CipherPrune: Efficient and Scalable Private Transformer Inference

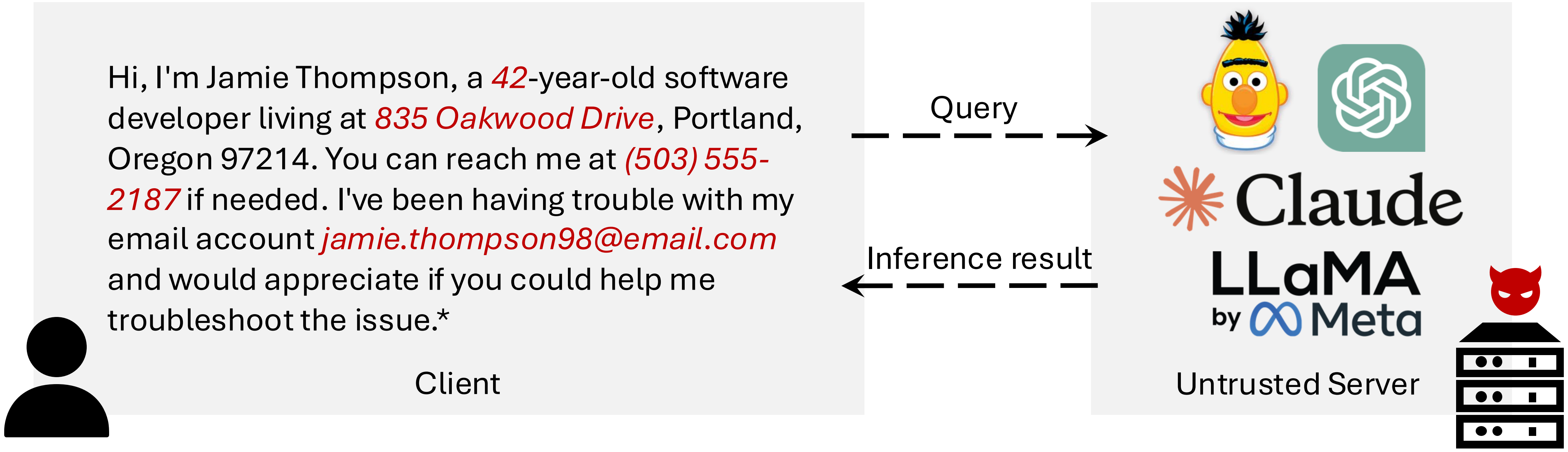
Yancheng Zhang[1], Jiaqi Xue[1], Mengxin Zheng[1]
Mimi Xie[2], Mingzhe Zhang[3], Lei Jiang[4], Qian Lou[1]

[1]University of Central Florida
[2]University of Texas at San Antonio
[3]Ant Research
[4]Indiana University Bloomington

ICLR'25
24th Apr. 2025

Data Privacy is Important during Transformer Inference

Transformer-based models are widely used to process highly confidential information.



Hi, I'm Jamie Thompson, a *42*-year-old software developer living at *835 Oakwood Drive*, Portland, Oregon 97214. You can reach me at *(503) 555-2187* if needed. I've been having trouble with my email account *jamie.thompson98@email.com* and would appreciate if you could help me troubleshoot the issue.*

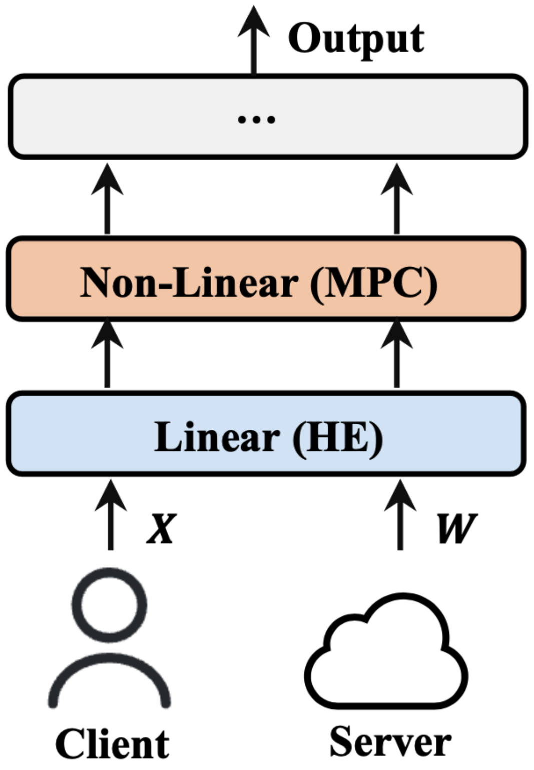
User queries can often contain private information like name, age, address, phone number, email.......

