HorizonML: A Decentralized Framework for Parallel Model Training

Independent Study — Spring 2025

Name: Deeptanshu Sankhwar
Advisor: Prof. Eric Keller

Motivation: Breaking Centralized AI Control

Problem: ML training is monopolized by a few companies with large GPU farms (OpenAI, Google, Anthropic).

Centralization leads to:

- Biased models
- Closed access
- Rent-seeking compute APIs

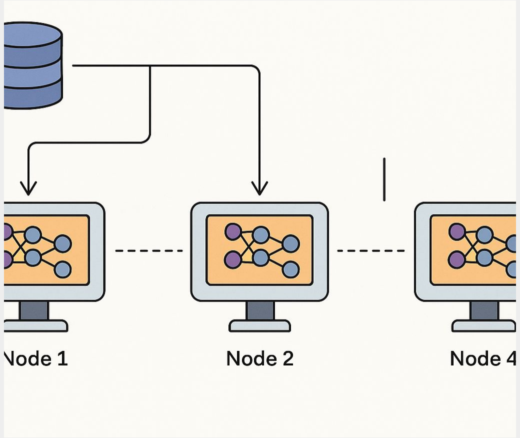
Vision: Democratize model training by distributing the process across independently operated nodes, without trusting any single entity.

Related Work

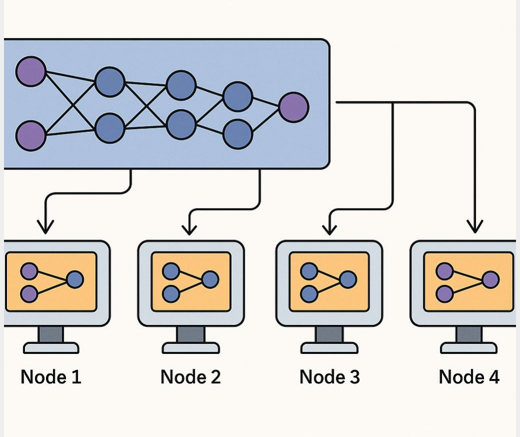
Approach	Key Idea	Limitation
Data Parallelism (PyTorch DDP)	Split batches, sync gradients	Still centralized orchestrator
Model Parallelism (GPipe, PipeDream)	Split model layers	Not decentralized, still server-bound
Bittensor	Proof-of-Intelligence network	Narrow to inference rewards, limited training structure
Proof-of-Learning	Verifiable training claims	Still experimental, not integrated into training infra

HorizonML Architecture Overview

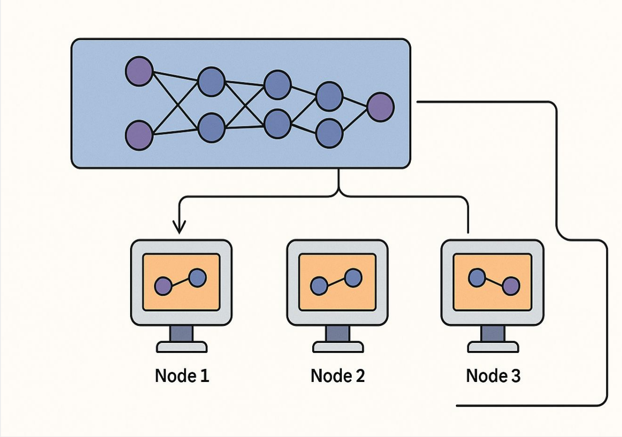
Data-Parallel Training



Layer-Parallel Training



Tensor-Parallel Training



Implementation Setup

Model: ResNet (split across nodes)

Dataset: CIFAR-10 + ImageNet (subset)

Workers: 5 simulated Docker nodes

Parallelism Strategies:

- Data parallelism
- Layer-wise model parallelism
- Tensor-wse model parallelism

Communication Stack:

- REST (baseline)

Data Parallelism

Each node trains full model on data shard

Problems:

Requires full model memory on each node

Noisy gradients due to small batch sync

High communication cost

Layer-wise Model Parallelism

Each node handles **a slice of the model**

Forward/backward pass is distributed

Advantages:

