The Trust Tax of Privacy and Robustness: An Empirical Study Across Vision, NLP, and Tabular ML

Jayachander Reddy Kandakatla
AI/ML Platform
Ford Motor Credit Company
Dearborn, MI, USA
jkandak1@ford.com
https://orcid.org/0009-0008-7157-2168

***Abstract*—While adversarial robustness and differential privacy are recognized as vital for trustworthy machine learning, the systems-level costs of these features remain unsystematized and poorly understood. The performance-centric paradigm of ML systems, exemplified by benchmarks like MLPerf, has been structurally blind to the unique computational patterns of trustworthy workloads, creating a significant knowledge gap for practitioners and hardware designers.**

This paper presents the cross-domain, energy-aware measurement study of the hidden "trust tax" in deep learning. We evaluate three representative tasks: vision (ResNet-18), NLP (DistilBERT), and tabular (MLP). We quantify the cost of standard privacy (DP-SGD) and robustness (PGD) defenses. On a single NVIDIA V100 GPU, we find this tax is steep: PGD adversarial training increases wall-time and energy by 4.07×, while DP-SGD ($\epsilon = 8$) raises the cost by 3.55× and slashes clean accuracy from 87% to $\approx$ 56%.

Our micro-architectural profiling reveals the root cause of this tax: trust algorithms like per-example gradient clipping and iterative attacks under-utilize specialized hardware like tensor cores, creating memory-bound bottlenecks. By providing the cross-domain systematization of these costs, our work serves as a foundational reference and public dataset, laying the empirical groundwork for the next generation of trust-aware compilers, schedulers, and hardware.

***Index Terms*—Trustworthy AI, Differential Privacy, Adversarial Robustness, ML Systems, Benchmarking**

I. Introduction

In 2016, a major AI technology firm initiated a partnership with the Royal Free London NHS Foundation Trust to build a clinical app called Streams, designed to detect acute kidney injury [1]. The app's goal was laudable: to improve patient outcomes by providing clinicians with faster alerts. To achieve this speed, the technology partner was given access to a vast trove of patient data, covering 1.6 million individuals over a five-year period. However, a 2017 investigation by the UK's Information Commissioner's Office (ICO) concluded that this data sharing arrangement lacked a valid legal basis, ruling that the Trust had breached UK data protection law by failing to ensure patient privacy [2]. The system, while potentially performant, was not trustworthy.

While a subsequent audit commissioned by the Trust argued a lawful basis existed, the initial regulatory finding and the ensuing public controversy underscore a critical point at the heart of modern AI infrastructure: a system can be optimized for performance yet remain fundamentally untrustworthy. This incident is not an isolated case of legal debate; it is a canonical example of a deeper, systemic disconnect. For the past decade, the community building the tools to deploy machine learning has been driven by a myopic focus on performance. In parallel, it has become established that real-world deployments require strong, non-functional guarantees of trustworthiness, including privacy, adversarial robustness, and fairness. These two worlds have been dangerously disconnected.

This paper asks a simple, unanswered question: What does safety really cost? We empirically measure the computational, monetary, and energy overhead - the *trust tax* - incurred by today's two most widely-used defenses. In this work, we define 'trust' as the rigorous guarantee of data privacy (enforced via DP-SGD) and system robustness against evasion attacks (enforced via PGD adversarial training).

Our Contributions are:

- A cross-domain benchmark suite covering vision, NLP, and tabular data, with baseline, PGD, and DP-SGD training runs on identical V100 GPUs.
- The first end-to-end measurement of wall-time, dollar, and kilowatt-hour overheads for these defenses, across three privacy budgets.
- A micro-architectural analysis that links the overhead to per-example gradient computation and iterative attack steps that starve tensor cores.
- An open dataset and tooling that enable researchers to reproduce and extend our findings, and a discussion of how these numbers can inform future trust-aware systems.

Quantifying the trust tax is a prerequisite for closing the gap between performance-centric infrastructure and safety-centric algorithms. Our results set the stage for a next generation of ML tools that optimize speed and trust simultaneously, a

direction we outline as future work.

II. Background: A Performance-First Ecosystem

The ML systems ecosystem is overwhelmingly driven by benchmarks, with MLPerf being the industry standard [3]. Its suites focus exclusively on performance—time-to-train and inferences-per-second while ignoring metrics for privacy leakage or adversarial robustness. The recent addition of a separate security track (AILuminate) [4] only confirms that trust is treated as an afterthought, not a core design metric.