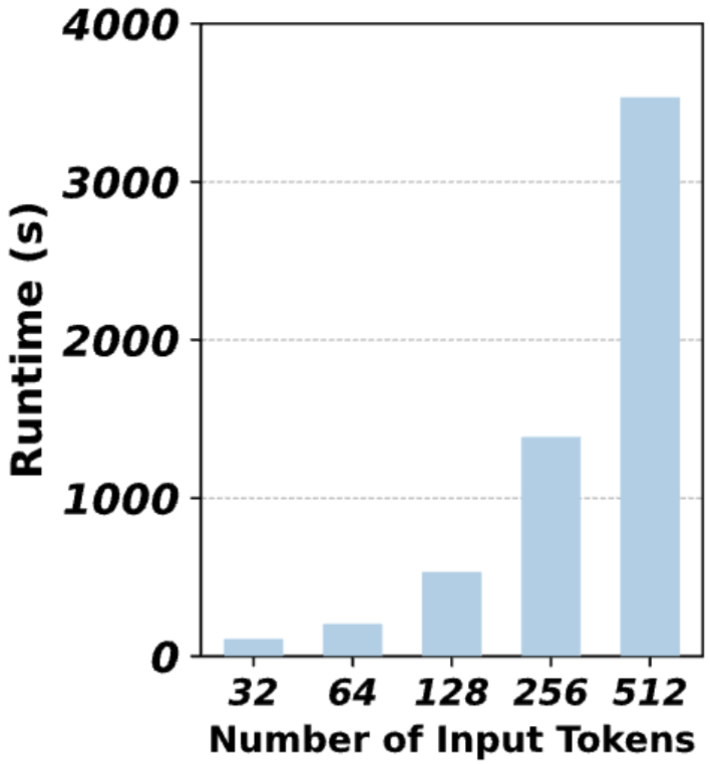
Direct data sharing leads to privacy breaches

*fake content generated by ChatGPT for presentation purpose only

Secure Two-party Transformer Inference

Homomorphic Encryption (HE) and Multi-party Computation (MPC) enables private Transformer inference



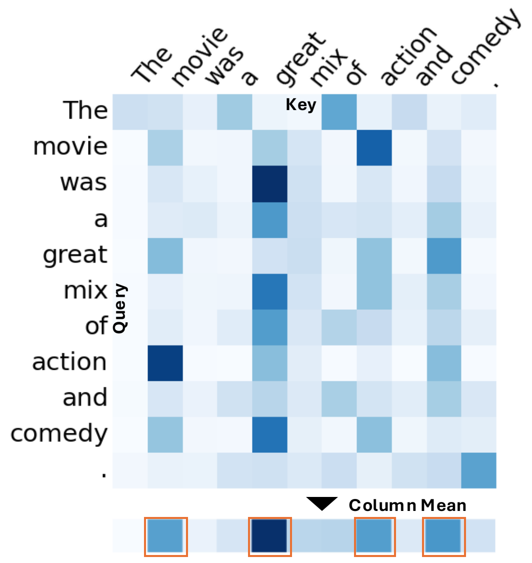


- By secretly sharing the input and intermediate features, the input and model privacy is protected.
- The efficiency and scalability of private Transformer inference is hindered by the quadratic complexity of attention module.

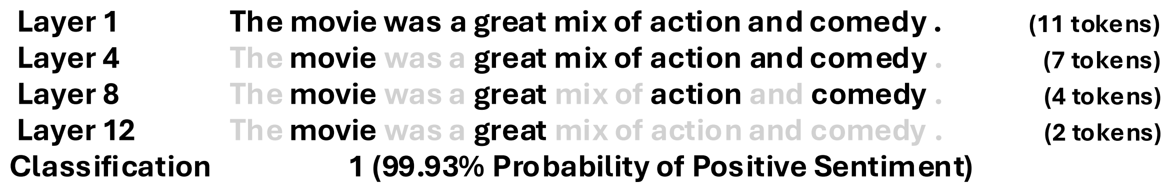
Our Motivation

Not all tokens are equally important.

➢ *Unimportant tokens* can be removed without degrading the model performance.



