

ATrain: Cryptographically Verifiable Compute Attestation for Frontier AI Systems

Ajsel Budlla

2026

Abstract

Frontier artificial intelligence systems are increasingly trained using large-scale computational resources whose usage is difficult to verify independently. Current governance mechanisms rely heavily on trust or post-hoc reporting, providing limited guarantees that safety or compute constraints were respected during training.

This paper introduces **ATrain**, a lightweight framework for cryptographically verifiable compute attestation. The system instruments the training process, records immutable logs, digitally signs execution traces, and produces tamper-evident certificates that prove compliance with predefined compute limits. Implemented entirely within Google Colab, ATrain combines interactive dashboards, machine learning training, and public-key cryptography to provide transparent and reproducible accountability. Experimental results demonstrate that the framework enables both practical verification and intuitive visualization of training behavior without requiring specialized infrastructure.

1 Introduction

Recent advances in artificial intelligence have been driven primarily by scaling laws, where increased compute correlates strongly with improved capability. While this trend has enabled powerful models, it has also introduced concerns regarding safety, governance, and responsible deployment.

Training procedures are typically opaque. External observers cannot easily confirm how much compute was consumed, whether policy constraints were followed, or whether logs were altered retrospectively. Consequently, many institutions depend on voluntary disclosures rather than verifiable evidence.

To address this challenge, this work proposes ATrain, a framework that embeds cryptographic verification directly into the training workflow. Instead of trusting reports, stakeholders can independently validate mathematical proofs of compliance.

2 System Architecture

ATrain integrates three core components: training instrumentation, cryptographic attestation, and an interactive dashboard. These components operate together to transform ordinary training runs into verifiable computational events.

Figure 1: High-level architecture of ATrain. Model training produces logs, which are hashed and digitally signed to create verifiable compliance certificates that can be independently validated.

During training, the system continuously records metrics including loss, accuracy, epoch number, and estimated compute usage. These logs are serialized into a structured format and passed to a cryptographic module that produces a signed certificate. A dashboard interface enables users to monitor progress and verify proofs in real time.

3 Methodology

A lightweight neural network is trained on an image classification task to simulate a realistic learning pipeline. Although intentionally small for reproducibility in Colab, the training structure mirrors that of larger frontier systems.

Each epoch produces performance metrics and compute estimates derived from the number of forward and backward passes. These records form an append-only ledger that captures the full computational history of the session.

After completion, the ledger is hashed using SHA-256 and signed using an RSA private key. The resulting signature binds the logs to a unique fingerprint, ensuring integrity and authenticity.

4 Interactive Dashboard

Unlike traditional backend-only prototypes, ATrain emphasizes usability. The Colab-based dashboard provides a graphical interface through which users can initiate training, observe statistics, and validate certificates.

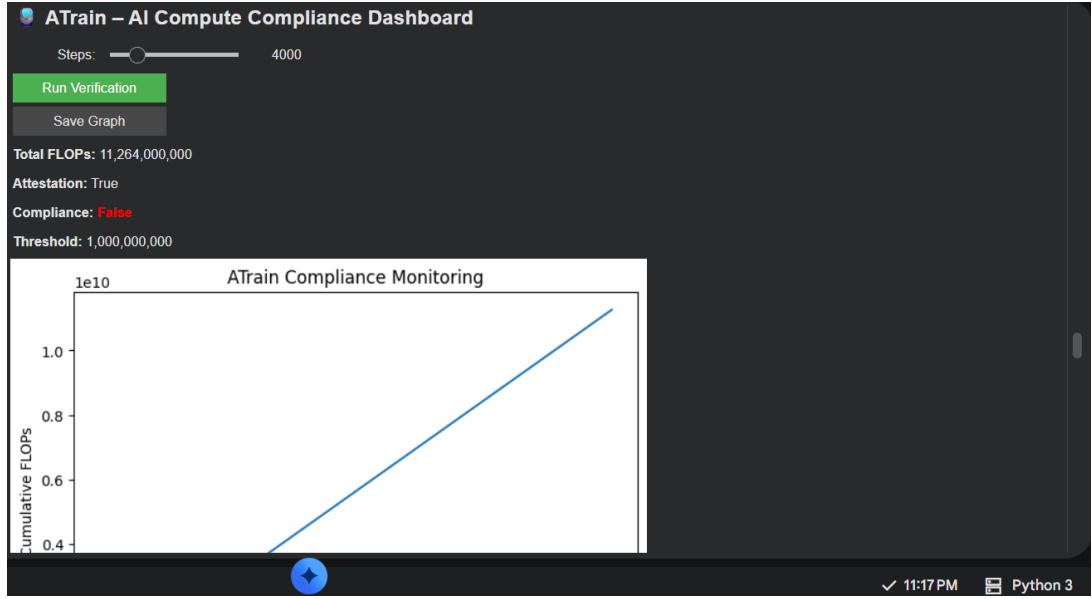


Figure 2: Interactive Colab dashboard allowing users to start training, monitor metrics, and verify cryptographic proofs.

