Privacy Isn't Free: Benchmarking the Systems Cost of Privacy-Preserving ML

Nnaemeka Obiefuna [1] **Samuel Oyeneye** [1] **Similoluwa Odunaiya** [1] **Iremide Oyelaja** [1] **Steven Kolawole** [1 2]

Abstract

Privacy-preserving machine learning techniques are increasingly deployed in hybrid combinations, yet their system-level interactions remain poorly understood. We introduce PRIVACYBENCH, a comprehensive framework that reveals non-additive behaviors in privacy technique combinations with significant performance and resource implications. Evaluating Federated Learning (FL), Differential Privacy (DP), and Secure Multi-Party Computation (SMPC) across ResNet18 and ViT models on medical datasets, we uncover striking disparities: while FL and FL+SMPC preserve utility with modest overhead, FL+DP combos exhibit severe convergence issues—accuracy drops from 98% to 13%, training time increases 16×, and energy consumption rises 20×. PRIVACYBENCH provides the first systematic evaluation framework to jointly track utility, cost, and environmental impact across privacy configs. These findings demonstrate that privacy techniques cannot be treated as modular components and highlight critical considerations for deploying privacy-preserving ML systems in resource-constrained environments.

1. Introduction

Privacy-preserving machine learning (PPML) systems are increasingly deployed in production environments where practitioners must balance theoretical privacy guarantees with practical system constraints. While individual techniques like Federated Learning (FL), Differential Privacy (DP), and Secure Multi-Party Computation (SMPC) are well-studied, their hybrid deployments—now standard in industry (Geyer et al., 2017; Rahaman et al., 2024)—remain poorly characterized in terms of computational overhead, energy consumption, and convergence behavior.

This gap has significant practical implications. Through systematic benchmarking, we demonstrate that privacy techniques exhibit complex interaction effects that can severely impact system performance. Adding DP to FL doesn't merely reduce accuracy—it can cause training convergence failure (98% to 13%) while dramatically increasing computational costs (16× training time, 20× energy consumption).

Current evaluation practices treat privacy methods in isolation (Caldas et al., 2018; Wei et al., 2023), but production systems require understanding of how technique combinations affect deployment feasibility. The lack of systematic resource monitoring and configuration management tools makes it difficult for practitioners to make informed design decisions about privacy system architecture.

We introduce PRIVACYBENCH, a reproducible benchmarking framework that quantifies the full system cost of privacy-preserving ML deployments. Our framework addresses key evaluation gaps through: (1) systematic evaluation of privacy technique combinations (FL, DP, SMPC), (2) integrated resource monitoring including energy tracking via CodeCarbon (Posthuma, 2025), (3) comprehensive instrumentation of training time, memory usage, and convergence behavior, and (4) YAML-based configuration management enabling controlled comparative studies across privacy techniques and model architectures.

Using ResNet18 and ViT models on medical imaging datasets, we systematically evaluate individual privacy techniques and their hybrid combinations, providing the first comprehensive analysis of privacy-utility-cost trade-offs in realistic deployment scenarios. Our engineering-focused benchmark reveals which privacy combinations are computationally viable and which may be prohibitively expensive, supporting evidence-driven privacy system design.

Key contributions: (1) First systematic evaluation of hybrid privacy technique combinations across utility, compute cost, and energy; (2) Reproducible benchmarking framework with expansive resource monitoring; (3) Evidence of non-additive privacy behaviors ranging from beneficial to severely detrimental; and (4) Open-source benchmarking platform for community extension.

[*]Equal contribution [1]ML Collective [2]Carnegie Mellon University. Correspondence to: Nnaemeka Obiefuna <itsdonmonc@gmail.com>, Samuel Oyeneye <samueloyeneye1@gmail.com>.

Proceedings of the 3^{rd} Efficient Systems for Founddation Models Workshop at the International Conference on Machine Learning. PMLR 267, 2025.

2. Related Work

Privacy-preserving machine learning techniques—Federated Learning (FL) (Zhang et al., 2021), Differential Privacy (DP) (Wei et al., 2021), and Secure Multi-Party Computation (SMPC) (Zhou et al., 2024)—are increasingly deployed in hybrid configurations. FL+DP combinations are standard in industry (El Ouadrhiri & Abdelhadi, 2022), while SMPC provides secure aggregation in federated settings (Adhikari & Adhikari). However, computational and energy costs of hybrid deployments remain poorly characterized.