(a) Sample Attention Map



Layer 1	**The movie was a great mix of action and comedy .**	**(11 tokens)**
Layer 4	**The movie was a great mix of action and comedy .**	**(7 tokens)**
Layer 8	**The movie was a great mix of action and comedy .**	**(4 tokens)**
Layer 12	**The movie was a great mix of action and comedy .**	**(2 tokens)**
Classification	**1 (99.93% Probability of Positive Sentiment)**	

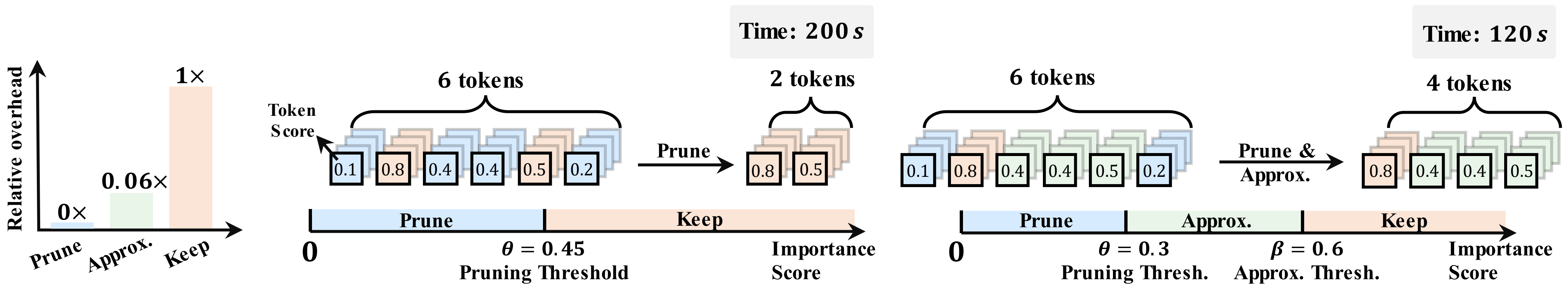
(b) Token Pruning for Sentiment Analysis

- The importance score can be computed from the attention map

- Unimportant tokens do not impact accuracy significantly and can be pruned during inference.

Our Motivation

- *Less important tokens* are important to the final performance and cannot be removed.
- Their computation can be approximated by low-degree polynomials during private inference



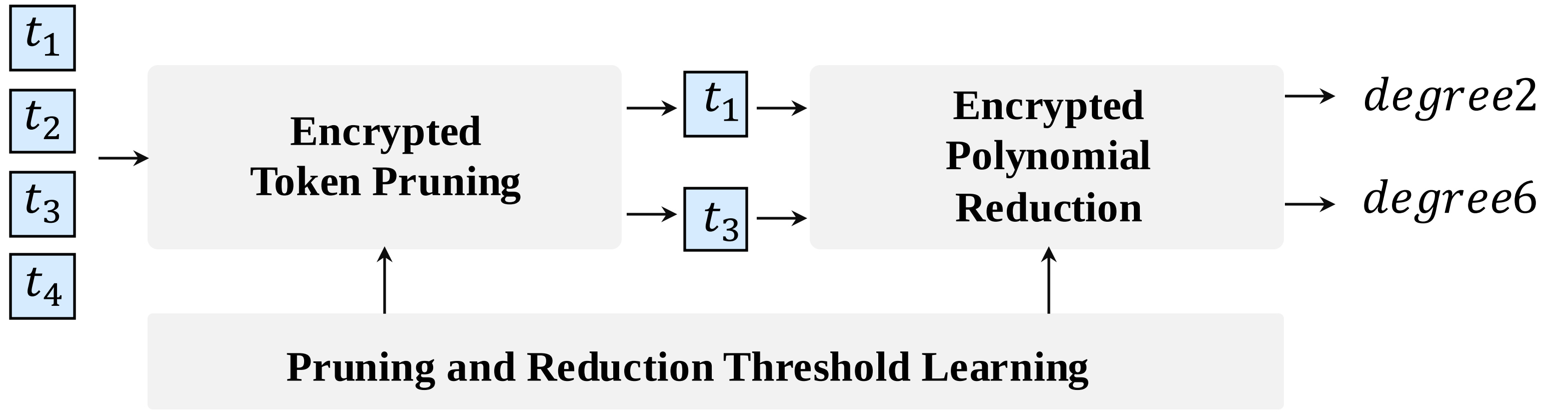
(a) Secure computation overhead (b) Token pruning (c) Token pruning with dynamic approximation

- Computing all remaining tokens with high-degree polynomials is expensive
- Approximating *Less important tokens* can preserve necessary information, while improving the efficiency