Lower memory footprint

No gradient aggregation = more stable convergence

Easy to decentralize

Tensor-wise Model Parallelism

Forward/backward pass is **distributed within a layer**

Results from all nodes are **aggregated after partial tensor computation**

Advantages

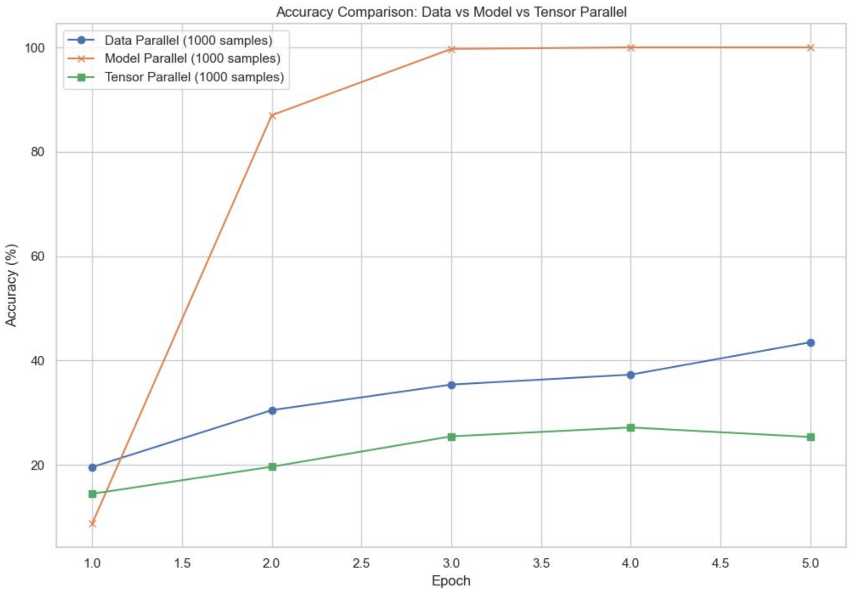
✅ Lower memory footprint

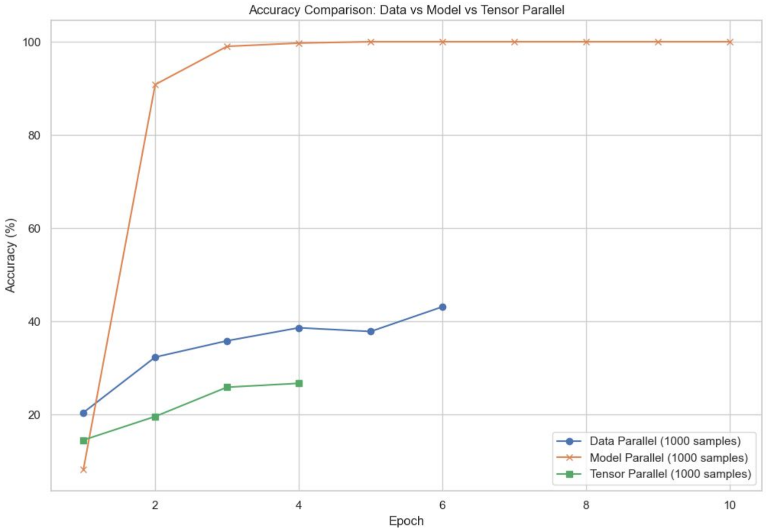
✅ Enables fine-grained parallelism **within large layers**

✅ Scales across resource-constrained nodes

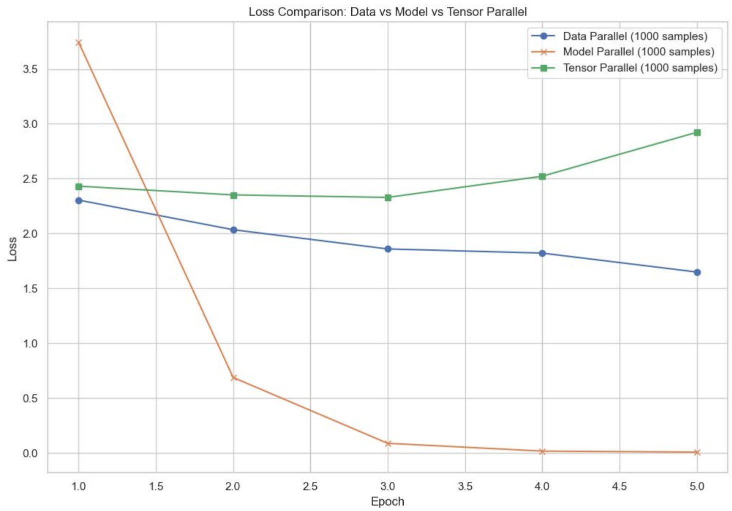
✅ Compatible with transformer-like architectures

