Truth Table Net: A Neural Network for Torus-FHE

A. Benamira, T. Guérand, T. Peyrin, S. Saha

Nanyang Technological University

27th April 2023

First, thank you for the invitation.

Adrien BENAMIRA

- Engineering & Applied Mathematics - University Paris Saclay
- NTU PhD student since 2020 supervised by Thomas Peyrin
- Research focus:
  1. Cryptanalysis & Deep Learning (DL)
  2. DL reliability

Deep Neural Networks (DNNs): 2012-2023

In $\approx$ 10 years:

Deep Neural Networks (DNNs): 2012-2023

In $\approx$ 10 years:

- Image
- Text
- Reinforcement learning
- Generation (image & text)

Deep Neural Networks (DNNs): 2012-2023

In $\approx$ 10 years:

- Image
- Text
- Reinforcement learning
- Generation (image & text)

Possible thansk to:

- Cloud/GPU computing
- Open-source datasets
- Large community (researchers + investors/companies)

Deep Neural Networks (DNNs): 2012-2023

In $\approx$ 10 years:

- Image
- Text
- Reinforcement learning
- Generation (image & text)

Possible thansk to:

- Cloud/GPU computing
- Open-source datasets
- Large community (researchers + investors/companies)

$\rightarrow$ But what about trust ?

DNNs Challenges in 2023

DNNS reliability & trustworthiness are poor:

DNNs Challenges in 2023

DNNS reliability & trustworthiness are poor:

1 - Interpretability

- Black box

DNNs Challenges in 2023

DNNS reliability & trustworthiness are poor:

1 - Interpretability

- Black box

2 - Formal Verification

- No proven guarantees for properties (fairness, robustness)

DNNs Challenges in 2023

DNNS reliability & trustworthiness are poor:

1 - Interpretability

- Black box

2 - Formal Verification

- No proven guarantees for properties (fairness, robustness)

3 - Privacy

- In Cloud settings
- No trust between user and model provider

DNNs Challenges in 2023

DNNS reliability & trustworthiness are poor:

1 - Interpretability

- Black box

2 - Formal Verification

- No proven guarantees for properties (fairness, robustness)

3 - Privacy

- In Cloud settings
- No trust between user and model provider

→ Critical industries: Military, Energy, Finance, Healthcare - 

DNNs Privacy in Cloud

DNNs Privacy in Cloud

- Fully Homomorphic Encryption (FHE)
- Guarantee data user privacy (GDPR - CCPA)
- Limits?

DNNs Privacy in Cloud

- Fully Homomorphic Encryption (FHE)
- Guarantee data user privacy (GDPR - CCPA)
- Limits?

Table: 10K different clients, 1 CIFAR-10 img./client with FHE/HE

DNNs Privacy in Cloud

- Fully Homomorphic Encryption (FHE)
- Guarantee data user privacy (GDPR - CCPA)
- Limits?

Table: 10K different clients, 1 CIFAR-10 img./client with FHE/HE

Method FHE/HE	Method ML	Cost 512GB - 3.6$/hour	Accuracy	Time 1core /client	Memory /client
CKKS	Resnet	≈18K$	92.5%	38min	384GB
	Lola	≈1.7K$	74.1 %	97min	12GB
TFHE	General DNN	≈53K$	62.3%	62h	14GB

DNNs Privacy in Cloud

- Fully Homomorphic Encryption (FHE)
- Guarantee data user privacy (GDPR - CCPA)
- Limits?

Table: 10K different clients, 1 CIFAR-10 img./client with FHE/HE

Method FHE/HE	Method ML	Cost 512GB - 3.6$/hour	Accuracy	Time 1core /client	Memory /client
CKKS	Resnet	≈18K$	92.5%	38min	384GB
	Lola	≈1.7K$	74.1 %	97min	12GB
TFHE	General DNN	≈53K$	62.3%	62h	14GB

→ How to decrease cost for production ?

AI Privacy in Cloud with FHE

- How to decrease cost ?
- 2 **complementary** strategies

AI Privacy in Cloud with FHE

- How to decrease cost ?
- 2 **complementary** strategies

Strategy 1 : enhance FHE technology for ML

- FHE.org & Zama

AI Privacy in Cloud with FHE

- How to decrease cost ?
- 2 **complementary** strategies

Strategy 1 : enhance FHE technology for ML

- FHE.org & Zama

Strategy 2 : enhance ML technology for FHE

- Our approach

AI Privacy in Cloud with FHE

- How to decrease cost ?
- 2 **complementary** strategies

Strategy 1 : enhance FHE technology for ML

- FHE.org & Zama

Strategy 2 : enhance ML technology for FHE

- Our approach

GOAL : Design a Neural Network for Torus-FHE

AI Privacy in Cloud with FHE

- How to decrease cost ?
- 2 **complementary** strategies

Strategy 1 : enhance FHE technology for ML

- FHE.org & Zama

Strategy 2 : enhance ML technology for FHE

- Our approach

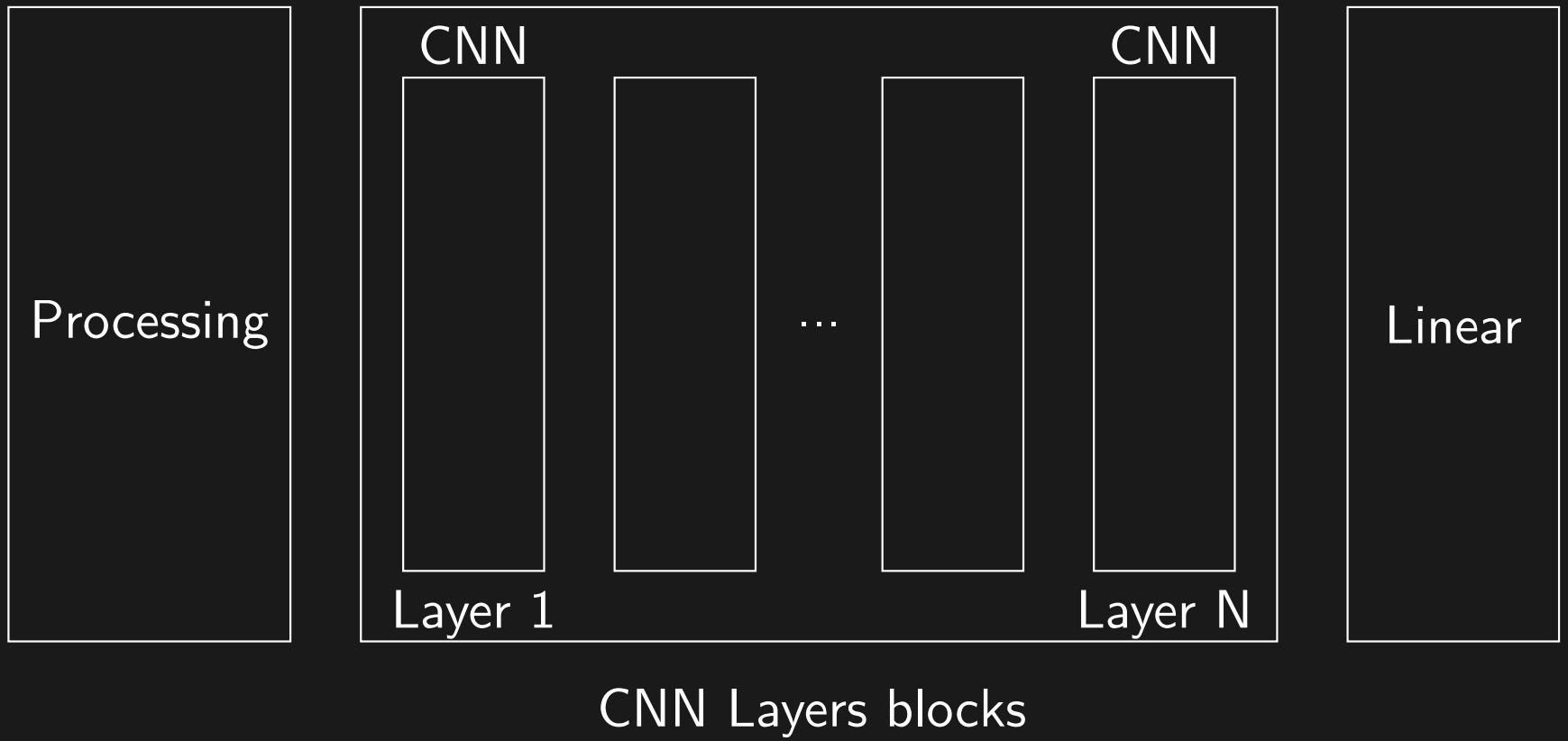
GOAL : Design a Neural Network for Torus-FHE

