香港科技大学（广州）
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

MoC-System: Efficient Fault Tolerance for Sparse Mixture-of-Experts Model Training

Weilin Cai, Le Qin, Jiayi Huang

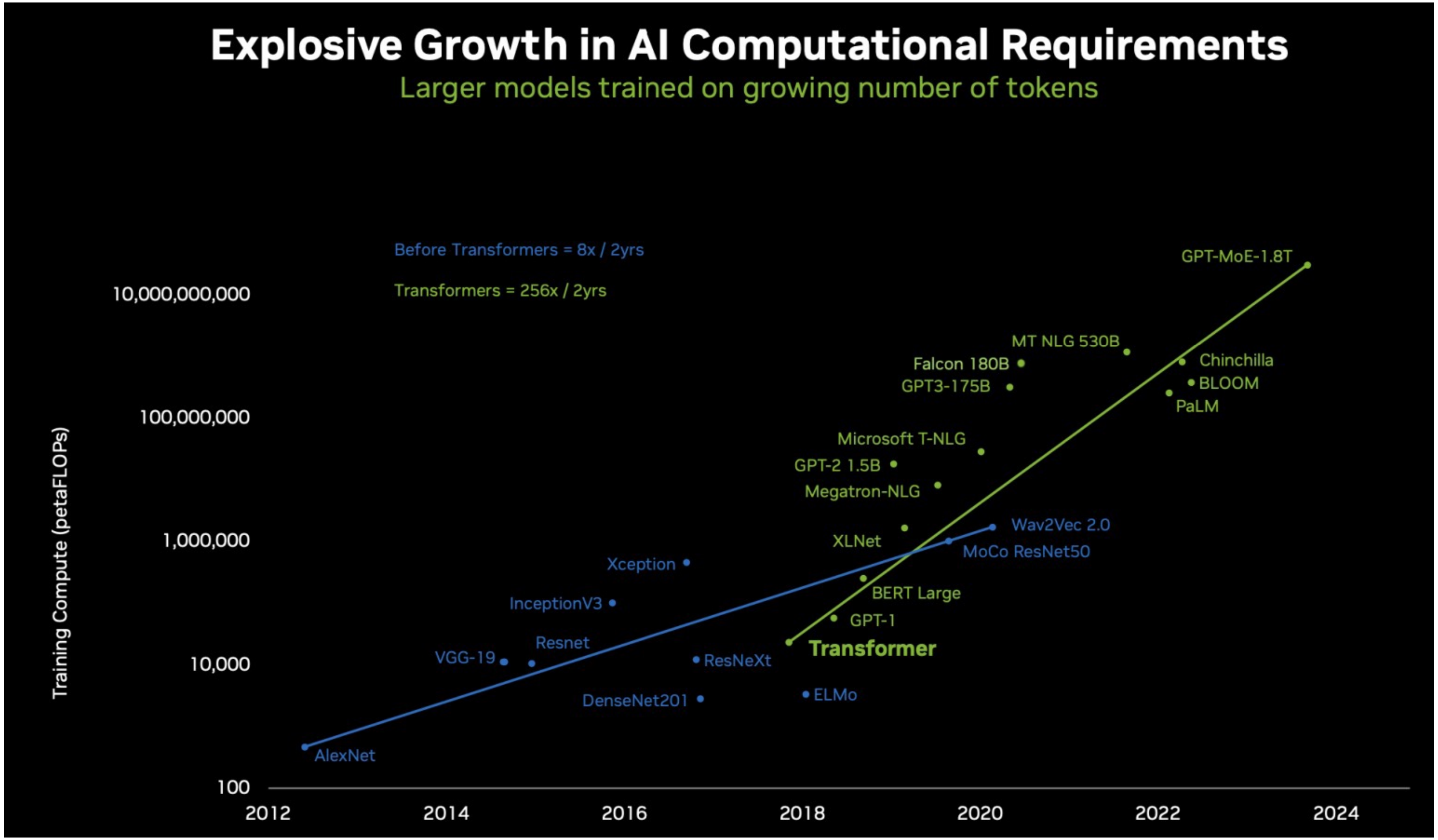
Success of Large Language Models (LLMs)

deepseek

Copilot

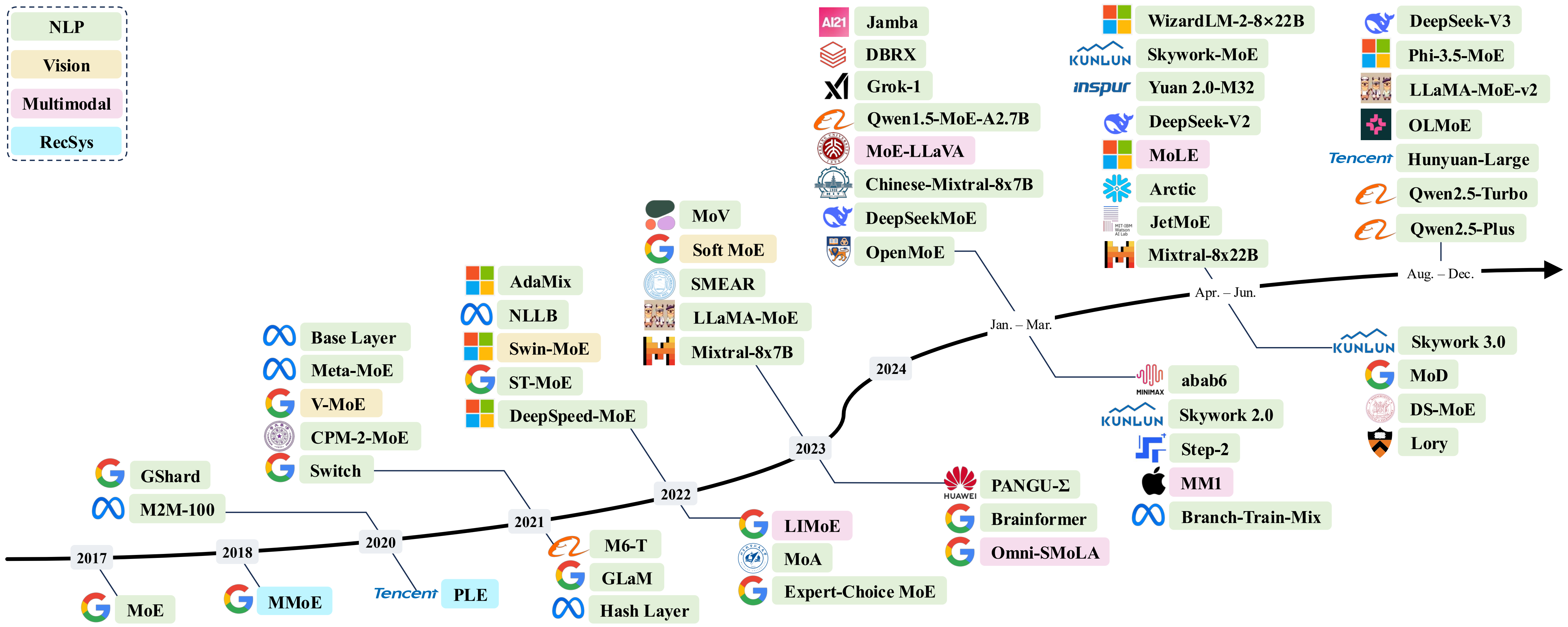
Scaling Laws of LLMs

- Model Size ↑
- Data Size ↑
- Computation ↑



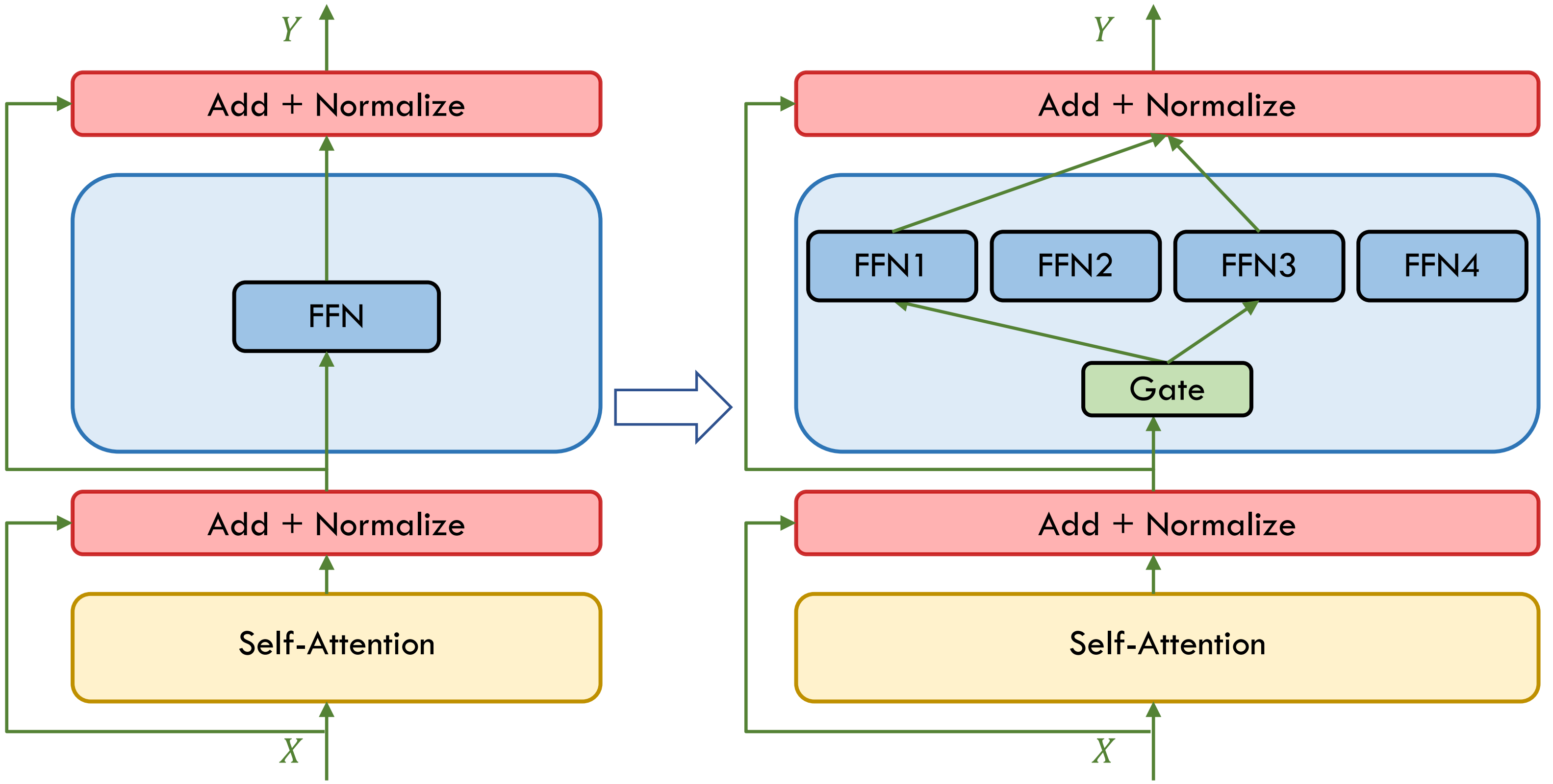
Source: NVIDIA, GTC March 2024 Keynote.

Mixture-of-Experts (MoE) Model



[1] Cai, Weilin, et al. "A Survey on Mixture of Experts in Large Language Models." IEEE TKDE (2025).

