ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

A SYSTEMATIC REVIEW OF ENERGY EFFICIENT DEPLOYMENT STRATEGIES FOR LARGE LANGUAGE MODELS IN REAL WORLD APPLICATIONS

Sweta Kumari[1], Dr. Shashank Swami[2]
[1]*MTech Student, Vikrant University Gwalior, MP.*
[2]*Assistant Professor, Department of Computer Science & Engineering, Vikrant University, Gwalior MP*

Article DOI: https://doi.org/10.36713/epra26156
DOI No: 10.36713/epra26156

ABSTRACT

The widespread use of Large Language Models (LLMs) in practice has become an issue of mounting concern regarding energy usage and sustainability, as inference loads rapidly begin to comprise a larger portion of the total cost of the AI systems. It is a review of the literature published on energy-efficient deployment strategies of LLMs addressing the energy challenges at the deployment stage, methods of optimizing models at the deployment stage, and system-level and hardware-aware serving strategies. The inference-focused workloads, heterogeneity in the infrastructure between the cloud, edge, and on-device, and overheads at the system level are discussed as the main contributors to deployment-time energy wastefulness. The review also evaluates the use of model compression, parameter-efficient adaptation, conditional computation, adaptive serving models, and accelerator-aware execution to reduce energy expenditure within real-world restrictions. In general, the article points to the necessity of more coherent and comprehensive deployment models and universalized energy assessment procedures as the only way to make LLMs usable in large-scale production settings in a scalable and sustainable manner.

KEYWORDS: *Large Language Models; Energy-Efficient Deployment; Inference Optimization; Green AI; Edge–Cloud Computing*

1. INTRODUCTION

The fast commercialization of Large Language Models (LLMs) into real-world uses, e.g. conversational agents, decision-support systems, enterprise automation, education, healthcare, and edge-based intelligent services, has fundamentally altered how intelligent systems are applied at scale, but this mass adoption has also shown a major sustainability issue of increasing energy consumption during model inference and deployment. Although the research conducted to date has mostly looked at the computational and environmental cost of training large models, the deployment phase contributes a larger portion of the total energy footprint to continued inference requirements, stringent latency, mixed infrastructure environments, and a requirement to provide services that are consistently available and reliable [1]. The increasing size and ability of LLMs create practical limits to their scalability, cost-efficiency, and environmental responsibility, especially in cloud-edge architectures and resource-constrained systems. Following these new issues, this systematic review paper seeks to consolidate and critically review the state of art in energy-efficient deployment strategies of LLMs with particular objectives that will include:

- Delimiting and classifying the main energy consumption sources in the actual deployment of LLM to clouds, edges, and on-devices conditions;
- Surveying and streamlining model-level optimization methods like compression, parameter-saving adaptation, and conditional computation to lower the cost of inference time energy;
- Analyzing hardware and system-level deployment strategies such as runtime optimization, serving models, accelerator-aware execution, and edge cloud co-design models; and
- Shedding light on transparent problems, trade-offs, and future research avenues on developing scalable, energy-aware pipelines deploying LLM.

The review aims to offer a systematic basis upon which researchers and practitioners can develop an architecture of LLM deployment into real-world working environments by unifying scattered evidence on machine learning systems, green AI, and deployment engineering literature.

2. ENERGY CHALLENGES IN REAL-WORLD DEPLOYMENT OF LLMS

While much focus has been on the energy and carbon costs of training large language models, the phase of deployment now takes an even bigger portion of their lifetime energy footprint. In contrast to training, which is a one-time or periodical procedure, a real-world deployment is a continuous inference and variable workload, hard latency, and in heterogeneous infrastructural environments. Scaling LLMs presents a unique range of energy challenges, which arise due to the complexity of the model and the system-level constraints as well as the specifics of real-life applications. The realization of these issues is paramount in developing deployment pipes that are computationally efficient and environmentally friendly.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

