基于秘密分享的大语言模型密态推理

程珂[1,2]，夏昱珩[1]，代川云[1]，付家瑄[1]，祝幸辉[1,2]，沈玉龙[1,2]
（1.西安电子科技大学计算机科学与技术学院，陕西 西安 710126；
2.西安电子科技大学陕西省网络与系统安全重点实验室，陕西 西安 710071）

摘　要：大语言模型推理服务可能导致用户输入提示信息泄露给服务器端或专有模型权重泄露给用户。安全多方计算、同态加密等密码学技术为解决上述问题提供了可行方案，但由于计算和通信开销过大，在处理大语言模型推理任务时难以实际应用。基于此，提出了基于轻量级秘密分享的大语言模型密态推理方案，在不泄露用户输入和模型权重的前提下，高效精准地实现大语言模型推理。实验表明，相较现有先进工作，所提方案密态推理效率提升1.2~10倍，通信开销减少20%~90%。

关键词：隐私保护；大语言模型；秘密分享；安全多方计算；密态推理
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025115

Cryptographic inference for large language model via secret sharing

CHENG Ke[1,2], XIA Yuheng[1], DAI Chuanyun[1], FU Jiaxuan[1], ZHU Xinghui[1,2], SHEN Yulong[1,2]
1. School of Computer Science and Technology, Xidian University, Xi'an 710126, China
2. Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China

Abstract: Inference services based on large language models may lead to the leakage of user input hints to the server or proprietary model weights to the user. Cryptographic techniques such as secure multi-party computation and homomorphic encryption provide feasible solutions to the above problems, but they are still difficult to apply practically to the task of inference over large language models due to the excessive computational and communication overhead. Based on this, a lightweight secret-sharing-based cryptographic inference scheme for large language models was proposed, by which inference could be performed efficiently and accurately while ensuring that neither user inputs nor model parameters were revealed. The experimental results show that the proposed scheme improves the efficiency by 1.2~10 times and reduces the communication cost by 20%~90% compared with the existing state-of-the-art works.

Keywords: privacy-preserving, large language model, secret sharing, secure multi-party computation, cryptographic inference

收稿日期：2025–02–21；**修回日期：**2025–06–13
通信作者：祝幸辉，xhzhu@xidian.edu.cn
基金项目：国家重点研发计划基金资助项目（No.2023YFB3107500）；国家自然科学基金资助项目（No.62402358, No.62220106004, No.92467201）；陕西省自然科学基础研究计划基金资助项目（No.2025JC-YBQN-869）；陕西省技术创新引导计划（基金）资助项目（No.2023KXJ-033）；山东省重点研发计划基金资助项目（No.2023CXPT056）；陕西省科学技术协会青年人才托举计划基金资助项目（No.20240138）；香港狮子山网络空间安全实验室研究课题基金资助项目（No.LRL24004）；中央高校基本科研业务费专项资金资助项目（No.ZDRC2202, No.KYFZ25005）

Foundation Items: The National Key Research and Development Program of China (No.2023YFB3107500), The National Natural Science Foundation of China (No. 62402358, No. 62220106004, No. 92467201), The Natural Science Basic Research Program of Shaanxi Province (No. 2025JC-YBQN-869), The Technology Innovation Leading Program of Shaanxi Province (No. 2023KXJ-033), The Key Research and Development Program of Shandong Province of China (No. 2023CXPT056), The Young Talent Fund of Association for Science and Technology in Shaanxi Province (No. 20240138), The Open Topics from the Lion Rock Labs of Cyberspace Security (No.LRL24004), The Fundamental Research Funds for the Central Universities (No.ZDRC2202, No.KYFZ25005)

0　引言

基于 Transformer 结构的大语言模型（LLM, large language model）在机器翻译、文本摘要、智能问答和情感分析等任务上效果显著。相较传统循环神经网络，Transformer 模型以其独特的注意力机制处理输入序列、捕捉长距离依赖关系，同时利用并行计算的特性缩短训练时间以提升实时数据处理能力[1]。以 ChatGPT 为代表的大语言模型推理服务发展迅速，服务提供商将训练好的大语言模型部署在云端，用户上传输入数据即可获取推理结果。然而，一方面，用户的输入数据可能包含身份、医疗、金融等敏感信息，若这些数据在传输和推理过程未充分保护，可能导致用户隐私泄露；另一方面，由模型持有者直接向用户提供模型权重的方式也不切实际，因为模型权重通常属于公司机密，并且算力较低的用户终端难以支撑大语言模型的部署与执行。

