# Breaking the Layer Barrier: Remodeling Private Transformer Inference with Hybrid CKKS and MPC

**Tianshi Xu**[1], Wen-jie Lu[2], Jiangrui Yu[1], Yi Chen[1],
Chenqi Lin[1], Runsheng Wang[1], Meng Li[1].
*[1]Peking University, [2]TikTok.*

# Outline

- Background of Private Transformer Inference
- Motivation of BLB
- Method
- Experimental Results

# Outline

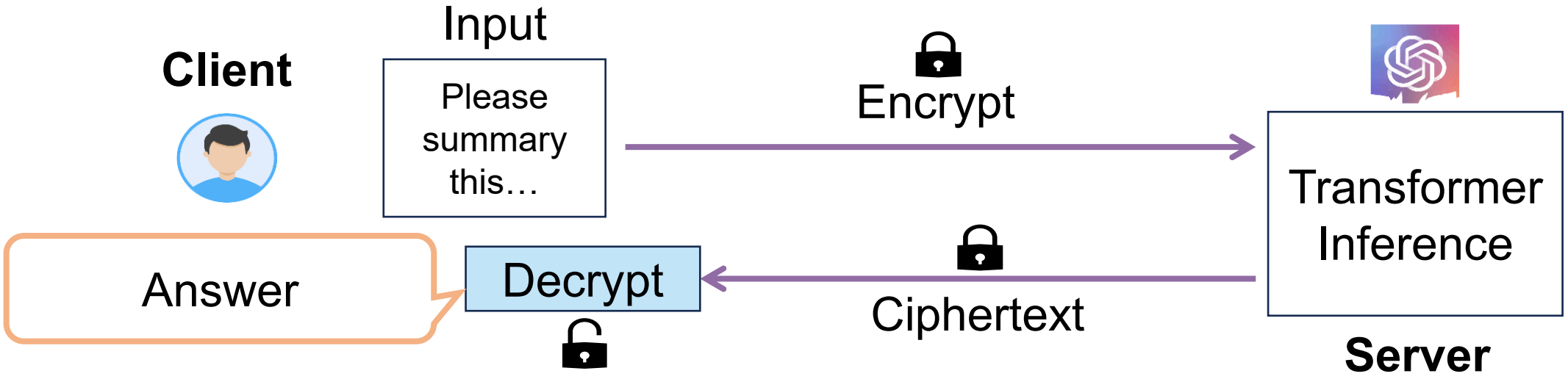
- ❑ Background of Private Transformer Inference
- ❑ Motivation of BLB
- ❑ Method
- ❑ Experimental Results

# Outline

- **Privacy-Preserving Inference for Transformer**
  - Transformer-based LLMs have been deployed in many data sensitive scenarios

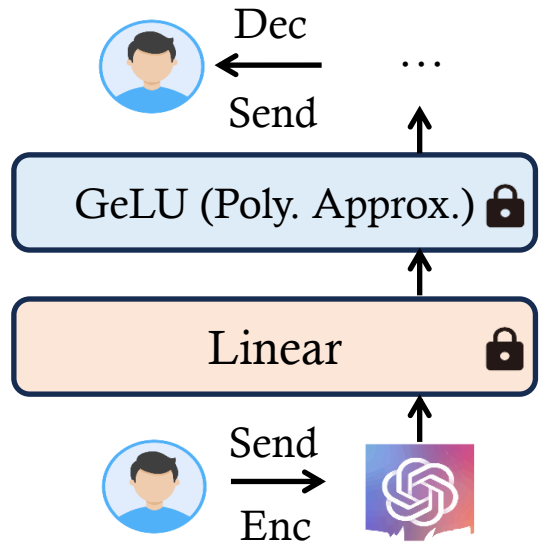
# Outline

- **Privacy-Preserving Inference for Transformer**
  - Transformer-based LLMs have been deployed in many data sensitive scenarios
  - Cryptography-based private inference offers strong provable protection of both model and user input



# Outline

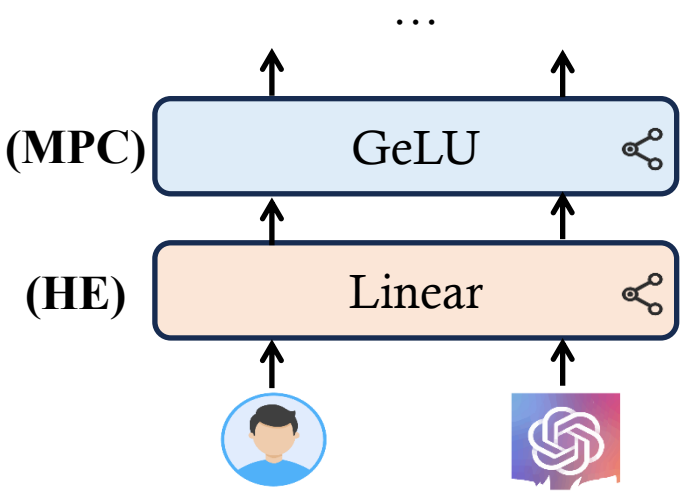
- **Problems of Private Transformer Inference**
  - Fully Homomorphic Encryption (**FHE**)
    - Fail to handle nonlinear layer (Bad accuracy and excessive computation due to bootstrapping)




**(b) Non-interactive FHE**

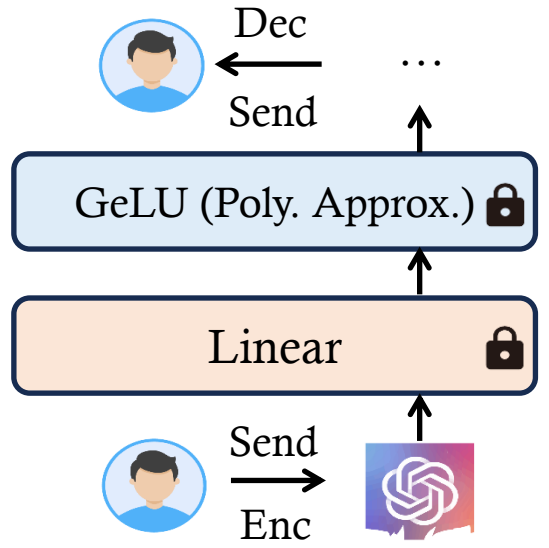
# Outline

- **Problems of Private Transformer Inference**
  - Fully Homomorphic Encryption (**FHE**)
    - Fail to handle nonlinear layer (Bad accuracy and excessive computation due to bootstrapping)
  - Homomorphic Encryption + Secure Multi-party Computation (hybrid **HE/MPC**)
    - High accuracy, but excessive communication (60GB for a BERT-base inference)