2.1. Inference-Centric Energy Consumption and Model Complexity

In the world of production, the energy footprint of inference can be significantly greater than that of model training due to the fact that deployed large language models are scaled at rate and must be online 24/7. In contrast to training which is epidemic, inference is a continuing workload which increases with size of the users and the magnitude of the application. The architectural features of LLMs, that is, the number of parameters is large, the number of transformer layers is deep and the attention mechanisms have a quadratic dependence on the length of input sequences, lead to a high per-query energy consumption. The latter is enhanced by the fact that token-by-token autoregressive generation promises sequential computation and restricts the possibility of executing it in parallel [2]. Therefore, long-context reasoning, document summarization, text analysis in legal or clinical context, and conversational agents with language memory have disproportionately high energy expenses because both incremental token generation and the complexity of computations increase with each incremental token generation.

Low-latency response demanded by real-time systems further limits the ability to operate on a energy-efficient basis since requirements to maintain response-time guarantees diminish the effectiveness of aggressive batching and asynchronous request service, which make the overall utilization of the hardware less efficient and heightens the amount of energy wasted in every query. In real-world deployment to do so, this difficulty can be experienced by the confluence of large model sizes, lengthy input contexts and real-time generation demands, which together exacerbate the computational load and constraint system-level optimization. With the trend of LLMs becoming more multimodalized, tool-augmented, and operating longer chain-of-thought inference, the energy burden of inference time should grow instead of decreasing [3]. The path that it follows highlights the view that model complexity is not only a training-time issue but a key factor in deployment inefficiency that requires specific optimization mechanisms to be explicitly considered, which explicitly considers the energy-performance trade-offs of real-world inference workloads.

2.2. Infrastructure Heterogeneity: Cloud, Edge, and On-Device Constraints

The use of large language models cuts across heterogeneous systems such as cloud data centers, enterprise servers, edge platforms, and on-device systems each with unique limitations on energy availability, memory capacity, thermal dissipation, and latency tolerance that define the viability of deployment strategies. Cloud computing facilities offer scaleable computing and storage to meet large models and high throughput inference but come with serious sustainability issues (increasing cost of operation, cooling, and carbon emissions given continuously high-intensity workloads). By comparison, edge and on-device deployments have much more severe power, battery, and thermal constraints, making broad scale LLM deployment either architecturally simplified, model-thinned, or selective offloading to backend servers impractical [4]. The diversity of deployment environments poses a challenge to the creation of general purpose energy optimization strategies since strategies that work in the context of data centers, including large-batch processing and centralized serving, might not be applicable in resource-constrained and latency-sensitive edge settings. This means that working deployment pipelines should be built with adaptive and context sensitive strategies to fit model size, model accuracy, model execution locality and serving mechanisms to infrastructural properties, otherwise energy optimization work will be viewed as a compartmented and narrow spectrum of real world impact.

2.3. System-Level Overheads and Operational Inefficiencies

On top of the computational complexity of the model itself a significant fraction of the energy used to run real world LLM systems comes from system-level overheads incurred in contemporary inference pipelines. When it comes to real-world implementations, model execution is not the only energy that is used since there are other operational parts as well, such as.

- Network communication overhead related with transferring massive input prompt and generated output between distributed services,
- Model loading and memory management overheads incurred in the initialisation of containers and reco-initiating its service,
- Orchestration cost due to scaling mechanisms that commonly spin up and spin down instances in reaction to the varying workload, and
- Reliability mechanisms including redundancy, replication, and failover which demand having many model replicas to guarantee service continuity.

These auxiliary processes are usually running continuously in the background, creating a constant baseline of energy cost which is not very sensitive to inference demand at any given time.