安全多方计算、同态加密等密码学技术为解决上述问题提供了可行方案，神经网络密态推理也因此成为研究热点。该技术通过在加密的用户输入和模型权重上执行推理任务，确保敏感信息不会在传输和推理过程中泄露。文献[2-12]基于秘密分享、同态加密等密码学原语设计了隐私保护的卷积神经网络推理方案，对于 LeNet、AlexNet 等小型神经网络模型，能够实现单张图片的毫秒级密态推理。

然而，不同于传统神经网络，Transformer 结构包含大量 GELU（Gaussian error linear unit）、Softmax 和 LayerNorm 等复杂非线性函数，导致基于密码学的大语言模型密态推理仍面临诸多挑战。文献[13]直接利用安全多方计算进行处理开销较大，在 Bert-Base 模型上单次密态推理时间超过 10 min。为了降低计算开销，文献[14-15]删除或替换部分复杂的非线性操作，但改变模型结构导致推理精度降低。文献[16-17]采用近似方法将非线性计算转化为线性计算，并基于同态加密、混淆电路等技术进行方案设计，可以在一定程度上保证较高的计算精度，但仍会产生大量计算和通信开销。

本文提出了基于算术秘密分享的大语言模型密态推理方案，在保证数据隐私的同时，提高安全计算效率和推理准确性。主要贡献如下。

1)基于算术秘密分享设计了安全基础计算协议，包括安全查找表协议、安全分享模式转换协议和安全除法协议。所提协议与现有查找表方案[18]相比，在线阶段计算效率提高了 10~20 倍；与现有分享模式转换协议[2]相比，通信开销降低了 50%；与现有安全除法协议[4]相比，通信轮次和开销减少了 70%。

2)将复杂非线性函数安全计算问题转换为查找表问题，设计了通信高效的 GELU 和 LayerNorm 函数安全计算协议，与现有先进工作相比，安全 GELU 函数的通信开销减少了 90%~95%，安全 LayerNorm 函数的计算效率提高了 2.5 倍，通信开销减少了 60%。

3)基于上述协议构建大语言模型密态推理方案，在密码学半诚实模型下证明了方案的安全性。在 Bert-Base、Bert-Large 和 GPT-2 等大语言模型上进行了实验验证，与现有先进工作相比，所提方案计算效率提升了 1.2~10 倍，通信开销减少了 20%~90%。

1　相关工作

神经网络推理服务所面临的数据安全问题受到广泛关注，国内外研究团队针对神经网络密态推理展开了一系列研究，表 1 对比了主流的隐私保护神经网络模型推理方案，其中，SS 表示秘密分享，FSS 表示函数秘密分享，HE 表示同态加密，FHE 表示全同态加密，MPC 表示安全多方计算，CNN 表示卷积神经网络。现有研究主要关注 CNN 等传统神经网络的密态推理。Demmler 等[19]提出安全两方计算框架 ABY（Arithmetic-Boolean-Yao），支持算术、布尔、Yao 等 3 种秘密分享密码学原语，并允许这些原语之间可相互转换。在 ABY 的基础上，后续发展了性能更优的 ABY2.0[20]以及支持安全三方计算的 ABY3[3]。Liu 等[2]在此基础上设计了基于算术秘密分享的安全外包神经网络推理协议，通过算术秘密分享实现数据加密和密文分发，确保外包计算环境下数据的隐私性和完整性。CrypTen[4]是 Facebook 开发的面向深度神经网络的安全多方计算平台，其底层依赖于深度学习框架 PyTorch，并支持加法秘密分享和布尔秘密分享的向量化执行。CryptGPU 在此基础上利用 GPU 加速所有运算操作，显著提高了神经网络在密文状态下的训练和推理效率[5]。Cryptflow2[6]基于布尔秘密分享和加法秘密分享设计了一系列交互式协议，实现了 ReLU 层、最大池化层等非线性层的安全计算。基于上述子协议构造的神经网络安全计算协议取得了较优的