Problem Statement

During secure two-party Transformer inference, the server and client jointly:

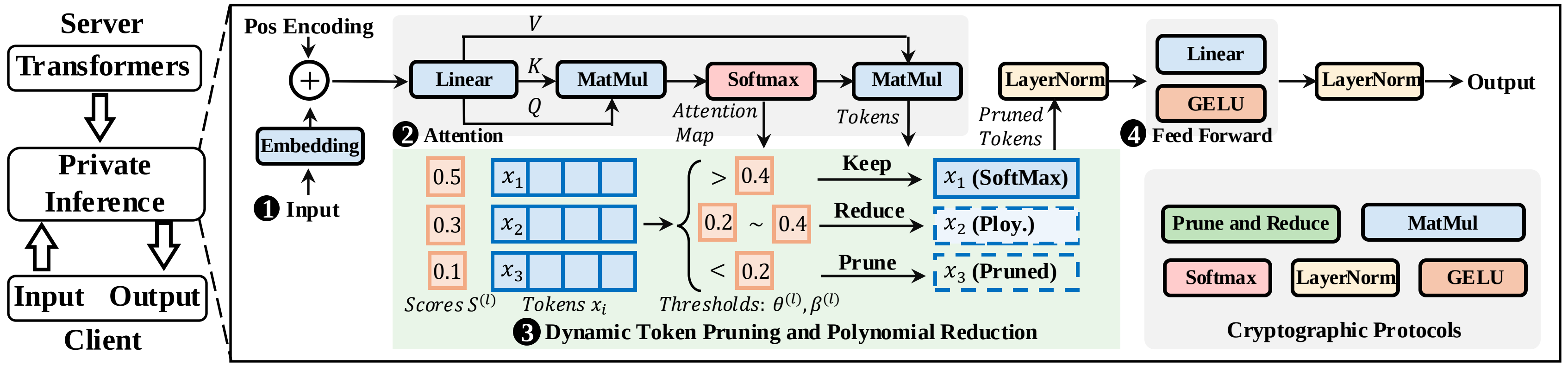
- Prune *unimportant tokens*
- Reduce the polynomial degree for *less important tokens*



- ➢ Security. The tokens and their location should be protected during token pruning.
- ➢ Accuracy. The token pruning and polynomial reduction should maintain high accuracy.
- ➢ Efficiency. The token pruning should speedup private Transformer inference.

CipherPrune Overview

The server and client jointly remove less important tokens, without leaking additional information.

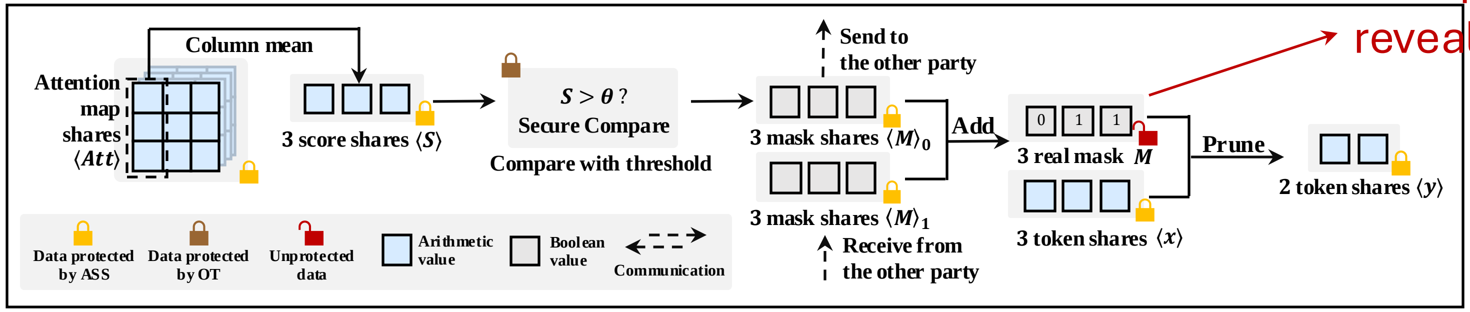


- The importance score can be computed locally over the secret shares
- After the SoftMax protocol, the Oblivious Transfer (OT) based pruning and mask protocols are invoked.
- Unimportant tokens are removed, and less important tokens are approximated.

