SoK: Cryptographic Neural-Network Computation

Lucien K. L. Ng[1], Sherman S. M. Chow[2]

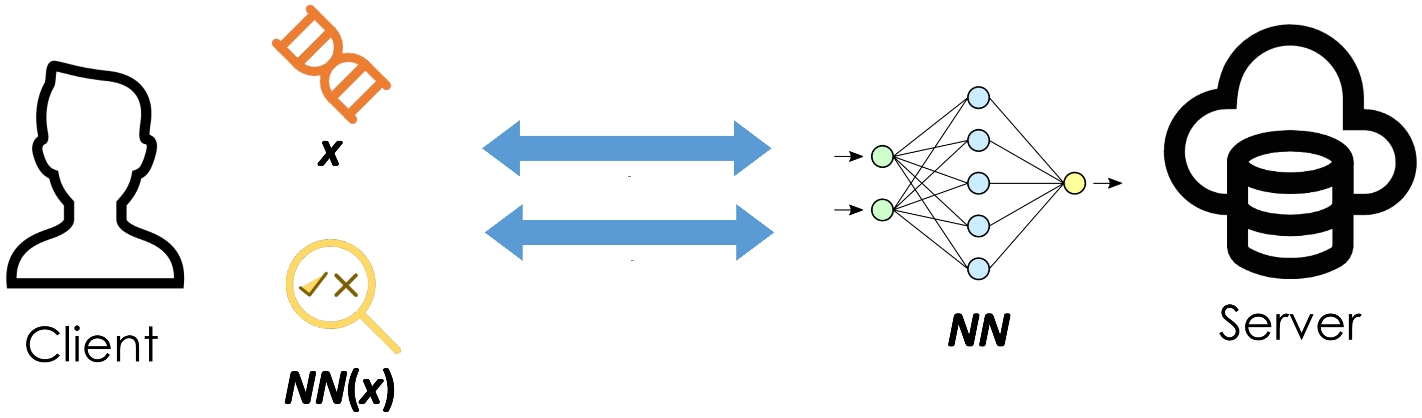
[1]Georgia Institute of Technology, USA
luciengkl@gatech.edu

[2]Chinese University of Hong Kong, Hong Kong
sherman@ie.cuhk.edu.hk

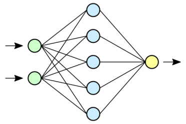
Privacy-preserving Neural Network (PPNN)

- Privacy Services
  - Oblivious Inference ⊆ Outsourced Inference
  - Outsourced Inference ⊆ Outsourced Training
  - Outsourced Training ⊆ Private Training
- Our Motivations
- Highlights of Three Types of Frameworks
- Evaluation over WAN

Oblivious Inference



Hide *Model Parameters* (mainly)

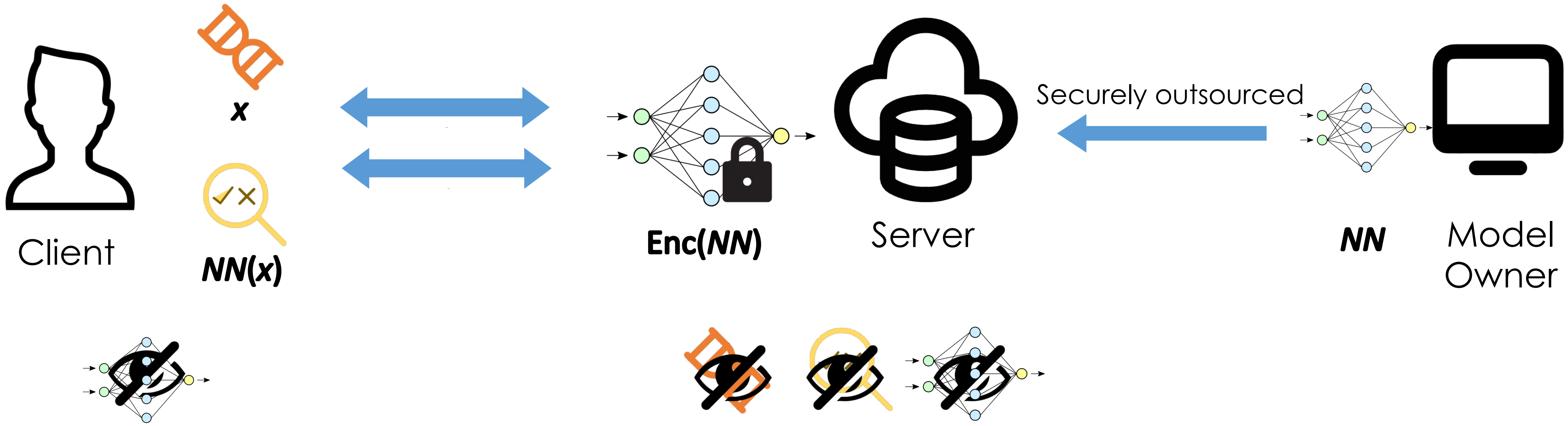


Intellectual property

Cannot learn either x or *NN*(x)