(a) Interactive HE/MPC

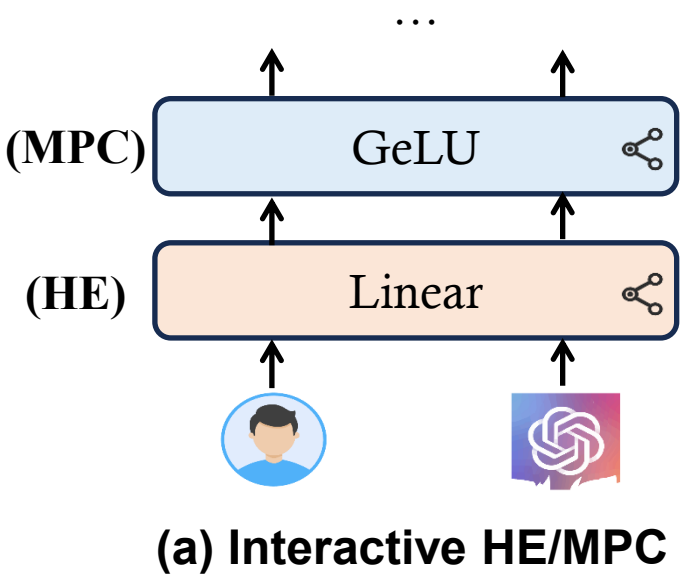
| | Accuracy | Communication | Computation |
|---|---|---|---|
| HE/MPC | ☺ | ☹ | ☺ |
| FHE | ☹ | ☺ | ☹ |
| BLB (ours) | ☺ | ☺ | ☺ (GPU) |




(b) Non-interactive FHE

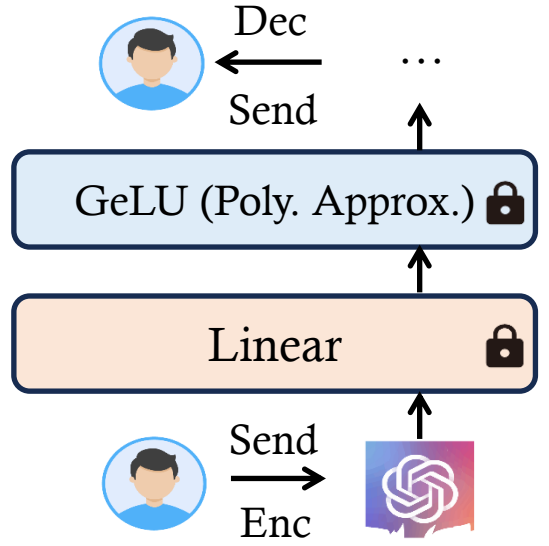
# Outline

- **Problems of Private Transformer Inference**
  - Fully Homomorphic Encryption (**FHE**)
    - Fail to handle nonlinear layer (Bad accuracy and excessive computation due to bootstrapping)
  - Homomorphic Encryption + Secure Multi-party Computation (hybrid **HE/MPC**)
    - High accuracy, but excessive communication (60GB for a BERT-base inference)
  - Optimization Golden Rule: Accuracy > Communication > Computation




(a) Interactive HE/MPC

| | Accuracy | Communication | Computation |
|---|---|---|---|
| HE/MPC | ☺ | ☹ | ☺ |
| FHE | ☹ | ☺ | ☹ |
| BLB (ours) | ☺ | ☺ | ☺ (GPU) |




(b) Non-interactive FHE

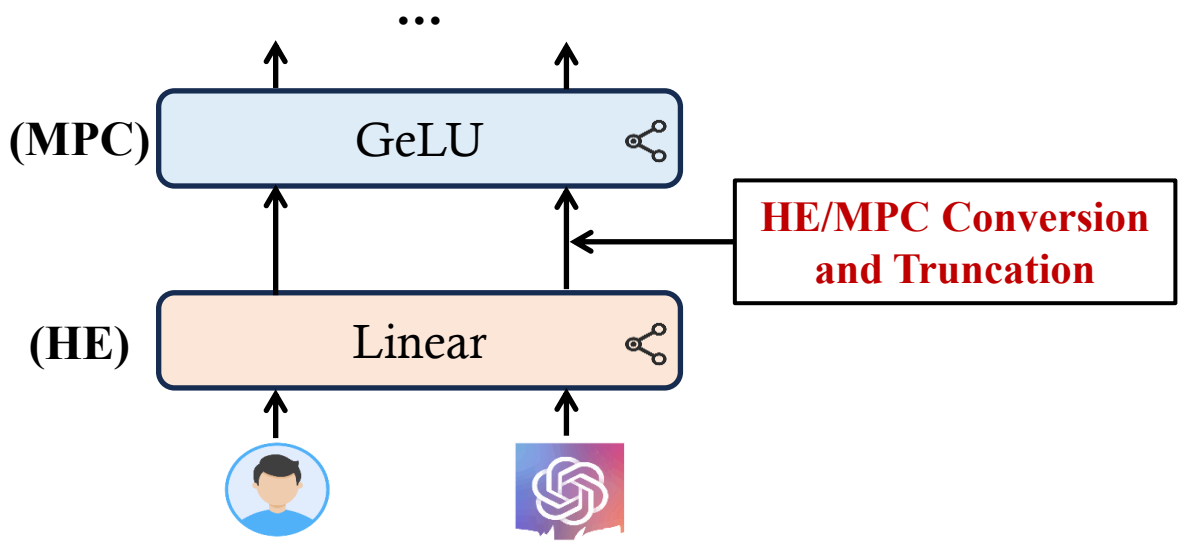
# Outline

- ❑ Background of Private Transformer Inference
- ❑ Motivation of BLB
- ❑ Method
- ❑ Experimental Results

# Motivation

## Private Inference with Hybrid HE/MPC

- Linear layers: Homomorphic Encryption (HE)
- Nonlinear layers: Secure Multi-party Computation (MPC)

