上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Trusted LLM

Log Creative

June 11th, 2024

Contents

1 LLM

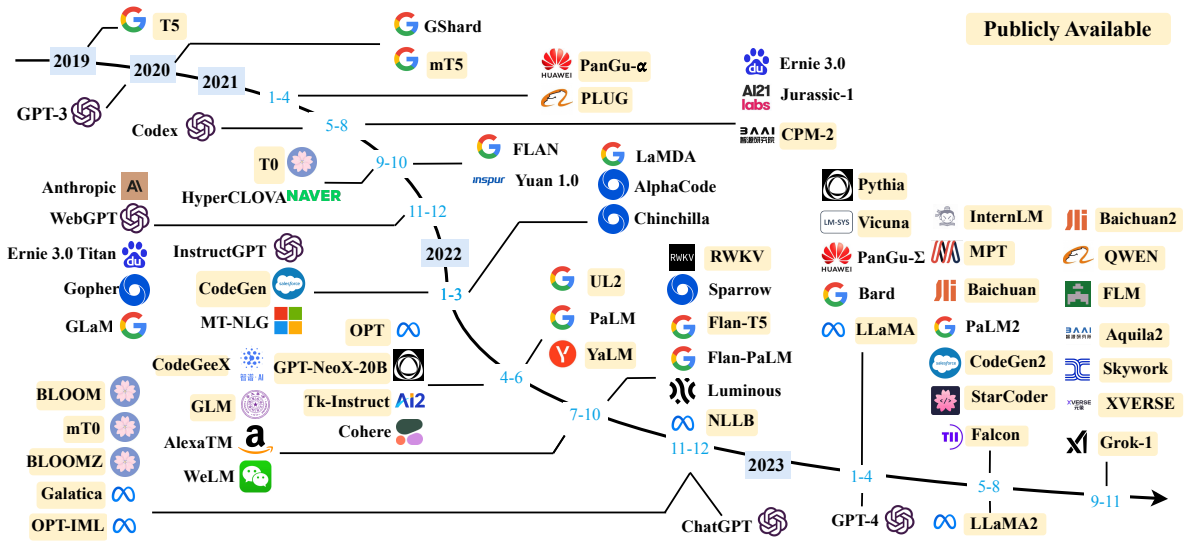


Figure 1 A timeline of existing large language models in recent years [1]

NLP. NLP (Natural Language Processing) can be categorized into the subfields of AI (Artificial Intelligence) and linguistics. NLP mainly explores how to make machines understand, process, reproduce, and generate natural languages [2].

LLM. Large Language Models (LLMs) refer to Transformer [3] language models that contain hundreds of billions (or more) of **parameters**, which are trained on massive text **data**. The advent of LLMs (Large Language Models) marks a significant milestone in NLP and generative AI [4], which have exhibited remarkable capabilities in tackling complex tasks, exemplified by their ability to engage in human-like conversations as demonstrated by ChatGPT [5]. Consequently, this has propelled significant research endeavors shown in Figure 1 with blooming research accomplishments across both the open-source models and commercial models.

Applications of LLM. LLMs are revolutionizing a multitude of industries by their ability to process and generate human-like text. They are instrumental in enhancing **NLP tasks** such as translation, summarization, and question-answering. In **healthcare**, LLMs expedite literature retrieval and analysis, aiding in drug discovery and genomics. They also contribute to **robotics** by facilitating natural language understanding for robots, and in the field of **computer science**, they assist in code generation and debugging. Moreover, LLMs are being **fine-tuned or adapted** for specific applications, leading to more personalized and accurate responses, e.g., Apple Intelligence [6] could apply LLMs to Siri to help with everyday tasks based on the personal context and fine-tuned or adapted models shown in Figure 2. The rapid advancements in LLM research are continuously expanding their applications, making them a pivotal tool in the progression of AI [7].

