Does Fully Homomorphic Encryption Need Compute Acceleration?

Rashmi Agrawal, Leo de Castro,
Rabia Yazicigil, Anantha Chandrakasan, Vinod Vaikuntanathan, Chiraag Juvekar, Ajay Joshi

BOSTON UNIVERSITY

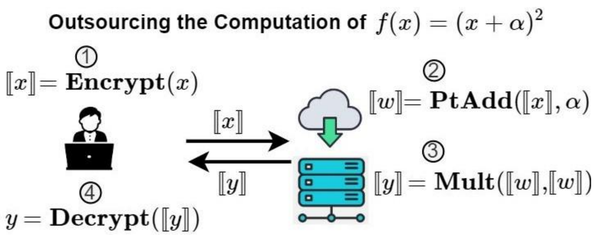
MIT CSAIL

Fully Homomorphic Encryption

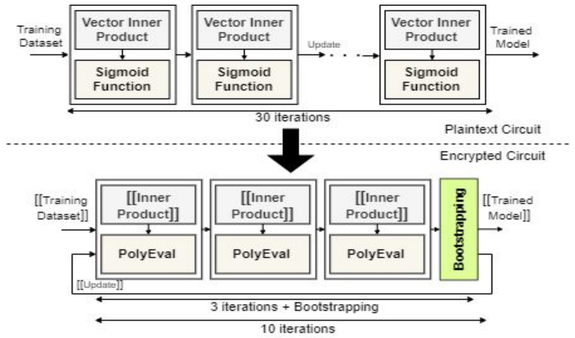
- Enables computation over **encrypted data**.
- Supports **any** arithmetic circuit.
- Requires **bootstrapping** for unlimited computation.

Applications

Secure computation outsourcing



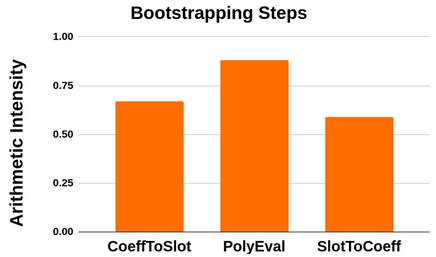
Secure AI Training: Logistic Regression

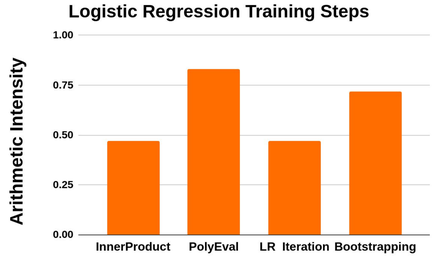


Problem: **Bootstrapping is the major performance bottleneck**

Bootstrapping in Hardware

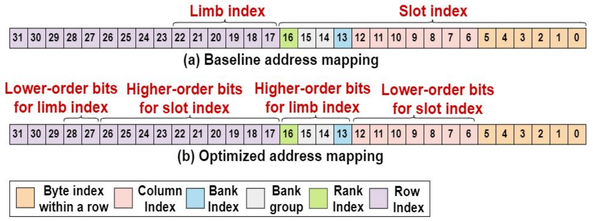
- Bootstrapping has low **arithmetic intensity**.
- Bottleneck is the **memory bandwidth**.





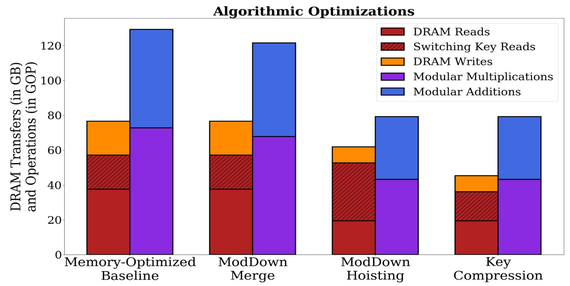
Our Contributions

Memory access optimizations: Optimize **low-level memory** organization for faster access for different orientations.



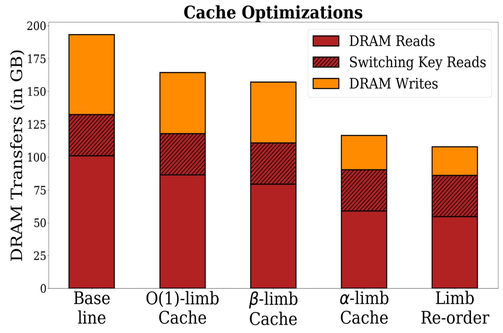
Algorithmic optimizations: Improve the arithmetic intensity of the **low-level operations**.

- Computation vs. memory **trade-off.**
- Improved homomorphic **matrix-vector product.**



Caching optimizations: Re-order the operations to maximize cache utilization.

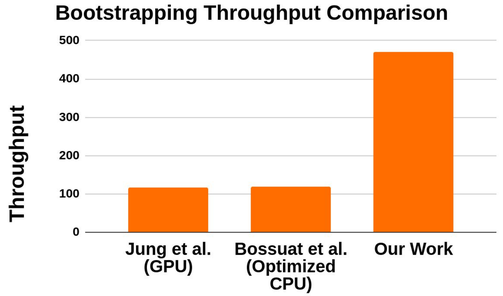
- **Custom tool** to enumerate various optimizations and select parameters to **optimize throughput**.



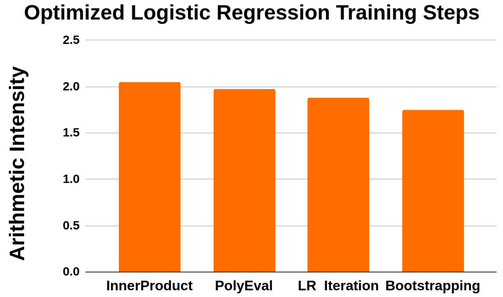
Results

- Significant improvements in **bootstrapping throughput**.

$$\text{throughput} = \frac{(\#\text{ of plaintext slots}) \cdot (\#\text{ of compute levels}) \cdot (\#\text{ bits of precision})}{\text{bootstrapping runtime}}$$



- Improved **arithmetic intensity.**



Future Perspectives: Memory bandwidth is still the **bottleneck**.

- Silver lining: FPGAs can achieve **near-ASIC performance at a fraction of the cost**.
- **Best near-term HE accelerator.**