights the adaptability and contextual awareness capabilities of each method, emphasizing their applications in multilingual and multimodal contexts.

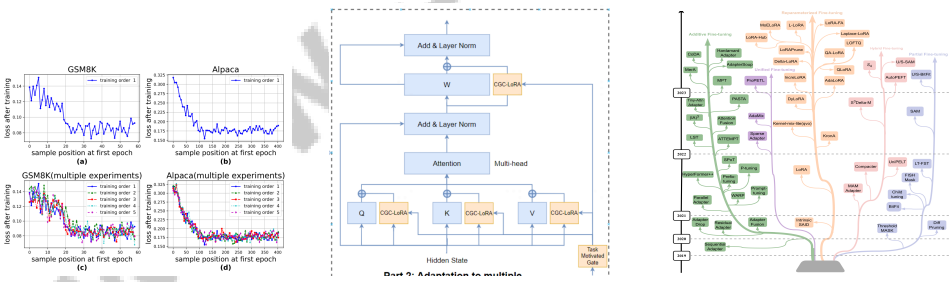
Continued pretraining and supervised fine-tuning (SFT) are pivotal for enhancing large language models (LLMs). Continued pretraining involves additional training on domain-specific datasets, enriching the model's specialized knowledge and comprehension, particularly in fields like scientific literature [6]. SFT further refines these models using labeled datasets, aligning them with specific tasks to boost precision. A two-stage SFT approach has notably improved performance in multilingual medical benchmarks [10], with hyperparameter tuning playing a crucial role in maintaining language generation accuracy [16]. The MinorSFT loss function enhances training effectiveness, reducing deviation during SFT [55].

Incorporating Mixture of Experts (MoE) models into pretraining and fine-tuning exemplifies optimization potential. MoE architectures distribute feed-forward network parameters among experts, enhancing performance across tasks [8]. Contrastive learning-guided fine-tuning further adapts LLMs to specific environments [56], and the MOE method combines expert models with a base model to improve task-specific outcomes [26].

Advanced techniques such as parameter-efficient fine-tuning (PEFT) methods, including LoRA, IA3, bottleneck adapters, and prefix tuning, optimize a subset of model parameters for better language adaptation [13]. These methods, alongside adaptive pruning strategies, mitigate overfitting from repeated pretraining data. MoEC organizes experts into clusters to share diverse training samples, reducing overfitting risks [47].

FLAN-MOE models use a prefix language model objective to boost task-specific performance [9], while the LLM2LLM strategy employs a teacher LLM to augment datasets, enhancing fine-tuning [57]. MoPs update relevant prompts for each task, minimizing interference in multi-task scenarios [17].

Recent studies highlight the importance of continuous pretraining and SFT in developing context-aware AI systems, enhancing LLM adaptability for clinical tasks and knowledge injection. Continuous pretraining establishes a solid foundation for fine-tuning, effectively incorporating new, domain-specific knowledge. Innovative methods like heterogeneous feedback integration and memory-guided architectures optimize these processes, ensuring LLMs retain extensive factual knowledge and generate accurate, contextually relevant responses [60, 61, 21, 62, 63]. These methodologies enhance adaptability and optimize performance across multilingual and multimodal contexts, advancing AI systems.



(a) Loss after training for different training orders on GSM8K and Alpaca[58]

(b) Adaptation to Multiple Tasks with CGC-LoRA[59]

(c) Fine-tuning Techniques Evolution Over Time[64]

Figure 7: Examples of Training Techniques: Continued Pretraining and Supervised Fine-Tuning (SFT)

As depicted in Figure 7, these examples illustrate methods to enhance LLM performance. The first image shows the impact of training orders on loss reduction, emphasizing sample position significance. The second highlights model adaptation to multiple tasks using CGC-LoRA, integrating an attention mechanism within an MLP framework. The third image depicts fine-tuning techniques' evolution, categorizing advancements into Additive, Unified, and Partial Fine-tuning, providing a visual narrative of their development in response to LLM complexity [58, 59, 64]. Table 1 presents a comprehensive analysis of different methods employed in the training, optimization, and adaptability of large language models, illustrating the diverse strategies and their contextual applications.

4.2 Full Finetuning

Method Name	Parameter Optimization	Resource Efficiency	Performance Evaluation
PSM[58]	Parameter Merging Technique	Reduce Training Imbalance	Accuracy And Win-rate
AtS[65]	Linear Model Fit	Reduced Resource Consumption	Pearson Correlation
SLERP[66]	Direct Preference Optimization	Model Merging Techniques	Benchmark Tests
SFT[60]	Hyperparameter Tuning	Systematic Dataset Generation	Accuracy Scores

Table 2: Comparison of finetuning methods for large language models (LLMs), detailing their parameter optimization strategies, resource efficiency, and performance evaluation metrics. The table highlights the distinct approaches utilized by each method to enhance LLM capabilities and maintain high performance across various tasks.

Full finetuning involves adjusting all LLM parameters to enhance task-specific performance, contrasting with partial techniques that modify only a subset. This approach can achieve higher accuracy by leveraging the model's entire capacity. A notable technique is parameter-selection merging, which combines fine-tuned models by selecting parameters based on probabilities, mitigating training imbalances and enhancing robustness [58].

The AtS method optimizes resource usage by selecting near-optimal LLMs from progressively smaller datasets while maintaining high performance [65]. SLERP (Spherical Linear Interpolation) facilitates smooth interpolation between model parameters, integrating diverse model strengths for a robust LLM [66].

Evaluating different finetuning strategies is crucial for understanding their impact on model performance, especially in QA tasks. Benchmarks highlight the importance of data selection in the finetuning process [67]. SFT updates LLMs with curated datasets containing new knowledge, improving real-world accuracy [60].