MoE Architecture

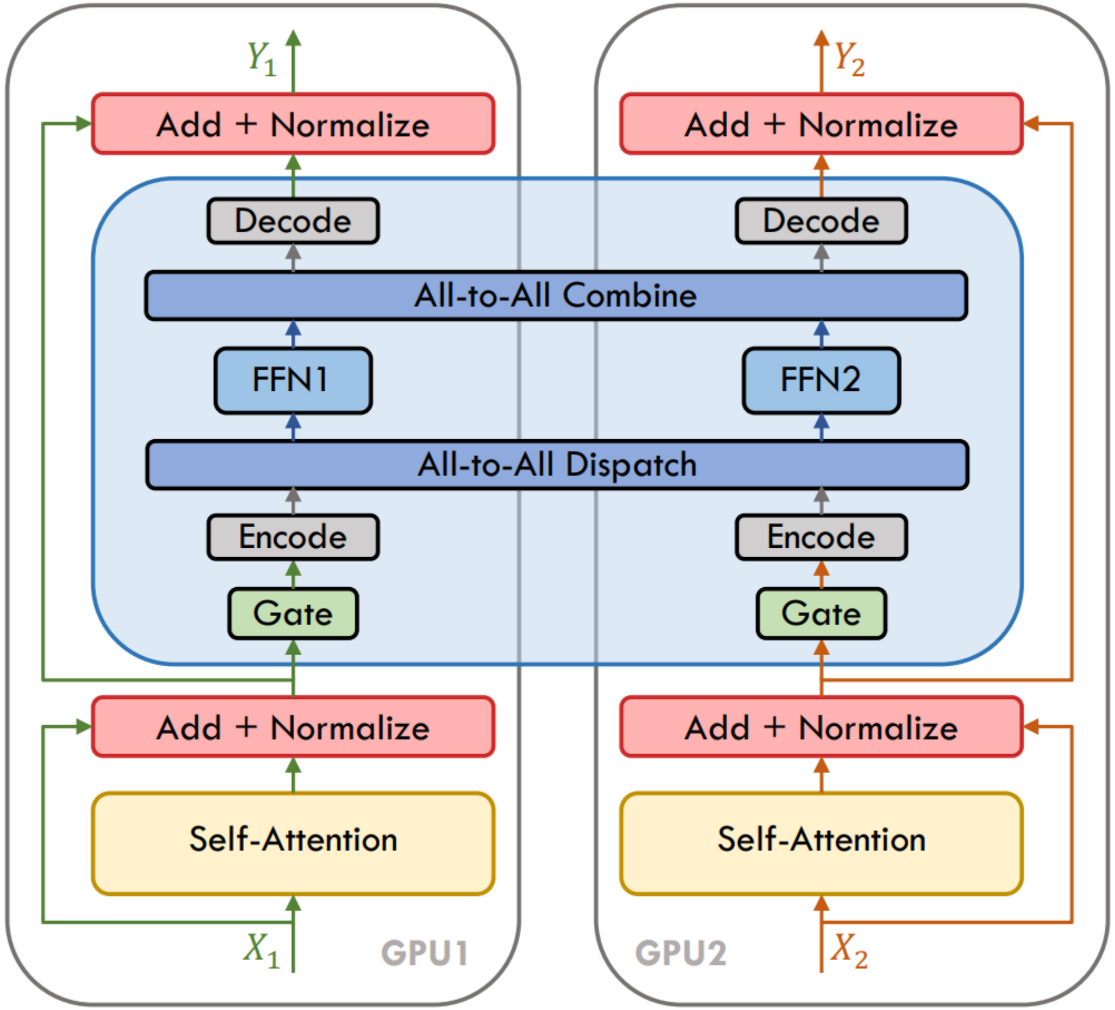


Dense Model **Sparse Mixture-of-Experts (MoE) Model**

Increase the number of parameters

Without correspondingly increasing the computation.

Expert Parallelism for Distributed Training



Fault in LLM Training

Component	Category	Interruption Count	% of Interruptions
Faulty GPU	GPU	148	30.1%
GPU HBM3 Memory	GPU	72	17.2%
Software Bug	Dependency	54	12.9%
Network Switch/Cable	Network	35	8.4%
Host Maintenance	Unplanned Maintenance	32	7.6%
GPU SRAM Memory	GPU	19	4.5%
GPU System Processor	GPU	17	4.1%
NIC	Host	7	1.7%
NCCL Watchdog Timeouts	Unknown	7	1.7%
Silent Data Corruption	GPU	6	1.4%
GPU Thermal Interface + Sensor	GPU	6	1.4%
SSD	Host	3	0.7%
Power Supply	Host	3	0.7%
Server Chassis	Host	2	0.5%
IO Expansion Board	Host	2	0.5%
Dependency	Dependency	2	0.5%
CPU	Host	2	0.5%
System Memory	Host	2	0.5%

Unexpected interruptions during a 54-Day Llama 3 405B [2] pre-training on a 16,000-GPU cluster.

[2] Grattafiori, Aaron, et al. "The llama 3 herd of models." arXiv preprint arXiv:2407.21783 (2024).

Fault in LLM Training

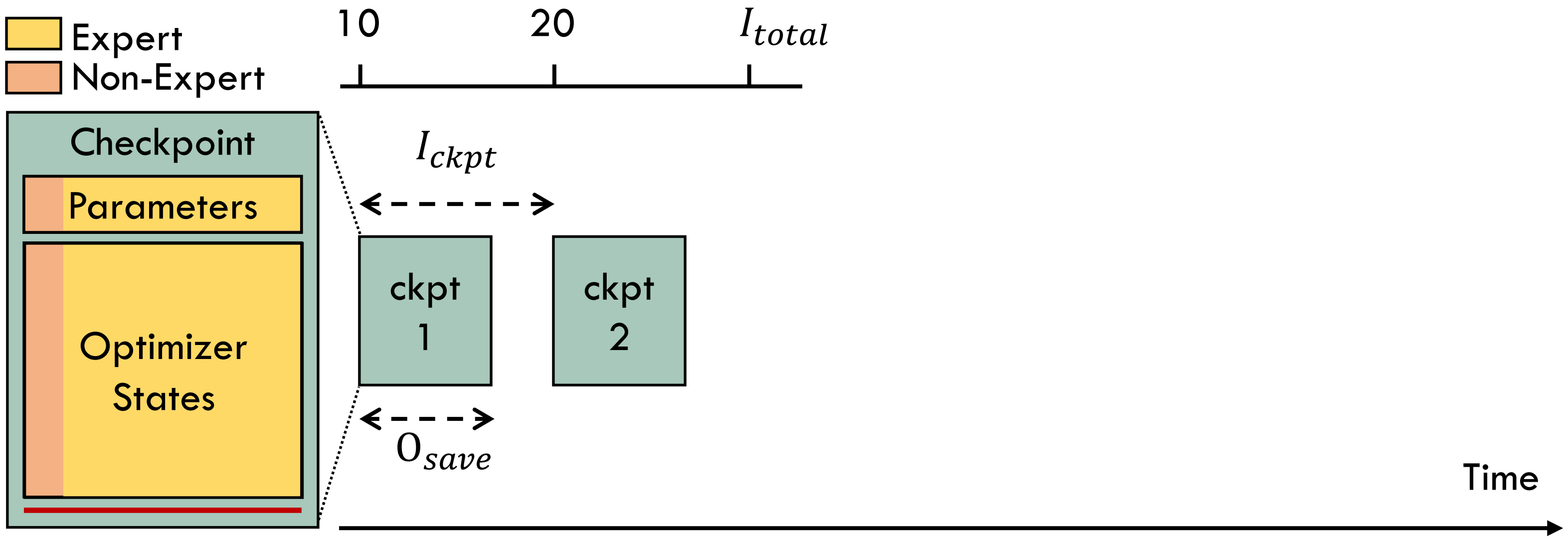
Component	Category	Interruption Count	% of Interruptions
Faulty GPU	GPU	148	30.1%
GPU HBM3 Memory	GPU	72	17.2%
Software Bug	Dependency	54	12.9%
Network Switch/Cable	Network	35	8.4%
Host Maintenance	Unplanned Maintenance	32	7.6%
GPU SRAM Memory	GPU	19	4.5%
SSD	Host	3	0.7%
Power Supply	Host	3	0.7%
Server Chassis	Host	2	0.5%
IO Expansion Board	Host	2	0.5%
Dependency	Dependency	2	0.5%
CPU	Host	2	0.5%
System Memory	Host	2	0.5%

On average, a fault occurs every 2 hours.

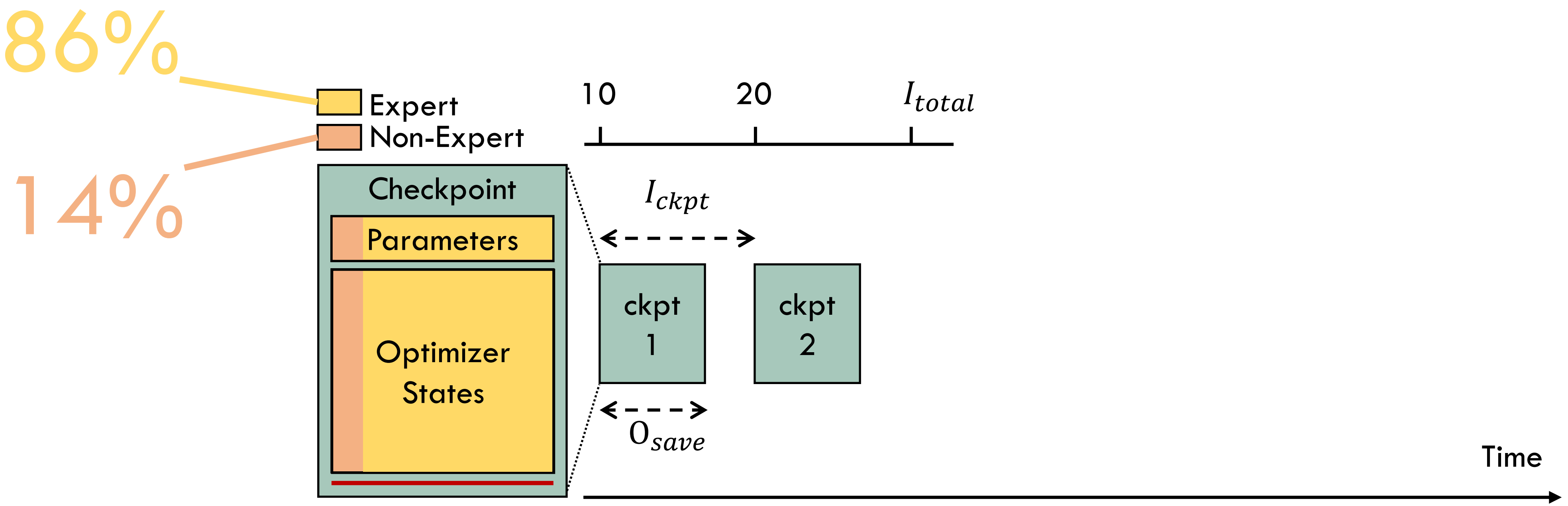
Unexpected interruptions during a 54-Day Llama 3 405B [2] pre-training on a 16,000-GPU cluster.

[2] Grattafiori, Aaron, et al. "The llama 3 herd of models." arXiv preprint arXiv:2407.21783 (2024).

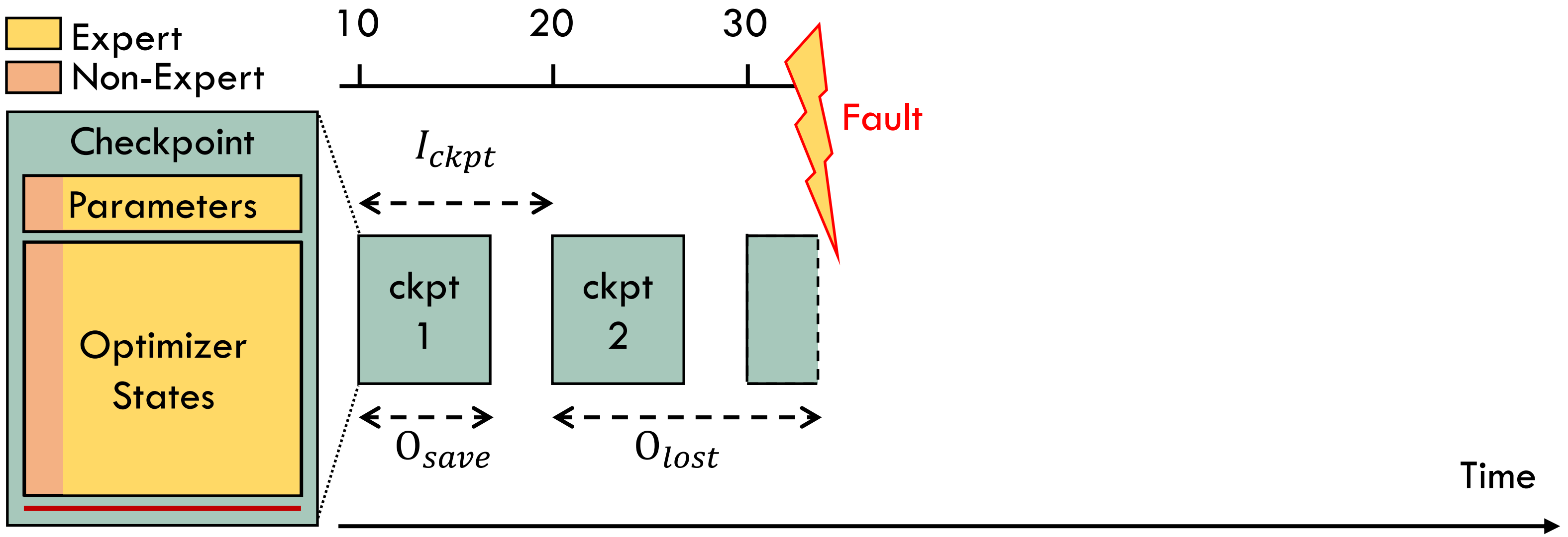
Fault Tolerance (Checkpoint) in LLM Training



