TrainVerify

Equivalence-Based Verification for Distributed LLM Training

SOSP 2025

Yunchi Lu[M, MS] Youshan Miao[MS] Cheng Tan[N, MS] Peng Huang[M] Yi Zhu[MS] Xian Zhang[MS] Fan Yang[MS]

University of Michigan · Microsoft · Northeastern University

Error-prone Model Parallelization

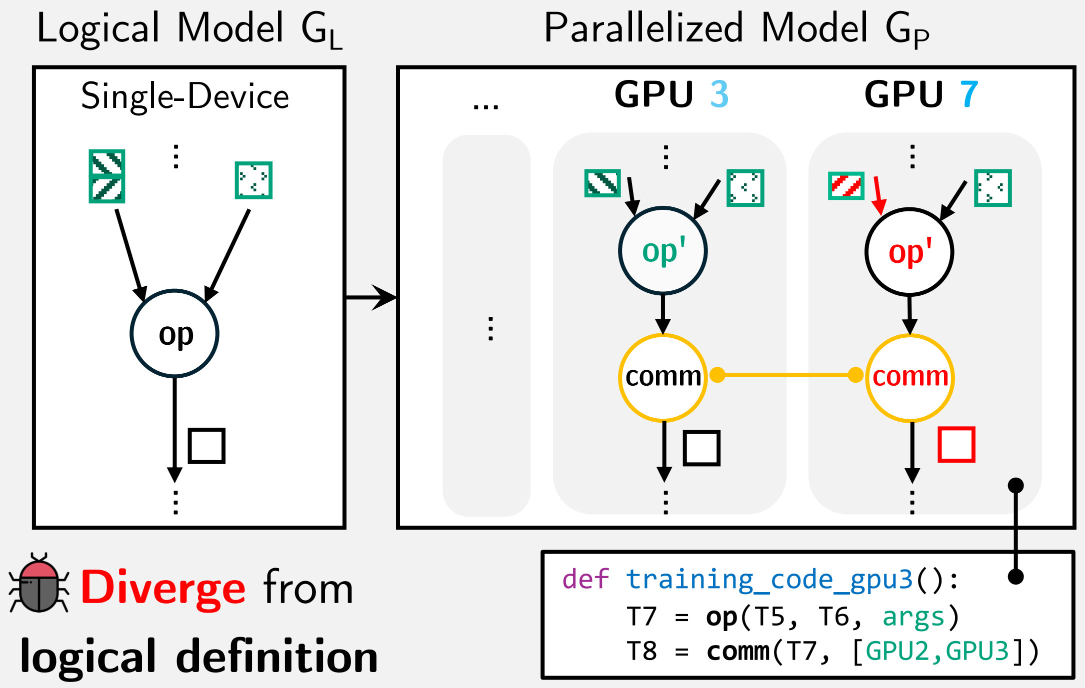
- Large-scale distributed training is *a must*
- Complex parallel training: messy, error-prone

Parallelization Bugs

- A major class of *elusive* and *costly* bugs
- Wrong tensor / op partitioning; communication
- **Common** across training frameworks
- ⚠ Silent, challenging to diagnose

Insight: Corretness lie in Execution Plans

- *Materialized* repr capturing parallelization logic
- Typically structured as Dataflow Graphs