Table 2 provides a comprehensive comparison of different finetuning methods employed to enhance large language model performance, focusing on their parameter optimization, resource efficiency, and evaluation metrics. Full finetuning is pivotal in enhancing LLM task-specific performance. By employing techniques like parameter-selection merging, AtS, and SLERP, researchers can significantly improve LLM capabilities. These methods facilitate domain-specific knowledge integration and enhance instruction-following abilities, ensuring LLMs remain effective across various applications, including scientific literature understanding and specialized topic modeling. This optimization addresses challenges such as constructing high-quality training corpora and generating diverse instructional prompts, ultimately leading to improved benchmarks across domains [6, 20, 50]. These methodologies refine model performance and contribute to AI systems' evolution, ensuring adaptability and precision in a changing technological landscape.

4.3 Innovative Training Strategies

Innovative training strategies enhance LLM capabilities and efficiency, focusing on optimizing data utilization and adaptability. The ATP (Adaptive Two-stage Pruning) method recovers up to 91% of dense model performance while pruning 40% of parameters, maintaining performance with reduced computational demands [12].

The LLM2LLM framework targets specific model weaknesses, optimizing the data augmentation budget and improving performance [57]. The MinorSFT method introduces a sample-level dynamic coefficient during SFT, balancing learning strength by minimizing deviation from the original model [55].

These innovative strategies, including integrating continual pretraining and SFT, enhance LLM training by infusing specialized scientific knowledge and improving instruction-following capabilities. This approach addresses challenges in understanding scientific literature and ensures LLM effectiveness in tackling complex AI challenges. Document-wise memory architectures and guidance mechanisms refine LLM performance by optimizing content retention and recall, bolstering relevance in both scientific and general contexts [61, 6]. Future research is expected to focus on optimizing these strategies for multimodal applications and enhancing efficiency across various hardware platforms, paving the way for versatile and effective AI systems.

5 Parameter Efficiency and Optimization

5.1 Parameter Efficiency: LORA and PEFT

Parameter efficiency is crucial for optimizing large language models (LLMs), especially in environments with limited resources. Techniques like Low-Rank Adaptation (LoRA) and Parameter-Efficient Fine-Tuning (PEFT) minimize trainable parameters while maintaining model performance. LoRA achieves this by freezing most model weights and tuning only a small subset, thus reducing computational overhead [12, 68]. This method effectively lowers computational costs without compromising effectiveness.

PEFT strategies, including adapter-based methods, integrate specialized knowledge from multiple models while minimizing resource usage [26, 13]. These approaches achieve comparable or superior performance with fewer resources, making them ideal for efficiency-critical applications. MoELoRA, an innovative PEFT method, treats LoRA as a Mixture of Experts, utilizing contrastive learning to enhance expert specialization and further improve parameter efficiency [56].

Eliseev et al. introduce caching mechanisms to enhance parameter efficiency by predicting future expert needs, reducing loading times during inference [8]. The LLM2LLM framework enhances LLM performance in low-data regimes, demonstrating significant benchmark improvements through strategic fine-tuning [57]. Advanced fine-tuning techniques like LoRA, memory fine-tuning, and hybrid approaches enable LLMs to excel in specific tasks while significantly reducing traditional fine-tuning costs. These strategies optimize resource utilization and enhance adaptability to specialized domains, making LLMs more effective across various applications, including scientific literature comprehension [69, 6, 21]. The focus on parameter efficiency is crucial for developing scalable and adaptable AI systems across diverse applications.

5.2 Techniques for Parameter Efficiency

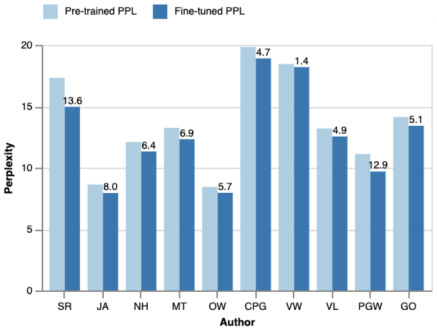
Enhancing parameter efficiency in LLMs is essential for optimizing performance while minimizing computational costs. Various techniques address the challenges of full fine-tuning large pre-trained language models (PLMs), which often require significant resources. Parameter-efficient fine-tuning (PEFT) methods, such as those in the LLM-Adapters framework, enable adapter-based PEFT tailored for specific tasks [70].

The Generate-Filter-Train (GFT) method integrates data generation, filtering, and parameter-efficient fine-tuning to enhance text classification performance in low-resource scenarios [71]. In Mixture-of-Experts (MoE) models, techniques like ESFT improve fine-tuning efficiency by focusing on the most relevant experts, while MoE-F uses real-time performance metrics to optimize predictive accuracy [72].

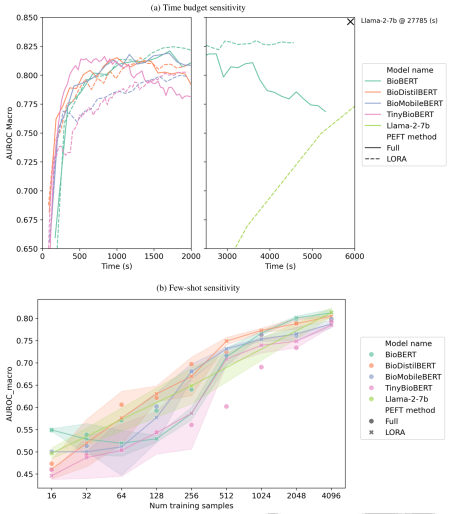
Utilizing multiple PEFT modules with Bayesian optimization enhances LLM response personalization for specific tasks [73]. Additionally, request batching and model parallelism strategies improve execution efficiency, contributing to parameter optimization [20]. Innovative pruning methods, such as those by Kim, identify and remove less critical blocks to enhance efficiency [74]. The strategic placement of Low-Rank Adaptations (LoRAs) in specific model layers optimizes language adaptation [13].