Current benchmarks focus on individual privacy methods. LEAF (Caldas et al., 2018) provides standardized FL datasets, FedScale (Lai et al., 2022) enables large-scale FL simulations, and DPMLBench (Wei et al., 2023) evaluates DP methods across datasets. No existing benchmark evaluates hybrid privacy configurations or systematically tracks computational resources—training time, memory usage, and energy consumption—that determine deployment feasibility.

PRIVACYBENCH addresses these gaps through systematic evaluation of hybrid privacy combinations with integrated resource monitoring including energy tracking via CodeCarbon (Rajput et al., 2024), and reproducible YAML-based configuration management.

3. Benchmark Design

PRIVACYBENCH provides a systematic evaluation environment emphasizing reproducible experimentation and comprehensive resource monitoring through modular configuration management, automated instrumentation, and extensible architecture.

3.1. System Architecture and Privacy Techniques

The benchmark features a plugin-based architecture enabling systematic evaluation of privacy technique combinations. Core components include YAML-based configuration files for experimental specification without code modification, automated tracking of training time and energy consumption via CodeCarbon, runtime privacy toggles for controlled studies, and deterministic execution with comprehensive seed control (configuration examples in Appendix C.2).

We evaluate three PPML techniques individually and in hybrid combinations: **Federated Learning (FL)** via Flower framework with 3-client non-IID partitions and 5 federated rounds; **Differential Privacy (DP)** using Opacus with privacy budgets $\epsilon = 0.5, 1.0$, $\delta = 1 \times 10^{-5}$; and **Secure Multi-Party Computation (SMPC)** through custom secure aggregation with additive secret sharing. Hybrid configurations include FL+DP and FL+SMPC, reflecting industry practice. Complete parameters are in Appendix B.

3.2. Models, Datasets, and Instrumentation

We employ ResNet18 (11.7M parameters, learning rate 2.1×10^{-4}) and ViT-Base (86.6M parameters, learning rate 5×10^{-5}) using Adam optimizer for up to 50 epochs. We employ early stopping—monitoring validation accuracy with a 7-epoch patience—and save the best model checkpoint. Evaluation uses privacy-sensitive medical imaging: Alzheimer MRI binary classification and ISIC skin lesion multi-class classification with non-IID federated partitioning (details in Appendix B.2).

PRIVACYBENCH's instrumentation captures standard classification metrics (accuracy, F1-score, MCC, precision, recall, ROC-AUC), resource profiling (training time, CPU/GPU memory), and energy monitoring (kWh consumption, CO_2 emissions via CodeCarbon). All experiments use standardized hardware for consistent measurements (specifications in Appendix C). The complete benchmark will be publicly released for community validation and extension.

4. Experimental Results

Our systematic evaluation reveals that privacy techniques exhibit non-additive behaviors with significant implications for system deployment. Through PRIVACYBENCH's comprehensive instrumentation, we demonstrate that while some privacy combinations preserve utility with acceptable overhead, others exhibit severe performance degradation and resource requirements that may render them impractical for deployment.

4.1. Beyond Modular Assumptions

Table 1: Key Performance and Resource Metrics (Alzheimer MRI Classification)

Config--uration	Accu--racy	Training Time (min)	Energy (kWh)	Overhead Factor
CNN Baseline	0.98	9.8	0.026	1.0×
FL	0.98	14.7	0.036	1.4×
FL+SMPC	0.98	17.5	0.041	1.6×
FL+DP	0.13	235.6	0.734	24.0×
ViT Baseline	0.99	54.1	0.119	1.0×
FL	0.96	40.1	0.104	0.9×
FL+SMPC	0.96	40.4	0.104	0.9×

The results reveal striking disparities in how privacy techniques interact. Federated learning maintains near-baseline performance for both architectures while adding modest computational overhead (40-50% increase in training time). Notably, FL can even provide efficiency gains for transformer architectures, reducing ViT training time by 26% compared to centralized training.

