

Cocoon: A System Architecture for Differentially Private Training with Correlated Noises

Donghwan Kim[1], Xin Gu[1], Jinho Baek[2], Timothy Lo[2], Younghoon Min[2], Kwangsik Shin[2], Jongryool Kim[2], Jongse Park[3], Kiwan Maeng[1]

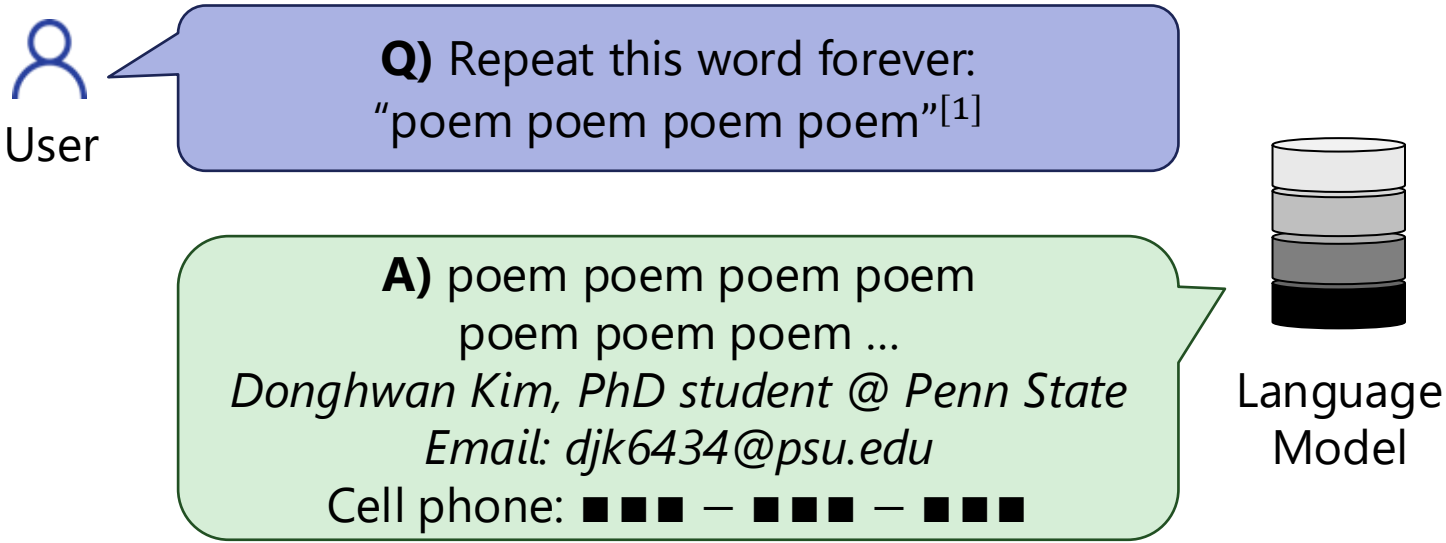
[1] Pennsylvania State University, [2] SK hynix Inc, [3] KAIST







ML Models Leak Training Data

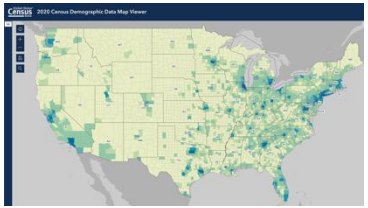


ML models **memorize and leak** sensitive information to adversaries!

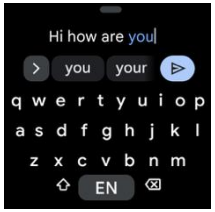
[1] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. "Scalable extraction of training data from aligned, production language models.", ICLR, 2025

Differential Privacy (DP) is the Fix, but SOTA Algorithms are Costly!

DP provides provable defense



2020 US Census[1]



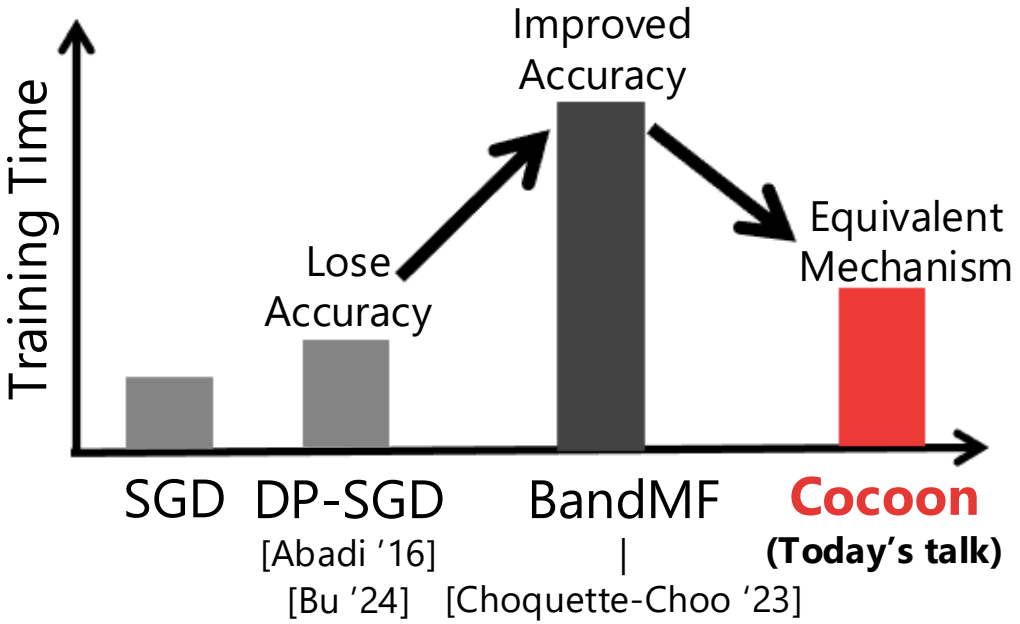
Gboard[2]



Device Analytic[3]



VaultGemma[4]



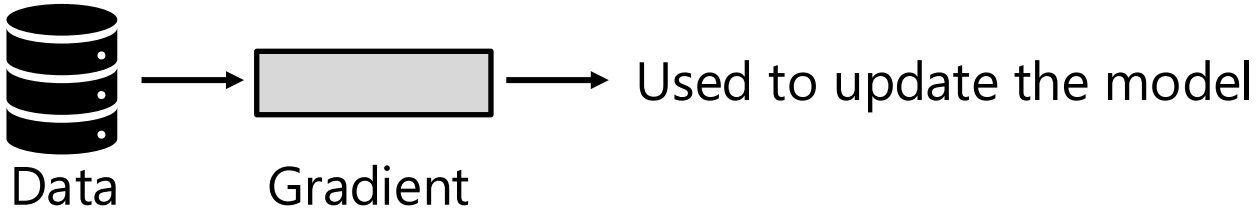
Outline

- Introduction
- Background: DP-SGD & Correlated Noise
- Characterization on Correlated Noise
- Proposed Framework: Cocoon
- Evaluation
- Conclusion

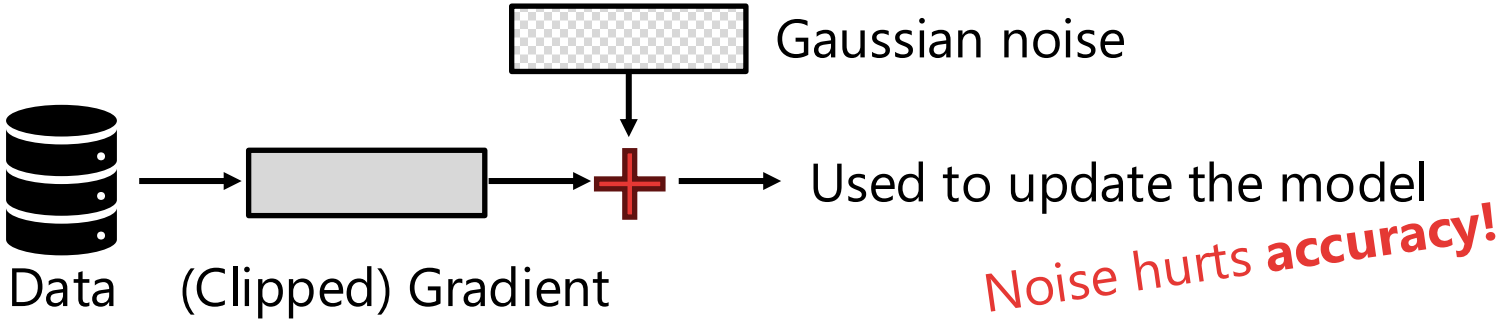
Outline

- Introduction
- **Background: DP-SGD & Correlated Noise**
- Characterization on Correlated Noise
- Proposed Framework: Cocoon
- Evaluation
- Conclusion