These techniques demonstrate significant parameter reductions, facilitating more efficient and accessible LLM deployment across applications. Continued emphasis on parameter efficiency is essential for optimizing resource consumption—computational power, memory, and energy—while enabling effective deployment in specialized areas like scientific literature understanding and clinical decision-making [75, 76, 50, 21, 6].

As illustrated in Figure 8, the examples provided showcase the intricate balance between model performance and resource utilization. The first example compares perplexity scores between pre-trained and fine-tuned PPL models across various authors, generally showing reduced perplexity in fine-tuned models, with noted exceptions. The second example analyzes model performance sensitivity concerning time budgets and few-shot learning scenarios, highlighting how different models respond to varying constraints. This examination emphasizes optimizing parameter efficiency and the diverse strategies employed by models to maintain performance under operational constraints [77, 76].



(a) Pre-trained and fine-tuned PPL perplexity comparison[77]



(b) Time Budget Sensitivity and Few-Shot Sensitivity of Various Models[76]

Figure 8: Examples of Techniques for Parameter Efficiency

5.3 Challenges and Solutions in Parameter Optimization

Parameter optimization in LLMs faces several challenges requiring innovative solutions to enhance efficiency and performance. A primary issue is the limited integration of computational modules in existing PEFT methods, leading to suboptimal downstream task performance [56]. This limitation can cause catastrophic forgetting during adaptation, where previously learned capabilities degrade [13]. Traditional training methods also struggle with efficiently unlearning specific tasks, causing performance degradation [15].

The complexity of dynamically adding or pruning layers is another challenge, as it is NP-hard, increasing computational costs and risking feature collapse in deeper networks. Additionally, transitioning from dense to sparse structures in MoE models can lead to performance declines due to communication overhead, which is critical for evaluating MoE model efficiency. Current memory systems in LLMs primarily operate offline, limiting adaptive learning from new queries in real-time. While recent research suggests LLMs could adopt autonomous learning strategies through direct text interaction, challenges like catastrophic forgetting persist, indicating the need for incremental learning capabilities to enhance adaptability in diverse applications [61, 78, 50, 79, 80].

In multilingual contexts, generating incoherent or irrelevant responses and producing text in incorrect languages during zero-shot cross-lingual generation complicates parameter optimization. Output instability and refusal to evaluate certain translations further underscore the necessity for robust optimization techniques. Collecting, cleaning, and labeling additional data can be costly and time-consuming, especially in specialized domains [57].

Proposed solutions include developing PEFT techniques that focus on the most relevant neurons to reduce computational costs and enhance performance. Adaptive training methods that adjust to task-specific dataset quality can mitigate overfitting risks and improve adaptability. Refining hyperparameter optimization processes through automated methods, such as blackbox optimization techniques like the NOMAD algorithm, can enhance model performance by systematically exploring hyperparameter space for optimal configurations, especially in LoRA applications [50, 81, 21].

The strategic use of continued pretraining is crucial for recovering performance after aggressive pruning, particularly in models exceeding 13B parameters. Optimizing data mixture ratios and refining reinforcement learning approaches can prevent performance trade-offs and ensure consistent model capabilities. Although MinorSFT introduces additional computational complexity, it offers pathways for better task adaptation [55].

Future research should explore optimizations in cluster size selection, routing strategies, and the applications of MoEC in various domains and tasks [47]. These solutions, coupled with ongoing research, promise to overcome current parameter optimization limitations, enabling efficient and scalable LLM deployment across diverse applications. As the field evolves, focusing on dynamic adaptation and efficient resource utilization will be key to advancing large language models' capabilities.

5.4 Impact of Parameter Efficiency on Model Performance

Parameter efficiency significantly impacts optimizing LLM performance, enhancing computational resource management while maintaining or improving accuracy. Techniques like token-adaptive routing, exemplified by AdaMOE, demonstrate how parameter efficiency can reduce floating point operations (FLOPs) while boosting accuracy across various tasks [35]. This underscores the positive effect of parameter efficiency on model performance through effective resource allocation.

The Apollo-MoE model, particularly within the medical domain, illustrates the benefits of parameter efficiency by improving performance across 50 languages while addressing data scarcity and scalability challenges [53]. Similarly, the MC-MoE framework achieves notable compression of MoE-LLMs, outperforming comparable dense models and highlighting its potential for resource-constrained environments [82].

Depth pruning methods, such as those proposed by Kim, enhance inference speed while maintaining competitive performance, especially under limited batch sizes [74]. The LocMoE strategy further exemplifies improvements in training efficiency, significantly reducing training time per epoch compared to classical routers without sacrificing accuracy [37].

In multilingual contexts, parameter-efficient methods like those proposed by Almaamari mitigate catastrophic forgetting and enhance performance across languages, improving overall efficacy [83]. The integration of pre-gated MoE algorithms, explored by Hwang, optimizes expert migration by overlapping it with current MoE block execution, reducing migration latency and boosting performance [84].

The MAP-Neo-7B model, which achieved superior performance on various benchmarks, exemplifies how parameter efficiency can yield competitive results against proprietary models, emphasizing the critical role of parameter management [85]. The GRIFFIN framework utilizes structured sparsity in activation flocking behavior to prune feedforward neurons effectively, illustrating the potential of efficient pruning techniques [86]. MoELoRA consistently outperforms LoRA across tasks, achieving significant improvements in math and common-sense reasoning, reinforcing the value of parameter-efficient strategies [56].

Collectively, these innovations highlight the transformative impact of parameter efficiency on LLM performance, enabling high accuracy while minimizing resource consumption. This facilitates the effective deployment of LLMs across diverse applications, particularly in specialized areas such as scientific literature understanding and systematic reviews, ensuring adaptability and efficiency in varying computational environments. By integrating continual pre-training and supervised fine-tuning, LLMs can be tailored to address specific challenges across different fields, enhancing their performance and accessibility in real-world settings [6, 20, 87].