SMPC-based secure aggregation composes well with

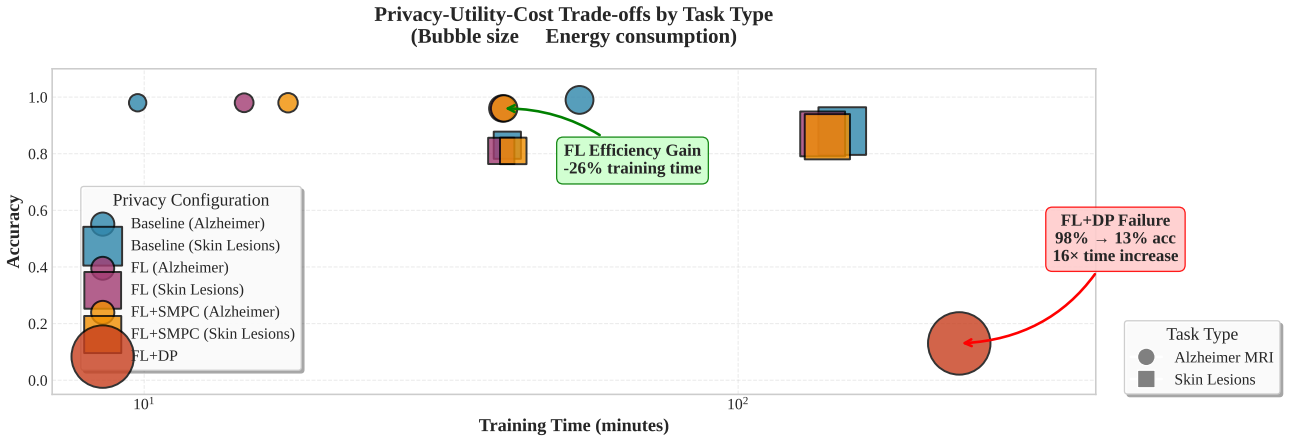

Figure 1: Privacy-Utility-Cost Trade-offs Across PPML Configurations and Task Types. Bubble size represents energy consumption; circles indicate Alzheimer MRI classification, squares indicate skin lesion classification. **Key findings:** FL and FL+SMPC achieve near-baseline performance with modest overhead, while FL+DP causes catastrophic failure (98% → 13% accuracy) with 16× training time increase. Notably, FL configurations can achieve efficiency gains (e.g., -26% training time for ViT on Alzheimer dataset), demonstrating that privacy techniques exhibit non-additive behaviors ranging from beneficial to destructive.

federated learning, introducing minimal additional overhead (18% over FL alone) while preserving model utility. This suggests that cryptographic privacy techniques can integrate smoothly when operating at compatible abstraction levels.

However, the FL+DP combination exhibits severe convergence issues. CNN accuracy drops from 98% to 13%—performance indistinguishable from random guessing—while training time increases 16× and energy consumption rises 20× compared to FL alone. This represents a complete breakdown of the learning process rather than gradual utility degradation.

4.2. Architectural Dependencies in Privacy System Perf

Our evaluation across CNN and transformer architectures reveals important architectural sensitivities. While ResNet18 models show consistent behavior across privacy techniques (excluding FL+DP failure), ViT models demonstrate different resource scaling patterns.

ViT models require substantially more computational resources (5.5× training time, 4.6× energy consumption) but maintain high utility under federated and SMPC configurations. Interestingly, federated training can improve efficiency for transformers, suggesting that distributed training may better suit their computational characteristics.

The complete performance metrics across both datasets and all configurations are provided in Appendix A.1, with detailed resource utilization analysis in Appendix A.2.

4.3. Resource Cost Analysis: Beyond Algorithmic Perf

The resource analysis reveals that privacy system design decisions have far-reaching implications beyond model accuracy. FL+DP's 24× computational overhead transforms privacy from a design consideration into a budget constraint that could render privacy initiatives economically unviable.

Current cost models for privacy-preserving ML appear inadequate. Organizations planning FL+DP deployments based on individual technique overhead would face resource requirements an order of magnitude beyond expectations. This miscalculation could significantly impact the feasibility of privacy-preserving deployments, particularly in resource-constrained environments.

Energy consumption patterns follow similar trends, with FL+DP requiring 20× more energy than FL alone. As computational sustainability becomes increasingly important, such energy requirements raise questions about the environmental impact of privacy-preserving ML systems.

4.4. Understanding Privacy System Failures

The FL+DP convergence failure pattern suggests fundamental compatibility issues rather than poor hyperparameter selection. The complete performance collapse—from medical-grade accuracy to random guessing—indicates that differential privacy noise, calibrated for centralized training, may become destructively amplified in federated settings.

We hypothesize that the combination of gradient signal attenuation from federated training (due to data heterogeneity and limited local updates) with DP noise injection creates a signal-to-noise ratio below the threshold required for meaningful learning. This explains why performance doesn't degrade gradually but collapses entirely.