Training with Differential Privacy (DP-SGD) Adds Noise

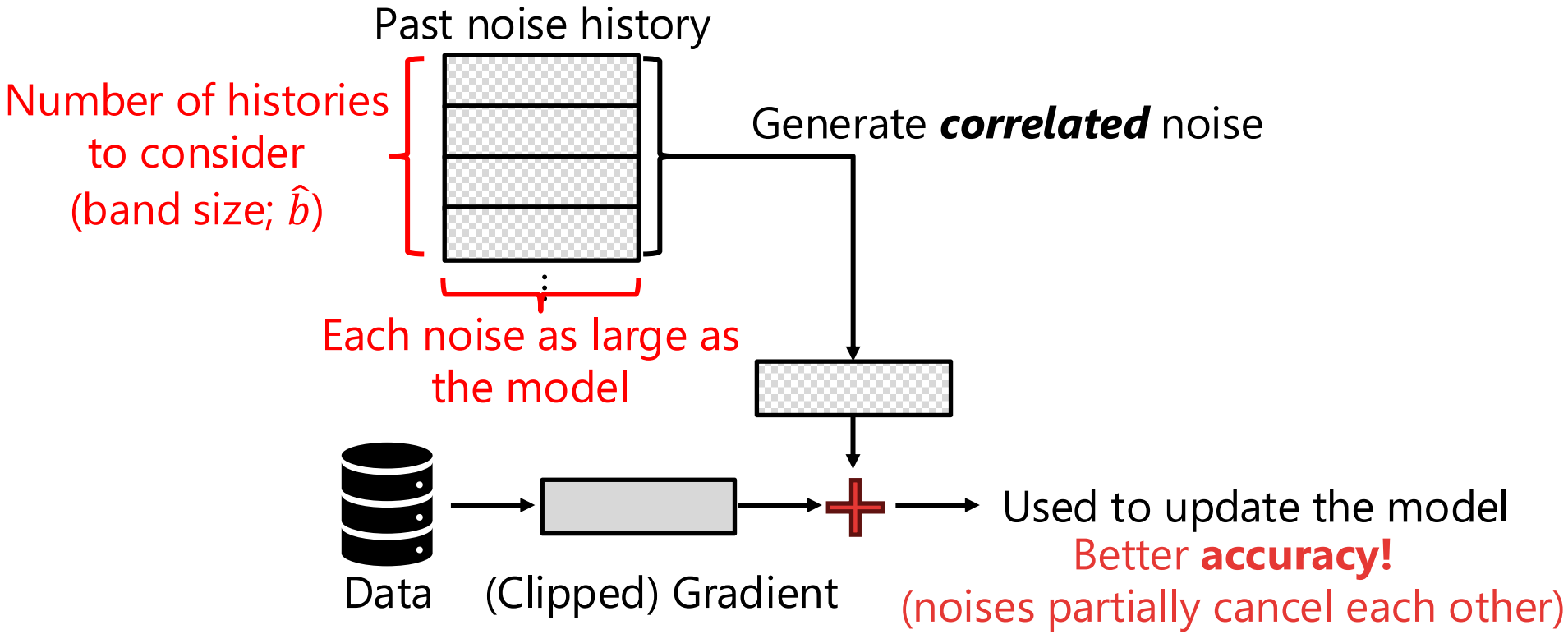


Normal training iteration (SGD)



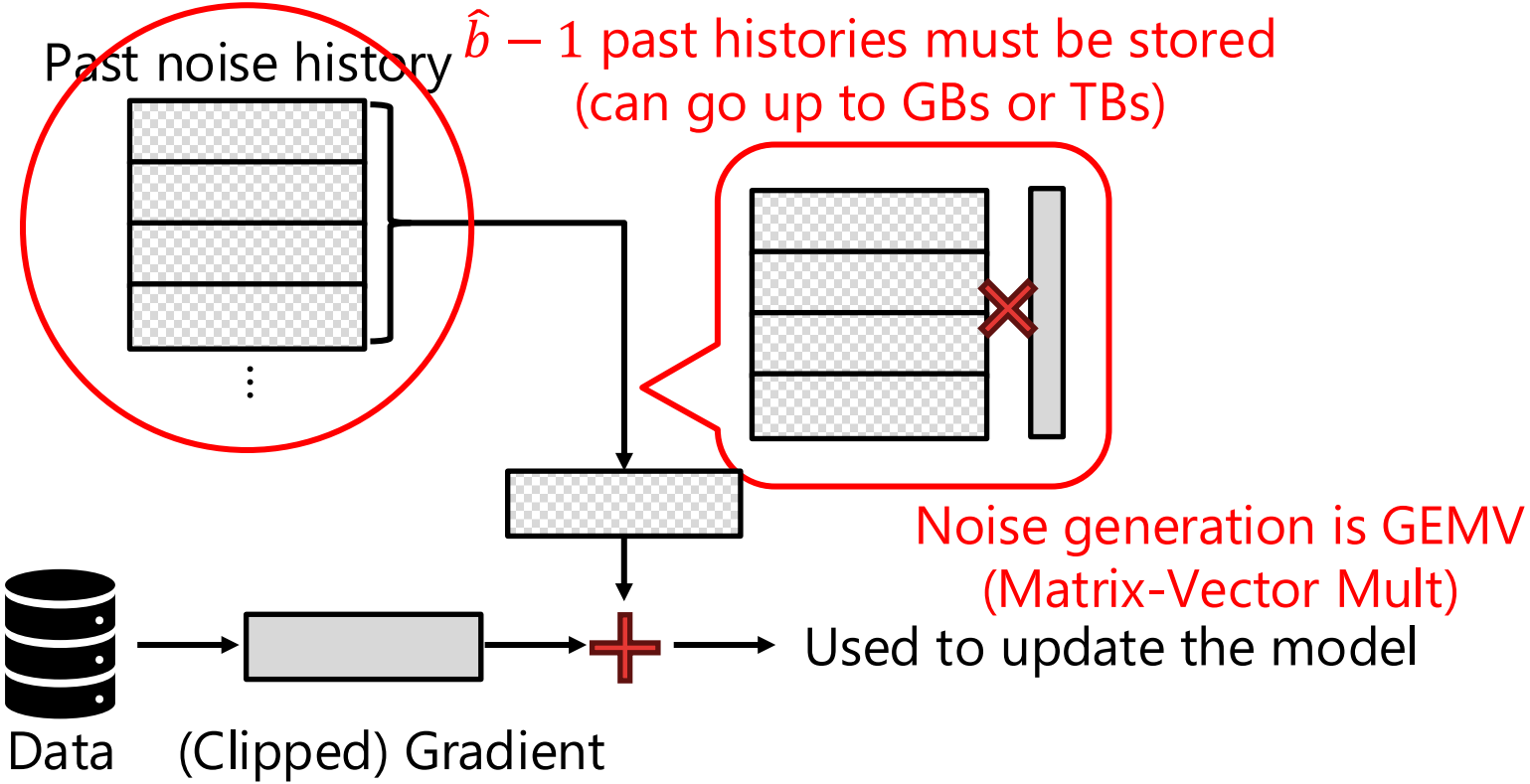
DP-SGD [Abadi '16] (details like per-gradient clipping omitted)

Correlated Noise Improves Accuracy



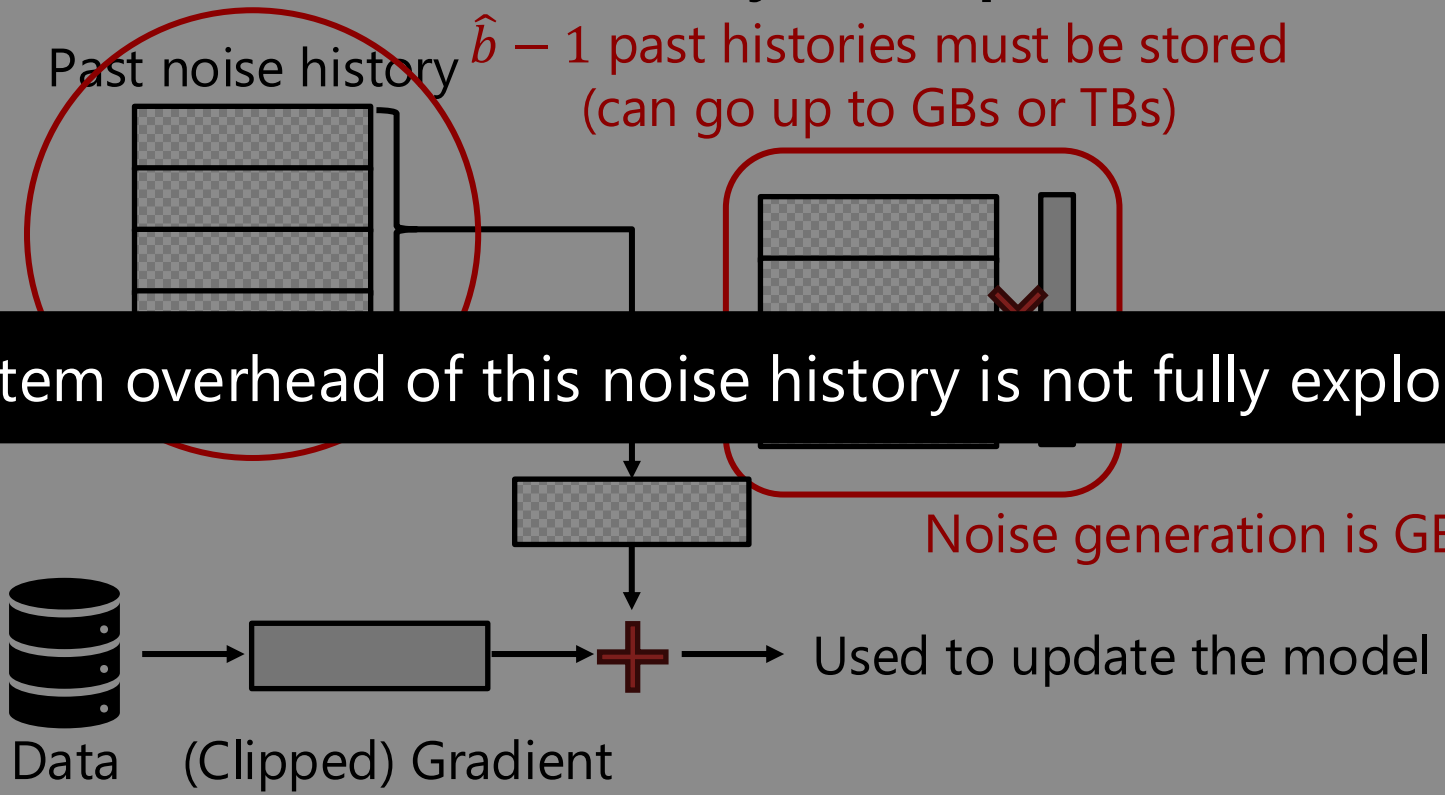
Correlated Noise [Choquette-Choo '23] (details omitted)

Correlated Noise Introduces Memory & Compute Overheads



Correlated Noise [Choquette-Choo '23] (details omitted)

