Yu TANG, Linbo QIAO, Lujia YIN, Peng LIANG, Ao SHEN, Zhilin YANG, Lizhi ZHANG, Dongsheng LI, 2025. Training large-scale language models with limited GPU memory: a survey. *Frontiers of Information Technology & Electronic Engineering*, 26(3):309-331. https://doi.org/10.1631/FITEE.2300710

Training large-scale language models with limited GPU memory: a survey

Key words: Training techniques; Memory optimization; Model parameters; Model states; Model activations

Yu TANG, Dongsheng LI
E-mail: tangyu14@nudt.edu.cn; dsli@nudt.edu.cn
ORCID: Yu TANG, https://orcid.org/0000-0002-8595-1547
Dongsheng LI, https://orcid.org/0000-0001-9743-2034

GPU memory wall

- Further advancement of large-scale models is substantially hindered by the limited GPU memory.

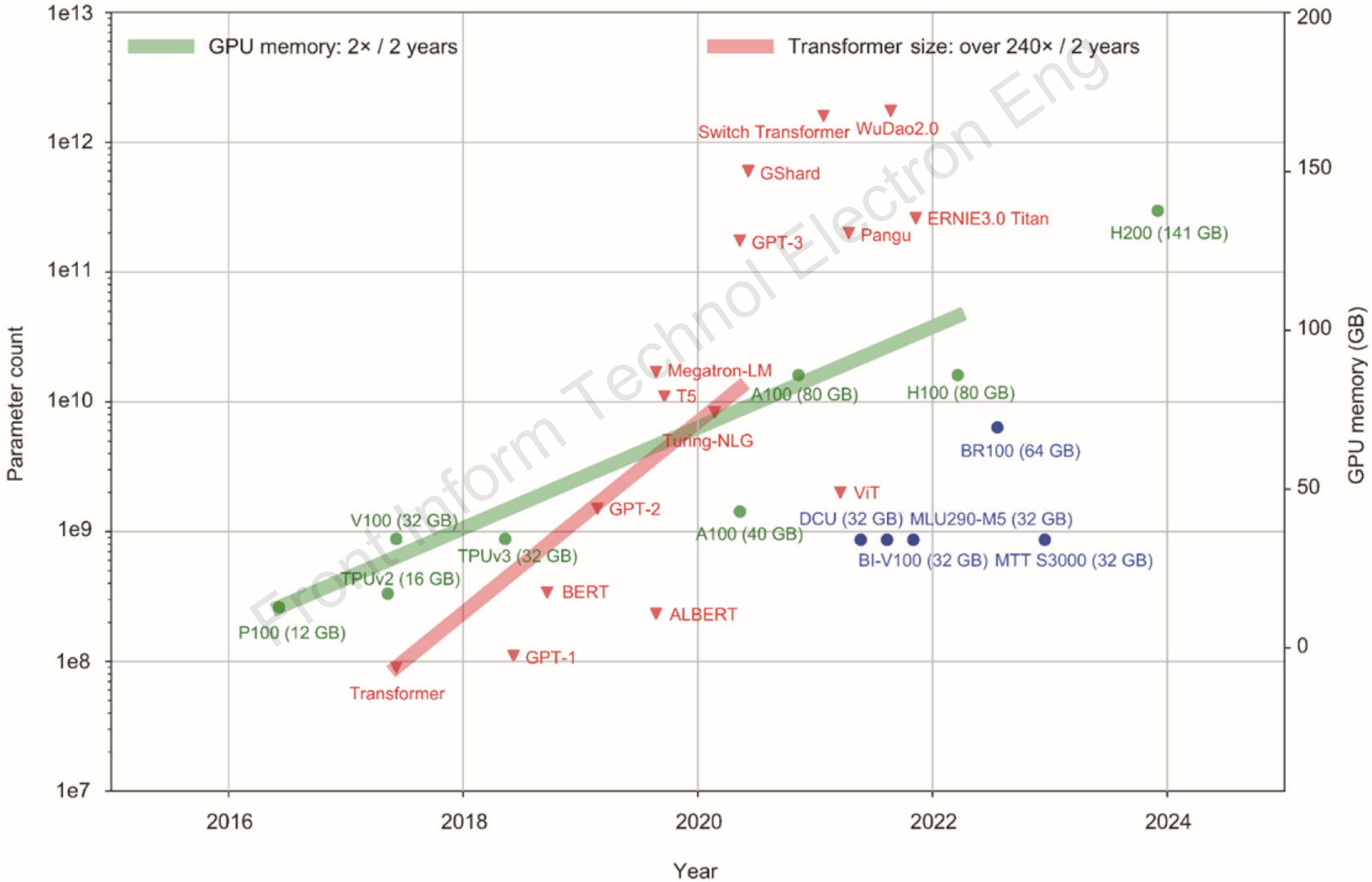
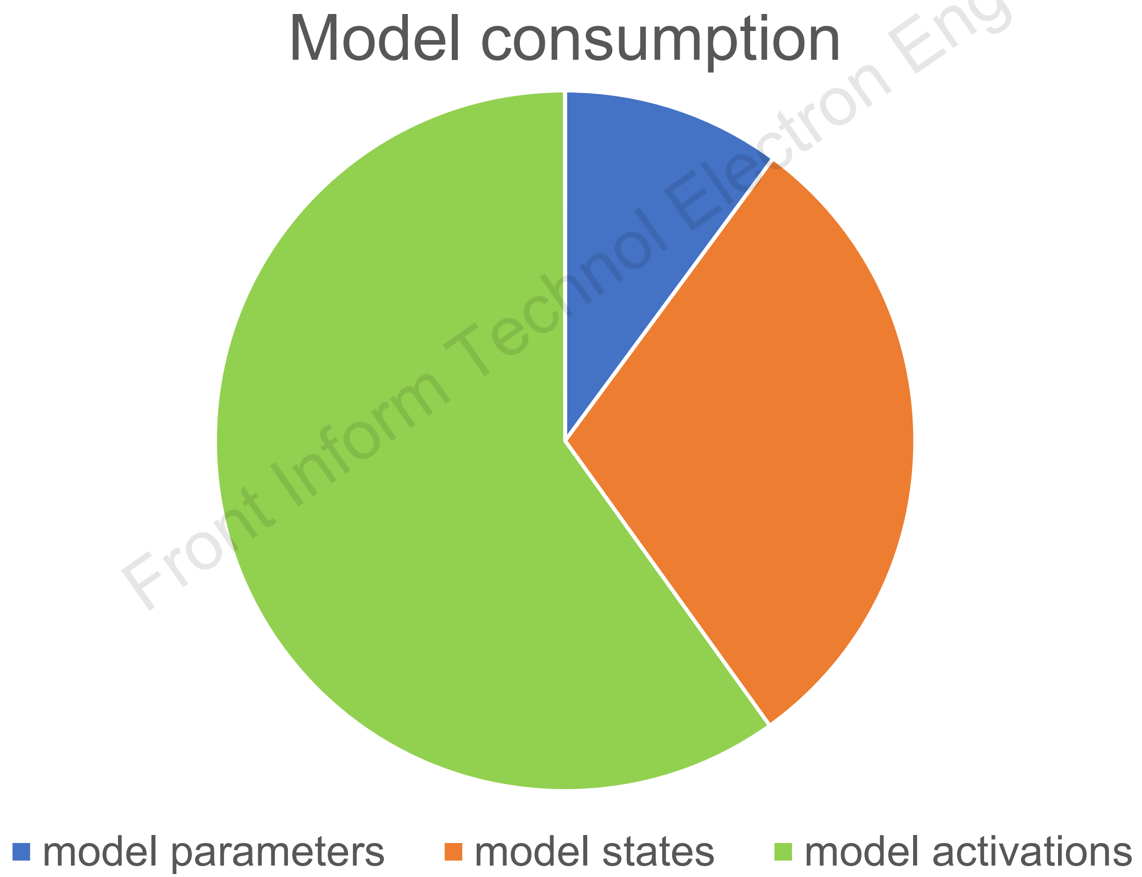


Fig. 1 Development of large-scale models' sizes and GPU memory capacity in recent years. It is obvious that the sizes of these models are increasing more and more rapidly and far beyond the capacity of GPU. References to color refer to the online version of this figure

GPU memory consumption analysis

- In training large-scale models, most memory consumption comes from model parameters, model states, and model activations.