→ Achieved with Truth Table-based approaches

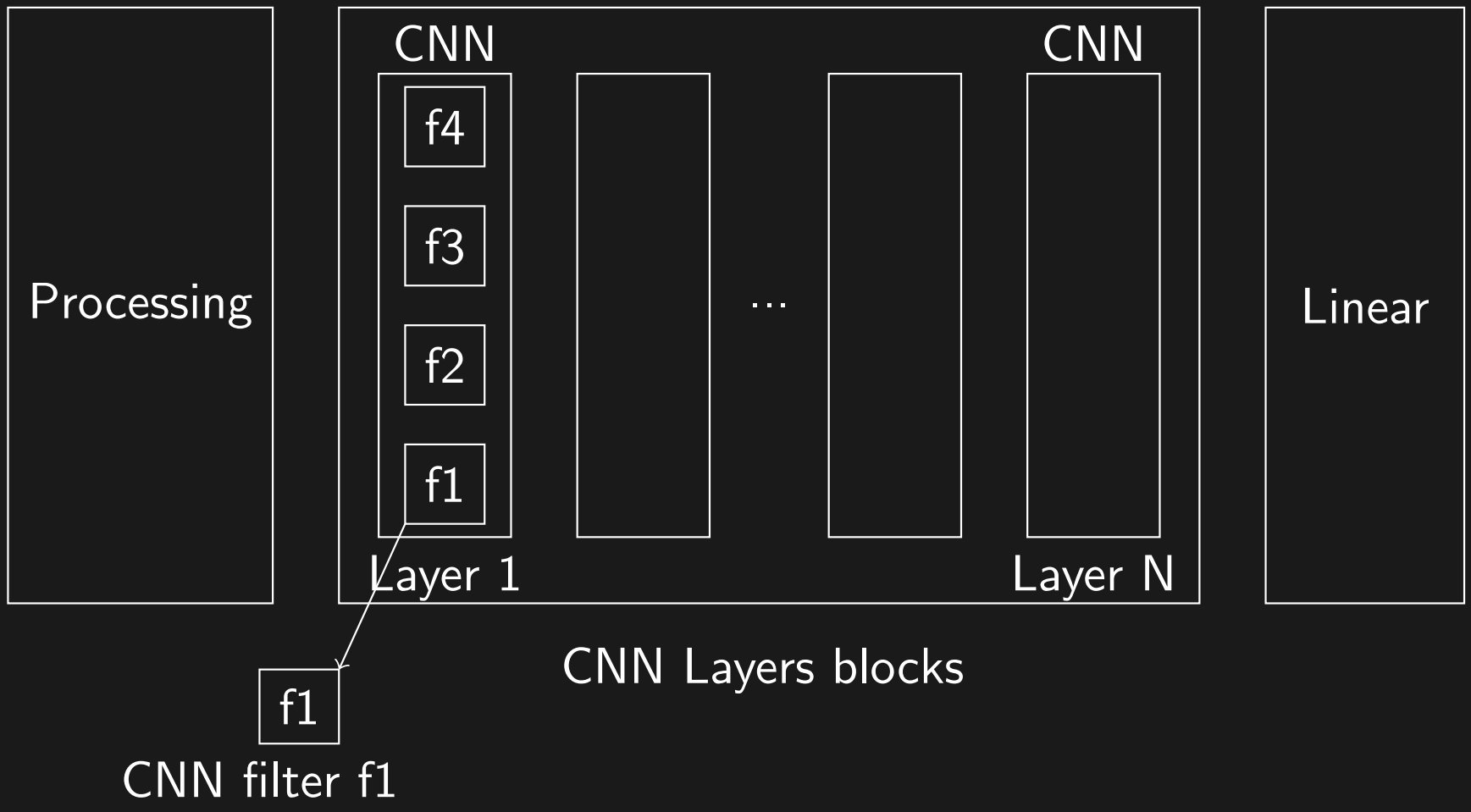
Can we invent Deep Convolutional Neural Net (DCNN) for Torus-FHE technology ?

DCNN overview

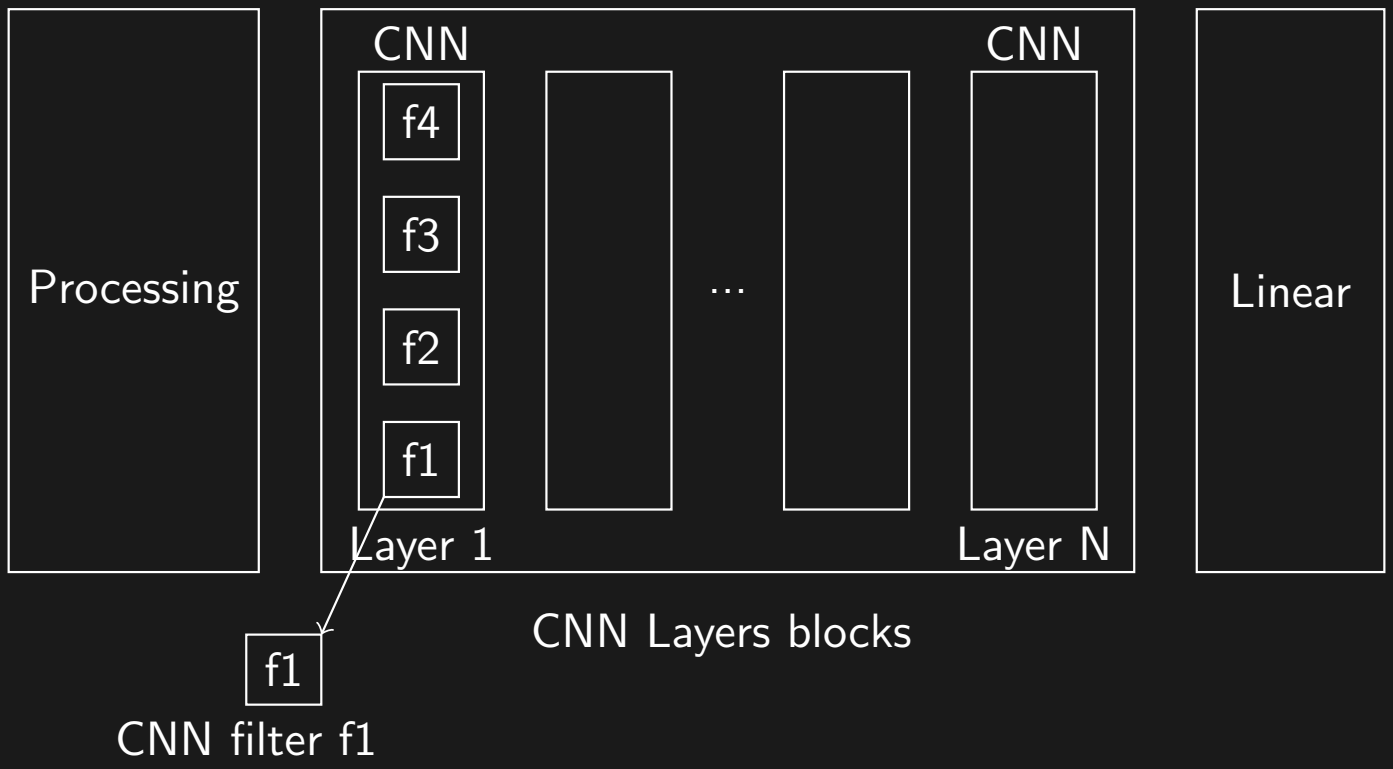
DCNN overview



DCNN overview



DCNN overview



GOAL : Transform each CNN filter into a truth table

Can we invent Deep Convolutional Neural Net (DCNN) for Torus-FHE technology ?