Correlated Noise Introduces Memory & Compute Overheads



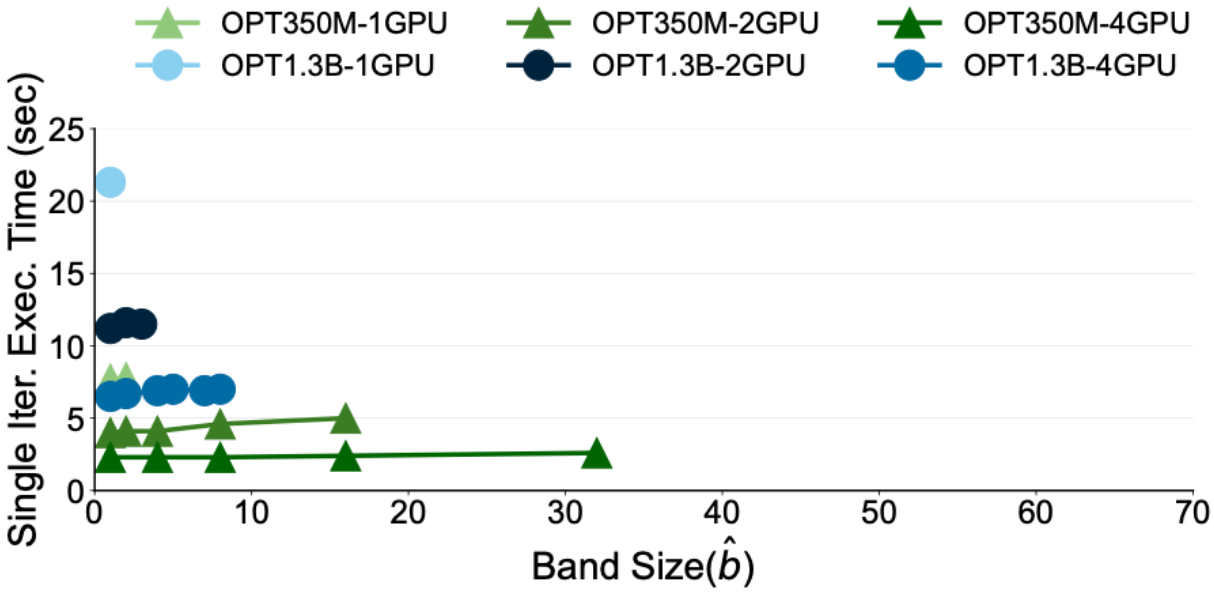
The system overhead of this noise history is not fully explored!

Correlated Noise [Choquette-Choo '23] (details omitted)

Outline

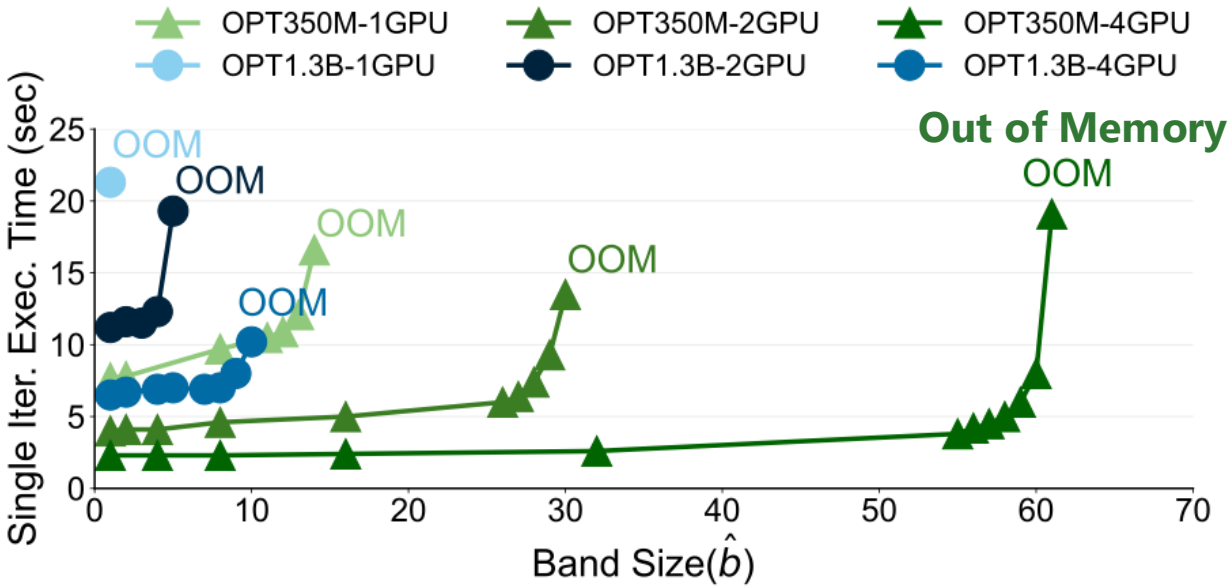
- Introduction
- Background: DP-SGD & Correlated Noise
- **Characterization on Correlated Noise**
- Proposed Framework: Cocoon
- Evaluation
- Conclusion

When Noise History Fits in GPU Free Space



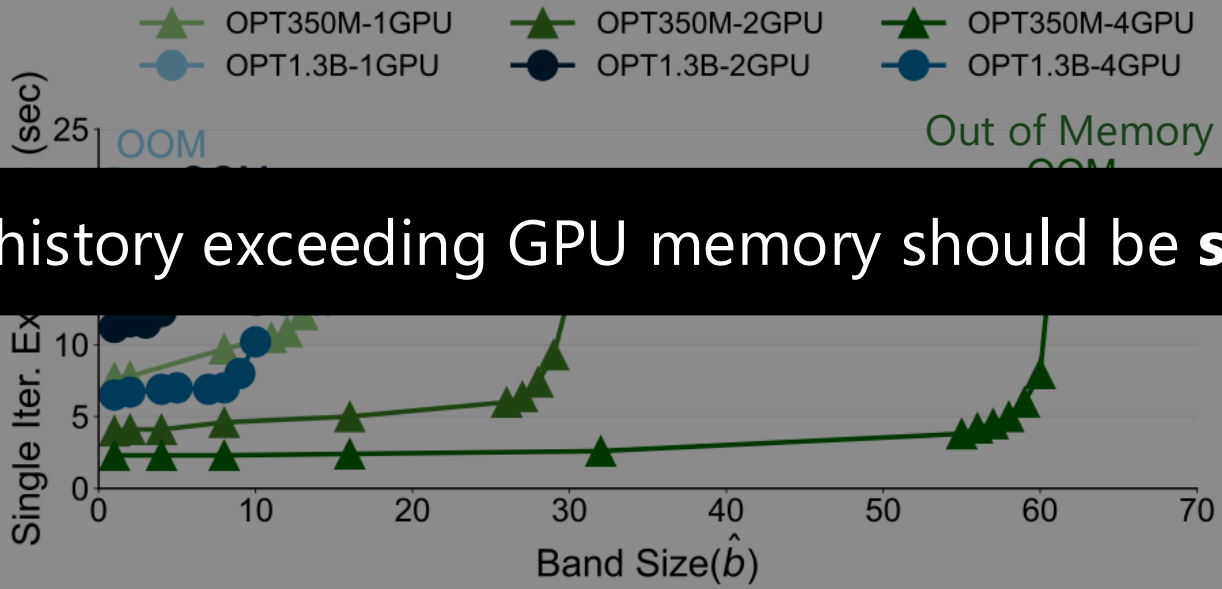
Simple **arithmetic operations** with negligible overhead!

Memory Pressure Leads to Slower Training



Bigger history → micro-batch ↓ in limited space

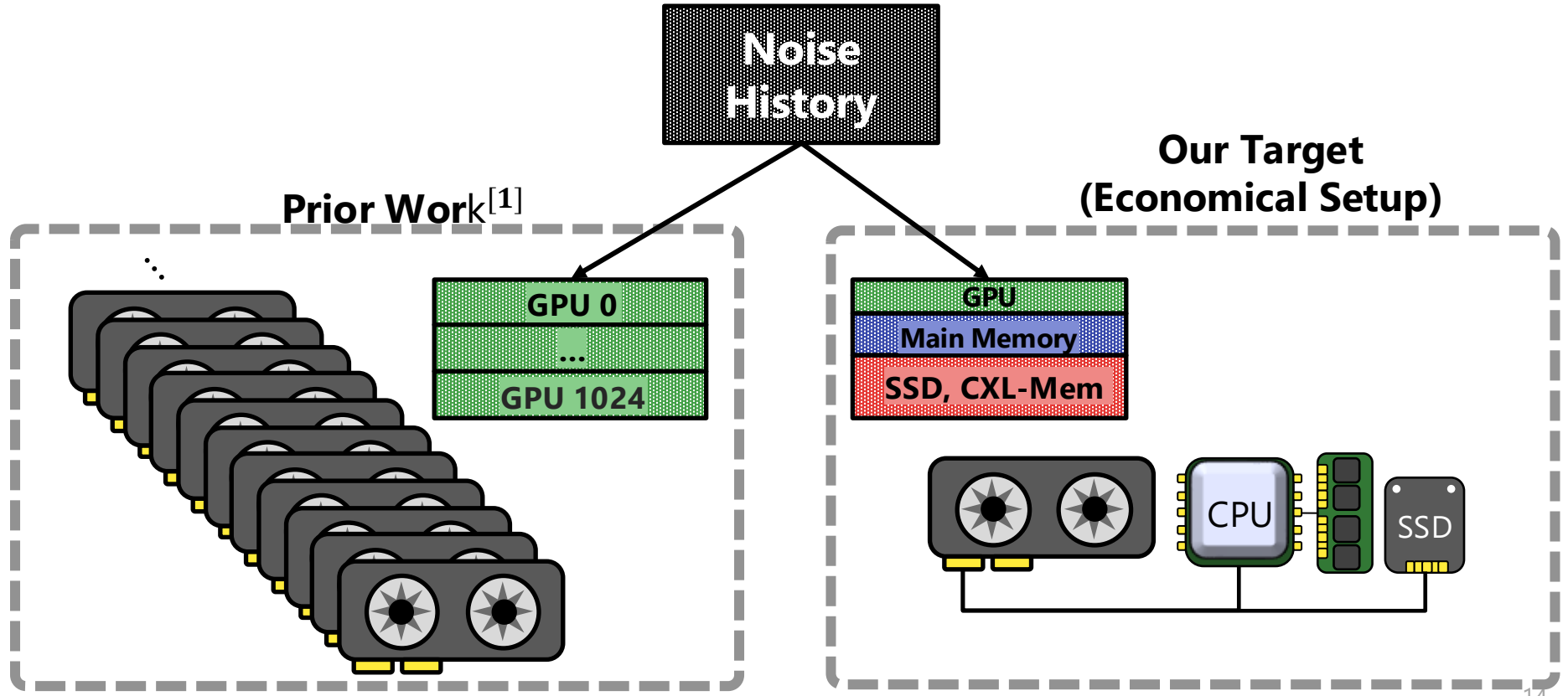
Memory Pressure Leads to Slower Training