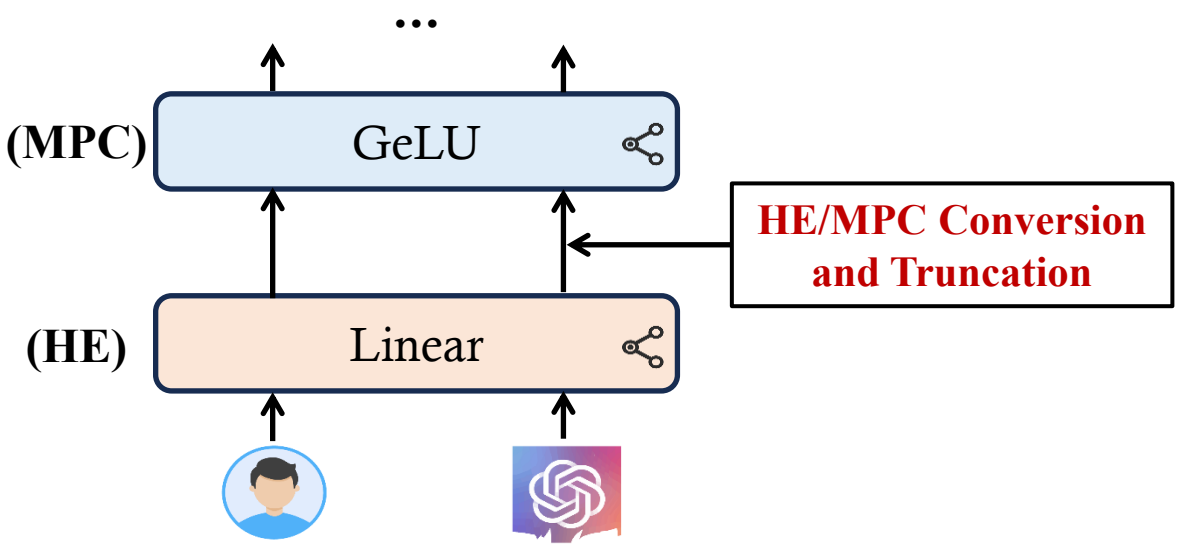



(a) Hybrid HE/MPC framework

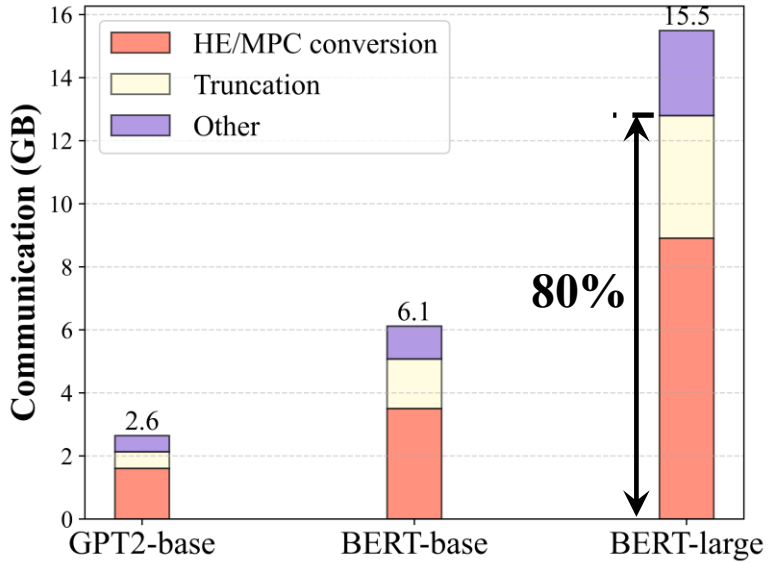
# Motivation

## Private Inference with Hybrid HE/MPC

- Linear layers: Homomorphic Encryption (HE)
- Nonlinear layers: Secure Multi-party Computation (MPC)
- Communication comes from: HE/MPC Conversions and Truncations cost dominates (>80%)




(a) Hybrid HE/MPC framework



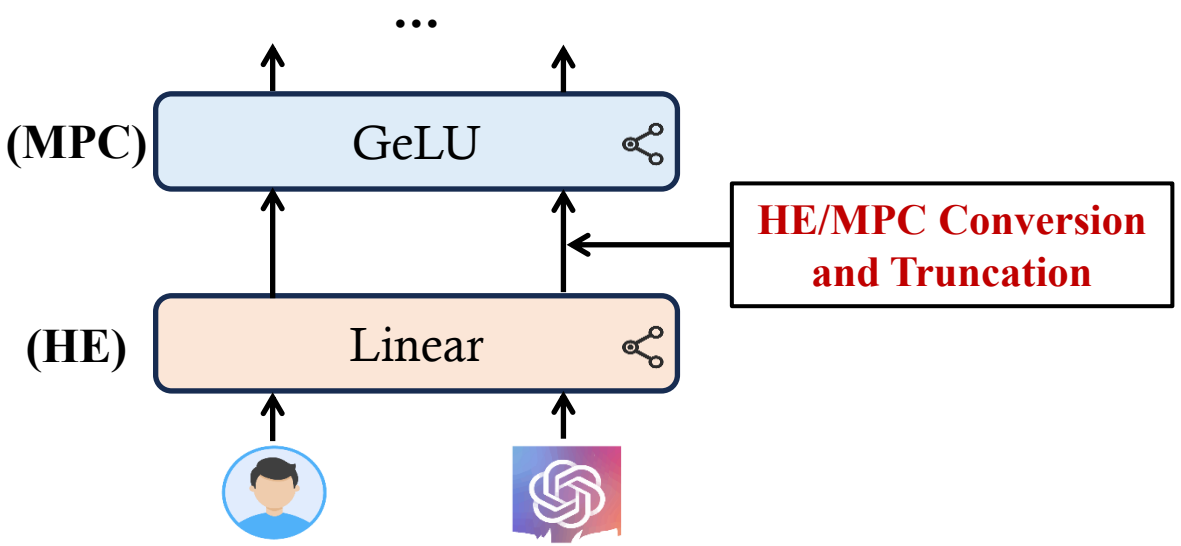

(b) Communication breakdown

# Motivation

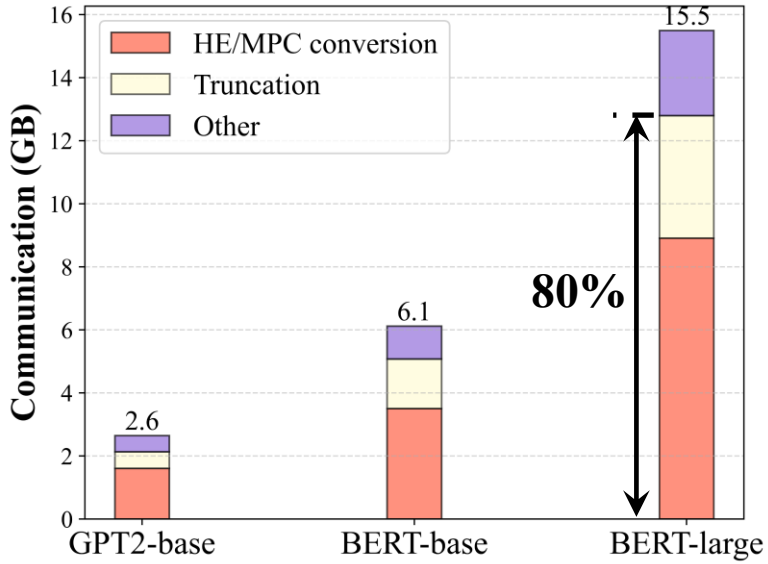
## Private Inference with Hybrid HE/MPC

- ❑ Linear layers: Homomorphic Encryption (HE)
- ❑ Nonlinear layers: Secure Multi-party Computation (MPC)
- ❑ Communication comes from: HE/MPC Conversions and Truncations cost dominates (>80%)

❑ **Our Goal:** Reduce communication overhead in hybrid HE/MPC for Transformer inference




(a) Hybrid HE/MPC framework

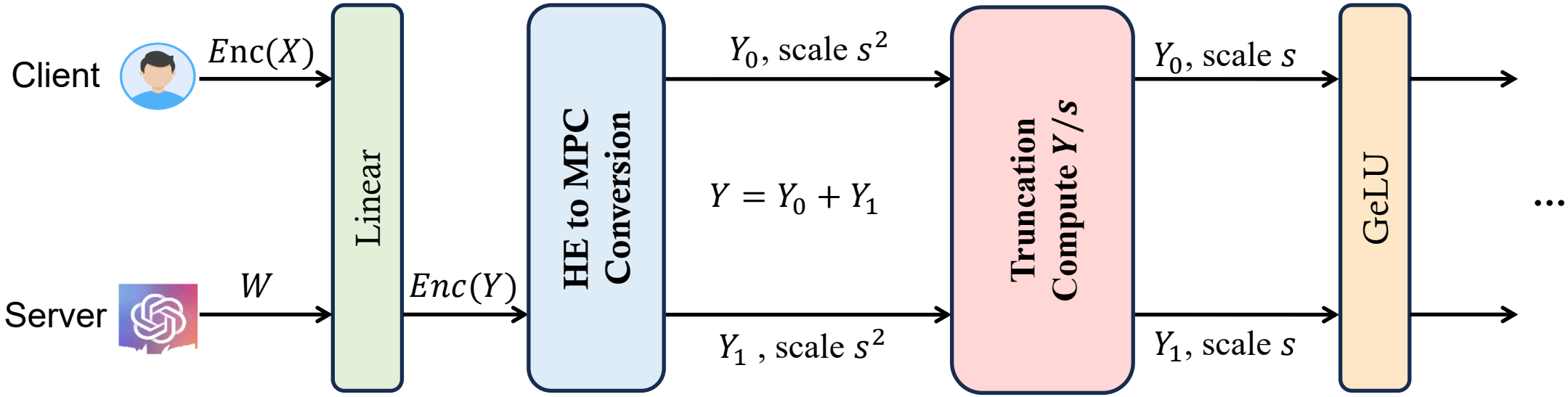



(b) Communication breakdown

# Motivation

## HE/MPC Conversions and Truncations cost dominates (>80%)

- HE/MPC (Secret Share) Conversion
- Truncation: Restore scale in fixed-point arithmetic