6 Retraining and Finetuning Experts

Optimizing Large Language Models (LLMs) through the Mixture of Experts (MoE) framework hinges on mechanisms that enhance performance. Knowledge sharing and expert specialization are pivotal in leveraging diverse expertise within model architectures, thereby strengthening LLM capabilities in complex tasks and influencing performance dynamics within MoE architectures.

6.1 Knowledge Sharing and Expert Specialization

Within MoE frameworks, knowledge sharing and expert specialization are critical for optimizing LLMs. MoE architectures promote expert specialization by fine-tuning each expert for specific tasks or domains, enhancing overall model performance and adaptability. Frameworks like Self-MoE dynamically activate relevant experts based on task requirements, increasing flexibility and efficiency [88]. MoE architectures facilitate the distribution and integration of specialized knowledge

among experts, allowing them to leverage insights from one another. This leads to improved performance and robustness, as demonstrated by CartesianMoE, which uses a "multiplication" method for knowledge sharing to address routing accuracy and expert collaboration challenges [6, 39, 33]. Dynamic routing capabilities ensure the engagement of the most relevant experts for specific tasks, optimizing computational resource use. Strategic expert module design fosters highly specialized experts excelling in domain-specific language processing or multilingual translation, broadening LLM applicability. Integrating knowledge sharing and expert specialization within MoE frameworks enhances LLM efficiency and scalability, enabling dynamic allocation of computational resources to the most relevant experts. This approach addresses routing accuracy and collaboration challenges while facilitating advanced, contextually aware AI systems capable of managing complex tasks with improved precision across various applications, including natural language processing and computer vision [28, 39, 44, 33].

6.2 Dynamic Expert Selection and Adaptation

Dynamic expert selection and adaptation are vital for optimizing LLM performance within the MoE framework. These processes involve real-time allocation of computational resources to the most suitable experts based on input data requirements, enhancing model efficiency and effectiveness. The dynamic nature of expert selection allows models to adapt to varying tasks and domains, leveraging specialized knowledge from different experts for optimal performance [26]. Advancements focus on improving expert selection and adaptation mechanisms. Token-adaptive routing, as exemplified by the AdaMOE framework, allows flexible token allocation to experts, optimizing resource utilization and reducing computational overhead [35]. This enhances model accuracy and its ability to handle diverse tasks by dynamically adjusting expert activation based on input characteristics. Innovations like the Lory framework introduce fully differentiable MoE architectures that enhance expert utilization through causal segment routing and similarity-based data batching, promoting specialization and refining routing processes [32]. Adaptive routing strategies, such as those proposed by Zeng, allow tokens to choose between true and null experts based on specific needs, optimizing resource allocation and improving performance [35]. Real-time performance metrics inform gating mechanisms in frameworks like MoE-F, ensuring expert selection based on current performance to optimize predictive accuracy and model efficiency [72]. Pre-gated MoE algorithms, explored by Hwang, optimize expert migration by overlapping it with current MoE block execution, reducing latency and enhancing performance [84]. These strategies enable models to achieve high performance and adaptability across various applications, ensuring relevance and effectiveness in addressing complex AI challenges. Methodologies like AHAM leverage domain-specific adaptations for improved topic modeling in scientific literature, while frameworks convert expert intuition into quantifiable features for enhanced predictive analytics. Specialized LLMs like SciLitLLM highlight the importance of continual pre-training and supervised fine-tuning for scientific tasks, illustrating the critical role of expert-driven methodologies in advancing AI capabilities [6, 50, 89, 5].

6.3 Integration of Specialized Expert Modules

Integrating specialized expert modules within LLMs enhances model adaptability and performance across diverse tasks. This strategy embeds domain-specific knowledge into LLMs, enabling exceptional performance in specialized applications while retaining generalization capabilities. Balancing task-specific fine-tuning with generalization is crucial, as emphasized by benchmarks that highlight the need to prevent overfitting and ensure robust performance across domains [90]. Continuous pretraining effectively integrates specialized modules, as evidenced by its success in clinical LLMs, achieving state-of-the-art results and allowing models to assimilate new information continually, improving adaptability to evolving data landscapes and overall performance [63]. Augmenting LLMs with relational databases proves effective in accurately answering database-related questions, demonstrating the potential of integrating external knowledge sources [91]. Optimizing expert module efficiency can involve selectively removing or skipping experts based on their contribution to performance, reducing memory usage and increasing inference speed [28]. This ensures engagement of the most relevant experts, optimizing resource allocation. Fuse Distillation condenses knowledge into a shared module during training, facilitating the integration of multilingual and domain-specific knowledge and impacting model adaptability [23]. In the medical domain, the MILE framework illustrates how integrating various Parameter-Efficient Fine-Tuning methods can enhance training efficiency and performance [92]. However, the effectiveness of integrations is influenced by the

quality of external models, as seen with the MindMerger method, where the multilingual model's capabilities significantly affect overall performance [93]. Cloud-based solutions for scalable LLM inference offer platforms for integrating specialized modules, addressing cost optimization and resource management challenges [20]. These solutions facilitate the deployment of specialized modules at scale, ensuring efficient resource utilization and cost-effective operation. Future research could enhance speech-output QA capabilities and close the performance gap between multimodal models and text-only counterparts, further expanding specialized module integration [94]. Data-centric training approaches, such as those employed by Ziya2, have shown significant improvements, underscoring the impact of specialized data integration on adaptability and performance [95]. The UNLEARN method provides a framework for selectively removing knowledge from LLMs by identifying and manipulating task-specific subspaces within weight matrices, refining specialization without compromising generalization [15]. These advancements highlight the transformative potential of integrating specialized expert modules within LLMs, paving the way for more adaptable and efficient AI systems capable of tackling a wide array of tasks with precision.