表 1 国内外主流的隐私保护神经网络模型推理方案对比

方案	技术路线	参与方数量	网络模型	激活函数	安全计算效率	通信开销
Liu 等[2]	SS	2	CNN	ReLU	较低	较高
CrypTen[4]	SS	2/3	CNN、Bert、GPT	ReLU、GELU、Softmax	较高	较低
CryptGPU[5]	SS	2/3	CNN、Bert、GPT	ReLU、GELU、Softmax	较高	较低
Cryptflow2[6]	SS	2	CNN	ReLU	较高	较高
AriaNN[7]	FSS	2	CNN	ReLU	较高	较低
SecureML[8]	SS	2	CNN	ReLU、Softmax	较低	较高
Falcon[9]	SS	3	CNN	ReLU	较高	较低
任艳丽等[12]	HE	2	CNN	ReLU	较低	较低
THE-X[14]	HE	2	Bert	ReLU	较低	较高
MPCFormer[15]	SS	2	Bert	GELU	较低	较低
Iron[13]	HE、MPC	2	Bert	GELU、Softmax	较低	较高
BOLT[16]	HE	2	Bert	GELU、Softmax	较低	较高
BumbleBee[21]	HE	2	Bert	GELU、Softmax	较低	较低
SecForme[17]	SS	2	Bert	GELU	较高	较高
Zhang 等[22]	FHE	2	Bert	GELU、Softmax	较低	无
本文方案	SS	2	Bert、GPT	GELU、Softmax	较高	较低

推理速度和准确性，但 Cryptflow2 实质上通过增加交互轮次来降低整体计算量，对通信稳定性和通信速率具有较高要求。为降低非线性层安全计算的通信轮次，AriaNN[7]基于函数秘密分享设计了常数级通信轮次的 ReLU、最大池化层安全计算协议。SecureML[8]设计了基于秘密共享的定点数算术计算协议，提出 Sigmoid 的分段计算和 Softmax 的近似计算方法。Falcon[9]基于秘密分享设计三方非线性函数安全计算协议，并支持半诚实大多数威胁模型下的隐私保护神经网络推理。任艳丽等[12]基于 CKKS（Cheon-Kim-Kim-Song）同态加密设计了密态卷积神经网络推理方案，降低了在线阶段的通信开销。

相较传统神经网络模型，Transformer 模型权重更大，在密态数据上实现 Transformer 模型推理更具挑战。THE-X[14]使用 ReLU 函数和多项式的组合替换 GELU 激活层、Softmax 层和 LayerNorm 层，进而评估同态加密下的 BERT-tiny 模型，但该方案在较小的模型上需要较大的计算开销，且存在较大的精度损失，并且每个 ReLU 函数计算结果会公开给用户，存在隐私泄露风险。MPCFormer[15]提出了一种基于算术秘密分享的 BERT-base 模型推理系统，采用知识蒸馏解决非线性函数密态计算引起的精度下降问题。尽管该方案在大型数据集上表现尚可，但在 RTE 等小规模数据集上蒸馏模型的泛化能力不足，模型精度下降幅度超过 5%，限制了其实用性。此外，该方案需要在原模型训练完成后进行额外的蒸馏过程，增加了系统复杂性和训练成本。Iron[13]结合同态加密和安全多方计算设计了矩阵乘法协议，并且对非线性函数进行优化以保证计算的安全性和准确率。BOLT[16]和 BumbleBee[21]在此基础上利用同态加密优化了安全矩阵乘法协议，降低了线性计算的通信开销，然而对于非线性函数，其仍依赖于高次多项式近似拟合，导致密态推理计算和通信开销仍然较大。SecFormer[17]基于分段多项式和傅里叶级数处理 Transformer 模型中的复杂非线性层，提高了安全计算的效率，但计算过程仍会产生大量通信开销，且需要对训练好的模型进行额外的蒸馏，增加了部署难度，难以应用于实际场景。Zhang 等[22]基于残数数制（RNS）-CKKS 全同态加密技术提出了第一个非交互的 Transformer 模型推理方案，其中非线性函数安全计算无须通信开销，但该方案在 Bert-Base 等模型上密态推理时需要大量计算开销，导致推理时延较大。

2　预备知识

2.1　系统架构和威胁模型

本文采用如图1所示的双云外包计算架构[2,23]，用户和模型持有者在本地对输入数据和模型权重等隐私信息进行算术秘密分享，并将分享份额分别发送给2台云服务器，云服务器之间在秘密分享数据上执行一系列安全交互计算协议完成大语言模型推理，并将秘密分享的推理结果返回给用户。

安全计算协议在半诚实安全模型[19-20]下执行，即参与方会按照协议规定忠实地执行各项操作，但会试图窥探或者推断与原始数据相关的隐私信息。在该安全模型下，攻击者不会主动破坏协议的执行过程，但会通过在协议执行过程中接收到的所有信息来推断其他参与方的私密信息。