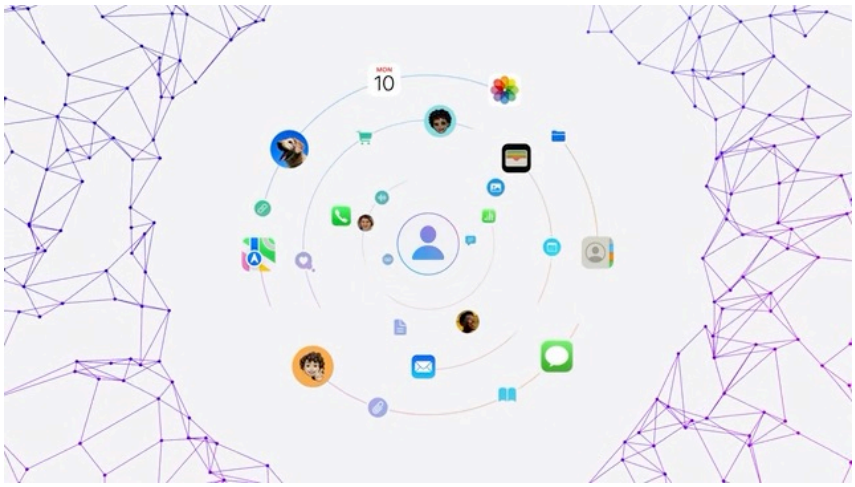


Figure 2 Apple Intelligence could understand the intent of the user on everyday tasks based on LLM

Origin of LLMs' intelligence. The outstanding capabilities of LLMs can be attributed to multiple factors [4], such as the usage of large-scale raw texts from the Web as **training data** (e.g., PaLM [8] was trained on a large dataset containing more than 700 billion tokens), the design of transformer architecture with **a**

large number of parameters (e.g., GPT-4 [9] is estimated to have in the range of 1 trillion parameters), and advanced training schemes that **accelerate the training process**, e.g., low-rank adaptation (LoRA) [10], quantized LoRA [11], and pathway systems [12]. Moreover, their outstanding instruction following capabilities can be primarily attributed to the implementation of **alignment with human preference**. Prevailing alignment methods use reinforcement learning from human feedback (RLHF) [5], which shapes the behavior of LLMs to more closely align with human preferences, thereby enhancing their utility and ensuring adherence to ethical considerations.

Trustworthy challenges of LLMs. However, the rise of LLMs also introduces concerns about their trustworthiness. Unlike traditional language models, LLMs possess unique characteristics that can potentially lead to trustworthiness issues. There are mainly 6 dimensions concerning the trustworthiness of LLMs [4]:

- **Truthfulness.** Truthfulness in AI systems refers to the accurate representation of information, facts, and results. LLMs tend to have hallucinations in a dialog where misinformation or outdated information could be provided, which is primarily due to the noise in the training data and the lack of generalization capability in the Transformer architecture.
- **Safety.** Safety in LLMs is crucial for avoiding unsafe or illegal outputs and ensuring engagement in healthy conversations. LLMs could be jailbroken by attacks to output knowledge that is illegal or toxic, which is due to the sophisticated prompt engineering and the lack of safety protocols under the hood.
- **Fairness.** Fairness is the ethical principle of ensuring that LLMs are designed, trained, and deployed in ways that do not lead to biased or discriminatory outcomes and that they treat all users and groups equitably. The biased information could be outputted due to data biases and private information in large training datasets.
- **Robustness.** Robustness is defined as a system's ability to maintain its performance level under various circumstances. This is mainly related to the deployment architecture of LLMs to make the throughput available most of the time to meet high user expectations.
- **Privacy.** Privacy encompasses the norms and practices aimed at protecting human autonomy, identity, and dignity. (1) For individuals, in the absence of stringent safeguards, sensitive personal information within the training dataset becomes susceptible to misuse, potentially leading to privacy breaches. This issue is especially acute in the healthcare sector, where maintaining the confidentiality of patient data is of utmost importance [13]. (2) For companies, private domains have accumulated substantial volumes of data through advancements in AI algorithms. Nevertheless, privacy concerns and commercial competition tend to isolate such data sources, hindering direct collaboration and knowledge sharing. What's more, real-world data privacy regulations frequently impose restrictions on direct data sharing among isolated entities, thereby exacerbating concerns relating to data scarcity and privacy protection.
- **Machine Ethics.** Machine ethics ensure the moral behaviors of man-made machines utilizing AI, commonly referred to as AI agents. This is mainly related to the ethical property of the training data.

As a result, **data privacy** and **correct execution** (output the desired information) are key challenges for trusted LLMs. LLM is of a great scale, as a result, the **efficiency** often matters.