Reducing memory of model parameters

- Multi-GPU training: Model Parallel, Pipeline Parallel, Mixed Parallel
- Mixed-precision training
- Specific model design: ReZero

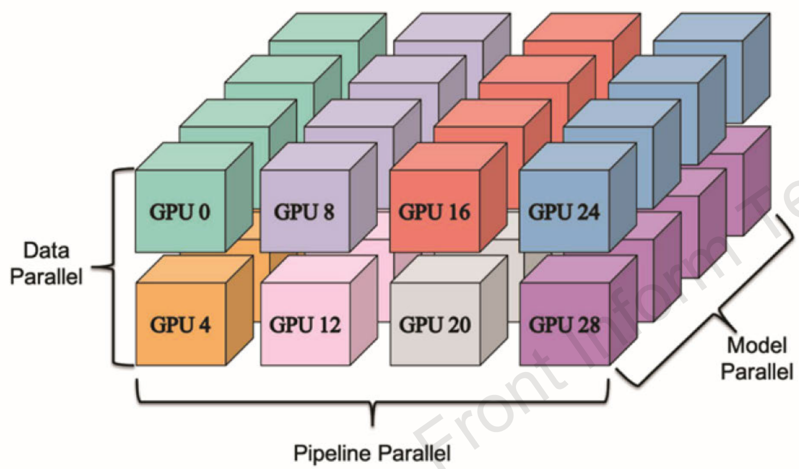


Fig. 5 Overview of 3D Parallel, which consists of Data Parallel, MP, and Pipeline Parallel. These parallel dimensions shard GPUs as their degrees in the training system

