# DELPHI: Cryptographic Inference for Neural Networks


**Pratyush Mishra**

Ryan Lehmkuhl

Akshayaram Srinivasan

Wenting Zheng

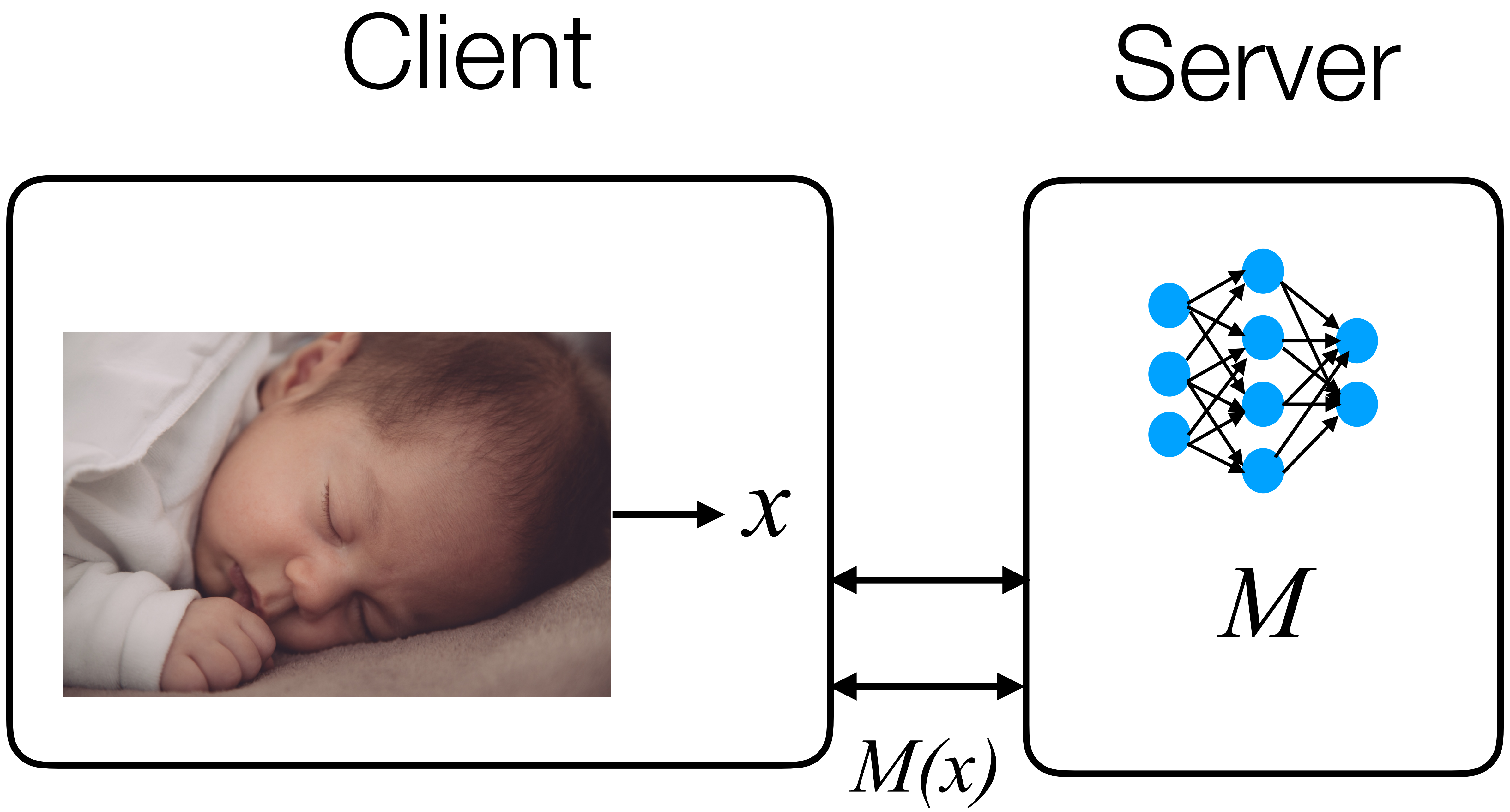
Raluca Ada Popa

UC Berkeley

# Neural Network Inference

A growing number of applications use neural networks in user interactions

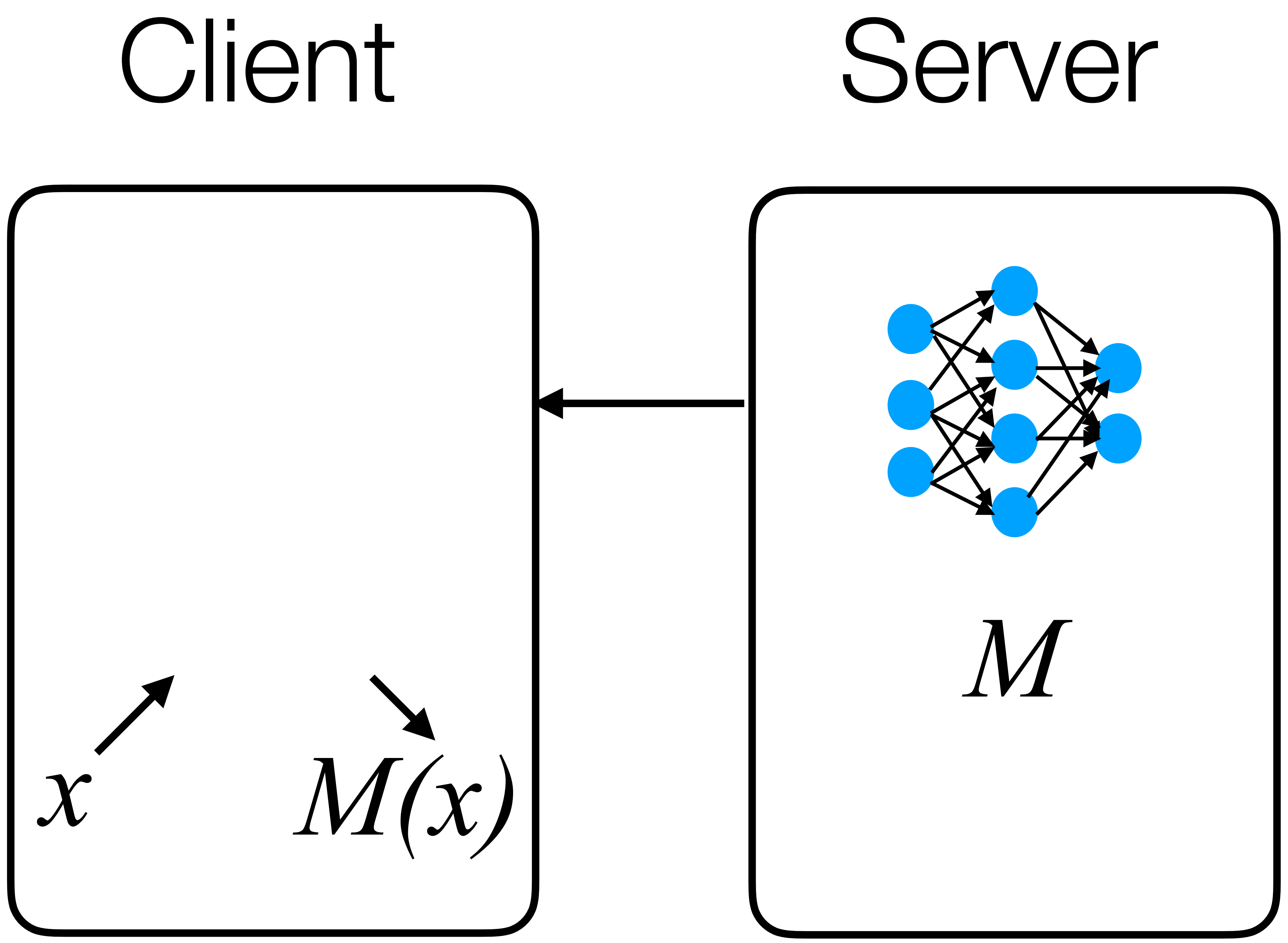
- Home monitoring: detect and recognize visitors
- Baby monitor: motion detection to alert parents




