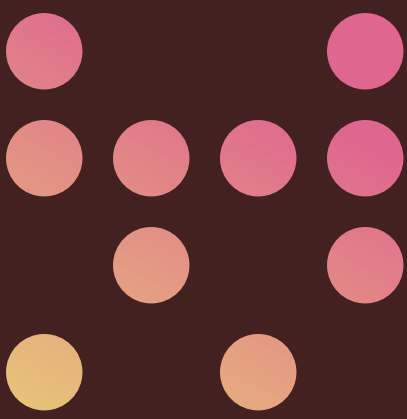




TrainingWorxSM

by MythWorx

Enterprise AI at Scale: A Modern Approach to Cost-Efficient Model Training

- MINIMAL GPU INFRASTRUCTURE
- COST-EFFICIENT
- MODULAR ARCHITECTURE
- INCREMENTAL AI TRAINING

Introduction

Enterprises today are sitting on **unprecedented volumes** of proprietary text, documents, communications, and operational data.



Unlocking the value of this information requires large language models (LLMs) capable of understanding internal terminology, domain-specific workflows, and the unique language patterns of the business. The challenge is that training or continuously updating such models is traditionally expensive—often prohibitively so.

Large-scale dense model training has historically demanded massive GPU clusters, long uninterrupted training windows, and high operational risk.

Even a 7-billion-parameter model can cost hundreds of thousands of dollars (or more) to train when the dataset spans hundreds of billions of tokens. Organizations looking to remain competitive by continuously improving their AI often find themselves constrained by infrastructure availability and budget realities.

The Challenge of Traditional Dense Training

Traditional LLM training treats the entire model as a single, indivisible entity

Every parameter is loaded, activated, and updated for every batch of data. When the dataset is large — such as an enterprise corpus spanning 500 billion internal tokens — this approach becomes extremely resource-intensive.

Dense training requires multi-GPU parallelism to achieve usable throughput

In practice, this means clusters of 64, 128, or even more GPUs operating simultaneously, all connected via high-speed networking. These clusters are costly on the cloud and expensive to purchase and maintain on-premises.

Furthermore, dense training is brittle. A failure in the middle of training, a node loss, or a system interruption often forces the model to revert to a recent checkpoint, losing some or all progress. The operational burden increases as the scale grows, making dense training both financially and operationally risky.

➡ Finally, dense training offers limited flexibility.

To expand a model's capacity, the entire network must be rebuilt and retrained. To incorporate new information, the model must be fine-tuned or retrained in a way that risks overwriting existing knowledge. In short, dense training is costly, rigid, and difficult to adapt to real-world enterprise needs.

