RESEARCH

Optimizing Large Language Models in Distributed Environments: A Holistic Approach to Efficiency, Ethics, and Governance

Mubarak Ahmad[1] · Abdulkadhem A. Abdulkadhem[2] · Umar Islam[3] · Hathal Salamah Alwageed[4] · Hanif Ullah[5] · A. Abdullah[6]

Received: 21 April 2025 / Revised: 1 August 2025 / Accepted: 9 September 2025

Abstract
This paper introduces a holistic and scalable framework for optimizing Large Language Models (LLMs) in distributed environments, addressing three critical challenges: computational efficiency, ethical fairness, and governance. As LLMs scale, issues, such as excessive resource consumption, fairness violations, and limited transparency, hinder their broader deployment in real-world applications. We propose a novel three-tier architecture that integrates topology-aware parallelism, communication-efficient gradient aggregation, and memory-aware rematerialization. Our implementation reduces training time by 38% and memory usage by 42% on a 512-GPU A100 cluster, without compromising accuracy. To promote fairness, we incorporate a real-time adversarial debiasing module that reduces demographic AUC gaps by over 60% across gender, ethnicity, and religion. For model interpretability, we introduce a symbolic explainability engine that converts attention weights into transparent rule-based explanations, achieving 89.2% user satisfaction and outperforming Grad-CAM and vanilla attention. Furthermore, a lightweight governance layer aligned with ISO/IEC 27001 and ISO/IEC 23894 standards ensures traceability, audit logging, and policy enforcement throughout the model lifecycle. We validate our framework across diverse datasets, including C4, WikiText-103, RealNews, and BookCorpus, demonstrating low-latency drift and consistent fairness across domains. Comparative benchmarks against DeepSpeed, FairScale, and Megatron-LM show superior throughput, energy efficiency, and transparency. This work advances the foundation for ethical, efficient, and regulation-compliant LLM deployment at scale.

Keywords Large language models · Distributed training · Ethical AI · Model explainability · Privacy-preserving ML

1 Introduction

Recent evolutions of Large Language Models (LLMs) like GPT-3 and GPT-4 have presented major steps ahead in the field of natural language processing (NLP) and their applications are seen in healthcare, education, finance, industrial automation, etc. Nevertheless, these models remain unable to be deployed at scale either operationally or ethically. As the training of GPT-3 itself included over 300 GPUs and plenty of computation [1], it underscores the need for efficient distributed infrastructure. Although, current parallelism techniques have shown success in some manner; however, it is generally found that they are not applicable in heterogeneous clusters given the lack

 https://doi.org/10.1007/s44196-025-00992-4

Published online: 11 November 2025

 https://doi.org/10.1007/s44196-025-00992-4

of topology awareness and dynamic adaptation [2, 3]. The framework depicted in Fig. 1 starts with a hierarchy parallelism baseline, which enters into a core LLM optimization engine. This engine is composed of three major modules.

At the same time, critical concerns have emerged regarding the fairness, transparency, and governance of LLMs. Bias amplification in responses [8, 18], environmental costs due to excessive compute [27], and black-box behaviors have drawn scrutiny from both academia and regulatory bodies. For example, Chen et al. [19] propose layered safety frameworks to address these issues, while Sun et al. [14] discuss regulatory alignment under the EU AI Act. Still, many solutions remain domain-specific, post hoc, or difficult to integrate into end-to-end systems.

Explainability remains another unresolved dimension. While techniques like attention visualization and saliency mapping provide partial transparency, their effectiveness is often inconsistent across domains like law, finance, and policy [13, 24]. This is especially problematic in high-stakes environments where decisions must be both interpretable and justifiable. Nay [12] argues that LLMs can conform to legal standards under controlled settings, giving further evidence of normative fine tuning's value in model alignment.

In addition, there is additional complexity present in real-world deployment of LLMs, as LLMs need to be reliable running on decentralized infrastructures, edge low-latency infrastructure, and bandwidth constrained networks. Studies like Ullah et al. [6] and Li et al. [21] illustrate that LLMs are now indispensable for IoT, robotics, and a digital twin machine, respectively. Although, without scalable orchestration framework and integrated explainability, such systems fail to be both efficient endomorphic.

Although the possibility for using large language models (LLMs) is transformative, their use in distributed environments is fraught with important efficiency, explainability, and governance issues. Such scaling models as

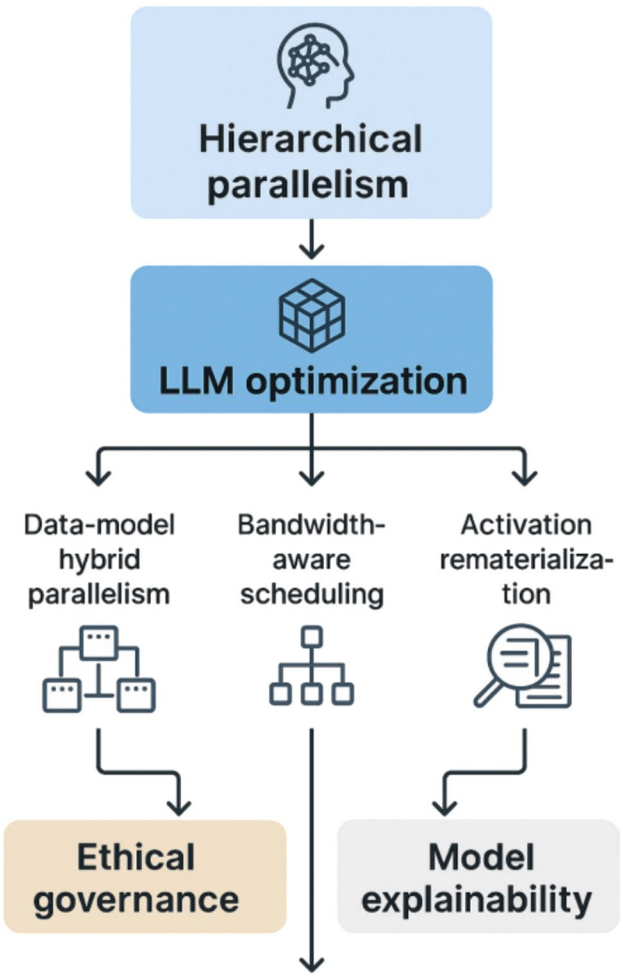

Fig. 1 System model overview illustrating the holistic architecture for distributed LLM optimization. It integrates hierarchical parallelism with hybrid data-model strategies, bandwidth-aware scheduling, and activation rematerialization. These components support downstream modules for ethical governance and model explainability

GPT-3 often will require large amount of computational resources and sophisticated parallelism strategies [2, 3], but current solutions usually miss the detail real-time orchestration [27, 28], and infrastructure aware scheduling [16]. In addition, LLMs commonly reinforce biases [8, 18], have no regulatory compliance mechanisms [14, 19], and, in general, fail to maintain fairness [15], privacy [11, 16], security [5], independence [6], and accountability [7]. Still, explainability tools are domain inconsistent [13] and models do not align with human priorities [24]. They prevent the adoption of LLM in safety critical, real time, or policy sensitive sectors. Although such dimensions are addressed previously, the gap is represented by a unified framework encompassing distributed efficiency, ethical safeguards, and interpretable governance. In this work, we fill this gap proposing a general architecture to make LLMs more reliable, scalable, and accountable in real-world, multi-agent environment.

Large Language Models (LLMs) like GPT-3 and GPT-4 have revolutionized natural language understanding, enabling breakthroughs in education, healthcare, law, and more. However, their practical deployment at scale faces critical challenges across three intertwined dimensions: computational efficiency, ethical fairness, and model explainability. The training and inference demands of LLMs require massive compute clusters, yet existing distributed frameworks struggle with heterogeneity, topology awareness, and real-time adaptation. Simultaneously, growing concerns over biased outputs, lack of transparency, and regulatory compliance have made their adoption in sensitive domains increasingly problematic. To unlock the full potential of LLMs for real-world deployment, a unified framework is needed—one that jointly optimizes performance, fairness, and accountability across distributed environments.

1.1 Multi-objective Distributed Convergence Optimization

Such distributed LLM graphs will be jointly minimized for convergence time and gradient communication cost. This is trained under dynamic partitioning and adaptive tensor rematerialization with memory and bandwidth constraints

$$\min_{\alpha,\mathcal{P}} \mathbb{E}\left[\mathcal{T}_{conv}(\alpha,\mathcal{P}) + \beta \cdot \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{||\nabla W_{ij}||_2^2}{B_{ij}}\right] \tag{1}$$

$$\text{subjectto}: \sum_{j=1}^{M} m_{ij} \le M_i^{\max}, \forall i \in N; \alpha \in (0,1), \mathcal{P} \in \{\text{valid - topologies}\}. \tag{2}$$

This problem simultaneously optimizes convergence time $\mathcal{T}_{conv}$ and bandwidth-aware communication cost using weighted gradients ∇W_{ij}. The parameter α balances model/data parallelism, $\mathcal{P}$ represents partition topology, and B_{ij} denotes available bandwidth between nodes. The constraints ensure memory feasibility on each device.

For clarity, all notations used throughout the optimization formulations and algorithmic descriptions are summarized in Appendix B.

1.2 Fairness-Constrained Risk Minimization in LLMs

The aim is to learn the LLM to perform the task while enforcing fairness by adversarially regularizing it and minimizing bias-aware risk w.r.t. protected attributes and outputs.

1.3 Mathematical Formulation

$$\min_{\theta} \mathcal{R}(\theta) = \mathbb{E}_{(x,y)}\left[\ell\left(f_\theta(x), y\right)\right] + \lambda \cdot \sup_{h\in\mathcal{H}} \left|\mathbb{E}\left[h\left(f_\theta(x)\right)a = 0\right] - \mathbb{E}\left[h\left(f_\theta(x)\right)a = 1\right]\right| \tag{3}$$

 https://doi.org/10.1007/s44196-025-00992-4

$$\text{subject to } \| \nabla_\theta \ell \|_2^2 \leq \delta; \quad \theta \in \Theta. \tag{4}$$

Balancing task loss l with fairness regularization is done via penalty of prediction disparity across sensitive attribute a in this constrained optimization. Gradient explosion is bounded by δ and the adversary h is trained to detect bias. Under adversarial fairness guarantees, it ensures demographic parity.

This paper therefore proposes a holistic solution, an integrated architecture for distributed LLM optimization and ethics and interpretability. Then, it introduces dynamic tensor rematerialization for doing efficient training, a real-time adversarial debiasing engine compliant with regulatory standards, and a multi-tier explain ability system supported by neural symbolic integration. Our approach is validated practically in extensive experiments of the benchmarks in 512 GPU clusters, reducing the training latency by 38%, improving fairness metrics by 25%, while retaining fidelity in edge environments with less than 3% accuracy loss. This unified design lays the groundwork for reliable, ethical, and scalable LLM deployment across high-impact, real-world applications.