User data is sensitive

Server's model is proprietary

# Client-side inference

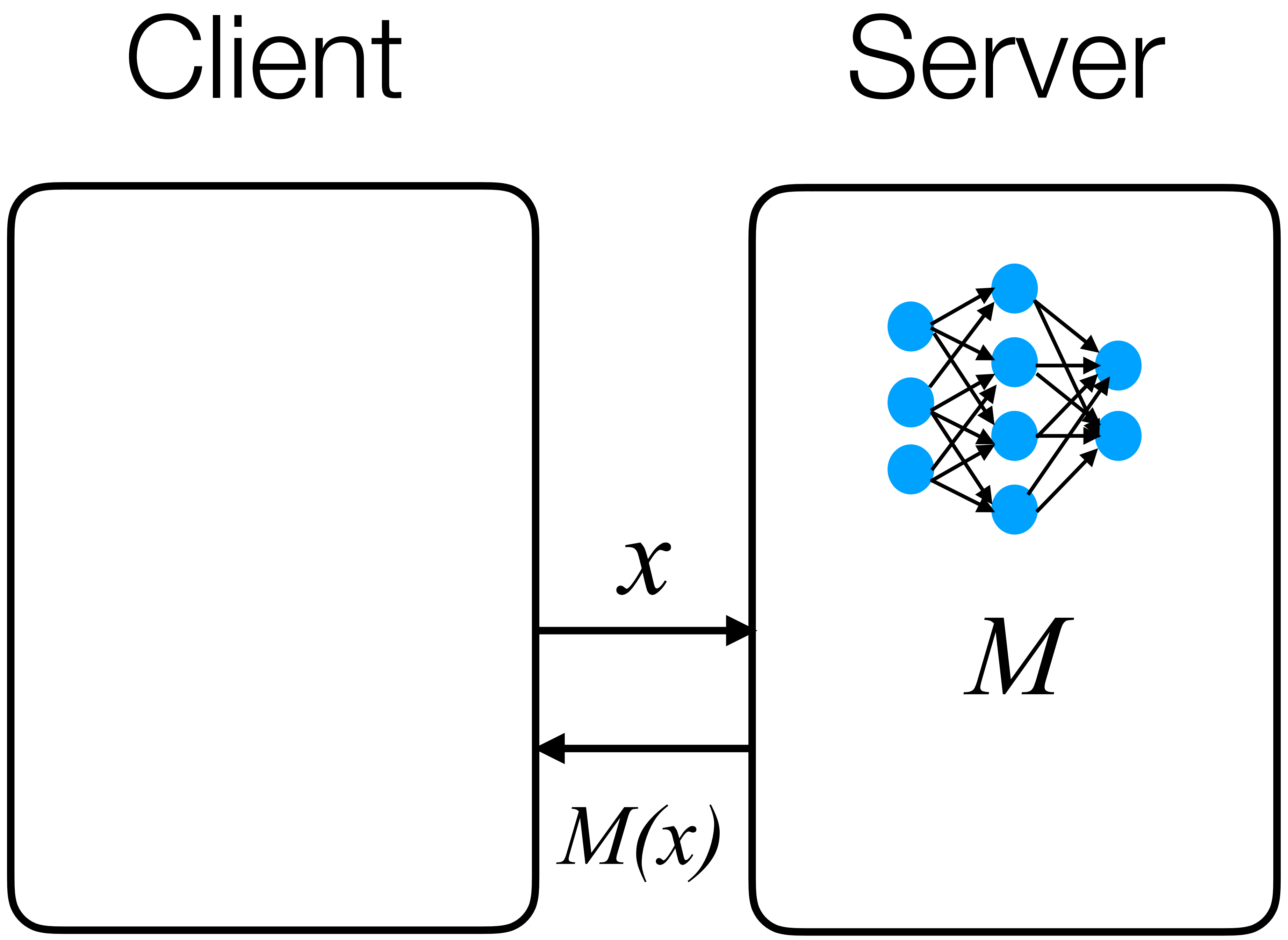



Client sees server's model!

This reveals model weights and leaks information about private training data

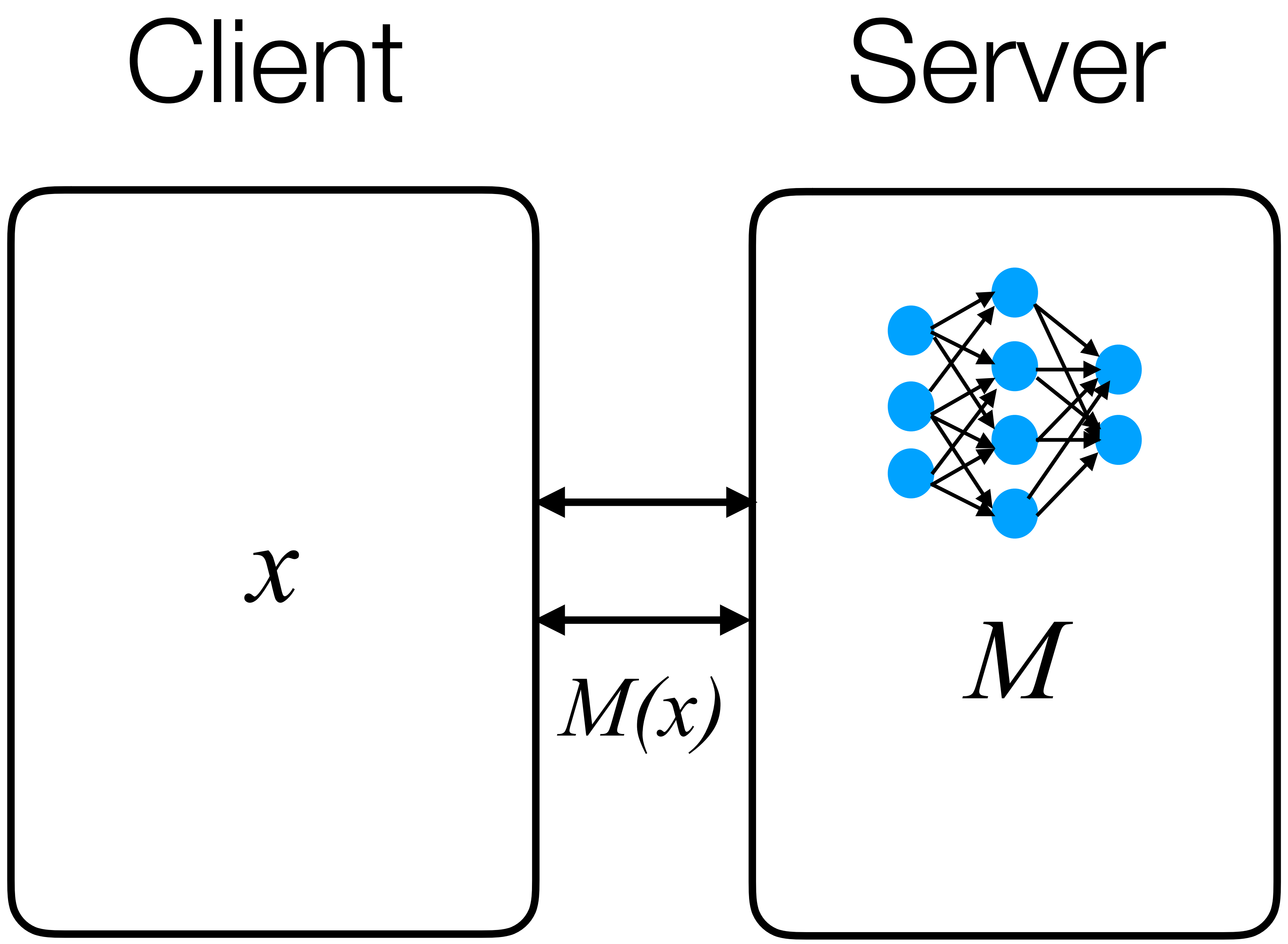
[SRS17], [CLEKS18], [MSCS18]

# Server-side inference




Server sees client data!