$$\text{Design a DCNN} \iff \begin{cases} \text{Preprocessing layer (client side)} \\ \text{Interconnected truth tables} \iff \text{CNNs} \\ \text{Linear classification layer (4-bit)} \end{cases}$$

$$\rightarrow \text{Truth Table Net} = \text{TTnet}$$

$$\text{LTT} \iff \text{Look Up Table}$$

$\rightarrow$ TTnet is 100% compatible with TFHE Zama technology

2. Truth Table Net
   - LTT Design
     - General
     - Tractability Exhaustive Property
   - TTnet Design
   - Performances

Key features of Learnable Truth Table (LTT) block

Key features of Learnable Truth Table (LTT) block

- Equivalent to a **enumerable Boolean function**

Key features of Learnable Truth Table (LTT) block

- Equivalent to a **enumerable Boolean function**
- **Complete distribution computable in 2^n operations**, with $n \leq 6$, independantly of the architecture

Key features of Learnable Truth Table (LTT) block

- Equivalent to a **enumerable Boolean function**
- **Complete distribution computable in 2^n operations**, with $n \leq 6$, independantly of the architecture
- Design based on a **neural network** for ease of training

Key features of Learnable Truth Table (LTT) block

- Equivalent to a **enumerable Boolean function**
- **Complete distribution computable in 2^n operations**, with $n \leq 6$, independantly of the architecture
- Design based on a **neural network** for ease of training
- **Scalable and accurate** when applied to large datasets

Design rules for Learnable Truth Table (LTT)

Design rules for Learnable Truth Table (LTT)

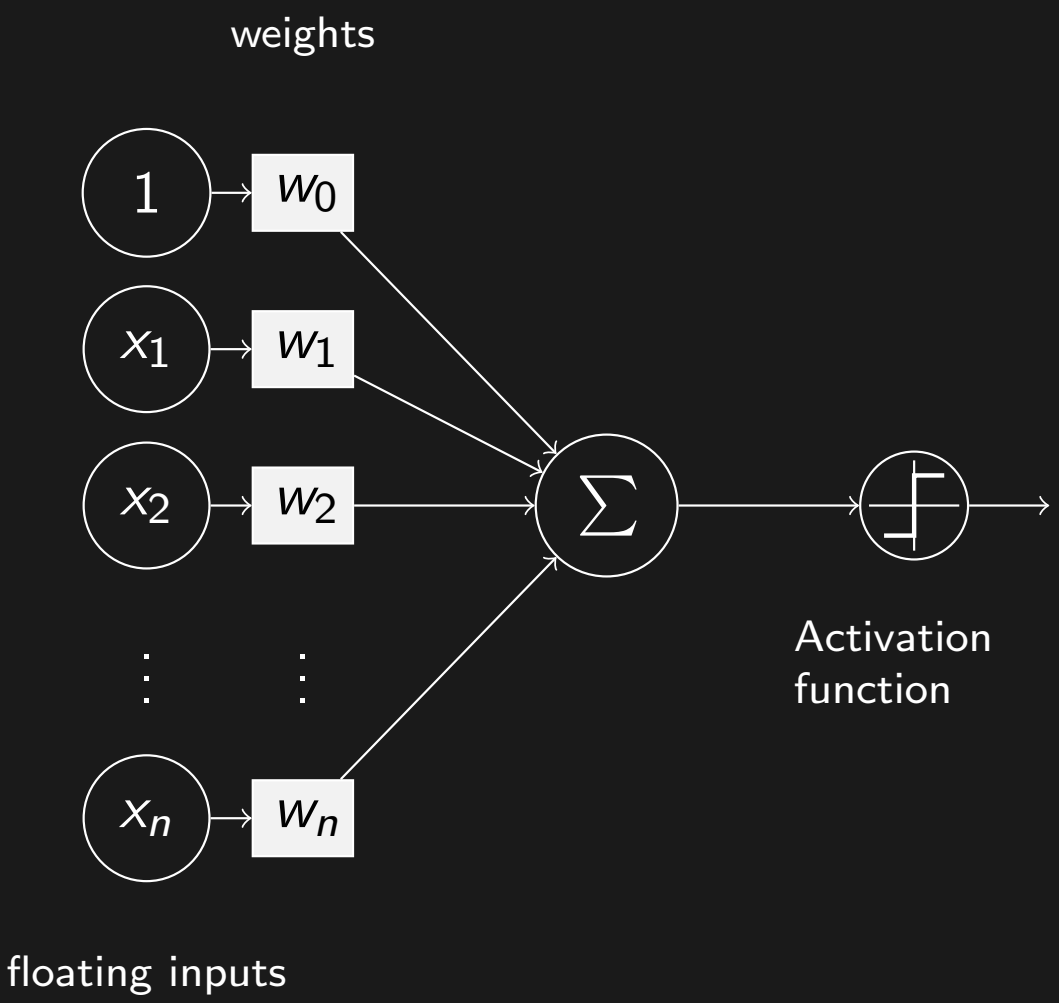
- Equivalent to a **small exhaustive Boolean function**
  1. LTT inputs/outputs are binary
  2. LTT input size is "small" : $n \leq 6$

Design rules for Learnable Truth Table (LTT)

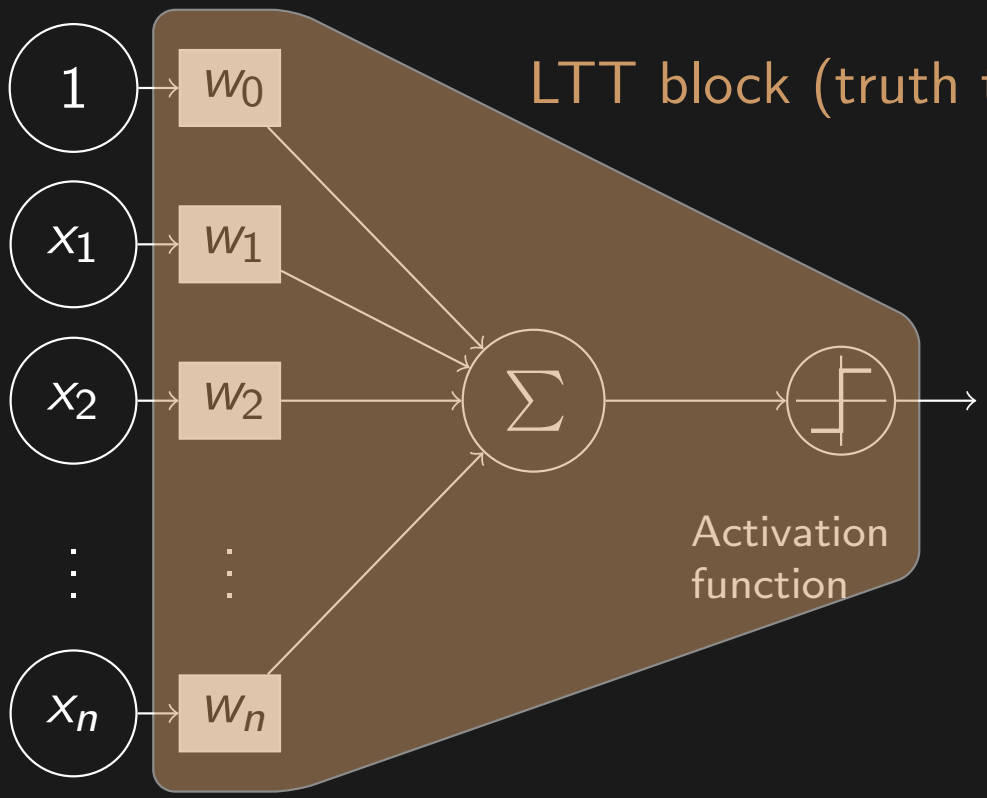
- Equivalent to a **small exhaustive Boolean function**
  1. LTT inputs/outputs are binary
  2. LTT input size is "small" : $n \leq 6$

- **Scalable and accurate** when applied to large datasets
  1. LTT is nonlinear in between the step functions