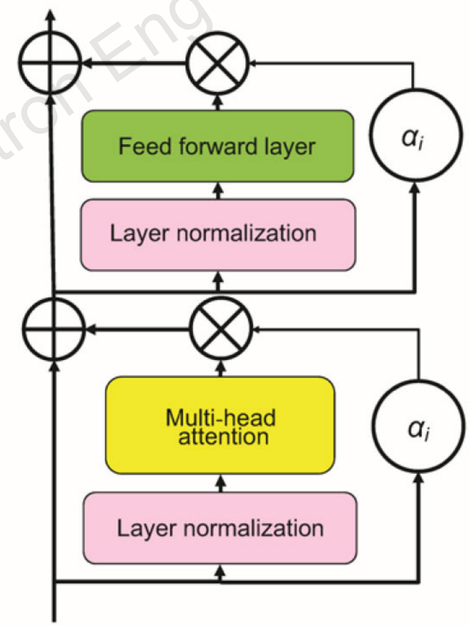
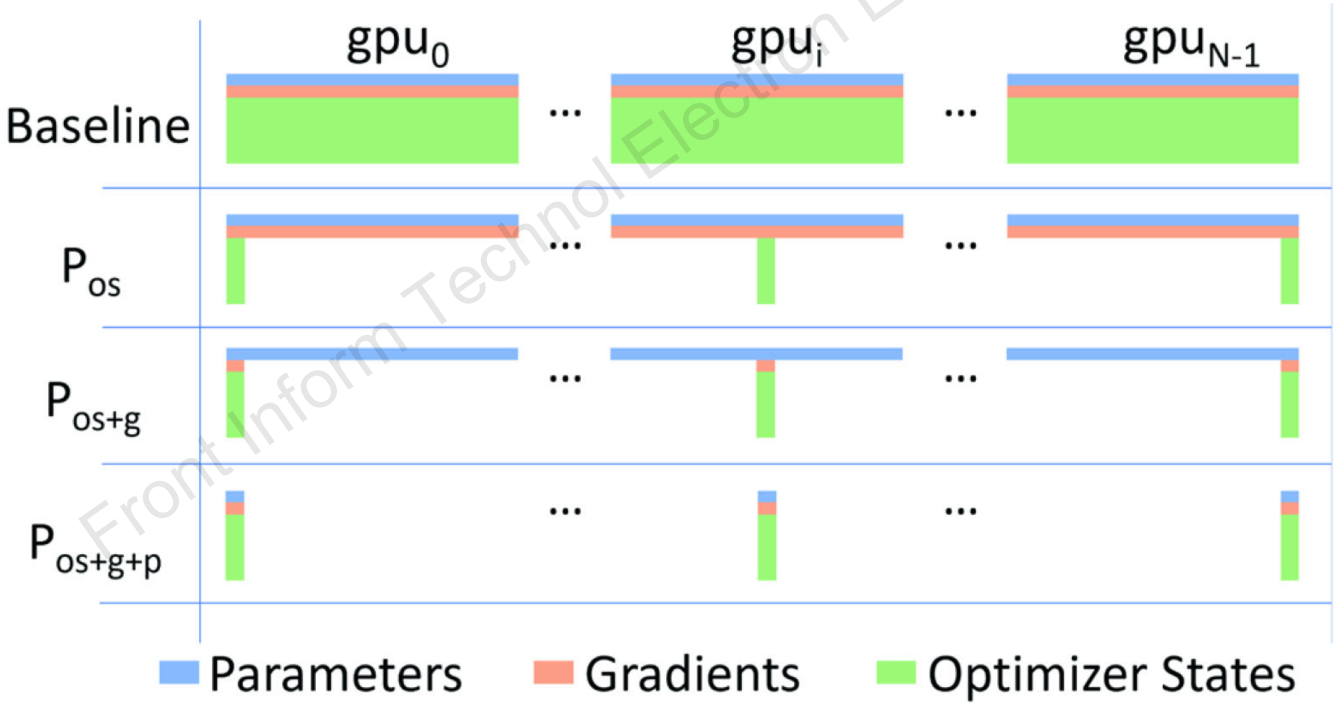


Fig. 9 Architecture of the ReZero Transformer. The parameter α determines whether the layer is calculated in the process

Reducing memory of model states

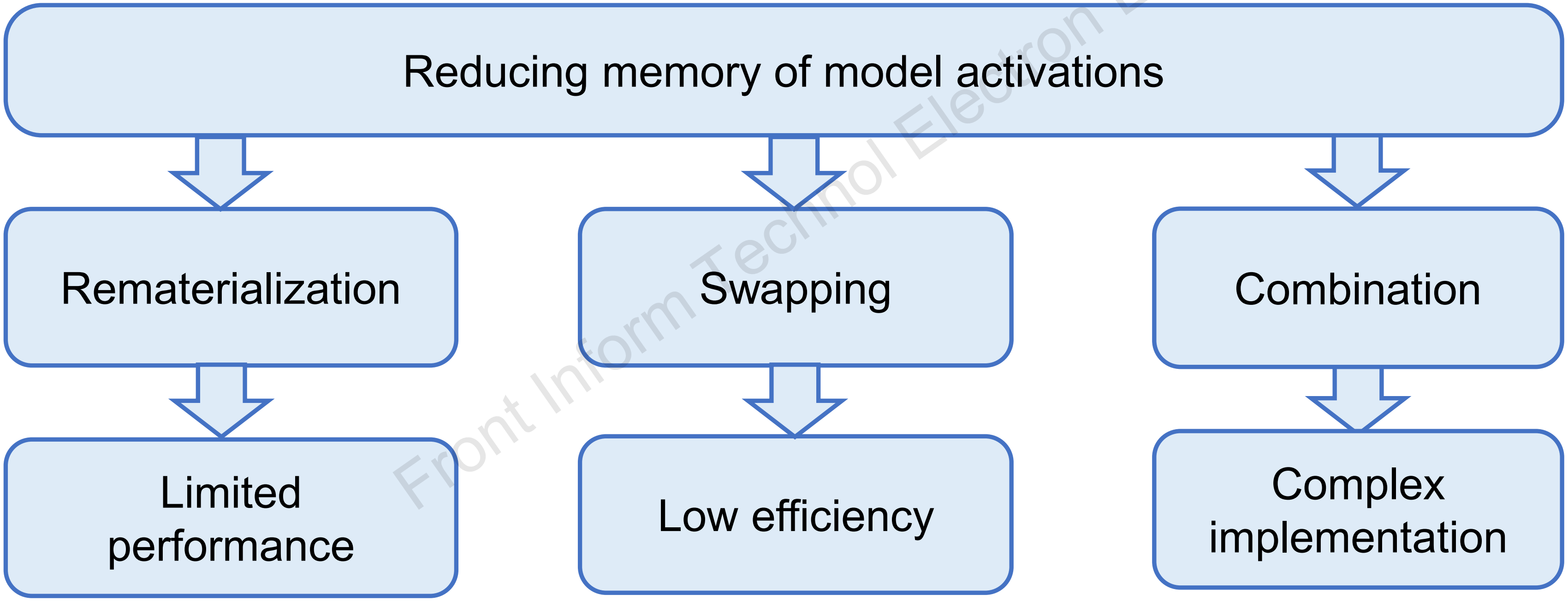
- ZeRO (Zero Redundancy Optimizer)
- PatrickStar
- Adafactor
- CAME
- DeepZeRo