Diverge from **logical definition**

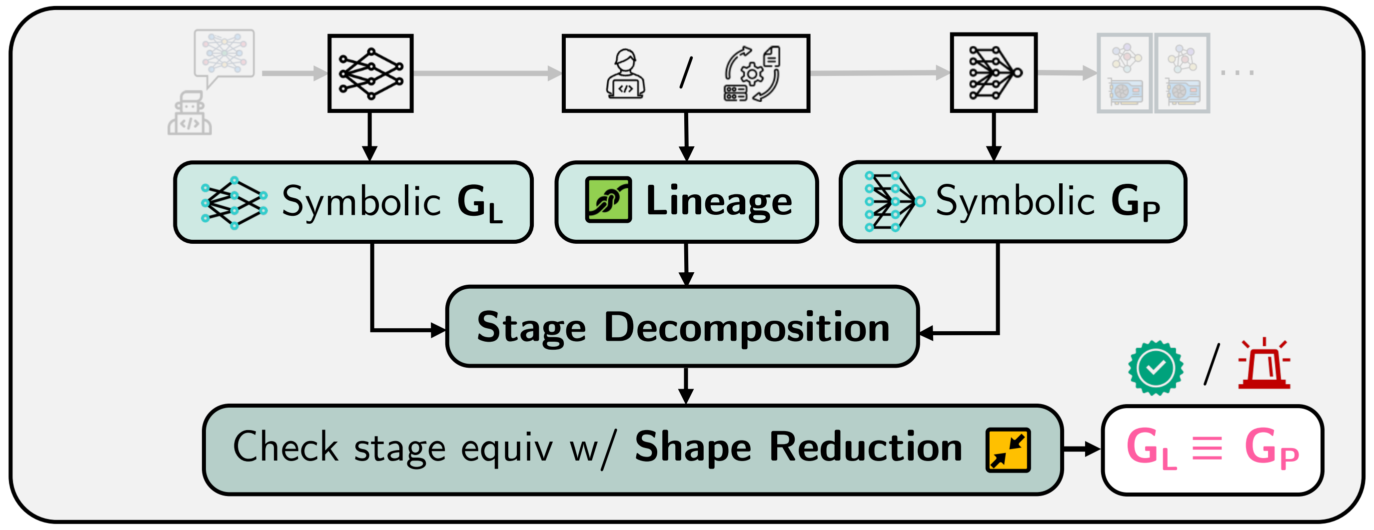
Current Practices & Limitations

- Testing: Undermined by **numerical noise**
- **Full-scale** 1-GPU exec **infeasible** for large models

Goal

- Correctness **guarantee** for parallellization
- **Symbolically** ensure algebraic equivalence between distributed and logical models
- Verification scales to large distributed training

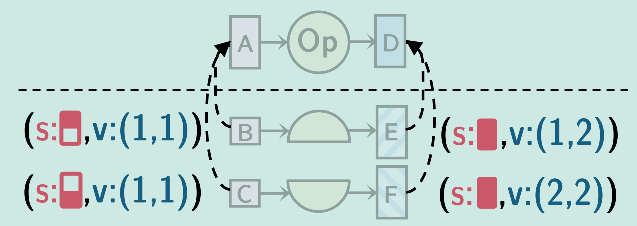
TrainVerify Design



Symbolic DFGs

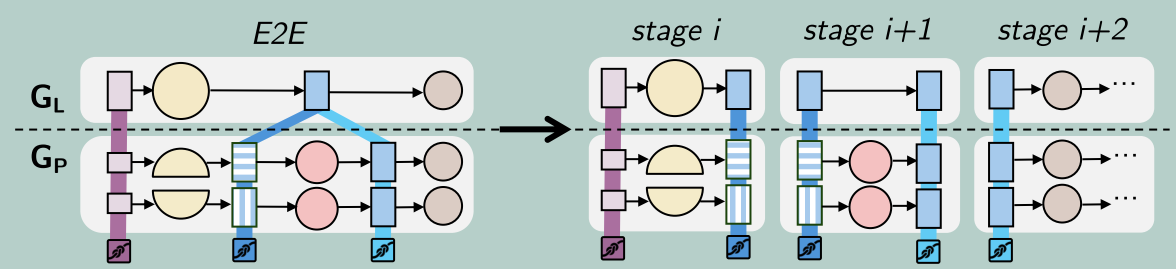
Original graph structure
Tensors w/ **symb elems**
Symbolic operators

Tensor Lineage



Stage Decomposition

- Address complexity from **deep** model architectures
- E2E verification → fine-grained tractable stages
- **Independently verifiable**, executed **in parallel**