# Secure inference goals




Client (& server) should learn *only* prediction $M(x)$

Server should not learn private client input $x$

Client should not learn private model weights $M$

# Prior work on secure inference

| Protocol type | FHE based | 2PC based | Desired |
|---|---|---|---|
| Examples | CryptoNets, CHET, TAPAS | SecureML, Gazelle, MiniONN | **Delphi** |
| **Performance** | | | |
| **Functionality/ Accuracy** | | | |

# Delphi

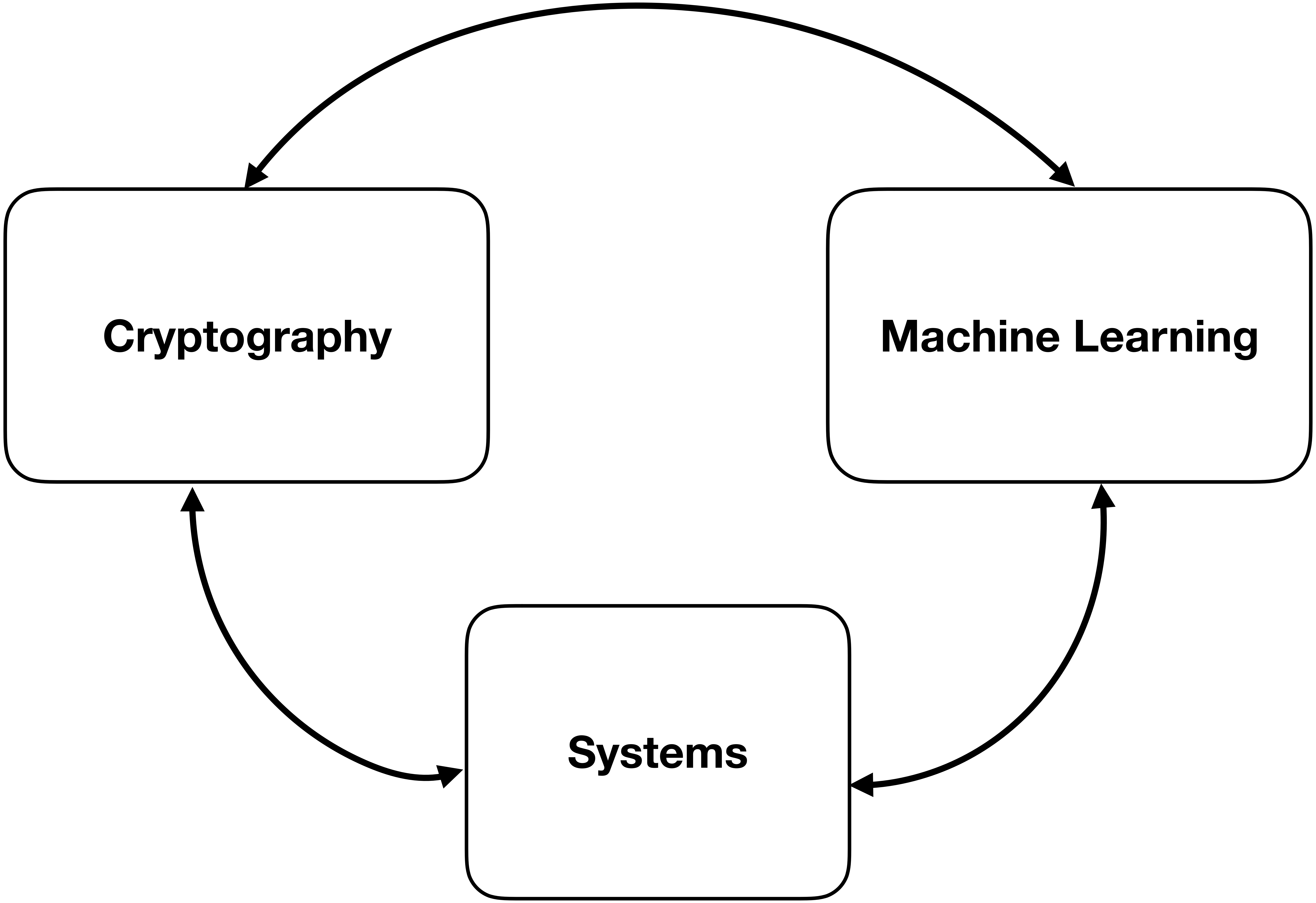
Cryptographic system for secure inference
on convolutional neural networks

**Security:** achieves **semi-honest simulation-based security**

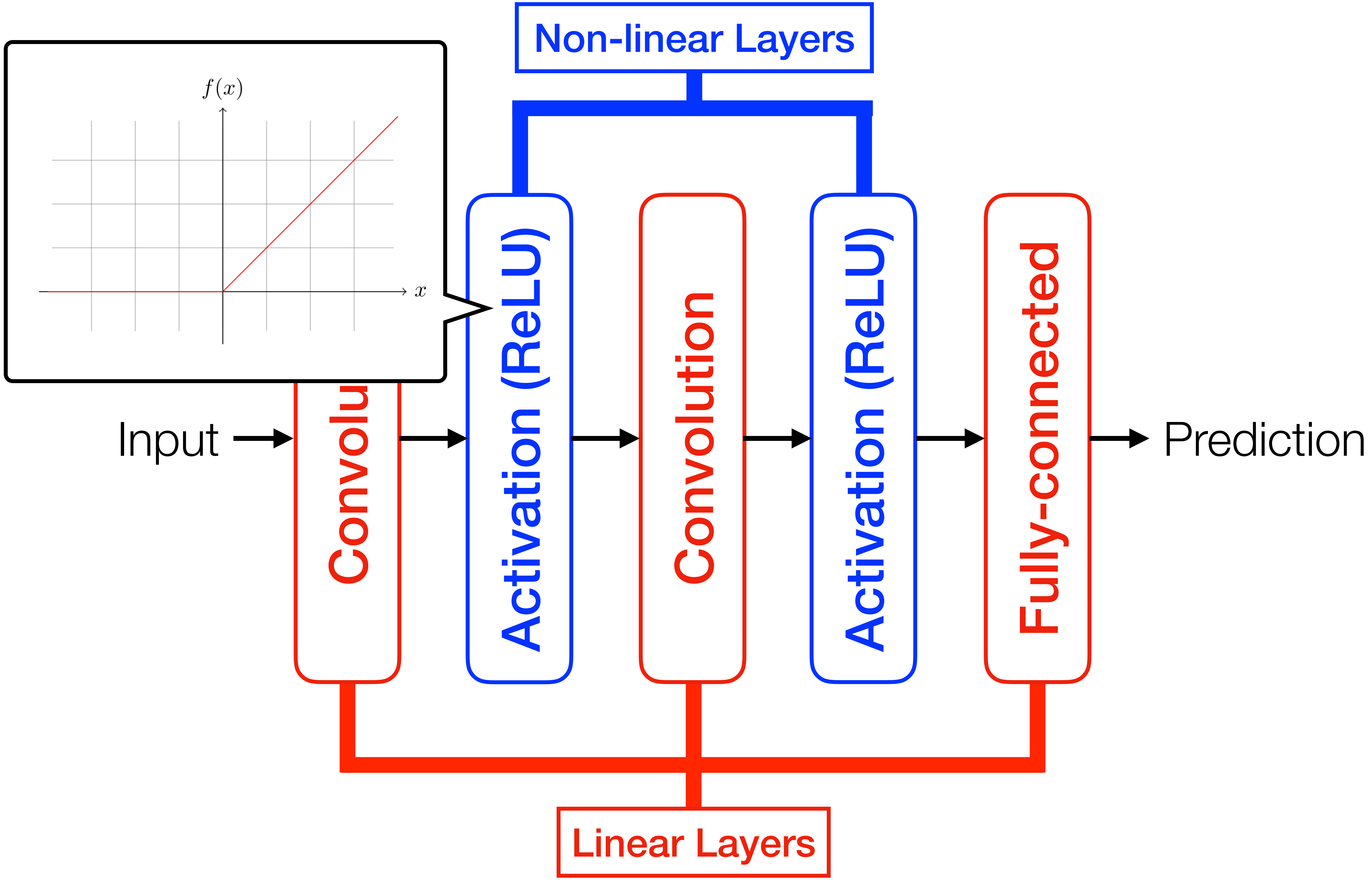
**Functionality:** supports **arbitrary CNNs**