2 Trusted LLMs

The next-generation internet, with its focus on trusted identity (entity authentication, identity management and system), trusted data (message authentication, blockchain and oracle), and trusted computation (multi-party computation, verifiable computing, smart contract and TEE-based computing), is expected to significantly influence the field of LLM to tackle the challenges mentioned in Section 1. This paper mainly focuses on **trusted computation** for LLM to preserve the privacy of data and prove the correctness of execution.

Data privacy. Addressing the challenge of utilizing private domain data for modeling purposes, while simultaneously upholding data privacy, is an issue of great significance. Existing solutions tackle this problem by employing privacy-preserving computation techniques. Three primary approaches to privacy-preserving computation are currently prevalent [13]: (1) cryptography-based methods that primarily focus on secure multi-party computation (SMPC); (2) confidential computing, which utilizes trusted hardware such as trusted execution environments (TEE); (3) federated learning (FL), a technique that integrates privacy-preserving measures into collaborative modeling. Section 2.1 and Section 2.2 will introduce the current state-of-the-art. Those methods are expected to be used widely to construct LLMs with more trustworthiness.

Correct execution. As laws and regulations around LLMs evolve and tighten, developing practical tools to verify the legitimacy of these models has become crucial. Section 2.3 will introduce the proving method based on zero-knowledge proof.

2.1 Sigma: LLM with efficient 2PC

2PC (2-party computation), which is a specific scenario of MPC (multi-party computation), could be described as follows. There are two parties P_0 and P_1 with inputs x_0 and x_1 and they wish to compute a public function $y = f(x_0, x_1)$ without revealing anything more than the function output y to each other. In a preprocessing phase that is independent of the inputs to the function x_0 and x_1, correlated randomness is generated and made available to P_0 and P_1.

While SMPC (secure multi-party computation) ensures robust security, it may not adequately address efficiency requirements. Sigma [14] is a system that advances the state-of-the-art for secure inference of transformer-based models along multiple dimensions. Like CrypTen [15], Sigma works in 2PC with preprocessing model and uses GPU acceleration, but is an order of magnitude **more efficient** in latency and communication while providing standard 2PC security guarantees. Sigma maintains the model accuracy under secure inference through precise approximations of complex non-linearities and scales efficiently to GPT models with billions of parameters.

Function secret sharing (FSS) [16] is a recent paradigm for obtaining efficient 2PC protocols with a preprocessing phase. Sigma is the first end-to-end system for secure transformer inference based on FSS. By constructing new FSS-based protocols for complex machine learning functionalities, such as Softmax, GeLU and SiLU, and also accelerating their computation on GPUs, Sigma improves the latency of secure inference

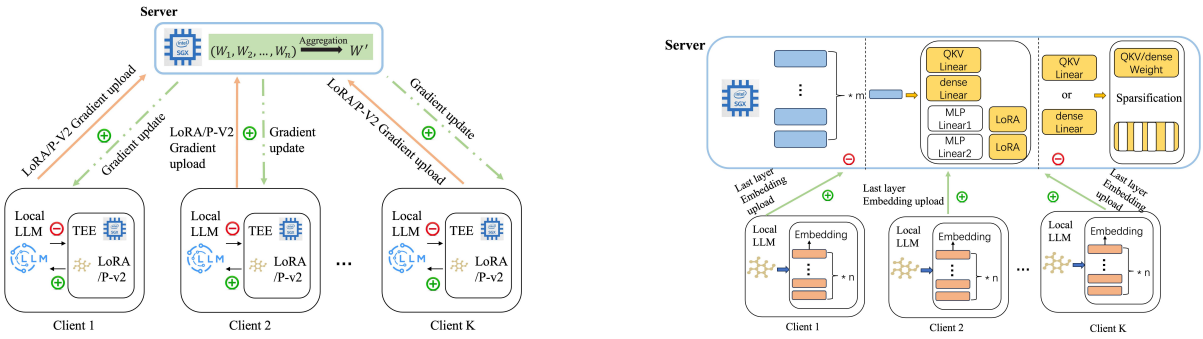
of transformers by 11 – 19× over the state-of-the-art that uses preprocessing and GPUs. Sigma is the first secure inference of GPT models, which shows the potential of MPC in use on LLMs.