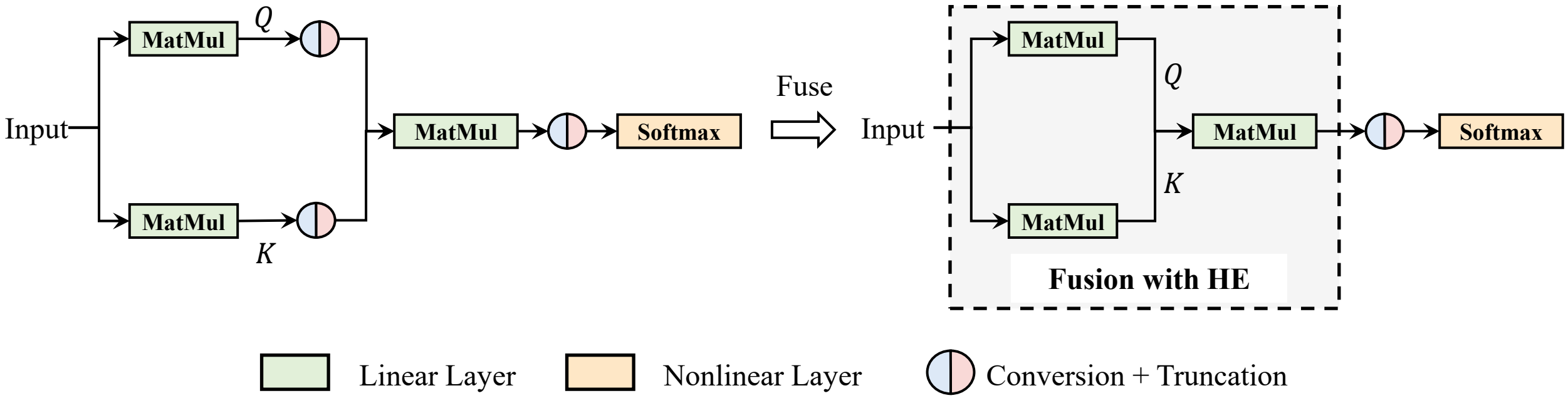



$Enc(\cdot)$: Encryption  $Y_0, Y_1$: secret share of $Y$

# Motivation

## Previous Solution: Linear Layer Fusion

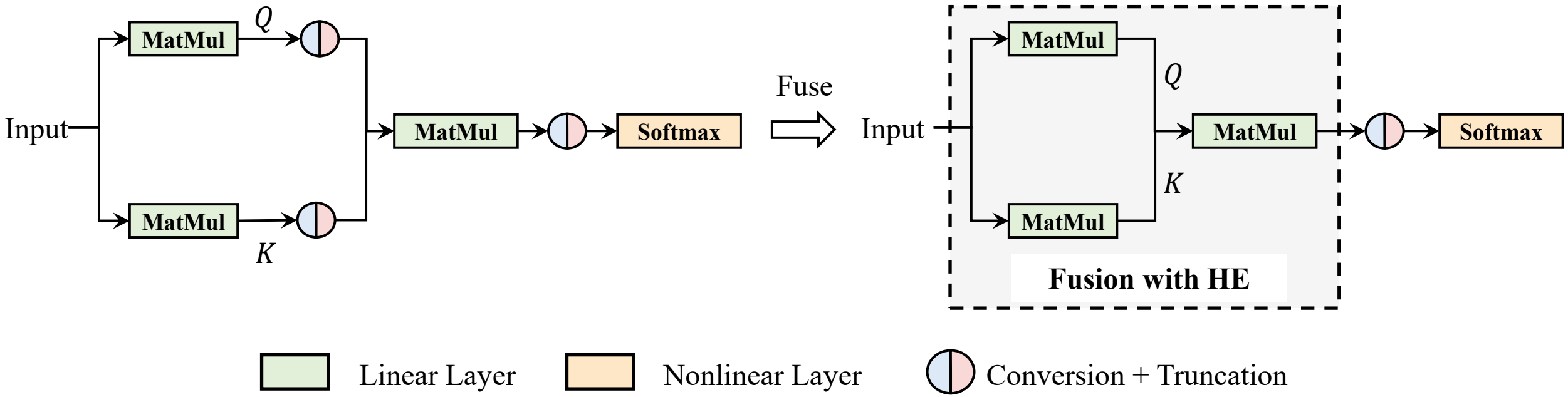
- Using HE to evaluate consecutive linear layers within a single communication round
- Eliminating all conversions and truncations between consecutive linear layers



# Motivation

## Previous Solution: Linear Layer Fusion

- Using HE to evaluate consecutive linear layers within a single communication round
- Eliminating all conversions and truncations between consecutive linear layers
- Example: $QK^T, Q = XW_Q, K = XW_K$



# Motivation

## Three Limitations of Linear Layer Fusion

### ❑ Limitation 1: Limited Flexibility

❑ Only two consecutive Matrix Multiplications $QK^T$ can be fused in one Transformer block

Only here can be fused!