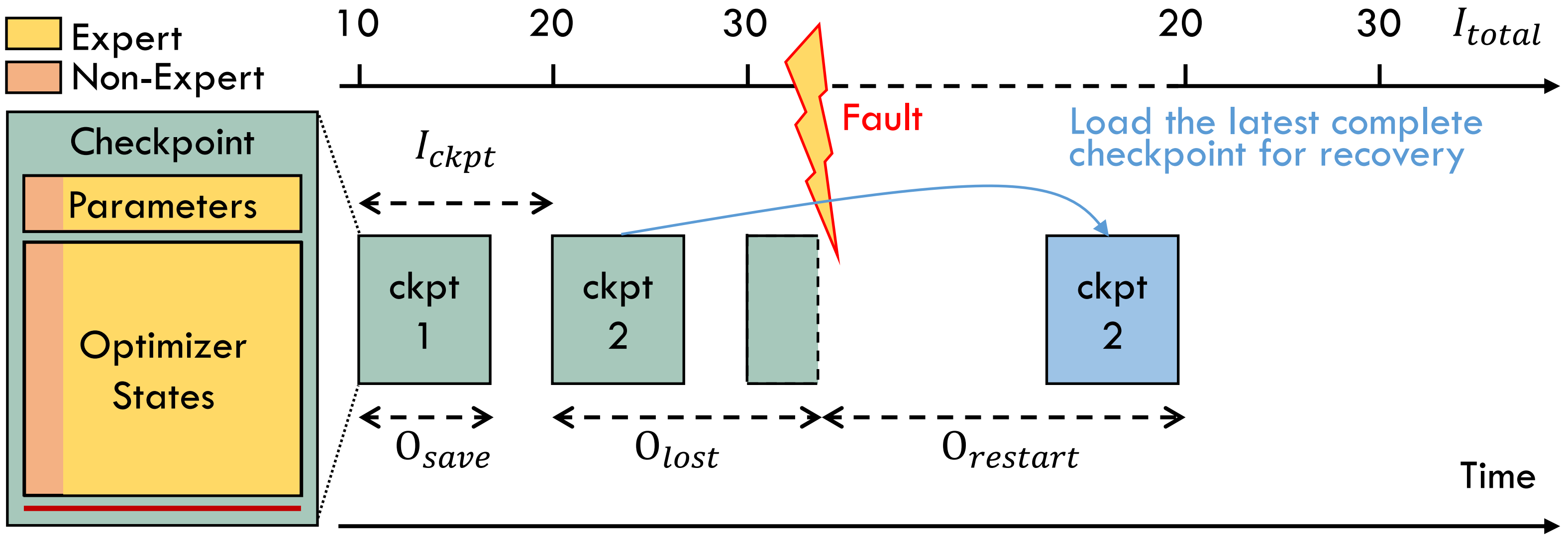
Fault Tolerance (Checkpoint) in LLM Training



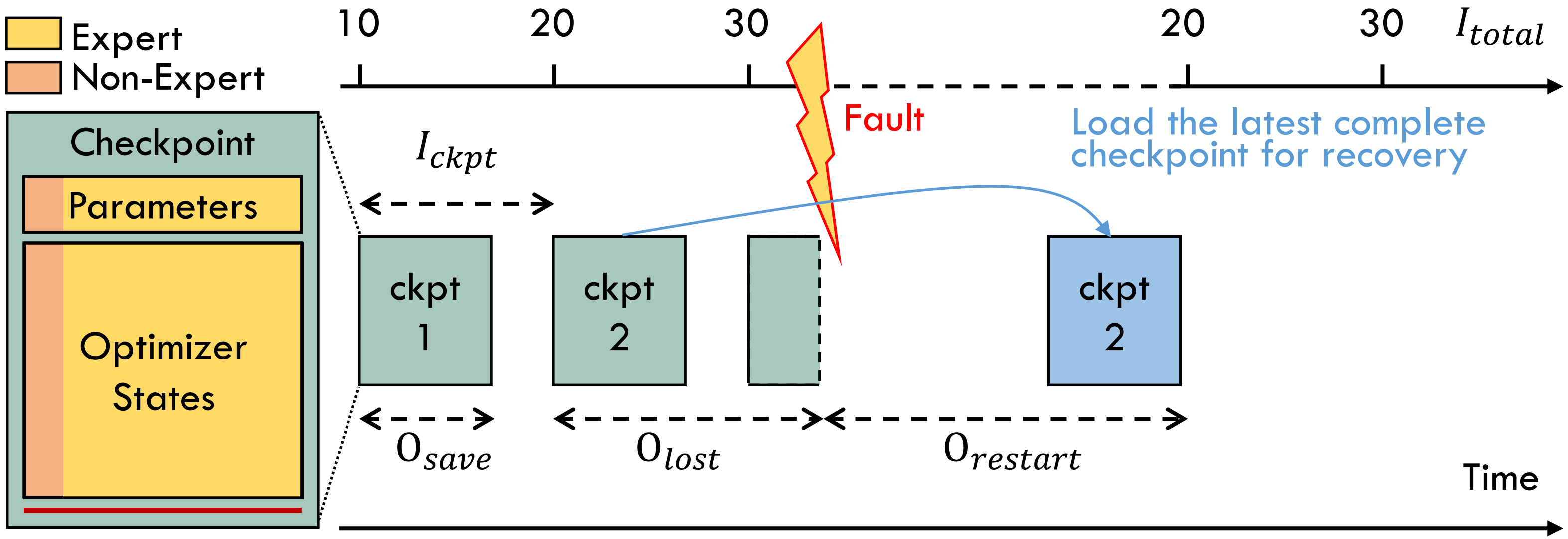
Fault Tolerance (Checkpoint) in LLM Training



Fault Tolerance (Checkpoint) in LLM Training

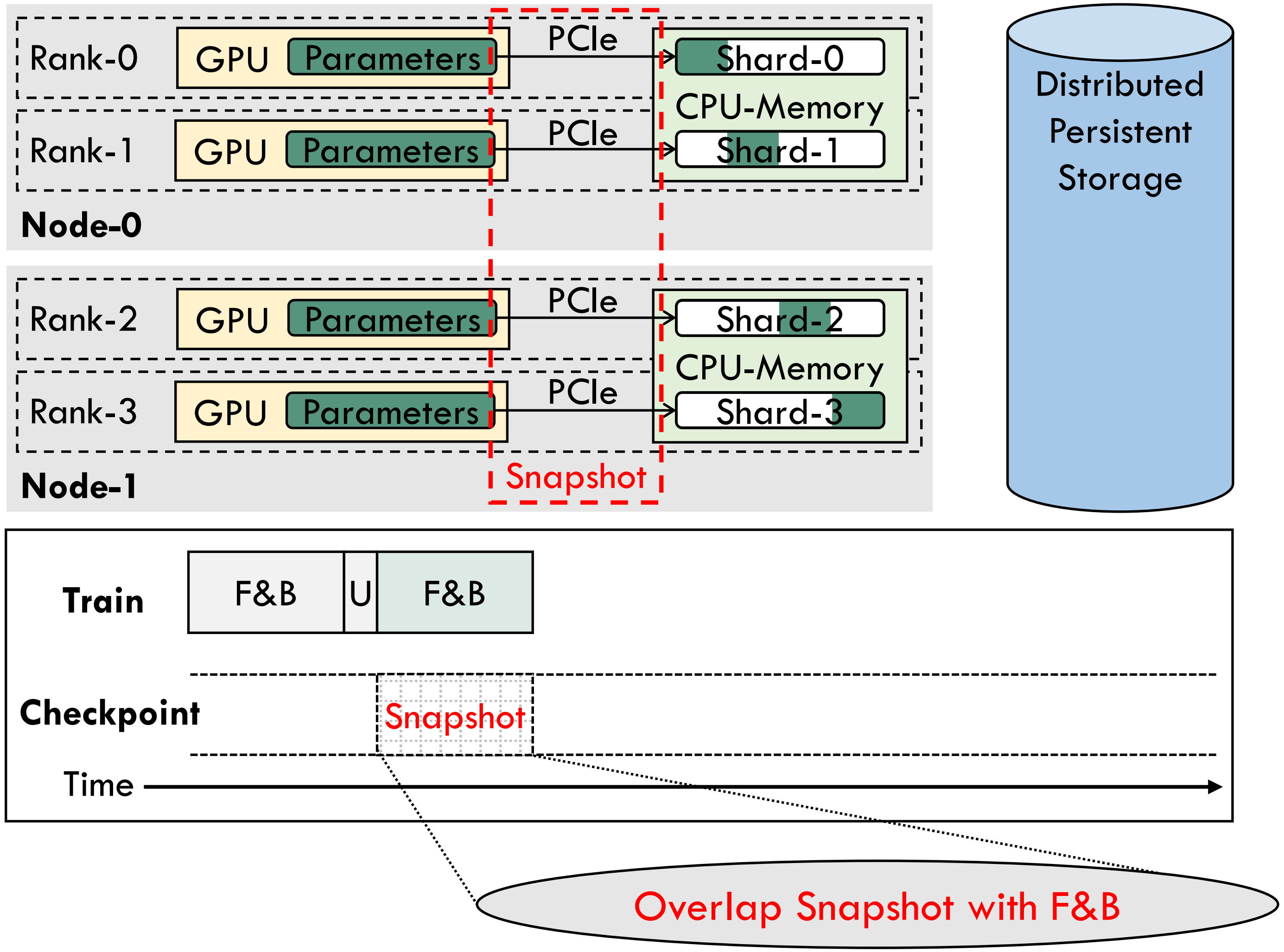


Fault Tolerance (Checkpoint) in LLM Training

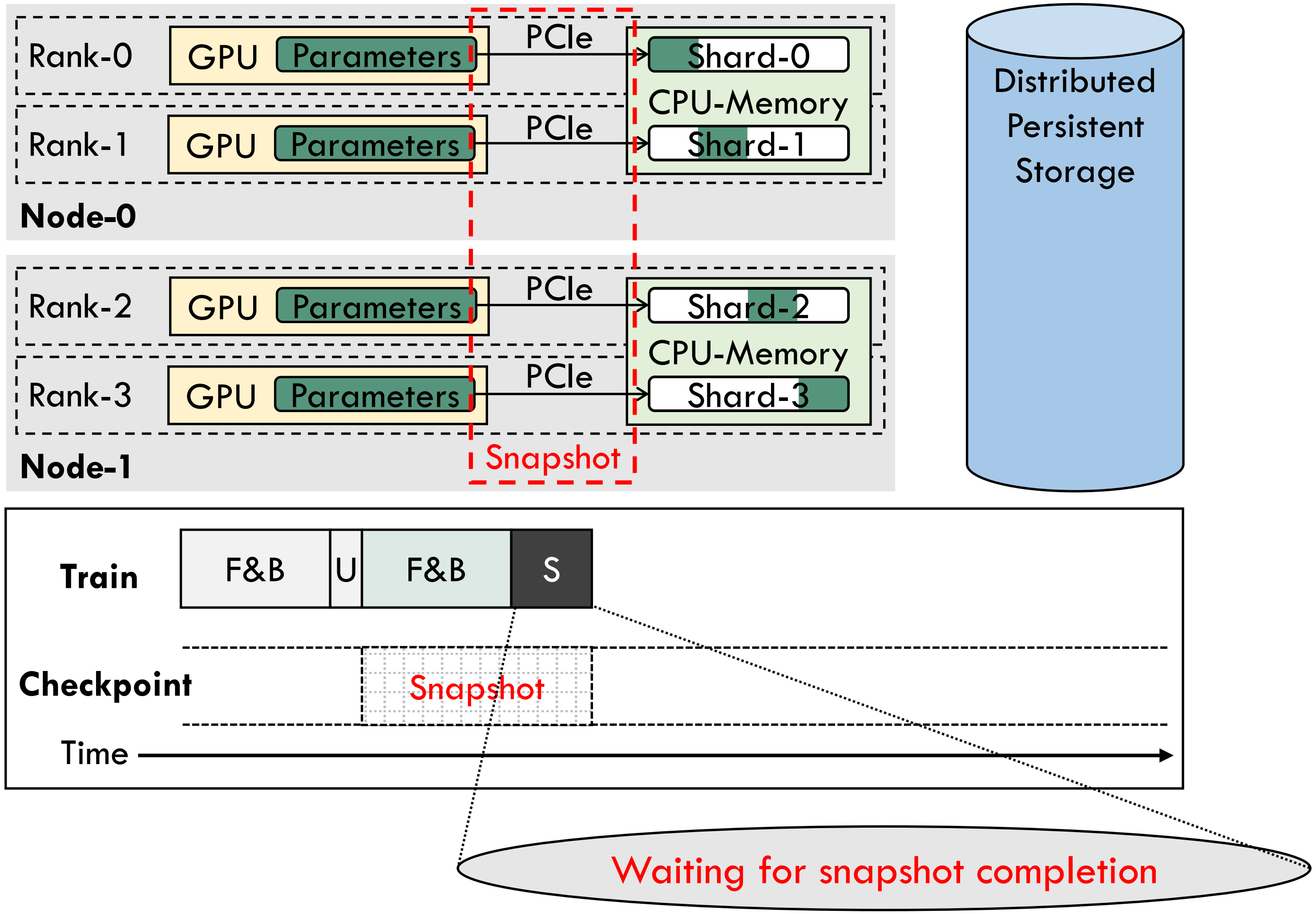


$$O_{ckpt} = O_{save}\frac{I_{total}}{I_{ckpt}} + \sum_{i=1}^{N_{fault}}\left(O_{restart}^{i} + O_{lost}^{i}\right)$$