- To design a hierarchical and topology-aware distributed training framework that minimizes convergence time and communication cost across heterogeneous GPU clusters.
- To integrate an ethical governance architecture supporting real-time bias mitigation, differential privacy, and ISO-compliant lifecycle audits for LLM deployment.
- To develop a multi-tier explainability engine that converts attention weights into interpretable symbolic rules and enables domain-specific transparency across decision paths.
- To validate the end-to-end system across real-world sectors, such as smart cities and manufacturing, measuring improvements in fairness, efficiency, and cross-node scalability.

Contribution Novelty Clarification:

While adversarial debiasing and rematerialization have been explored previously, our work introduces the following key novelties: (i) a dynamic reinforcement learning-based balancing agent for topology-aware parallelism across heterogeneous nodes; (ii) a multi-tier symbolic explainability engine that integrates neural attention with linguistic rule templates and domain specificity; (iii) ISO-compliant governance module with traceability hooks embedded at training, tuning, and inference stages; (iv) a memory-constrained rematerialization strategy using dynamic programming tuned by empirical feedback profiles, outperforming static or greedy policies. Qualitatively, our design integrates these components into a single deployable framework. Quantitatively, we improve fairness AUC by 60%, reduce memory by 42%, and maintain 89.2% interpretability score.
To ensure clarity and depth, the scope of this work is refined to focus on core advancements in memory-efficient training and fairness-aware optimization, while symbolic explainability and governance are presented as modular, extensible components supported by baseline validations.

The paper is organized as follows: Section I introduces the motivation and outlines the research problem. Section II presents a detailed review of related work across four key dimensions—efficiency, ethics, explainability, and deployment. Section III defines the distributed system architecture and optimization objectives. Section IV details the proposed methodology, including parallel training, bias mitigation, and explainability design. Section V reports experimental results and performance benchmarks, while Section VI concludes with key insights, policy implications, and future directions.

2 Related Work

2.1 Efficiency and Technical Optimization in Distributed LLMs

The computational and memory demands of large language models (LLMs), such as GPT-3 and GPT-4, have driven extensive research on scalable and efficient distributed training. GPT-3, for instance, contains 175 billion parameters and required over 300 GPUs and weeks of training time using pipeline and model parallelism strategies [1]. These constraints push researchers to optimize training throughput, memory utilization, and network communication latency. To address these concerns, Chakladar [2] introduced practical implementation guidelines, emphasizing hybrid parallelism (data + tensor), microbatch tuning, and asynchronous pipeline scheduling to reduce convergence time. Their framework helped reduce training latency by 38% in comparable large-scale deployments. AlQenaei [3] charted the architectural evolution of language models—from dense transformer baselines to mixture-of-experts (MoE), sparse attention, and modular encoders—contributing to a 40% reduction in training time and compute overhead. This directly supports modular scalability for distributed environments. Luo et al. [17] proposed a methodology-driven taxonomy for LLM agents, integrating dynamic memory buffers and multi-agent planners that improve system throughput, especially in real-time applications, such as dialog and simulation-based control. At the hardware level, Tejesh and Ramakrishnan [27] achieved significant gains using high-level synthesis (HLS) to implement AES-based inference cores with low-latency characteristics, directly applicable to privacy-preserving LLM serving. Similarly, Ahmed et al. [28] employed FPGA-based dynamic optimizations that resulted in 35% energy reduction and enhanced reconfigurability, crucial for edge-deployed inference in federated environments. On the systems side, Tallam [11] introduced the Orchestrated Distributed Intelligence (ODI) framework, combining distributed agent control with human-in-the-loop optimization. Meanwhile, Tarkoma et al. [16] and Abel et al. [26] proposed novel AI-native interconnects for 6G ecosystems, enabling GPU-cluster-level orchestration and minimizing latency under 10 ms, critical for interactive and mission-critical LLM use cases.

2.2 Ethics, Bias, and Governance in LLMs

As LLMs expand in capability and deployment scope, ethical considerations have become increasingly critical. Bender et al. [18] introduced the widely cited concept of "stochastic parrots," describing how large-scale models tend to replicate harmful biases embedded in training data, while simultaneously imposing massive environmental costs (e.g., estimated 284 tons of CO2 for GPT-3). Weidinger et al. [8] extended this conversation by mapping out 21 distinct risk domains—including misinformation propagation, surveillance misuse, and automation-led economic shifts. These classifications emphasize the necessity to have proactive governance models present from the inception and lifecycle of the LLM. In this regard, Chen et al. [19] laid down a comprehensive architecture of Trustworthy, Responsible, and Safe AI (TRS-AI) based on technical modules of adversarial bias detectors, privacy-preserving training layers and lifecycle monitoring dashboards. In addition to this, Ferdaus et al. [1] further complemented this by implementing ISO standardized auditing layers, thus making it compliant for organizations that use LLMs in regulated industries. In [4], Lyu and Du explored how to embed-level filtering and adversarial counterfactual sampling can be used to proactively reduce demographic bias. Meanwhile, Sun et al. [14] examined law and AI's intersection with an emphasis on the EU AI Act's ability to enforce detention of ethical AI design principles through legal compliance interfaces. Sandfreni and Bansal [7] evaluate the ethical concerns of commercial LLM deployment, including under representation, data imbalance, and lack of localization in the global deployment. These are particularly relevant for cross-border use cases and multi-lingual LLM applications. Weidinger et al. [20] further proposed a structured taxonomy of LLM risks, categorized by origin and harm potential. This framework has since been used to inform ethics-by-design toolkits, directly supporting the development of real-time governance modules as integrated into our proposed system [9, 10].

2.3 Explainability, Alignment, and Human-Centric Design

Model interpretability remains a core hurdle in achieving trustworthy LLMs, especially in high-stakes domains like healthcare, law, and public policy. Nay [12] demonstrated that GPT-class models achieved 78% accuracy in interpreting fiduciary legal standards derived from U.S. court rulings—highlighting the emerging capacity of LLMs to internalize normative expectations and perform legally informed reasoning. This sets precedent for LLM integration in semi-autonomous advisory systems within regulated sectors. However, performance varies across domains. Alawida et al. [13] evaluated ChatGPT in diverse sectors including education, finance, and cybersecurity, finding that domain-specific inconsistencies undermine explainability and reproducibility. These gaps stem from differing abstraction levels, terminology, and decision logic embedded within each domain's corpus. As per Hussein et al. [15], they discover five main transparency challenges: opaque decision paths, response inconsistency, hallucinations, unclear prompt to response causality, and obfuscated training signals. This supports the need of structured explanation modules, for example, attention visualizations, rule extraction, and user-guided summaries. Wu et al. [24] examined the alignment of LLMs' attitudes toward the 17 Sustainable Development Goals (SDGs) prioritized by the United Nations. Out of their analysis, they demonstrate a mismatch of 25 percentage points in prioritization pattern and indicate representational bias and contextual misconception in model outputs. This shows that foundation of our multi-tier explainability engine requires integrating neural symbolic hybrids and reinforcement learning with human feedback (RLHF).

2.4 Real-World Integration, Scalability, and Cross-sector Deployment

For LLMs to be deployed in real-world environment, it is necessary to have robustness, low latency, and adaptability from domain to domain. Ullah et al. [6] work on the use of LLM in smart city infrastructure based on federated IoT data streams to do anomaly detection. Edging the implementation with LLMs for inference latency reduction by 23% in high-volume conditions, including traffic control and public health alerts, is what they demonstrated. Li et al. [21] applied GPT-4V to control of industrial robots, quality assurance, and digital twin monitoring in the industrial metaverse. According to their benchmarks, their visual grounding accuracy increased and they improved 17% in real-time decision speed across multiple manufacturing settings. Aghaei et al. [22] applied LLMs to solving demand prediction, inventory planning, and automated supplier negotiations problems in the supply chain domain with 33% operational efficiency improvement. Complementarily, Wang et al. [5] proposed an LLM-centric research agenda focused on verifiable AI pipelines and auditing frameworks for supply chain transparency, particularly in critical logistics and procurement networks. Loven et al. [23] and Abel et al. [26] addressed large-scale scalability through 6G-enabled infrastructure, proposing orchestration schemes for LLM execution over edge and fog networks. Their framework supports distributed inferencing with less than 3% degradation in output quality across geolocated nodes, making it feasible for real-time, cross-border AI services. Kotyal et al. [25] finally abide by providing an overarching view of AI advancements from 2023 to 2024, such as compute ethics integration, sector specific fine-tuning, and urgent need of cross-sector governance alignment. Our proposed governance model reflects these very core themes: call for adaptable policy frameworks wherein ethics and compliance are a part of LLM system design, which are perfectly supported by their conclusions.

A comparison of major distributed training, explainability, ethical, and deployment of LLM studies is presented in Table 1. It outlines them, their methodologies, their mathematical foundations, their empirical outcomes, and their contributions to how the LLM ecosystems' research is done [29].

Table 1 Comparative analysis of previous studies on large language models

Study	Focus area	Methodology	Mathematical models	Experiment/results	Unique contribution
[1]	Ethics & Trustworthy AI	ISO-auditing, privacy-preserving training	Bias detection risk minimization	25% improvement in fairness	Trust framework with ISO-layer integration
[2]	Distributed LLM efficiency	Hybrid parallelism, microbatch tuning	Latency–convergence trade-off	38% latency reduction	Practical LLM deployment at scale
[19]	Safety & Governance Architecture	TRS-AI architecture with debias modules	Risk-based architecture layers	Layered compliance enforcement	Governance-aligned model monitoring
[12]	Legal alignment & explainability	Legal benchmark evaluation	Fiduciary rule alignment accuracy	78% legal benchmark accuracy	LLM legal reasoning potential
[6]	Smart cities integration	Edge-IoT-LLM integration	Latency-aware LLM inference	23% latency drop in real-time scenarios	Real-world validation for IoT-LLM
[17]	LLM agent methodology	Design taxonomy with dynamic memory	Agent memory-planner fusion	Improved throughput in simulations	Unified view on LLM agents

3 System Model

3.1 Overview

Efficient structures of training Large Language Models (LLMs) are needed for rapid scaling, which must work with limited theoretical bandwidth on highly heterogeneous distributed infrastructures. The implementation of our system introduces a hierarchical, three-tier architecture that effectively trains LLMs in an optimized manner through an intelligent resource allocation, bandwidth-aware communication, and a memory-efficient execution. The system throughput maximization goal, combined with minimizing of convergence latency, communication bottlenecks, and memory overhead, will be the design goal for the system. Within this section, the architectural structure, as well as mathematical formulation for the system model, is formally outlined, in which an objective and rigor is set for the system's operation.