Here is a Classic perceptron



LTT General idea



n-bit binary input

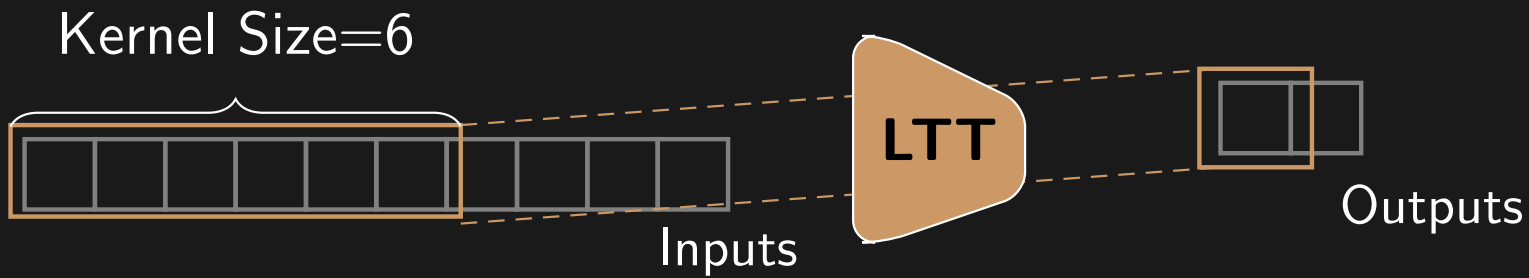
$\rightarrow$ the main issue is to reduce n while keeping high performances

Focus on CNN filters

- Reducing n for all functions is impossible
- Focus on the well used CNNs tool
- CNN filter has, by design, small size inputs

Animation video

1D-CNN filter / LTT block with 1-channel input



Truth Table, 2^6					
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
...	...	...	...	...	...
1	1	1	1	1	0
1	1	1	1	1	1

LTT
1
1
0
...
0
1

- LTT block / CNN filter transformation into a truth table

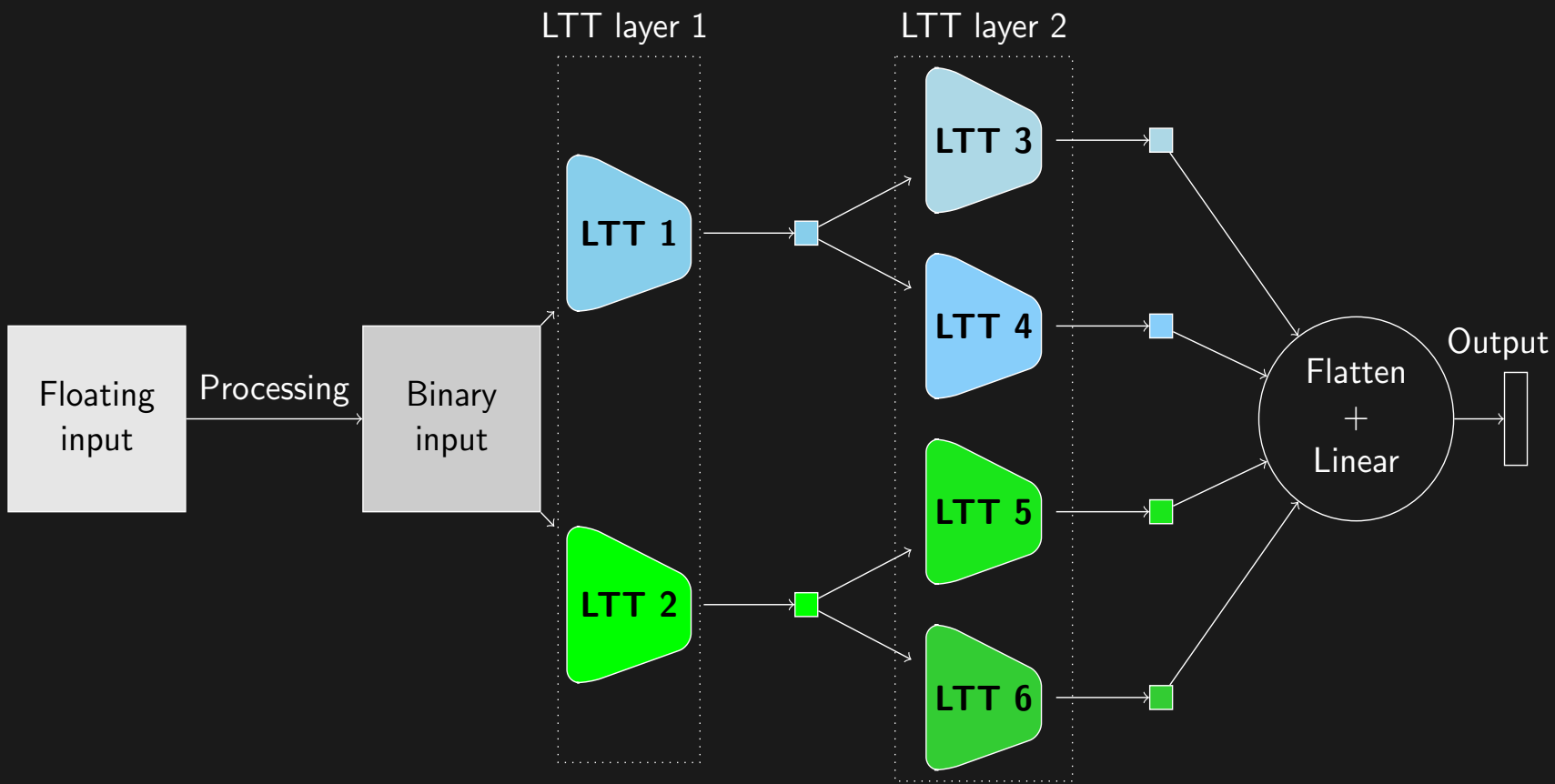
Rules Design for Learnable Truth Table (LTT)

- Equivalent to a **small exhaustive Boolean function** ✓
  1. LTT inputs/outputs are binary ✓
  2. LTT input size is "small" : $n \leq 6$ ✓

Rules Design for Learnable Truth Table (LTT)

- Equivalent to a **small exhaustive Boolean function** ✓
  1. LTT inputs/outputs are binary ✓
  2. LTT input size is "small" : $n \leq 6$ ✓
- **Scalable and accurate** when applied to large datasets
  1. LTT is nonlinear in between the step functions

Rules Design for TTnet



LTT Design ✓

- LTT inputs/outputs are binary
- LTT input size is "small" : $n \leq 6$
- Non linear LTT function to reach scalability

TTnet Design ✓

CNN filter $\iff$ LTT $\iff$ truth table of size n with $n \leq 6$

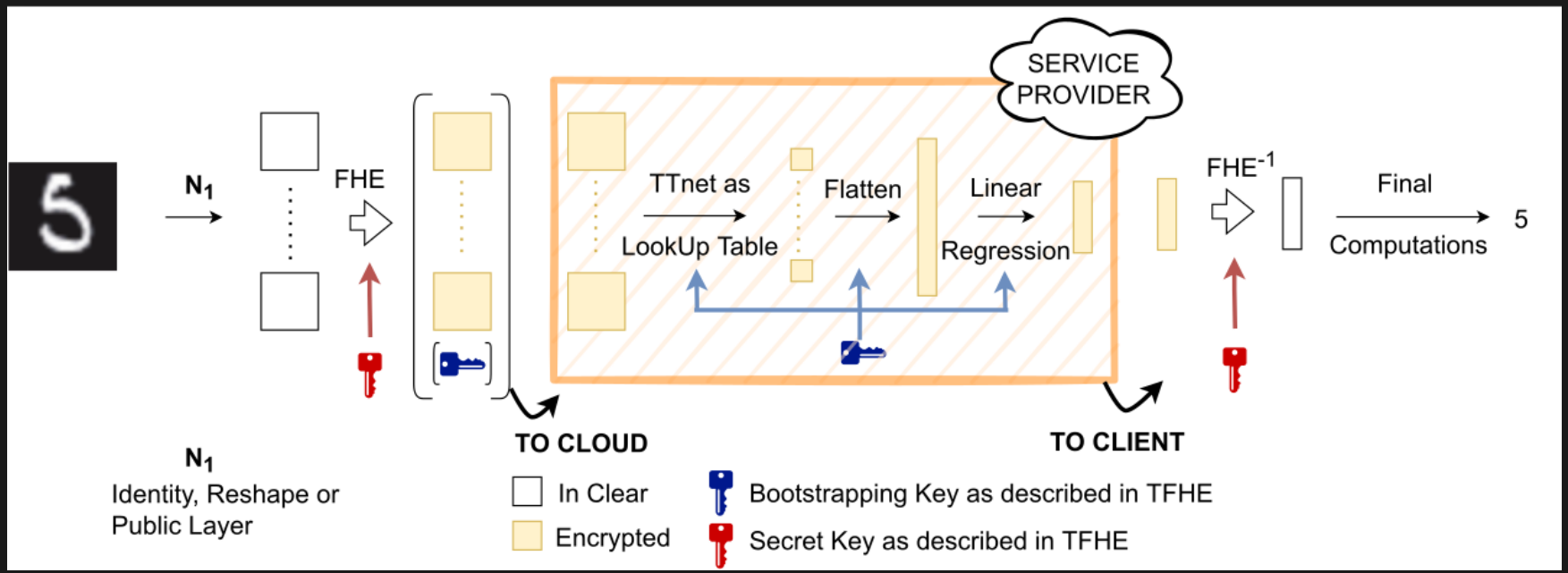
$$\text{TTnet} \iff \begin{cases} \text{Preprocessing layer} \\ \text{Interconnected truth tables} \iff \text{CNNs} \\ \text{Linear classification layer (sparse binary layer)} \end{cases}$$

What are the accuracies on CIFAR10 & ImageNET?

