Noga Gercsak

Charlotte, NC

Privacy-Aware Collaborative DNN Inference: Adaptive Differential Privacy for Edge Computing

INTRODUCTION

The number of Internet of Things (IoT) devices is projected to grow to **21.1 billion globally**, making it one of the most integral components of our **data**. IoT devices are commonly used in object recognition, human activity recognition, health monitoring, and environmental sensing.

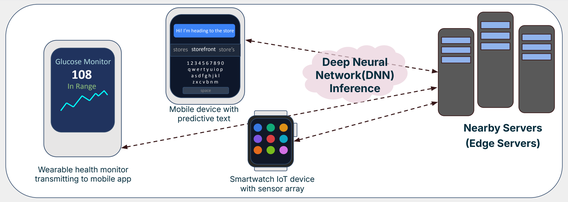

Figure created by Noga Gercsak using Microsoft PowerPoint, 2026.

Fig. 1: Model partitioning for DNN inference at the edge.

Implications of Cybersecurity in IoT Devices

- **Healthcare:** Wearable health monitors transmit intermediate data to edge servers in real time
  - Interception exposes patient vitals, diagnoses, and behavioral patterns
- **February 2025:** A single unprotected IoT database (Mars Hydro) breach exposed over 2.7 billion records
- **Autonomous vehicles:** Self-driving systems rely on continuous DNN inference at the edge
  - Privacy breach could compromise navigation data or cause accidents
- Even abstract intermediate representations (like the feature maps shown above) retain enough visual/semantic information to **reconstruct** sensitive input **data** through model inversion **attacks**

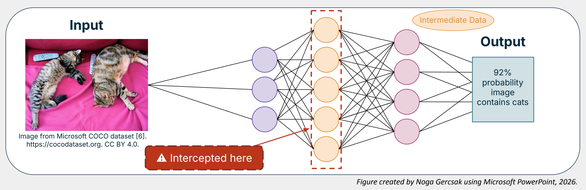

Figure created by Noga Gercsak using Microsoft PowerPoint, 2026.

Fig. 2: Intermediate Data Interception in Split Neural Network Inference

OBJECTIVES

RESEARCH QUESTION:

How can adaptive privacy budget allocation improve privacy-utility trade-offs in collaborative DNN inference compared to static differential privacy?

Cost = α × Energy + β × Time + γ × Privacy Loss

α (Energy Weight)

- Minimizing power consumption
- Higher α favors earlier cut points with sooner offloads
- Unit: millijoules (mJ)

β (Time Weight)

- Prioritizes minimizing total end-to-end inference latency
- Higher β favors configurations that reduce client + transmission + server processing time
- Unit: seconds (s)

γ (Privacy Weight)

- Prioritizes minimizing privacy loss (1/ε)
- Higher γ → smaller ε → stronger privacy, less accuracy
- Unit: dimensionless (ε is a unitless privacy budget parameter)

By tuning α, β, and γ, the framework adapts to different deployment scenarios: a hospital prioritizes γ, a real-time autonomous vehicle prioritizes β, a battery-powered sensor prioritizes α.

METHODOLOGY

The framework was tested on real IoT hardware across 20 configurations (varying both where the model is split and how much privacy noise is added) to measure the accuracy, energy, and latency trade-offs of adaptive vs. fixed differential privacy.

Differential Privacy (DP) Implementation

Data to Transmit
Partial image analysis from edge device

→ **Identify Layer Type**
Convolutional, pooling, or deep

→ **Calculate Sensitivity**
How much can one input shift the output?

→ **Set Noise Level**
More privacy = more noise

→ **Add Noise to Data**
$x + N(0, \sigma^2)$

→ **Send Protected Data**
Server cannot reconstruct original

Figure created by Noga Gercsak using Microsoft PowerPoint, 2026.

ε = 8 (weak privacy) ↔ ε = 0.5 (strong privacy)

Fig. 3: Noise Addition Pipeline

Before sending data to the server, the edge device adds carefully calculated random noise. More noise means stronger privacy but slightly lower accuracy. The key innovation is that the noise amount is calculated individually for each layer type, rather than using one fixed value for everything.