7 Transfer Learning in LLMs

7.1 Mitigating Forgetting and Enhancing Transfer Learning

Addressing catastrophic forgetting while improving transfer learning in LLMs is crucial for their sustained efficacy across various tasks and domains. Catastrophic forgetting, which leads to the erosion of previously learned knowledge during new data fine-tuning, poses significant challenges to model safety and performance [90]. The Lifelong-MoE method leverages a Mixture of Experts (MoE) architecture to retain prior knowledge while integrating new information, enhancing LLM adaptability across domains [96]. This underscores the importance of dynamic expert selection in balancing knowledge retention with task-specific specialization.

The Tree Generation (TG) approach offers a model-agnostic solution to preserve general knowledge while facilitating domain-specific adaptation, enhancing LLM adaptability without compromising foundational knowledge [97]. Integrating Lifelong-MoE and TG into LLM training provides a robust framework for mitigating forgetting and enhancing transfer learning. Employing a hybrid approach that combines continual pre-training (CPT) and supervised fine-tuning (SFT), as demonstrated by SciLitLLM for scientific literature understanding, addresses challenges like high-quality training corpora and diverse instructional inputs. The Sequential Fusion method enables domain-specific knowledge integration, allowing LLMs to update capabilities without extensive retraining, achieving notable performance improvements in specialized areas like medical and economics-management datasets [98, 6]. The ongoing research focus on mitigating forgetting and optimizing transfer learning is pivotal in advancing LLM capabilities.

7.2 Multilingual and Multimodal Adaptation

Adapting LLMs to multilingual and multimodal tasks enhances performance across diverse linguistic and modality contexts. Multilingual adaptation addresses challenges such as cross-lingual transfer and language-specific nuances, especially in low-resource settings where traditional models often underperform. Techniques like zero-shot cross-lingual transfer and language-specific fine-tuning help bridge the performance gap between high-resource and low-resource languages, enhancing LLM multilingual capabilities [16].

Incorporating multimodal data—including text, audio, video, and images—into LLMs expands their applicability. The integration of modality-specific encoders within a sparse Mixture of Experts (MoE) architecture, exemplified by Uni-MoE, enhances processing efficiency for diverse modalities, enabling LLMs to tackle complex multimodal tasks [8]. This approach leverages complementary information from different modalities, improving overall task performance and understanding.

Developing robust benchmarks for evaluating LLM performance in multilingual and multimodal contexts is critical for advancing adaptation efforts. These benchmarks provide insights into LLM capabilities by identifying strengths and weaknesses across tasks, guiding enhancements in training strategies and model architectures. Tailored approaches such as continual pre-training and supervised fine-tuning address challenges related to domain knowledge and task familiarity. Additionally, benchmarks facilitate innovative memory architectures that optimize document-related content

retention, enhancing generative capabilities [61, 6, 87]. Optimizing multilingual and multimodal data integration allows LLMs to achieve greater adaptability and effectiveness across applications, from translation and content generation to complex data analysis.

7.3 Domain-Specific Adaptation

Domain-specific adaptation of LLMs is essential for enhancing performance in specialized fields, such as medicine, where precision and domain knowledge are critical. The challenge lies in effectively fine-tuning large models without incurring high computational costs while maintaining performance and generalization capabilities [92]. This necessitates developing efficient fine-tuning techniques tailored to the unique requirements of specific domains.

Parameter-efficient fine-tuning methods, such as Low-Rank Adaptation (LoRA) and adapters, optimize a subset of model parameters, integrating domain-specific knowledge with minimal computational overhead. These techniques preserve generalization capabilities while enhancing performance in specialized tasks, particularly in the medical domain [92]. Continuous pretraining of LLMs on domain-specific datasets allows models to accumulate specialized knowledge, improving comprehension of domain-specific language and concepts. This approach enhances the ability to perform complex tasks, such as medical diagnosis or treatment recommendations, by leveraging rich contextual information from specialized data [92].

Integrating external knowledge sources, such as relational databases or domain-specific ontologies, further enhances LLM adaptability. By incorporating structured knowledge into the model architecture, LLMs improve their capacity to address domain-specific queries and perform tasks requiring a deep understanding of specialized content [92]. Domain-specific adaptation involves a combination of efficient fine-tuning techniques, continuous pretraining on specialized datasets, and the integration of external knowledge sources. These strategies collectively enhance LLM performance and adaptability in specialized domains, as evidenced by models like SciLitLLM, which excels in scientific literature comprehension, and the Sequential Fusion method, which significantly improves accuracy in tasks requiring expert reasoning across diverse datasets [98, 6]. As research advances, domain-specific adaptation will remain a key driver in optimizing LLMs for specialized applications, ensuring their relevance across diverse fields.

8 Conclusion

The exploration of optimization techniques for Large Language Models (LLMs) highlights their pivotal role in transforming artificial intelligence and natural language processing. The Mixture of Experts (MoE) architecture stands out for its ability to enhance scalability and performance, especially in multilingual settings, effectively addressing the challenge of overfitting in low-resource languages. Pre-gated MoE systems contribute to improved performance and memory efficiency, enabling the creation of larger, more capable LLMs across various NLP tasks. Notably, smaller MoE models with numerous experts demonstrate superior quality and inference efficiency compared to larger models with fewer experts.

Parameter-efficient approaches, such as Low-Rank Adaptation (LoRA) and Parameter-Efficient Fine-Tuning (PEFT), have been instrumental in optimizing computational resources while maintaining model performance. Techniques like Neuron-level Fine-Tuning (NeFT) consistently outperform full-parameter fine-tuning, offering promising pathways for enhancing LLM training efficiency. The DLO framework exemplifies how dynamic vertical scaling can achieve performance levels comparable to traditional dense models, while significantly reducing training costs and enhancing inference efficiency.

Continued pretraining and supervised fine-tuning (SFT) are essential for refining LLM capabilities, with autonomous learning methods showing substantial improvements over traditional approaches. The strategic selection of training data is crucial for optimizing SFT in question-answering tasks. Dynamic adaptation strategies, such as those in the APA framework, lead to significant improvements in unlearning efficiency and recommendation performance.