**Efficiency:**

- improves **bandwidth** (9x) and **inference latency** (22x)
- can **utilize GPU/TPU for linear layers**
- evaluated on **realistic workloads** (CIFAR-100, ResNet-32)

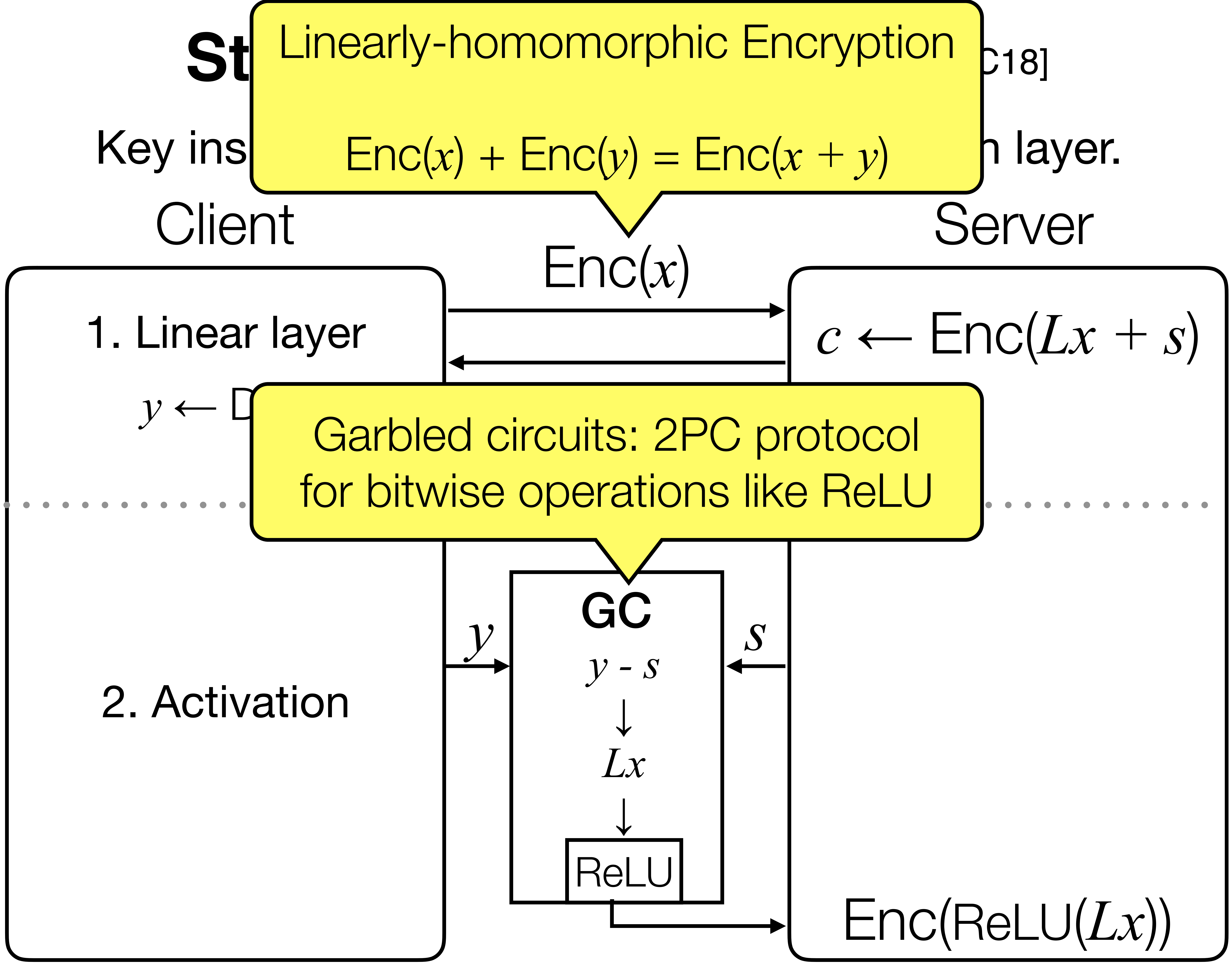

Cryptography
Machine Learning
Systems

# Recap: Convolutional Neural Networks

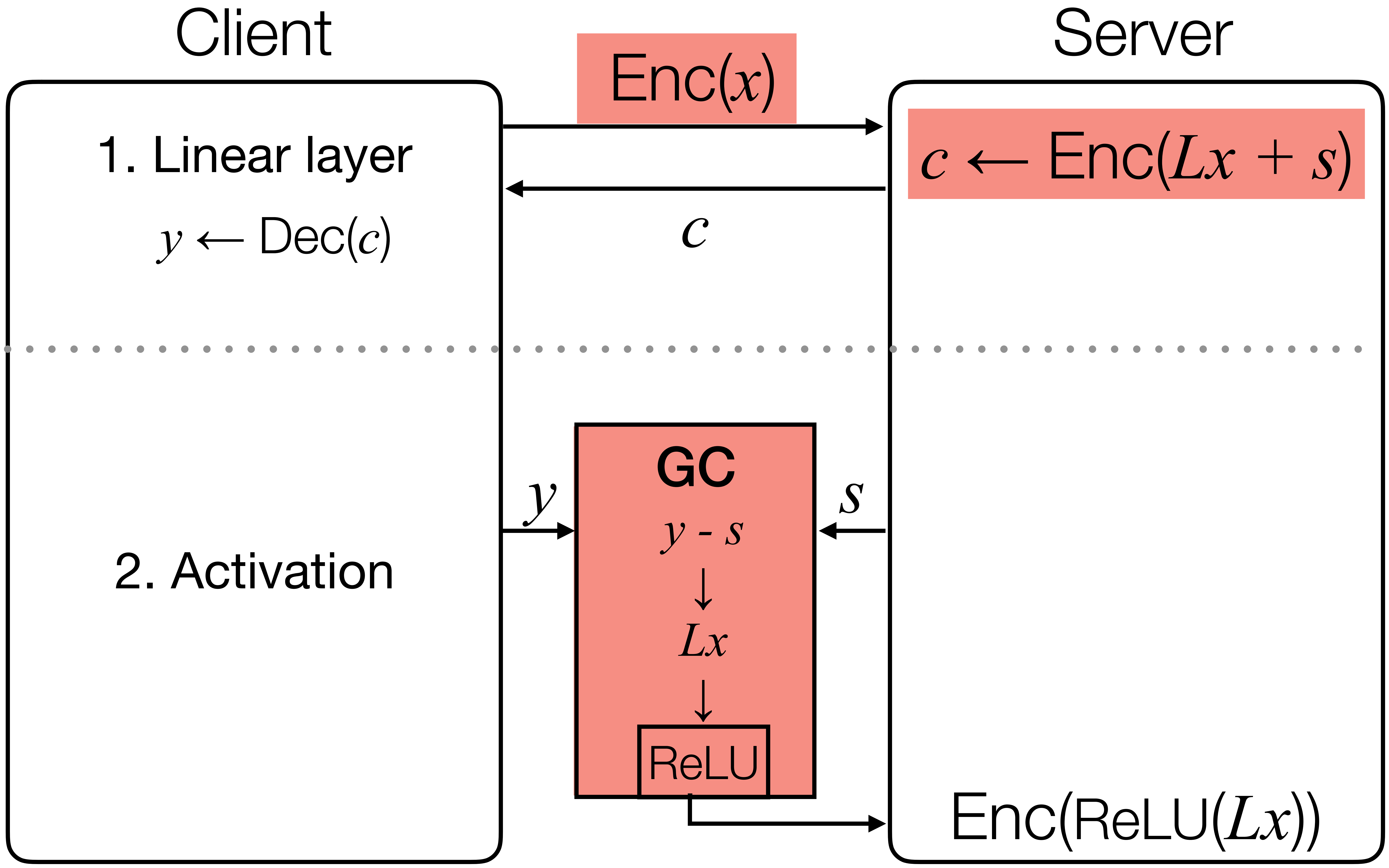


St [C18]

Key ins… layer.