Besides these structural overheads, during runtime inefficiencies also reduce the energy proportionality of deployed systems. Poor scheduling of requests, under-utilisation of hardware resources during periods of low loads, and poor batching are some of the issues that may lead to the occurrence of an energy consumption that is not proportional to the intensity of workload thus reducing the efficiency gains of optimized models. In real serving conditions System designers have to deal with a complex interaction of communication overheads, orchestration costs, reliability provisioning, and workload scheduling constraints in the real serving environment, all of which significantly contribute to the total energy footprint of LLCM services. This discussion has clearly shown that energy efficiency in the implementation of LLM is not dictated entirely by the model architecture or the number of parameters used in it but is critically influenced by the choices in system design and the practice of runtime management which controls the provisioning, coordination and utilization of computational resources when they are operated in a dynamical state.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

3. MODEL-LEVEL OPTIMIZATION STRATEGIES

Model-level optimization is the most direct and immediate type of intervention to minimise the energy and compute footprint of large language models in deployment. The aim of these strategies is to redesign, compress or adapt model architectures and parameters so that inference can be less resource-intensive without compromising reasonable levels of task performance [5]. Model-level methods have so far been applied to modify the internal structure or parameterization of LLMs instead of operating ex post at the system level, to affect memory consumption, computational complexity, and energy consumption at the origin. Such inherent efficiency gains are frequently a precondition to viable practicability in real world settings that have finite resources in energy, such as a deployed edge and hybrid cloud.

3.1. Model Compression and Parameter-Efficient Adaptation

A key direction of recent research on energy-efficient deployment of large language models has been compression and parameter-efficient adaptation methods aimed at directly minimising the memory footprint and arithmetic intensity of the inference workloads, which are the key contributors to the energy consumption of a deployment in resource-constrained settings.

Pruning techniques are meant to detect and remove redundant parameters, neurons, attention heads or even whole blocks of architecture that add small amounts to predictive performance, with the structured method of pruning being especially appealing to deployment systems since it produces sparsity patterns in hardware-friendly forms, which can be transformed into quantifiable speedups to inference and energy-saving, though the aggressive approach can result in loss of performance or even instability of generative tasks that can depend on distributed representations across layers [6]. Quantization also reduces computational and memory requirements by reducing the numerical precision of model parameters and activations, which may be executed more quickly with reduced energy consumption on modern low-bit-optimized accelerators, but excessive selective impairment of representational capacity can limit the degree to which quantization can be used, at the risk of functional reliability [7].

Knowledge distillation offers an alternative avenue to efficiency, transferring representational capacity and task-specific proficiencies of large teacher models to small student architectures, which can be deployed with minimal energy to domain-specific applications in which frontier-scale model capacity is not required, but the distillation process can cause inductive biases on the student. Fine-tuning methods that are more parameter-efficient, such as adapters, prefix tuning and low-rank adaptation, further specialise the deployment landscape by allowing task adaptation without changes or storage of the overall parameter set of a base model, allowing it to be reused across tasks and resulting in management overheads in multi-task serving environments as modules accumulate.

3.2. Architectural Innovations and Conditional Computation

In addition to direct changes in parameters, the increasing literature has covered architectural solutions that can enhance efficiency in deployment by adding conditional computation and selective activation control to large language models. These methods aim to separate model capacity and inference-time cost so that systems can be expressively powerful, with only a sub-network of computational pathways becoming active at any given time following an input [8]. This kind of rethinking of architecture is especially necessary in real world deployment environments, where long-context reasoning, multimodal input, and dynamic user loads otherwise can become prohibitively expensive in terms of energy consumption. Conditional computation frameworks would treat the model as a monolithic computational unit that needs to be executed entirely when responding to each input, in contrast to conditionally executing the unit in response to informational needs of each input instance.

- **Sparse and Structured Attention Mechanisms.**

Attention in conventional transformers grows quadratically with input length, and therefore long-context processing is one of the biggest consumers of inference-time energy. The sparse and structured attention mechanisms solve this bottleneck by limiting attention computations to a set of relevant tokens, based on locality, learning patterns, or sparsity structure patterns. These techniques enable important energy savings to long-document, multi-turn dialogue, and retrieval-intensive applications by lowering the computational complexity, as well as the overhead of memory access, of the interactions that need to be considered in the token scheduler. Nevertheless, the construction of efficient sparsity patterns is still task-specific and attention limitations which are not effectively selected can reduce levels of generative coherence and contextual comprehension.