Future research should focus on developing robust multilingual datasets to reduce bias and toxicity, particularly in underrepresented languages. Enhancements to post-pretraining applications and improvements in synthetic data generation quality and safety are critical areas for further exploration.

Investigating additional quantization techniques for scalability and effectiveness in automatic grading systems presents promising opportunities. Moreover, the applicability of these methods to low-resource languages and human-involved evaluations for deeper insights into translation quality requires further investigation.

The UNLEARN method demonstrates significant gains in removing unwanted knowledge from LLMs without negatively impacting related tasks, achieving high levels of forgetting on targeted tasks while maintaining performance close to the original model. Future research should strengthen the robustness of ATP under extreme sparsity and explore its applications for full-parameter fine-tuning. Key insights include a structured approach to tool learning, identifying challenges, and outlining future research directions, emphasizing the need for a unified framework for tool learning. Further exploration of hyperparameter tuning within this framework and integration with techniques like prompt tuning and few-shot learning is recommended. Enhancing benchmarks to include multilingual capabilities and examining the impact of diverse training data on model performance are also crucial areas for future research. Identifying metrics to assess model fitting during training will refine the use of MinorSFT and explore its applicability in various contexts. The study concludes that creating low-cost MoEs from trained expert models is a viable strategy for enhancing model performance, with gate-less and Noisy MoE configurations yielding competitive results. MoPs effectively mitigate prompt training interference and enhance performance across multi-source and multi-task scenarios, representing a promising approach for improving the efficiency and effectiveness of prompt-based learning systems.

This survey emphasizes the immense potential for optimizing LLMs, paving the way for more efficient and effective AI applications across diverse domains. As research advances, dynamic adaptation, parameter efficiency, and robust transfer learning will continue to drive progress in LLM capabilities. Future studies should explore applications in low-resource languages and assess translation quality through human evaluations, ensuring the continued relevance and effectiveness of LLMs in dynamic environments.

References

[1] Shenbin Qian, Constantin Orăsan, Diptesh Kanojia, and Félix do Carmo. Are large language models state-of-the-art quality estimators for machine translation of user-generated content?, 2024.

[2] Jinxin Liu, Shulin Cao, Jiaxin Shi, Tingjian Zhang, Lunyiu Nie, Linmei Hu, Lei Hou, and Juanzi Li. How proficient are large language models in formal languages? an in-depth insight for knowledge base question answering, 2024.

[3] Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-moe: Scaling unified multimodal llms with mixture of experts, 2024.

[4] Wei Wang and Qing Li. Dynamic universal approximation theory: The basic theory for transformer-based large language models, 2024.

[5] 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.

[6] Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. Scilitllm: How to adapt llms for scientific literature understanding, 2025.

[7] Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. Revealing the structure of language model capabilities, 2023.

[8] Artyom Eliseev and Denis Mazur. Fast inference of mixture-of-experts language models with offloading, 2023.

[9] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. Mixture-of-experts meets instruction tuning:a winning combination for large language models, 2023.

[10] Meng Zhou, Surajsinh Parmar, and Anubhav Bhatti. Towards democratizing multilingual large language models for medicine through a two-stage instruction fine-tuning approach, 2024.

[11] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis, 2023.

[12] Lei Lu, Zhepeng Wang, Runxue Bao, Mengbing Wang, Fangyi Li, Yawen Wu, Weiwen Jiang, Jie Xu, Yanzhi Wang, and Shangqian Gao. All-in-one tuning and structural pruning for domain-specific llms, 2024.

[13] Jenny Kunz. Train more parameters but mind their placement: Insights into language adaptation with peft, 2024.

[14] 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.

[15] Tyler Lizzo and Larry Heck. Unlearn efficient removal of knowledge in large language models, 2024.

[16] Nadezhda Chirkova, Sheng Liang, and Vassilina Nikoulina. Empirical study of pretrained multilingual language models for zero-shot cross-lingual knowledge transfer in generation, 2024.

[17] 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.

[18] Xinyu Zhao, Guoheng Sun, Ruisi Cai, Yukun Zhou, Pingzhi Li, Peihao Wang, Bowen Tan, Yexiao He, Li Chen, Yi Liang, Beidi Chen, Binhang Yuan, Hongyi Wang, Ang Li, Zhangyang Wang, and Tianlong Chen. Model-glue: Democratized llm scaling for a large model zoo in the wild, 2024.

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

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

[21] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024.

[22] Da Song, Xuan Xie, Jiayang Song, Derui Zhu, Yuheng Huang, Felix Juefei-Xu, and Lei Ma. Luna: A model-based universal analysis framework for large language models, 2024.

[23] Haoran Xu, Weiting Tan, Shuyue Stella Li, Yunmo Chen, Benjamin Van Durme, Philipp Koehn, and Kenton Murray. Condensing multilingual knowledge with lightweight language-specific modules, 2023.

[24] Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. A survey on large language models with multilingualism: Recent advances and new frontiers, 2025.

[25] Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, and Philip S. Yu. Llmrec: Benchmarking large language models on recommendation task, 2023.

[26] Rhui Dih Lee, Laura Wynter, and Raghu Kiran Ganti. Flexible and effective mixing of large language models into a mixture of domain experts, 2024.

[27] Zihan Qiu, Zeyu Huang, Shuang Cheng, Yizhi Zhou, Zili Wang, Ivan Titov, and Jie Fu. Layerwise recurrent router for mixture-of-experts, 2024.

[28] Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models, 2024.

[29] Manxi Sun, Wei Liu, Jian Luan, Pengzhi Gao, and Bin Wang. Mixture of diverse size experts, 2024.

[30] Josef Pichlmeier, Philipp Ross, and Andre Luckow. Performance characterization of expert router for scalable llm inference, 2024.

