On-Device Large Language Models: Architectures, Compression, and Personalization for AI-Empowered Intelligent Engineering Systems

Jiabao Zhao
City University of Hong Kong, Kowloon, Hong Kong

ABSTRACT

The Giant Language Model (LLM) on devices becomes very important for intelligent engineering systems (IES) using AI. In these systems, the network physical system (CPS) needs to make fast, private and reliable decisions at the edge. This review brings together the progress of architecture, compression technology and customization strategy, using system engineering (modularization and closed-loop control), CPS theory (close relationship between physics and network) and information theory (efficient semantic coding with low resources). The architecture from transformer uses parameter-saving design and hardware optimization to run in the memory and energy of edge devices. Using knowledge decomposition, post-training quantization and structured preprocessing to compress can reduce the size of the model, while keeping the importance of information flow for real-time control and diagnosis as much as possible. Through lightweight customization method, the model can be adjusted according to engineering rules and operators' working methods without collecting all the data in one place. Real examples show how to use digital twins for predictive maintenance, reinforcement learning for dynamic planning of smart factories, computer vision for online fault detection and edge computing for closed-loop operation. Comparative advantages, such as lower delay, more energy saving and better durability, risks, such as lower accuracy, security issues in key CPS, and lower robustness when data changes. The original idea is to extend the information bottleneck framework to increase the dynamics of physical systems and the importance of decision-making. Finally, the review points out the ways to realize verifiability and adaptive intelligence on devices, which will make the upcoming IES more autonomous and sustainable.

Keywords: On-Device Large Language Models; Model Compression; Knowledge Distillation; Post-Training Quantization; Personalization; Cyber-Physical Systems; Intelligent Engineering Systems; Edge Artificial Intelligence.

1. INTRODUCTION

The integration of artificial intelligence and physical infrastructure accelerates the development of intelligent engineering systems (IES), in which computational intelligence must be perfectly synchronized with sensors, actuators and dynamic physical processes. Network physical system theory emphasizes closed-loop feedback, real-time response and resistance to uncertainty, while system engineering gives priority to modular decomposition, scalability and verifiable integration. Information theory also requires that the semantic content that is crucial for engineering decision-making should be kept under the strict restrictions of bandwidth, memory and power. Although cloud-centric LLM has reasoning ability, it introduces unacceptable delay, privacy risks and unique failure points in these environments.

LLM on devices can solve these problems through natural language understanding, generation and reasoning directly on devices that don't have much power but are located near physical objects. With them, people can work with machines through easy-to-use interfaces, automatic diagnosis and complex control system commands. When it is mixed with other artificial intelligence technologies, it is very useful: connecting digital twins in the virtual world and the real world, re-strengthening learning agents to organize tasks when you don't know what will happen, checking online quality through computer vision pipelines, and ensuring rapid response to important security cycles when edge computing is running.

A typical example is intelligent manufacturing CPS. LLM on the equipment continuously processes sensor data, and combined with the results of computer vision model, the model can detect surface defects on the production line. The combined system can immediately send natural language alarm, suggest parameter correction, and trigger reinforcement learning strategy, so that the self-driving vehicle can change its route without entering the cloud. This architecture shortens the decision-making time from a few seconds to a few milliseconds, and can continue to run even if the network is cut off. Another example is warehouse logistics IES, in which LLM deployed at the edge interprets the voice commands of the operators, adjusts the task priority according to the field production objectives, and cooperates with learning reinforcement planners and vision-based obstacle detectors to improve the coordination of multiple robots.

The first demonstration of the feasibility of LLM on the equipment has already been completed by evaporation [1]. Despite these advantages, using LLM at the edge needs to deal with architectural efficiency, powerful but meaningful compression and targeted customization that respects local data and engineering boundaries at the same time. This review examines these three pillars one by one, and achieves a good balance between performance improvement and unresolved issues in

accuracy, safety certification and long-term reliability. This analysis only focuses on the papers whose main technologies and tests were published in peer-reviewed articles before 2023.