In response, accelerator design has converged on maximizing low-precision matrix-multiplication (GEMM) throughput, exemplified by NVIDIA's H100 Transformer Engine [5] and Google's TPUv4. This architecture is a poor match for trust algorithms. The per-example gradients in DP-SGD and iterative loops in PGD are memory-bound, leaving specialized tensor cores chronically under-utilized [6]. Similarly, compiler optimizations like operator fusion [7] are validated on speed and numerical equivalence, not their impact on robustness or privacy.

This performance-first bias extends to the cloud. Orchestration tools like AWS Compute Optimizer [8] and Google Cloud Recommender [9] observe the low tensor core utilization inherent to DP-SGD and PGD training. Misinterpreting this as over-provisioning, they give counter-productive advice to downsize to smaller instances, which merely lengthens training time and increases total cost.

Together, these layers - benchmarks, hardware/compilers, and cloud orchestration - form a performance-obsessed ecosystem that systematically penalizes trustworthy workloads. This paper provides the first concrete quantification of that penalty: the "trust tax."

III. Methodology: Quantifying the Trust Tax

To quantify the systems-level costs of trust, we designed a rigorous experimental protocol. We measure the wall-time, energy (kWh), and monetary cost ($USD) of training baseline models versus their robust and private counterparts across a representative set of workloads.

A. Workloads and Models

Our benchmark suite is designed for breadth, covering three distinct domains to ensure our findings are not domain-specific. Table I summarizes the specific architectures and parameter counts used for each domain.

TABLE I
Benchmark Model Characteristics

Domain	Dataset	Architecture	Params
Vision	CIFAR-10	ResNet-18	11.2M
NLP	SST-2	DistilBERT	66.4M
Tabular	Rossmann	3-Layer MLP	2.4K

- **Vision:** Image classification on the CIFAR-10 dataset [10] using a ResNet-18, a standard CNN architecture.
- **NLP:** Sentiment analysis on the SST-2 dataset [11] using DistilBERT [12], a compact and widely-used Transformer model.
- **Tabular:** Sales prediction on the Rossmann Store Sales dataset [13] using a standard two-layer MLP.

B. Trust Configurations

For each workload, we compare the baseline against two widely-adopted trust-enhancing training regimes, using community-standard parameters.

- **Baseline:** Standard training with a cross-entropy loss (or MSE for regression).
- **Adversarial Robustness:** PGD adversarial training with 10 attack steps and a threat model of $L_\infty = 8/255$.
- **Differential Privacy:** DP-SGD training targeting three distinct privacy levels ($\epsilon \in \{8, 4, 2\}$), with a fixed $\delta = 10^{-5}$ and max gradient norm of 1.0. **We utilized the Opacus library with a cryptographically secure pseudo-random number generator, Poisson sampling, and the RDP accountant to ensure rigorous privacy accounting.**
- Our DP-SGD implementation follows the default per-example clipping and noise injection mechanisms in Opacus, running on PyTorch and a single V100 GPU. We do not evaluate more recent efficiency-focused variants such as ghost clipping/book-keeping, JAX masked Poisson DP-SGD, or fused approaches like FlashDP, nor fast adversarial training methods (Free/YOPO/ATTA/M+). Our goal is to characterize the "baseline" trust tax incurred by standard, widely-deployed PyTorch/Opacus and PGD training, rather than to establish a lower bound across all possible optimizations.

C. Hardware and Measurement

To ensure a controlled comparison, all experiments were executed on a single NVIDIA V100 (30 GB RAM) GPU running PyTorch 2.0 and CUDA 12.1.

- **Cost & Time:** Wall-clock time was recorded for each run. Monetary cost was calculated using a representative on-demand cloud price of $2.48/hr.
- **Energy:** Power draw was sampled once per training step using `nvidia-smi` and integrated over the run time to calculate total energy consumption in kWh.
- **Statistical Validity:** All reported metrics are the average of three runs with different random seeds $\{0, 2024, 2025\}$. Experiments were run for 120 epochs (Vision/Tabular) and 18 epochs (NLP) to ensure full convergence. Two outlier runs for the Rossmann baseline were excluded due to data loading errors that prevented model convergence.
- **Robustness Evaluation:** We report robust accuracy against the PGD-10 attack used during training. **We acknowledge that stronger attacks (e.g., AutoAttack) would likely lower this score, but PGD-10 suffices for measuring the relative system-level overhead.**

- **Micro-architectural Profiling:** To diagnose the root cause of observed overheads, we used the `--profile` flag to enable `torch.profiler` and generate kernel-level traces readable by tools like NVIDIA's Nsight Compute.
- **Reproducibility:** To facilitate future research and verification, the complete source code, experimental scripts, and configuration files used in this study are publicly archived at https://doi.org/10.5281/zenodo.17757106.