e.g., financial data, healthy status

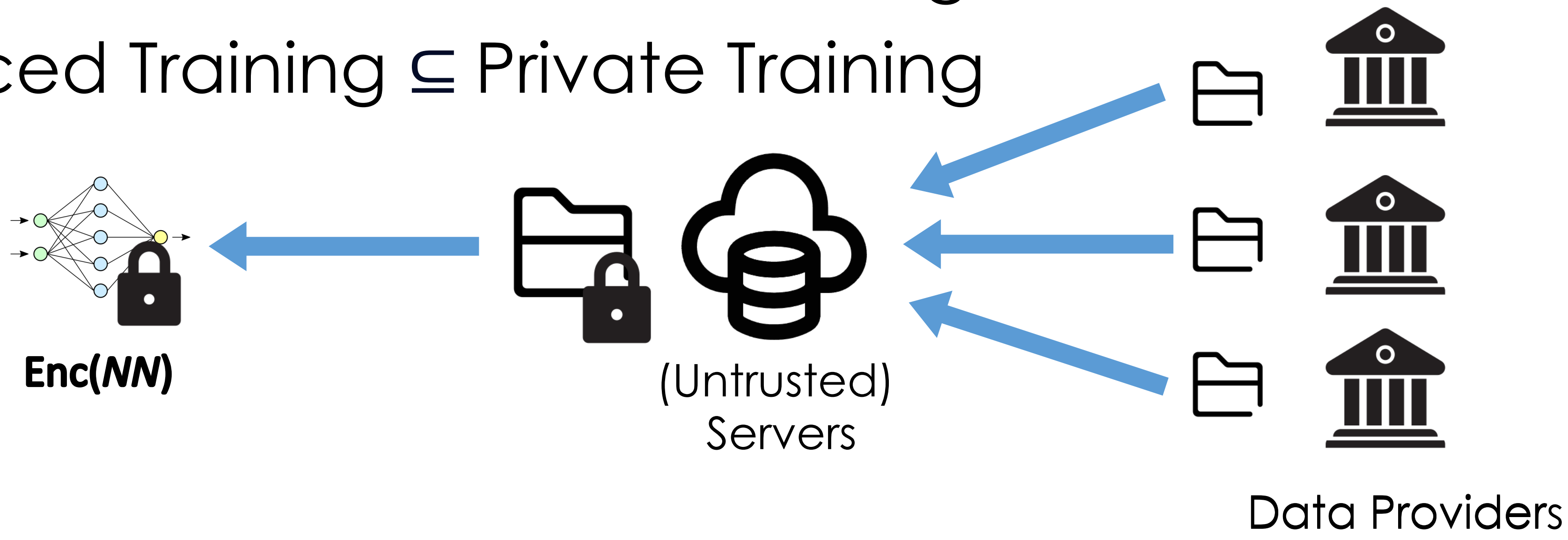
Outsourced Inference



- Oblivious Inference ⊆ Outsourced Inference

Outsourced/Private Training

- #Data Providers = 1 ⇒ Outsourced training
- #Data Providers ≥ 1 ⇒ Private training
- Outsourced Training ⊆ Private Training



- Outsourced Inference ⊆ Outsourced Training
  - Inference is a sub-routine of training

Our Goals

- Dissect the rapid development
  - (*e.g.*, the genealogy)
- Help newcomers dive right into the crux
- Avoid reinventing the wheel
- Highlight open problems and challenges
  - (This talk will briefly mention some)
- Aid in fair comparison

Out of Scope

Use of trusted hardware / trusted execution environment

- That's why “Cryptographic” in our title

Slalom [ICLR]
4.7×10^4 im/hr for ImageNet
with VGG-16

2020

2021

Inference

Training

Goten [AAAI]
2×10^5 im/hr on CIFAR-10
~90% acc
with VGG-16

❌ Membership inference, model inversion

- Orthogonal defenses

Florian Tramèr, Dan Boneh.
Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware.

Lucien K. L. Ng, Sherman S. M. Chow, Anna P. Y. Woo, Donald P. H. Wong, Yongjun Zhao.
Goten: GPU-Outsourcing Trusted Execution of Neural Network Training.

Out of Scope (cont.)