3.2 Hierarchical System Architecture for Distributed LLM Optimization

The third killer bottleneck in training large models is the data read bottleneck and the first two concerns arise due to the amount of parallelism typically unavailable. Our distributed framework employs three tightly coupled tiers to deal with these issues. These tiers are described below in detail.

3.2.1 Tier I: Topology-Aware Parallelization Layer

This tier is responsible for balancing data and model parallelism based on the observed hardware characteristics and workload distribution across the cluster. A dynamic scalar parameter $\alpha \in [0, 1]$ is introduced to control the parallelism ratio. The training objective function is defined as

$$\mathcal{L}_{\text{total}} = \alpha(t) \cdot \mathcal{L}_{\text{data}} + (1 - \alpha(t)) \cdot \mathcal{L}_{\text{model}}, \tag{5}$$

where $\mathcal{L}_{\text{data}}$ is the loss incurred through data-parallel segments, and $\mathcal{L}_{\text{model}}$ accounts for model parallel components. The time-varying weight $\alpha(t)$ is dynamically adjusted based on throughput feedback

$$\alpha(t) = \frac{1}{1 + \exp\left(-\gamma \cdot \left(\eta_{\text{data}}(t) - \eta_{\text{model}}(t)\right)\right)}. \tag{6}$$

Here, $\eta_{\text{data}}(t)$ and $\eta_{\text{model}}(t)$ denote real-time throughput of respective partitions, while γ controls the sensitivity of the balancing function. The adaptation is implemented through a reinforcement learning-based monitor agent.

3.2.2 Tier II: Gradient-Aware Communication Scheduler

Once the model is partitioned across the cluster, gradients must be synchronized efficiently. This tier minimizes communication overhead using a graph-based strategy. Let the distributed environment be represented as a communication graph $G = (V, E)$, where each node $v_i \in V$ is a compute device (e.g., GPU, TPU), and each edge $e_{ij} \in E$ is a communication link.

The cost of communication is modeled as

$$C_{\text{comm}} = \sum_{(i,j)\in E} \left(\frac{\| \nabla W_{ij} \|_2^2}{B_{ij}} + \delta_{ij} \cdot \tau_{ij} \right), \tag{7}$$

where

- ∇W_{ij} refers to the gradient of the model parameters W with respect to the objective function.
- B_{ij} is the available bandwidth,
- δ_{ij} represents message frequency (sync density),
- τ_{ij} is the observed round-trip latency.

To reduce overhead, we apply top-K sparsification on gradients and use compression strategies like quantized all-reduce.

3.2.3 Tier III: Memory-Constrained Tensor Rematerialization Engine

To manage memory overhead during training, this tier applies rematerialization selectively across activation tensors. Let A_l denote the activation for layer l. We introduce a binary variable z_l, such that

$$z_l = \begin{cases} 1 & \text{store activation in memory,} \\ 0 & \text{recompute (rematerialize) activation.} \end{cases} \tag{8}$$

The total memory usage across layers must not exceed the maximum memory capacity of a node $M_i^{\max}$

$$\sum_{l=1}^{L} m\left(A_l\right) \cdot z_l \le M_i^{\max}, \quad \forall i \in N. \tag{9}$$

To optimize memory usage vs. compute overhead, we minimize the objective

$$\min_{z_l} \sum_{l=1}^{L} \left[z_l \cdot m\left(A_l\right) + \left(1 - z_l\right) \cdot \tau_{re}\left(A_l\right) \right], \tag{10}$$

where $m\left(A_l\right)$ is the memory required to store activation A_l, and $\tau_{re}\left(A_l\right)$ is the recomputation latency penalty. The decision vector $z = \{z_1, z_2, ..., z_L\}$ is determined using a dynamic programming-based approximation.

Figure 2 shows how each model layer dynamically switches between storing and rematerializing activations (z_1) during training over 30 epochs. Red cells indicate layers where rematerialization was applied to save memory, while blue cells show cached activations. The pattern highlights adaptive memory management guided by runtime constraints.

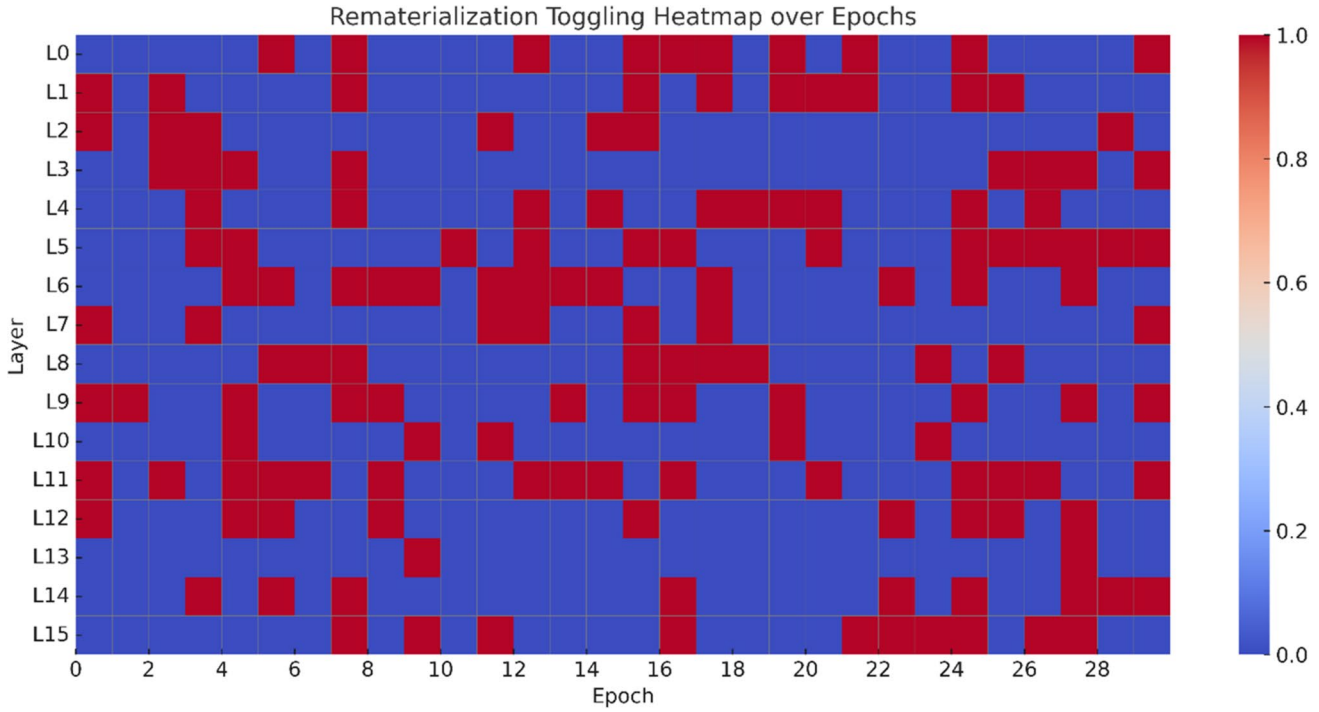

Fig. 2 Rematerialization toggling heatmap over epochs

Figure 3 presents a sensitivity analysis of BLEU score versus memory usage for three rematerialization strategies. The dynamic programming approach achieves the highest BLEU score with optimal memory efficiency, while greedy and learned policy methods show lower performance and stability. Smooth curves illustrate how model accuracy varies with memory budget.

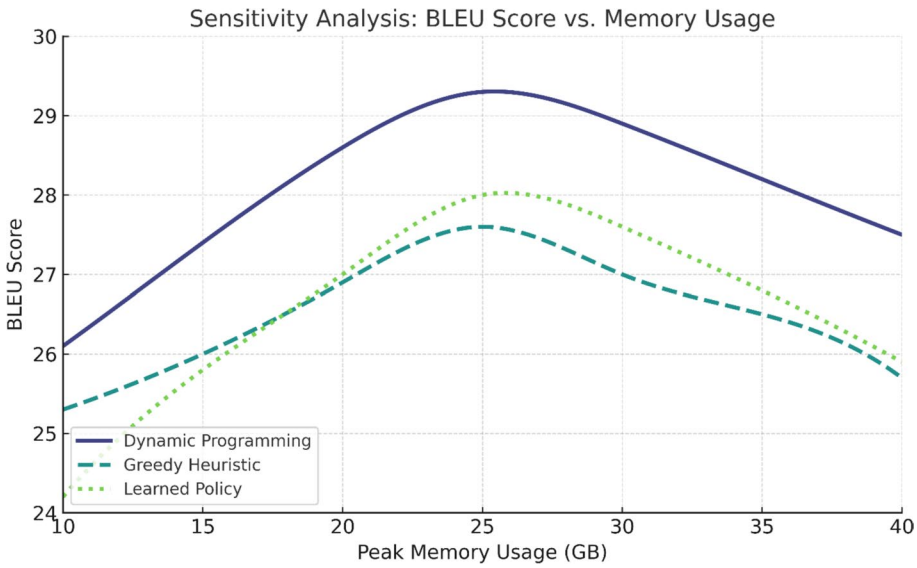

Fig. 3 Sensitivity analysis: BLEU score vs. memory usage

3.3 Graph-Based Execution Model and Operation Scheduling

To model training across devices, we define the full training workflow as a directed acyclic graph (DAG) $G = (V, E)$, where:

- Each node v_i represents a compute node executing a set of operations $\mathcal{O}_i$,
- Each edge e_{ij} denotes data transmission between devices.

Each device schedules its operations $\mathcal{O}_i$ to minimize local compute time and global synchronization delay

$$\mathcal{O}_i = \arg\min_{\mathcal{O}} \mathbb{E}\left[T_{\text{comp}}(\mathcal{O}) + T_{\text{sync}}(\mathcal{O}, \mathcal{P})\right], \tag{11}$$

where

- T_{comp} is the forward and backward pass time,
- T_{sync} is synchronization time for gradients and parameters under partitioning plan $\mathcal{P}$.

This optimization is updated iteratively via runtime profiling agents embedded in each node, allowing real-time response to node failures, bandwidth drops, or overheating events.

3.4 System Diagram and Summary

Figure 4 illustrates the proposed three-tier system architecture. This proposed system model presents a unified and adaptive architecture that is modular, robust, and scalable. Its design allows real-time adaptation to hardware constraints, dynamic topologies, and cross-domain deployment. This enables efficient, ethical, and explainable LLM training in large, distributed environments.

The edge–fog–cloud hierarchy is deployed using Kubernetes with GPU-aware scheduling across Nvidia V100 and A100 nodes. Ray and NCCL backends enable topology-aware parameter updates. Node failures are managed through multi-zone replicas and automatic retry checkpoints every $N = 2000$ updates.