[31] Xin Chen, Hengheng Zhang, Xiaotao Gu, Kaifeng Bi, Lingxi Xie, and Qi Tian. Pipeline moe: A flexible moe implementation with pipeline parallelism, 2023.

[32] Zexuan Zhong, Mengzhou Xia, Danqi Chen, and Mike Lewis. Lory: Fully differentiable mixture-of-experts for autoregressive language model pre-training, 2024.

[33] Ziyue Li and Tianyi Zhou. Your mixture-of-experts llm is secretly an embedding model for free, 2024.

[34] Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training, 2024.

[35] Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models, 2024.

[36] Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient moe training for multitask multilingual models, 2021.

[37] Jing Li, Zhijie Sun, Xuan He, Li Zeng, Yi Lin, Entong Li, Binfan Zheng, Rongqian Zhao, and Xin Chen. Locmoe: A low-overhead moe for large language model training, 2024.

[38] Weize Liu, Yinlong Xu, Hongxia Xu, Jintai Chen, Xuming Hu, and Jian Wu. Unraveling babel: Exploring multilingual activation patterns of llms and their applications, 2024.

[39] Zhenpeng Su, Xing Wu, Zijia Lin, Yizhe Xiong, Minxuan Lv, Guangyuan Ma, Hui Chen, Songlin Hu, and Guiguang Ding. Cartesianmoe: Boosting knowledge sharing among experts via cartesian product routing in mixture-of-experts, 2025.

[40] Qiong Wu, Zhaoxi Ke, Yiyi Zhou, Xiaoshuai Sun, and Rongrong Ji. Routing experts: Learning to route dynamic experts in multi-modal large language models, 2025.

[41] Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Y. Wu. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models, 2024.

[42] Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models, 2024.

[43] Shanshan Zhong, Shanghua Gao, Zhongzhan Huang, Wushao Wen, Marinka Zitnik, and Pan Zhou. Moextend: Tuning new experts for modality and task extension, 2024.

[44] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts, 2024.

[45] Longfei Yun, Yonghao Zhuang, Yao Fu, Eric P Xing, and Hao Zhang. Toward inference-optimal mixture-of-expert large language models, 2024.

[46] Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models, 2024.

[47] Yuan Xie, Shaohan Huang, Tianyu Chen, and Furu Wei. Moec: Mixture of expert clusters, 2022.

[48] Xianzhi Du, Tom Gunter, Xiang Kong, Mark Lee, Zirui Wang, Aonan Zhang, Nan Du, and Ruoming Pang. Revisiting moe and dense speed-accuracy comparisons for llm training, 2024.

[49] Haiyang Huang, Newsha Ardalani, Anna Sun, Liu Ke, Hsien-Hsin S. Lee, Anjali Sridhar, Shruti Bhosale, Carole-Jean Wu, and Benjamin Lee. Towards moe deployment: Mitigating inefficiencies in mixture-of-expert (moe) inference, 2023.

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

[51] Ruisi Cai, Yeonju Ro, Geon-Woo Kim, Peihao Wang, Babak Ehteshami Bejnordi, Aditya Akella, and Zhangyang Wang. Read-me: Refactorizing llms as router-decoupled mixture of experts with system co-design, 2024.

[52] Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router, 2024.

[53] Guorui Zheng, Xidong Wang, Juhao Liang, Nuo Chen, Yuping Zheng, and Benyou Wang. Efficiently democratizing medical llms for 50 languages via a mixture of language family experts, 2025.

[54] Rui Kong, Yuanchun Li, Qingtian Feng, Weijun Wang, Xiaozhou Ye, Ye Ouyang, Linghe Kong, and Yunxin Liu. Swapmoe: Serving off-the-shelf moe-based large language models with tunable memory budget, 2024.

[55] Shiming Xie, Hong Chen, Fred Yu, Zeye Sun, and Xiuyu Wu. Minor sft loss for llm fine-tune to increase performance and reduce model deviation, 2024.

[56] Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models, 2024.

[57] Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. Llm2llm: Boosting llms with novel iterative data enhancement, 2024.

[58] Yiming Ju, Ziyi Ni, Xingrun Xing, Zhixiong Zeng, hanyu Zhao, Siqi Fan, and Zheng Zhang. Mitigating training imbalance in llm fine-tuning via selective parameter merging, 2024.

[59] Chao Song, Zhihao Ye, Qiqiang Lin, Qiuying Peng, and Jun Wang. A framework to implement 1+n multi-task fine-tuning pattern in llms using the cgc-lora algorithm, 2024.

[60] Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, Tolga Aktas, and Todd Hendry. Injecting new knowledge into large language models via supervised fine-tuning, 2024.

[61] Bumjin Park and Jaesik Choi. Memorizing documents with guidance in large language models, 2024.

[62] Ryan Aponte, Ryan A. Rossi, Shunan Guo, Franck Dernoncourt, Tong Yu, Xiang Chen, Subrata Mitra, and Nedim Lipka. A framework for fine-tuning llms using heterogeneous feedback, 2024.

[63] Clément Christophe, Tathagata Raha, Svetlana Maslenkova, Muhammad Umar Salman, Praveen K Kanithi, Marco AF Pimentel, and Shadab Khan. Beyond fine-tuning: Unleashing the potential of continuous pretraining for clinical llms, 2024.

[64] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023.

[65] Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. Selecting large language model to fine-tune via rectified scaling law, 2024.

[66] Wei Lu, Rachel K. Luu, and Markus J. Buehler. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities, 2024.

[67] Junjie Ye, Yuming Yang, Qi Zhang, Tao Gui, Xuanjing Huang, Peng Wang, Zhongchao Shi, and Jianping Fan. 60 data points are sufficient to fine-tune llms for question-answering, 2025.

[68] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts, 2024.

[69] Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Hao, and Nishil Talati. Understanding the performance and estimating the cost of llm fine-tuning, 2024.

[70] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models, 2023.