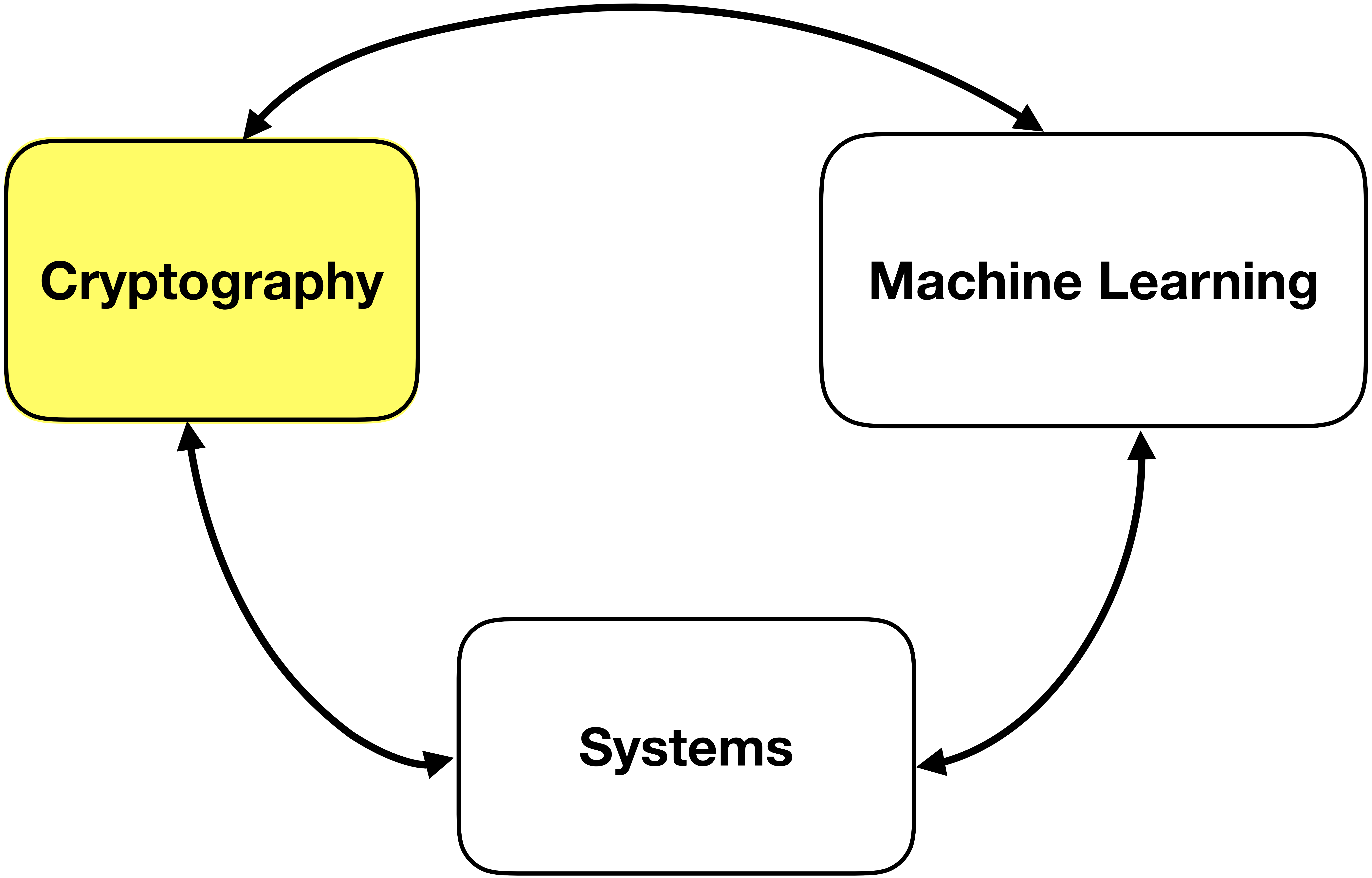


# Expensive parts of Gazelle

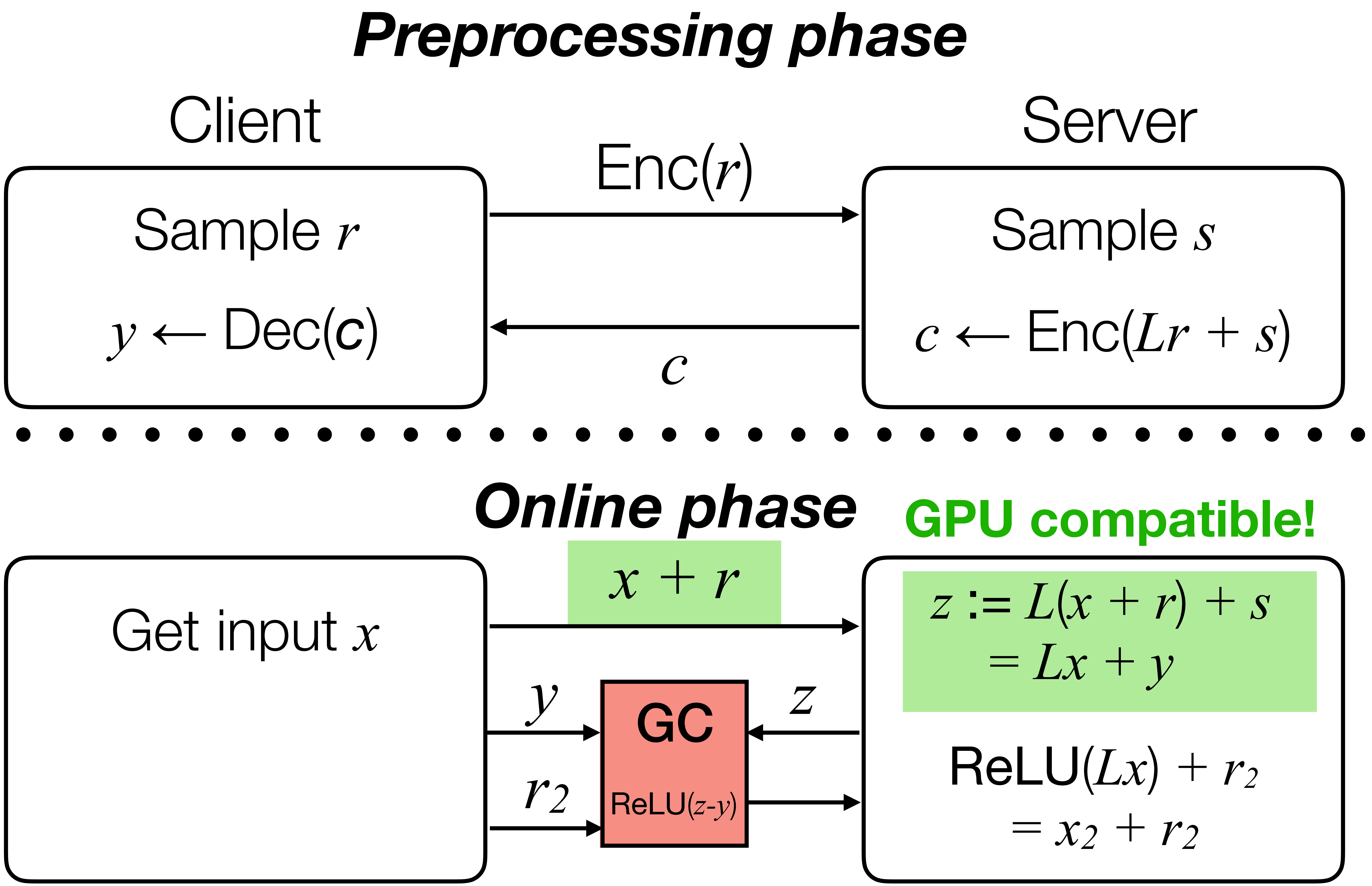



**no GPU!**

For ResNet-32, per inference: **~600MB** communication, and **~82 sec** latency.