How TrainingWorxSM Changes the Landscape

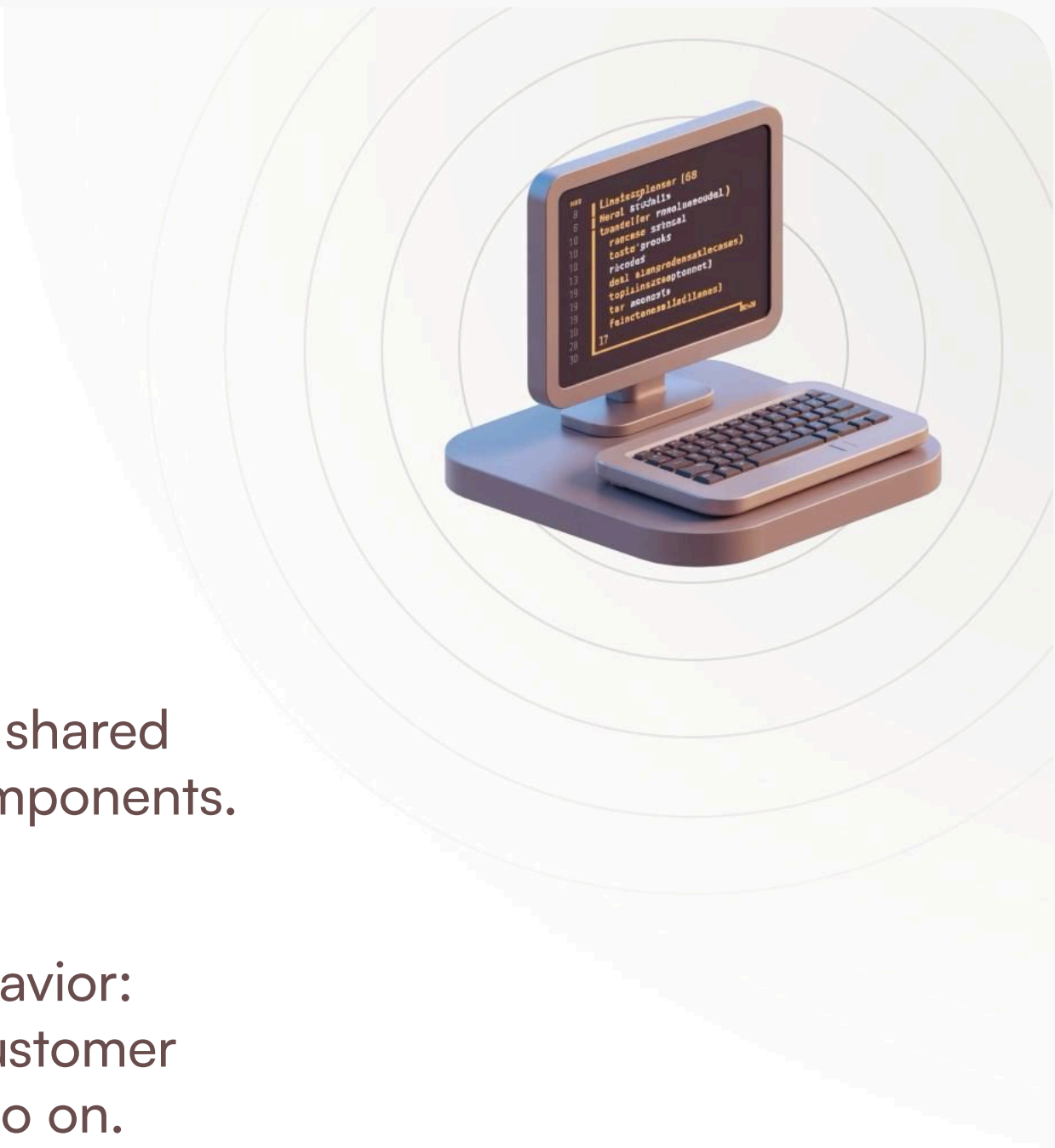
TrainingWorxSM takes a fundamentally different approach



Instead of training all parameters together, TrainingWorxSM organizes the model into a shared trunk and a library of modular “expert” components.



These experts handle domain-specific behavior: legal language, biomedical terminology, customer service patterns, financial reasoning, and so on.



Under TrainingWorxSM, the training process is broken into sequential steps:

Training one expert at a time

Only the shared trunk and a single expert module are loaded and updated at any given moment. This dramatically reduces memory pressure and allows training on a small number of GPUs.

Progressively adding new experts

Each expert learns from a different slice of the enterprise’s internal data, resulting in a distributed understanding of multiple domains.

Aligning experts into a coherent whole

After training, lightweight alignment layers ensure that the experts work together smoothly.

Teaching a routing mechanism

Finally, a router learns to activate the right experts at the right time during inference.

The result is a unified model whose effective capacity can exceed that of much larger dense models, without the associated training cost.

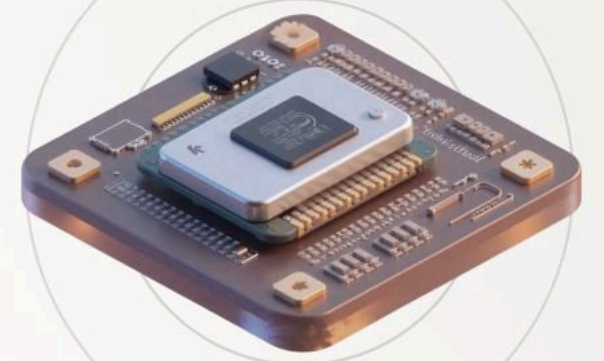
Comparison Chart

Below is a comparison of traditional dense training versus TrainingWorxSM across the dimensions that matter most to enterprises training a 7B model on 500 billion internal tokens.