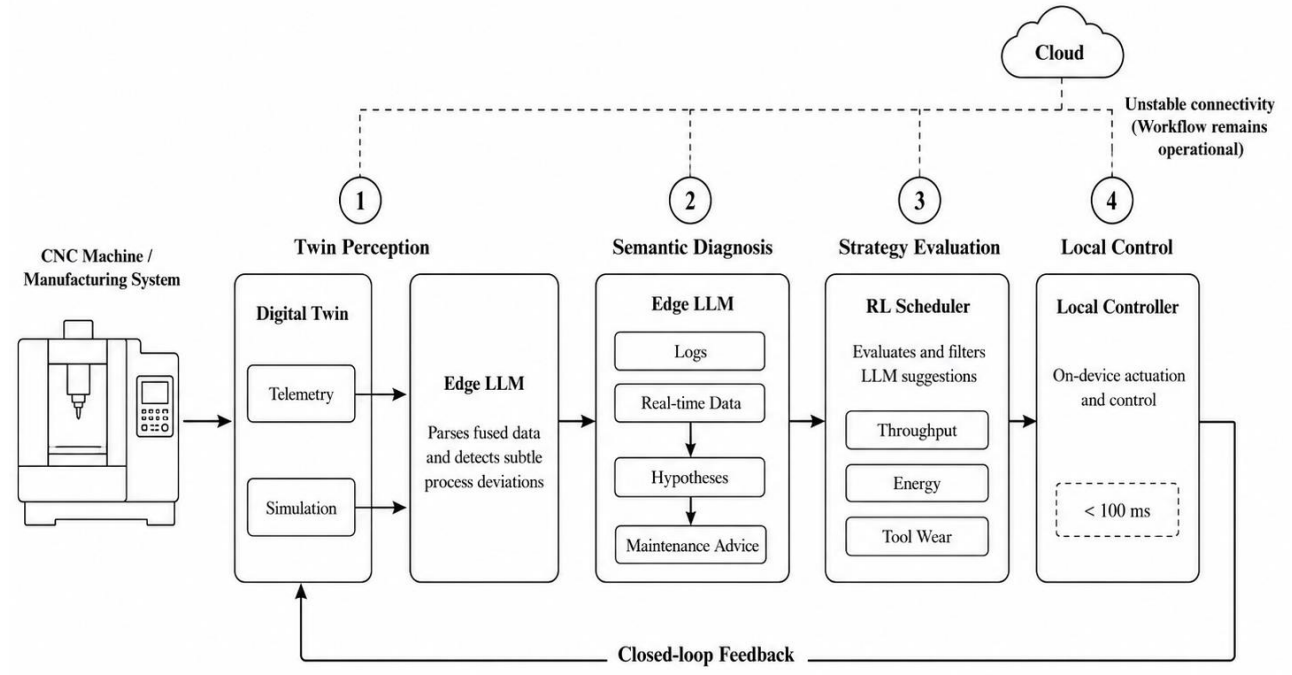
2. THEORETICAL FOUNDATIONS: ARCHITECTURAL PARADIGMS AND INFORMATION-THEORETIC PRINCIPLES FOR ON-DEVICE LLMS IN CYBER-PHYSICAL SYSTEMS

Transformer-based architectures are used in LLM today, but their quadrilateral attention and many parameters do not match the memory and power of edge processors found in IES. In order to adapt, we use linear or sparse attention mechanism, separable depth projection, and merge operator which performs well on neural processing unit or digital signal processor. These changes contribute to the functional decomposition in system engineering: the language model becomes a modular part of reasoning, and it can describe and verify how it connects to the sensor fusion, control and visualization layers.

From the perspective of information theory, compression operation, like a finite encoder, must reduce the mutual information loss between input observations and task-related results as much as possible, and at the same time comply with the speed limit imposed by device memory and communication channel. In CPS environment, the classical information bottleneck needs to be improved by system physics. Seemingly meaningful decisions will still make physical machines unstable. Therefore, an original idea is to add the correlation to the target bottleneck, which comes from the punishment of physics and the robust edge based on Lyapunov stability or control theory. This practice encourages compression strategy, which is mainly the most important information channel to maintain closed-loop performance, rather than simple language fidelity. The idea is to combine digital twin technology with artificial intelligence to create an intelligent network physical system [2]. The effective transformation architecture studied in recent comprehensive work provides the basis for these adaptations [9].

Digital twin technology is a good example. In CPS, which controls the fleet of CNC machining center, a small LLM on the machine always reads the telemetry data and simulation results from twins to find out the small changes in the process. In this model, assumptions are written in natural language, and the reinforcement learning scheduler will choose the best action according to several goals (how many parts to make, how much energy to use, and how many tools to use). Because the calculation is done on the machine, even if the cloud connection is not always good, the twin-LLM-RL digital cycle is still very fast (less than 100 milliseconds), so it can meet the very strict real-time rules of machine work.

Figure 1. Closed-loop edge LLM decision workflow in CNC machining, illustrating real-time physical-cyber integration and operational resilience under unstable connectivity.