Dataset	n	Accuracy
MNIST	6	98%
CIFAR-10	6 - 25	70% - 87%
ImageNet	16	45% (Top-1)

3. Results
   - TFHE : TTnet for scaling Torus FHE with concrete
   - Tabular datasets
   - Image datasets

General: $n \leq 8$ and 1 mutihead E-AE LTT layer



$\rightarrow$ Trade-off between accuracy & FHE running time performances.

Tabular datasets: fully private

	FHE family	#CPU **cores**	**Adult**		**Cancer**		**Diabetes**	
			Acc.	Time	Acc.	Time	Acc.	Time
ETHZ/CCS22	CKKS	64	81.6%	420s	-	-	-	-
TAPAS	TFHE	16	-	-	97.1%	3.5s	54.9%	250s
Ours + Zama		4	85.3%	5.6s	97.1%	1.9s	57.0%	1.2s

Tabular ADULT datasets: Zama comparison

Model	Xgboost - Zama			Ours + Zama	
	6bits	4bits	2bits	Big	Small
Accuracy	86.0%	85.6%	75.6%	85.3%	83.8%
FHE Time	142s	40s	8.7s	5.6s	0.5s
Memory	15MB	3MB	1.4MB	3.4MB	3.4MB
Cost (100K clients)	$\approx$ 10$	$\approx$ 4$	$\approx$ 4$	$\approx$ 4$	$\approx$ 4$

$\rightarrow$ Ready for production

$\rightarrow$ Bonus : TTnet offers global & exact interpretability

Image datasets

Table: 10K different clients, 1 CIFAR-10 img./client with FHE/HE

Method FHE/HE	Method ML	Cost 512GB - 3.6$/hour	Accuracy	Time /client	Memory /client
CKKS	Resnet	≈18K$	92.5%	38min	384GB
	Lola	≈1.7K$	74.1 %	97min	12GB
TFHE	General DNN	≈53K$	62.3%	62h	14GB
TFHE	TTnet BIG	≈60$	74.1%	38min	0.8GB
	TTnet SMALL	≈7$	62.2%	38min	0.08GB

Image datasets

Table: 10K different clients, 1 CIFAR-10 img./client with FHE/HE

Method FHE/HE	Method ML	Cost 512GB - 3.6$/hour	Accuracy	Time /client	Memory /client
CKKS	Resnet	≈18K$	92.5%	38min	384GB
	Lola	≈1.7K$	74.1 %	97min	12GB
TFHE	General DNN	≈53K$	62.3%	62h	14GB
TFHE	TTnet BIG	≈60$	74.1%	38min	0.8GB
	TTnet SMALL	≈7$	62.2%	38min	0.08GB

→ (Almost) Ready to production

- Conclusion
  - Future works
  - Take home message

Future works for TTnet & FHE

- TTnet preprocessing at the **service provider side**
- TTnet as a Boolean circuit with **Rust**
- TTnet **FHE training** on tabular datasets
- TTnet HE with **CKKS & OpenFHE**
- **Imagenet & NLP** with TTnet FHE

Conclusion : Take-home message

- A new neural network based on **truth table/ Look-Up Table**
- FHE + TTnet ready to production for tabular dataset, almost ready for image dataset
- TTnet already relevant for **multiple applications**

- We have positions for PhD + PostDocs
- Partnership with Industry is more than welcome
- TTnet is under **NTU Licensing**

Thank you for your attention

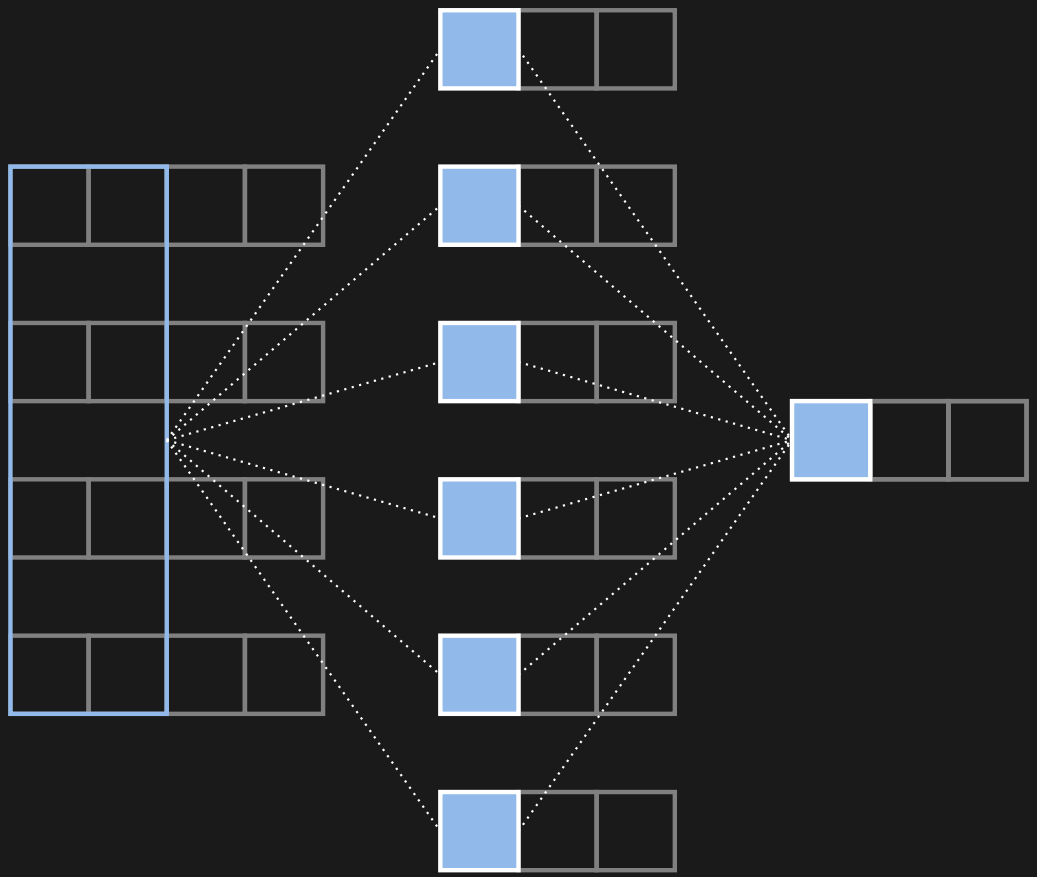
Do you have any question?