2.2 Federated LLM with TEE

LLMs integrated with Trusted Execution Environments (TEEs) represent a significant advancement in secure computing. TEEs provide a secure area of the main processor that guarantees the confidentiality and integrity of the code and data loaded inside it. LLMs with TEE could have the following benefits:

- **Data Privacy.** TEEs can ensure that sensitive data used by LLMs, such as personal information or proprietary business data, is processed securely, preventing unauthorized access and leaks.
- **Model Protection.** LLMs often contain proprietary algorithms and structures. TEEs can protect these models from being reverse-engineered or tampered with, preserving intellectual property.
- **Secure Inference.** When LLMs perform inference tasks, TEEs can ensure that the computation is performed in a secure environment, which is crucial for applications in sensitive domains like healthcare or finance.
- **Auditability and Compliance.** With TEEs, it's possible to provide verifiable logs of LLM operations, which can be crucial for regulatory compliance and audit trails.

Meanwhile, the distributed (federated) LLM is an important method for co-training the domain-specific LLM using siloed data. However, maliciously stealing model parameters and data from the server or client side has become an urgent problem to be solved. TEEs can facilitate secure federated learning, where multiple parties collaboratively train an LLM without exposing their individual datasets. This is particularly useful for privacy-preserving distributed learning scenarios. [17] proposes a secure distributed LLM based on model slicing with secure communication executed in the TEE and general environments through lightweight encryption.



(a) Method 1 (b) Method 2

Figure 3 Block diagram of the distributed LLM, where the green plus sign indicates encryption and the red minus sign indicates decryption.

Shown in Figure 3, [17] proposes two methods: (a) Method 1 slices the LLM, deploys the sensitive part of the structure in the TEE, and protects the transmission between the TEE and GPU through OTP (One Time

Pad) [18] to achieve the protection effect of the model parameters and data. (b) In Method 2, after each client collects the embedding of all the data, it is uploaded to the server's TEE using OTP encryption. The server's TEE receives the data and does the decryption for further finetuning the model.

Both methods have pros and cons. Method 1 is for small memory (consumer-grade) TEE-shielded LLM Partition; while Method 2 is for large memory TEE-shielded LLM Partition. Method 1 solves the security problem, but its client generates a higher number of transmissions between the GPU and TEE during training. Secondly, during the process of encryption and decryption, there is a loss of numerical accuracy due to truncation errors, resulting in a slight degradation of accuracy. It is ideal to use Method 2 when the device is of industry grade.

Furthermore, Apple Intelligence [6], shown in Figure 4, also makes use of TEEs (marked as Secure Enclave in the figure) to provide the Private Cloud Compute (PCC) feature of LLMs and other large-scale models on both the on-device side and the Apple silicon server side. It is very promising for TEE to be widely used in federated LLM inferencing!