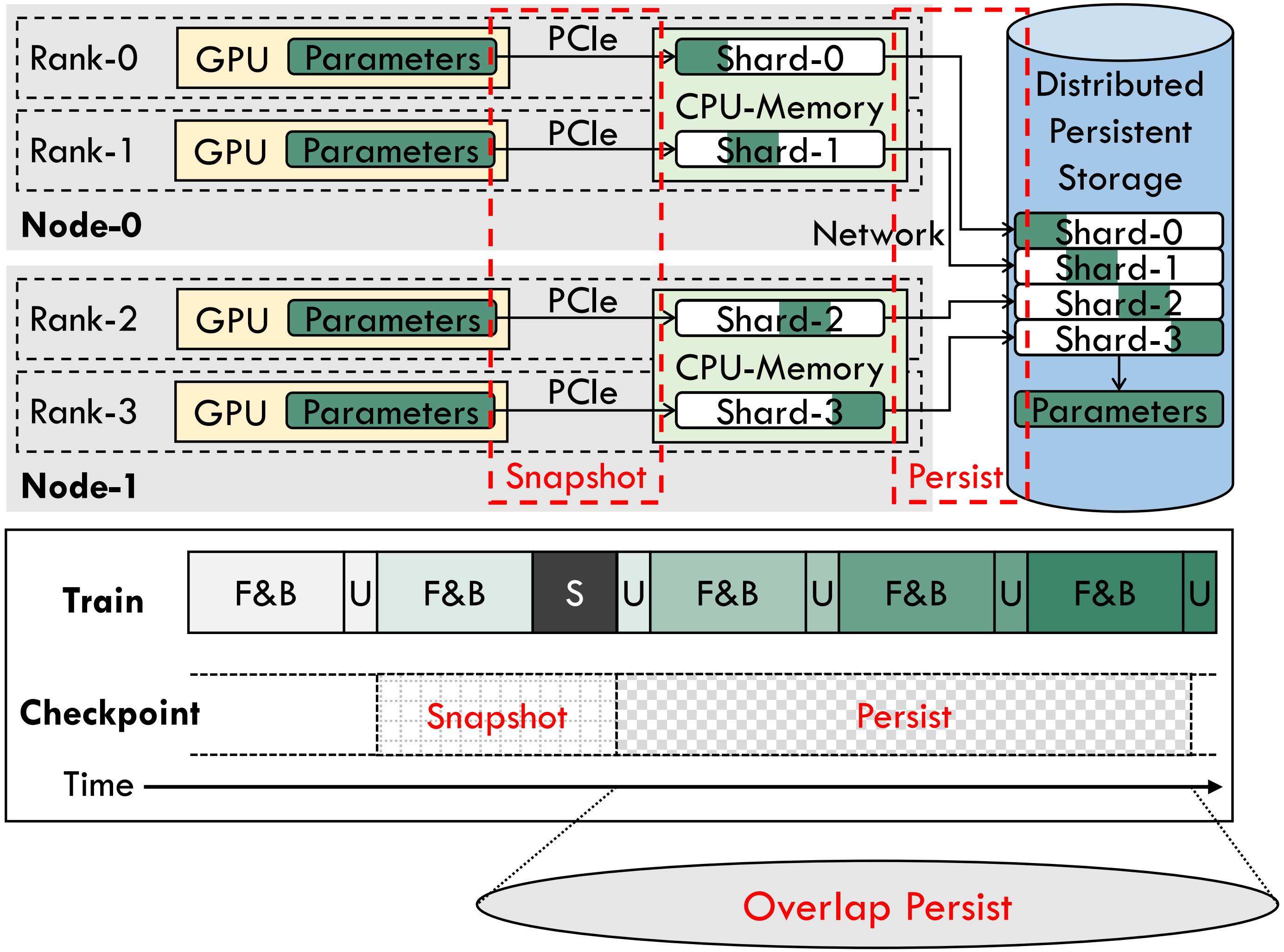
Checkpointing Workflow



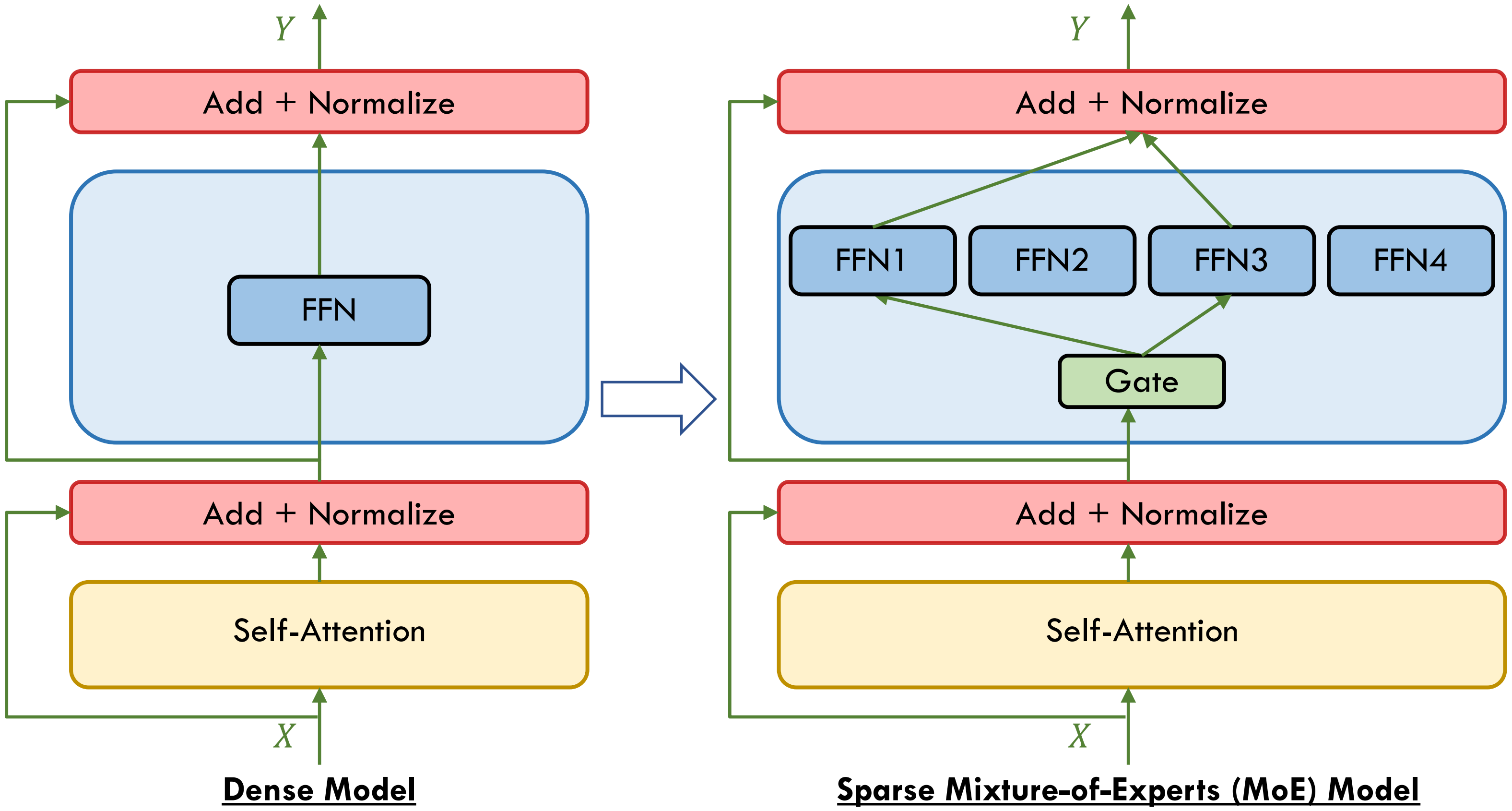
Checkpointing Workflow



Checkpointing Workflow



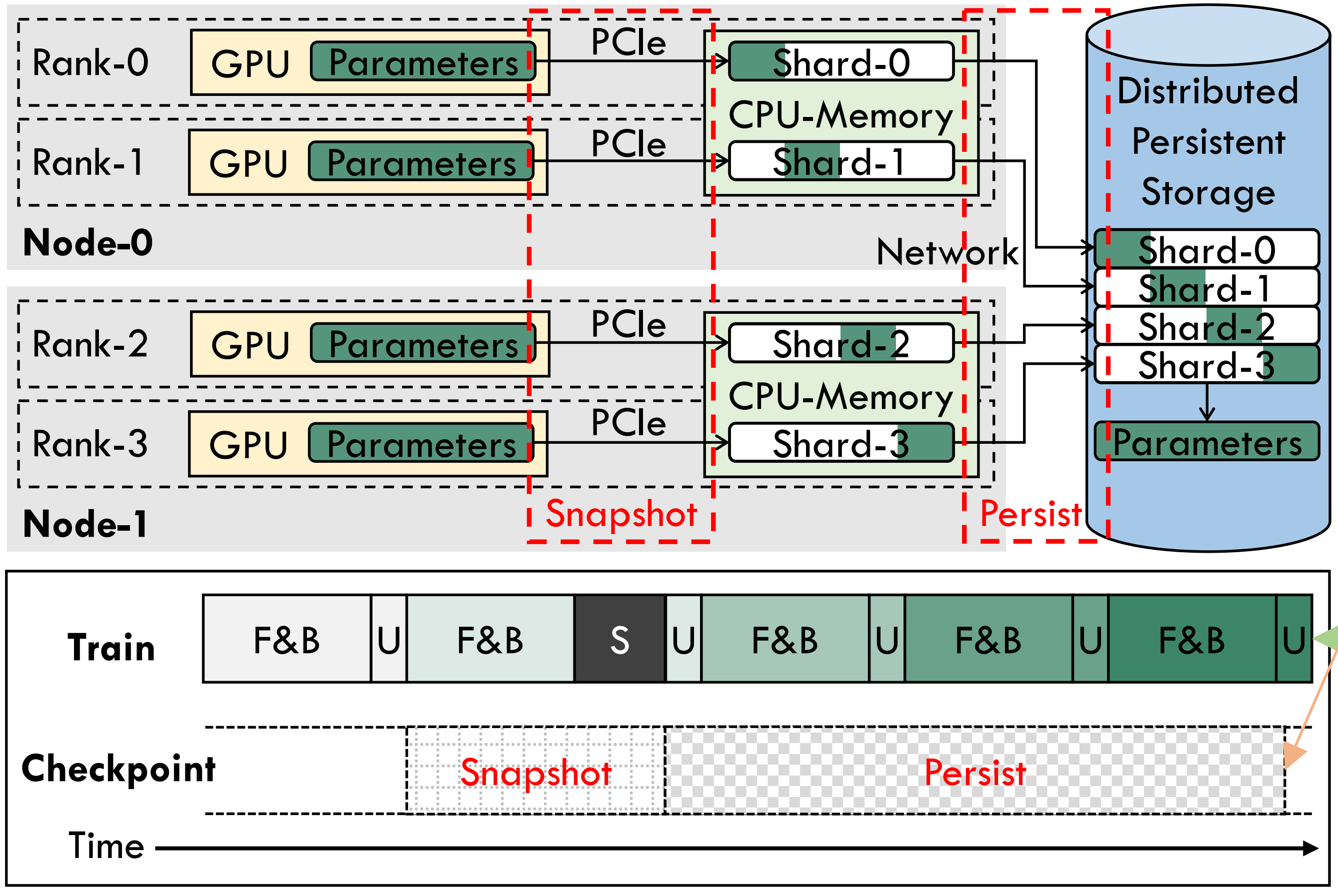
New Challenges in Checkpointing of MoE Model



Increase the number of parameters

Without a corresponding increase in iteration computation.

New Challenges in Checkpointing of MoE Model



Increasing the number of parameters results in
(1) larger storage burden,
(2) longer durations for Snapshot and Persist.

Without a corresponding increase in F&B duration for overlap.

Insensitive Expert Parameters

Expert is insensitive to a limited number of training updates.

1. MoE models generally require larger volumes of pre-training data [3,4,5].
2. Fine-tuning only the non-expert parameters performs good accuracy[6].
3. Fine-tuning only the expert parameters leads to a drastic reduction in accuracy [6].