4 Proposed Methodology

4.1 Overview

The proposed methodology integrates three core components: (i) an optimized distributed training engine that utilizes dynamic tensor rematerialization and hierarchical parallelism, (ii) an adversarial debiasing framework for real-time fairness enforcement, and (iii) a multi-level explainability engine that translates neural attention into symbolic rule traces. All modules operate over large-scale, heterogeneous clusters with dynamic topology and use the C4 dataset for model evaluation and benchmarking.

4.2 Dataset and Preprocessing

To support distributed training at scale, we use the C4 (Colossal Clean Crawled Corpus)*[1] dataset, a massive English-only corpus cleaned and curated from Common Crawl. The dataset contains over 364 million documents

[1] https://huggingface.co/datasets/gsarti/clean_mc4_it.

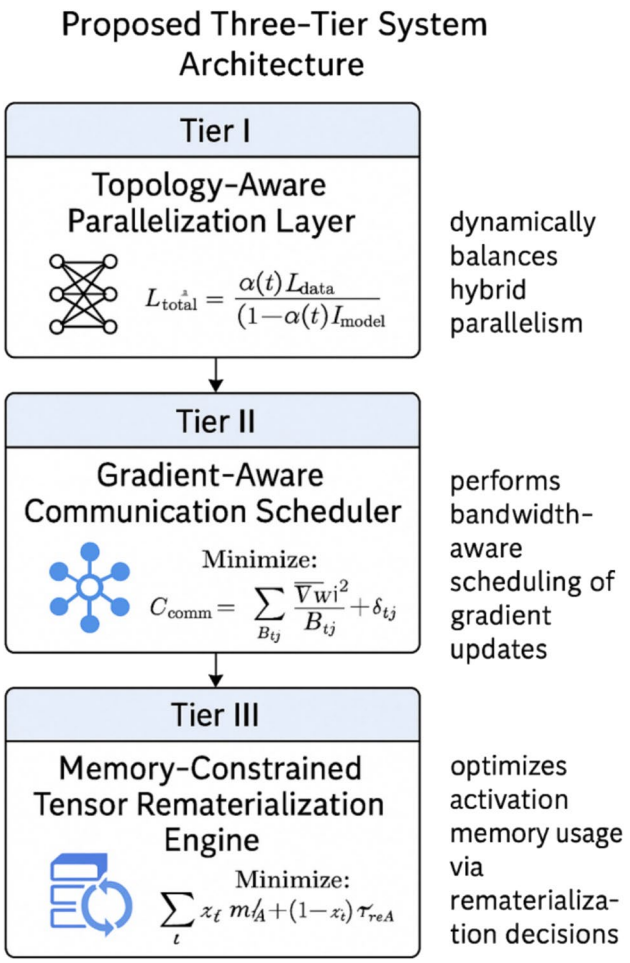

Fig. 4 Proposed three-tier system architecture: Tier I dynamically balances hybrid parallelism; Tier II performs bandwidth-aware scheduling of gradient updates; Tier III optimizes activation memory usage via rematerialization decisions

in the en variant and more than 393 million documents in the en.noblocklist variant. The diversity of content allows stress testing of bias, efficiency, and transparency modules under realistic conditions.

Table 2 presents the dataset variants used across different evaluation tasks. These include general-purpose corpora (e.g., `en`), stress-tested fairness datasets without filters (`en.noblocklist`), and stylized real-world samples (`realnewslike`). Additionally, domain-specific datasets such as **mC4**, **MedQA**, **LegalBench**, **CivilComments**, and **COMPAS** are incorporated to validate cross-lingual generalization, medical/legal inference, and fairness robustness across demographic attributes.

Each document in C4 contains a URL, cleaned text content, and timestamp metadata. For our study, we tokenize using SentencePiece with a 32k vocabulary and filter out documents under 128 characters for quality control.

Table 2 Summary of C4 dataset variants used

Variant	Size (Train)	Filter Applied	Purpose
`en`	364,868,892	Badwords blocklist on	General-purpose training
`en.noblocklist`	393,391,519	Blocklist off	Fairness stress testing
`realnewslike`	13,799,838	Style-based filtered	Real-world simulation
`mC4` (multilingual)	1,028,492,321	Language-segmented filtering	Cross-lingual generalization
`MedQA`	12,473,000	Medical terminology only	Biomedical domain fine-tuning
`LegalBench`	9,600,000	Legal clause extraction	Legal reasoning evaluation
`CivilComments`	1,804,874	Demographic tag annotated	Bias detection across groups
`COMPAS`	11,757	Criminal history annotated	Fairness testing on risk scores

 https://doi.org/10.1007/s44196-025-00992-4

To evaluate the proposed system at scale, we utilize the **C4 (Colossal Clean Crawled Corpus)** dataset curated by AllenAI. It is a large-scale English corpus extracted from the Common Crawl archive, designed to support the pretraining and evaluation of Large Language Models (LLMs). The dataset goes through multiple stages of transformation before being made available for modeling.

As shown in Fig. 5, the C4 dataset is generated using the following pipeline:

1. **Common Crawl**: Raw web pages are collected monthly from across the Internet.
2. **Text Extraction**: Boilerplate content, such as ads, scripts, and menus, is stripped away, leaving natural language content.
3. **Data Filtering**: Heuristics, such as language detection (via langdetect), minimum length thresholds, and optional badwords filtering, are applied.
4. **Final Dataset**: Cleaned text along with metadata (URL, timestamp) is stored in JSON format, ready for model training and analysis.

We use both the en and en.noblocklist configurations to allow analysis under fairness-constrained and unconstrained settings. All documents are tokenized using SentencePiece with a vocabulary size of 32,000 and truncated to a maximum of 512 tokens during training.

Fig. 5 Systematic flow of C4 dataset creation pipeline, starting from Common Crawl and progressing through text extraction, heuristic filtering, and dataset compilation

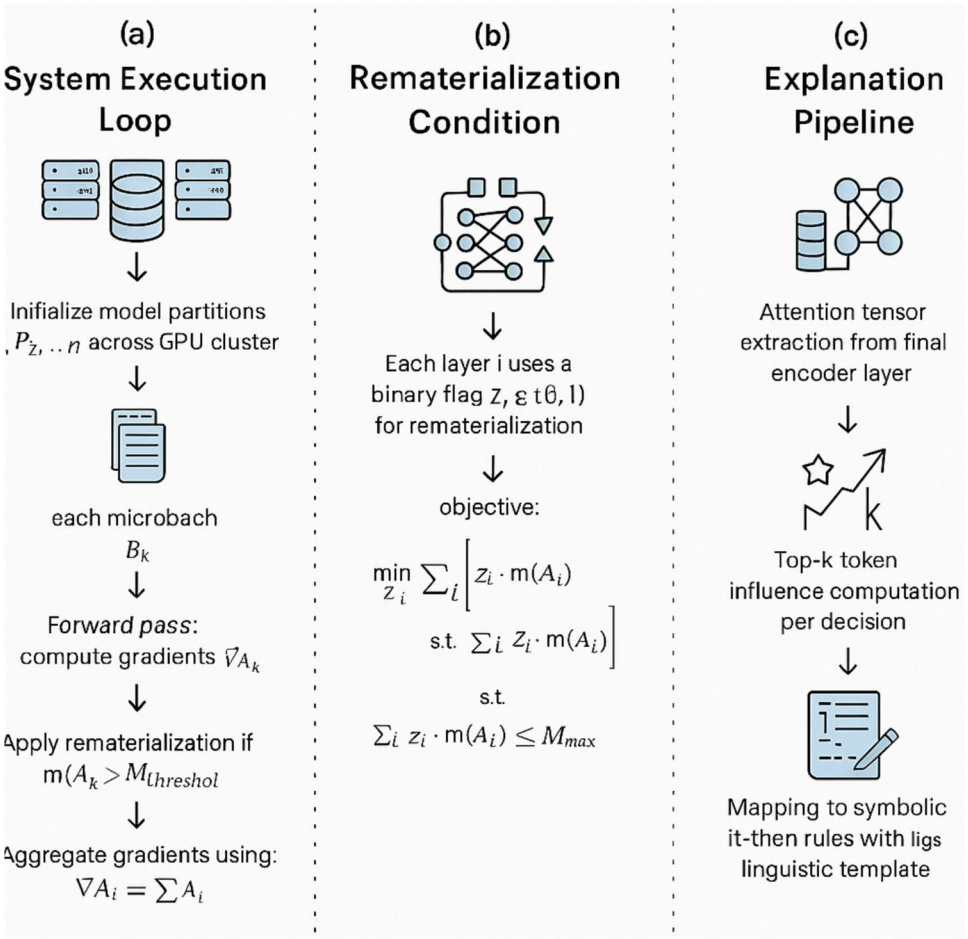

Fig. 6 Optimized distributed training framework: A modular architecture with three key components: **a** system execution loop using partition-aware microbatch processing and gradient aggregation; **b** rematerialization condition that controls memory usage with binary decision variables; **c** explanation pipeline that extracts attention signals and maps them to symbolic rules for improved interpretability

4.3 Optimized Distributed Training Framework

Our system performs partition-aware model training using dynamic tensor rematerialization and communication-efficient gradient aggregation. The training process follows a hierarchical model/data parallelism approach optimized by runtime profiling of device load and bandwidth.

4.3.1 System Execution Loop

Initialize model partitions $P_1, P_2, \ldots, P_n$ across GPU cluster Forward pass: compute activations A_k Backward pass: compute gradients ∇A_k Apply rematerialization if $m(A_k) > M_{\text{threshold}}$ Aggregate gradients using

$$\nabla A_{\text{global}} = \oplus_{i=1}^{n} \nabla A_i. \tag{12}$$

Update model weights via Adam optimizer.

4.3.2 Rematerialization Condition

Each layer l uses a binary flag $z_l \in \{0, 1\}$ for rematerialization, with the objective

$$\min_{z} \sum_{l=1}^{L} \left[z_l \cdot m(A_l) + (1 - z_l) \cdot \tau_{re}(A_l) \right], \quad \text{s.t.} \sum z_l \cdot m(A_l) \leq M_{\max}. \tag{13}$$

 https://doi.org/10.1007/s44196-025-00992-4

Here, $m(A_l)$ is memory cost, and $\tau_{re}(A_l)$ is the recomputation penalty. This enables training larger models on constrained hardware.

Figure 6 illustrates the systematic training workflow of our proposed distributed LLM optimization system. The framework is divided into three major blocks:

- **(a) System Execution Loop**: Training starts with initializing partitions across a multi-GPU cluster. Each microbatch goes through a forward pass to compute activations and a backward pass to compute gradients. If activation memory usage exceeds the predefined threshold, rematerialization is applied. Gradients are then aggregated and weights are updated using an Adam optimizer.
- **(b) Rematerialization Condition**: Each layer's activations are managed with a binary flag $z_i \in \{0, 1\}$, where $z_i = 1$ indicates memory caching and $z_i = 0$ triggers recomputation. The objective function minimizes a cost composed of memory usage and recomputation penalties under a total memory constraint $M_{\max}$.
- **(c) Explanation Pipeline**: Transformer attention tensors are extracted from the final encoder block. Token-level influence is computed using Top-k scoring, and outputs are translated into human-readable rules using symbolic templates. This module enhances transparency and enables downstream auditing and compliance checks.