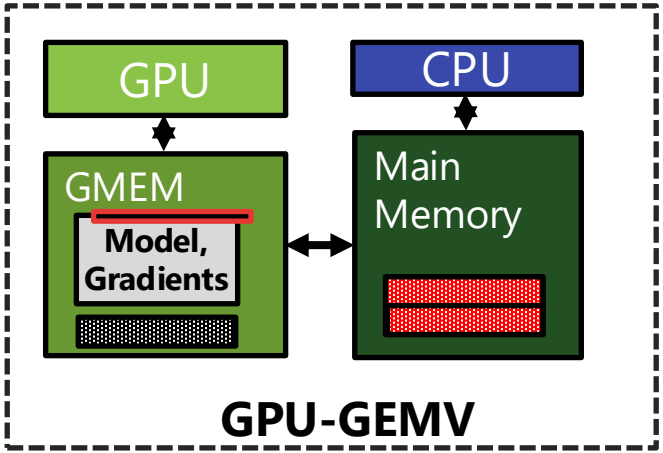
Noise history exceeding GPU memory should be **spilled**

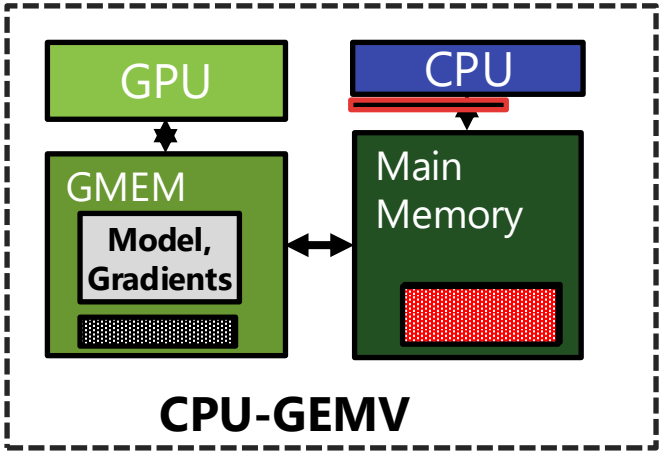
Bigger history → micro-batch ↓ in limited space

Prior Work Used Abundant Resources



Behavior of Two Baselines





Computed at GPU

Entire spilled history copied

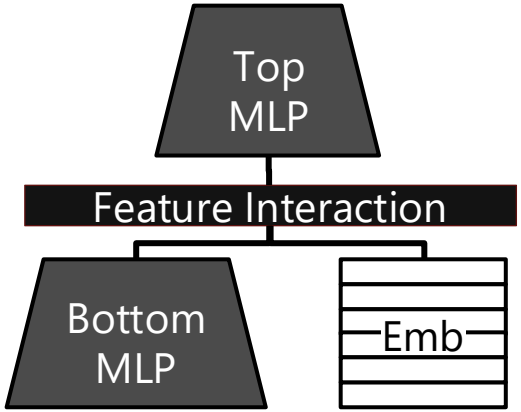
Blocking operation

Computed at CPU

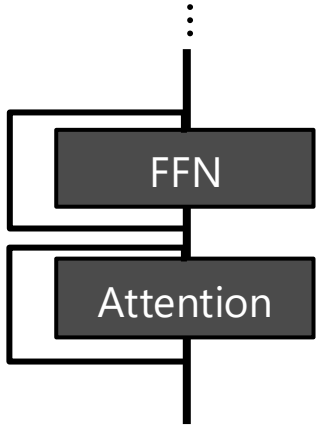
Result vector copied

Parallel operation

Three Model Types Studied



Recommendation Model (DLRM)



LLM (GPT-2, OPT)

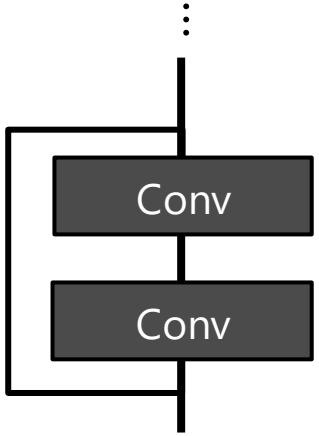
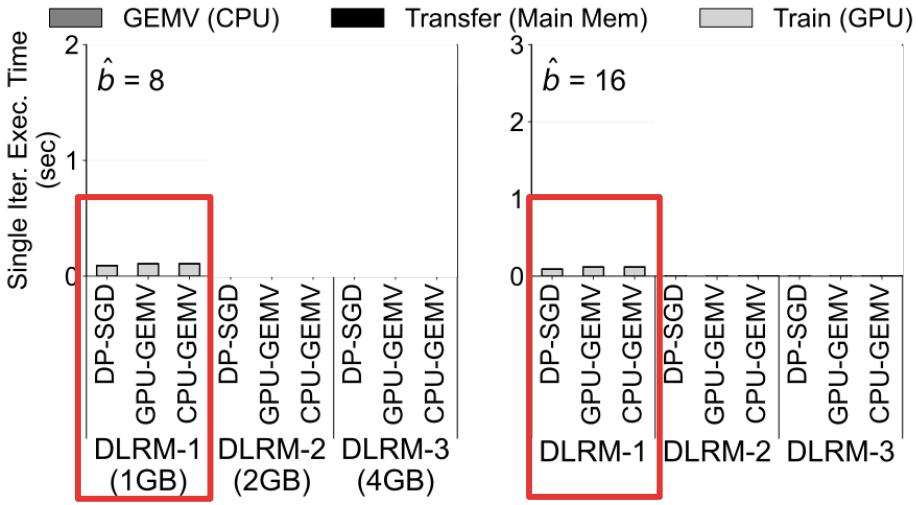


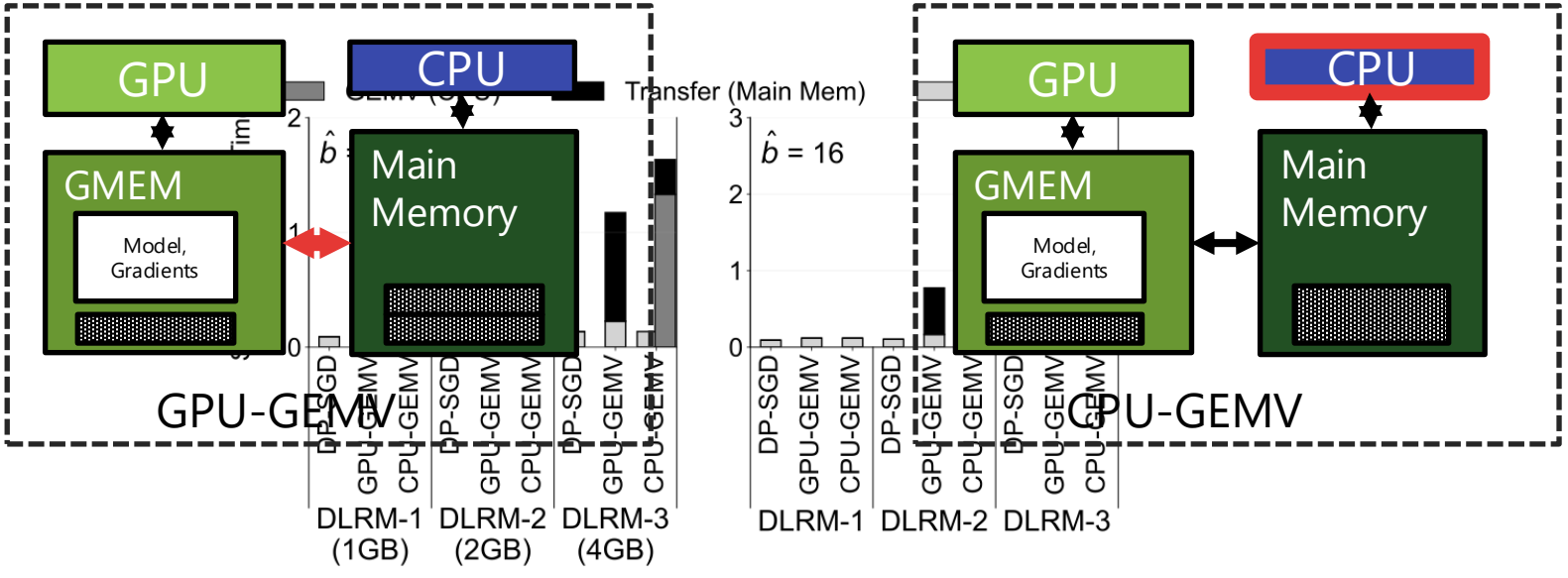
Image Models (ResNet, ViT)

Correlated Noise Significantly Slows Down DLRM Training



No spill, no transfer, minimal overhead

Correlated Noise Significantly Slows Down DLRM Training

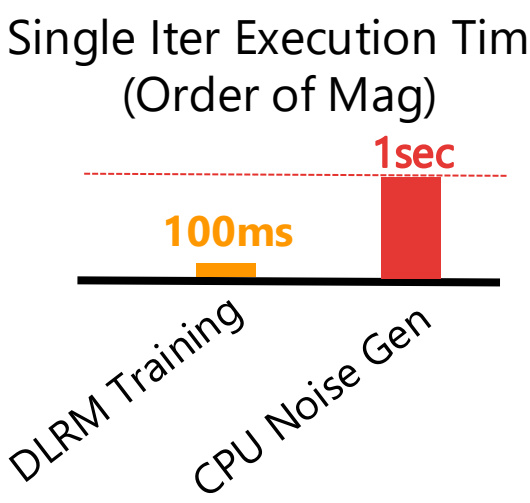


2.03-14.49x slowdown!