[71] Parth Patwa, Simone Filice, Zhiyu Chen, Giuseppe Castellucci, Oleg Rokhlenko, and Shervin Malmasi. Enhancing low-resource llms classification with peft and synthetic data, 2024.

[72] Raeid Saqur, Anastasis Kratsios, Florian Krach, Yannick Limmer, Jacob-Junqi Tian, John Willes, Blanka Horvath, and Frank Rudzicz. Filtered not mixed: Stochastic filtering-based online gating for mixture of large language models, 2024.

[73] Kai Zhang, Yejin Kim, and Xiaozhong Liu. Personalized llm response generation with parameterized memory injection, 2025.

[74] Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened llama: Depth pruning for large language models with comparison of retraining methods, 2024.

[75] Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, Xinyuan Song, Carl Yang, Yue Cheng, and Liang Zhao. Beyond efficiency: A systematic survey of resource-efficient large language models, 2024.

[76] Niall Taylor, Upamanyu Ghose, Omid Rohanian, Mohammadmahdi Nouriborji, Andrey Kormilitzin, David Clifton, and Alejo Nevado-Holgado. Efficiency at scale: Investigating the performance of diminutive language models in clinical tasks, 2024.

[77] Xinyue Liu, Harshita Diddee, and Daphne Ippolito. Customizing large language model generation style using parameter-efficient finetuning, 2024.

[78] Ke Ji, Junying Chen, Anningzhe Gao, Wenya Xie, Xiang Wan, and Benyou Wang. Llms could autonomously learn without external supervision, 2024.

[79] Zhiyu Hu, Yang Zhang, Minghao Xiao, Wenjie Wang, Fuli Feng, and Xiangnan He. Exact and efficient unlearning for large language model-based recommendation, 2024.

[80] Junhao Zheng, Shengjie Qiu, and Qianli Ma. Can llms learn new concepts incrementally without forgetting?, 2024.

[81] Christophe Tribes, Sacha Benarroch-Lelong, Peng Lu, and Ivan Kobyzev. Hyperparameter optimization for large language model instruction-tuning, 2024.

[82] Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts llms gains more, 2025.

[83] Mohammed Al-Maamari, Mehdi Ben Amor, and Michael Granitzer. Mixture of modular experts: Distilling knowledge from a multilingual teacher into specialized modular language models, 2024.

[84] Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. Pre-gated moe: An algorithm-system co-design for fast and scalable mixture-of-expert inference, 2024.

[85] Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, Raven Yuan, Tuney Zheng, Wei Pang, Xinrun Du, Yiming Liang, Yinghao Ma, Yizhi Li, Ziyang Ma, Bill Lin, Emmanouil Benetos, Huan Yang, Junting Zhou, Kaijing Ma, Minghao Liu, Morry Niu, Noah Wang, Quehry Que, Ruibo Liu, Sine Liu, Shawn Guo, Soren Gao, Wangchunshu Zhou, Xinyue Zhang, Yizhi Zhou, Yubo Wang, Yuelin Bai, Yuhan Zhang, Yuxiang Zhang, Zenith Wang, Zhenzhu Yang, Zijian Zhao, Jiajun Zhang, Wanli Ouyang, Wenhao Huang, and Wenhu Chen. Map-neo: Highly capable and transparent bilingual large language model series, 2024.

[86] Harry Dong, Beidi Chen, and Yuejie Chi. Prompt-prompted adaptive structured pruning for efficient llm generation, 2024.

[87] Dmitry Scherbakov, Nina Hubig, Vinita Jansari, Alexander Bakumenko, and Leslie A. Lenert. The emergence of large language models (llm) as a tool in literature reviews: an llm automated systematic review, 2024.

[88] Junmo Kang, Leonid Karlinsky, Hongyin Luo, Zhen Wang, Jacob Hansen, James Glass, David Cox, Rameswar Panda, Rogerio Feris, and Alan Ritter. Self-moe: Towards compositional large language models with self-specialized experts, 2024.

[89] Yuncheng Yang, Yulei Qin, Tong Wu, Zihan Xu, Gang Li, Pengcheng Guo, Hang Shao, Yuchen Shi, Ke Li, Xing Sun, Jie Yang, and Yun Gu. Leveraging open knowledge for advancing task expertise in large language models, 2024.

[90] Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models, 2024.

[91] Zongyue Qin, Chen Luo, Zhengyang Wang, Haoming Jiang, and Yizhou Sun. Relational database augmented large language model, 2024.

[92] Jiawei Chen, Yue Jiang, Dingkang Yang, Mingcheng Li, Jinjie Wei, Ziyun Qian, and Lihua Zhang. Can llms' tuning methods work in medical multimodal domain?, 2024.

[93] Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. Mindmerger: Efficient boosting llm reasoning in non-english languages, 2024.

[94] Maohao Shen, Shun Zhang, Jilong Wu, Zhiping Xiu, Ehab AlBadawy, Yiting Lu, Mike Seltzer, and Qing He. Get large language models ready to speak: A late-fusion approach for speech generation, 2024.

[95] Ruyi Gan, Ziwei Wu, Renliang Sun, Junyu Lu, Xiaojun Wu, Dixiang Zhang, Kunhao Pan, Junqing He, Yuanhe Tian, Ping Yang, Qi Yang, Hao Wang, Jiaxing Zhang, and Yan Song. Ziya2: Data-centric learning is all llms need, 2024.

[96] Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cu. Lifelong language pretraining with distribution-specialized experts, 2023.

[97] Zilun Zhang, Yutao Sun, Tiancheng Zhao, Leigang Sha, Ruochen Xu, Kyusong Lee, and Jianwei Yin. Preserving knowledge in large language model with model-agnostic self-decompression, 2024.

[98] Xin Zhang, Tianjie Ju, Huijia Liang, Ying Fu, and Qin Zhang. General llms as instructors for domain-specific llms: A sequential fusion method to integrate extraction and editing, 2024.

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.