2.2　算术秘密分享

算术秘密分享（ASS, arithmetic secret sharing）[19]是一种轻量级秘密分享方案，本文方案基于安全两方计算模型构建，各参与方记为$P=\left(P_0,P_1\right)$，主要使用(2,2)−算术秘密分享模式。

(2,2)−分享（记作$[\![\cdot]\!]$）。在$[\![\cdot]\!]$分享形式下，环$\mathbb{Z}_{2^n}$上的一个秘密x被2个参与方分享，每个参与P_i持有秘密分享份额$[\![x]\!]_i$，其中$i\in\{0,1\}$，且满足$x=[\![x]\!]_0+[\![x]\!]_1$，$x=x_0+x_1$用于表示上述过程。除非另有说明，所有计算都是在环$\mathbb{Z}_{2^n}$上进行的。为了描述简洁，将在描述中省略$\bmod 2^n$。

基于上述(2,2)−分享技术，可在半诚实模型下实现以下基础协议（协议中$i\in\{0,1\}$）。

1)分享协议（记作share）：参与方P_i在环$\mathbb{Z}_{2^n}$生成一个随机数r，令$[\![x]\!]_i=x-r$，并将随机数r发送给参与方P_{1-i}。参与方P_{1-i}令$[\![x]\!]_{1-i}=r$。

2)恢复协议（记作restore）：参与方P_i将分享份额$[\![x]\!]_i$发送给参与方P_{1-i}，后者计算$x=[\![x]\!]_i+[\![x]\!]_{1-i}$以恢复原始数据$x$。

3)安全加法协议：给定秘密分享值$[\![x]\!]$和$[\![y]\!]$，输出其加和结果$[\![x+y]\!]$。具体来说，参与方P_i本地计算$[\![x+y]\!]_i=[\![x]\!]_i+[\![y]\!]_i$。此外，对于共同持有的常数$c$，$[\![x+c]\!]$只需要计算$\left(x_0+c,x_1\right)$即可得到正确的输出，该过程可由$P_i$在本地计算得到。

4)安全乘法协议：给定秘密分享值$[\![x]\!]$和$[\![y]\!]$，输出其乘积结果$[\![x\cdot y]\!]$，可通过乘法三元组（Beaver三元组）[24]实现该安全计算协议。具体来说，参与方P_i持有预先生成的三元组$\left([\![a]\!]_i,[\![b]\!]_i,[\![c]\!]_i\right)$，满足$c=a\cdot b$。各参与方在本地计算$[\![e]\!]_i=[\![x]\!]_i-[\![a]\!]_i$和$[\![f]\!]_i=[\![y]\!]_i-[\![b]\!]_i$，随后，双方执行restore($e$)和restore($f$)以恢复$e$，$f$。各参与方计算$[\![x\cdot y]\!]=i\cdot e\cdot f+[\![a]\!]_i\cdot f+[\![b]\!]_i\cdot f+[\![c]\!]_i$得到最终结果。

乘法协议中的乘法三元组可由可信第三方提前生成并分发给计算的参与方，也可由计算参与方通过同态加密或不经意传输提前生成，具体生成协议参考文献[19]。需要说明的是，当环为$\mathbb{Z}_2$时，若将加减法替换为异或（⊕）操作，乘法替换为与（∧）操作，则上述秘密分享形式称为布尔秘密分享或布尔分享，用$[\![\cdot]\!]^{\mathrm{B}}$表示，对应的分享和恢复协议