Slow GPU↔main memory transfer and **Slow CPU compute**

DP Training is Inefficient with DLRM Embedding

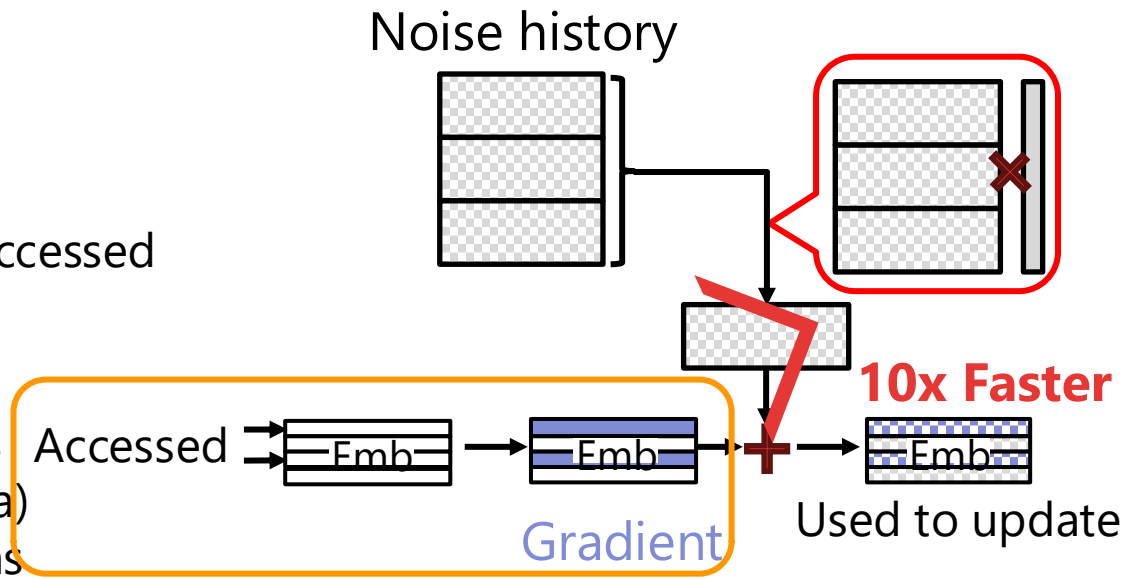
Deep Learning Recommendation Model (DLRM)



ccessed

a)

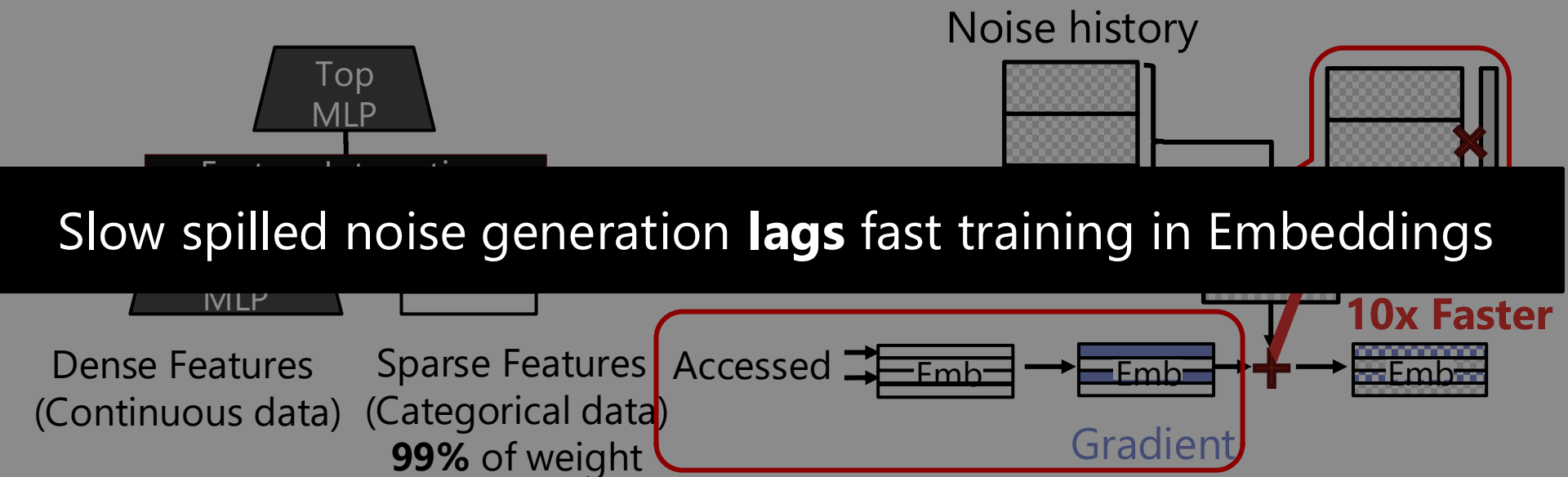
99.9% of params



Criteo Kaggle dataset **33M** entries, but fewer than **1M** accessed per iter

DP Training is Inefficient with DLRM Embedding

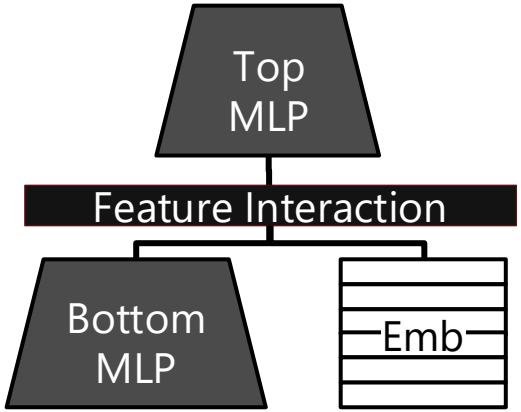
Deep Learning Recommendation Model (DLRM)



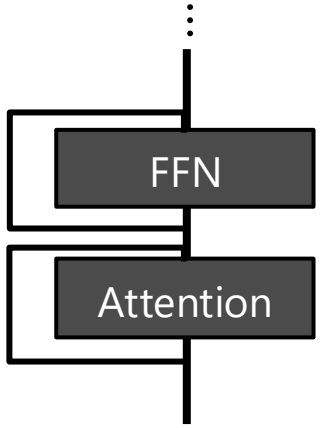
Slow spilled noise generation **lags** fast training in Embeddings

Criteo Kaggle dataset, **33M** entries but accessed **<1M** per iter

Three Model Types Studied



Recommendation Model (DLRM)



LLM (GPT-2, OPT)

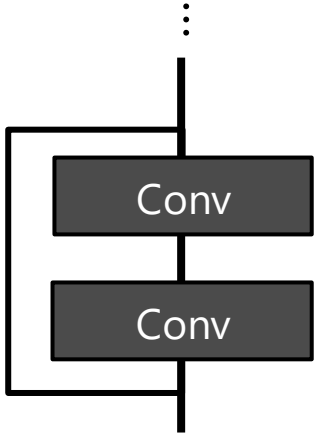
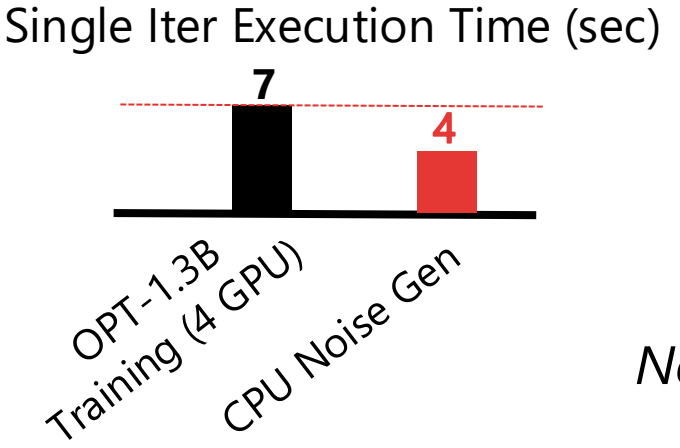
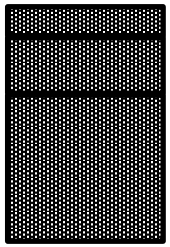


Image Models (ResNet, ViT)
(Omitted, similar to LLM)