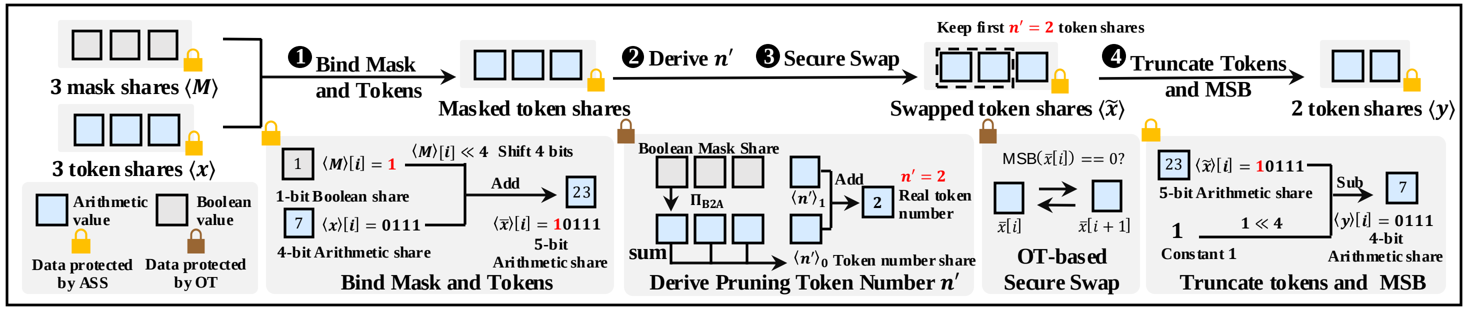
Encrypted Token Pruning

We can evaluate the importance of tokens privately and reveal the pruning mask.

The location of the unimportant tokens are revealed

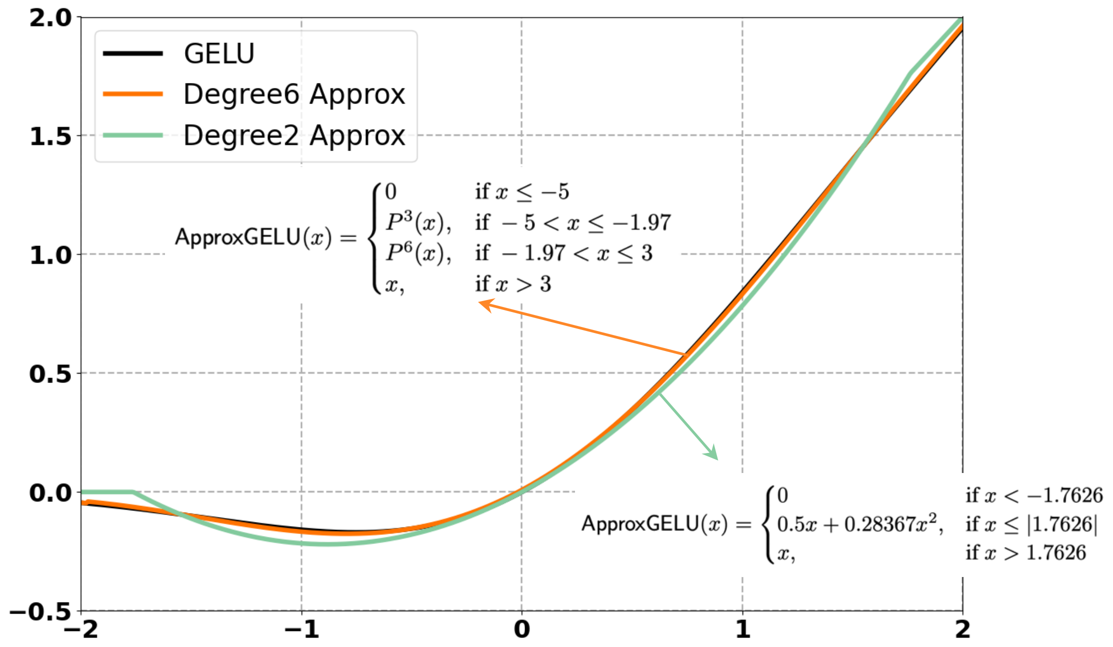


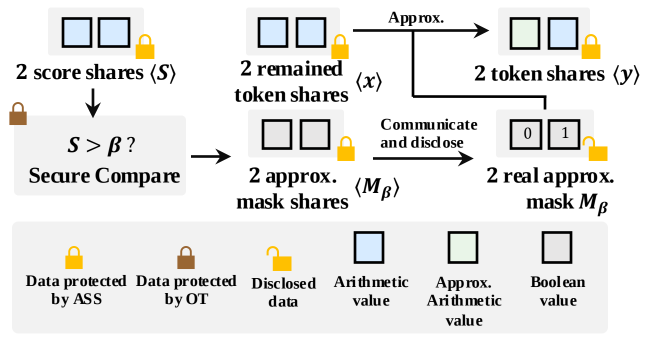
We can protect the pruning mask via mask-token binding and secure swap.