Cryptography
Machine Learning
Systems

# Delphi: Optimizing Linear layers



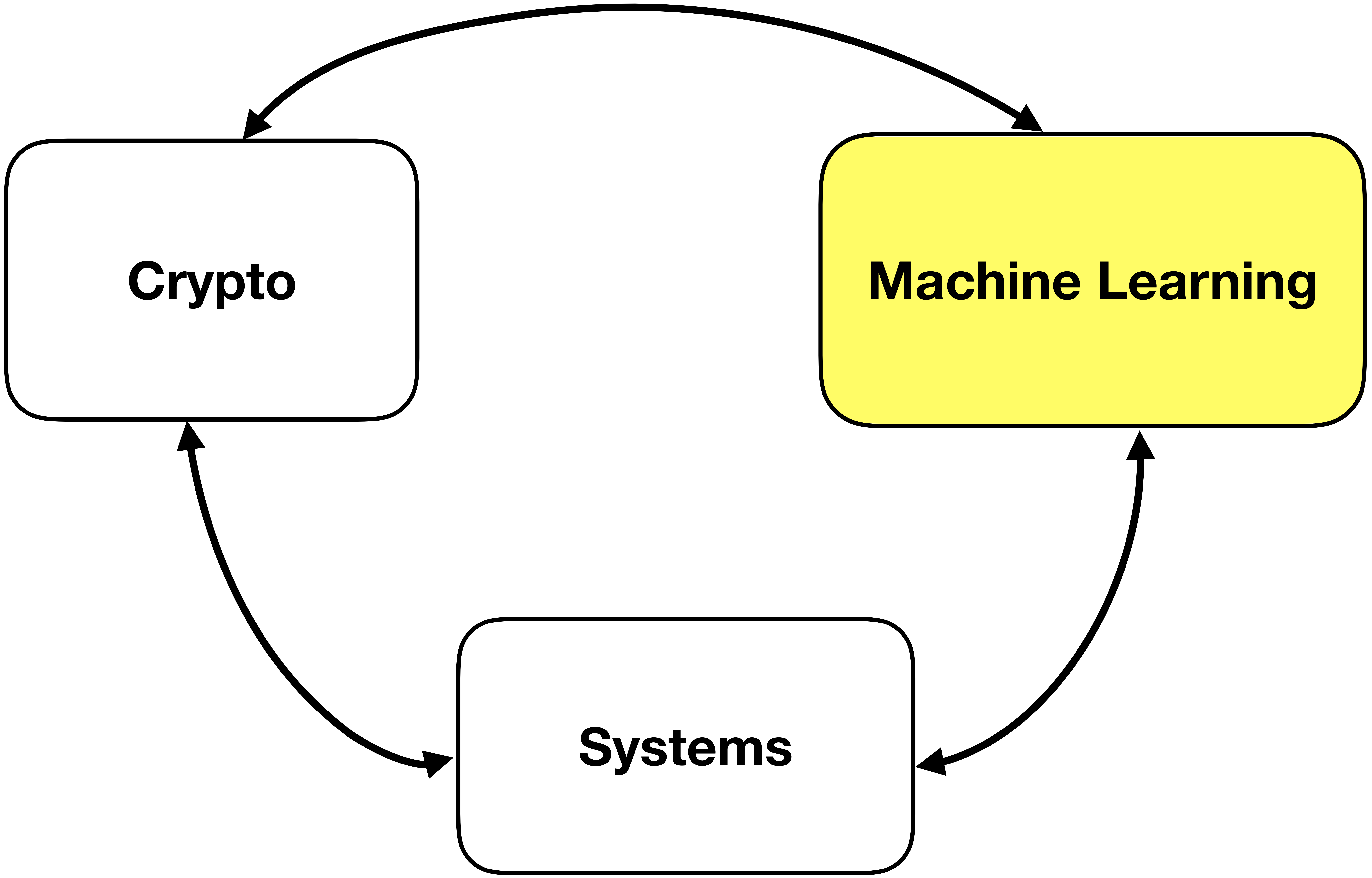

Per inference:
~~>600MB~~ ~350MB
communication,
~~~82 s~~ ~13 s
latency
~~~

# Delphi: Optimizing Non-linear Activations

**Problem:** ReLU is cheap for CPUs, but **costly** in 2PC.

**Solution Idea:** Replace ReLUs with quadratic activations, which *are* cheap in 2PC
[CryptoNets, SecureML]