System Architecture

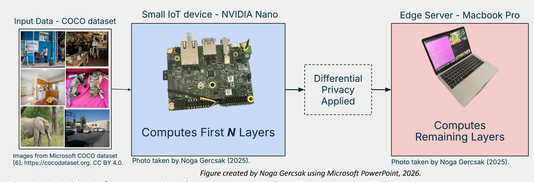

Figure created by Noga Gercsak using Microsoft PowerPoint, 2026.

Fig. 4: Split Inference Architecture: Edge Device to Edge Server

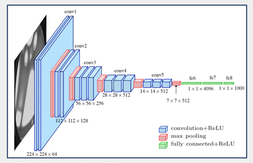

Fig. 5: VGG16 Architecture

VGG16 was chosen because its 16 layers can be split at any point, making it ideal for testing where exactly to "hand off" computation. It's also widely used in real applications like medical image analysis and smart security systems, which are the exact situations where protecting user privacy matters most.

Experimental Protocol

Dataset:	1,000 COCO 2017 images across 10 categories, preprocessed to 224×224; chosen to represent real-world visual diversity without domain bias.
Configuration space:	4 partition points (layers 3, 5, 8, 12) × 5 epsilon values (ε = 0.5–8.0) = 20 experimental configurations, spanning the full privacy-performance spectrum.
Measurement rigor:	Each configuration ran 50 inference trials (with 2 warm-up runs for cache stabilization) at 5Hz energy monitoring, yielding statistically robust results rather than single-run snapshots.

RESULTS

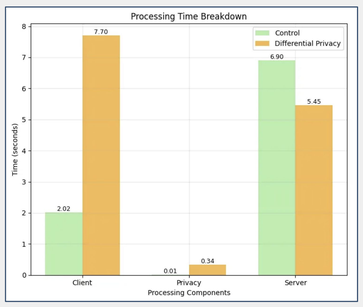

Fig. 6: Processing time distribution comparing baseline and differential privacy implementations across client-side, privacy-specific, and server-side operations.

Graphic created by Noga Gercsak using Python (VS Code), matplotlib, and numpy, 2025.

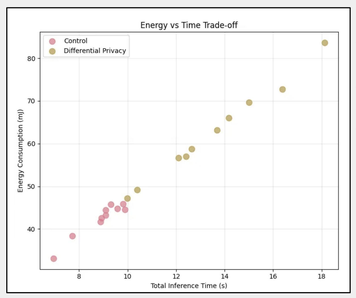

Fig. 7: Energy-latency trade-off analysis showing operational measurements for systems with and without differential privacy. Three distinct clusters are observed: 8–12 seconds with 40–55 mJ, 12–15 seconds with 55–70 mJ, and 15–18 seconds with 70–85 m

Graphic created by Noga Gercsak using Python (VS Code), matplotlib, and numpy, 2025.

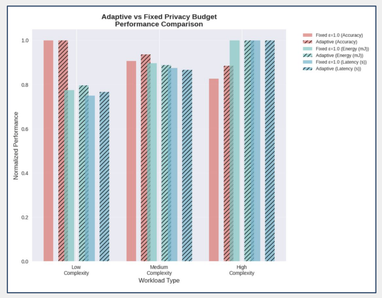

Fig.8: Adaptive versus fixed privacy budget performance comparison across low, medium, and high complexity workloads. Metrics shown include accuracy (normalized), energy consumption (mJ), and latency (seconds) for fixed ε = 1.0 and adaptive privacy budget allocation.

Graphic created by Noga Gercsak using Python (VS Code), matplotlib, and numpy, 2025.

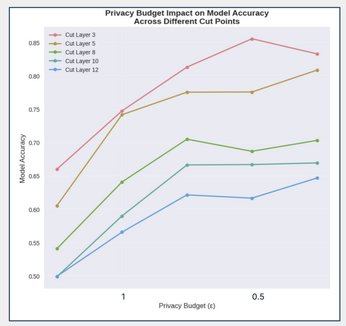

Fig. 9: Privacy budget impact on model accuracy across different cut points. Lower ε values indicate stronger privacy guarantees but reduced accuracy. Cut points at layers 3, 5, 8, 10, and 12 show varying sensitivity to privacy budget changes.