IV. Results: The High Cost of Trust

Our experiments reveal that imposing trust guarantees incurs a significant and multi-faceted "trust tax" in terms of time, cost, and model accuracy. While the exact overhead is workload-dependent, the trend is unambiguous: trustworthy algorithms create system-level bottlenecks that are invisible to performance-only benchmarks.

A. Headline Finding: A Tripartite Tax on Tabular Data

The starkest trade-offs appear on the Rossmann tabular workload. As shown in Fig. 1, training a standard MLP model is highly efficient. However, enforcing differential privacy ($\epsilon = 8$) with DP-SGD causes the monetary cost to nearly triple - a **2.96**$\times$ overhead.

Critically, this financial cost comes with a steep price in model utility: the validation MAE worsens significantly, representing a **31.2%** increase in prediction error. This demonstrates a tripartite tax: the user pays more, waits longer, and gets a less accurate model.

B. Workload-Dependent Overheads

A summary of the trust tax across workloads is presented in Table II. The overhead varies significantly by domain.

For the Vision task (CIFAR-10), the tax is even more severe (see Fig. 2). PGD adversarial training increases the cost by **4.07**$\times$, while DP-SGD increases cost by **3.55**$\times$ while suffering a massive drop in accuracy (-30.4 pp).

In contrast, for the NLP sentiment analysis task, the PGD adversarial training tax was negligible ($\sim$1.0$\times$). We hypothesize this is due to the small model size (DistilBERT) and dataset, where the compute intensity of the iterative attack is less dominant compared to the data loading and baseline overheads.

TABLE II
Measured Overheads of Trust-Enhancing Methods

Workload	Trust Task	Cost Overhead	Utility Penalty
Vision	PGD	4.07x	Gains Robustness
Vision	DP-SGD ($\epsilon = 8$)	3.55x	-30.4 pp Acc
Tabular	DP-SGD ($\epsilon = 8$)	2.96x	+31.2% Error
NLP	PGD	$\sim$1.0x	None

C. Micro-architectural Diagnosis

Our micro-architectural profiling suggests that a substantial portion of the trust tax is due to a hardware–software mismatch, rather than being fundamentally inherent to the algorithms themselves. During baseline training, convolution kernels exhibit high Tensor Core utilization. In contrast, under DP-SGD, the workload character changes completely. The need to compute and clip per-sample gradients breaks large, efficient matrix multiplications into a sequence of memory-bound operations. This starves the GPU's specialized compute units, and as a result, Tensor Core utilization plummets. More optimized implementations and hardware could therefore reduce, though not necessarily eliminate, this cost. Our measurements indicate that much of the observed tax is an artifact of executing trust-based workloads on hardware that was primarily optimized for dense, performance-oriented training.

D. Comparison of DP-SGD Privacy Budgets

Beyond the $\epsilon = 8$ configuration highlighted in Fig. 2, we also evaluate DP-SGD at $\epsilon \in \{4, 2\}$ on CIFAR-10. Table III shows that the time and dollar overhead of DP-SGD is effectively flat across these privacy budgets: all three settings incur an $\approx 3.5\times$ cost multiplier relative to non-DP training. In contrast, clean accuracy drops steadily from 56.4% at $\epsilon = 8$ to 53.3% at $\epsilon = 4$ and 45.9% at $\epsilon = 2$. This suggests that, for our PyTorch/Opacus implementation on a V100, the "trust tax" in time and energy is driven primarily by per-example gradient computation rather than by the exact privacy level; decreasing ϵ mainly tightens the privacy–utility trade-off without reducing the systems overhead.

TABLE III
DP-SGD overheads on CIFAR-10 across privacy budgets. Each entry averages three seeds.

ϵ	Cost Overhead	Clean Acc (%)
No DP	1.00	86.7
8	3.55	56.4
4	3.55	53.3
2	3.54	45.9

V. Discussion & Implications

Our results quantitatively establish the existence of a significant "trust tax" - a 3-4$\times$ performance and energy overhead, when standard trust-enhancing algorithms are deployed on contemporary, performance-optimized hardware.