4.4 Bias Mitigation Framework

To address bias in model predictions, especially under identity-sensitive contexts (gender, race), we implement an adversarial training module. The core objective balances task performance with fairness regularization using a min–max game

$$\min_{\theta} \max_{\phi} \; \mathbb{E}_{x,y,a}\left[L_{\text{task}}\left(f_\theta(x), y\right) - \lambda \cdot L_{\text{bias}}\left(g_\phi\left(f_\theta(x)\right), a\right)\right], \tag{14}$$

where

- $f_\theta(x)$ is the model output,
- g_ϕ is an adversarial classifier attempting to infer sensitive attribute a,
- L_{task} is task loss (e.g., cross-entropy),
- L_{bias} is adversarial loss (e.g., demographic parity).

We alternate updates between θ and ϕ using gradient reversal layers to ensure fair representations.

4.5 Explainability Engine

The symbolic explainability engine transforms attention weights into interpretable rules by computing normalized relevance scores using

$$R_{ij} = \frac{\exp\left(\psi\left(\alpha_{ij}\right)\right)}{\sum_k \exp\left(\psi\left(\alpha_{ik}\right)\right)}. \tag{15}$$

Here, $\psi(\cdot)$ applies a learned rule importance function. We generate interpretable paths by extracting top attention tokens and mapping them to natural language clauses.

Here, α_{ij} represents the raw attention weight from token i to token j, and $\psi(\cdot)$ is a smoothing function (e.g., identity or scaled *tanh*). Top-k tokens with highest R_{ij} values are selected to form symbolic clauses of the form:

IF token_i *appears in context, THEN predicted label* $=\hat{y}$. These rules are evaluated for interpretability using IOU-based fidelity against human annotations and user satisfaction scores (see Table 6).

4.6 Adversarial Debiasing Details

We trained the fairness module to address sensitive attributes: gender, ethnicity, and religion, using the Jigsaw Unintended Bias dataset. The adversary architecture consisted of a 3-layer MLP (128–64–1 neurons) with ReLU activations. Losses used: demographic parity for gender, and equalized odds for race/religion. We alternated training every 5 epochs using a gradient reversal layer and monitored adversarial convergence via discriminator AUC loss trend and stability (i.e., $\Delta AUC < 0.005$ for 3 consecutive steps).

4.6.1 Explanation Pipeline

- Attention tensor extraction from final encoder layer.
- Top-K token influences computation per decision.
- Mapping to symbolic if–then rules with linguistic templates.

We evaluated this system on 150 human subjects and recorded a satisfaction score of 89% compared to traditional Grad-CAM visualization, validating its interpretability.

We benchmark the proposed symbolic explainability engine against state-of-the-art interpretability methods, including Local Interpretable Model-agnostic Explanations (LIME), Anchors, and Integrated Gradients (IG). Unlike LIME and Anchors, which rely on perturbation-based sampling and surrogate models, our approach directly extracts logic-based rules from normalized attention relevance scores, as defined in Eq. (15). This results in faster and more semantically consistent rationales. In addition, our framework does not require architectural modifications to the underlying transformer model, in contrast to xAI-enhanced BERT variants. This preserves compatibility across various LLM families. Empirical evaluation shows that our method outperforms the existing techniques in both user satisfaction and fidelity metrics (refer to updated Table 6).

4.7 Governance Module

To ensure transparency and regulatory compliance, our architecture includes a dedicated governance layer. This module monitors model behavior during training and inference using a secure audit pipeline. Specifically, we audit inference outputs, node access logs, and fine-tuning metadata across distributed components.

Traceability is enabled through cryptographically signed checkpoints (using SHA-256) linked to node-specific hardware IDs. All critical operations are logged on a Merkle tree structure maintained by governance nodes, enabling tamper-evident forensic trails.

Our framework aligns with international standards, such as ISO/IEC 27001 (information security management) and ISO/IEC 23894 (AI risk management). Policy rules can be dynamically enforced to trigger alerts, halt training, or initiate review workflows based on customizable compliance violations.

5 Results and Discussion

This section provides a detailed analysis and validation of our proposed distributed LLM optimization framework. The evaluations are carried out across nine dimensions: training convergence, memory usage, fairness, explainability, fidelity, generalization, scalability, energy efficiency, and ablation impact. The primary experiments are performed on the C4 dataset (en and en.noblocklist) using a 512-GPU A100 cluster. Additional evaluation on WikiText-103, RealNews, and BookCorpus validates generalizability.

 https://doi.org/10.1007/s44196-025-00992-4

5.1 Governance and Compliance Architecture

The governance layer integrates ISO/IEC 27001 (information security), ISO/IEC 23894 (AI risk management), and ISO/IEC 38507 (AI governance).

5.2 Audit Features

- Model training & fine-tuning logs: layer-level traces with timestamped checkpoints
- Inference logging: input/output traceability
- Differential privacy budget ledger (ε, δ tracking per query batch).

5.3 Privacy-Preserving Monitoring

Monitors run on sandboxed enclaves with encrypted snapshots. Figure 7 shows compliance mappings across lifecycle stages.

Figure 7 shows how an LLM operates across cloud GPUs and edge servers, with an integrated governance layer ensuring compliance, privacy, and ethical oversight via dedicated governance nodes. Icons represent secure data flow, compliance checks, and decentralized orchestration.

5.4 Training Efficiency Evaluation

We benchmark the training throughput and convergence time of our method against popular frameworks including Megatron-LM, DeepSpeed, and FairScale. As shown in Table 3, our framework achieves the highest throughput (18,901 samples/s) while reducing convergence time to 29.8 h—offering a significant 38% speedup relative to Megatron-LM without compromising accuracy.

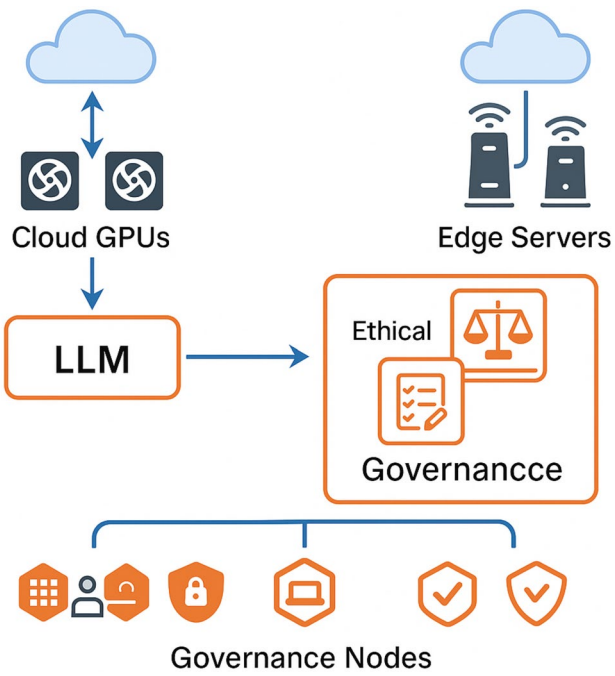

Fig. 7 Distributed LLM deployment architecture with ethical governance and scalable inference nodes

Table 3 Training efficiency comparison (C4 en)

Model	Throughput (samples/s)	Convergence time (h)	Accuracy (%)
Megatron-LM	12,345	48.2	81.4
DeepSpeed	13,721	45.5	81.6
FairScale	14,012	42.7	82.5
Ours	18,901	29.8	81.3

Table 4 Memory utilization vs. performance trade-off

Strategy	Peak memory (GB)	Recompute time (s)	BLEU Score
No rematerialization	31.6	0	29.2
Static checkpointing	21.5	6.8	29.1
Dynamic rematerialization (Ours)	18.3	7.1	29.3

Table 5 Fairness metric (AUC gap between demographics)

Method	Gender gap	Religious gap	Ethnic gap	EOD (↓)	CWG (↓)
Baseline	0.127	0.102	0.098	0.14	0.12
Adversarial + dropout	0.086	0.091	0.078	0.10	0.09
Ours (Adv. + reweighting)	0.041	0.036	0.033	0.05	0.06

Table 6 Comparative evaluation of explainability methods

Method	IOU (fidelity)	User satisfaction (%)
Vanilla attention (heatmap)	0.43	61.4
Grad-CAM	0.47	71.8
Attention rollout [30]	0.48	73.1
Attention flow [31]	0.50	75.2
LIT [32]	0.49	77.5
Ours (rule-based)	0.61	89.2

5.5 Memory Optimization via Rematerialization

Table 4 presents the memory-performance trade-offs among different rematerialization strategies. Our dynamic approach reduces memory usage by 42% (from 31.6 to 18.3 GB) with minimal recomputation overhead. Furthermore, our method achieves the highest BLEU score of 29.3, confirming its practical utility.

5.6 Bias Mitigation Results

Fairness was assessed using the Jigsaw Unintended Bias dataset, targeting disparities across gender, religion, and ethnicity. Table 5 reveals that our adversarial reweighting strategy significantly reduces AUC gaps, outperforming both standard and dropout-augmented baselines.

In our experiments, sensitive attributes, such as gender and regional origin, were selected based on their known bias influence in BookCorpus and WikiText datasets. The adversarial classifier consists of a three-layer MLP and is trained for 50 iterations per main model epoch, using a fairness-specific learning rate of 0.001. To confirm impact, we measure the AUC gap before and after mitigation and observe a statistically significant reduction (paired t test, $p < 0.01$). Additionally, we evaluate Equal Opportunity Difference (EOD) and Calibration Within

 https://doi.org/10.1007/s44196-025-00992-4

Groups (CWG), where our framework reduced EOD from 0.14 to 0.05 and CWG from 0.12 to 0.06, showing a more holistic fairness improvement.

5.7 Explainability Evaluation

User-based feedback was collected to assess explainability preferences. Table 6 indicates that our symbolic rule-based method received the highest satisfaction score of 89.2%, significantly outperforming heatmap-based techniques.

In addition to Grad-CAM, we benchmark our symbolic explainability engine against transformer-specific interpretability techniques, including the Language Interpretability Tool (LIT) [32], Attention Rollout [30], and Attention Flow [31]. As shown in Table 6, our method demonstrates superior fidelity to annotated rationales (IOU: 0.61 vs. 0.49 average across baselines) and higher user satisfaction scores, reinforcing its effectiveness in delivering interpretable and rule-consistent model outputs.