Differential privacy

- Different concepts of privacy
- Different research challenges
- *e.g.*, the curse of dimensionality in lang. model
  - [Du-Yue-Chow-Wang-Huang-Sun@CCS23]

Federated learning

- It leaks the models to the data providers
- Often uses "sum of PRF" techniques
  - [Naor-Pinkas-Reingold@EuroCrypt99]
  - [Chase-Chow@CCS09]
  - [Bonawitz et al.@CCS17]

Highlight of PPNN Development

Inference

Training

2016

CryptoNets [ICML]
300s for a query
on MNIST

2017

SecureML (2PC) [SP]
3.3×10^5 im/hr on MNIST
93% accuracy

2018

Gazelle [USS]
0.03s for MNIST
3.56s for <u>CIFAR-10</u>

(CIFAR-10 is a harder
visual dataset than MNIST)

2020

Trident (<u>4PC</u>) [NDSS]
4×10^7 im/hr on MNIST

2021

GForce [USS]
<u>0.3s</u> for CIFAR-10
with <u>VGG-16</u>

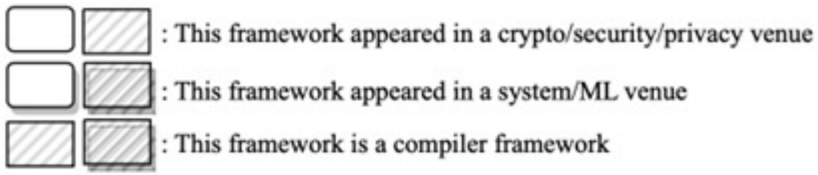
(VGG-16 has more linear layers
than those used before)

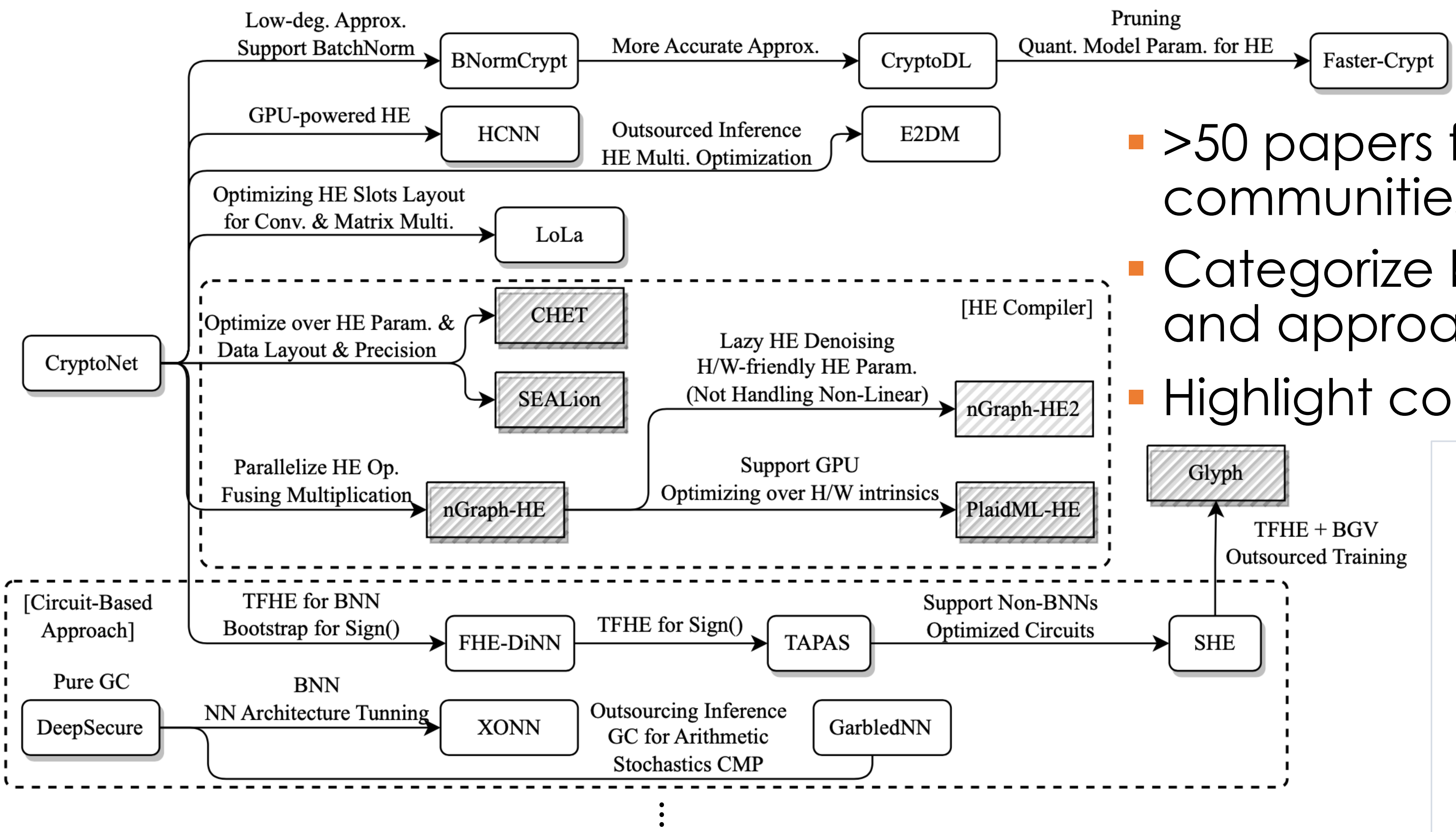
CryptGPU [SP]
4×10^4 im/hr on CIFAR-10
~84% acc
with <u>VGG-16</u>