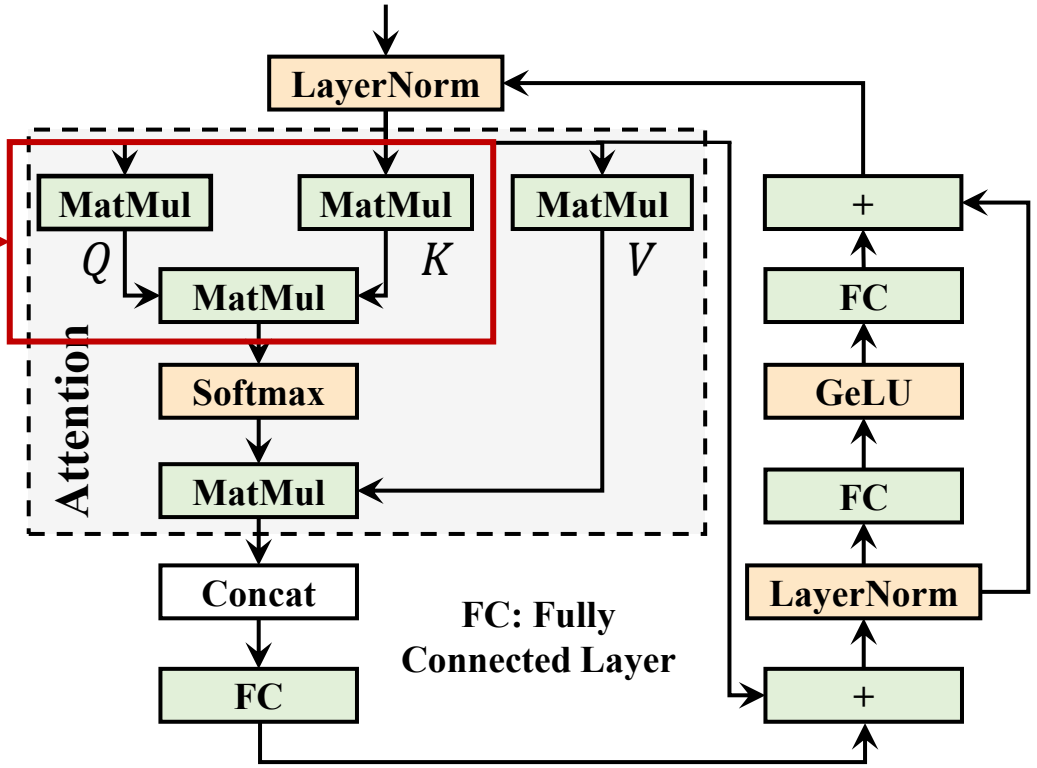
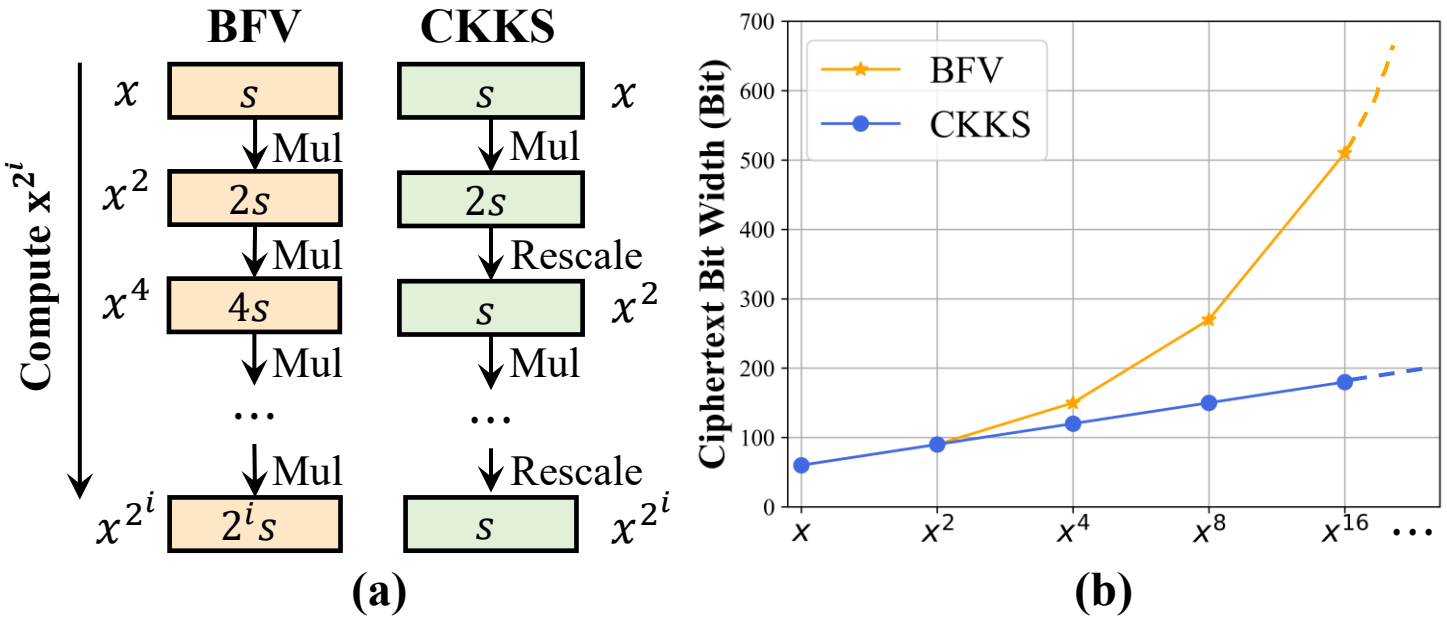



**Fig:** A Transformer block

# Motivation

## Three Limitations of Linear Layer Fusion

- **Limitation 2: High Ciphertext Bit Width using BFV/MPC**
  - BFV cannot reduce scale, needs exponential scale growth with multiplication depth
  - Exponential growth of scale, plaintext bit-width and ciphertext bit-width



# Motivation

## Three Limitations of Linear Layer Fusion

- **Limitation 3: Suboptimal MatMul Protocol after Fusion**
  - Ciphertext-Ciphertext (ct-ct) Matrix Multiplication in HE is extremely complicated!
  - Adjusting the packing of intermediate ciphertexts involves complex HE operations

# Motivation

## Three Limitations of Linear Layer Fusion

- **Limitation 3: Suboptimal MatMul Protocol after Fusion**
  - Ciphertext-Ciphertext (ct-ct) Matrix Multiplication in HE is extremely complicated!
  - Adjusting the packing of intermediate ciphertexts involves complex HE operations
  - BOLT(S&P'24) requires approximately 20× more computationally-intensive HE rotations for fused ct-ct MatMul compared to unfused Ciphertext-Plaintext (ct-pt) protocol

# Outline

- ❑ Background of Private Transformer Inference
- ❑ Motivation of BLB
- ❑ Method
- ❑ Experimental Results

# Core Idea: Breaking the Layer Barrier

**Observation:** Nonlinear layers in Transformer consist of many small linear operators

(e.g., Mul, Add)

$$\mathrm{LayerNorm}(\mathbf{X})_{i,j} = \frac{\boldsymbol{\gamma}_j(\mathbf{X}_{i,j} - \boldsymbol{\mu}_i)}{\boldsymbol{\sigma}_i} + \boldsymbol{\beta}_j$$

$$\mathrm{ApproxGELU}(x) = \begin{cases} x & \text{if } x > 2.7, \\ a|x|^4 + b|x|^3 + c|x|^2 & \\ \quad + d|x| + e + 0.5x & \text{if } |x| \le 2.7, \\ 0 & \text{if } x < -2.7. \end{cases}$$

# Core Idea: Breaking the Layer Barrier

**Observation:** Nonlinear layers in Transformer consist of many small linear operators

(e.g., Mul, Add)

$$\mathrm{LayerNorm}(\mathbf{X})_{i,j} = \frac{\boldsymbol{\gamma}_j(\mathbf{X}_{i,j} - \boldsymbol{\mu}_i)}{\boldsymbol{\sigma}_i} + \boldsymbol{\beta}_j$$