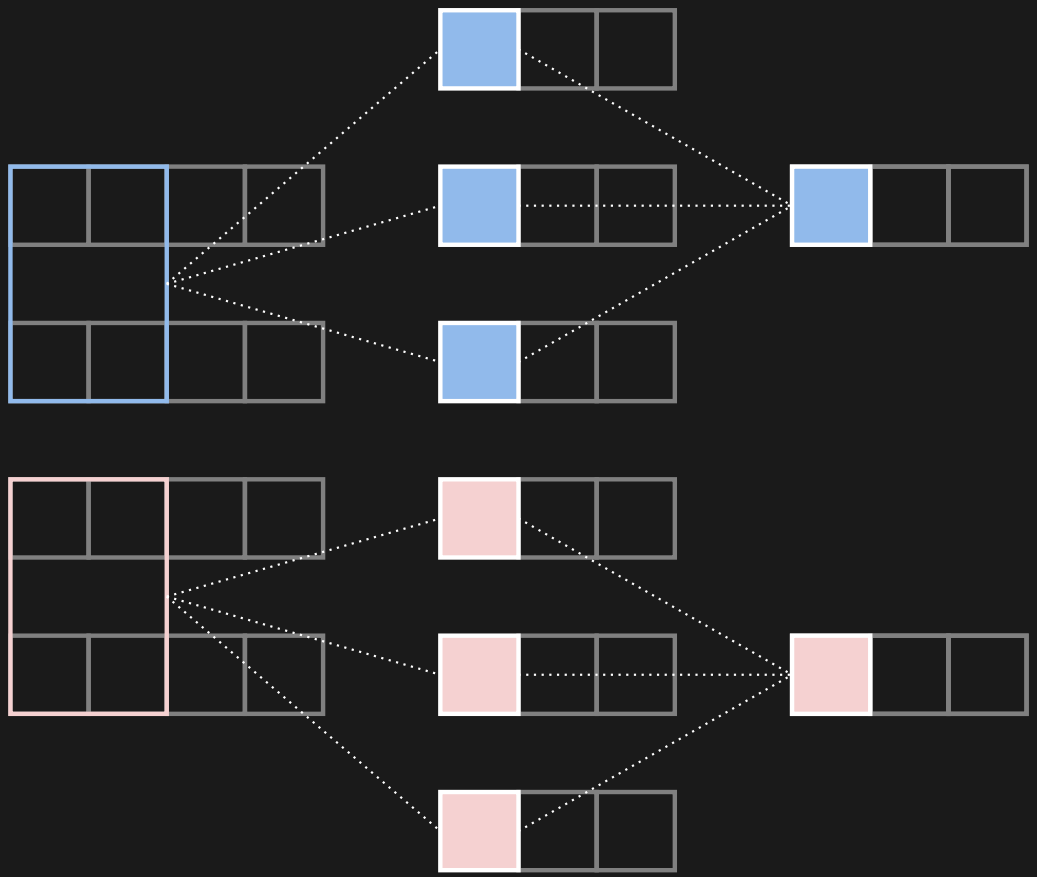
Accepted Papers

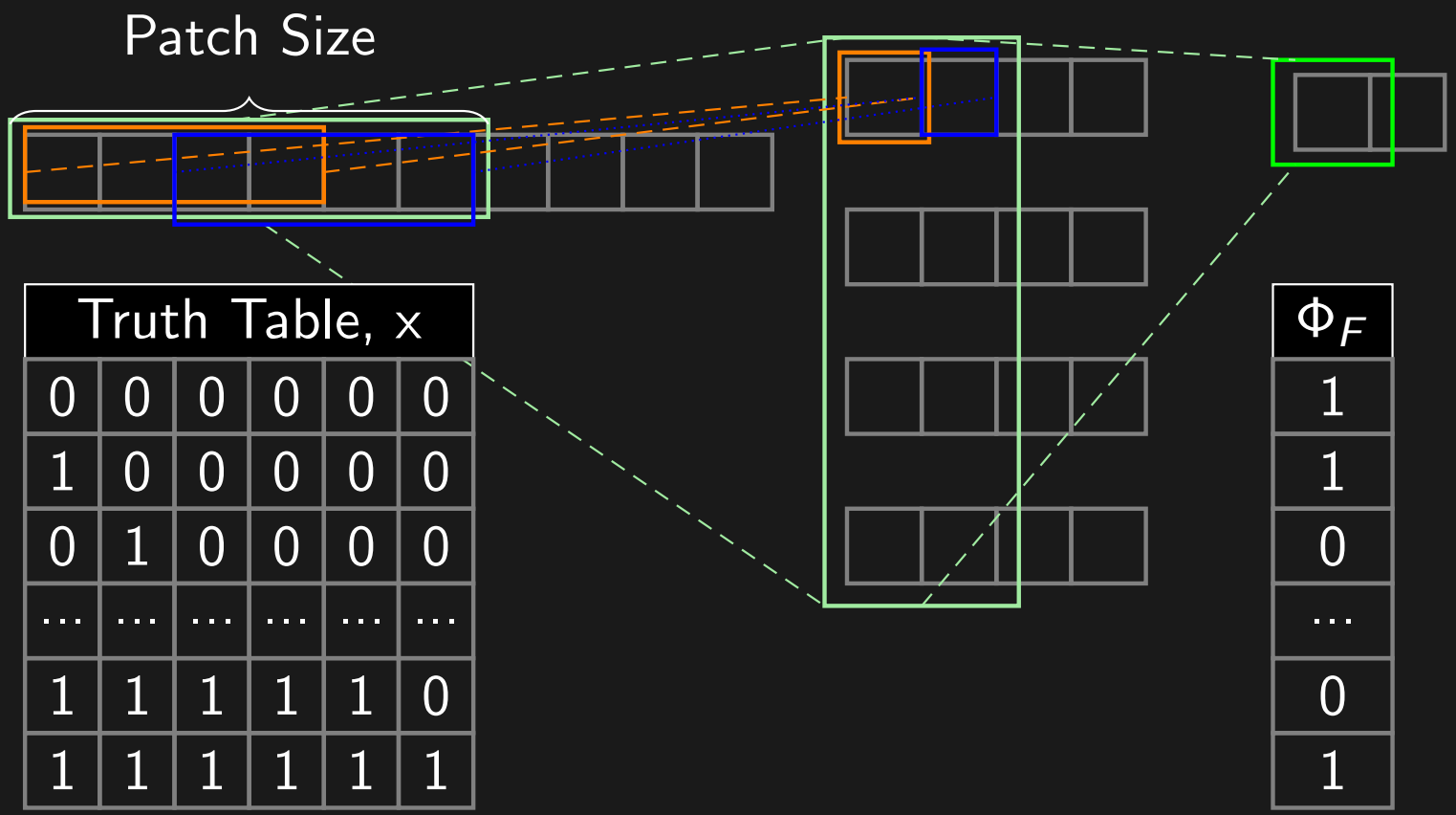
1. A deeper look at machine learning-based cryptanalysis A Benamira, D Gerault, T Peyrin, QQ Tan; *EUROCRYPT 2021*
2. Truth-Table Net: A New Convolutional Architecture Encodable By Design Into SAT Formulas A Benamira, T Peyrin, B Hooi; *ECCV 2022 Workshop AROW*
3. Peek into the Black-Box: Interpretable Neural Network using SAT Equations in Side-Channel Analysis T Yap, A Benamira, S Bhasin, T Peyri; *TCHES 2023*