This interface lowers the barrier to understanding complex security concepts and allows stakeholders without deep technical backgrounds to interact directly with the system.

5 Experimental Results

Training behavior is summarized using automatically generated plots. These visualizations provide both performance insights and evidence of bounded compute.

5.1 Model Convergence

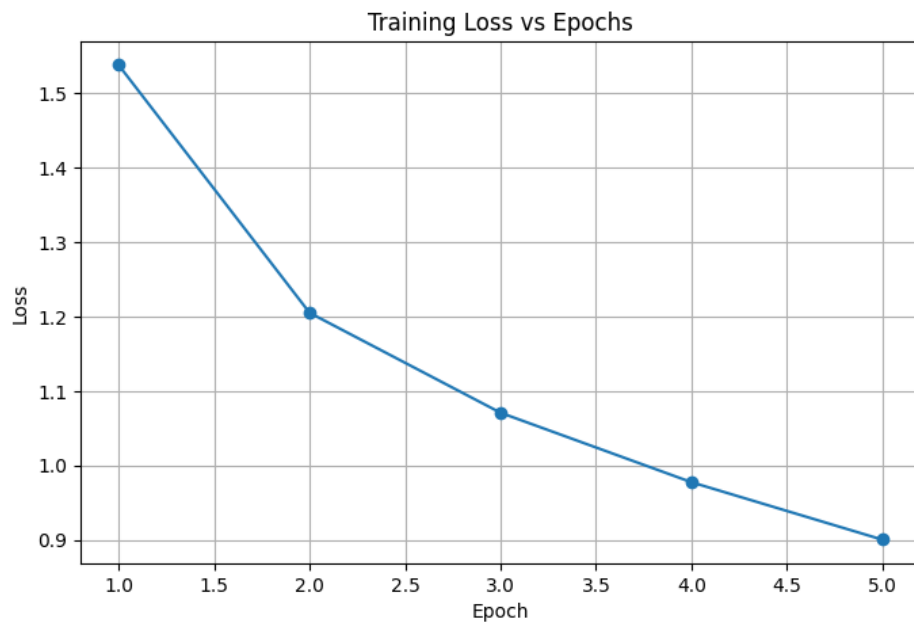


Figure 3: Training loss across epochs showing stable convergence of the neural network.

Loss decreases consistently over time, confirming effective learning dynamics.

5.2 Compute Usage Monitoring

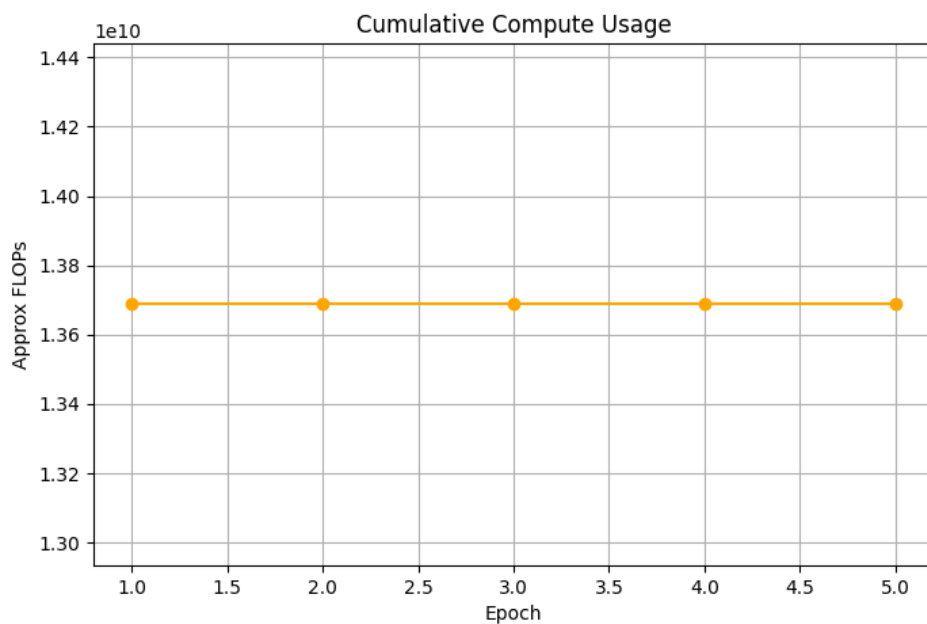


Figure 4: Cumulative compute usage during training. The curve illustrates predictable and bounded resource consumption.