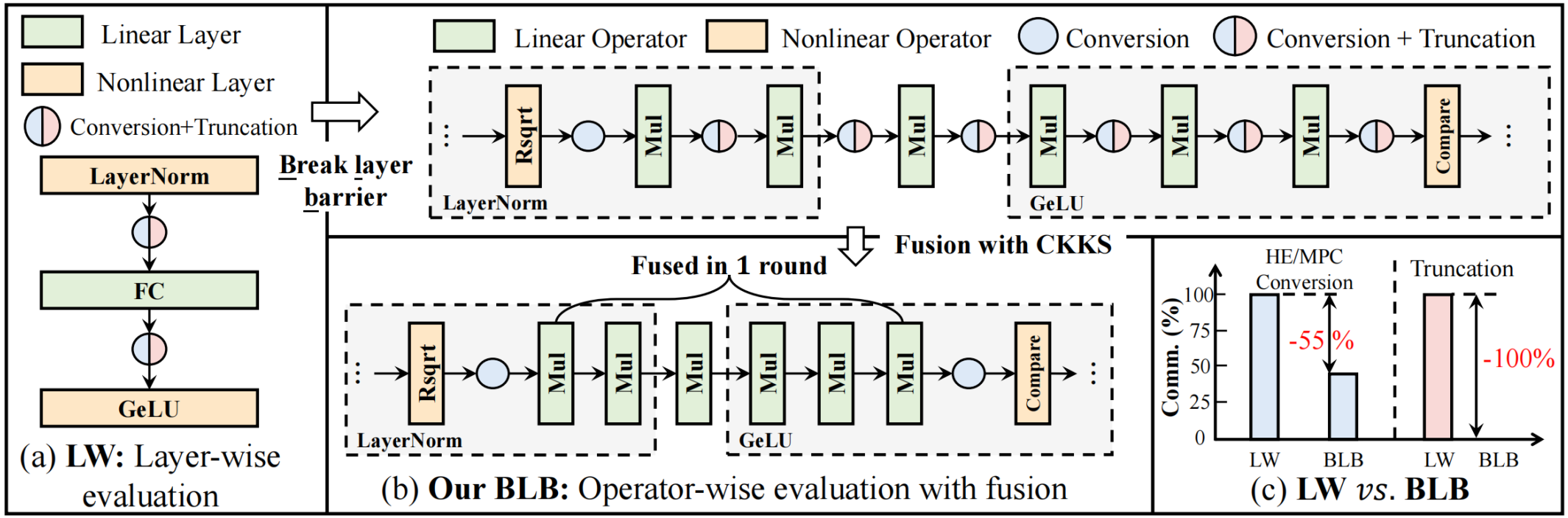
$$\mathrm{ApproxGELU}(x) = \begin{cases} x & \text{if } x > 2.7, \\ a|x|^4 + b|x|^3 + c|x|^2 & \\ \quad + d|x| + e + 0.5x & \text{if } |x| \le 2.7, \\ 0 & \text{if } x < -2.7. \end{cases}$$

**Intuition**: Nonlinear layers should no longer be treated as indivisible units. A finer-grained analysis is required.

# Core Idea: Breaking the Layer Barrier

**BLB:** From **layer-wise (LW) evaluation** to **operator-wise evaluation with fusion**

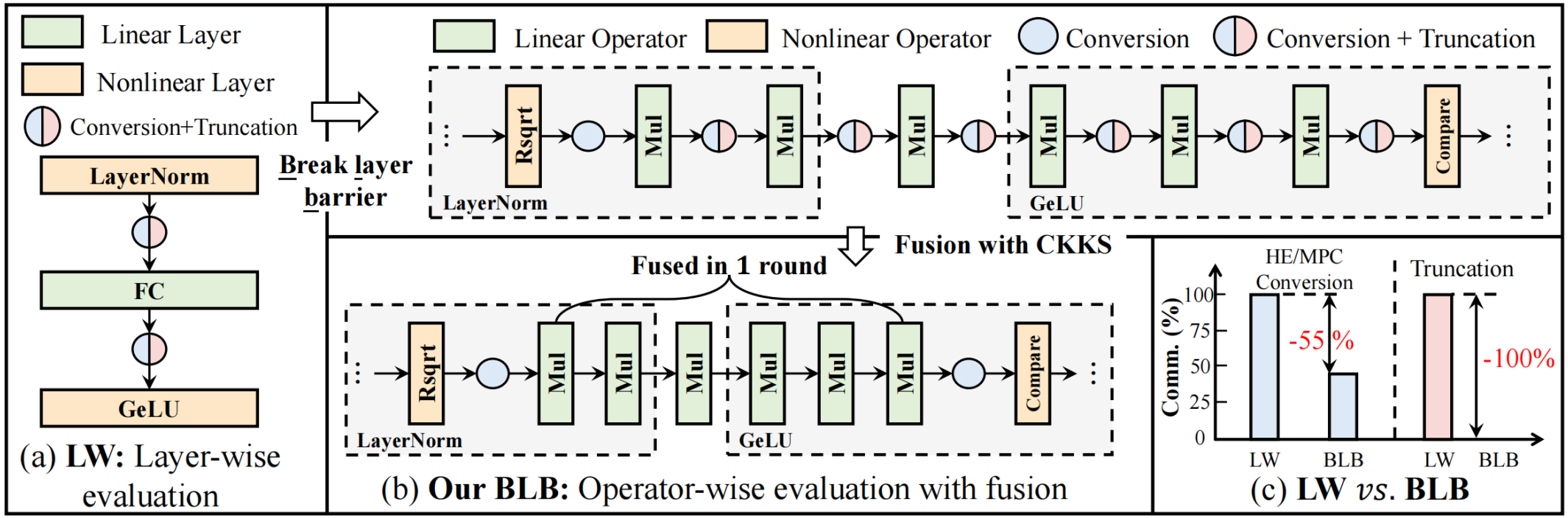
- Decompose layers into operators, classifying them into linear operators and nonlinear operators



# Core Idea: Breaking the Layer Barrier

**BLB:** From **layer-wise (LW) evaluation** to **operator-wise evaluation with fusion**

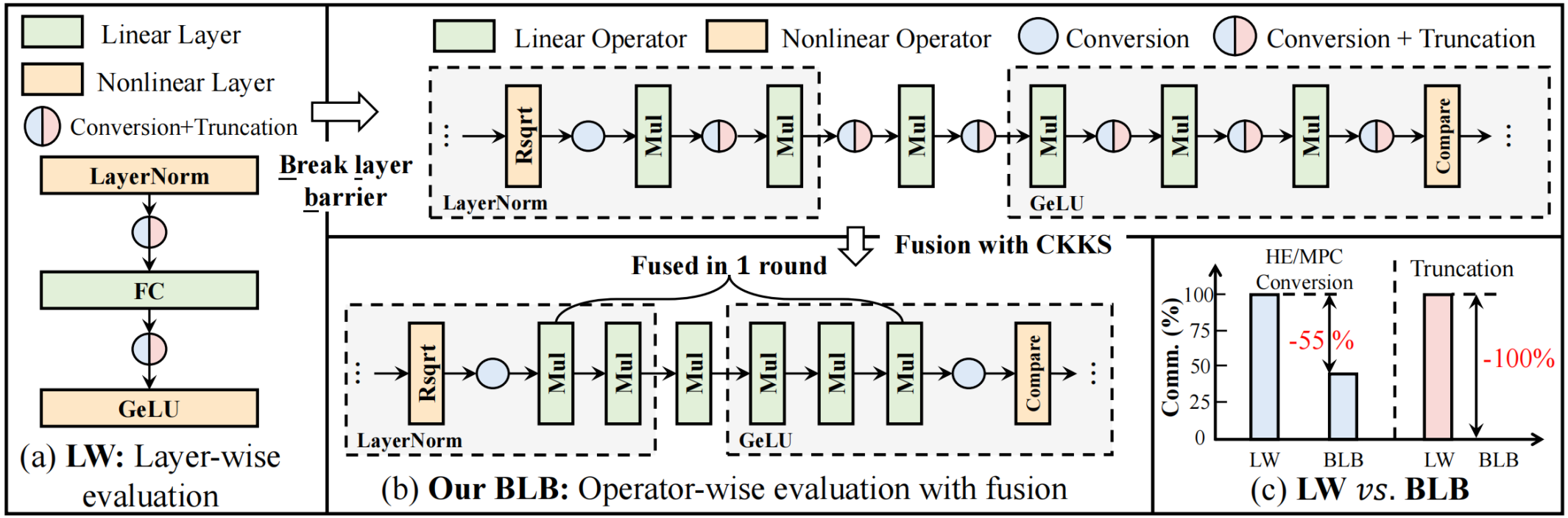
- Decompose layers into operators, classifying them into linear operators and nonlinear operators
- Fuse adjacent linear operators across layers (evaluate them within a communication round)