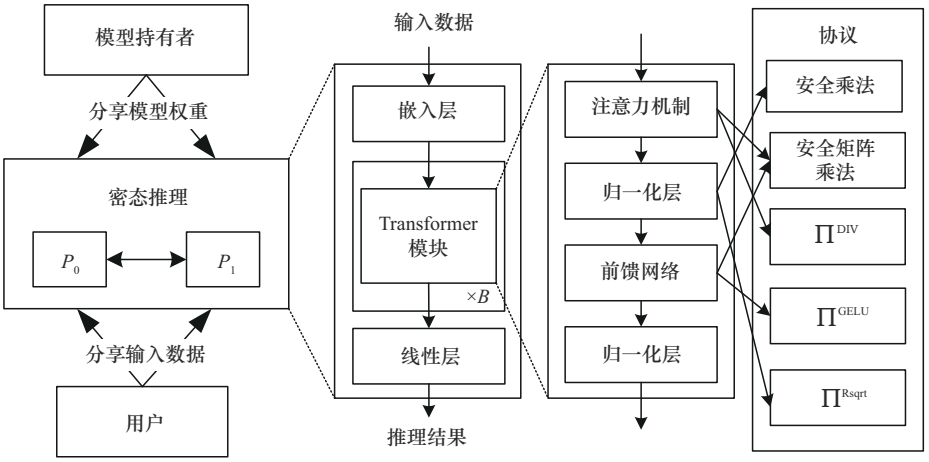

图1　系统模型

记为 share^{B} 和 restore^{B}。

2.3 Transformer 模型

Transformer 模型是 Vaswani 等[1]提出的用于自然语言处理和其他序列到序列任务的深度学习模型架构，如图 2 所示，由编码器（左侧）和解码器（右侧）组成。编码器负责深入理解输入文本，为每个输入构建相应的语义表示。解码器则在编码器输出的语义基础上，结合其他输入信息生成目标序列。BERT[25]是一个仅包含编码器的 Transformer 模型，在 GLUE（general language understanding evaluation）基准上超越了当时所有先进模型；GPT 则是一个仅包含解码器的 Transformer 模型，广泛应用于智能问答等任务[26]。典型的 Transformer 模块主要包含以下部分。

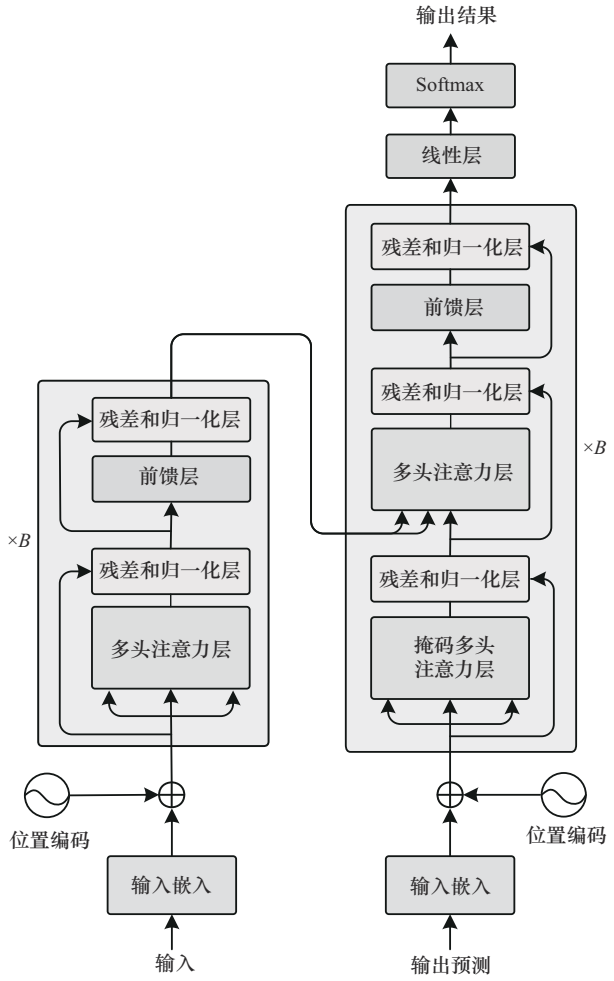

图 2 Transformer 模型架构

1)符号嵌入。Transformer 将自然语言转换为一个记号序列，每个记号是一个大小为 d_{model} 的一维向量。通过词嵌入矩阵将记号映射到相应的嵌入向量中。

2)自注意力机制。自注意力机制的核心是捕捉向量之间的相关性，使模型关注输入序列中的不同部分并分配不同的注意力权重，从而在处理序列的同时考虑所有位置的信息。自注意力机制按以下方式将输入的查询矩阵和一组键值对映射到输出

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_{\text{model}}}}\right)\boldsymbol{V} \tag{1}$$

其中，$\boldsymbol{Q}$ 是查询矩阵，$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 分别是键矩阵和值矩阵。对于输入向量 $\boldsymbol{x} \in \mathbb{R}^k$，令 $x_{\max} = \max\left(x_0,x_1,\cdots,x_{k-1}\right)$，则 Softmax 函数的输出向量 $\boldsymbol{y} \in \mathbb{R}^k$ 为

$$y_i = \frac{\mathrm{e}^{x_i}}{\sum_{j=0}^{k-1}\mathrm{e}^{x_j}} = \frac{\mathrm{e}^{x_i - x_{\max}}}{\sum_{j=0}^{k-1}\mathrm{e}^{x_j - x_{\max}}} \tag{2}$$