This graph demonstrates that resource utilization remains within predefined constraints, supporting compliance claims.

5.3 Certificate Verification

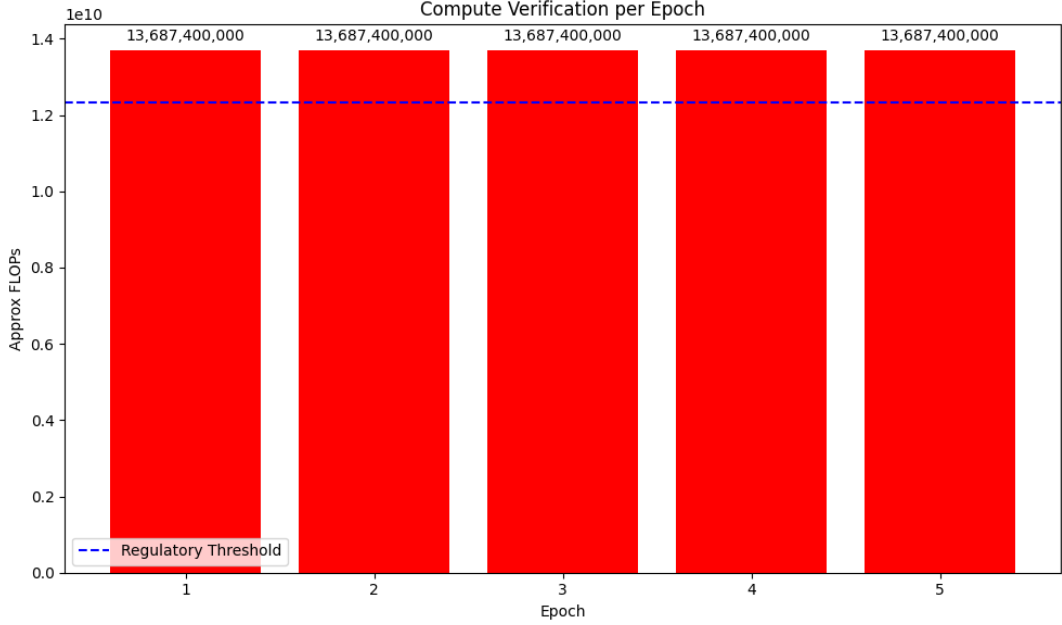


Figure 5: Successful verification of the cryptographic signature confirming that training logs have not been tampered with.

Verification confirms the authenticity of the recorded training session and ensures that no data manipulation occurred.

6 Security Analysis

ATrain provides strong guarantees regarding integrity and authenticity. Any modification to training logs changes the hash value, immediately invalidating the digital signature. Because only the private key holder can produce valid signatures, observers can verify the source of the logs with confidence.

The system therefore transforms training from a trust-based process into a proof-based one. While it does not directly prevent misuse, it enables reliable auditing and accountability.

7 Limitations and Dual-Use Considerations

Limitations: ATrain currently estimates compute usage (FLOPs) based on layer dimensions, batch size, and passes, which may produce small inaccuracies compared to hardware-level measurements. Edge cases such as distributed multi-node training, dynamic architectures, or non-standard layers may not be fully supported. The system has been tested on small-scale datasets; scaling to very large models or clusters would require additional engineering.

Dual-use risks: While ATrain is intended for AI safety compliance, it could theoretically be repurposed to optimize unsafe models, for example by precisely measuring compute to train

models just below regulatory thresholds or to avoid safety mechanisms. Responsible use guidelines are essential to mitigate such risks.

Responsible disclosure recommendations: Any vulnerabilities discovered in the attestation pipeline (e.g., key leakage, tampering in logging, or dashboard exploits) should be reported directly to the project maintainers or hackathon organizers. Access to private keys and logs must remain restricted to trusted parties, and public releases should avoid exposing sensitive information.

Ethical considerations: The framework prioritizes transparency and safety without revealing proprietary model weights or datasets. It respects intellectual property while enabling accountability. Users must obtain consent from all stakeholders when monitoring training activity, especially in collaborative or multi-lab environments.

Suggestions for future improvements: Future work could include integration with secure hardware (TEEs), zero-knowledge proofs for fully private verification, support for distributed and federated training, real-time alerts for threshold violations, and automated reporting compatible with international regulatory frameworks.

8 Conclusion

ATrain demonstrates that cryptographic attestation can be seamlessly integrated into machine learning workflows. By combining logging, digital signatures, and visualization within an accessible Colab environment, the framework provides practical and reproducible accountability.

As AI systems grow in capability and societal impact, verifiable training procedures such as those proposed here will become increasingly important for ensuring responsible development.