Detailed failure analysis, including convergence patterns and resource breakdown, is provided in Appendix E.

4.5. Implications for Privacy System Design

Our findings suggest that privacy techniques cannot be treated as modular components that can be arbitrarily combined. FL+SMPC succeeds because both techniques operate at compatible abstraction levels—federated coordination and cryptographic aggregation complement rather than conflict with each other.

FL+DP (i.e., FL + centralized DP with server-side fixed noise) fails because it merges two incompatible assumptions: federated learning counts on averaging diverse, noisy gradients, while differential privacy requires tightly controlled, convergence-preserving noise. Under our settings, these assumptions clash, resulting in catastrophic performance loss.

These findings imply that privacy systems can't simply stack techniques. Instead, architectures must be co-designed to account for inter-method interactions and ensure fundamental compatibility..

5. Discussion and Future Directions

Our findings through PRIVACYBENCH raise important questions about privacy-preserving ML system design that extend beyond the specific techniques and configurations we evaluated.

Rethinking Privacy System Architecture The contrasting success of FL+SMPC versus FL+DP failure suggests that privacy technique compatibility may be predictable based on their operational abstractions. Techniques operating at similar levels (federated coordination + cryptographic aggregation) compose successfully, while those with conflicting assumptions (distributed learning + centralized noise calibration) exhibit fundamental incompatibilities. This points toward developing formal frameworks for assessing privacy technique compatibility before deployment.

Resource Accessibility and Privacy Equity The $16\times$ training time and $20\times$ energy increases observed with FL+DP raise critical questions about equitable access to privacy-preserving technologies. If effective privacy techniques require order-of-magnitude more computational resources, this could create a two-tiered system where privacy becomes accessible only to well-resourced organizations. This is critical for healthcare, small research centers, and resource-constrained regions where privacy is paramount.

Evaluation Methodology Gaps Current privacy research largely evaluates techniques in isolation, potentially missing the interaction effects we document. Our findings suggest that benchmarking practices should systematically test technique combinations and include resource cost analysis alongside traditional utility metrics. The development of adversarial benchmarking approaches that specifically probe failure modes could help identify problematic combinations before deployment.

Generalizability and Scope Our evaluation focuses on medical imaging with specific model architectures and a three-client federated setup. Important questions remain about how these findings generalize across domains, scales, and architectural choices. The architectural dependencies we observe—particularly the efficiency gains seen with federated ViT training—warrant investigation across different model families and problem domains.

Research Directions PRIVACYBENCH enables several promising research directions: developing formal compatibility frameworks for privacy technique combinations, designing privacy systems that gracefully degrade under resource constraints, and investigating architectural choices that improve privacy-utility-cost trade-offs. The systematic evaluation capabilities provided by our benchmark can support these investigations while ensuring reproducible and comparable results.

6. Conclusion

Through PRIVACYBENCH, we demonstrate that privacy-preserving machine learning techniques exhibit complex, non-additive behaviors when combined, challenging assumptions about modular privacy system design. Our systematic evaluation reveals that while FL and SMPC compose successfully with minimal overhead, FL+DP combinations can result in severe performance degradation and prohibitive resource requirements. These findings have important implications for privacy system deployment. Privacy techniques cannot be layered arbitrarily without careful consideration of their fundamental compatibility and resource implications. Expanding privacy regulations and computational sustainability concerns demand evaluation frameworks exposing incompatibilities and resource costs before deployment.

PRIVACYBENCH provides this foundation through comprehensive resource monitoring, systematic configuration management, and reproducible experimental infrastructure. Our benchmark enables the community to move beyond isolated technique evaluation toward holistic privacy system design that balances theoretical guarantees with practical deployment constraints. By treating privacy-preserving ML as a systems engineering challenge—explicitly modeling method interactions, resource limits, and operational needs—PRIVACYBENCH offers the tools and methodology for this shift. In future work, we will expand the framework to cover additional privacy techniques, data modalities, and practical use cases.

References

Adhikari, T. and Adhikari, T. Privacy-preserving data aggregation in federated learning for secure iot applications.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

El Ouadrhiri, A. and Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE access*, 10: 22359–22380, 2022.

Geyer, R. C., Klein, T., and Nabi, M. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Lai, F., Dai, Y., Singapuram, S., Liu, J., Zhu, X., Madhyastha, H., and Chowdhury, M. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International conference on machine learning*, pp. 11814–11827. PMLR, 2022.