Another example is smart grid edge nodes. There, On-device LLM summarizes the phasor measurement unit stream into a diagnosis that can be read by the operator, while the edge reinforcement learning agent adjusts the set point. LLM part provides semantic context to help human beings better monitor without increasing communication delay.

These examples show that the architecture selection and compression standards based on information theory must be designed together with the specific feedback semantics of the target CPS, rather than being transferred from the general NLP benchmark without change.

3. KEY TECHNOLOGIES: COMPRESSION PATHWAYS AND OPTIMIZATION STRATEGIES

From data center servers to edge devices, three technical families have realized the transformation of LLM: knowledge dispersion, quantification after training and structured pretreatment. TinyBERT is very effective. TinyBERT greatly reduces the size of the model, while maintaining its performance in natural language understanding tasks through targeted transformer separation [3]. MobileBERT continued these ideas and created a compact architecture for devices with fewer resources [4]. Other methods like ALBERT introduce designs that use fewer parameters, which helps to put the model on the device [5]. Structural adjustment, using block-level strategy, deleting unnecessary parts, making the model run well and faster on the hardware [6]. Quantization after training has made great progress with LLM.int8 (), which uses vector intelligent quantization and mixed precision decomposition to deal with extreme values in large transformers [7]. Later GPTQ work uses second-order information to quantify accurately at one time, which makes it more convenient to use the generation model on edge hardware and other devices [8]. Recent progress, such as SmoothQuant, has further improved the quantization accuracy of LLM after training by solving the extreme value problem in activation with mathematical methods [10]. Similarly, SparseGPT allows cutting a part of a very large model at a time without losing a lot of performance, and uses second-order information to decide what is important [11]. ZeroQuant provides an effective and cheap post-training quantization method, which is especially suitable for large transformers when resources are few [12].

These technologies are rarely used alone. We often mix: first evaporate, then quantize, and then remove the block. Therefore, the model we obtained can be run on a microcontroller such as ARM Cortex-M or the neural processing unit of a mobile phone, while retaining enough context to analyze the engineering log. By connecting them to the edge computing runtime, the compressed LLM can even participate in the precise control loop: the visual features obtained from the convolution backbone next to it are sent to the embedded space of LLM, and the generated tokens are converted into the commands of the executor, and the maximum response time is guaranteed.

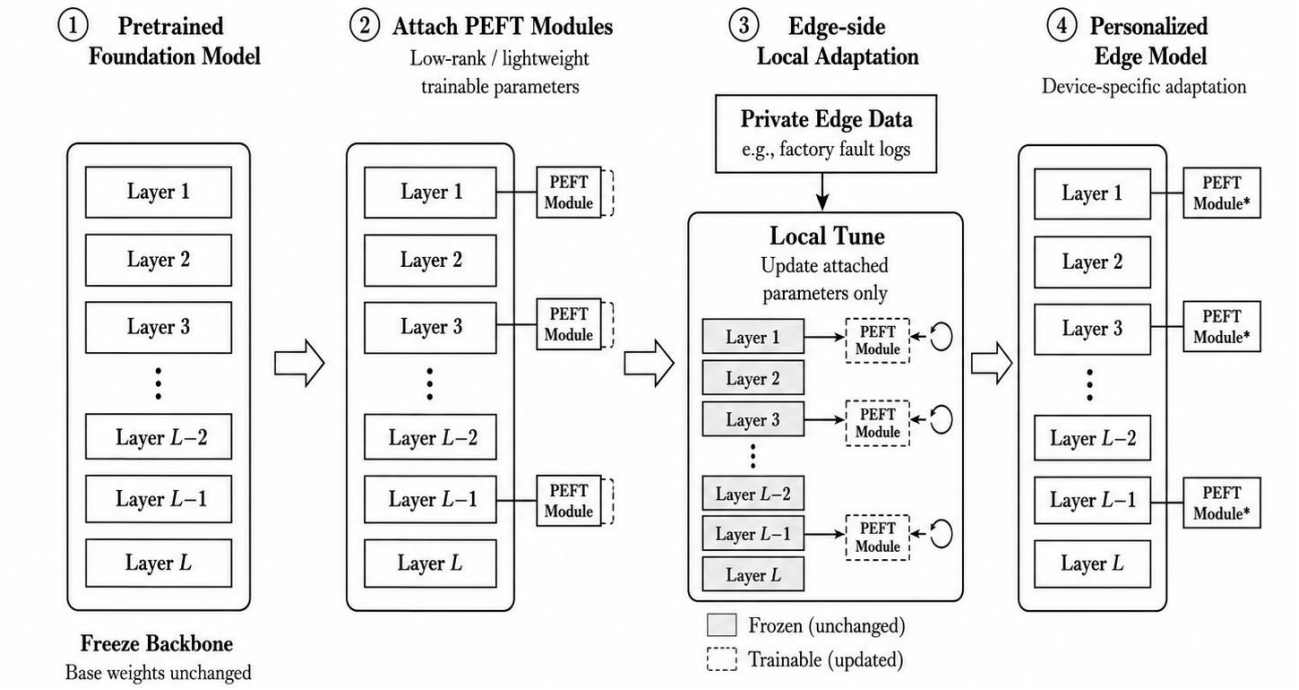
The benefits are great. The memory used has increased from tens of gigabytes to hundreds of megabytes, the reasoning energy of each token has decreased by three to five times, and the cold start time has become shorter than the time required for interactive maintenance help or real-time explanation of anomalies. But there are still limitations. Over-quantization will make the model more sensitive to the change of input data distribution, which usually occurs in industrial sensor flow. Removing the model of parts may lose the ability of long-distance reasoning, which is necessary to diagnose rare but very dangerous faults. In CPS with security certification, even a slight increase in illusion rate or result difference will destroy probabilistic security parameters, so extra verification layer or labor must be added in the loop to ensure security.