多头注意力（MHA, multi-head attention）模块由多个并行的自注意力模块组成。

3)前馈网络（FFN, feed forward network）。前馈网络由 2 个全连接层和一个激活函数组成，用于捕捉序列中的非线性关系。对于输入矩阵 $\boldsymbol{X}$，FFN 定义为

$$\text{FFN}(\boldsymbol{X}) = \text{GELU}(\boldsymbol{X}W_1 + b_1)W_2 + b_2 \tag{3}$$

其中，W_1、W_2 和 b_1、b_2 分别是第一个和第二个全连接层的权重和偏移量，GELU 是高斯误差线性单元激活函数，定义为

$$\text{GELU}(x) = 0.5x\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}\left(x+0.044\,715x^3\right)\right)\right) \tag{4}$$

4)层归一化（layer normalization）。层归一化用于规范化神经网络各层的激活分布，对于输入向量 $\boldsymbol{x} \in \mathbb{R}^k$，令 $m = \frac{\sum x_i}{k}$ 和 $v = \frac{\sum\left(x_i - m\right)^2}{k}$ 表示均值和方差。记模型参数为 γ，β，$z_i = x_i - m$，$i \in \{0,1\}$，则层归一化的输出向量 $\boldsymbol{y} \in \mathbb{R}^k$ 为

$$y_i = \gamma\frac{x_i - m}{\sqrt{v}} + \beta = \gamma\frac{z_i}{\sqrt{\frac{\sum z_i^2}{k}}} + \beta \tag{5}$$

3 大语言模型密态推理方案

大语言模型基于 Transformer 模型构建，由一系列线性层和非线性层组成，线性层主要包含嵌入层、矩阵乘法等线性操作，非线性层主要包含

GELU、Softmax、LayerNorm 等非线性函数计算。本文直接采用文献[2,19]中基于算术秘密分享的安全计算协议实现 Transformer 模型中的线性层，因此本节主要针对 Transformer 模型的非线性函数进行安全计算协议设计。这些协议可以按照适当尺寸顺序拼接，每一层的输入和输出都是加法秘密分享形式，并且都在相同的环 $\mathbb{Z}_{2^n}$ 上，因此可以组成任意的 Transformer 模型。接下来，介绍本文提出的安全基础计算协议，并在此基础上设计安全 GELU 激活函数、安全 Softmax 函数和安全 LayerNorm 函数等安全非线性层计算协议。

3.1　安全基础计算协议

3.1.1　安全查找表协议

查找表协议通过预计算所有可能的函数值，从而实现函数的快速求解。对于输入数据 $x \in \mathbb{Z}_M$，安全查找表协议输出为 $\boldsymbol{T}[x] \in \mathbb{Z}_L$，其中，$\boldsymbol{T}[x] \in \mathbb{Z}_L^M$ 为公共的查找表，即函数的值域，$\mathbb{Z}_M$ 为输入数据的取值范围，$\mathbb{Z}_L$ 为输出数据的取值范围。通过内积计算实现查找表算法，对于各参与方，给定输入数据 $[\![x]\!]$ 和公共查找表 $\boldsymbol{T}$，双方计算向量 $[\![\boldsymbol{u}]\!]$ 满足 $\boldsymbol{u}$ 为 x 的独热（one-hot）向量，即仅在 x 处为 1，其余位置为 0，可以通过计算 $[\![\boldsymbol{T}[x]]\!] = [\![\boldsymbol{u} \odot \boldsymbol{T}]\!]$ 得到查找表结果。表 2 给出了查找表算法样例，给定函数 $f(x) = 2x + 1$，$x \in [0,7]$，预计算所有函数值存入表 $\boldsymbol{T}$ 中。为求解 x=5 对应的函数值，构建 x 的 one-hot 向量 $\boldsymbol{u}$，将 $\boldsymbol{u}$ 与 $\boldsymbol{T}$ 对应元素相乘并求和即可得到计算结果，即 $\boldsymbol{T}[x] = \boldsymbol{u} \odot \boldsymbol{T} = \sum_{i=0}^{8} \boldsymbol{u}[i] \cdot \boldsymbol{T}[i] = 11$。