Posthuma, M. The energy consumption and carbon footprint of https. B.S. thesis, University of Twente, 2025.

Rahaman, M., Arya, V., Orozco, S. M., and Pappachan, P. Secure multi-party computation (smpc) protocols and privacy. In *Innovations in Modern Cryptography*, pp. 190–214. IGI Global, 2024.

Rajput, S., Widmayer, T., Shang, Z., Kechagia, M., Sarro, F., and Sharma, T. Enhancing energy-awareness in deep learning through fine-grained energy measurement. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–34, 2024.

Wei, C., Zhao, M., Zhang, Z., Chen, M., Meng, W., Liu, B., Fan, Y., and Chen, W. Dpmlbench: Holistic evaluation of differentially private machine learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2621–2635, 2023.

Wei, W., Liu, L., Wu, Y., Su, G., and Iyengar, A. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pp. 797–807. IEEE, 2021.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

Zhou, I., Tofigh, F., Piccardi, M., Abolhasan, M., Franklin, D., and Lipman, J. Secure multi-party computation for machine learning: A survey. *IEEE Access*, 2024.

A. Complete Experimental Results

A.1. Comprehensive Performance Metrics

Table 2: Complete Results Across All Configurations and Datasets

Configuration	Dataset	Acc	F1	MCC	Prec	Rec	AUC
CNN Baseline	Alzheimer	0.98	0.98	0.97	0.98	0.98	1.00
CNN Baseline	Skin Lesions	0.83	0.83	0.75	0.83	0.83	0.96
ViT Baseline	Alzheimer	0.99	0.99	0.98	0.99	0.98	1.00
ViT Baseline	Skin Lesions	0.88	0.88	0.82	0.88	0.88	0.98
FL (CNN)	Alzheimer	0.98	0.98	0.97	0.98	0.98	1.00
FL (CNN)	Skin Lesions	0.81	0.81	0.73	0.81	0.81	0.96
FL (ViT)	Alzheimer	0.96	0.96	0.94	0.97	0.96	1.00
FL (ViT)	Skin Lesions	0.87	0.87	0.82	0.87	0.87	0.98
FL+SMPC (CNN)	Alzheimer	0.98	0.98	0.97	0.98	0.98	1.00
FL+SMPC (CNN)	Skin Lesions	0.81	0.81	0.73	0.81	0.81	0.96
FL+SMPC (ViT)	Alzheimer	0.96	0.96	0.93	0.96	0.96	1.00
FL+DP (CNN)	Alzheimer	0.13	0.03	0.00	0.02	0.13	0.50
FL+DP (CNN)	Skin Lesions	–*	–*	–*	–*	–*	–*

Results not available due to computational constraints

A.2. Computational Resource Analysis

Table 3: Complete Resource Utilization Metrics

Configuration	Dataset	Time (sec)	CPU (GB)	GPU (GB)	Energy (kWh)	CO_2 (kg)
CNN Baseline	Alzheimer	585	0.15	0.64	0.026	0.011
CNN Baseline	Skin Lesions	2,452	0.15	0.64	0.112	0.048
ViT Baseline	Alzheimer	3,244	1.98	4.08	0.119	0.051
ViT Baseline	Skin Lesions	8,983	1.00	3.76	0.413	0.178
FL (CNN)	Alzheimer	884	0.0005	0.60	0.036	0.016
FL (CNN)	Skin Lesions	2,395	0.0011	0.60	0.102	0.044
FL (ViT)	Alzheimer	2,405	0.0010	3.72	0.104	0.045
FL (ViT)	Skin Lesions	8,325	0.0012	3.72	0.362	0.156
FL+SMPC (CNN)	Alzheimer	1,048	0.0005	0.97	0.041	0.018
FL+SMPC (CNN)	Skin Lesions	2,509	0.0011	0.97	0.105	0.045
FL+SMPC (ViT)	Alzheimer	2,422	0.0009	3.72	0.104	0.045
FL+SMPC (ViT)	Skin Lesions	8,478	0.0011	3.72	0.369	0.159
FL+DP (CNN)	Alzheimer	14,137	0.0007	0.97	0.734	0.069