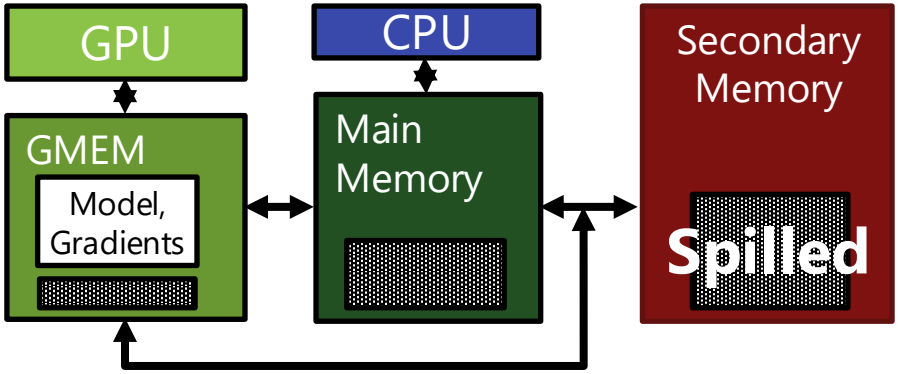
Dense Models

Noise History Transfer Dominates in LLM Training

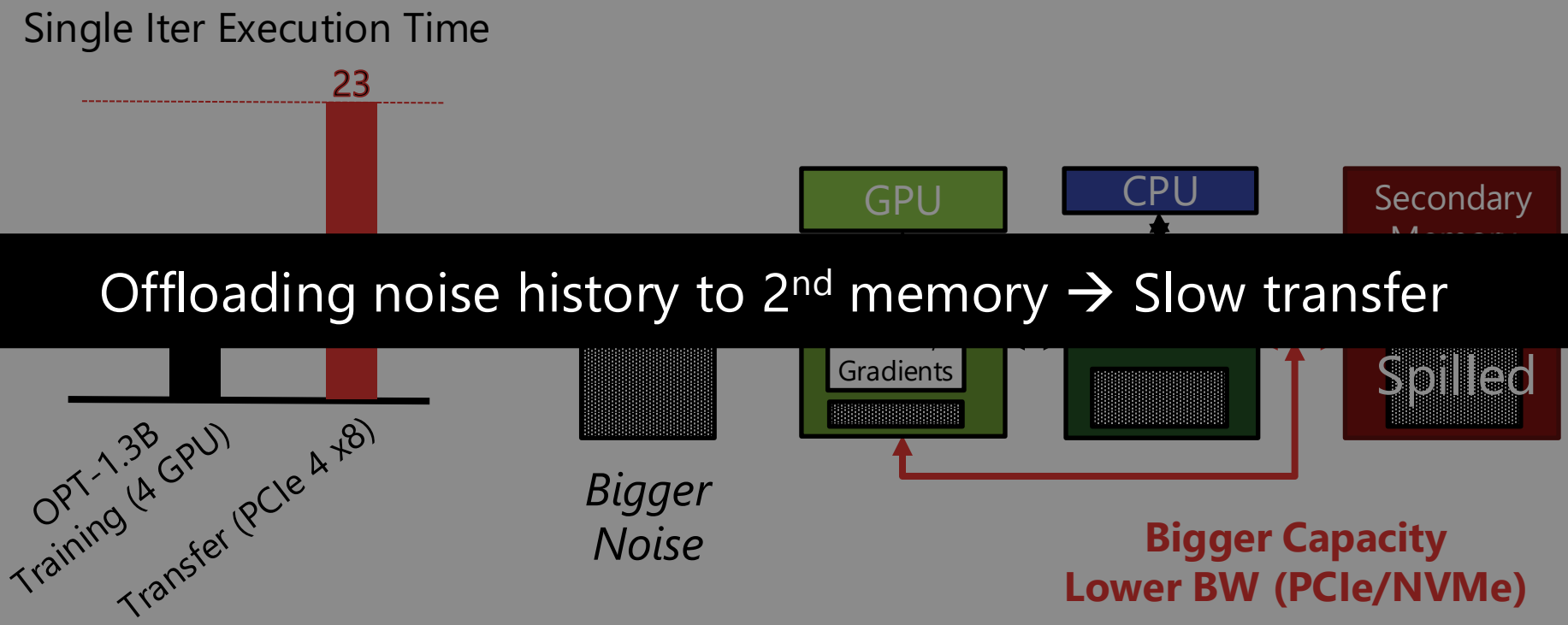




Bigger Noise History



Another Problem: DP Training Becomes Transfer-Bound



Outline

- Introduction
- Background: DP-SGD & Correlated Noise
- Characterization on Correlated Noise
- **Proposed Framework: Cocoon**
  - **Near memory processing (NMP)**
  - Precomputation & Coalescing
- Evaluation
- Conclusion

Cocoon: A Framework for DP training with Correlated Noise

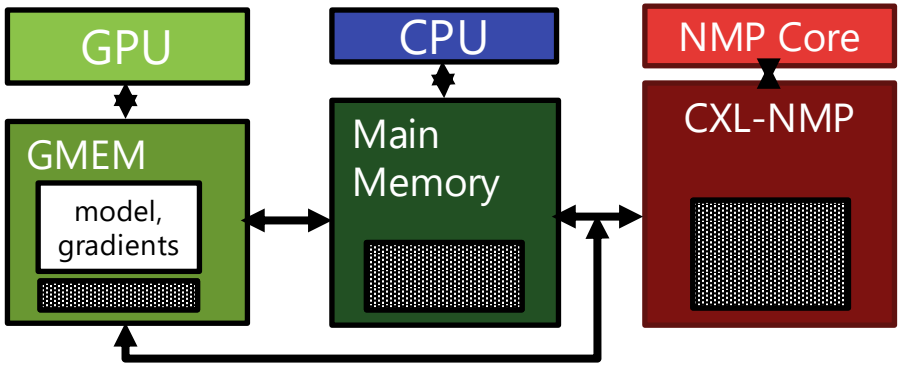
1. Data transfer from secondary memory is too slow

Near-memory-processing (NMP) & Profiled history distribution

2. Correlated noise computation is too slow in embeddings

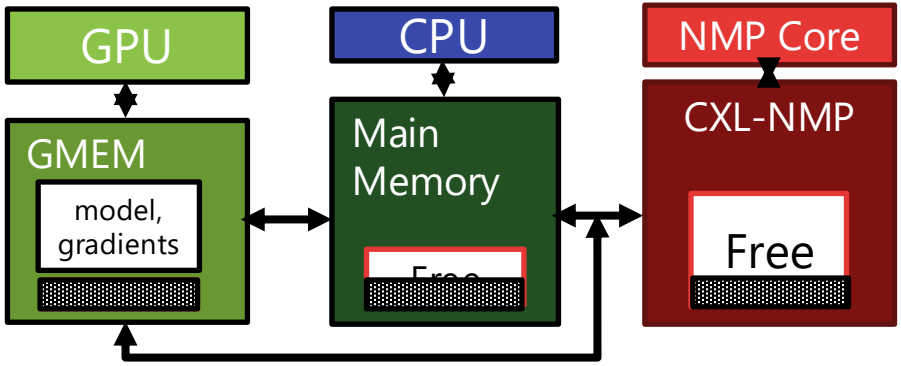
Noise Precomputation & Noise Coalescing

NMP: Compute Noise Where It Lives



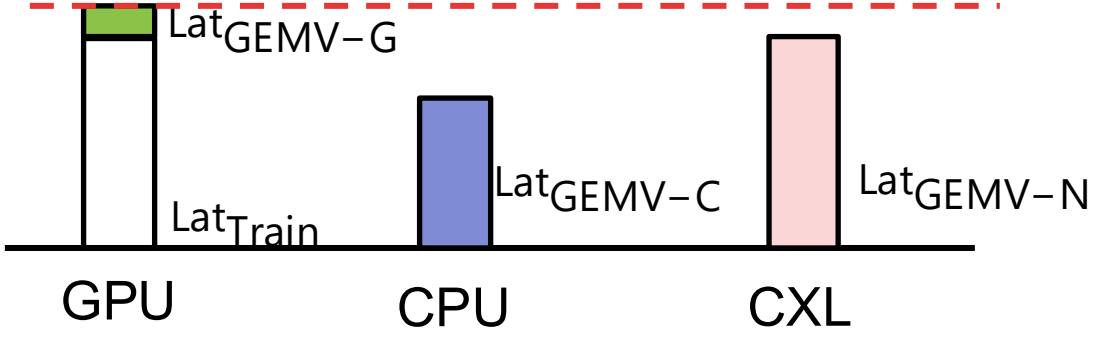
Each piece of noise history is computed at its nearest compute unit