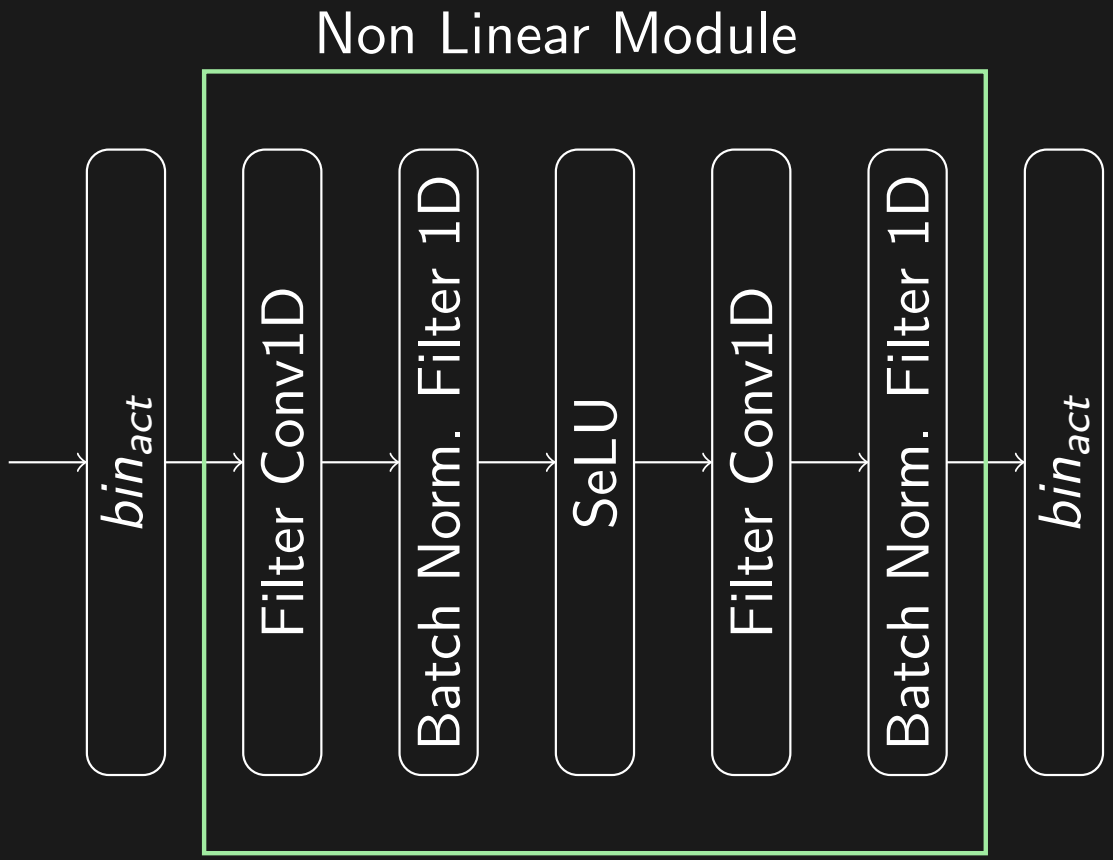
Submitted Papers

1. A Scalable, Interpretable, Verifiable & Differentiable Logic Gate Convolutional Neural Network Architecture From Truth Tables A Benamira, T Guérand, T Peyrin, T Yap, BH Kuen-Yew ; *Submitted at ICML 2023*
2. TT-TFHE: a Torus Fully Homomorphic Encryption-Friendly Neural Network Architecture A Benamira, T Guérand, T Peyrin, S Saha; *Submitted at ICML 2023*






Patch Size
Truth Table, x
0 0 0 0 0 0
1 0 0 0 0 0
0 1 0 0 0 0
... ...
1 1 1 1 1 0
1 1 1 1 1 1
Φ_F
1
1
0
...
0
1



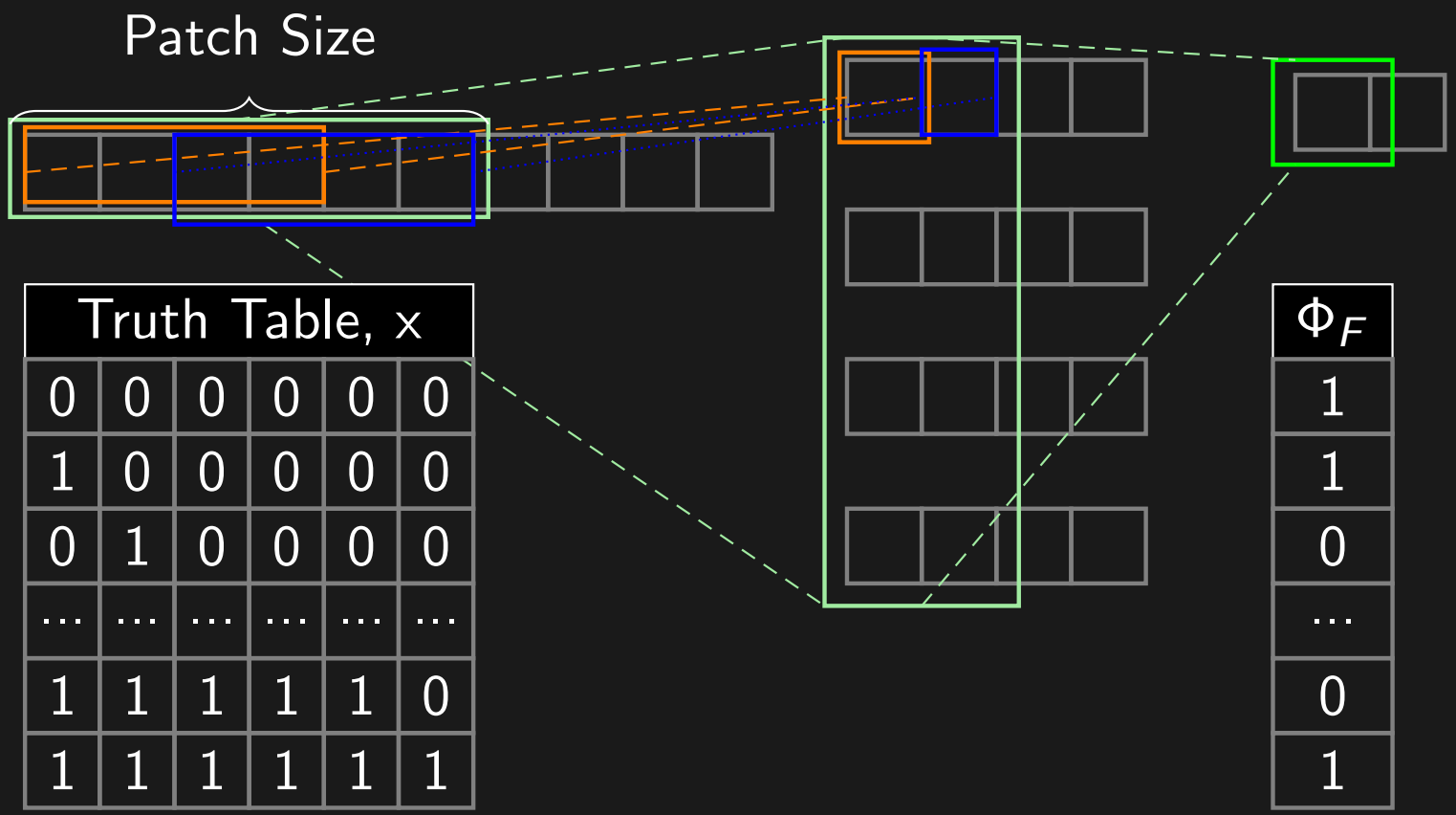
Architecture description of one family of LTT block in 1 dimension: Expanding-AutoEncoder LTT (E-AE LTT)

Image datasets

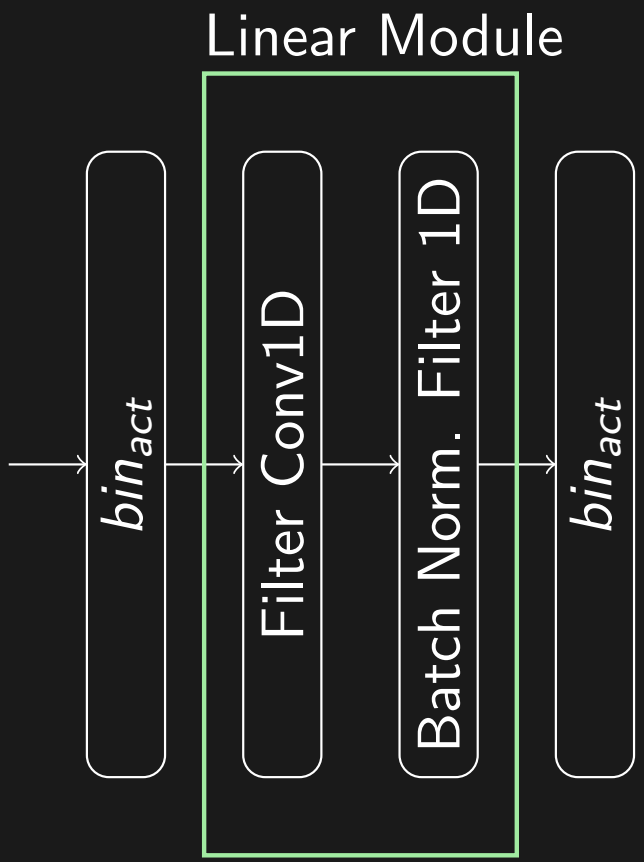
TFHE-based schemes		Full-Pr ($\emptyset/N$)				$VGG_{1B}/\emptyset$	VGG_{1L}/N		VGG_{1B}/N
		TAPAS	GateNet	Zama	**Ours**	**Ours**	Zama	**Ours**	**Ours**
#CPU cores		16	2	6	4	4	6	4	4
MNIST	Acc. (%)	98.6	98.8*	97.1	97.2	97.5	-	98.2	98.1
	Time	37h	44h*	115s	83.6s	0.04s	-	8.7s	7s
CIFAR-10	Acc. (%)	-	80.5*	-	-	70.4	62.3	69.4/72.1	74.1/75.3
	Time	-	3920h*	-	-	0.4s	29m	9.5m/6.2h	9.5m/6.2h