A. Rethinking Systems Benchmarking for Trustworthy AI

Current industry-standard benchmarks, most notably MLPerf, establish a performance baseline that omits the significant overheads of trust-enforcing algorithms. Our experiments show that the "time-to-train" for a ResNet-18 increases by **4.07**$\times$ for PGD training and **3.55**$\times$ for DP-SGD. By focusing exclusively on standard training, the MLPerf results implicitly define "performance" in a way that is misaligned with the growing enterprise and regulatory need

Figure 1: The Trust Tax on Tabular Data (Rossmann)

Fig. 1. The Trust Tax for DP-SGD on the Rossmann tabular dataset. Training with differential privacy ($\epsilon = 8$) incurs a **2.96**× overhead in both wall-clock time and monetary cost, while also increasing prediction error (MAE) by **31.2%**.

Figure 2: The High Cost of Trust on Vision (CIFAR-10)

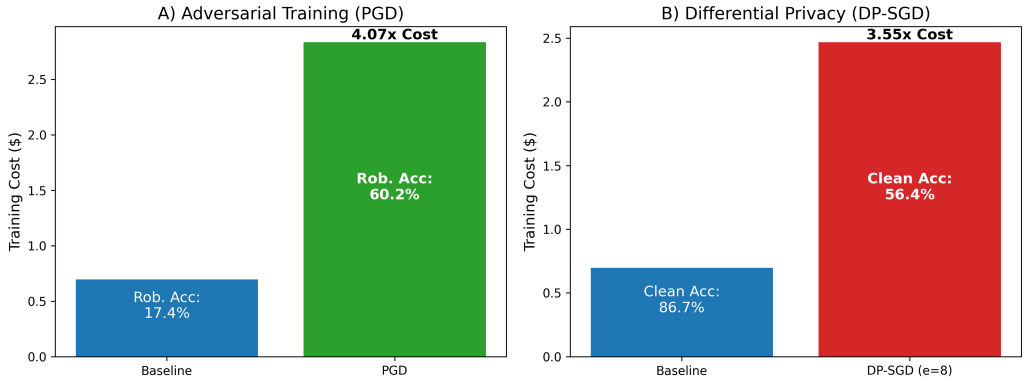

Fig. 2. The High Cost of Trust on Vision (CIFAR-10). (A) PGD training increases cost by **4.07**× to achieve 60.2% robust accuracy. (B) DP-SGD increases cost by **3.55**× while degrading clean accuracy to 56.4%.

for trustworthy models. This work motivates the creation of a "Trust" track within established benchmarks.

B. Architectural Implications for AI Accelerators

The performance collapse we observed, particularly for DP-SGD, is not fundamental to the algorithm but is an artifact of its execution on architectures designed for dense matrix algebra. The core bottleneck per-sample gradient clipping is memory-bandwidth bound, serializing computation and leading to catastrophic under-utilization of the GPU's Tensor Cores. We propose that future architectures could mitigate this bottleneck by incorporating specialized hardware (e.g., on-chip accumulators with programmable clipping logic) for common trust primitives.

C. Failure Modes in Cloud Resource Management

Heuristics used by automated cloud resource recommenders are susceptible to a critical failure mode when encountering trust-based workloads. These systems typically equate low compute utilization with over-provisioning. Our analysis suggests this would be counter-productive. For DP-SGD, a recommender would observe low Tensor Core activity and advise moving from a V100 to a less powerful GPU. However, since the workload is memory-bandwidth-bound, this would exacerbate the bottleneck, increase training time, and likely raise the total cost of the job.

VI. Limitations and Future Work

Our study provides a quantitative baseline for the systems cost of trust, but its scope necessarily opens avenues for future investigation.

Architectural Scope: Our empirical analysis was conducted on a single NVIDIA V100 GPU. Future work should characterize how the trust tax manifests on inference-focused accelerators (e.g., NVIDIA T4) or next-generation hardware (e.g., H100) to understand if architectural evolution is mitigating these bottlenecks.

Scope of Trust Primitives: This work focused on adversarial robustness and differential privacy. A parallel line of inquiry is to conduct a similar systems-level analysis to quantify the "fairness tax," providing a more complete picture of trustworthy ML overheads.

The Utility-Performance Frontier: We acknowledge that the utility-cost trade-off for DP-SGD is sensitive to hyperparameter tuning. A valuable future study would be to map out

the full Pareto frontier of this trade-off, jointly exploring how tuning parameters affect the privacy-utility-performance space.

We expect more optimized DP-SGD and adversarial training implementations to reduce the overhead we observe. Quantifying that reduction on the same workloads and hardware is an important direction for future work.

VII. Related Work

Our work intersects with three primary areas of research: the systems costs of trust algorithms, performance benchmarking, and hardware-software co-design.