[3] Artetxe, Mikel, et al. "Efficient large scale language modeling with mixtures of experts." arXiv preprint arXiv:2112.10684 (2021).
[4] Fedus, William, Barret Zoph, and Noam Shazeer. "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity." Journal of Machine Learning Research 23.120 (2022): 1-39.
[5] Xue, Fuzhao, et al. "Go wider instead of deeper." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 8. 2022.
[6] Zoph, Barret, et al. "St-moe: Designing stable and transferable sparse expert models." arXiv preprint arXiv:2202.08906 (2022).

Insensitive Expert Parameters

Expert is insensitive to a limited number of training updates.

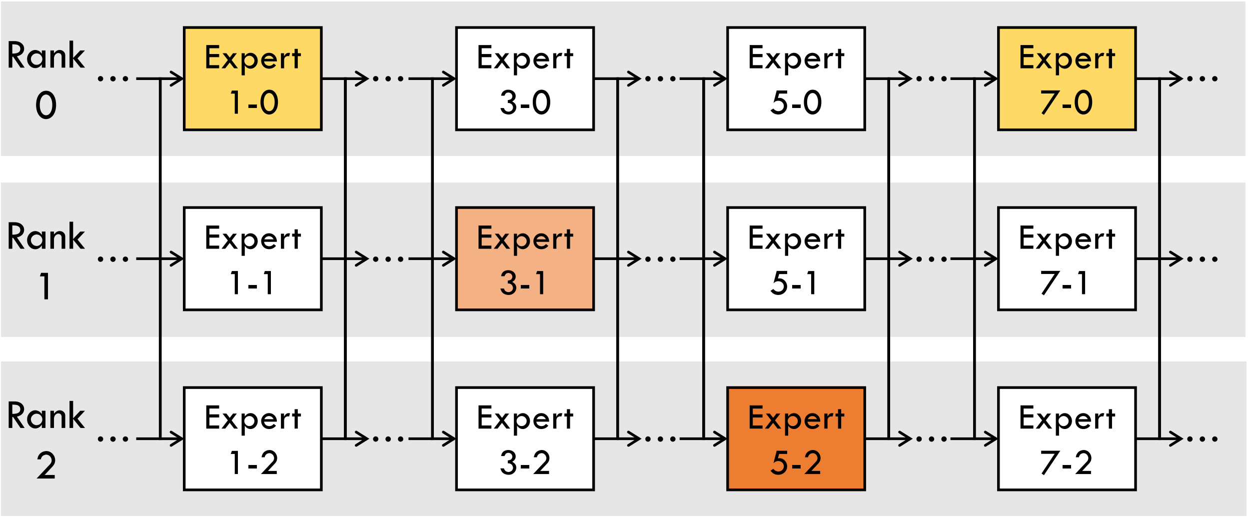
1. MoE models generally require larger volumes of pre-training data [3,4,5].
2. Fine-tuning only the non-expert parameters performs good accuracy[6].
3. Fine-tuning only the expert parameters leads to a drastic reduction in accuracy [6].

	HS	PIQA	WG	BoolQ	ARC-C	OBQA	RTE	AVG.
Base (OLMoE)	57.99	80.52	68.59	74.46	47.27	44.80	54.51	61.16
Finetune w.o. expert	58.58	81.88	68.51	76.82	48.72	45.20	63.54	63.32
Finetune all parameters	58.34	81.34	70.40	79.11	48.38	45.00	66.06	64.09

[3] Artetxe, Mikel, et al. "Efficient large scale language modeling with mixtures of experts." arXiv preprint arXiv:2112.10684 (2021).
[4] Fedus, William, Barret Zoph, and Noam Shazeer. "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity." Journal of Machine Learning Research 23.120 (2022): 1-39.
[5] Xue, Fuzhao, et al. "Go wider instead of deeper." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 8. 2022.
[6] Zoph, Barret, et al. "St-moe: Designing stable and transferable sparse expert models." arXiv preprint arXiv:2202.08906 (2022).

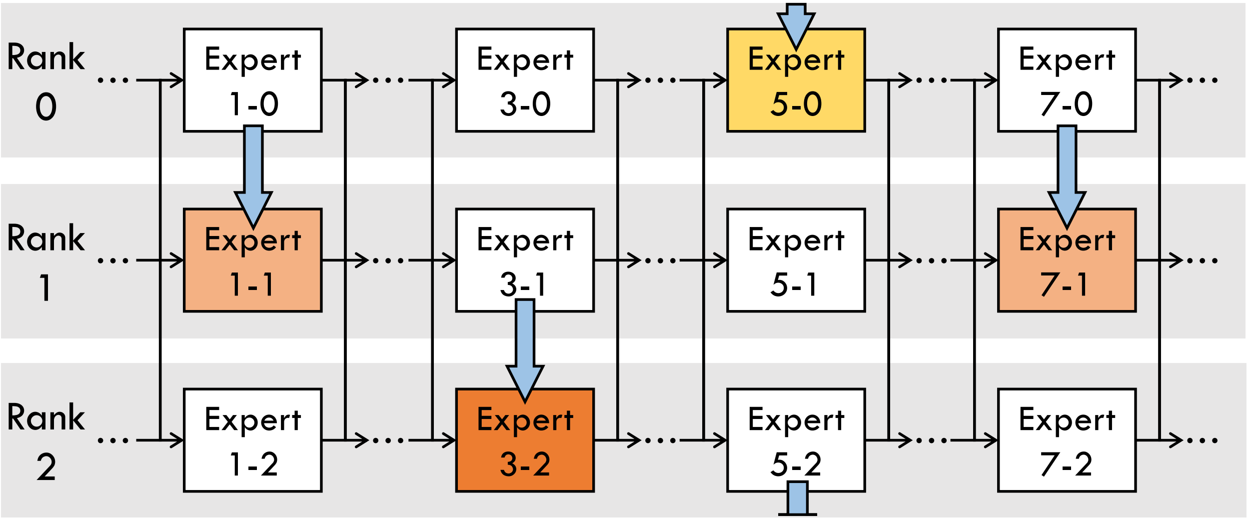
Partial Experts Checkpointing (PEC)

Iteration: i



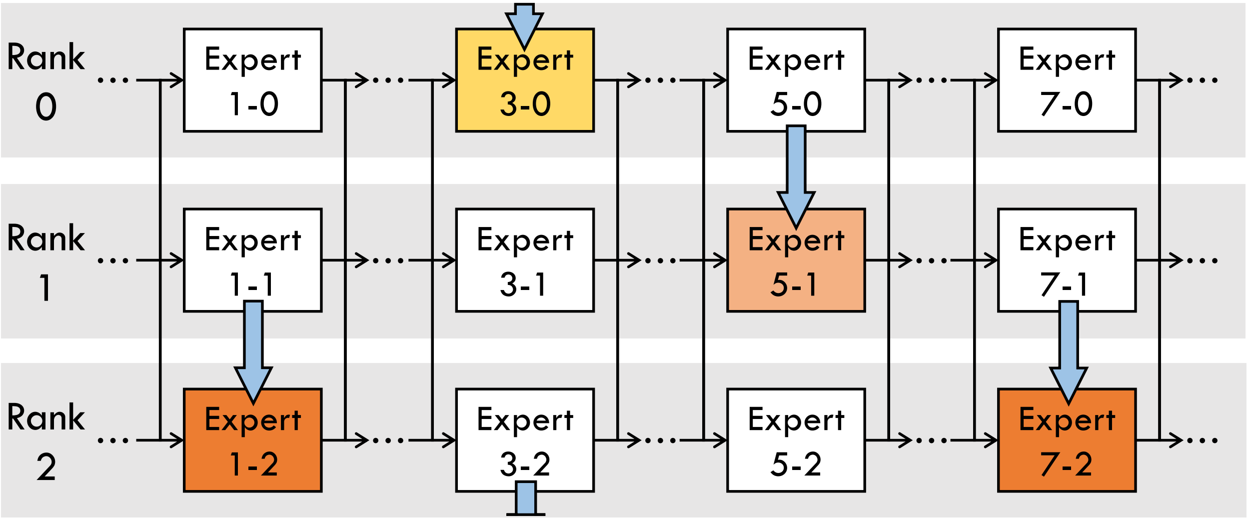
Partial Experts Checkpointing (PEC)

Iteration: i+1



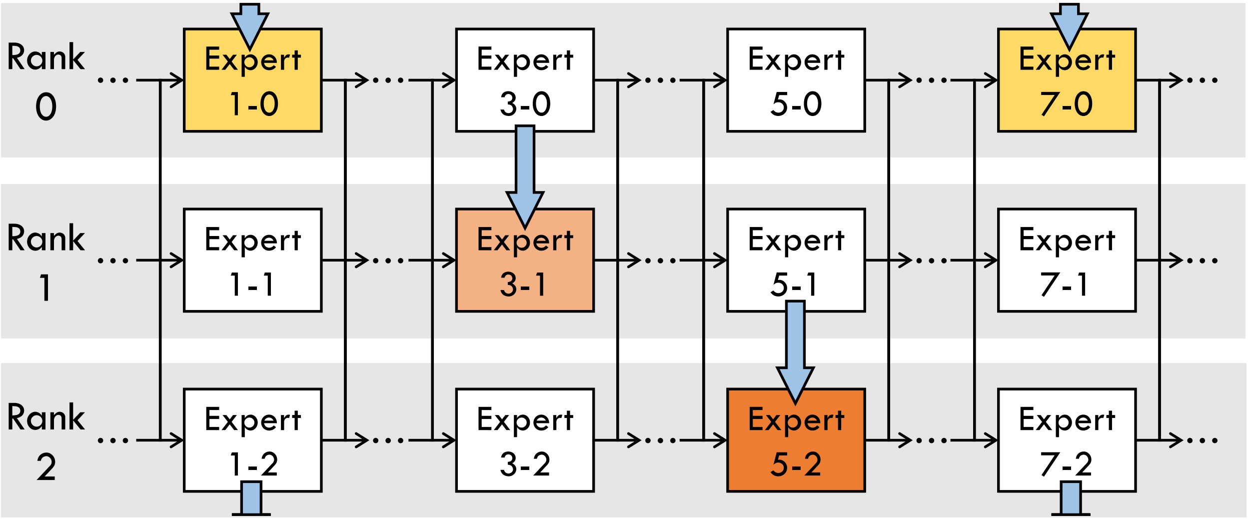
Partial Experts Checkpointing (PEC)

Iteration: i+2



Partial Experts Checkpointing (PEC)

Iteration: i+3



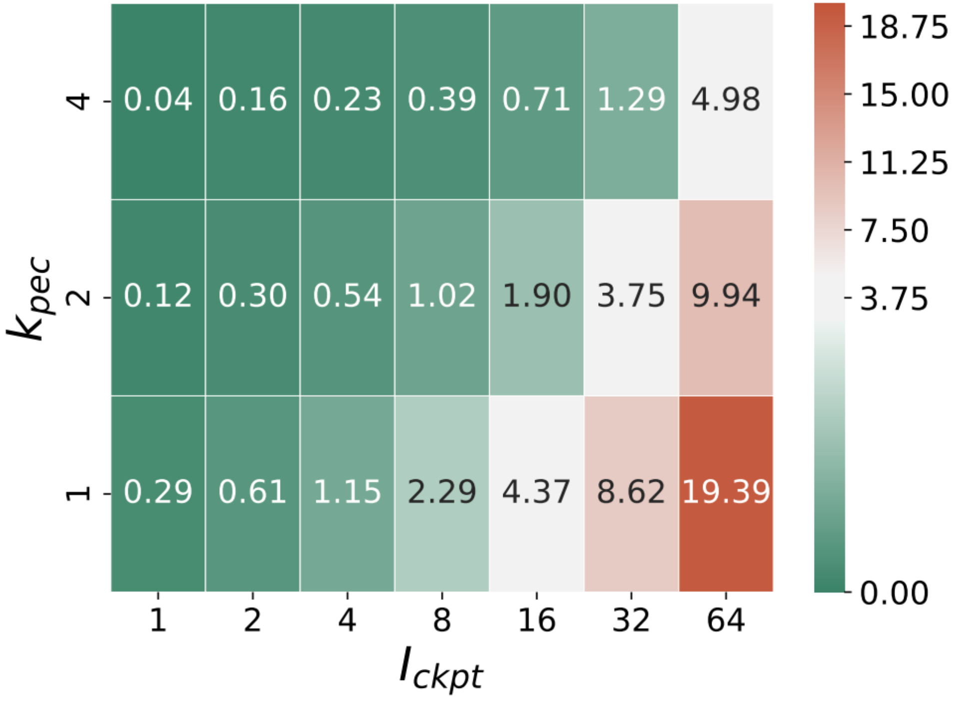
$$C_{full} \approx (P_{ne} + P_e) \cdot (B_w + B_o) \Longrightarrow C_{pec} \approx \left(P_{ne} + \boxed{\frac{K_{pec}}{N}} P_e\right) \cdot (B_w + B_o)$$