# Core Idea: Breaking the Layer Barrier

**BLB:** From **layer-wise (LW) evaluation** to **operator-wise evaluation with fusion**

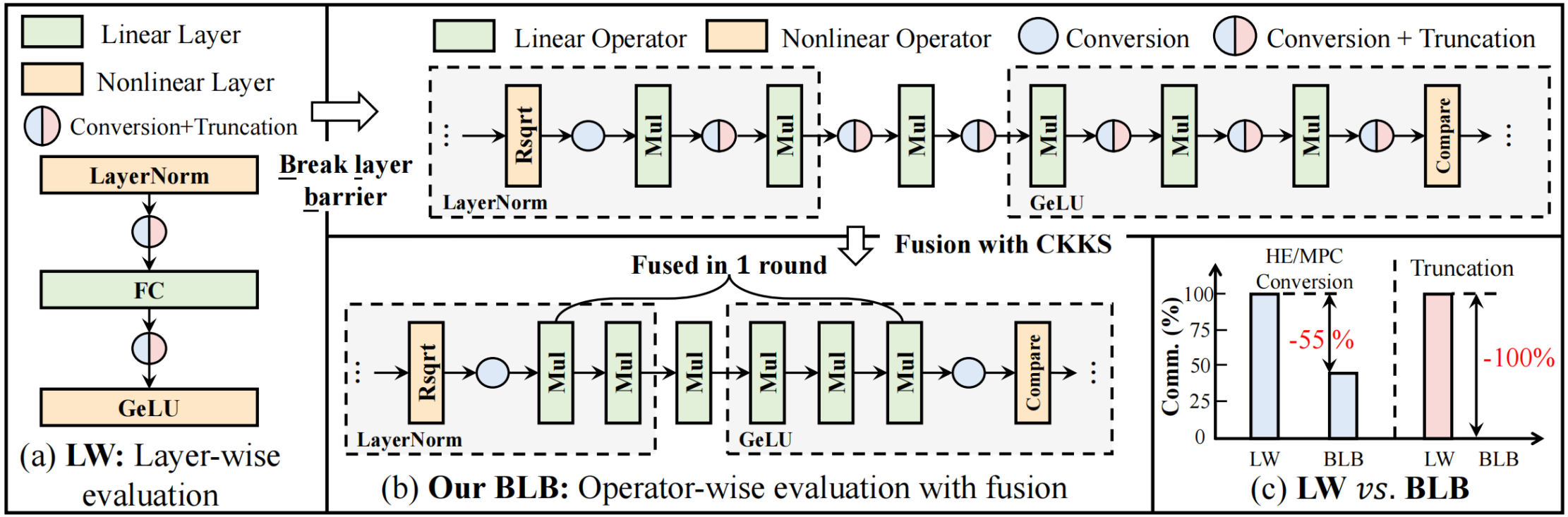
- Decompose layers into operators, classifying them into linear operators and nonlinear operators
- Fuse adjacent linear operators across layers (evaluate them within a communication round)
- Eliminate all conversions and truncations between adjacent linear operators



# Core Idea: Breaking the Layer Barrier

**Example:** LayerNorm + Fully Connected Layer (FC) + GeLU

- The last two ops in LayerNorm, the subsequent FC Layer, the first three ops in GeLU can be fused
- 100% Truncation and 55% HE/MPC Conversion can be eliminated after fusion



# BLB Framework

## Three Technical Highlights

1. **FineGrainFusion:** Systematic operator-level fusion patterns across Transformer layers

| Type | Name | Description |
|---|---|---|
| *Linear operators* | | |
| Identity | $\text{ewadd}_{cc}, \text{ewadd}_{cp}$ | Element-wise addition of ct-ct, ct-pt |
| | $\text{ewmul}_{cc}, \text{ewmul}_{cp}$ | Element-wise multiplication of ct-ct, ct-pt |
| Expansion | $\text{sadd}_{cc}, \text{sadd}_{cp}$ | Scalar addition of ct-ct, ct-pt |
| | $\text{smul}_{cc}, \text{smul}_{cp}$ | Scalar multiplication of ct-ct, ct-pt |
| Reduction | sum | Sum in a specified dimension of the input |
| Transformation | $\text{matmul}_{cc}, \text{matmul}_{cp}$ | MatMul of ct-ct, ct-pt |
| *Nonlinear operators* | | |
| cmp | | $[\![\mathbf{1}\{x < y\}]\!]^B \leftarrow \text{cmp}([\![x]\!]^M, [\![y]\!]^M)$ |
| mux | | $[\![b \cdot x]\!]^M \leftarrow \text{mux}([\![b]\!]^B, [\![x]\!]^M)$ |
| rec, rsqrt | | Reciprocal, Reciprocal Sqrt [47] |

**Operator Categorization**

| First op \ Second op | Identity | Expansion | Reduction | Transformation |
|---|---|---|---|---|
| **Identity** | Identity | Expansion | Reduction | Transformation |
| **Expansion** | Expansion | × | Identity | Expansion |
| **Reduction** | Reduction | Identity | × | × |
| **Transformation** | Transformation | × | Reduction | Transformation |

**Fusion pattern analysis for linear operators**

# BLB Framework

## Three Technical Highlights

2. **CKKS+MPC Conversion:** First secure protocol to support CKKS/MPC conversion
   - We use CKKS instead of BFV to control the ciphertext bitwidth growth
   - Previous CKKS-MPC conversion protocol is insecure, leaking computation results