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin E. Lauter, Michael Naehrig, John Wernsing. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy.

Payman Mohassel, Yupeng Zhang.
SecureML: A System for Scalable Privacy-Preserving Machine Learning.

Our Genealogy

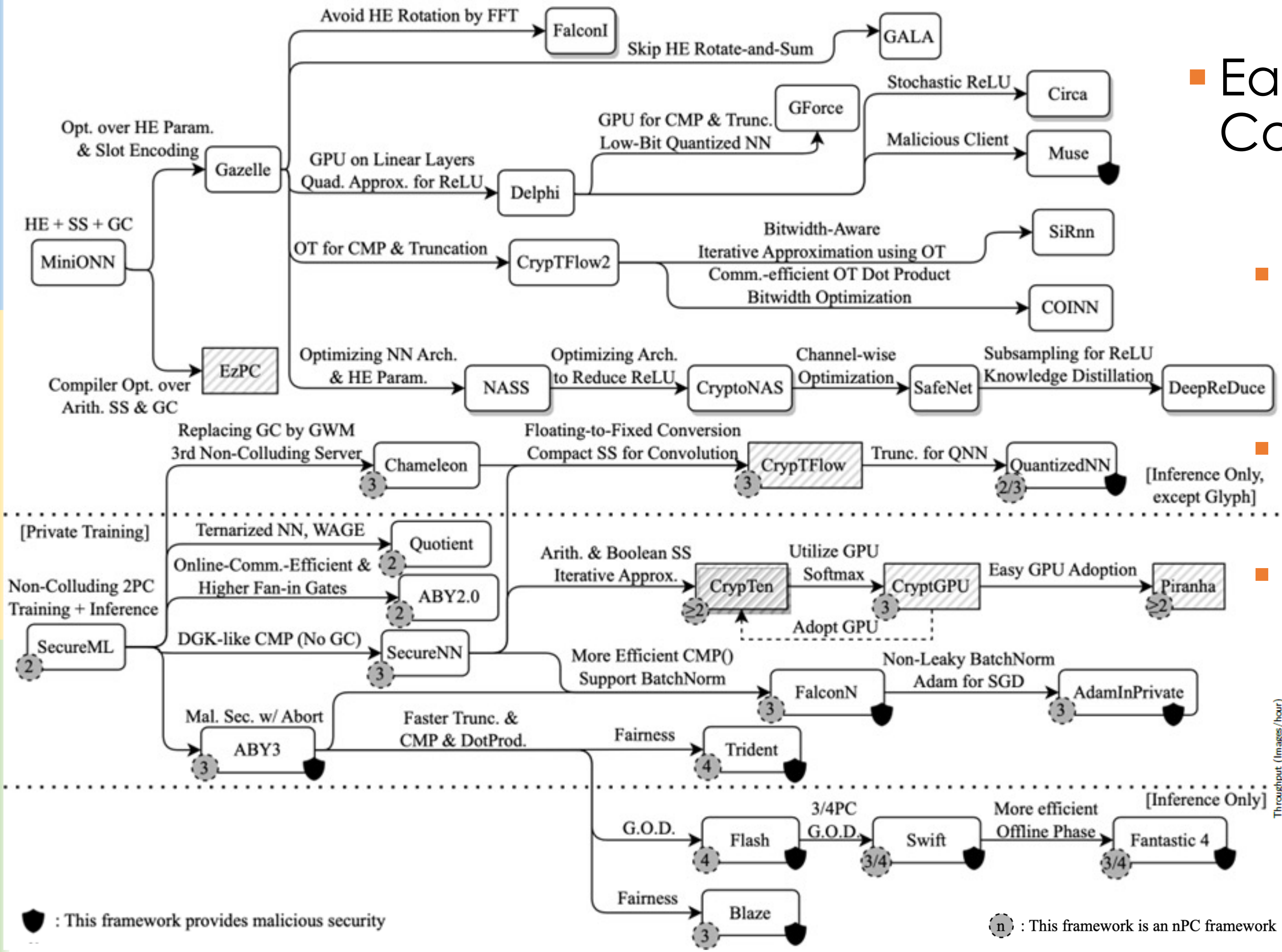




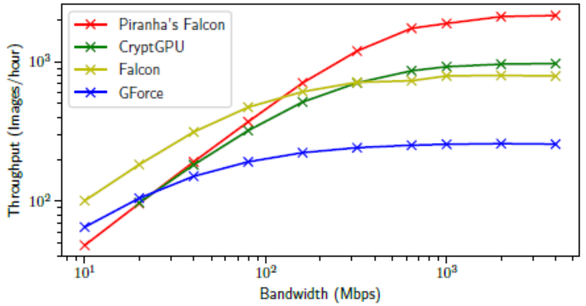
- >50 papers from different communities in 2016-22
- Categorize by their setting and approach
- Highlight contributions

(to be continued on the next slide)

Server-Client:
GForce [USS21]
COINN [CCS21]
CHET [PLDI19]
MPC:
CryptGPU [SP21]
FalconN [PETS21]
Pure-HE:
SHE [NeurIPS19]



- Easier and Fair Comparison
  - Non-colluding Assumption?
  - Benchmarks
  - Re-evaluation on WAN



Framework Type vs. Privacy Service

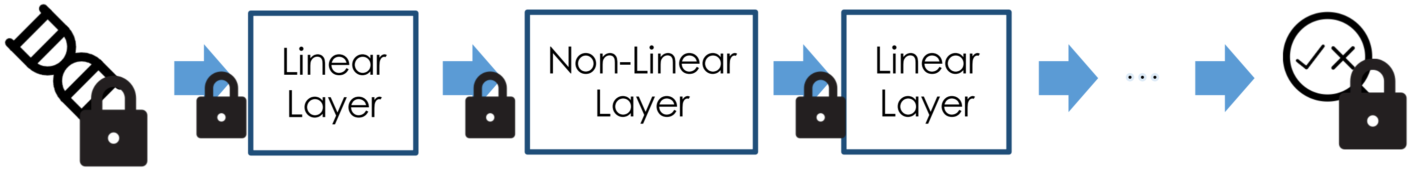
Framework Type	Oblivious Inference	Outsourced Inference	Outsourced /Private Training
Pure-LHE	✓	O	✗
Mixed	✓	✗	✗
MPC-based	✓	✓	O

✓: All frameworks support O: Only some support ✗: No framework supports

- LHE: Linear-Homomorphic Encryption; MPC: Secure Multi-party Computation
- Two other "less popular" framework types in our paper
  - Torus-based fully-homomorphic encryption
  - Pure garbled circuit

Paradigms for NN Computations

- Handle <u>linear layers</u> and <u>non-linear layers</u> *differently*



- Linear: *e.g.*, Convolution, Matrix Multiplication
  - Each output entry is an inner product of some input entries
  - Output $y_i = \sum_j w_j \cdot w_j$, where w_j and x_j are from the inputs

(I) Pure LHE for Oblivious Inference

- Client: secret key holder, can decrypt [x] into the result x
- Server: owner of model w

- Linear Layers: $[y_i] = \sum_j w_j \cdot [x_j]$
  - [x] denotes encryption of x

- ML technique helps