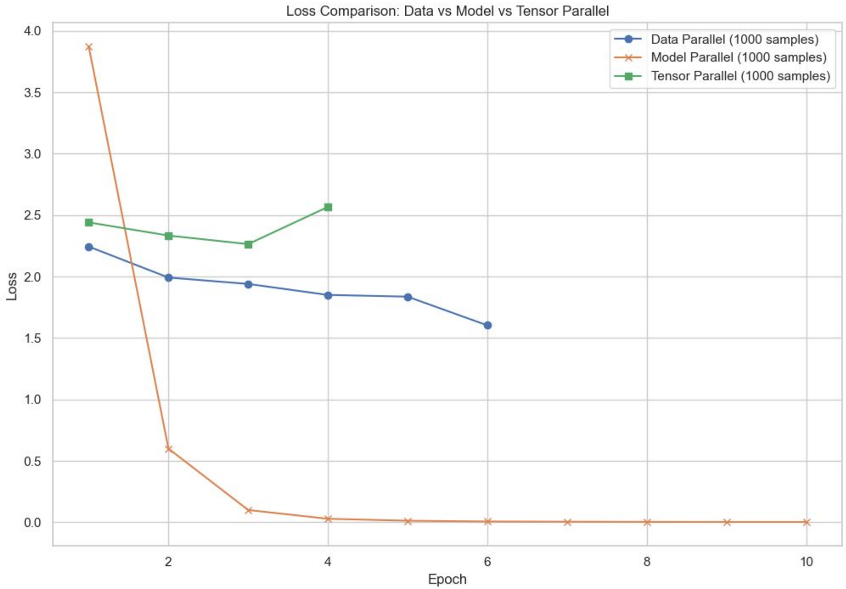
Head-to-Head Comparison



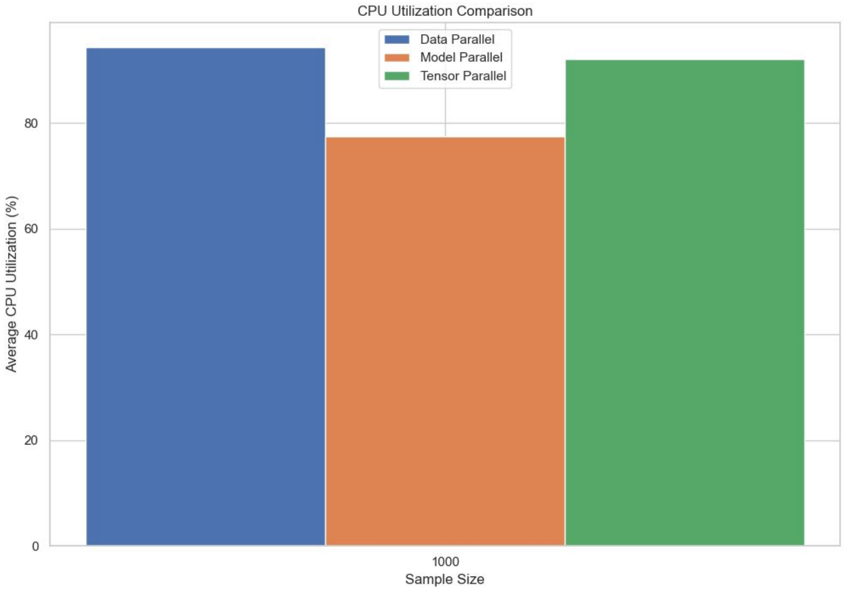


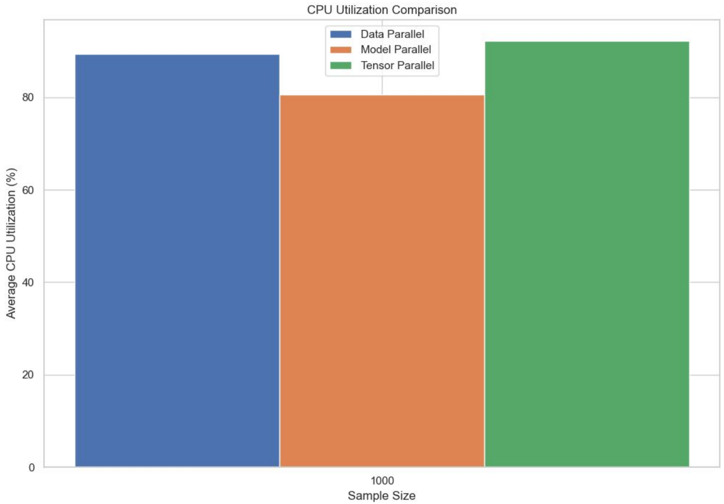
Head-to-Head Comparison



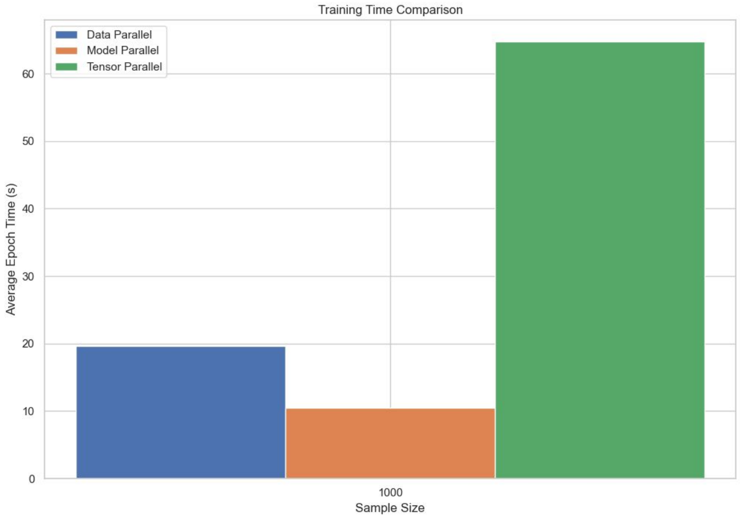


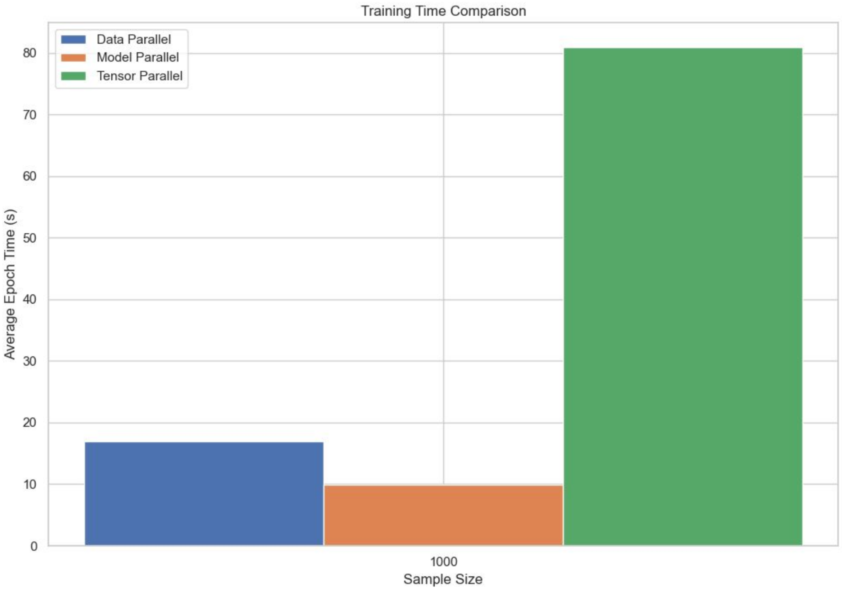
Head-to-Head Comparison





Head-to-Head Comparison





What We Achieved

Created a modular training framework that:

Runs across independent nodes

Enables **model splitting without centralized coordination**

Demonstrated **better training quality** using model parallelism under decentralized assumptions

What's Next: Towards a Decentralized AI Chain

Introduce **blockchain layer** for:

- Logging training contributions
- Rewarding correct updates (Proof of Learning / ZKML)

Transform nodes into **agents**:

- Compete or collaborate on training
- Earn reputation or tokens

Move toward a **trust-minimized model training protocol**

Vision: HorizonML as a Base Layer for Open AI Infrastructure

Our study focuses on evaluating the market landscape, consumer trends, and competition pertinent to the new product.

“Just as Ethereum democratized finance, HorizonML aims to democratize intelligence.”

Open training economy	Permissionless participation	On-chain coordination, off-chain training	Foundation for autonomous AI networks

Conclusion

Reimagined distributed training as a decentralized protocol

Built & evaluated competing strategies

Validated model split approach as more scalable and stable

Set the groundwork for training coordination without centralized compute monopolies

Thank you!