Impact on Model Accuracy

Proportion of Lost Tokens (PLT): $$PLT = \frac{1}{N_{moe}} \sum_{i=1}^{N_{moe}} \frac{\sum_{j=1}^{N_{fault}} L_{i,j}(I_{ckpt}, K_{pec}, F)}{T_i \cdot TopK_i}$$

Impact on Model Accuracy

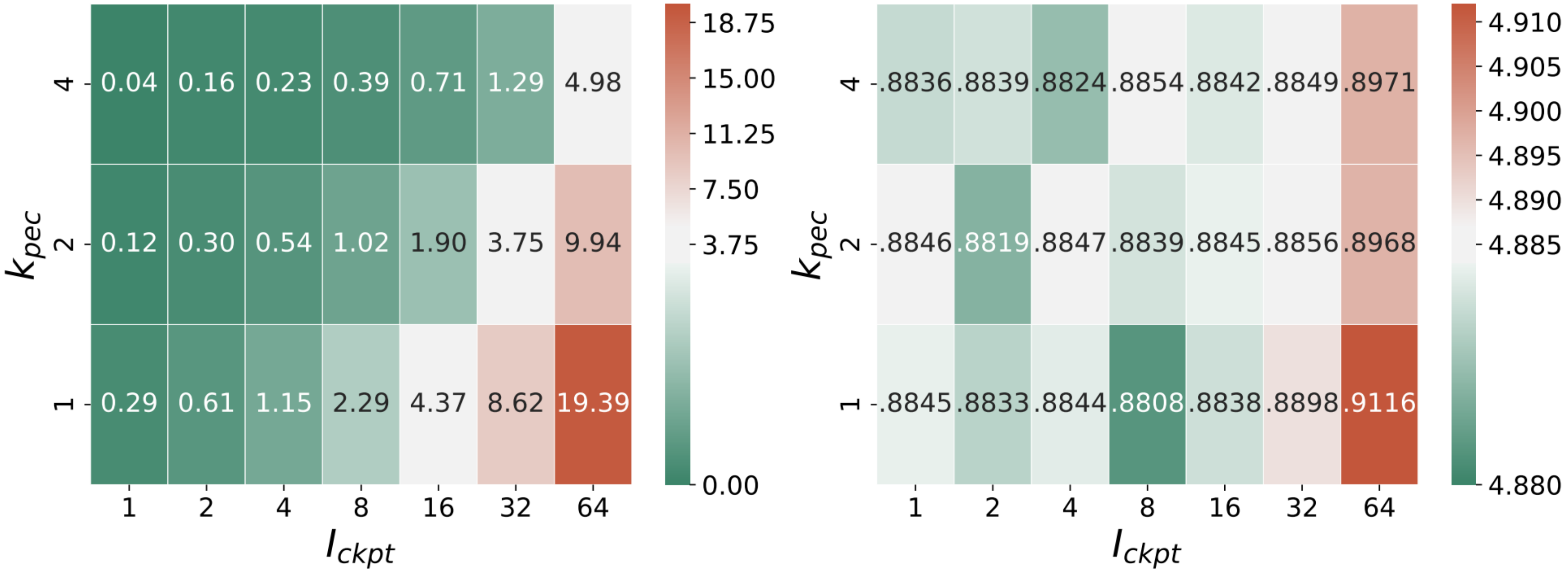
Proportion of Lost Tokens (PLT): $PLT = \frac{1}{N_{moe}} \sum_{i=1}^{N_{moe}} \frac{\sum_{j=1}^{N_{fault}} L_{i,j}(I_{ckpt}, K_{pec}, F)}{T_i \cdot TopK_i}$



(a) PLT (%)

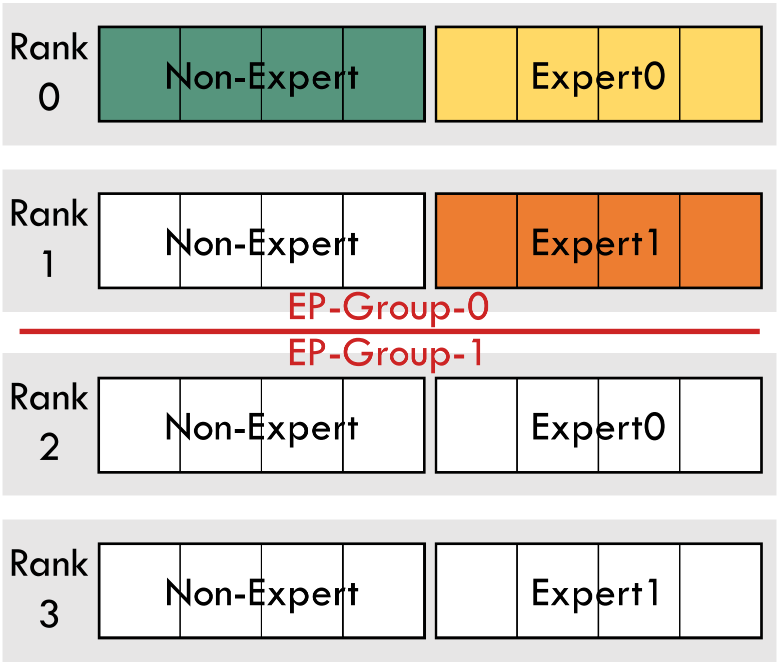
Impact on Model Accuracy

Proportion of Lost Tokens (PLT): $$PLT = \frac{1}{N_{moe}} \sum_{i=1}^{N_{moe}} \frac{\sum_{j=1}^{N_{fault}} L_{i,j}(I_{ckpt}, K_{pec}, F)}{T_i \cdot TopK_i}$$



(a) PLT (%) (b) Validation loss

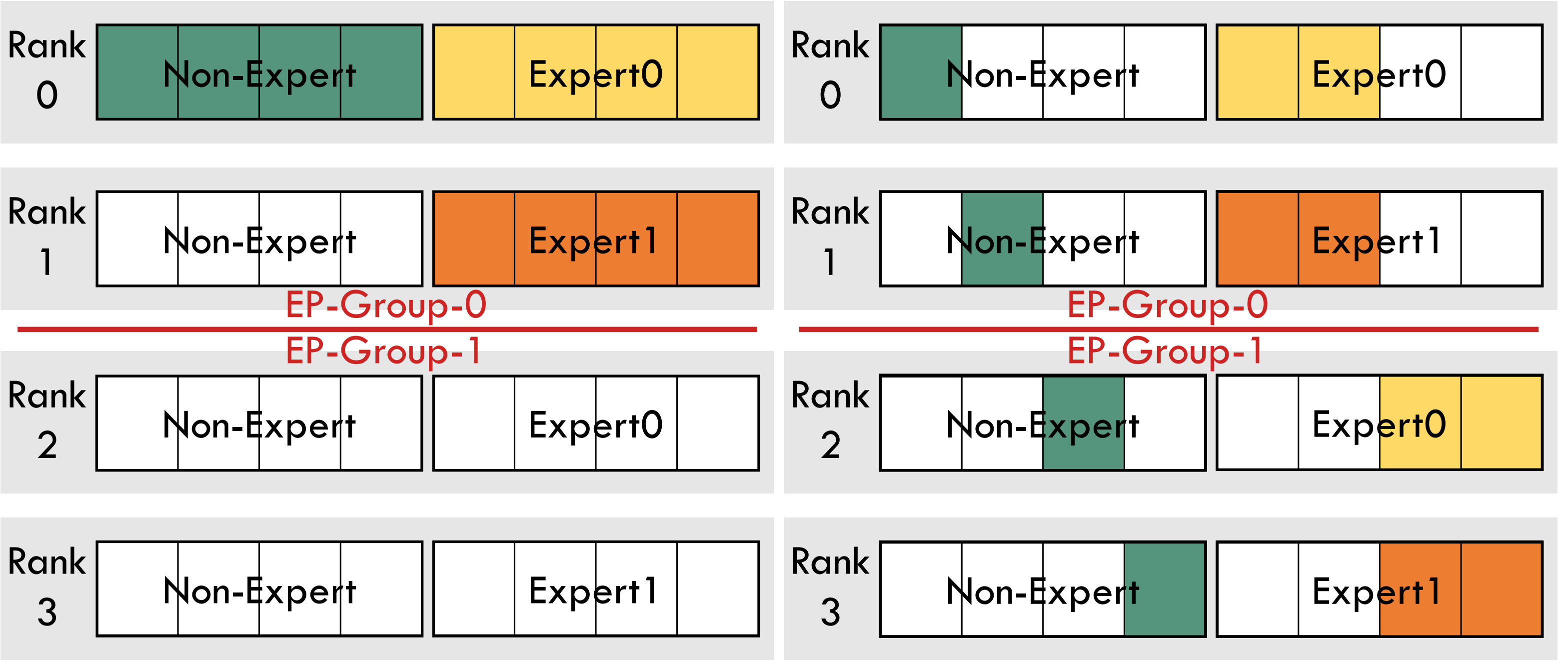
Fully Sharded Checkpointing



(a) Baseline

Fully Sharded Checkpointing

- Equal Sharding for Expert Part
- Equal Sharding for Non-Expert Part

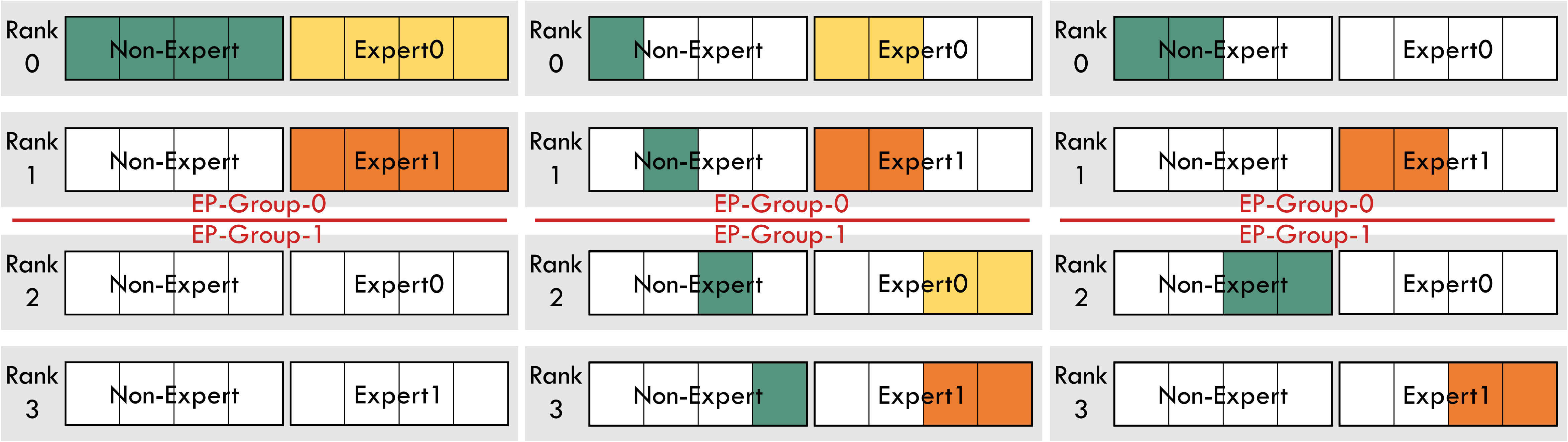


(a) Baseline

(b) Equal Sharding for Non-Expert Part

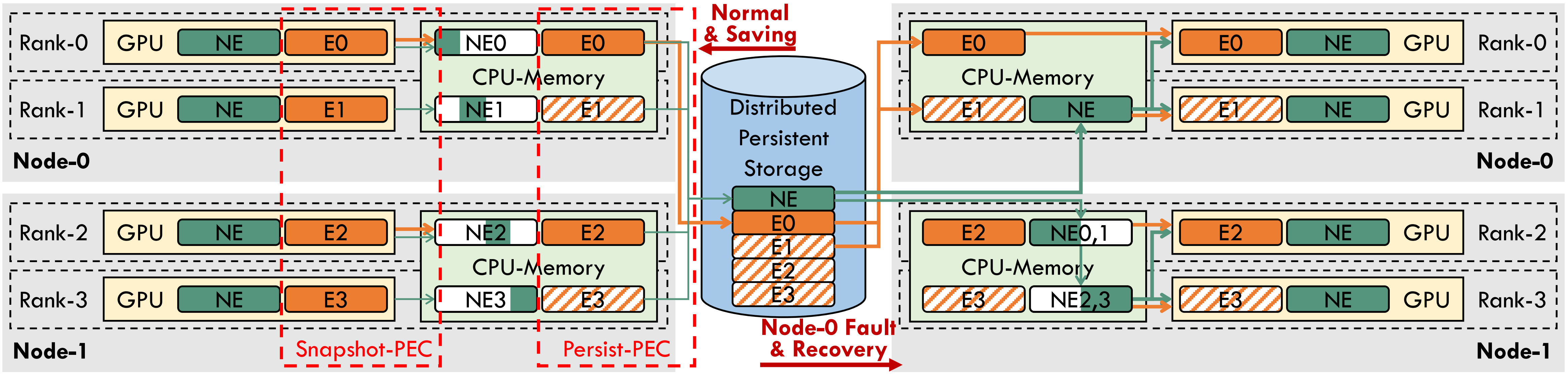
Fully Sharded Checkpointing

- Equal Sharding for Expert Part
- Equal Sharding or Adaptive Sharding (with PEC) for Non-Expert Part