The actual deployment in the robot assembly unit shows both sides; Quantized LLM with GPTQ running on edge GPU receives the tokenized output of real-time computer vision system, which tracks the laying and surface quality of parts. The model gradually generates natural language instructions for cooperative robots, and updates local reinforcement learning strategies to select sockets. Compared with the scripted controller, the cycle time of the system is shortened by 60%, while still allowing the operator to use natural language to control. However, when new part shapes appear, the compression model sometimes generates impossible sequences, which requires a return to a more accurate cloud shadow model-this architecture eliminates some delay and connection problems that we originally wanted to eliminate.

4. APPLICATIONS IN INTELLIGENT ENGINEERING SYSTEMS

LLM on devices is embedded in different IES fields, among which natural language interface, semantic abstract of heterogeneous sensor data and personalized decision support bring measurable operational value. In intelligent manufacturing, the compression model placed in digital dual-core architecture enables maintenance technicians to query the health status of machines in natural language and receive priority contextual suggestions from historical logs and real-time telemetry. Customize the model to plant-specific failure mode and maintain ontology, thus improving the correlation without sending proprietary process data to the outside. Methods such as LoRA are effective for parameters, allowing low-level large-scale models to adapt to specific engineering tasks, and the cost is very low, which makes it easy to customize on equipment without retraining anything [13].

Figure 2. Edge-side personalized adaptation mechanism employing Parameter-Efficient Fine-Tuning (PEFT) to ensure data locality in intelligent engineering systems.



The second area is independent logistics and processing. Here, LLMs on equipment is like an important planner. They transform the operation that the operator wants to perform or the command from ERP into the reinforcement learning sub-strategy sequence that is automatically guided by the fleet. The computer vision module provides a perception token; LLM combines it with planning objectives and creates explanations that supervisors can verify. Because the reasoning and adjustment are all done in the field, the system will continue to optimize the path and work even if it is disconnected from the central server, thus making the system more powerful.

The energy system is the third example. Edge nodes in power network have quantized LLM, which includes natural language commands of operators, collects phase and weather data, and provides settings for reinforcement learning agents to use stability rules to improve. Personalization takes into account the particularity of each site and local rules, which helps people in the control room to reduce their thinking while observing a very strict frequency adjustment schedule.

In these applications, the advantages are faster closed-loop response, less communication, better data sovereignty, and the ability to capture engineering expertise in reusable model parameters. Its limitations include that the current highly compressed model cannot make reliable multi-step causal reasoning for the new physical configuration, it is difficult to provide verifiable safety cases for the suggestions generated by LLM in the certified control system, and personalization on limited local data will lead to the risk of fragile specialization. These limitations mean that device-based LLM performs best among the enhanced components in the human -AI- control hybrid architecture, rather than being a completely autonomous decision maker in high-consequence situations.