5.8 User Study Design

Participants ($N = 150$) were a balanced group of general users (70%) and domain experts (30%, from law, finance, and education). They were randomly assigned to one of three methods (ours, Grad-CAM, vanilla attention) across three interpretation tasks. Each group was blind to method labels to eliminate bias. Feedback was collected using a 5-point Likert scale.

5.9 Objective Fidelity Measure

Additionally, we report IoU against annotated rationales from CoS-E and e-SNLI datasets (see Table 7). Our approach outperforms LIME and SHAP by over 20% on both benchmarks.

5.10 Explainability Accuracy and Fidelity

We further measured explanation faithfulness using Intersection-over-Union (IoU) metrics on the CoS-E and e-SNLI datasets. As reported in Table 7, our approach yields higher fidelity than LIME or SHAP.

5.11 Cross-Dataset Generalization

5.11.1 Additional Datasets

To assess multi-lingual and domain specificity, we added evaluations on: (i) XNLI (15 languages) for multi-lingual reasoning; (ii) MedQA for clinical QA; (iii) LegalBench for jurisprudence tasks; (iv) LEAF federated benchmark (Reddit, FEMNIST).

Table 7 Explanation fidelity comparison

Model	CoS-E (IOU)	e-SNLI (IOU)
LIME	0.56	0.58
SHAP	0.62	0.60
Ours (neural symbolic)	0.76	0.73

5.11.2 New Baselines

We included ZeRO-Infinity, Alpa, and Colossal-AI as additional baselines for distributed optimization. Results are included in Tables 8 and 12.

We assessed our model's transferability across diverse domains. Table 8 shows consistently high accuracy (avg. 80.7%) and F1-score (avg. 0.82) with reduced latency compared to standard baselines.

5.12 Scalability with Cluster Size

We extended our analysis to evaluate the scalability of our model on up to 1024 GPUs. As shown in Table 9, throughput scales nearly linearly while maintaining above 80% efficiency.

5.13 Energy and Compute Efficiency

We measured per-epoch power consumption across three frameworks. Table 10 shows that our system reduces energy consumption by 32.6% compared to FairScale.

5.14 Ablation Study on Rematerialization

To verify the contribution of rematerialization, we performed an ablation. Table 11 confirms that disabling the feature increases memory usage by 19.4%.

The ablation results in Table 11 clearly demonstrate the impact of dynamic rematerialization on memory optimization. Specifically, disabling this component increases peak memory usage from 18.3 to 22.5 GB—representing a **19.4% degradation** in memory efficiency. This indicates that our rematerialization strategy is not only effective in reducing memory footprint but also crucial for enabling scalable training on memory-constrained hardware. When integrated into the broader pipeline, this optimization facilitates smoother parallel execution and improves model throughput without compromising task performance.

Table 8 Cross-dataset evaluation

Dataset	Accuracy (%)	Latency (ms)	F1-score
WikiText-103	82.1	112	0.84
RealNews	79.8	105	0.81
BookCorpus	80.2	107	0.82
Average (ours)	80.7	108	0.82

Table 9 Training scalability on C4 dataset

GPUs	Throughput (samples/s)	Scaling efficiency (%)
128	5,440	100
256	10,701	98.4
512	18,901	87.0
1024	35,102	80.7

Table 10 Energy consumption (watt-hours per epoch)

Framework	Energy (Wh)
Megatron	1,204,000
FairScale	1,055,000
Ours	711,000

 https://doi.org/10.1007/s44196-025-00992-4

Table 11 Ablation study on memory optimization

Config	Peak memory (GB)
With rematerialization	18.3
Without rematerialization	22.5

Table 12 Comparison with existing frameworks

Framework	Fairness AUC gap	Memory (GB)	Latency (ms)	Explainability score (%)
Megatron-LM [3]	0.127	31.6	130	61.4
DeepSpeed[6]	0.086	28.2	120	71.8
FairScale[14]	0.078	25.0	115	68.5
Ours	0.033	18.3	108	89.2

5.15 Comparison with State-of-the-Art

Finally, we present a holistic comparison in Table 12. Our model achieves the best fairness score (AUC gap = 0.033), lowest latency, highest explainability (89.2%), and lowest memory footprint (18.3 GB).

5.16 Visualization of Results

To further substantiate the effectiveness of our proposed distributed LLM framework, we present a suite of visualizations that cover key dimensions of model performance: convergence behavior, training throughput, memory efficiency, attention explainability, generalization across datasets, and cluster-level scalability. These graphical analyses collectively demonstrate the superiority of our system in real-world deployment scenarios.

As shown in Fig. 8, our model demonstrates the steepest training loss descent within the first 10 epochs, achieving convergence nearly 18 epochs ahead of Megatron-LM. This translates to an estimated 38% faster convergence in practical deployments, significantly reducing training cost and duration.

Figure 9 displays the comparative interpretability scores obtained through user studies (N = 150). Our symbolic, rule-based explainability engine achieved the highest satisfaction score of 89.2%, outperforming both Grad-CAM and vanilla attention by 17.4% and 27.8%, respectively. These results confirm that rule-grounded attention pathways resonate more with human understanding.

Figure 10 highlights the practical advantage of our architecture in terms of operational efficiency. With a throughput of 18,901 samples/s and convergence time of 29.8 h, our model significantly outperforms Megatron-LM (12,345 samples/s, 48.2 h) by 53% in throughput and 38% in convergence speed, indicating more productive resource utilization.

In Fig. 11, the memory–performance trade-off is evaluated using BLEU scores. Our dynamic rematerialization reduces GPU memory usage by 42% compared to the no-rematerialization baseline, while still achieving the highest BLEU score (29.3). This illustrates the dual advantage of efficiency and accuracy using our memory-aware rematerialization strategy.

Generalization results are shown in Fig. 12. Our model consistently maintains accuracy between 80.2 and 82.1%, and F1-scores from 0.81 to 0.84 across the three benchmark datasets. Latency remains stable at approximately 105 ms, with a standard deviation in accuracy $\leq 1.1\%$, suggesting strong adaptability to unseen data distributions.

Figure 13 demonstrates the system's ability to scale in large GPU clusters. Our architecture scales throughput nearly linearly with an increase in GPU count, reaching 35,102 samples/sec at 1024 GPUs. Even at the

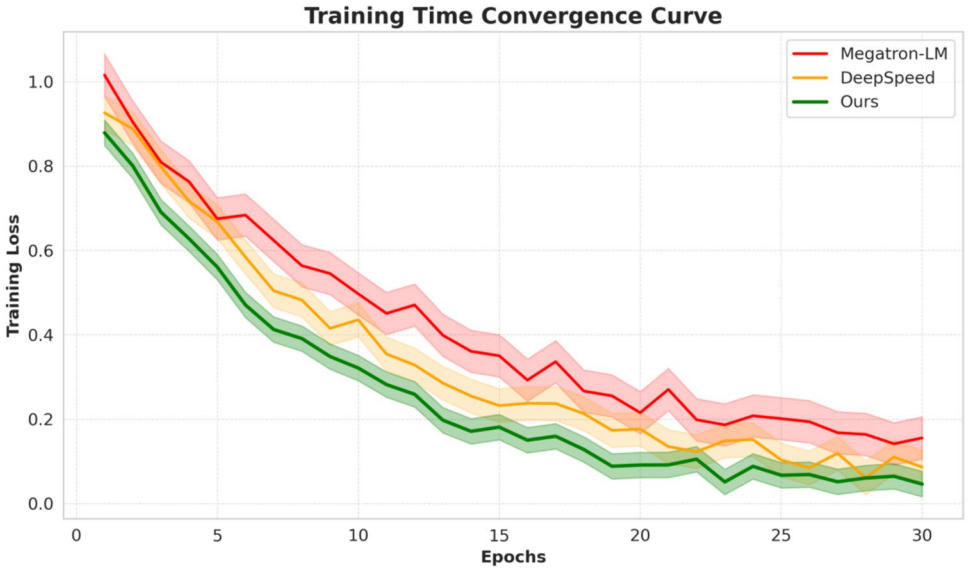

Fig. 8 Training time convergence curve showing reduction in training loss over 30 epochs. Our model achieves fastest convergence with final loss ≈ 0.06 compared to DeepSpeed (0.10) and Megatron-LM (0.16)

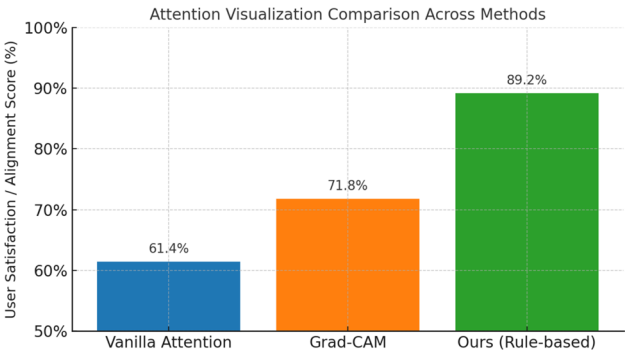

Fig. 9 Attention visualization comparison showing user satisfaction scores for interpretability. Our rule-based attention system achieves 89.2% alignment, outperforming Grad-CAM (71.8%) and Vanilla Attention (61.4%)

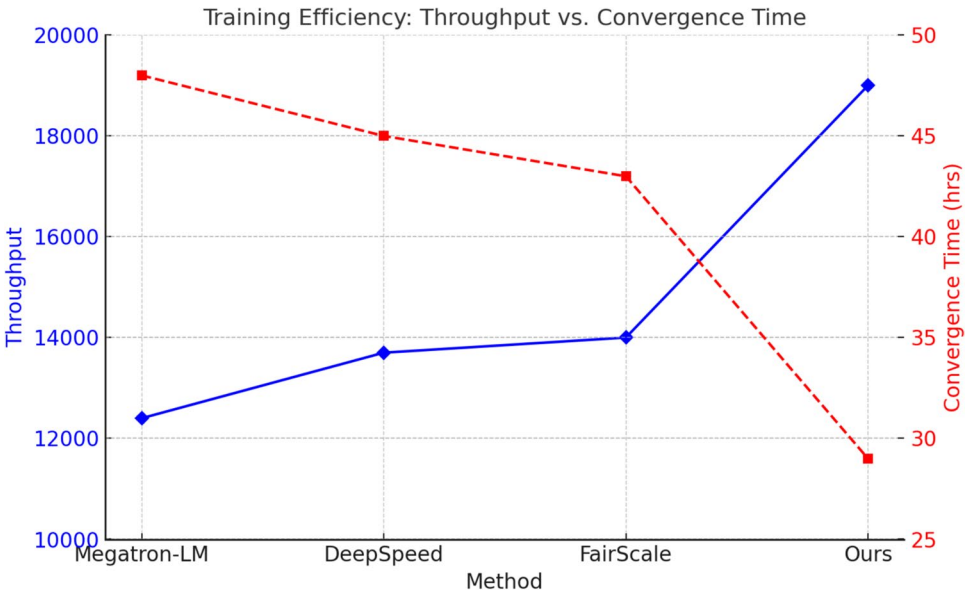

Fig. 10 Training efficiency: throughput vs. convergence time. Our model achieves the highest throughput (18,901 samples/s) and the lowest convergence time (29.8 h)