	Traditional Dense Training 7B Model, 500B Tokens	TrainingWorx SM Modular Training 7B Model, 500B Tokens	TrainingWorx SM Advantage
Peak Compute Requirement	Full 7B active per step. Requires multi-GPU high-memory nodes (A100/H100).	Only trunk + one expert active. Fits on 1—2 GPUs with CPU/NVMe offloading.	Cost / Hardware
Memory Footprint	All parameters + optimizer states must fit in VRAM. High memory pressure.	Parameters, grads, optimizer states sharded across CPU/NVMe. Small active footprint.	Infrastructure Cost
Training Throughput	Full gradient updates across 7B parameters for all 500B tokens. Cluster required for speed.	Each expert trained on a subset of the 500B tokens. Sequential jobs. Lower cost.	Cost / Flexibility
Total Compute Cost	High. Compute scales with full 7B per token across 500B tokens.	Lower. Compute scales with expert size (50—150M) + trunk, not full model.	Cost Efficiency
Scaling Beyond 7B	Requires retraining or upgrading to 13B/34B/70B models.	Add more experts to expand overall capacity without new dense model.	Capacity / Flexibility
Domain Specialization	Single FFN per layer must learn all domains. Cross-domain interference.	Experts can specialize per domain. Minimal interference.	Capacity / Specialization
Catastrophic Forgetting	Likely when training sequentially on large new datasets.	Very low. Earlier experts remain untouched.	Reliability
Update Frequency	Expensive. Requires large fine-tunes or partial retraining.	Frequent updates possible. Train new experts cheaply and integrate.	Speed / Flexibility
Failure Recovery	Single monolithic job. Restart required if failure occurs.	Only re-run the affected expert. Training is modular.	Risk Reduction
Inference Latency	Fixed cost for 7B FFNs per token.	Near 7B latency. Only top-k experts activated.	Efficiency
Inference Capacity	Fixed at 7B performance.	Behaves like a 40B—200B model due to expert library.	Capacity
Data Pipeline Fit	Designed for static, pre-built datasets.	Optimized for streaming data (Kafka/Flink) and continuous ingestion.	Engineering Flexibility
Operational Complexity	Simpler architecture. Complex cluster orchestration.	Modular architecture. Simpler hardware footprint, staged jobs.	Operational Flexibility
Long-Term Scalability	Scaling requires building bigger dense models.	Add experts indefinitely. Lifelong learning path.	Future-Proofing

Why TrainingWorxSM Matters for Enterprises

TrainingWorxSM **dramatically lowers the financial barrier** for model training while creating new opportunities for domain specialization and continuous improvement.



Instead of investing in massive clusters, enterprises can build models incrementally, using smaller equipment and smaller training cycles.

Moreover, TrainingWorxSM is future-proof

When new data or new requirements arise, the organization can simply train new expert modules and incorporate them into the existing model. There is no need for expensive full-model retraining, and no risk of overwriting previous knowledge.

By turning model expansion into an incremental process rather than a monolithic one, TrainingWorxSM empowers organizations to scale their AI capabilities as their business grows — without carrying the weight of traditional training costs.

Cost Comparison

7B Model, 500B Tokens

* This comparison reflects GPU infrastructure costs only.

On-Premises H100 Cluster Training

Purchasing a 64-GPU cluster requires a \$2M+ upfront investment. Factoring in rack infrastructure, power, cooling, and staff over a three-year amortization, a single training cycle effectively costs \$2.5M. While this improves long-term asset utilization, it demands massive capital expenditure and introduces significant operational complexity compared to more flexible cloud alternatives.

Cloud H100 Cluster Training

Standard cloud pre-training for a 7B model on 500B tokens costs roughly \$500,000 in GPU fees alone. With networking, storage, and orchestration, total expenses typically reach \$650,000 to \$900,000. While offering high throughput, this 128-GPU approach carries significant financial and operational risk compared to more efficient alternatives.

TrainingWorxSM Training on Minimal GPU Equipment

TrainingWorxSM slashes 7B model costs to ~\$75,000 by training one expert module at a time. This sequential approach limits peak requirements to a single expert plus the trunk, enabling training on 12 × H100 or A100 GPUs. On-premises setups further reduce expenses for alignment and router calibration.

Unlike dense training, TrainingWorxSM does not force the enterprise to gamble a large budget on a single long training run.

TrainingWorxSM jobs can be paused, resumed, re-run, or expanded incrementally as new data becomes available.

Scenario	Hardware	Resource Count	Cost Basis	Estimated GPU Infrastructure Cost
On-Premises H100 Dense Training	64× H100 GPUs	Owned equipment	CapEx + 3-year ops	~\$2.5m
Cloud H100 Dense Training	128× H100 GPUs	Cluster	~\$8 blended on-demand rate	~\$1.5m
TrainingWorx SM Training on Minimal Equipment	12× H100 GPUs	Cloud or on-prem	~\$8 blended on-demand rate	~150k

Conclusion

TrainingWorxSM by MythWorx is a breakthrough approach for enterprises seeking to unlock **the full value of their internal data** without incurring the prohibitive costs of traditional dense model training.

By enabling modular, sequential training with minimal hardware, TrainingWorxSM reduces costs by more than

90%

while providing a scalable, flexible path for long-term model evolution. It transforms how enterprises train, update, and maintain large language models—making advanced AI accessible to organizations of all sizes.