  - Pruning sets some small model parameters w_j to 0
  - Server can skip computing $w_j \cdot [x_j]$

Non-linear Layers in Pure-LHE frameworks

- Activation: "Simple" ones via Polynomial Approximation
  - $[y_i] = a_0 + a_1 \, [x_i] + a_2 \cdot [x_i] \cdot [x_i] + \cdots$
  - Approximation degrades accuracy

- Pooling: "Simple" Average Pooling
  - (additions with one division)
  - (non-linear) Max pooling usually gives higher accuracy

Bitwidth Issue

- Plaintext NN operates in floating point (numbers)
  - a much wider range than $\mathbf{Z}_q$, *i.e.*, integers, for [x]
  - 256-bit to represent a floating point
- High bitwidth → larger HE parameters → worse performance in LHE
- "*privately, efficiently, & accurately evaluate layers in low bitwidth?*"
- "*cater dynamic weights in secure training?*"
- "*guide non-cryptographers to select "tight" HE parameters?*"
  - Compilers (Sec VIII.B)

(II) Mixed Frameworks

- Solving 2 issues in pure-LHE frameworks
  - 1st issue: LHE computation is slow
- Use additive sharing
  - addition over shares
- Each op costs just a few CPU instructions
  - >100× faster than LHE ops (ignoring communication)
- Multiplications need pre-processing (*e.g.*, by LHE) in offline time
  - Online: The query became known, use pre-computed results

Comparison (CMP) in Non-linear Layer

- 2nd Issue: Polynomial approximation harms accuracy
- In many non-linear layers, *comparison* ($x \le y$) is a fundamental operation
  - $\mathrm{ReLU}(x) = \mathrm{Max}(x, 0)$, $\mathrm{Maxpool}(\{x\}_{0..3}) = \mathrm{Max}(x_0, x_1, x_2, x_3)$

- NN Architecture Search (*Delphi* [USS20])
  - Approx. only some CMP

Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, Raluca Ada Popa. Delphi: A Cryptographic Inference Service for Neural Networks.