- **Mixture-of-Experts and Conditional Routing Architectures.**

Mixture-of-experts (MoE) models are generalizations of the principle of conditional computation in which each input is dynamically routed to a small set of specialized sub-networks, or experts, as opposed to activating the entire parameter set during each inference pass. The selective activation paradigm is a scaling parameter that allows models to be scaled in number of parameters without a corresponding proportional increase in the inference-time energy cost since not all experts are involved in the computation of each request [9]. Deployment MoE architectures have provided a favorable trade-off between model capacity and operational efficiency especially in cloud environments where experts who may be different can be dispersed on hardware resources. However, the routing mechanism also adds complexity to system design, load balancing and memory management and imbalanced use of experts may result in unproductive inefficiencies that partially counteract the desired energy efficiency.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

- **Retrieval-Augmented and Hybrid Neuro-Symbolic Frameworks.**

A second architectural change in this direction is retrieval-augmented generation, which minimizes energy consumption by eliminating the large internal parameter stores and making use of external knowledge bases or document repositories. Smaller models can perform with the level of inference time computation required by much larger standalone LLMs by on-demand retrieval of task-relevant content, thus reducing the amount of computation required at inference time. Such a solution has effectively relieved the model of some of the burden of knowledge to an external memory system and allowed smaller architectures to perform very well in real-world applications as in question answering, enterprise search, and decision support. The retrieval pipelines however add further overhead in the system level via indexing, search latency and network communication and the net energy advantage presupposes the efficiency of the retrieval subsystem and how often external knowledge has to be accessed.

- **Adaptive Computation and Early-Exit Mechanisms.**

Adaptive computation methods also increase the efficiency of deploying models since they permit ad hoc models to conclude inference when an input can be resolved with high confidence using partial computation. Using early-exit mechanisms, the number of layers that are actually run in response to each request may be minimized by allowing there to be a layer that generates results when enough confidence has been attained. This is especially useful in high throughput, real-time applications that have a high percentage of possible queries that are routine or low-complexity. On the one hand, early-exit strategies can enormously minimize average energy usage, but they must be sensitive to ensure they do not over correctly decrease the performance of more complex or uncertain inputs, potentially requiring complete-depth computation.

4. SYSTEM-LEVEL AND HARDWARE-AWARE DEPLOYMENT STRATEGIES

While model-level optimizations can decrease the intrinsic computational cost of large language models, large language models can often be deployed with substantial savings in energy consumption in practice at both system and hardware levels, where inference workloads are coordinated, scheduled, and run on a heterogeneous collection of computing devices. In production, the serving structures, runtime administration choices and hardware usage patterns have a significant impact on the effectiveness of the LLM implementation, which jointly identify the effectiveness with which computational assets are allocated to changing workloads. In contrast to offline inference that is static, real-life LLM services need to support bursty traffic, a high level of latency constraints and reliability and as such, system-level design decisions take center stage in energy-conscious deployment. As a result, energy efficiency does not come out of small models but rather of an overall optimization of the entire serving stack, of request processing and execution pipelines as well as hardware configuration and infrastructure provisioning.

Table 1 provides a comparable overview of the representative literature covering system-level and hardware-aware deployment strategies of energy-efficient inference of large language models. The chosen works demonstrate the impact of runtime serving frameworks, hardware-adaptable scheduling, elastic scaling, and distributed inference architectures towards smaller energy usage of deployed LLMs in both cloud and edge environments.