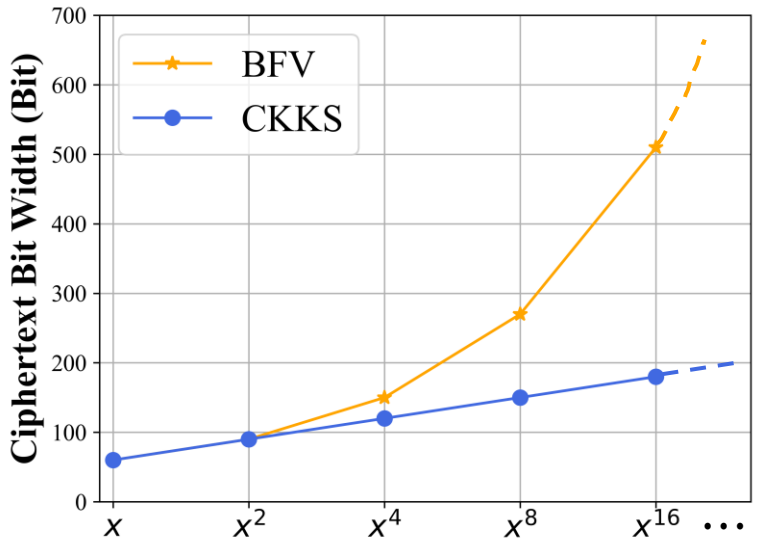



**Fig:** CKKS ct biwdith grows linearly

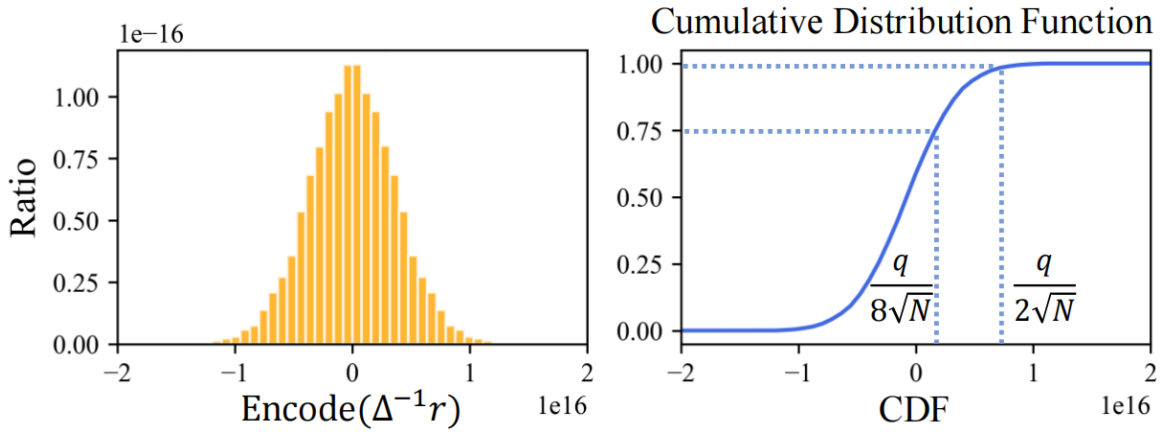



**Fig:** CKKS Encode function narrows random mask

# BLB Framework

## Three Technical Highlights

3. **Rotation-Efficient Ct-Ct MatMul:** 29 × and 8 × fewer HE rotations than BOLT (S&P'24) and Powerformer (ACL'25)

   - Optimized for $QK^T$ and $\mathrm{Softmax} \times V$; multi-head packing; BSGS optimization

| $\mathrm{matmul}_{cc}$ | | BOLT [45] | Powerformer [46] | BLB |
|---|---|---|---|---|
| $Q_h K_h^T$ $h \in [H]$ | # Rot | $O(\frac{md}{s}(m+\log_2 m))$ 18432 | $O(\frac{md}{s}(\frac{5}{2}m+4\sqrt{m}))$ 5856 | $O(\frac{md}{s}(\frac{\sqrt{d}}{H}+\sqrt{m})+m)$ 640 |
| | # CMult. | $O(\frac{m^2 d}{s})$ 2048 | $O(\frac{m^2 d}{s})$ 1024 | $O(\frac{m^2 d}{s})$ 1024 |
| $\mathrm{Softmax} \times V_h$ $h \in [H]$ | # Rot | $O(\frac{m^2 H}{s}(m+\log_2 m))$ 28560 | $O(\frac{m^2 H}{s}(\frac{5}{2}m+4\sqrt{m}))$ 6508 | $O(\frac{m^2 H}{s}(\sqrt{\frac{m}{H}}+\sqrt{m})+m)$ 1056 |
| | # CMult. | $O(\frac{m^2 d}{s})$ 1024 | $O(\frac{m^3 H}{s})$ 2048 | $O(\frac{m^3 H}{s})$ 2048 |

# Outline

- ❑ Background of Private Transformer Inference
- ❑ Motivation of BLB
- ❑ Method
- ❑ Experimental Results

# Experimental Results

## Experimental Setup

- Implementation: SEAL, EzPC, Phantom FHE (GPU)
- Models: GPT2-base, BERT-base, BERT-large
- Baselines
  - HE/2PC: Iron (NeurIPS'22), BOLT (S&P'24), Bumblebee (NDSS'25)
  - 3PC: SIGMA (PETS'24), MPCFormer (ICLR'23)
  - FHE: NEXUS (NDSS'25)

# Experimental Results

## Communication Reduction

- **21** × *vs.* BOLT (S&P'24)
- **2** × *vs.* Bumblebee (NDSS'25)

## Latency Reduction

- Up to **29** × on CPU
- Up to **13** × on GPU

| Model | Framework | Latency (min) | | | Comm. (GB) |
|---|---|---|---|---|---|
| | | LAN | $WAN_2$ | $WAN_3$ | |
| GPT2-base 64 input tokens | SIGMA [26]* | 7.7 | 25.1 | 38.5 | 28.7 |
| | MPCFormer [39]* | 2.1 | 7.2 | 10.5 | 7.3 |
| | BOLT [45] | 9.7 | 34.0 | 53.6 | 34.8 |
| | Bumblebee [41] | **2.9** | **5.2** | 12.3 | 2.5 |
| | BLB | 4.0 | 6.8 | **10.2** | **1.5** |
| | BOLT (GPU) [45] | 7.6 | 31.8 | 51.6 | 34.8 |
| | Bumblebee (GPU) [41] | 2.6 | 4.9 | 12.0 | 2.5 |
| | BLB (GPU) | **2.0** | **3.9** | **8.1** | **1.5** |
| BERT-base 128 input tokens | NEXUS [56] | 457 | 458 | 470 | 0.2 |
| | Iron [27] | 66.7 | / | / | 76.5 |
| | SIGMA [26]* | 7.8 | 30.4 | 48.6 | 34.4 |
| | MPCFormer [39]* | 5.5 | 18.0 | 27.5 | 12.1 |
| | BOLT [45] | 22.1 | 85.0 | 130.0 | 63.6 |
| | Bumblebee [41] | **5.0** | 12.3 | 22.1 | 5.8 |
| | BLB | 6.1 | **10.5** | **17.2** | **3.0** |
| | NEXUS (GPU) [56] | 19.9 | 20.8 | 32.5 | 0.2 |
| | BOLT (GPU) [45] | 15.5 | 77.9 | 122.7 | 63.6 |
| | Bumblebee (GPU) [41] | 4.3 | 11.6 | 21.3 | 5.8 |
| | BLB (GPU) | **2.5** | **6.6** | **13.2** | **3.0** |
| BERT-large 128 input tokens | Iron [27] | 92.0 | / | / | 220.0 |
| | SIGMA [26]* | 20.5 | 102.5 | 155.5 | 92.8 |
| | MPCFormer [39]* | 7.7 | 34.1 | 52.0 | 32.6 |
| | BOLT [45] | 57.6 | 222.0 | 274.4 | 158.9 |
| | Bumblebee [41] | **10.6** | 32.4 | 51.3 | 15.2 |
| | BLB | 15.1 | **29.0** | **37.7** | **7.8** |
| | BOLT (GPU) [45] | 43.8 | 208.2 | 260.6 | 158.9 |
| | Bumblebee (GPU) [41] | 9.7 | 30.8 | 49.2 | 15.2 |
| | BLB (GPU) | **6.6** | **16.2** | **24.9** | **7.8** |

* Frameworks that require three parties.

# Conclusion

- BLB breaks the **“one-layer, one-protocol”** paradigm
- Enables **operator-level fusion** with CKKS+MPC
- Achieves **significant communication and latency reduction**

*From layers to operators — it’s time to break the barrier*