- Use GPU to securely compute "linearized" CMP (*GForce* [USS21])
  - >30× faster than garbled circuit

Lucien K. L. Ng, Sherman S. M. Chow.
GForce: GPU-Friendly Oblivious and Rapid Neural Network Inference.

- *"How to implement an even more efficient CMP?"*
- *"What can other crypto primitives be made GPU/TPU-friendly?"*

(III) Non-Colluding Assumption

- Servers that will not reveal their secret to any other parties
- *m*PC framework assumes *m* non-colluding servers
- 3PC frameworks make a stronger assumption than 2PC ones
  - A server only needs to compromise 1 among 2 others instead of a fixed 1
- More servers, higher throughput
  - 3rd server can prepare Beaver's triplets
  - if only 2 servers, they need to interact
- Training needs millions of iteration of inferences

Framework	#	Guarantee
SecureML [13], Quotient [74], ABY2.0 [84]	2	–
CrypTen [77], Piranha [109]	≥2	–
QuantizedNN [72]	2/3	Abort
Chameleon [86], CrypTFlow [107]	3	–
CryptGPU [51]	3	–
ABY3 [88], SecureNN [68]	3	Abort
FalconN [69], AdamInPrivate [90]	3	Abort
Blaze [76]	3	Fair
Swift [89], Fantastic 4 [85]	3/4	G.O.D.
Flash [75]	4	G.O.D.
Trident [50]	4	Fair
GarbledNN [64], XONN [63]	–	Abort
Muse [114]	–	Client

Complex Function Evaluation

- BatchNorm can be reduced to 1 / sqrt(x) over secret x
- Softmax can be reduced to x / y and e^x over secret x & y

- "*How to efficiently & accurately approximate*
  x / y, 1 / sqrt(y), e^x, sigmoid(x), and tanh(x)
  for secret x and y?"

- "*How to realize <u>high throughput</u> and <u>accurate</u> private training without non-colluding assumptions?*"

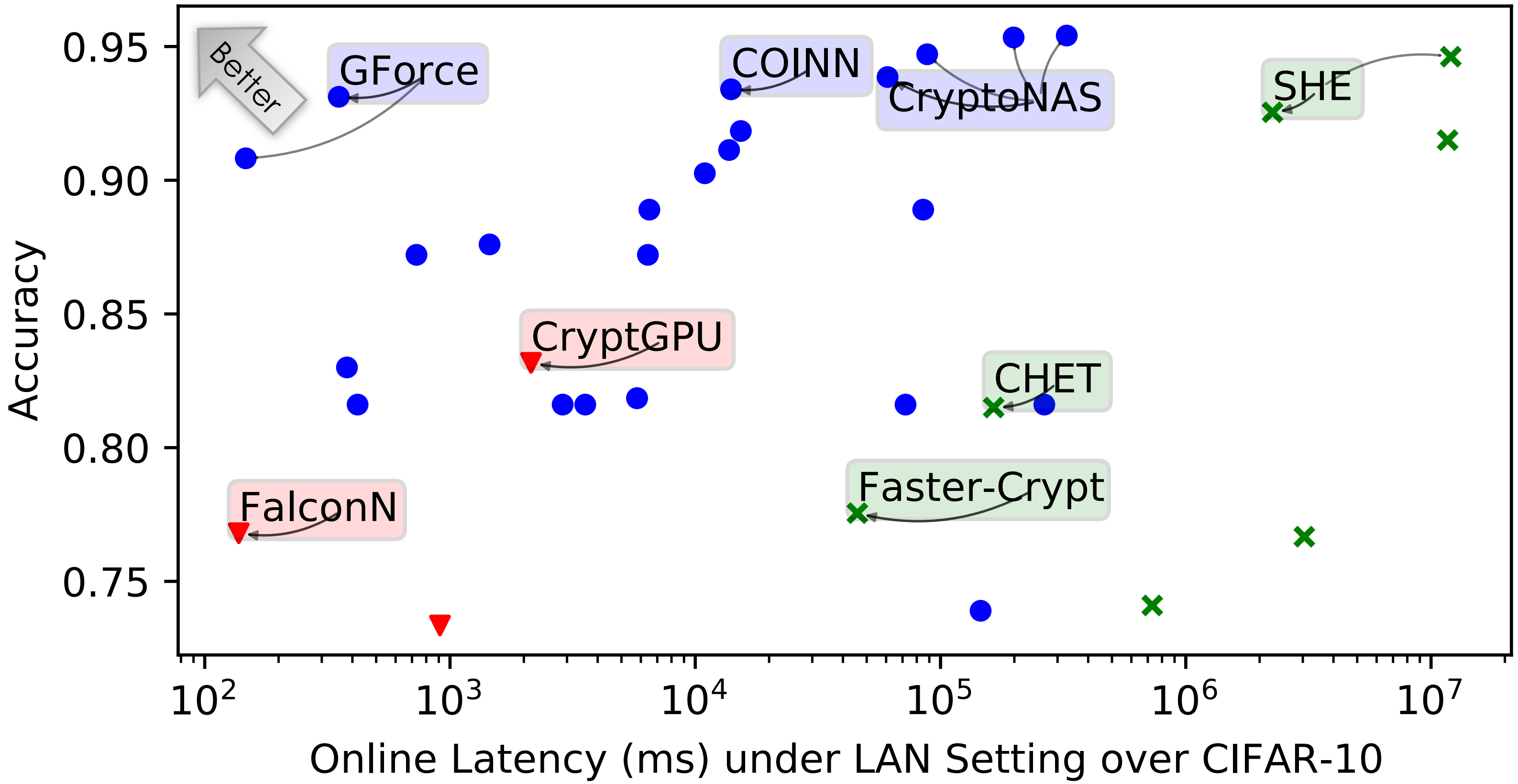
Framework Summary (▽: Oblivious Inference, ▼: Outsourced Inference, □: Outsourced Training, ■: Private Training)