Table 1: Comparative Summary of System-Level and Hardware-Aware Deployment Strategies for Energy-Efficient LLM Inference

Study	Hardware-Aware Contributions	Key Energy-Efficiency Insights	Relevance to This Review
Fernandez et al., 2025 [10]	Analysis of serving pipelines, batching strategies, and inference optimizations across hardware platforms	Showed that inference-time energy dominates lifecycle cost and can be reduced via batching and precision-aware serving	Supports system-level optimization and runtime energy management
Xie & Fang, 2025 [11]	Hardware-aware scheduling and lightweight inference framework for low-power devices	Demonstrated feasibility of generative AI on constrained hardware with energy-aware execution	Justifies edge–cloud collaboration strategies
Husom et al., 2025 [12]	Evaluation of quantized LLMs on edge hardware considering energy, latency, and accuracy trade-offs	Showed quantization significantly reduces energy consumption with moderate accuracy loss	Supports hardware-aware precision scaling
Zhu et al., 2025 [13]	Disaggregated expert parallelism and scalable serving architecture	Improved throughput-per-watt for MoE inference via expert disaggregation	Demonstrates system-level co-design for scalable, energy-efficient inference
Singh et al., 2025 [14]	Elastic scaling mechanisms for dynamic workload management	Reduced idle resource energy waste through adaptive scaling of experts	Supports load-aware scaling strategies
Behdin et al., 2025 [15]	End-to-end system optimization including hardware utilization and deployment pipelines	Highlighted real-world energy and cost savings through deployment-aware optimization	Grounds the review in real-world production settings

Recent system-level work has highlighted batching and precision-conscious serving pipelines (Fernandez et al., 2025), hardware-aware inference systems on low-power hardware (Xie and Fang, 2025), and elastic scaling mechanisms with mixture-of-experts models (Singh et al., 2025) as practical work

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

to enhance throughput-per-watt in practice in the deployment of the LLMs of interest.

4.1. Runtime Optimization, Hardware Utilization, and Edge–Cloud Co-Design

An array of optimization strategies at the runtime level can attempt to minimize the unnecessary computation and enhance the use of resources when inference is performed. Dynamic batching and adaptive request scheduling combine inference requests such that the use of accelerators is maximized without violation of latency requirements, which enhances throughput-per-watt at variable workloads. Even more interaction-based applications with repeatable query patterns can be used through caching schemes which store intermediate representation or common access embeddings to cut down on repeated computation. Strategies of early-exit, that enable part of the model to be run using low-complexity inputs, are used to help reduce the average energy consumption by preventing the unnecessary running of the full-depth inference. Simultaneously, the adaptive precision that is being served by the frameworks allow the dynamic adjustment of the numerical precision based on the workload characteristic and the accuracy requirement so that energy could be saved in the periods when the demand is low, or the approximate inference is tolerated.

Hardware-conscious deployment strategies further enhance these system-level benefits by matching model execution with the performance of dedicated accelerators including GPUs, TPUs and newer AI-specific chips designed to execute low-precision arithmetic and memory-efficient code. Compiler-level optimizations, kernel fusion, and memory layout transformations minimize data movement and synchronization overheads which are frequently the most significant sources of energy expenditure in large-scale inference pipelines. The architectural decisions are based on the properties of target accelerators, in the co-design of models and hardware, which further improves throughput-per-watt and energy inefficiencies caused by the mismatch between compute patterns and target accelerator hardware characteristics. Edge-cloud collaboration paradigms in instruments of distributed and hybrid deployment Teams: Edge-cloud collaboration paradigms encompass a pragmatic approach to energy-consumption, latency, and privacy tradeoffs in which lightweight pre-processing or partial inference is done at the edge, and heavyweight reasoning is done on cloud computations. The hybrid execution models minimize the needless data transmission and allows energy-aware allocation of workloads among heterogeneous resources. Together, these system level and hardware-conscious approaches support the perspective that the effective deployment of LLM is an end to end optimization problem, and needs a coordinated design at software frameworks, runtime management policies, hardware architecture, and at the infrastructure level provisioning, instead of individual improvements at a single level.

5. CONCLUSION

This review has identified the problem of energy-efficient implementation of large language models as a pressing issue because the inference overhead continues to grow to represent a significant portion of the total energy usage of real-world AI systems. The reviewed literature demonstrates that significant efficiency improvements come through joint implementation of the model-level optimizations, system-level runtime strategies, and hardware-aware deployment, but not through unilateral improvements in one of the layers. Although compression, conditional computation, adaptive serving, and accelerator-aware execution are effective techniques that can significantly lower the cost of deployment time in terms of energy, they need to be integrated with application needs and infrastructure limitations. In general, the results discuss the necessity to have holistic, end-to-end implementation models and uniform energy appraisal practices to facilitate the scalable and sustainable introduction of LLMs into practical settings.