Encrypted Polynomial Reduction

For less important tokens, we can use low-degree polynomials to further reduce the overhead.





2 score shares $\langle S \rangle$
$S > \beta$?
Secure Compare
2 remained token shares $\langle x \rangle$
Approx.
2 token shares $\langle y \rangle$
2 approx. mask shares $\langle M_\beta \rangle$
Communicate and disclose
0
1
2 real approx. mask M_β
Data protected by ASS
Data protected by OT
Disclosed data
Arithmetic value
Approx. Arithmetic value
Boolean value

- Low-degree polynomials are less accurate, but much more efficient.
- The reduction mask can be computed in the same way as the pruning mask.

Pruning and Reduction Threshold Learning

The layer-wise pruning and reduction thresholds are jointly optimized thorough a fine-tuning.

- We set the soft pruning masks M_θ and reduction masks M_β

$$M_\theta^{(l)}(x_i) = \sigma(\frac{S^{(l)}(x_i) - \theta^{(l)}}{T}) \qquad M_\beta^{(l)}(x_i) = \sigma(\frac{S^{(l)}(x_i) - \beta^{(l)}}{T})$$

Where l is the layer index, S is the importance score, θ and β is learnable thresholds, σ is the Sigmoid function and T is a hyperparameter.

- We optimize the thresholds θ and β by setting the loss as:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda(\mathcal{L}_{prune} + \alpha\mathcal{L}_{approx.})$$

Where the pruning loss and approximation loss is determined by the L_1 norm of the masks.

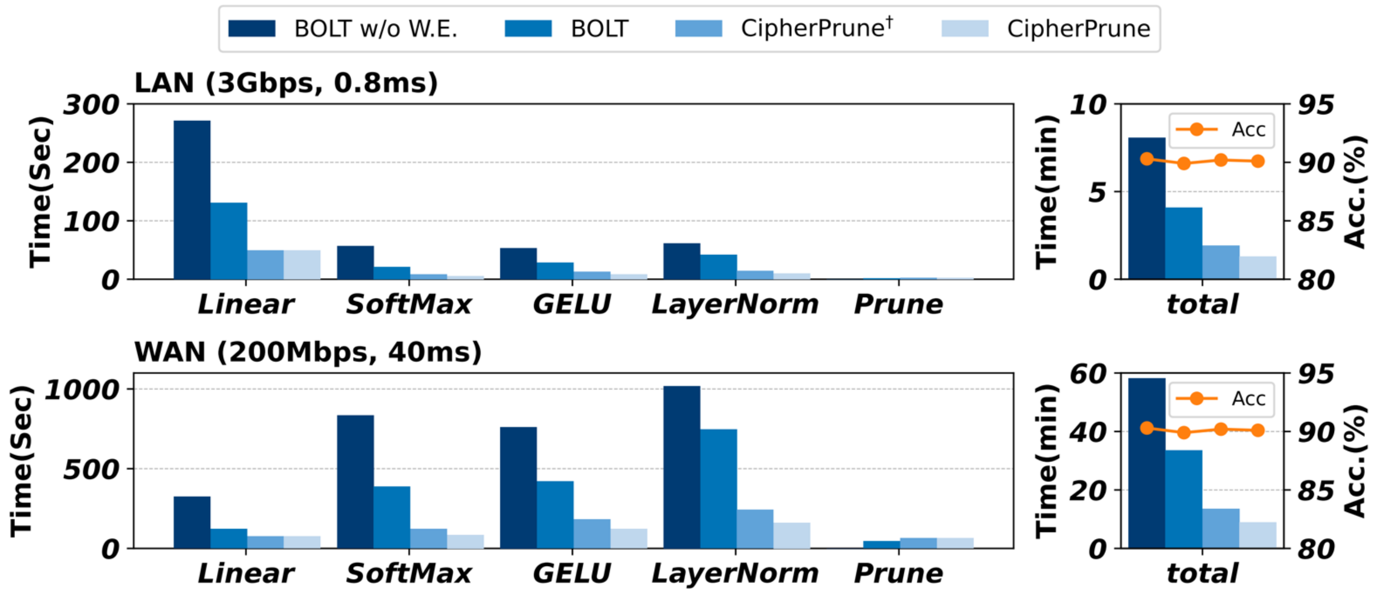
$$\mathcal{L}_{prune} = \frac{1}{L}\sum_{l=0}^{L-1}\left\|M_\theta^{(l)}(x)\right\|_1, \mathcal{L}_{approx.} = \frac{1}{L}\sum_{l=0}^{L-1}\left\|M_\beta^{(l)}(x)\right\|_1$$

Evaluation

CipherPrune improves the runtime and communication cost significantly, with only a marginal drop in accuracy.