	Framework	Basic Info.		Fixed-Point				Non-Linear			Optimization						Datasets				Crypto Tools			
		Reference	Year	Privacy Service	Trunc. & Wrap	Bitwidth	B/QNN	Poly. Approx.	CMP	Num. Method	Offline/Online	HE SIMD	Dyna. Weights	GPU	Optimize Arch.	Compiler	MNIST	CIFAR-10	CIFAR-100	ImageNet	GC/GMW	OT	SS	HE
Pure-HE	CryptoNets	[11]	16	▽	○	H	-	●	○	-	○	●	○	○	○	○	●	○	○	○	○	○	○	L
	BNormCrypt	[53]	17*	▽	○	H?	-	●	○	-	○	◐	○	○	○	○	●	○	○	○	○	○	○	L
	CryptoDL	[54]	17	▽	○	H?	-	●	○	-	○	◐	○	○	○	○	●	●	○	○	○	○	○	L
	Faster-Crypt	[55]	18*	▽	○	H	Q	●	○	-	○	◐	○	○	●	○	●	●	○	◐	○	○	○	L
	HCNN	[56]	21	▽	○	L	-	◐	○	-	○	◐	○	●	○	○	●	●	○	○	○	○	○	L
	E2DM	[57]	18	▼	○	H	-	◐	○	-	○	●	○	○	○	○	●	○	○	○	○	○	○	L
	nGraph-HE	[102]	19	▽	○	H	-	◐	○	-	○	◐	○	○	○	●	●	●	○	○	○	○	○	L
	nGraph-HE2	[105]	19	▽	○	H	-	○	○	-	○	◐	○	○	○	●	●	○	○	◐	○	○	○	L
	PlaidML-HE	[106]	19	▽	○	H?	-	◐	○	-	○	◐	○	●	○	●	○	○	○	○	○	○	○	L
Non-Colluding MPC	CrypTFlow	[107]	20	▼	●	H	-	○	◐	-	○	○	○	○	○	●	◐	◐	○	●	●	○	●	-
	ABY3	[88]	18	■	●	H	-	○	●	-	●	○	○	○	○	○	●	○	○	○	●	○	●	-
	Flash	[75]	20	▼	●	H	-	○	●	-	●	○	○	○	○	○	●	○	○	○	○	○	●	-
	Blaze	[76]	20	▼	●	H	-	○	●	-	●	○	○	○	○	○	○	○	○	○	○	○	●	-
	Swift	[89]	21	▼	●	H	-	○	●	-	●	○	○	○	○	○	●	◐	○	○	○	○	●	-
	Trident	[50]	20	■	●	H	-	○	●	-	●	○	○	○	○	○	●	○	○	○	○	○	●	-
	Fantastic 4	[85]	21	■	●	H	-	○	●	-	○	○	○	○	○	○	●	○	○	◐	○	○	●	-
	QuantizedNN	[72]	20	▼	●	H	Q	○	◐	-	○	○	○	○	○	○	○	◐	○	●	○	○	●	-
	AdamInPrivate	[90]	22	■	●	L	Q	○	○	I	●	○	●	○	○	○	●	●	○	○	○	○	●	-
	SecureNN	[68]	19	■	●	H	-	○	●	-	●	○	●	○	○	○	●	○	○	○	○	○	●	-
	FalconN	[69]	21	■	●	L	-	○	●	I	●	○	●	○	○	○	●	●	○	○	○	○	●	-
	CrypTen	[77]	21	■	●	H	-	○	◐	I	◐	○	○	◐	○	●	◐	○	○	◐	●	●	●	-
	CryptGPU	[51]	21	■	◐	H	-	○	◐	I	○	○	●	●	○	○	●	●	○	○	●	○	●	-
	Piranha	[109]	22	■	◐	H	-	○	◐	-	◐	○	○	●	○	●	●	●	○	○	○	○	●	-