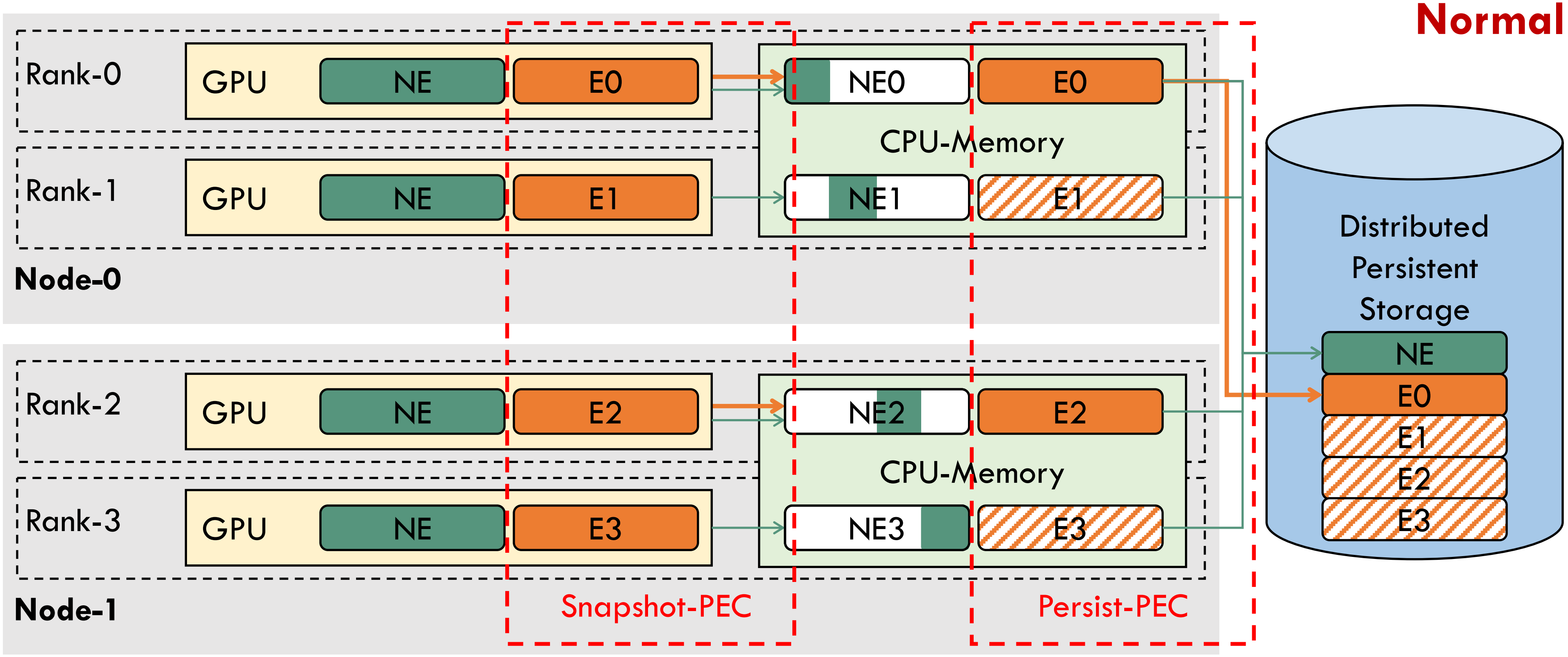
(a) Baseline (b) Equal Sharding for Non-Expert Part (c) Adaptive Sharding for Non-Expert Part

Two-Level Checkpointing Management



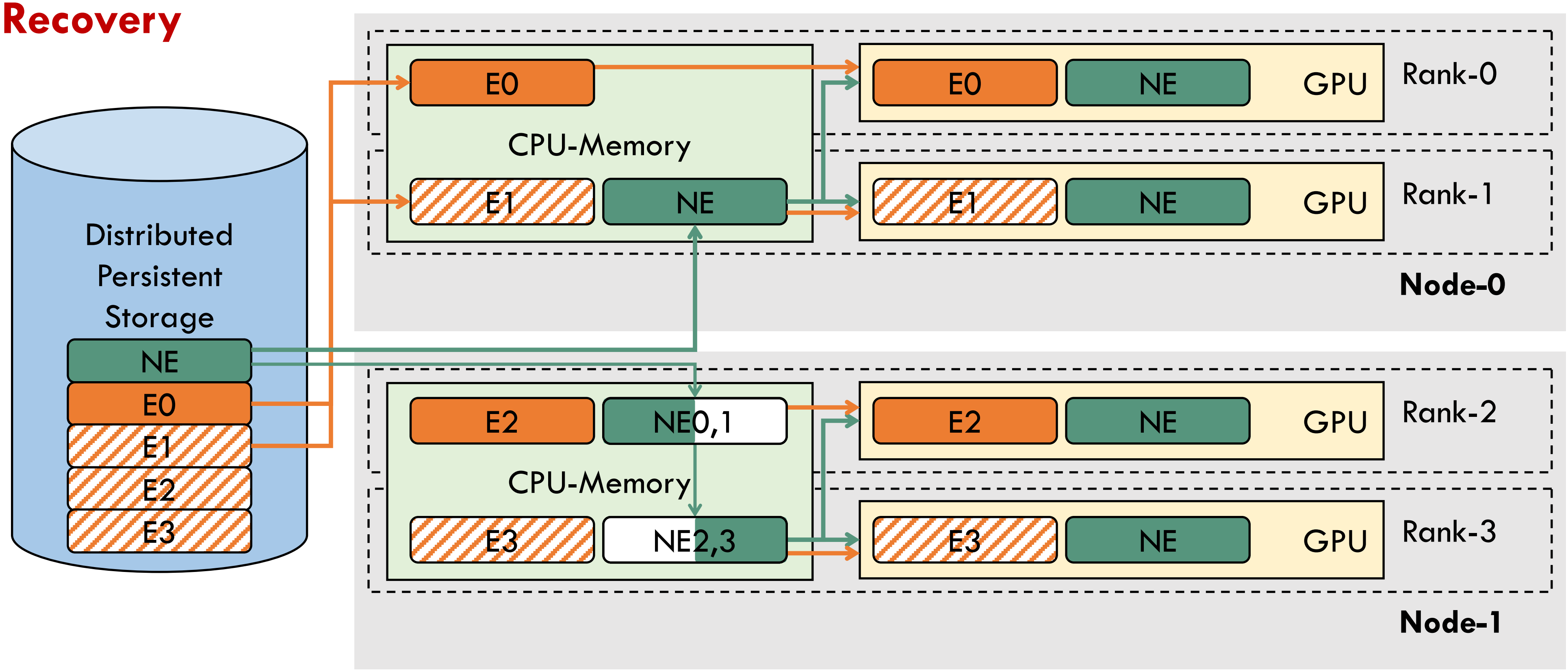
Two-Level PEC Saving

Normal & Saving

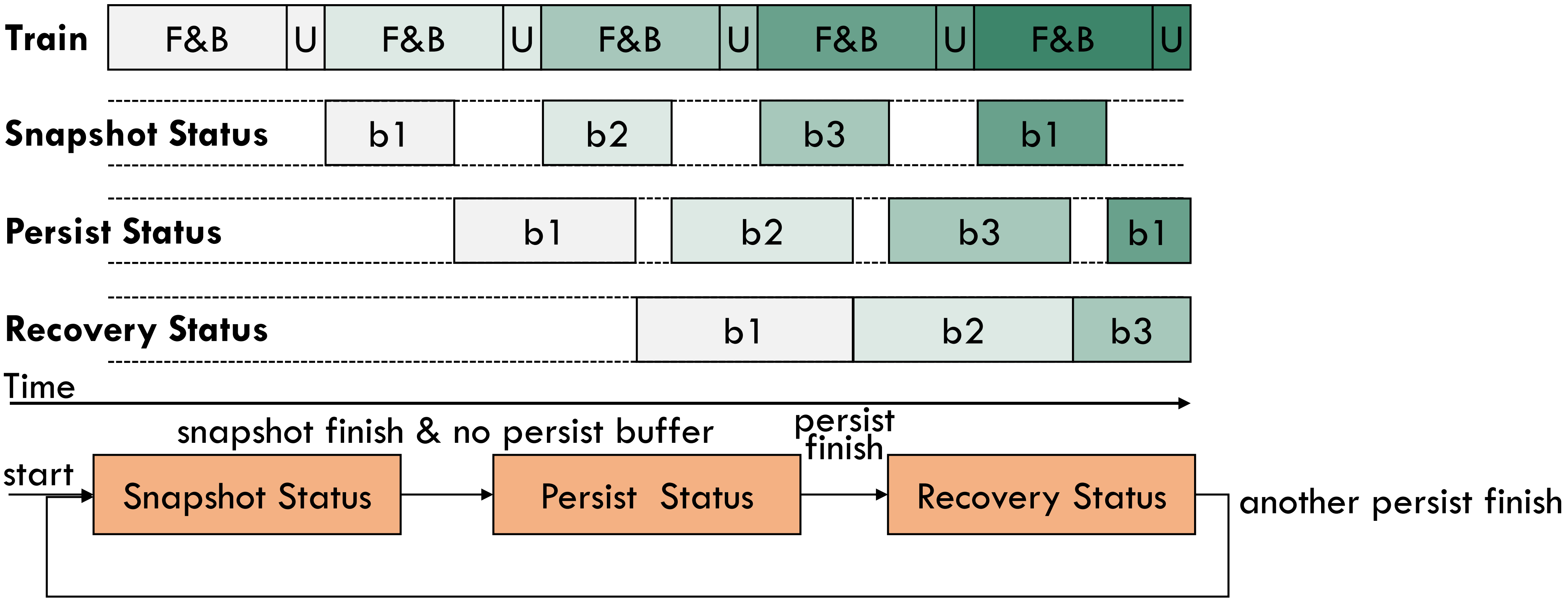


Two-Level PEC Recovery

Node-0 Fault & Recovery



Async Checkpointing with Triple-Buffer



Evaluation

- **Real-World Testing**
  - Megatron-DeepSpeed Framework [7]
  - 8-16 A800-SXM4-80GB GPUs
- **Simulation Testing**
  - ASTRA-SIM Simulator [8]

Table 1. Hyperparameters for experimental MoE models.

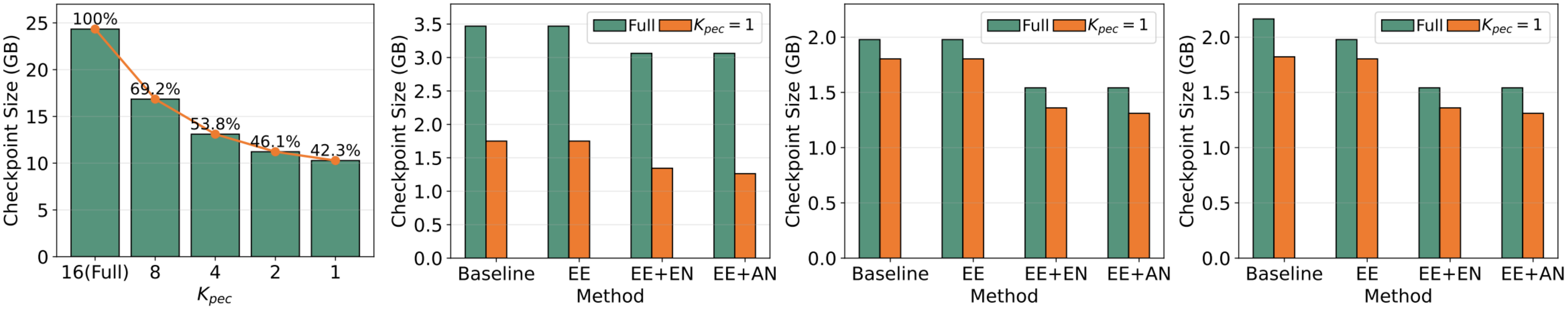
Parameter	GPT-125M-8E	GPT-350M-16E	SwinV2-MoE
Num. layers	12	24	[2, 2, 18, 2]
Hidden size	768	1024	96
Num. atten. heads	12	16	[3, 6, 12, 24]
Num. MoE layers	6	12	10
Num. experts/layer	8	16	8
Num. parameters	323M	1.7G	173M

Table 2. Configurations for GPT-350M-16E model training.