On memory usage

Dataset	Method	FHE type	Accuracy	Server RAM
CIFAR-10	CryptoNets	BFV	-	100 GB
	SHE	LTFHE	92.5%	$<$ 1 TB
	LoLa	BFV	74.1%	12 GB
	Lee et al.	CKKS	91.31%	384 GB
	Zama VGG_{1L}/N	TFHE	62.31%	14.8 GB
	Ours VGG_{1L}/N	TFHE	69.4%	0.82 GB
	Ours VGG_{1B}/N	TFHE	74.1%	0.82 GB


Patch Size
Truth Table, x
0 0 0 0 0 0
1 0 0 0 0 0
0 1 0 0 0 0
... ...
1 1 1 1 1 0
1 1 1 1 1 1
Φ_F
1
1
0
...
0
1

Architecture description of one family of LTT block: Linear LTT (Lin. LTT)

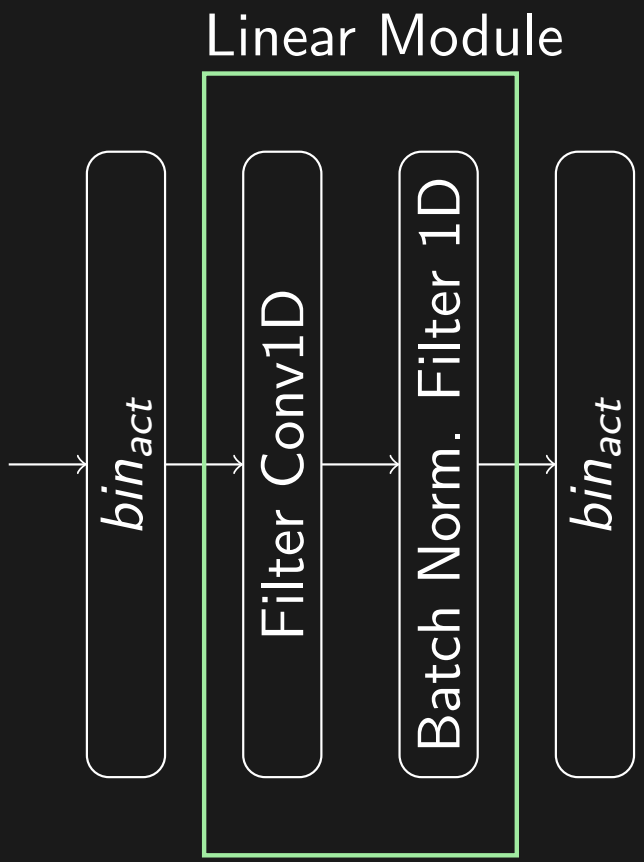


Limit

TTnet based on Lin. LTT can not scale

MNIST Accuracy : 89% with $n = 6$

Architecture description of one family of LTT block: Linear LTT (Lin. LTT)



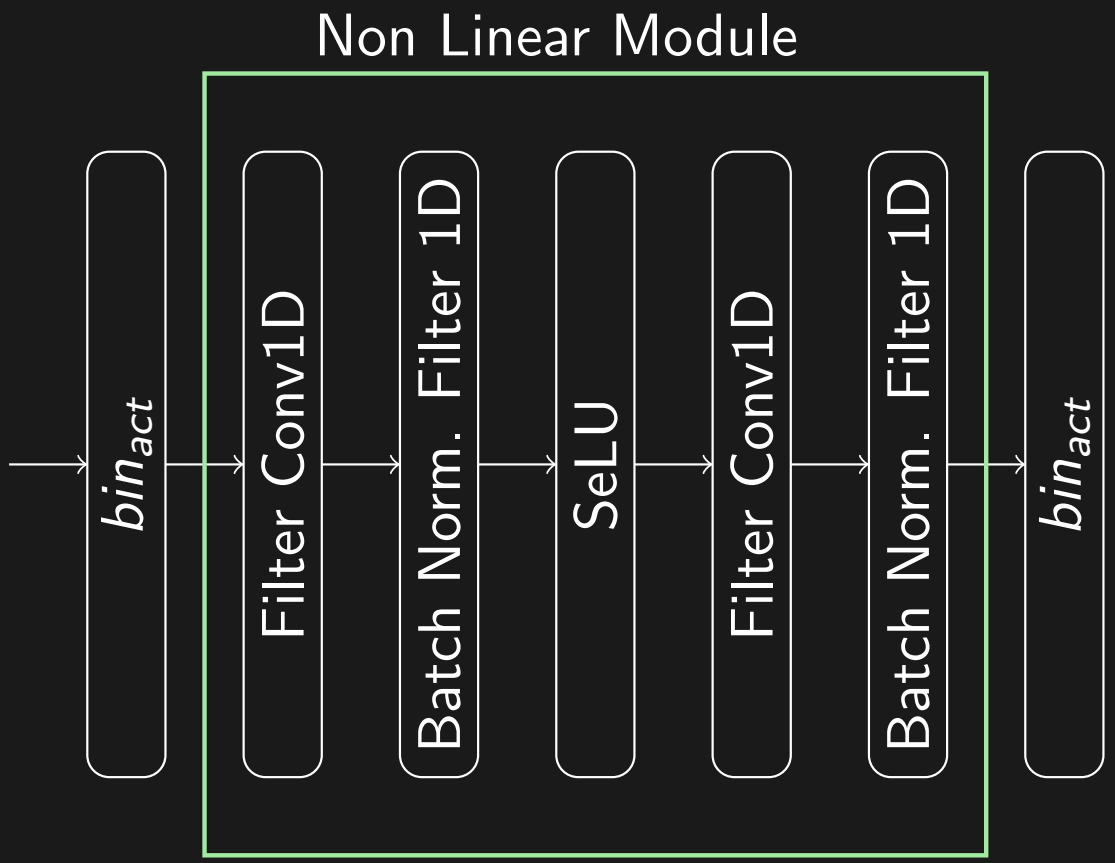
Limit
TTnet based on Lin. LTT can not scale

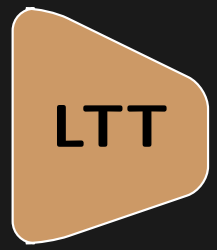
MNIST Accuracy : 89% with $n = 6$

We need to :

- Increase the number of parameters
- Use non linearity layer

Architecture description of an other family of LTT block: Expanding-AutoEncoder LTT (E-AE LTT)





MNIST Accuracy :

98% with $n = 6$

An intuition of why
"LTT is nonlinear in between the step functions"
solves the problem of scalability

Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet, Brendel & Bethge, ICLR 2019

Main idea : ResNET + Patches of size 9×9 are enough to classify Imagenet

Rules Design for Learnable Truth Table (LTT)

- Equivalent to a **small exhaustive Boolean function** ✓
  1. LTT inputs/outputs are binary
  2. LTT input size is "small" : $n \leq 6$

  BONUS: Optimal CNF with Quine–McCluskey algorithm

- **Scalable and accurate** when applied to large datasets ✓
  1. LTT is nonlinear in between the step functions

x_1	x_2	x_3	LTT
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Table: Example truth table of size 3.

With Quine–McCluskey algorithm, **Optimal DNF/CNF** -

$$\text{CNF} = (x_1 \vee x_3) \wedge (\overline{x_2} \vee x_3)$$

First, we train our Neural Network TTnet on the dataset.

Second, we convert our Neural Network TTnet in form of rules based model.

Finally, we delete Neural Network and we analyze the rules.

Use Case 3: formal verification

Dataset MNIST - Hand Written Digits

Goal: Predict the digit of an image

Dataset $\mathcal{D}$ dimensions: 28x28 input pixels variables $\Longrightarrow$ medium dataset for formal verification

Use Case 3: formal verification

Dataset MNIST - Hand Written Digits

Goal: Predict the digit of an image

Dataset $\mathcal{D}$ dimensions: 28x28 input pixels variables $\implies$ medium dataset for formal verification

Property: Given $(x, l) \in \mathcal{D}$, f and ϵ, we want to garantee that :
$\forall x' \; s.t. \; ||x - x'||_\infty < \epsilon, \; f(x') = f(x) = l$