REFERENCES

1. *Agrawal, R., Kumar, H., & Lnu, S. R. (2025, March). Efficient llms for edge devices: Pruning, quantization, and distillation techniques. In 2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS) (pp. 1413-1418). IEEE.*
2. *Behdin, K., Dai, Y., Fatahibaarzi, A., Gupta, A., Song, Q., Tang, S., ... & Simon, L. (2025). Efficient AI in Practice: Training and Deployment of Efficient LLMs for Industry Applications. arXiv preprint arXiv:2502.14305.*
3. *Cai, G., Tian, R., Yang, L., Jia, Y., Li, L., & Wang, J. (2026). Efficient Inference for Edge Large Language Models: A Survey. Tsinghua Science and Technology, 31(3), 1365-1380.*
4. *Fernandez, J., Na, C., Tiwari, V., Bisk, Y., Luccioni, S., & Strubell, E. (2025). Energy considerations of large language model inference and efficiency optimizations. arXiv preprint arXiv:2504.17674.*
5. *Ghosh, S. K., Raha, A., & Raghunathan, V. (2023). Energy-efficient approximate edge inference systems. ACM Transactions on Embedded Computing Systems, 22(4), 1-50.*
6. *Husom, E. J., Goknil, A., Astekin, M., Shar, L. K., KÃ¥ sen, A., Sen, S., ... & Soylu, A. (2025). Sustainable llm inference for edge ai: Evaluating quantized llms for energy efficiency, output accuracy, and inference latency. ACM Transactions on Internet of Things, 6(4), 1-35.*
7. *Husom, E. J., Goknil, A., Astekin, M., Shar, L. K., KÃ¥ sen, A., Sen, S., ... & Soylu, A. (2025). Sustainable llm inference for edge ai: Evaluating quantized llms for energy efficiency, output accuracy, and inference latency. ACM Transactions on Internet of Things, 6(4), 1-35.*
8. *Khan, S., Naz, N. S., Mazhar, T., Tariq, M. U., Shahzad, T., Guizani, S., & Hamam, H. (2025). Green AI techniques for reducing energy consumption in AI systems. Array, 100652.*
9. *Khan, T., Motie, S., Kocak, S. A., & Raza, S. (2025). Optimizing Large Language Models: Metrics, Energy Efficiency, and Case Study Insights. arXiv preprint arXiv:2504.06307.*
10. *Kristiani, E., Verma, V. K., & Yang, C. T. (2026). Deploying LLM Transformer on Edge Computing Devices: A Survey of Strategies, Challenges, and Future Directions. AI, 7(1), 15.*
11. *Singh, G., Yu, T., Li, H., Chen, C., Sadri, H., Zhang, Q., ... & Fan, Z. (2025). ElasticMoE: An Efficient Auto Scaling Method for Mixture-of-Experts Models. arXiv preprint arXiv:2510.02613.*
12. *Wang, R., Gao, Z., Zhang, L., Yue, S., & Gao, Z. (2025). Empowering large language models to edge intelligence: A survey of edge efficient LLMs and techniques. Computer Science Review, 57, 100755.*

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 2| February2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

13. *Whitmore, J., Hastings, C., Patel, A., & Brody, S. (2025). Efficient Inference of Large Language Models through Model Compression.*
14. *Xie, Y., & Fang, Q. (2025). An energy-aware generative AI edge inference framework for low-power IoT devices. Electronics, 14(20), 4086.*
15. *Zhu, R., Jiang, Z., Jin, C., Wu, P., Stuardo, C. A., Wang, D., ... & Liu, X. (2025). MegaScale-Infer: Serving Mixture-of-Experts at Scale with Disaggregated Expert Parallelism. arXiv preprint arXiv:2504.02263.*