H: >32-bit, L: ≤32, M: mixed, ?: unspecified; I: iterative, T: table lookup; L: LHE, T: TFHE; * marks the earliest appearance of the preprint;
◐: adopter of existing techniques / w/o acc. or end-to-end inf. results; ●: original contributor / with performance & accuracy;

(IV) Performance Evaluation

- *Mixed* frameworks minimize online *inference latency* on LAN



Server-Client:
GForce [USS21]
COINN [CCS21]
CryptoNAS [NeurIPS20]
CHET [PLDI19]

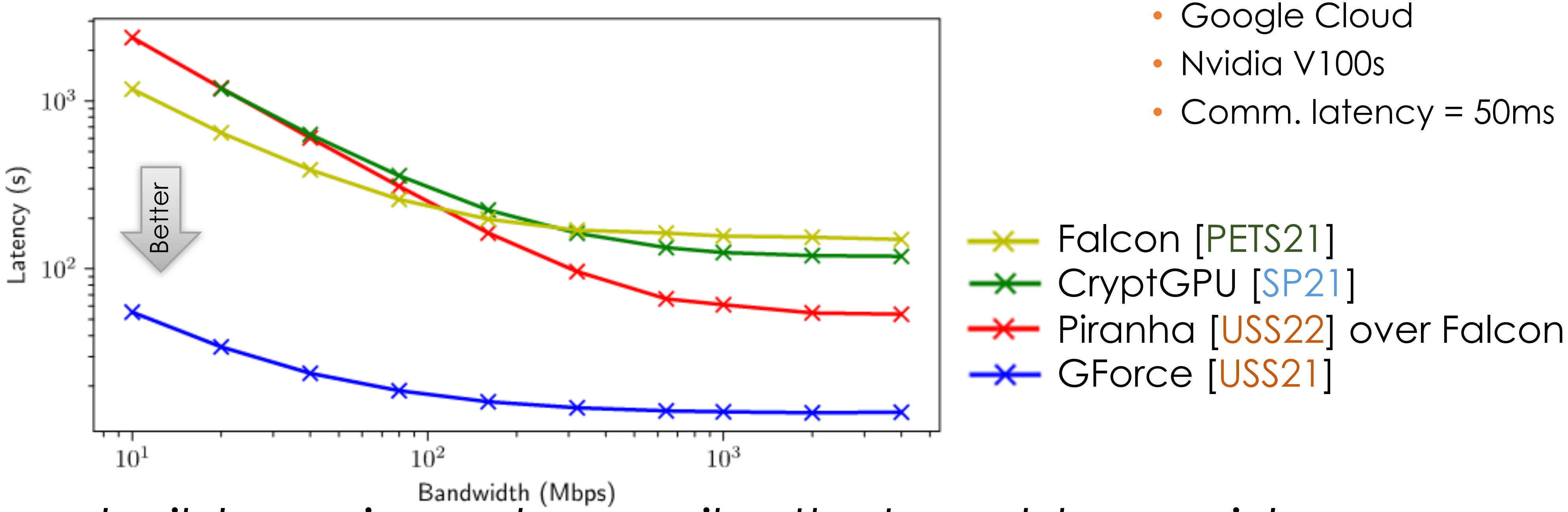
MPC (non-colluding servers):
CryptGPU [SP21]
FalconN [PETS21]

Pure-HE:
SHE [NeurIPS19]

Throughput: See our paper

Re-evaluation on WAN

- Run the state-of-the-art frameworks on the same hardware



- Google Cloud
- Nvidia V100s
- Comm. latency = 50ms

- *"Can we build a universal compiler that enables rapid prototyping and allow uniform experimental comparison?"*

Full Version: sokcryptonn.github.io

- More Details on Cryptography
- Interactive Charts and Genealogy
- Update to include new works
  - Contact us if you feel we missed your work!

luciengkl@gatech.edu | sherman@ie.cuhk.edu.hk