Method	BERT Medium			BERT Base			BERT Large		
	Time	Comm.	Acc.	Time	Comm.	Acc.	Time	Comm.	Acc.
IRON (Hao et al., 2022)	442.4	124.5	$87.7_{\pm 0.2}$	1087.8	281.0	$90.4_{\pm 0.1}$	2873.5	744.8	$92.7_{\pm 0.1}$
BOLT w/o W.E. (Pang et al., 2024)	197.1	27.9	$87.4_{\pm 0.3}$	484.5	59.6	$90.3_{\pm 0.1}$	1279.8	142.6	$92.6_{\pm 0.2}$
BOLT (Pang et al., 2024)	99.5	14.3	$87.2_{\pm 0.3}$	245.4	25.7	$89.9_{\pm 0.3}$	624.3	67.9	$92.4_{\pm 0.2}$
CipherPrune	43.6	6.7	$87.4_{\pm 0.2}$	79.1	9.7	$90.1_{\pm 0.2}$	157.6	18.4	$92.5_{\pm 0.1}$

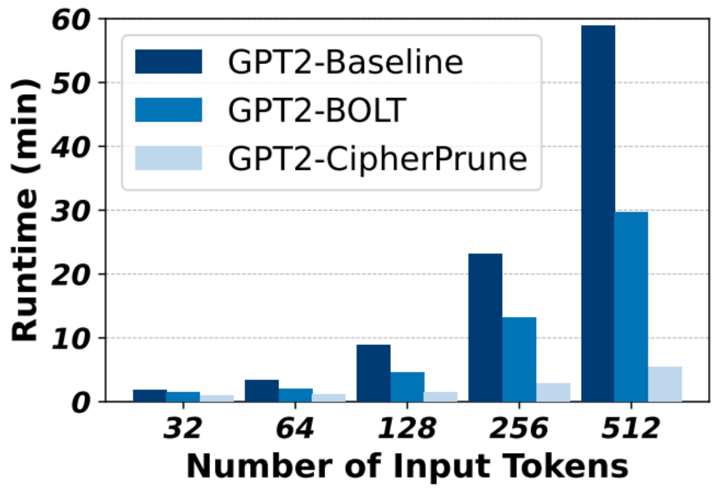
We breakdown the runtime on different protocols.



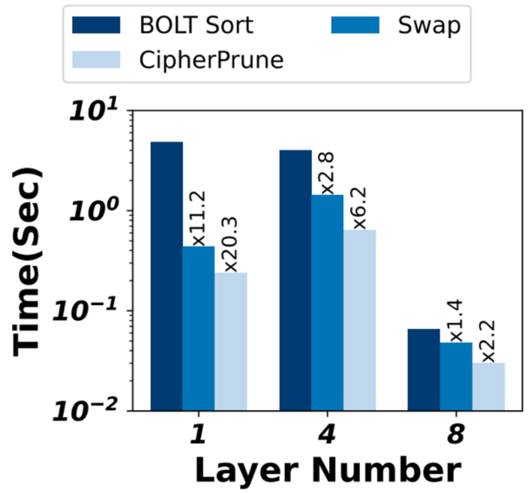
Evaluation

CipherPrune's layer wise pruning and reduction achieve better accuracy-efficiency trade-offs.

Method	Accuracy Metric on Tasks (%)				Time(Sec)
	MNLI	QNLI	SST2	MPRC	
BOLT w/o W.E.	84.75	90.32	91.74	90.53	484.5
BOLT	84.71	89.94	92.74	89.95	245.4
CipherPrune[†]	84.74	90.17	92.75	90.45	115.3
CipherPrune	84.68	90.11	92.66	90.18	79.1