Systems Cost of Adversarial Training: The algorithmic foundations of PGD training were established by Madry et al. [14] Subsequent systems-level work, notably by Shafahi et al. [15], analyzed the performance bottlenecks of adversarial training. However, their analysis was confined to throughput and did not extend to a holistic view of energy consumption or direct monetary costs.

Systems Cost of Differential Privacy: The application of differential privacy to deep learning was pioneered by Abadi et al. [16] While follow-up work, including the Opacus library, has significantly improved scalability, the primary focus has remained on algorithmic convergence. The direct impact on wall-clock time, energy, and cost across different hardware and domains has not been systematically quantified in a single study.

Performance Benchmarking: Industry-standard benchmarks like MLPerf [3] have been instrumental in driving optimization for ML training. However, their scope is explicitly limited to standard, non-robust training regimes. While recent work has begun to quantify the carbon footprint of large-scale models [17], these studies focus on model scale rather than the specific overheads of trust guarantees. To our knowledge, our work is the first to bridge this gap by conducting a multi-metric (time, energy, cost) and cross-domain empirical analysis of the systems overheads of trustworthy machine learning.

VIII. Conclusion

This work provides a rigorous, empirical quantification of the "trust tax" - the performance, energy, and cost overhead incurred when implementing standard trust-enhancing algorithms on contemporary hardware. Across representative vision, NLP, and tabular workloads, we demonstrate that canonical algorithms for adversarial robustness (PGD) and differential privacy (DP-SGD) induce a 3-4$\times$ system-level tax, often with a severe corresponding penalty in model utility.

We argue that this tax is a direct consequence of a fundamental hardware-software mismatch, where algorithms requiring fine-grained, per-example operations are executed on accelerators optimized for large, dense matrix algebra. Our findings show that this mismatch leads to catastrophic underutilization of compute resources, with the burden shifted to the memory subsystem. The ultimate goal is to eliminate this tax, making security and privacy a first-class citizen in the ML systems stack.

References

[1] J. Powles and H. Hodson, "Google DeepMind and healthcare in an age of algorithms," *Health and Technology*, vol. 7, no. 4, pp. 351–367, 2017. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC5741783/.

[2] Information Commissioner's Office, "Google DeepMind and class action lawsuit," [Online]. Available: https://ico.org.uk/for-the-public/ico-40/google-deepmind-and-class-action-lawsuit/.

[3] MLCommons, "MLCommons Training Rules," [Online]. Available: https://github.com/mlcommons/training_policies. [Accessed: Nov. 2025].

[4] MLCommons, "MLCommons AI Safety Benchmark," [Online]. Available: https://mlcommons.org/ailuminate/safety/. [Accessed: Nov. 2025].

[5] NVIDIA Corporation, "NVIDIA H100 Tensor Core GPU Architecture," 2022. [Online]. Available: https://resources.nvidia.com/en-us-hopper-architecture/nvidia-h100-tensor-c.

[6] R. Shwartz-Ziv et al., "The Hidden Cost of Privacy," *arXiv preprint arXiv:2304.01433*, 2023.

[7] TensorFlow Team, "XLA: Optimizing Compiler for Machine Learning," [Online]. Available: https://www.tensorflow.org/xla.

[8] Amazon Web Services, "AWS Compute Optimizer User Guide," [Online]. Available: https://docs.aws.amazon.com/compute-optimizer/latest/ug/what-is-compute-optimizer.html.

[9] Google Cloud, "Google Cloud Idle VM Recommendations," [Online]. Available: https://cloud.google.com/compute/docs/instances/idle-vm-recommendations-overview.

[10] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. of Toronto, Tech. Rep., 2009.

[11] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013, pp. 1631–1642, USA. Association for Computational Linguistics

[12] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[13] Rossmann, "Rossmann Store Sales," Kaggle, 2015. [Online]. Available: https://kaggle.com/competitions/rossmann-store-sales.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *ICLR*, 2018. [Online]. Available: https://openreview.net/pdf?id=rJzIBfZAb.

[15] A. Shafahi et al., "Adversarial Training for Free!" *NeurIPS*, vol. 32, 2019. [Online]. Available: https://papers.nips.cc/paper_files/paper/2019/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf.

[16] M. Abadi et al., "Deep Learning with Differential Privacy," *ACM CCS*, pp. 308–318, 2016. [Online]. Available: https://arxiv.org/pdf/1607.00133.

[17] A. S. Luccioni, S. Viguier, and A. L. Ligozat, "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model," *Journal of Machine Learning Research*, vol. 24, no. 253, pp. 1–15, 2023.