**Problem:** Training accurate quad. networks is difficult: algorithms are optimized for all-ReLU networks


Crypto
Machine Learning
Systems

# Delphi's Machine Learning Planner

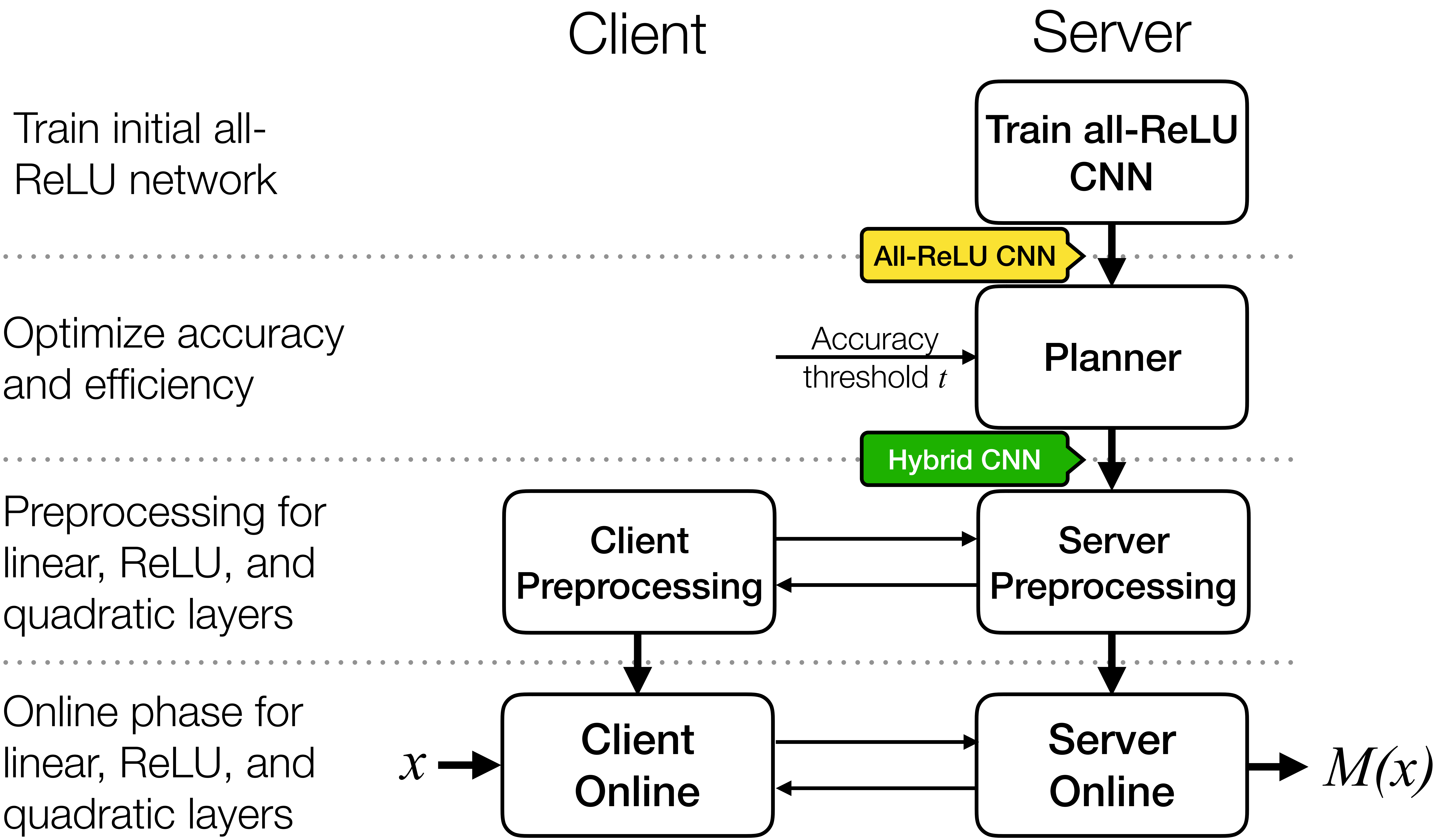

Better techniques for training hybrid networks

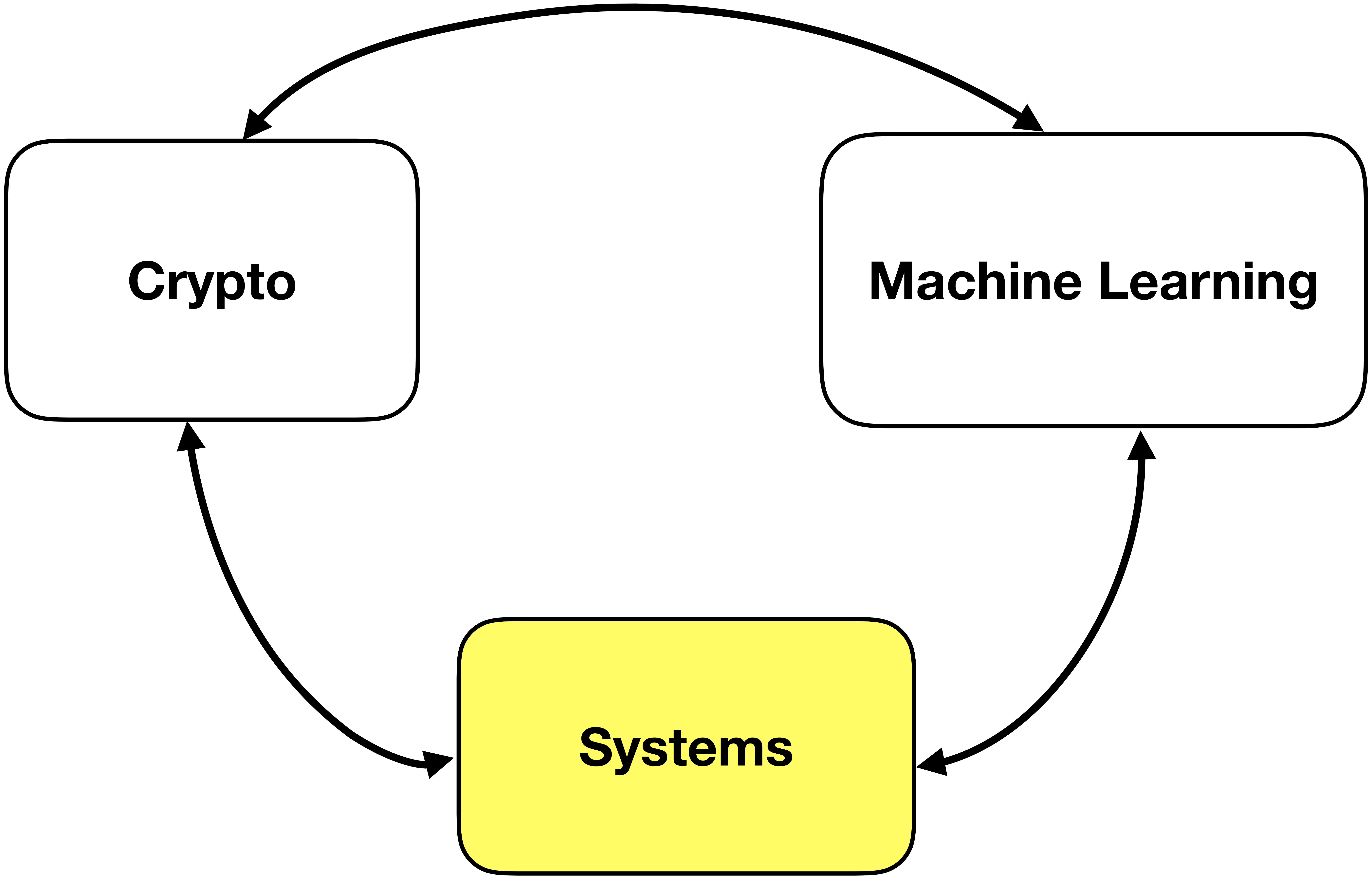
- Clipping gradients
- Blending in quadratic layers slowly