B. Detailed Experimental Configuration

B.1. Privacy Technique Parameters

Table 4: Differential Privacy Configuration

Parameter	Value	Description
Noise Multiplier	1.0	Gaussian noise multiplier for gradient perturbation
Max Grad Norm	1.0	L2 norm clipping threshold
Delta (δ)	1×10^{-5}	Privacy parameter for (ϵ,δ)-differential privacy
Epsilon (ϵ)	0.5, 1.0	Privacy budget values evaluated
Secure RNG	True	Cryptographically secure random number generation

Table 5: Secure Multi-Party Computation Configuration

Parameter	Value	Description
Clipping Range	8	Value range for secure quantization
Max Weight	2000 (CNN), 6000 (ViT)	Maximum weight value for quantization
Modulus	4,294,967,296	Field size for secret sharing arithmetic
Shares	3	Number of secret shares per value
Quantization Range	4,194,304	Precision range for fixed-point arithmetic
Reconstruction Threshold	2	Minimum shares needed for reconstruction

Table 6: Federated Learning Configuration

Parameter	CNN	ViT	Description
Clients	3	3	Number of participating clients
FL Rounds	5	5	Total federated training rounds
Local Epochs	15	10	Training epochs per client per round
Batch Size	32/64*	32	Local training batch size
Learning Rate	2.1×10^{-4}	5×10^{-5}	Client-side learning rate
Early Stopping	10 rounds	10 rounds	Patience for convergence
Data Partition	Non-IID	Non-IID	Client data distribution

Batch size 64 used for FL+DP configurations

Table 7: ResNet18 Configuration

Component	Specification
Parameters	11.7M
Input Size	$224 \times 224 \times 3$
Architecture	ResNet18 with pretrained ImageNet weights
Optimizer	Adam
Learning Rate	2.1×10^{-4}
Weight Decay	1×10^{-4}
Early Stopping Patience	7 steps
Max Epochs	50

B.2. Dataset Specifications

Preprocessing Pipeline:

- **Alzheimer MRI:** Resize to 224×224, normalize with ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$)
- **Skin Lesions:** Resize to 224×224, random horizontal flip ($p = 0.5$), random rotation (± 10), normalize with ImageNet statistics

C. System Infrastructure Details

C.1. Hardware Specifications

Software Environment

- **OS:** Debian GNU/Linux 11 (bullseye)
- **Python:** 3.10.x or later
- **Deep Learning:** PyTorch 2.x (CUDA 12.4, cuDNN 8.x), torchvision
- **Utilities:** Transformers, NumPy, SciPy, Pillow

Table 8: Vision Transformer Configuration

Component	Specification
Parameters	86.6M
Input Size	$224 \times 224 \times 3$
Architecture	ViT-Base/16 with pretrained ImageNet weights
Patch Size	16×16
Optimizer	Adam
Learning Rate	5×10^{-5}
Weight Decay	1×10^{-4}
Early Stopping Patience	7 steps
Max Epochs	50

Table 9: Dataset Statistics

Dataset	Classes	Train	Test	Size	Preprocessing
Alzheimer MRI (baseline)	4	60%	40%	224×224	Resize, normalize
Skin Lesions-ISIC (baseline)	8	60%	40%	224×224	Resize, augment, normalize
Alzheimer MRI (experiments)	4	92%	8%	224×224	Resize, normalize
Skin Lesions-ISIC (experiments)	8	92%	8%	224×224	Resize, augment, normalize

Table 10: Experimental Hardware Configuration

Component	Specification
GPU	2× NVIDIA Tesla T4 (15 GB each)
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz, 32 vCPUs (n1-standard-64)
RAM	120 GB DDR4
Storage	50 GB Balanced Persistent Disk (SCSI interface)
OS	Debian GNU/Linux 11 (bullseye)

- **Configuration:** YAML-based experiment management

Table 11: Software Dependencies

Package	Version
Python	3.12
PyTorch	2.6.0
Flower	1.15.2
Opacus	1.5.3
CodeCarbon	2.8.3
NumPy	2.0
Scikit-learn	1.6.1
CUDA	12.4

C.2. YAML Configuration Examples

Baseline CNN Configuration:

```
experiment:
  name: "cnn_baseline_alzheimer"
  dataset: "alzheimer"

model:
  architecture: "resnet18"
  pretrained: true
```

```
training:
  epochs: 50
  batch_size: 32
  learning_rate: 2.1e-4
  optimizer: "adam"
  early_stopping: 5

privacy:
  federated: false
  differential_privacy: false
  secure_mpc: false

logging:
  track_energy: true
  track_memory: true
  save_checkpoints: true
```