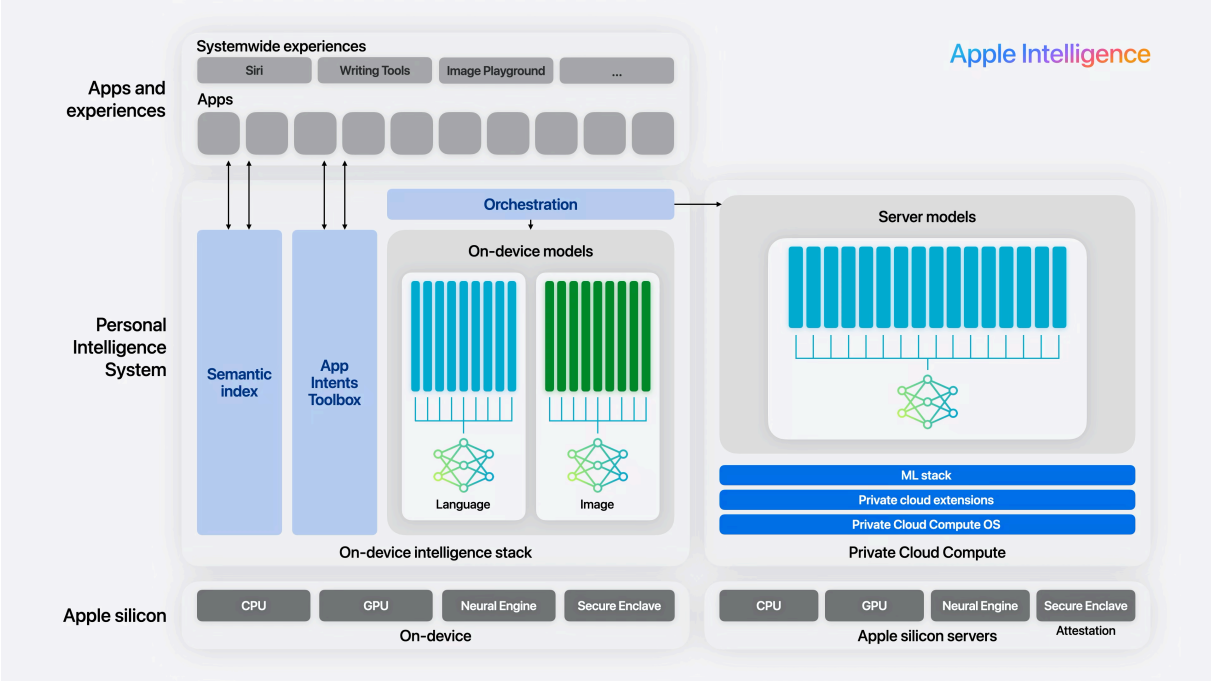


Figure 4 The architecture of Apple Intelligence [6]

2.3 zkLLM

Because ML (Machine Learning) is deployed behind these closed systems, there are increasing calls for transparency, such as releasing model weights. However, these service providers have legitimate reasons not to release this information, including for privacy and trade secrets. To bridge this gap, recent work has proposed using zero-knowledge proofs (ZKPs) for certifying computation with private models [19].

However, adapting existing ZKP techniques to modern LLMs, characterized by their immense scale, presents significant challenges. These models require substantial computational resources, which general-purpose ZKP frameworks, often unaware of LLM structure and limited in parallel computation support, struggle to provide.

For LLMs specifically, zkLLM [20] can address challenges related to the legitimacy of model outputs, offering solutions for non-arithmetic operations in deep learning and enabling zero-knowledge proofs for the attention mechanism (called zkAttn, which leveraging the foundation of tlookup). This can lead to the generation of correctness proofs for LLM inference processes, maintaining the privacy of model parameters while ensuring the integrity of the computations. Since zkLLM is implemented in CUDA, zkLLM emerges as a significant stride towards achieving efficient zero-knowledge verifiable computations over LLMs.

zkAttn. It is to achieve zero-knowledge verifiability with limited overhead for the attention mechanism, shown in Figure 5.

tlookup. It is the foundation of zkAttn. It is to address general non-arithmetic operations in deep learning. The tlookup design preserves the widely-used tensor-based structure, guaranteeing seamless compatibility with the established computational frameworks in deep learning. The protocol is shown in Figure 6.

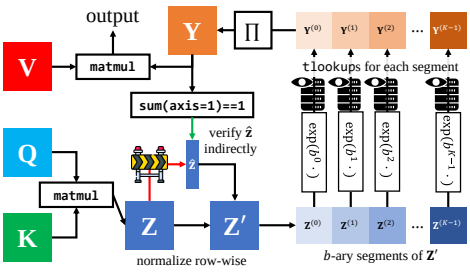


Figure 5 Overview of zkAttn

Protocol 1 tlookup

Require: The prover $\mathcal{P}$ knows $\mathbf{S} \in \mathbb{F}^D$. N, D are both powers of 2 such that N divides D.

procedure TLOOKUP-SETUP($\mathbf{T} \in \mathbb{F}^N$)
  return $[\![\mathbf{T}]\!] \leftarrow \mathtt{Commit}(\mathbf{T}; 0)$ ▷ No hiding required
end procedure
procedure $\mathcal{P}$.TLOOKUP-PREP($\mathbf{S} \in \mathbb{F}^D, \mathbf{T} \in \mathbb{F}^N$)
  Compute $\mathbf{m} = \mathbf{m}(\mathbf{S}, \mathbf{T})$ as (10)
  $\mathcal{P} \to \mathcal{V} : [\![\mathbf{S}]\!] \leftarrow \mathtt{Commit}(\mathbf{S})$
  $\mathcal{P} \to \mathcal{V} : [\![\mathbf{m}]\!] \leftarrow \mathtt{Commit}(\mathbf{m})$
end procedure
procedure $\langle \mathcal{P}, \mathcal{V} \rangle$.TLOOKUP-PROVE($[\![\mathbf{S}]\!], [\![\mathbf{m}]\!], [\![\mathbf{T}]\!]$)
  $\mathcal{V} \to \mathcal{P} : \beta \sim \mathbb{F}$
  $\mathcal{P}$ computes $\mathbf{A}, \mathbf{B}$ as (11)
  $\mathcal{P} \to \mathcal{V} : [\![\mathbf{A}]\!] \leftarrow \mathtt{Commit}(\mathbf{A}), [\![\mathbf{B}]\!] \leftarrow \mathtt{Commit}(\mathbf{B})$
  $\mathcal{P}$ and $\mathcal{V}$ run the sumcheck on (14), followed by the proofs of evaluation on $[\![\mathbf{A}]\!], [\![\mathbf{B}]\!], [\![\mathbf{S}]\!], [\![\mathbf{m}]\!]$ and $[\![\mathbf{T}]\!]$.
end procedure