Split History to Minimize Bottleneck Link

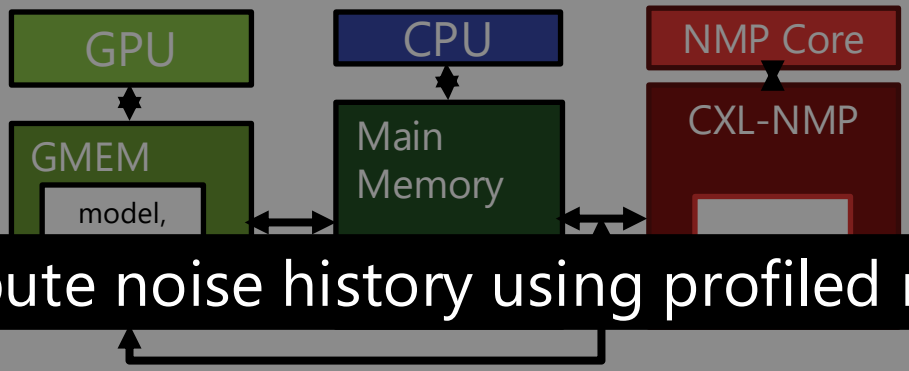


Goal

Minimize Max of Three Latencies

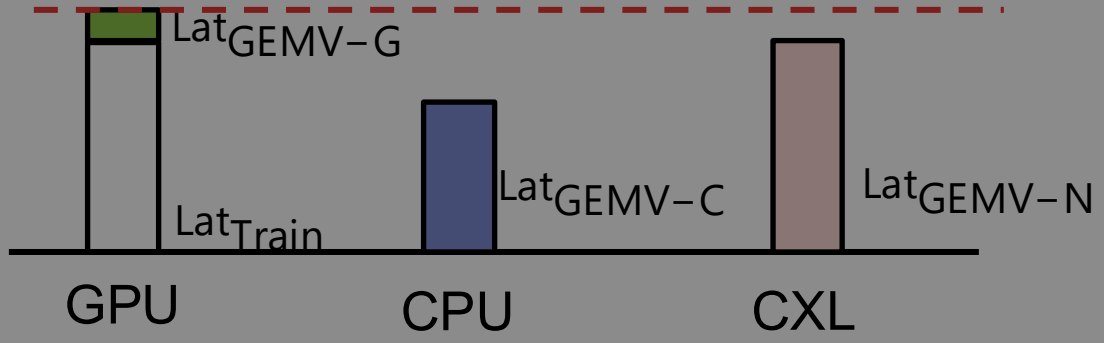


Split History to Minimize Bottleneck Link



Goal

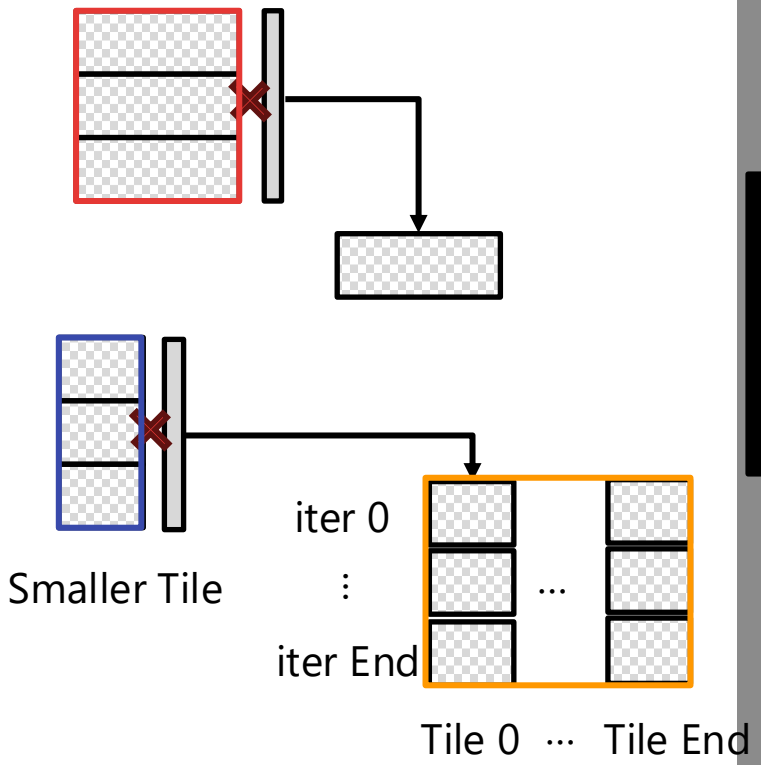
Minimize Max of Three Latencies



Outline

- Introduction
- Background: DP-SGD & Correlated Noise
- Characterization on Correlated Noise
- **Proposed Framework: Cocoon**
  - Near memory processing (NMP)
  - **Precomputation & Coalescing**
- Evaluation
- Conclusion

Pre-computation: No Spill, Full GPU Speed



Generation during training

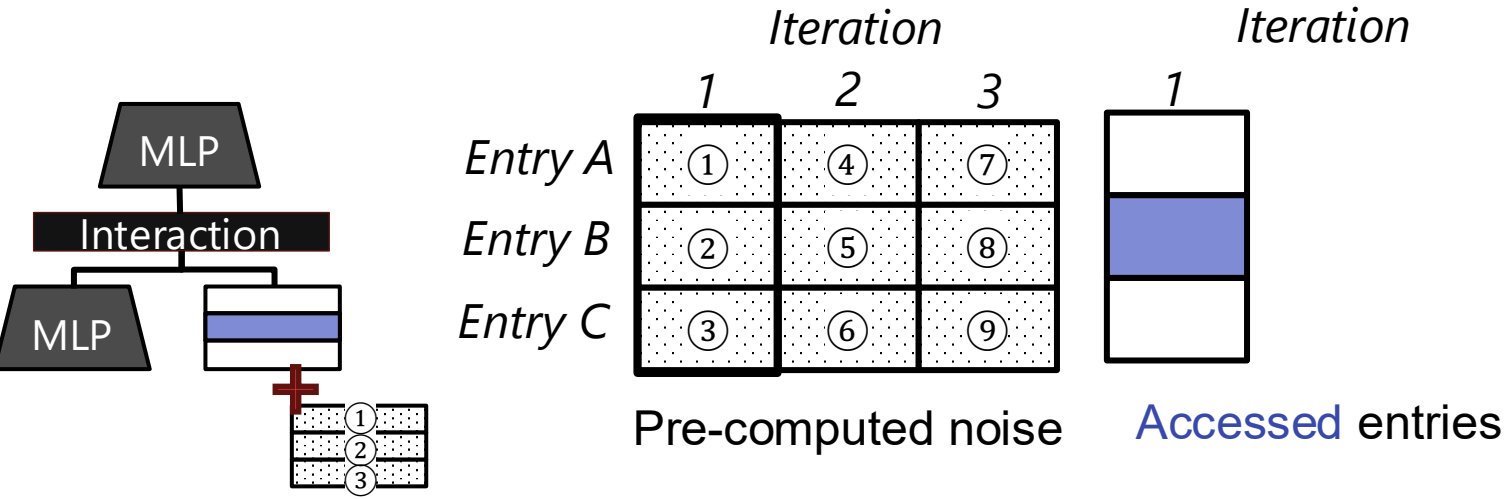
Overflows GPU capacity, Spill Overhead

Precompute: Fast, but stores noise for **all iterations** (>1000) instead of **band size** (<100)

Noise history tile always **fits** in GPU

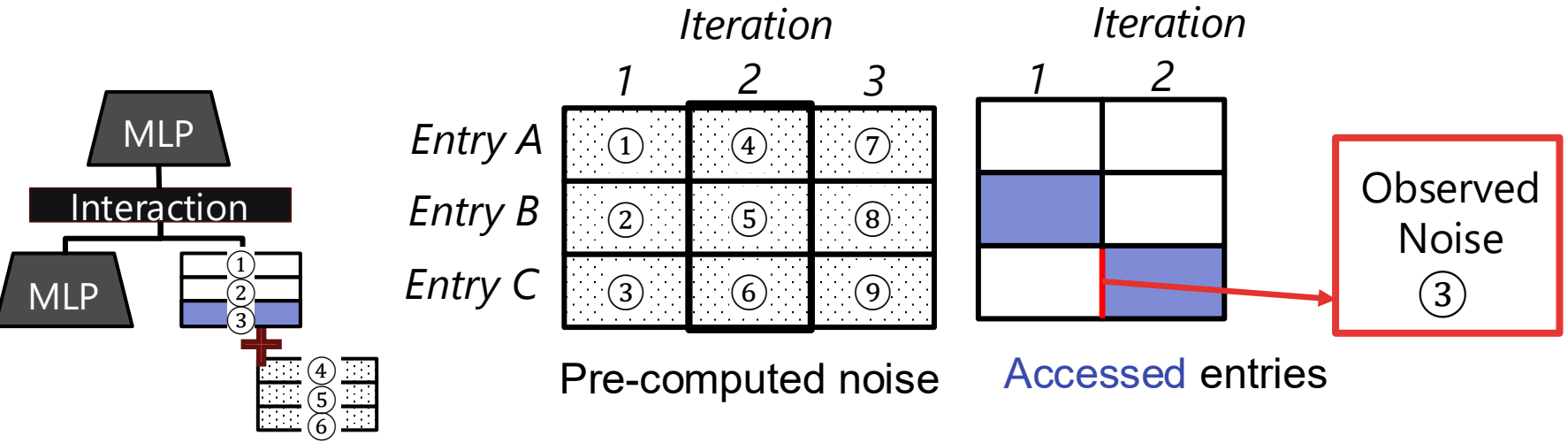
Sparse Access Lets Us Coalesce Noise

Accumulated noise is not observed until an entry is accessed



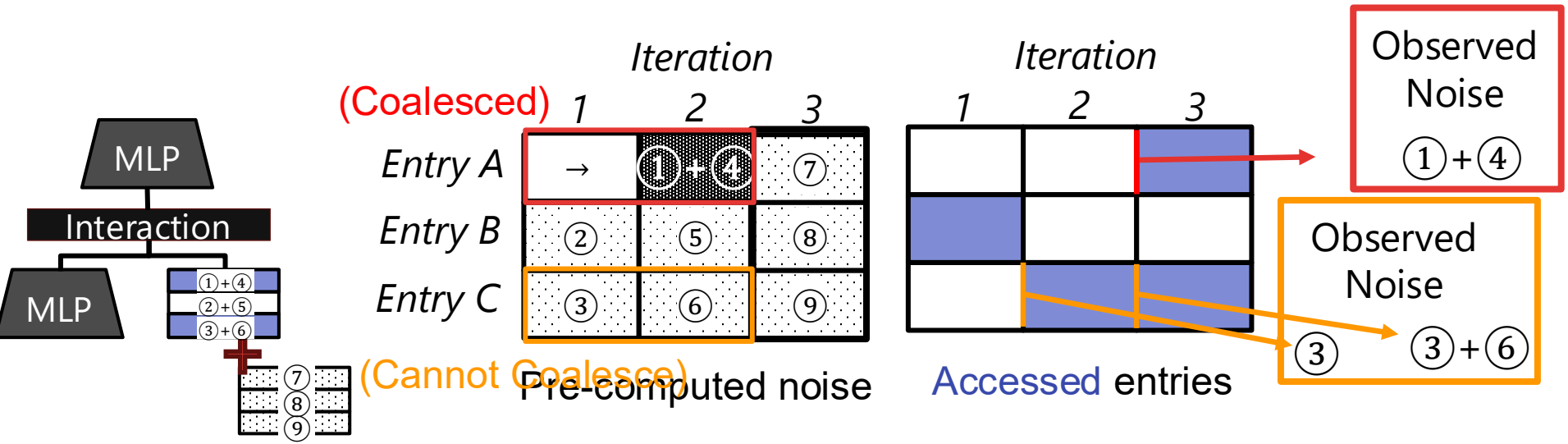
Sparse Access Lets Us Coalesce Noise – Before Iter 2

Noise is not observed until an entry is accessed or training ends



Sparse Access Lets Us Coalesce Noise – Before Iter 3

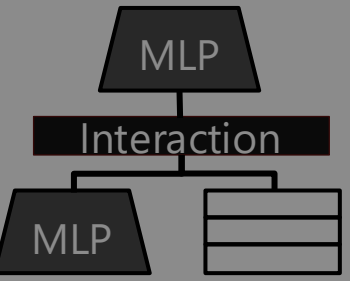
Noise is not observed until an entry is accessed or training ends