Specializing **Neural Architecture Search** to discover hybrid networks

- Adapt PBT algorithm
- Iterative exploration of search space

# Delphi's end-to-end workflow




Crypto
Machine Learning
Systems

# Implementation

Rust + C++ library with support for GPU acceleration

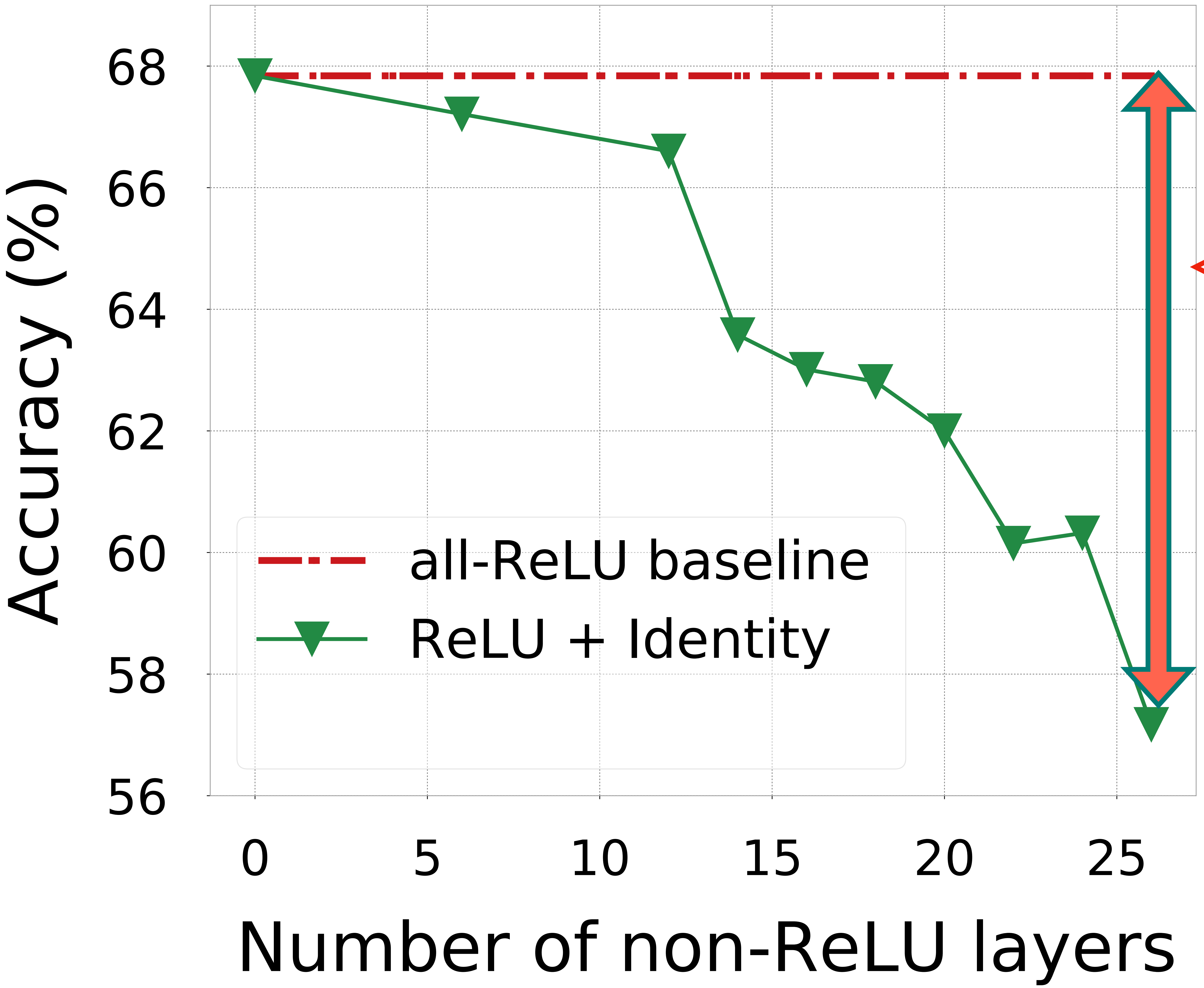
github.com/mc2-project/delphi

# Evaluation

1. Does Delphi's **planner preserve accuracy**?
2. Does Delphi's **protocol reduce latency & bandwidth**?

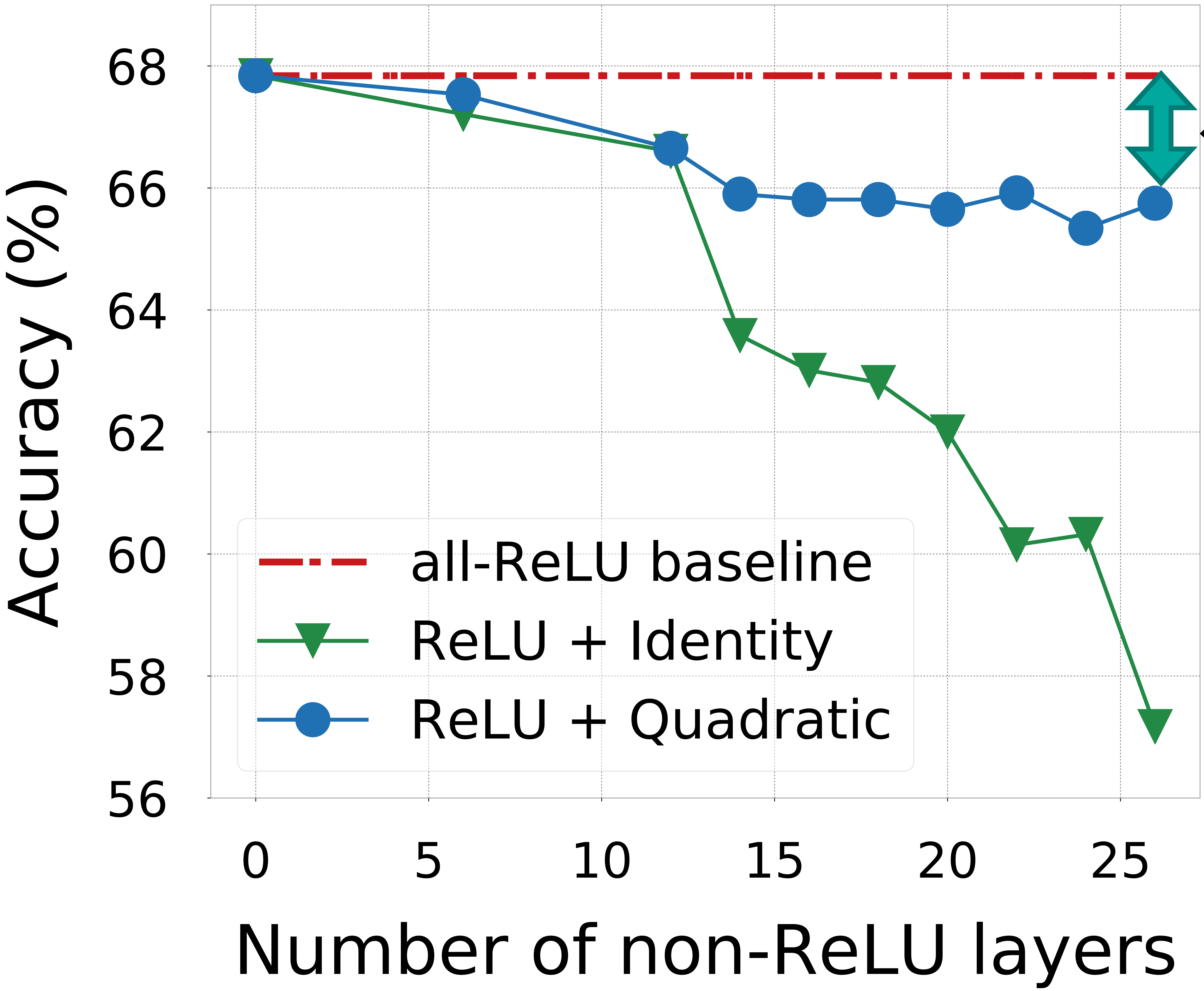
**Benchmark:** ResNet-32 network on CIFAR-100

# Planner accuracy




ReLUs are not redundant: accuracy loss > 10%

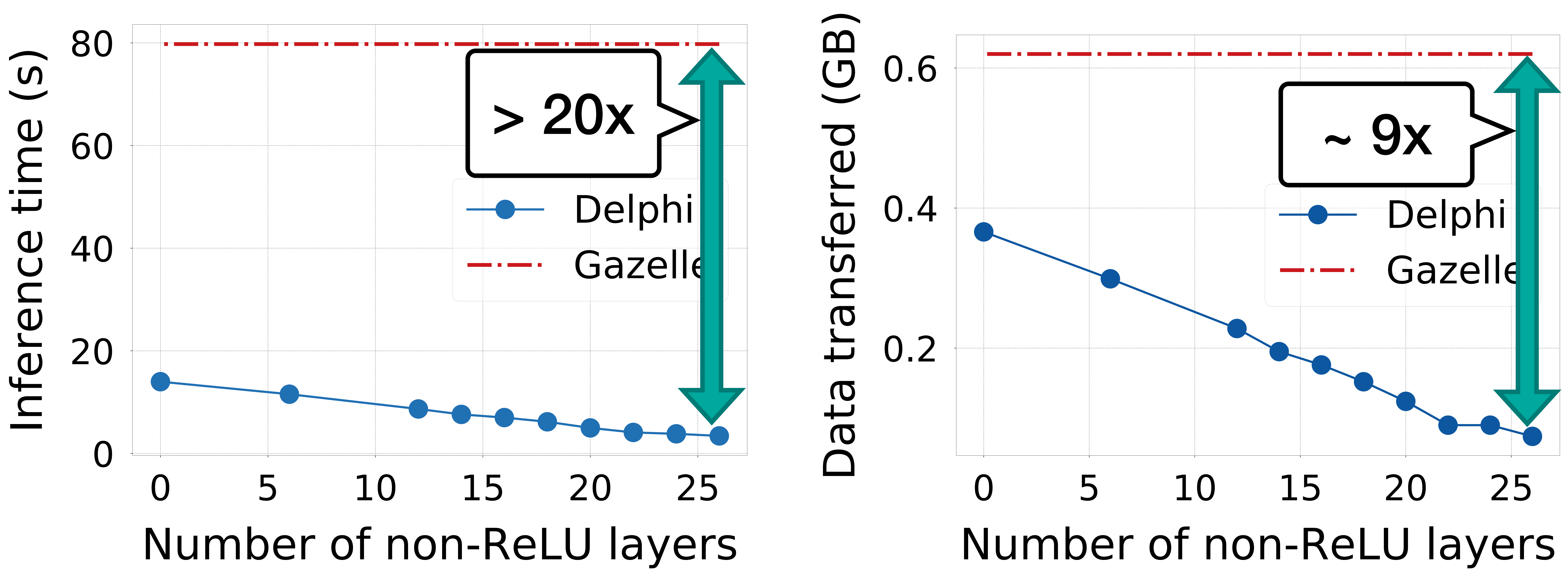
# Planner accuracy




Most efficient planned network achieves loss of < 2%

# Latency and communication

## Comparison with Gazelle



# Delphi

- Secure inference on convolutional neural networks
- 9-22x more efficient than prior work
- Combines techniques from systems, cryptography, and ML

ia.cr/2020/050

github.com/mc2-project/delphi