- All stages pass ✓; Failed stage → localized bug

Shape Reduction

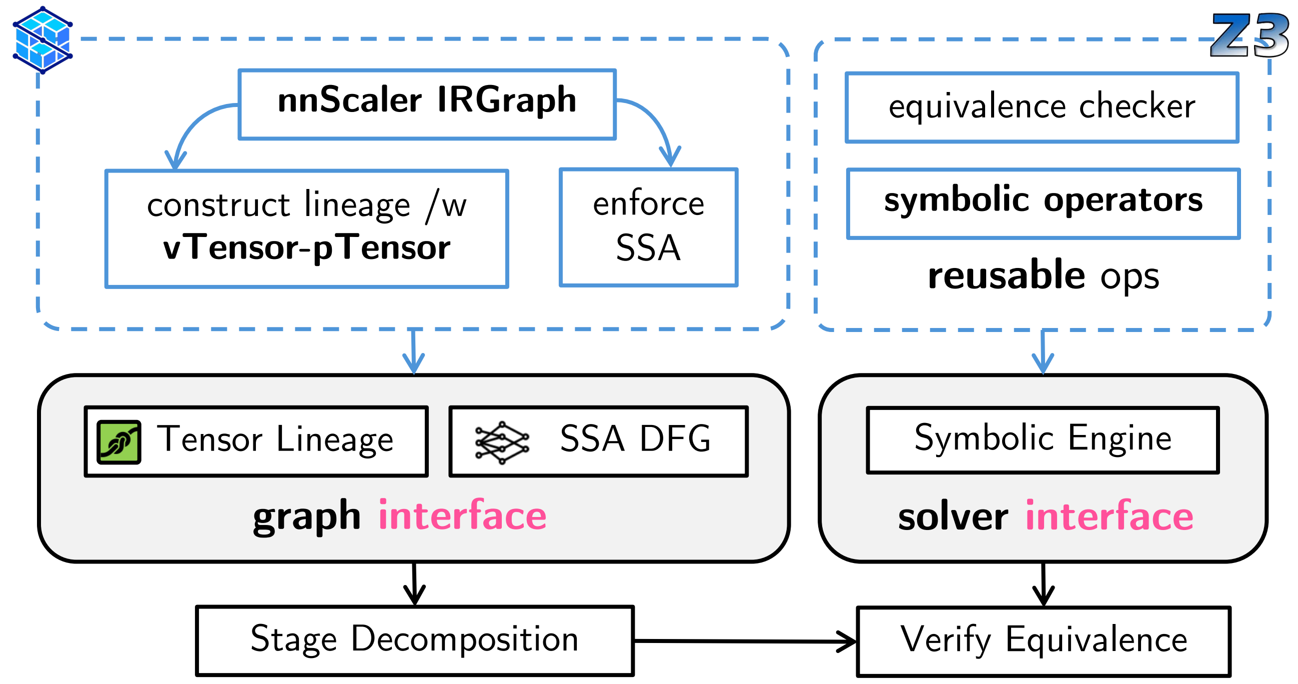
- Address complexity from **large** tensor volumes
- Exploit *repetition* in DNN operators

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

$$c_{1,1} = a_{1,1}\cdot b_{1,1} + a_{1,2}\cdot b_{2,1} + a_{1,3}\cdot b_{3,1}$$
$$c_{2,2} = a_{2,1}\cdot b_{1,2} + a_{2,2}\cdot b_{2,2} + a_{2,3}\cdot b_{3,2}$$

- **Shrink** each dimension
- **Preserve** structural and functional **properties**
- **Verified equivalence** on shape-reduced models **faithfully extends to the original**

Implementation

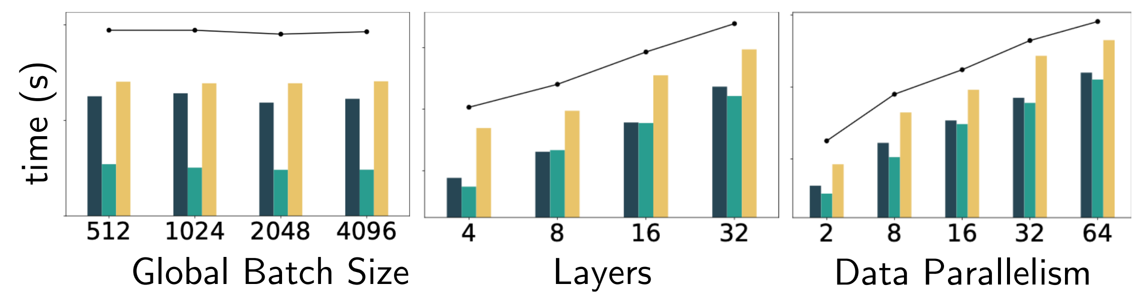


Evaluation

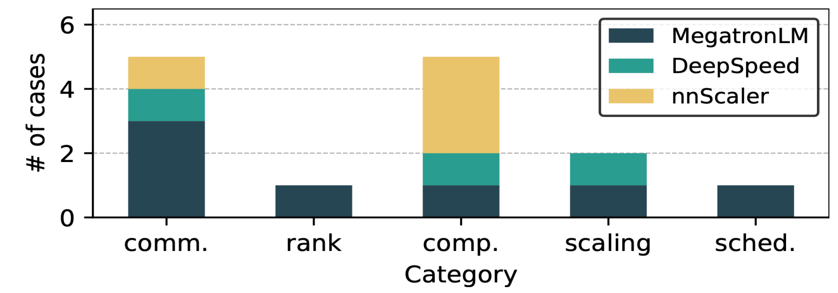
- Executed on **single** 32-core CPU w/ 1TB mem
- Successfully verified the ***largest*** models < 9 h

Model	Size	Para. w/ #GPUs	Verif. Time
Llama3	70B	512	2.4h
	405B	**8192**	**8.0h**
DeepSeek-V3	236B	512	2.4h
	671B	**2048**	**9.0h**

- Verification cost: **invariant** to original tensor shapes; **linear** to layeres and parallelization degrees



- Eliminate broad categories of bugs



- **Catch new bugs**: sharding non-partitionable dims; broken data dependencies; SSA violation