Configuration	Node	GPU	DP	TP	PP	EP	Experts/GPU
Case1	1	8	8	1	1	8	2
Case2	2	16	16	1	1	16	1
Case3	2	16	16	1	1	8	2

[7] Microsoft. 2022. Megatron-DeepSpeed. https://github.com/microsoft/Megatron-DeepSpeed
[8] Rashidi, Saeed, et al. "Astra-sim: Enabling sw/hw co-design exploration for distributed dl training platforms." 2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2020.

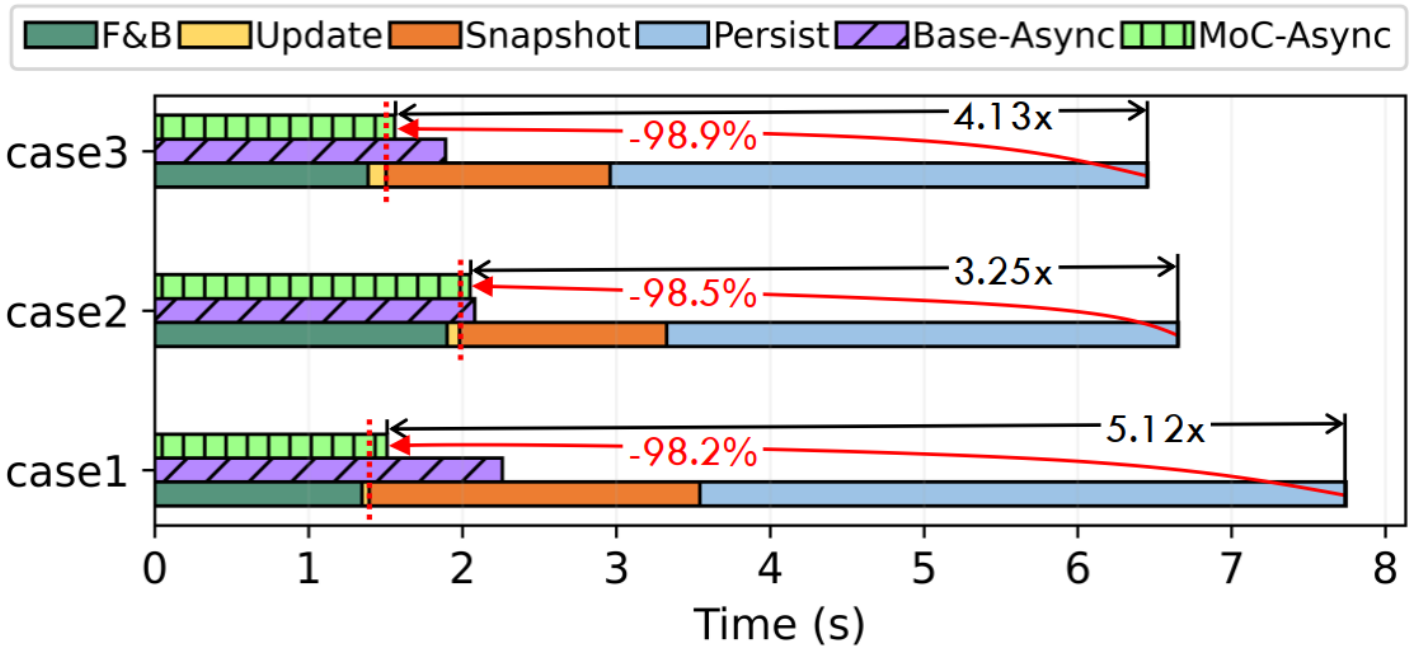
Checkpoint Size



(a) Total Checkpoint Size (b) Bottleneck Rank in Case1 (c) Bottleneck Rank in Case2 (d) Bottleneck Rank in Case3

MoC-System significantly reduces the checkpoint size across different scenarios.

Asynchronous Checkpointing



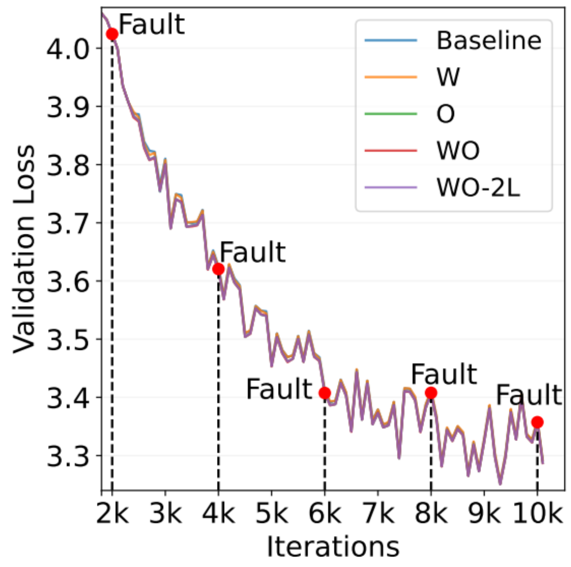
With asynchronous checkpointing, MoC-System can decrease the overhead of each checkpointing process by up to 98.9% and accelerate the training iteration by up to 5.12 times.

Impact on Model Accuracy

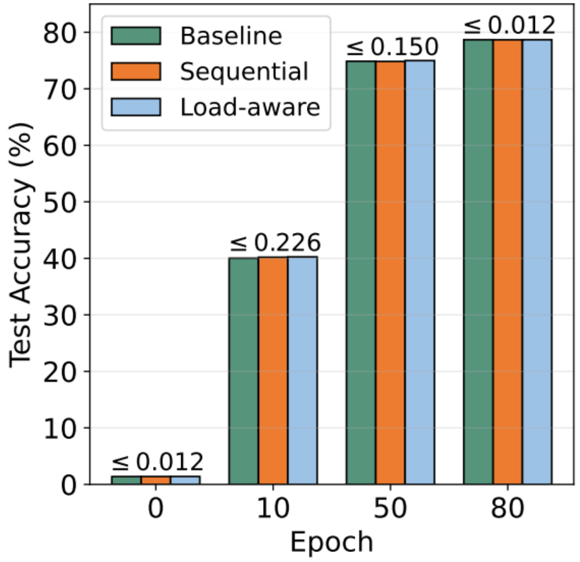
Method	**Ckpt**	**HellaSwag**	**PIQA**	**WinoGrande**	**BoolQ**	**ARC-E**	**OBQA**	**RACE**	**MathQA**	**Avg. (↑)**
Baseline	1	26.85	58.22	49.09	54.77	36.83	13.00	24.21	20.54	35.44
W	0.88	26.92	58.16	49.72	57.52	**37.84**	12.80	24.69	**20.84**	36.06
O	0.54	26.93	58.00	48.54	61.28	37.21	**13.40**	**25.26**	19.97	36.32
WO	0.42	26.91	58.38	49.33	61.31	37.33	13.20	24.50	20.20	36.40
WO-2L	0.42	**26.96**	**58.49**	**50.12**	**61.74**	37.12	13.20	24.40	20.13	**36.52**
Deviation	-	(0.06, 0.11)	(-0.22, 0.27)	(-0.55, 1.03)	(2.75, 6.97)	(0.29, 1.01)	(-0.20, 0.40)	(0.19, 1.05)	(-0.57, 0.30)	(0.62, 1.08)

Impact on Model Accuracy

Method	Ckpt	HellaSwag	PIQA	WinoGrande	BoolQ	ARC-E	OBQA	RACE	MathQA	Avg. (↑)
Baseline	1	26.85	58.22	49.09	54.77	36.83	13.00	24.21	20.54	35.44
W	0.88	26.92	58.16	49.72	57.52	**37.84**	12.80	24.69	**20.84**	36.06
O	0.54	26.93	58.00	48.54	61.28	37.21	**13.40**	**25.26**	19.97	36.32
WO	0.42	26.91	58.38	49.33	61.31	37.33	13.20	24.50	20.20	36.40
WO-2L	0.42	**26.96**	**58.49**	**50.12**	**61.74**	37.12	13.20	24.40	20.13	**36.52**
Deviation	-	(0.06, 0.11)	(-0.22, 0.27)	(-0.55, 1.03)	(2.75, 6.97)	(0.29, 1.01)	(-0.20, 0.40)	(0.19, 1.05)	(-0.57, 0.30)	(0.62, 1.08)



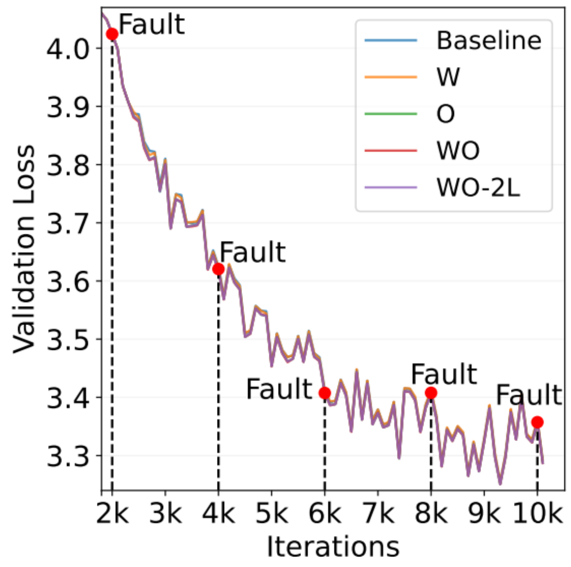
(a) GPT-350M-16E



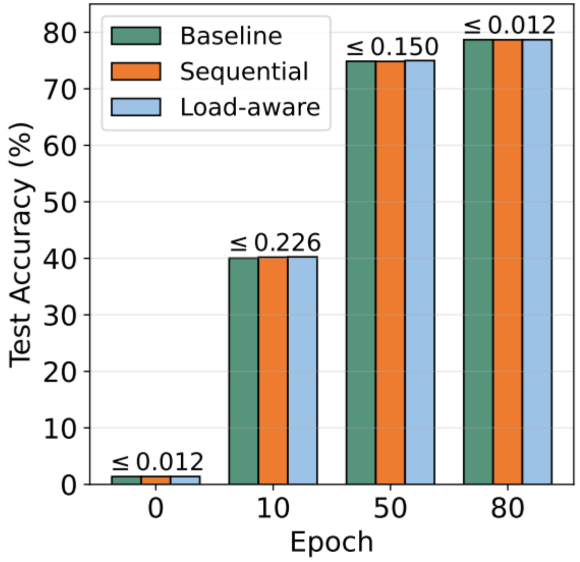
(b) SwinV2-MoE

Impact on Model Accuracy

Method	Ckpt	HellaSwag	PIQA	WinoGrande	BoolQ	ARC-E	OBQA	RACE	MathQA	Avg. (↑)
Baseline	1	26.85	58.22	49.09	54.77	36.83	13.00	24.21	20.54	35.44
W	0.88	26.92	58.16	49.72	57.52	**37.84**	12.80	24.69	**20.84**	36.06
O	0.54	26.93	58.00	48.54	61.28	37.21	**13.40**	**25.26**	19.97	36.32
WO	0.42	26.91	58.38	49.33	61.31	37.33	13.20	24.50	20.20	36.40
WO-2L	0.42	**26.96**	**58.49**	**50.12**	**61.74**	37.12	13.20	24.40	20.13	**36.52**
Deviation	-	(0.06, 0.11)	(-0.22, 0.27)	(-0.55, 1.03)	(2.75, 6.97)	(0.29, 1.01)	(-0.20, 0.40)	(0.19, 1.05)	(-0.57, 0.30)	(0.62, 1.08)



(a) GPT-350M-16E



(b) SwinV2-MoE

Method	HS	PIQA	WG	BQ	ARC	OBQA	RTE	Avg.
Base	57.99	80.52	68.59	74.46	47.27	44.80	54.51	61.16
FT-w.o.E	58.58	81.88	68.51	76.82	48.72	45.20	63.54	63.32
FT-Full	58.34	81.34	70.40	79.11	48.38	45.00	66.06	64.09
FT-PEC	58.78	81.45	70.24	79.17	48.23	45.00	65.58	64.06

Summary

- ☐ **Problem:** MoE models introduce new challenges for existing fault-tolerant strategies, necessitating specific optimizations to enhance efficiency.

Summary

- **Problem:** MoE models introduce new challenges for existing fault-tolerant strategies, necessitating specific optimizations to enhance efficiency.
- **Inspiration:** Expert parameters are insensitive to a limited number of training updates.

Summary

- **Problem:** MoE models introduce new challenges for existing fault-tolerant strategies, necessitating specific optimizations to enhance efficiency.

- **Inspiration:** Expert parameters are insensitive to a limited number of training updates.

- **MoC-System:**
  - Partial Expert Checkpointing to reduce the checkpoint size
  - Fully sharded checkpointing strategies
  - Two-level asynchronous checkpointing for snapshot and persist

Thanks and Questions

Email: wcai738@connect.hkust-gz.edu.cn

"MoC-System: Efficient Fault Tolerance for Sparse Mixture-of-Experts Model Training,"
Weilin Cai, Le Qin, Jiayi Huang, ASPLOS 2025.

香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)