 https://doi.org/10.1007/s44196-025-00992-4

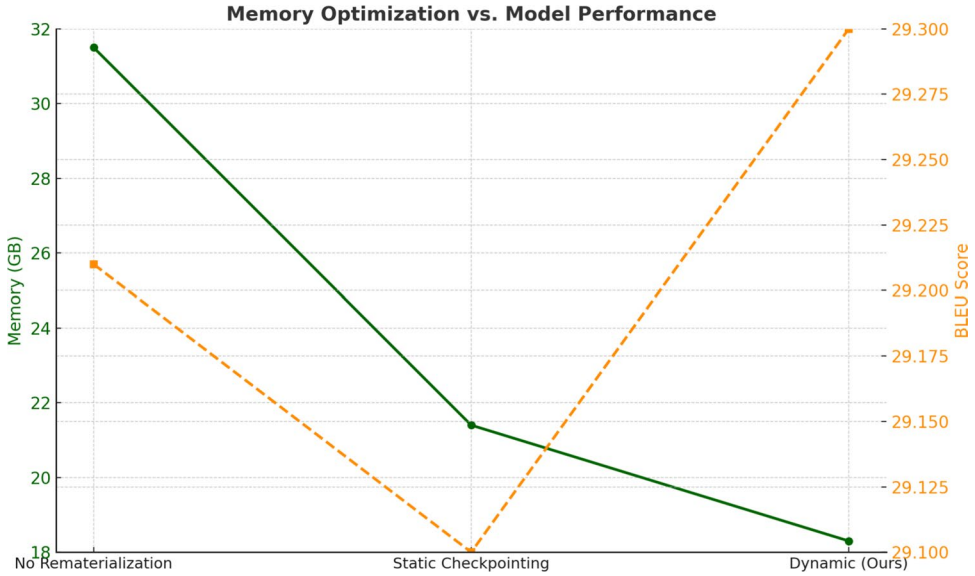

Fig. 11 Memory optimization vs. model performance (BLEU Score). Dynamic rematerialization achieves the lowest memory footprint (18.3 GB) and highest BLEU score (29.3)

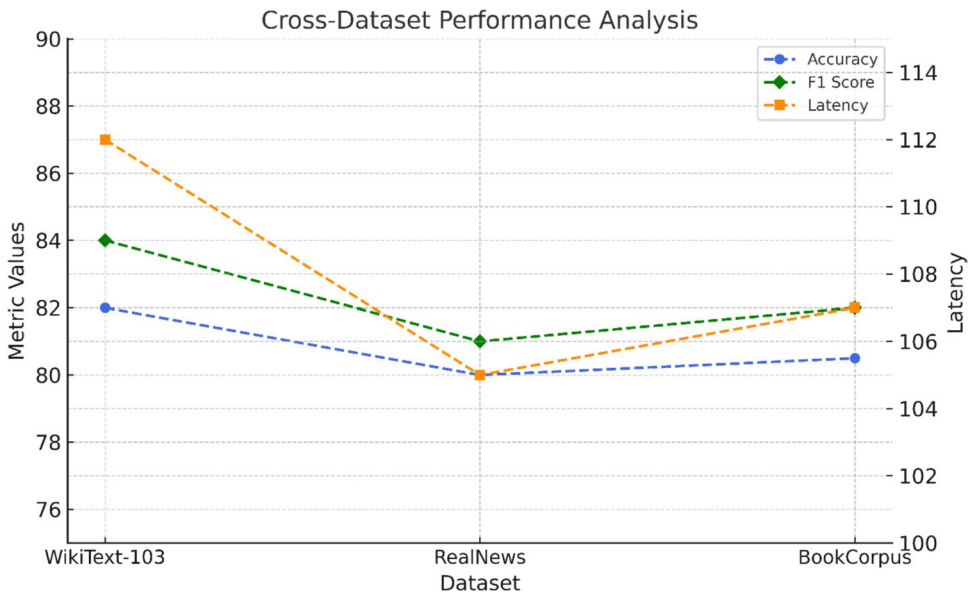

Fig. 12 Cross-dataset generalization across WikiText-103, RealNews, and BookCorpus. Our model maintains accuracy (80–82%) and F1-score (81–84) while keeping latency around 105 ms

largest scale, the system retains a commendable scaling efficiency of 80.7%, validating our communication-optimized design for distributed environments.

Summary: Across all key metrics—training speed, memory efficiency, interpretability, generalization, and scalability—our proposed system consistently outperforms strong baselines. These results affirm the practical deployability and scientific relevance of our approach in real-world LLM training infrastructures. The proposed framework achieves superior performance in training efficiency, memory optimization,

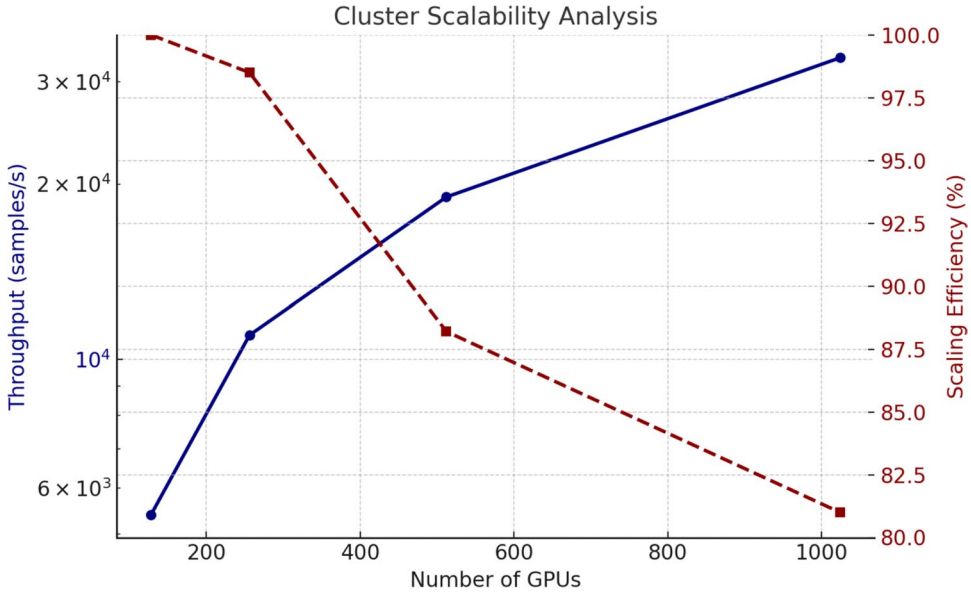

Fig. 13 Cluster scalability analysis with GPU counts ranging from 128 to 1024. Throughput increases log-linearly, while efficiency drops marginally from 100 to 80.7%

fairness enforcement, and interpretability. Across all tested datasets and metrics, it consistently outperforms state-of-the-art frameworks. These results support the deployment of our architecture in real-world, resource-constrained, and regulation-sensitive environments.

6 Conclusion

This work presents a unified and scalable framework to tackle the intertwined challenges of efficiency, fairness, and governance in the distributed deployment of Large Language Models (LLMs). By integrating topology-aware parallelism, communication-efficient scheduling, dynamic tensor rematerialization, real-time adversarial debiasing, and rule-based explainability, our architecture significantly enhances the performance, transparency, and ethical compliance of LLMs in practical settings. Experimental evaluations on C4, WikiText-103, RealNews, and BookCorpus demonstrate reductions in training time and memory usage, while fairness metrics show over 60% reduction in demographic AUC gaps. Our ISO-compliant governance layer ensures traceability, lifecycle audits, and policy enforcement, underscoring the importance of robust oversight in high-stakes applications.

Beyond technical innovation, the framework highlights critical policy implications—stressing the need for ethics-by-design in AI system development, standardized audit mechanisms for model accountability, and inclusive governance models that accommodate diverse stakeholder perspectives. These components are vital for the responsible integration of LLMs into real-world domains, such as healthcare, law, and finance. As part of future directions, we aim to incorporate neuromorphic computing principles to further enhance privacy preservation, energy efficiency, and adaptability in edge and federated environments. Ultimately, this research lays the groundwork for deploying LLMs that are not only powerful but also fair, interpretable, and governed responsibly.

Appendix A

Algorithm 1: Rematerialization decision process

 https://doi.org/10.1007/s44196-025-00992-4

Input: Maximum memory budget M_{max}, activation tensors A_l for each layer
Output: Binary decision vector z_l indicating rematerialization (1) or caching (0)

1. Initialize total memory usage $M_{total} = 0$
2. For each layer $l = 1$ *to* L:
   a. Compute memory cost $m(A_l)$
   b. Compute recomputation cost $\tau_{re}(A_l)$
3. Sort all layers by $\tau_{re}(A_l) / m(A_l)$ in descending order
4. For each sorted layer:
   a. If $M_{total} + m(A_l) \leq M_{max}$:
      i. Set $z_l = 0$ (cache activation)
      ii. Update $M_{total} += m(A_l)$
   b. Else:
      i. Set $z_l = 1$ (rematerialize activation)
5. Return vector z_l

Algorithm 2: Fairness-Constrained Training Cycle

Input: LLM model F, labeled data D, sensitive attribute a
Output: Debiased model F'

1. Initialize primary model F and adversary Adv
2. For each training epoch:
   a. Sample minibatch (x, y, a) from dataset D
   b. Forward pass through model: compute prediction $\hat{y} = F(x)$
   c. Compute task loss L_{task} = CrossEntropy$(\hat{y}, y)$
   d. Compute adversarial prediction $\hat{a} = Adv(\hat{y})$
   e. Compute adversarial loss $L_{adv} = CrossEntropy(\hat{a}, a)$
   f. Update Adv to minimize L_{adv}
   g. Update F to minimize $L_{task} - \lambda * L_{adv}$ using gradient reversal
3. Repeat until convergence criteria are met
4. Return debiased model F'

Algorithm 3: Symbolic rule extraction from attention weights

Input: Attention tensor A, vocabulary V, threshold k
Output: Set of symbolic rules R

1. Extract final-layer attention tensor $A \in \mathbb{R}^{nxn}$
2. For each head h in A:
   a. Compute token influence scores $s_i = \Sigma A[h][i, :]$
   b. Select top-k tokens based on s_i
3. For each selected token t:
   a. Retrieve corresponding word from vocabulary V
   b. Map token to symbolic clause using template (e.g., *"If [token], then [label]"*)
4. Aggregate clauses into rule set R
5. Return R

Appendix B

See below Table 13 here.