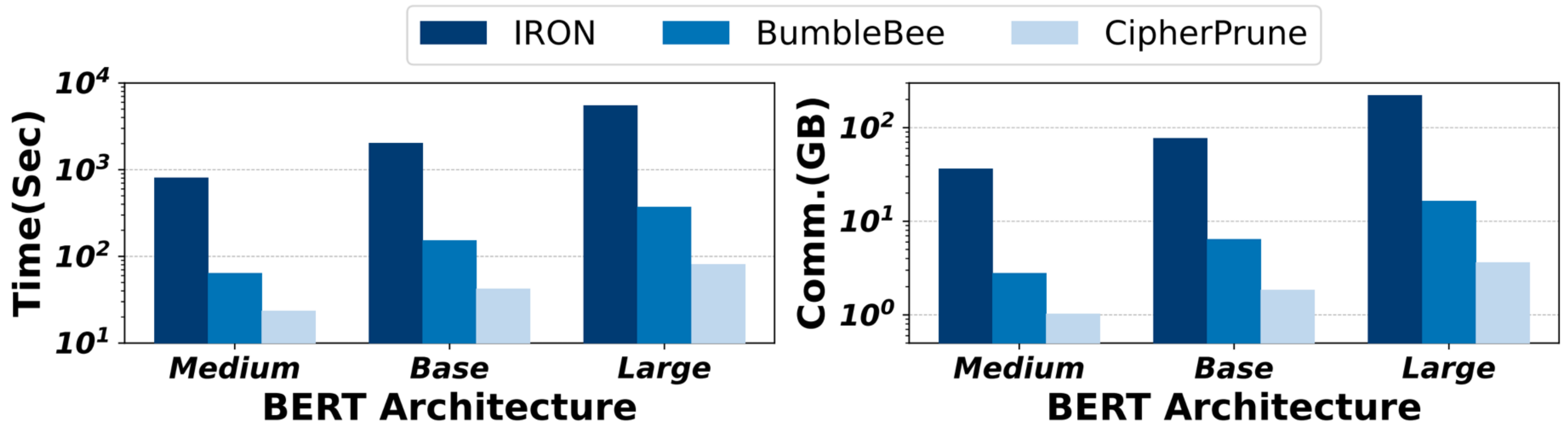
CipherPrune improves the quadratic complexity and is more scalable to long inputs.



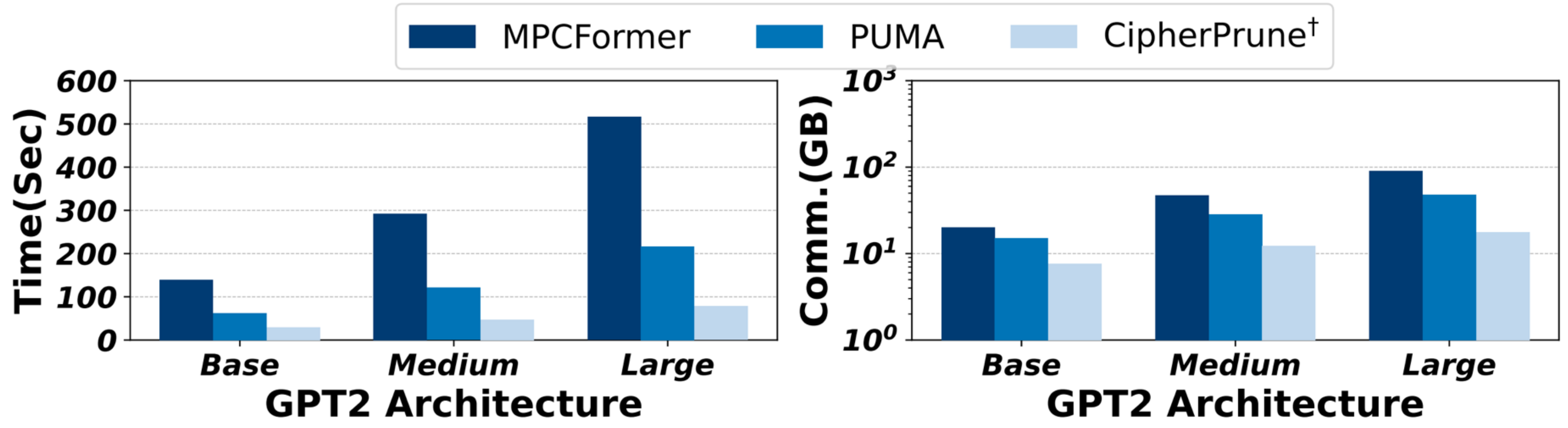
CipherPrune's pruning and masking protocols are more efficient than sorting-based methods

Evaluation

CipherPrune can be easily adapted to other MPC backends, such as SPU used by BumbleBee.



CipherPrune can also be easily adapted to the secure three-party computing setting as MPCFormer and PUMA.



Thank you!

Code

Paper