Figure 6 tlookup

In summary, ZKP could be used in LLM to fortify the legitimacy of LLMs in light of their transformative impact on various domains with this promising design of zkLLM.

3 Conclusion

This paper first introduces LLM in the field of NLP, which has been applied in many areas of our daily lives. Considering the unique characteristics of LLM with a large number of parameters and training data, and unique training or finetuning methods, LLM faces data privacy and correct execution challenges in the aspect of trustworthiness, as well as the efficiency to provide such trusted LLM features.

The next generation of the Internet could come to help. This paper mainly focuses on trusted computation methods for trusted LLM: (a) Sigma [14] provides an efficient method based on FSS to perform 2PC for LLM; (b) [17] provides federated methods to secure the computation of LLM inside TEEs. Apple Intelligence [6] also shows the potential to provide secure LLM feedback based on federated learning with TEEs. (c) zkLLM [20] provides an architecture to provide zero-knowledge proof of private LLM inferencing, which could fortify the legitimacy while keeping the model parameters secret.

It is promising for more LLMs to make use of the components of trusted computing to provide services with better privacy and provable safety, which is a great cross-innovation for the field of security with NLP!

References

[1] W. X. Zhao, K. Zhou, J. Li, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[2] K.-H. Chang, "Natural language processing: Recent development and applications," *Applied Sciences*, vol. 13, no. 20, p. 11 395, 2023.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] L. Sun, Y. Huang, H. Wang, *et al.*, "Trustllm: Trustworthiness in large language models," *arXiv preprint arXiv:2401.05561*, 2024.

[5] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 27 730–27 744.

[6] Apple Inc., *Platforms state of the union - WWDC 2024*, 2024. [Online]. Available: https://developer.apple.com/videos/play/wwdc2024/102/.

[7] H. Naveed, A. U. Khan, S. Qiu, *et al.*, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[8] R. Anil, A. M. Dai, O. Firat, *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[9] J. Achiam, S. Adler, S. Agarwal, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[10] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[11] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 10 088–10 115.

[12] P. Barham, A. Chowdhery, J. Dean, *et al.*, "Pathways: Asynchronous distributed dataflow for ml," in *Proceedings of Machine Learning and Systems*, D. Marculescu, Y. Chi, and C. Wu, Eds., vol. 4, 2022, pp. 430–449. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2022/file/37385144cac01dff38247ab11c119e3c-Paper.pdf.

[13] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated large language model: A position paper," *arXiv preprint arXiv:2307.08925*, 2023.

[14] K. Gupta, N. Jawalkar, A. Mukherjee, *et al.*, *SIGMA: Secure GPT inference with function secret sharing*, Cryptology ePrint Archive, Paper 2023/1269, 2023. [Online]. Available: https://eprint.iacr.org/2023/1269.

[15] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, "Crypten: Secure multi-party computation meets machine learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 4961–4973. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/2754518221cfbc8d25c13a06a4cb8421-Paper.pdf.

[16] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing: Improvements and extensions," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1292–1303.

[17] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, and L. Wang, "A fast, performant, secure distributed training framework for llm," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 4800–4804.

[18] F. Tramer and D. Boneh, "Slalom: Fast, verifiable and private execution of neural networks in trusted hardware," *arXiv preprint arXiv:1806.03287*, 2018.

[19] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "Zkml: An optimizing system for ml inference in zero-knowledge proofs," in *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024, pp. 560–574.

[20] H. Sun, J. Li, and H. Zhang, *Zkllm: Zero knowledge proofs for large language models*, 2024. arXiv: 2404.16109 [cs.LG].