Graphic created by Noga Gercsak using Python (VS Code), matplotlib, and numpy, 2025.

Graphic created by Noga Gercsak using Python (VS Code), matplotlib, and numpy, 2025.

DATA ANALYSIS

- **Optimal split point:** Partitioning at Layer 3 preserved the highest accuracy across all privacy levels, outperforming all deeper cut points
- **Minimal accuracy cost:** Adding full privacy protection reduced accuracy by only 2.2%, from 88% down to 85.8%
- **Adaptive privacy outperforms fixed:** Dynamically adjusting the privacy budget improved accuracy by up to 3.5% compared to a static setting
- **Deep partitioning is high risk:** Splitting at Layer 12 with moderate privacy reduced accuracy to 50%, equivalent to random guessing
- **Privacy protection has measurable energy costs:** Configurations grouped into three distinct ranges, where stronger privacy consistently required more energy and longer inference time

Tbl. 1: Key Summary of Result

Measure Type	Result	Observation
Accuracy (DP vs. baseline)	2.2% reduction at ε = 1.0	Privacy protection achievable with minimal utility loss
Client Processing Time	2.02s ⟶ 7.70s (+281%)	Overhead concentrated on edge device
Best Partition Point	Layer 3, all ε values	Early partitioning most privacy-resilient

Figure created by Noga Gercsak using Microsoft PowerPoint, 2026.

CONCLUSIONS

No DP-Baseline
88% Accuracy
0 Privacy protection
~44 mJ

→

With Adaptive DP
85.8% accuracy
Full DP guarantee
47–51 mJ (Layer 3)

- **Comparison to existing approaches:** Results align with Apple's real-world DP deployments (ε = 2–8), validating the framework against industry standards
- **Successfully demonstrated** that adaptive differential privacy can be integrated into collaborative edge inference with only 2.2% accuracy reduction on real IoT hardware
- **Early partition resilience:** Layer 3 maintained 83–85% accuracy across all privacy levels, providing a concrete deployment recommendation
- **Adaptive allocation validated:** Dynamic sensitivity calculation outperformed fixed budgets by up to 3.5% on high-complexity workloads, confirming the core hypothesis

FUTURE WORK

- Extend to MobileNet, EfficientNet, and transformer architectures
- Validate against live adversarial inference attacks
- Test under variable real-world network conditions

Improvements and Limitations

- Experiments conducted on specific hardware (Jetson Nano + MacBook Pro); may not generalize to all edge devices
- Privacy guarantees are theoretical; specific attack vectors were not empirically tested

KEY REFERENCES

[1] Apple Inc. (2017). Differential Privacy Technical Overview. apple.com/privacy
[2] Chaopeng, G., Zhengqing, L., & Jie, S. (2023). A privacy protection approach in edge-computing based on maximized DNN partition strategy with energy saving. Journal of Cloud Computing, 12. doi:10.1186/s13677-023-00404-y
[3] Cheng, Z., et al. (2025). Privacy-aware joint DNN model deployment and partitioning optimization for collaborative edge inference services. arXiv:2502.16091
[4] Dong, F., et al. (2023). Multi-exit DNN inference acceleration based on multi-dimensional optimization for edge intelligence. IEEE Transactions on Mobile Computing, 22. doi:10.1109/tmc.2022.3172402
[5] Errounda, F. Z., & Liu, Y. (2023). Adaptive differential privacy in vertical federated learning for mobility forecasting. Future Generations Computer Systems, 149. doi:10.1016/j.future.2023.07.033
[6] Lin, T. Y., et al. (2014). Microsoft COCO: Common objects in context. ECCV 2014. arXiv:1405.0312
[7] Rohini G. (2021). Everything you need to know about VGG16 [Image]. Medium. https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918