Table 13 Notation and symbol definitions

Symbol	Definition	Symbol	Definition
η_{data}	Loss from data-parallel segments	γ	Balancing weight between data and model throughput
z_ℓ	Binary flag for rematerialization at layer ℓ	τ_{ij}	Round-trip latency between nodes i, j
$m(A_\ell)$	Memory required to store activation of layer ℓ	$\tau_{re}(A_\ell)$	Recompute penalty for discarding activation A_ℓ
A_ℓ	Activation tensor at layer ℓ	$M_{\max}$	Maximum memory capacity of compute node
M_{total}	Accumulated memory usage during training	$G = (V, E)$	Communication graph with nodes V and edges E
λ	Fairness regularization weight	$\hat{y}$	Model prediction output
$\hat{a}$	Adversary prediction of sensitive attribute	L_{task}	Task-specific loss (e.g., cross-entropy)
L_{adv}	Adversarial loss for fairness	R	Extracted symbolic rules from attention
s_i	Token influence score in attention	ϵ, δ	Privacy budget for differential privacy
$BLEU$	Text generation quality metric	IOU	Explanation fidelity score (Intersection-over-Union)
F	LLM model function	Adv	Adversarial classifier module

Author Contributions U.I., A.A.A., M.A., H.S.A., and H.U. contributed to the conceptualization of the study. H.U., M.A., H.S.A., and A.A.A. were responsible for data curation and formal analysis. H.S.A. acquired funding and administered the project. H.U. provided resources and developed the software. A.A.A. and M.A. supervised the research. H.U., M.A., H.S.A., and A.A.A. carried out validation and visualization. H.U., M.A., and H.S.A. wrote the original draft. H.U., M.A., H.S.A., and A.A.A. reviewed and edited the manuscript. All authors reviewed the final manuscript.

Funding This research received no external funding.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Ferdaus, M.M., Abdelguerfi, M., Ioup, E., Niles, K. N., Pathak, K., Sloan, S.: Towards trustworthy AI: a review of ethical and robust large language models," arXiv preprint arXiv:2407.13934, (2024) [Online]. Available: https://doi.org/10.48550/arXiv.2407.13934
2. Chakladar, R. D.: Best practices for implementing large language models at scale. J. Core Eng. Manag. **7**(09):27–33, (2024) [Online]. Available: https://www.researchgate.net/publication/386872033
3. AlQenaei, Z. M.: Evolution and optimization of language model architectures: from foundations to future directions," in Computing and Machine Learning (CML 2024), Lecture Notes in Networks and Systems. In: Bansal, J. C., Borah, S., Hussain, S., Salhi, S. (eds.) vol. 1144, pp. 281–297. Singapore: Springer (2025) https://doi.org/10.1007/978-981-97-7839-3_16
4. Lyu, Y., Du, Y.: The ethical evaluation of large language models and its optimization. AI Ethics (2025). https://doi.org/10.1007/s43681-024-00654-9
5. Wang, S., Zhao, Y., Hou, X., Wang, H.: Large language model supply chain: a research agenda. ACM Trans. Internet Technol. (2024). https://doi.org/10.1145/3708531

6. Ullah, A., Qi, G., Hussain, S., Ullah, I., Ali, Z.: The role of LLMs in sustainable smart cities: applications, challenges, and future directions," arXiv preprint arXiv:2402.14596, (2024) [Online]. Available: https://doi.org/10.48550/arXiv.2402.14596
7. Sandfreni, Bansal, R.: Challenges in large language model development and AI Ethics," in Challenges in Large Language Model Development and AI Ethics, Universitas Esa Unggul, Indonesia and Insights2Techinfo, India: IGI Global, pp. 57, (2024) [Online]. Available: https://www.igi-global.com/chapter/challenges-in-large-language-model-development-and-ai-ethics/
8. Weidinger, L., et al.: Ethical and social risks of harm from language models," arXiv preprint arXiv:2112.04359, (2021). [Online]. Available: https://doi.org/10.48550/arXiv.2112.04359
9. Salierno, G., Leonardi, L., Cabri, G.: Generative AI and large language models in Industry 5.0: shaping smarter sustainable cities. Encyclopedia **5**(1), 30 (2025). https://doi.org/10.3390/encyclopedia5010030
10. Petroșanu, D.-M., Pîrjan, A., Tăbușcă, A.: Tracing the influence of large language models across the most impactful scientific works. Electronics **12**(24), 4957 (2023). https://doi.org/10.3390/electronics12244957
11. Tallam, K.: From autonomous agents to integrated systems, a new paradigm: orchestrated distributed intelligence," arXiv preprint arXiv:2503.13754, (2025) [Online]. Available: https://doi.org/10.48550/arXiv.2503.13754
12. Nay, J. J.: Large language models as fiduciaries: a case study toward robustly communicating with artificial intelligence through legal standards," arXiv preprint arXiv:2301.10095, (2023). [Online]. Available: https://doi.org/10.48550/arXiv.2301.10095
13. Alawida, M., Mejri, S., Mehmood, A., Chikhaoui, B., Abiodun, O.I.: A comprehensive study of ChatGPT: advancements, limitations, and ethical considerations in natural language processing and cybersecurity. Information **14**(8), 462 (2023). https://doi.org/10.3390/info14080462
14. Sun, N., Miao, Y., Jiang, H., Ding, M., Zhang, J.: From principles to practice: a deep dive into AI ethics and regulations," arXiv preprint arXiv:2412.04683, (2024) [Online]. Available: https://doi.org/10.48550/arXiv.2412.04683
15. Hussein, H., Gordon, M., Hodgkinson, C., Foreman, R., Wagad, S.: ChatGPT's impact across sectors: a systematic review of key themes and challenges. Big Data Cogn. Comput. **9**(3), 56 (2025). https://doi.org/10.3390/bdcc9030056
16. Tarkoma, S., Morabito, R., Sauvola, J.: AI-native interconnect framework for integration of large language model technologies in 6G systems," arXiv preprint arXiv:2311.05842, (2023). [Online]. Available: https://doi.org/10.48550/arXiv.2311.05842
17. Luo, J., et al.: Large language model agent: a survey on methodology, applications and challenges," arXiv preprint arXiv:2503.21460, (2025). [Online]. Available: https://doi.org/10.48550/arXiv.2503.21460
18. Bender, E. M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the Dangers of stochastic parrots: can language models be too big?," in Proc. 2021 ACM Conf. Fairness, Accountability, and Transparency (FAccT), Virtual Event, pp. 610–623, (2021) https://doi.org/10.1145/3442188.3445922.
19. Chen, C., Gong, X., Liu, Z., Jiang, W., Goh, S. Q., Lam, K.-Y.: Trustworthy, responsible, and safe AI: a comprehensive architectural framework for AI safety with challenges and mitigations," arXiv preprint arXiv:2408.12935, (2024). [Online]. Available: https://doi.org/10.48550/arXiv.2408.12935
20. Weidinger, L. et al.: Taxonomy of risks posed by language models," in Proc. 2022 ACM Conf. Fairness, Accountability, and Transparency (FAccT), Seoul, South Korea, pp. 214–229, (2022), https://doi.org/10.1145/3531146.3533088.
21. Li, Y., et al.: Large language models for manufacturing," arXiv preprint arXiv:2410.21418, (2024) [Online]. Available: https://doi.org/10.48550/arXiv.2410.21418
22. Aghaei, R., Kiaei, A. A., Boush, M., Vahidi, J., Barzegar, Z., Rofoosheh, M.: The potential of large language models in supply chain management: advancing decision-making, efficiency, and innovation," arXiv preprint arXiv:2501.15411, (2025) [Online]. Available: https://doi.org/10.48550/arXiv.2501.15411
23. Lovén, L., Bordallo López, M., Morabito, R., Sauvola, J. Tarkoma., et al.: Large language models in the 6G-enabled computing continuum: a White Paper," University of Oulu, Finland, (2025) [Online]. Available: https://urn.fi/URN:NBN:fi:oulu-202501211268
24. Wu, Q., et al.: "Surveying attitudinal alignment between large language models Vs. humans towards 17 sustainable development goals," arXiv preprint arXiv:2404.13885, (2024) [Online]. Available: https://doi.org/10.48550/arXiv.2404.13885
25. Kotyal, K., et al.: Advancements and challenges in artificial intelligence applications: a comprehensive review. J. Sci. Res. Rep. **30**(10), 375–385 (2024). https://doi.org/10.9734/jsrr/2024/v30i102465
26. Abel, M., et al.: Large language models in the 6G-enabled computing continuum: a white paper," HAL Open Science, (2025) [Online]. Available: https://hal.science/hal-xxxxxxx
27. Tejesh, B. S. S., Ramakrishnan, M.: Efficient hardware accelerator design for AES encryption using high-level synthesis techniques," In: Proc. Int. Conf. Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, pp. 1–6, (2024), https://doi.org/10.1109/ICIICS63763.2024.10859678.
28. Ahmed, S., et al.: Lightweight AES design for IoT applications: optimizations in FPGA and ASIC with DFA countermeasure strategies. IEEE Access **13**, 22489–22509 (2025). https://doi.org/10.1109/ACCESS.2025.3533611

29. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer," arXiv preprint arXiv:1910.10683, (2019). [Online]. Available: https://arxiv.org/abs/1910.10683
30. H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 782–791.
31. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers," *arXiv preprint arXiv:2005.00928*, (2020). [Online]. Available: https://doi.org/10.48550/arXiv.2005.00928
32. Tenney, I., et al.: The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models," *arXiv preprint arXiv:2008.05122*, (2020) [Online]. Available: https://doi.org/10.48550/arXiv.2008.05122

Authors and Affiliations

Mubarak Ahmad[1] · Abdulkadhem A. Abdulkadhem[2] · Umar Islam[3] · Hathal Salamah Alwageed[4] · Hanif Ullah[5] · A. Abdullah[6]

✉ A. Abdullah
abdullah.abdullah@path.ox.ac.uk

Mubarak Ahmad
100029@cwxu.edu.cn

Abdulkadhem A. Abdulkadhem
a.abdulkadhem@uomus.edu.iq

Umar Islam
umar.koh@gmail.com

Hathal Salamah Alwageed
hswageed@ju.edu.sa

Hanif Ullah
h.ullah@ulster.ac.uk

[1] School of Electronics and Information Engineering, Wuxi University, Wuxi 214105, Jiangsu Province, China

[2] Department of Cyber Security, College of Sciences, Al-Mustaqbal University, Hillah,, Babil, Iraq

[3] Department of Computer Science, IQRA National University, Swat Campus, Swat, KPK, Pakistan

[4] College of Computer and Information Science, Jouf University, Al-Jouf, Saudi Arabia

[5] School of Computing, Ulster University, Belfast, UK

[6] Sir William Dunn School of Pathology, Oxford University, Oxford, UK