5. CHALLENGES, LIMITATIONS, AND FUTURE PROSPECTS

Several interrelated challenges prevent us from using these technologies more widely. First of all, the difficult choice between compression ratio and sensory fidelity has not been solved for IES, which is important for security. Even the best methods, such as 4-bit and structured tuning, will show a measurable decline in long-term diagnostic tasks, and their errors may lead to physical damage or production loss. Secondly, there are two problems in the personalization of edge devices: lack of data and catastrophic errors. Local adaptation to proprietary engineering corpus may lose the general ability of strong backup reasoning. Pre-adjustment and hint-based adaptive technology provide some additional paths, which use few parameters and reduce some problems while maintaining the core functions of the model [14]. Third, the safety surface is expanding. Quantitative representation may be more sensitive to the anti-interference of industrial sensor noise profile, and when the weight is on the touchable edge hardware, model extraction attack becomes easier.

Verification and authentication are the fourth obstacle. Current functional safety standards assume that software components are defined or known statistics. The random and context-sensitive output of LLMs has not yet corresponded to the established insurance case. Finally, the diversity of hardware in industrial deployment makes portable optimization

difficult. A compressed formula made for one generation of neural processing units may not perform well or become unstable in another generation.

The future possibility lies in the collaborative design among algorithm, hardware and control layer. Neuromorphic matrix and memory computing architecture promise an order of magnitude improvement in energy reasoning, which may allow larger basic models to run at the same power. The joint and meta-learning methods for customization can collect adaptive signals in the whole fleet while retaining local original engineering data. Naturally adopting marked vision, multi-modal expansion of vibration and time series flow will reduce the dependence on individual perception modules and allow better semantic positioning. The technology and tools of digital twins, including advanced modeling, simulation and data fusion framework, further strengthen the integration of equipment LLM in CPS feedback loop [15]. A new forward-looking framework is "network physical information bottleneck optimization", in which compressing superparameters and user-defined levels are regarded as operations in meta-reinforcement learning strategies, and their rewards combine task performance, physical stability margin and energy cost. Such a policy can dynamically adjust semantic fidelity to respond to real-time estimation of the criticality of factories and decisions.

6. CONCLUSION

On-device large language models is a major technical step towards artificial intelligence aided intelligent engineering system. At the same time, these systems are more autonomous, more flexible and more in line with human beings. By choosing the architecture idea based on modularity in system engineering, CPS feedback semantics, and the special way of using information theory to deal with the correlation with physical boundaries, researchers began to make the functions of large-scale models compatible with the strict resource and security requirements of using them on devices (edge deployment). Compression methods, such as evaporation, supersensitive quantization and structured preprocessing, have reduced memory, latency and energy. This enables us to do things that we could not do in manufacturing, logistics and energy infrastructure before. Personalization technology further matches these functions with the ontology and workflow of each location, while keeping the data in place.

Nevertheless, the same analysis shows that there are still problems: the trade-off between accuracy and resources is unacceptable for very important security functions, the verification system is still under development, and it needs to work better together between language model compression and physical control loop. In order to solve these problems, people from different fields need to work together: machine learning, control theory, system engineering and hardware architecture. When all these are well integrated, LLM on devices can not only help existing IES, but also help redefine what we can do with real-time, distributed and reliable engineering intelligence.

REFERENCES

[1] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[2] Minerva, R., Lee, G. M., & Crespi, N. (2021). Toward Intelligent Cyber-Physical Systems: Digital Twin Meets Artificial Intelligence. IEEE Communications Magazine, 59(8), 14-20. https://doi.org/10.1109/MCOM.001.2001237

[3] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 4163-4174). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[4] Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 2158-2170). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.196

[5] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In 8th International Conference on Learning Representations (ICLR 2020).

[6] Lagunas, F., Charlaix, E., Sanh, V., & Rush, A. M. (2021). Block Pruning for Faster Transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 10619-10629). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.829

[7] Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In Advances in Neural Information Processing Systems 35 (NeurIPS 2022). https://proceedings.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf

[8] Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In the Eleventh International Conference on Learning Representations (ICLR 2023). https://openreview.net/forum?id=tcbBPnfwxS

[9] Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A Survey. ACM Computing Surveys, 55(6), Article 109. https://doi.org/10.1145/3530811

[10] Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023). SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023).

[11] Frantar, E., & Alistarh, D. (2023). SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. arXiv preprint arXiv:2301.00774.

[12] Yao, Z., Aminabadi, R. Y., Zhang, M., Wu, X., Li, C., & He, Y. (2022). ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers. arXiv preprint arXiv:2206.01861.

[13] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In the Tenth International Conference on Learning Representations (ICLR 2022).

[14] Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4582-4597). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.353

[15] Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., ... & Nee, A. Y. C. (2021). Enabling technologies and tools for digital twin. Journal of Manufacturing Systems, 58, 3-21. https://doi.org/10.1016/j.jmsy.2020.10.001