表 2　查找表算法样例

x	$\boldsymbol{T}[x]$	$\boldsymbol{u}$
0	1	0
1	3	0
2	5	0
3	7	0
4	9	0
5	11	1
6	13	0
7	15	0

安全查找表协议如算法 1 所示，密钥生成阶段由可信第三方执行，获取输入数据的取值范围并在该范围内生成随机数 r，产生 r 的 one-hot 向量 $\boldsymbol{v}$，并将 r 和 $\boldsymbol{v}$ 分享给 2 个计算参与方。在线评估阶段，参与方获取 $[\![x]\!]$，计算 $[\![\hat{x}]\!]_i = [\![x]\!]_i - [\![r]\!]_i$ 并恢复 $\hat{x}$，通过计算 $[\![\boldsymbol{u}[j]]\!]_i = [\![\boldsymbol{v}[(j - \hat{x}) \bmod M]]\!]_i$ 得到向量 $[\![\boldsymbol{u}]\!]$，使 $\boldsymbol{u}[j] = 1$ 当且仅当 $j = x$ 时成立。各参与方可通过 $[\![\boldsymbol{u}]\!]$ 和 $\boldsymbol{T}$ 得到查表结果。

算法 1　安全查找表协议 $\prod^{\mathrm{LUT}}$

输入　P_i 持有 $[\![x]\!]_i$ 和 $\boldsymbol{T}$

输出　$[\![z]\!] = [\![\boldsymbol{T}[x]]\!]$

密钥生成阶段 输入 M

1) $r \leftarrow \mathrm{rand}(M)$

2) 计算向量 $\boldsymbol{v}$ 满足 $\boldsymbol{v}[r] = 1$ 且对任意 $j \neq r$，$\boldsymbol{v}[r] = 0$

3) $[\![\boldsymbol{v}]\!]_0, [\![\boldsymbol{v}]\!]_1 = \mathrm{share}(\boldsymbol{v})$

4) $[\![r]\!]_0, [\![r]\!]_1 = \mathrm{share}(r)$

5) for $i \in \{0,1\}$，$k_i = [\![r]\!]_i \| [\![\boldsymbol{v}]\!]_i$

在线执行阶段

1) $[\![r]\!]_i \| [\![\boldsymbol{v}]\!]_i \leftarrow k_i$

2) P_i 在本地计算 $[\![\hat{x}]\!]_i = [\![x]\!]_i - [\![r]\!]_i$

3) 恢复 $\hat{x} = \mathrm{restore}([\![\hat{x}]\!]_i)$

4) 令 $[\![\boldsymbol{u}[j]]\!]_i = [\![\boldsymbol{v}[(j - \hat{x}) \bmod M]]\!]_i$

5) P_i 计算 $[\![\boldsymbol{T}[x]]\!]_i = \sum_{j=0}^{M} [\![\boldsymbol{u}[j]]\!]_i \cdot \boldsymbol{T}[j]$

6) P_i 输出 $[\![z]\!]_i = [\![\boldsymbol{T}[x]]\!]_i$

3.1.2　安全分享模式转换协议

安全分享模式转换协议在不泄露原始输入数据的情况下将单比特布尔秘密分享数据 $[\![x]\!]^{\mathrm{B}}$ 转为算术秘密分享 $[\![x]\!]$，其中 $x \in \{0, 1\}$，记为 $\prod^{\mathrm{B2A}}$，具体实现如算法 2 所示。

算法 2　安全分享模式转换协议 $\prod^{\mathrm{B2A}}$

输入　P_i 持有 $[\![x]\!]_i^{\mathrm{B}}$

输出　$[\![x]\!]_i$

密钥生成阶段

1) $r \leftarrow \mathrm{rand}(\{0,1\})$

2) $[\![r]\!]_0 [\![r]\!]_1 = \mathrm{share}(r)$

在线执行阶段

1) 计算 $[\![\hat{x}]\!]_i^{\mathrm{B}} = ([\![x]\!]_i^{\mathrm{B}} + [\![r]\!]_i) \bmod 2$

2)恢复 $\hat{x} = \text{restore}^{\text{B}}\left(\left[\!\left[\hat{x}\right]\!\right]_i^{\text{B}}\right)$

3)$\left[\!\left[x\right]\!\right]_i = i \cdot \hat{x} + \left[\!\left[r\right]\!\right]_0 - 2 \cdot \left[\!\left[r\right]\!\right]_0 \cdot \hat{x}$