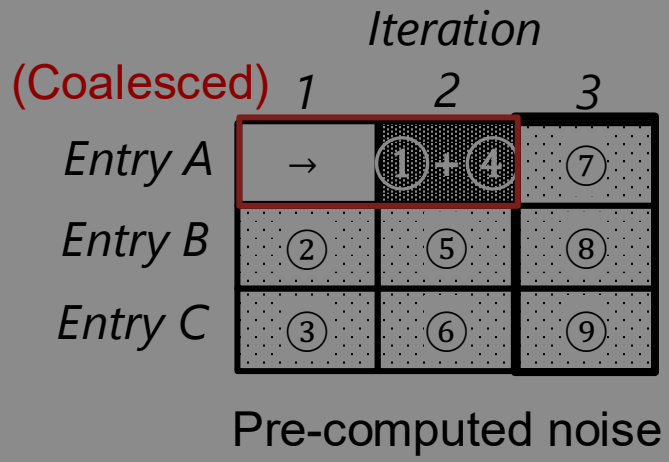


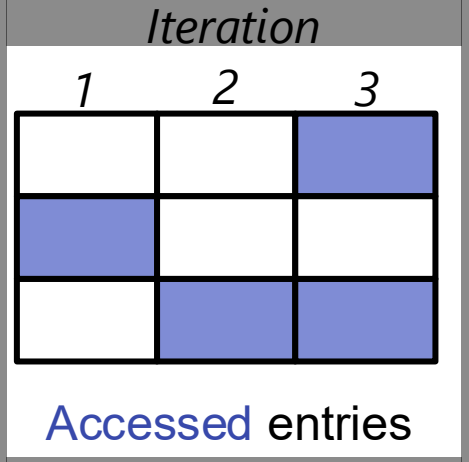
Defers the addition, coalescing until accessed, still equivalent

Sparse Access Lets Us Coalesce Noise – Before Iter 3

Noise is not observed until an entry is accessed or training ends







Access Pattern known in advance -> Coalesce while Precompute
Reduces memory over **50x**

Outline

- Introduction
- Background: DP-SGD & Correlated Noise
- Characterization on Correlated Noise
- Proposed Framework: Cocoon
- **Evaluation**
- **Conclusion**

Experimental Setup

GPU	
GPU type	4 NVIDIA RTX A5000 (24GB)
CPU	
CPU	Intel Xeon Gold 6330
Main Memory	8×DDR4 3200MHz 16GB (128GB)
CXL-NMP Prototype	
CXL Engine	Xilinx VP1502 FPGA
CXL MEM	DDR4 256GB

Training Workloads

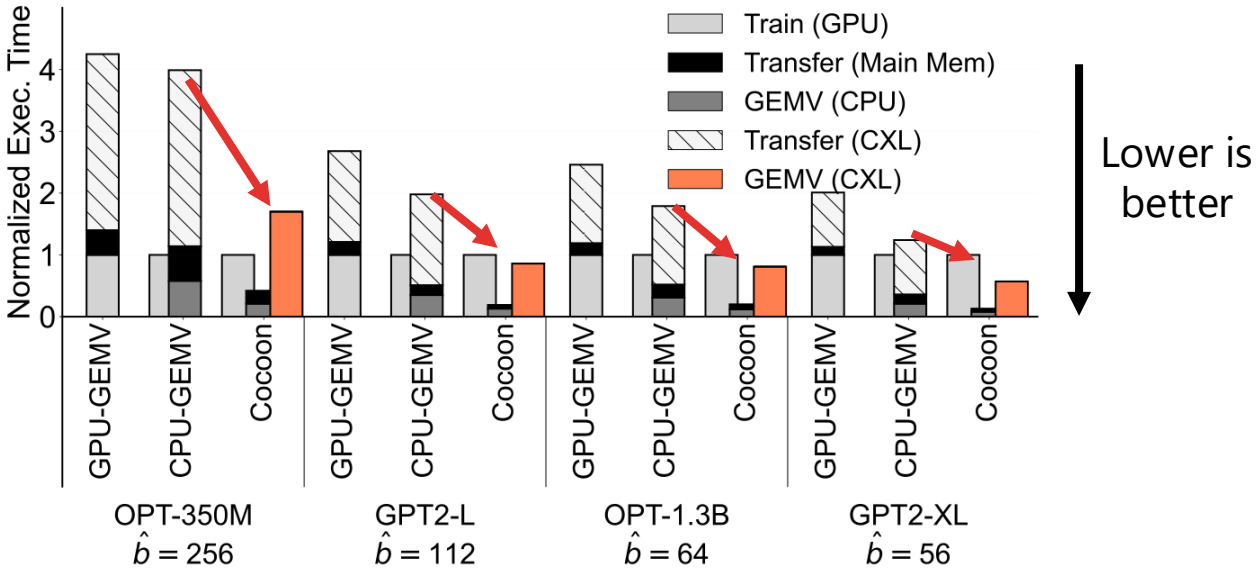
LLM	
GPT-2 *OPT*	E2E dataset
Recommendation Model	
DLRM	Criteo Kaggle, Synthetic
Image Classification	
CNN, ViT	ImageNet Dummy

Baselines

CPU-GEMV

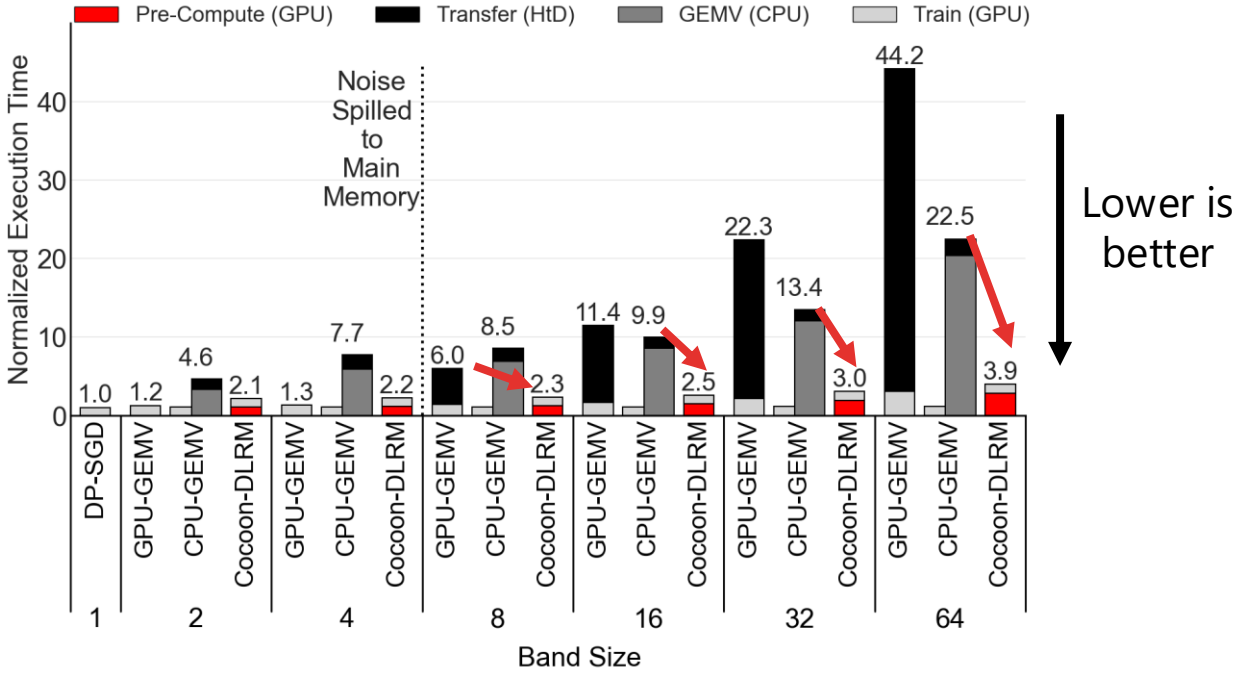
GPU-GEMV

NMP Improves Performance by Cutting Heavy Transfer Time



1.23-2.32x speedup over the baseline

DLRM Training Execution Time Decreases with Precomputation



Even with pre-computation, **2.67-4.87x** speedup
even after paying precompute cost

Conclusion

- Memory pressure of DP-correlated noise brings two problems
  - Data transfer from secondary device is too slow
  - Noise generation slower than sparse model training

- Our framework **Cocoon:**
  - Distributes noise history & Compute noise at nearest unit, 1.23-2.32x speedup
  - Exploits embedding sparsity, 2.67-4.87x e2e speedup



Cocoon: A System Architecture for Differentially Private Training with Correlated Noises

Donghwan Kim[1], Xin Gu[1], Jinho Baek[2], Timothy Lo[2], Younghoon Min[2], Kwangsik Shin[2], Jongryool Kim[2], Jongse Park[3], Kiwan Maeng[1]

[1] Pennsylvania State University, [2] SK hynix Inc, [3] KAIST







References

- ML Models Leak Training Data
  - [1] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. "Scalable extraction of training data from aligned, production language models.", ICLR, 2025
- Differential Privacy (DP) is the Fix, but SOTA Algorithms are Costly!
  - [1] John M Abowd. "The US Census Bureau adopts differential privacy". In Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining, 2018.
  - [2] Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. "Federated learning of Gboard language models with differential privacy." ACL Industry Track, 2023.
  - [3] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. "Privacy loss in Apple's implementation of differential privacy on MacOS 10.12." In arxiv.org, 2017.
  - [4] Amer Sinha, Thomas Mesnard, Ryan McKenna, Daogao Liu, Christopher A. Choquette-Choo, Yangsibo Huang, Da Yu, George Kaissis, Zachary Charles, Ruibo Liu, Lynn Chua, Pritish Kamath, Pasin Manurangsi, Steve He, Chiyuan Zhang, Badih Ghazi, Borja De Balle Pigem, Prem Eruvbetine, Tris Warkentin, Armand Joulin, and Ravi Kumar. "VaultGemma: A differentially private Gemma model." In arxiv.org, 2025.
  - [Abadi '16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy." CCS. 2016.
  - [Bu' 24] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. "Differentially private optimization on large model at small cost." ICML, 2023.
  - [Choquette-Choo '23] Christopher A Choquette-Choo, Arun Ganesh, Ryan McKenna, H Brendan McMahan, John Rush, Abhradeep Guha Thakurta, and Zheng Xu. "(Amplified) Banded Matrix Factorization: A unified approach to private training." NeurIPS, 2023.
- Prior Works Used Abundant Resource
  - [1] Ryan McKenna, "Scaling up the Banded Matrix Factorization Mechanism for Differentially Private ML", ICLR 2025