Rajbhandari Samyam, et al., 2020. ZeRO: memory optimizations toward training trillion parameter models. SC20: IEEE Int Conf for High Performance Computing, Networking, Storage and Analysis.

Reducing memory of model activations

- Rematerialization: Checkpoint, DTR, and Checkmate
- Swapping: SwapAdvisor, ZeRO-Offload, and ZeRO-Infinity
- Combination: SuperNeurons, Capuchin, and DELTA



Discussion

Future research	Outlook
Supporting new parallel methods	• Expert Parallel • Sequence Parallel • FlashAttention
Ensuring the convergence of optimizers	• Theoretical analysis • Stable convergence • Less memory consumption
Improving rematerialization and swapping	• Memory fragmentation • Low throughput • Complex implementation

References

[1] Gholami Amir, et al., 2024. AI and memory wall. IEEE Micro.

[2] Jain Paras, et al., 2020. Checkmate: breaking the memory wall with optimal tensor rematerialization. Proceedings of Machine Learning and Systems, p.497-511.

[3] Bachlechner Thomas, et al., 2021. ReZero is all you need: fast convergence at large depth. PMLR.

[4] Fang Jiarui, et al., 2022. Parallel training of pre-trained models via chunk-based dynamic memory management. IEEE Transactions on Parallel and Distributed Systems, 34(1):304-315.

[5] Ren Jie, et al., 2021. ZeRo-Offload: democratizing billion-scale model training. USENIX Annual Technical Conference.

[6] Rasley Jeff, et al., 2020. DeepSpeed: system optimizations enable training deep learning models with over 100 billion parameters. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

[7] Rajbhandari Samyam, et al., 2020. ZeRo: memory optimizations toward training trillion parameter models. SC20: IEEE International Conference for High Performance Computing, Networking, Storage and Analysis.

[8] Lai Zhiquan, et al., 2023. Merak: an efficient distributed DNN training framework with automated 3D parallelism for giant foundation models. IEEE Transactions on Parallel and Distributed Systems, 34(5):1466-1478.

[9] Liang Peng, et al., 2023. A survey on auto-parallelism of large-scale deep learning training. IEEE Transactions on Parallel and Distributed Systems, 34(8):2377-2390.

[10] Tang Yu, et al., 2024. DELTA: memory-efficient training via dynamic fine-grained recomputation and swapping. ACM Transactions on Architecture and Code Optimization, 21(4):1-25.