FL+DP Configuration:

```
experiment:
  name: "fl_dp_cnn_alzheimer"
  dataset: "alzheimer"

model:
  architecture: "resnet18"
  pretrained: true

federated:
  enabled: true
  clients: 3
  rounds: 20
  local_epochs: 50
  aggregation: "fedavg"

differential_privacy:
  enabled: true
  noise_multiplier: 1.0
  max_grad_norm: 1.0
  delta: 1e-5
  epsilon: 0.5

training:
  batch_size: 64
  learning_rate: 2.1e-4
  optimizer: "adam"
  early_stopping: 20

logging:
  track_energy: true
  track_memory: true
  save_model_updates: true
```

D. Implementation Details

D.1. Energy Monitoring Implementation

```
from codecarbon import EmissionsTracker

tracker = EmissionsTracker(
    project_name="privacybench",
    measure_power_secs=30,
    tracking_mode="machine",
    country_iso_code="\textcolor{red}{[YOUR\_COUNTRY\_CODE]}",
```

```
    region="\textcolor{red}{[YOUR\_REGION]}",
    output_file="emissions.csv"
)
```

D.2. Memory Tracking Implementation

```
import psutil
import torch

def track_memory():
    # CPU memory
    cpu_memory = psutil.virtual_memory()
    cpu_used_gb = cpu_memory.used / (1024**3)

    # GPU memory
    if torch.cuda.is_available():
        gpu_memory = torch.cuda.memory_stats()
        gpu_used_gb = gpu_memory['reserved_bytes.all.current'] / (1024**3)

    return cpu_used_gb, gpu_used_gb
```

E. Failure Analysis Details

E.1. FL+DP Convergence Analysis

The FL+DP configuration exhibits several characteristic failure patterns that suggest fundamental compatibility issues rather than implementation bugs:

1. **Rapid Performance Degradation:** Accuracy drops below 20% within the first 5 federated rounds, indicating early training collapse
2. **Gradient Explosion:** Local gradient norms consistently exceed the clipping threshold, suggesting instability in the optimization process
3. **High Variance:** Model updates show extreme variance across clients, preventing meaningful aggregation
4. **Poor Aggregation:** FedAvg algorithm struggles to find consensus among highly divergent client updates

E.2. Hypothesized Causes

Based on the observed failure patterns, we hypothesize several contributing factors:

- **Noise Amplification:** DP noise calibrated for centralized training becomes destructively amplified in federated settings where gradient signals are already attenuated
- **Signal Attenuation:** The combination of limited local epochs, non-IID data distribution, and DP noise reduces meaningful gradient information below the threshold required for learning
- **Interaction Effects:** The fundamental assumptions of federated learning (that noisy, heterogeneous updates can be meaningfully averaged) conflict with differential privacy assumptions (that controlled noise injection preserves learning)

E.3. Resource Overhead Analysis

The FL+DP configuration demonstrates significant computational overhead compared to other privacy techniques. Training time increases from 884 seconds (FL only) to 14,137 seconds (FL+DP), representing a 16$\times$ multiplicative factor. Energy consumption similarly rises from 0.036 kWh to 0.734 kWh, a 20$\times$ increase with important implications for deployment feasibility and environmental impact.

F. Reproducibility Checklist

F.1. Environment Setup

☐ Install Python 3.12 and above then create virtual environment

☐ Install dependencies from requirements.txt with exact versions (Table 11)

☐ Verify CUDA compatibility and GPU drivers

☐ Download and prepare datasets with preprocessing pipeline

☐ Configure CodeCarbon for energy tracking with appropriate region settings

☐ Set random seeds for reproducibility (seed = 42 used throughout)

F.2. Experiment Execution

☐ Run baseline experiments first to establish performance benchmarks

☐ Execute FL experiments with comprehensive resource monitoring enabled

☐ Validate SMPC implementation with known test vectors for correctness

☐ Monitor FL+DP experiments for early termination due to convergence failure

☐ Archive results, configurations, and logs for analysis

☐ Verify energy measurements with external monitoring tools if available

F.3. Result Validation

☐ Verify metric calculations against reference implementations

☐ Cross-validate energy measurements with alternative tracking methods

☐ Confirm statistical significance of performance differences across runs

☐ Document any deviations from expected results or implementation issues

☐ Compare resource utilization patterns with reported baselines