密钥生成阶段在 0 和 1 之间产生随机数，并将其分享给 2 个计算参与方。在线执行阶段，各参与方本地计算 $\left[\!\left[\hat{x}\right]\!\right]_i^{\text{B}} = \left(\left[\!\left[x\right]\!\right]_i^{\text{B}} + \left[\!\left[r\right]\!\right]_i\right) \bmod 2$ 盲话输入 $\left[\!\left[x\right]\!\right]_i^{\text{B}}$ 并公开 $\hat{x}$，计算 $\left[\!\left[x\right]\!\right]_i = i \cdot \hat{x} + \left[\!\left[r\right]\!\right]_0 - 2 \cdot \left[\!\left[r\right]\!\right]_0 \cdot \hat{x}$ 得到算术秘密分享结果。

3.1.3 安全除法协议

除法是 Softmax 函数的重要组成部分之一，为求解 $\frac{x}{y}$，可利用近似公式求解 y 的倒数 $\frac{1}{y}$，再结合乘法求解除法结果。对于在[0.5, 1)区间内的数值 b，倒数的近似公式为

$$\frac{1}{b} \approx w\left(1 + \varepsilon_0\right)\left(1 + \varepsilon_1\right) \tag{6}$$

其中，$w = 2.914\,2 - b$，$\varepsilon_0 = 1 - bw$，$\varepsilon_1 = \varepsilon_0^2$。由于该近似公式要求数值处于[0.5, 1)区间内，除数 y 需要被标准化处理为[0.5, 1)区间内的数据，对于任意正整数 y，一定存在一个整数 k，使 $2^k \leqslant y < 2^{k+1}$。因此，标准化 y 的过程可转化为确定整数 k，再将 y 与 $\frac{1}{2}^{k+1}$ 相乘。对于 k 的求解，将除数 $\left[\!\left[y\right]\!\right]$ 与 $2^1, 2^2, \cdots, 2^{n-1}$ 分别比较大小，得到比较结果 $\left[\!\left[d_j\right]\!\right]$，$j \in [1, n-1]$，各参与方将 $\left[\!\left[d_j\right]\!\right]$ 求和即可得 $\left[\!\left[k\right]\!\right]$。例如，$y = 7$，其对应的 $k = 2$，$y \geqslant 2^j$ 的比较结果序列 d_j 为 $\{1, 1, 0, 0, \cdots, 0\}$，$j \in [1, n-1]$，满足 $\sum_{j=1}^{n-1} d_j = 2 = k$。$\frac{1}{2}^{k+1}$ 可通过构建 $2^{-(k+1)}$ 的查找表，利用查找表算法求解。计算 $\left[\!\left[\frac{x}{2}^{k+1}\right]\!\right] = \left[\!\left[x\right]\!\right] \cdot \left[\!\left[2^{-(k+1)}\right]\!\right]$ 和 $\left[\!\left[\frac{y}{2}^{k+1}\right]\!\right] = \left[\!\left[y\right]\!\right] \cdot \left[\!\left[2^{-(k+1)}\right]\!\right]$，完成对输入数据的标准化。上述过程中比较大小操作针对 $\left[\!\left[y\right]\!\right]$ 的偏移量执行，因此可使用相同的密钥进行计算。本文采用 SIGMA[27] 中基于分布式点函数（DPF, distributed point function）[28] 的比较大小方案 DReLU，当输入为非负时，DReLU 输出 $\left[\!\left[1\right]\!\right]^{\text{B}}$，否则输出 $\left[\!\left[0\right]\!\right]^{\text{B}}$。由于该过程输出为布尔秘密分享，需要调用 $\prod^{\text{B2A}}$ 将其转为算术秘密分享。安全除法协议具体实现如算法3所示，其中，$\boldsymbol{T}_{\exp 2}[i] = \left\lfloor 2^{-i} \cdot 2^f \right\rfloor, i \in [1, n-1]$，$f$ 为定点数表示下的精度位数。

算法 3 安全除法协议 $\prod^{\text{DIV}}$

输入 P_i 持有 $\left[\!\left[x\right]\!\right]_i$，$\left[\!\left[y\right]\!\right]_i$，盲化因子 $\left[\!\left[r\right]\!\right]_i$，查找表 $\boldsymbol{T}_{\exp 2}$

输出 $\left[\!\left[z\right]\!\right]_i = \left[\!\left[\frac{x}{y}\right]\!\right]_i$

1)P_i 计算 $\hat{y} = \text{restore}\left(\left[\!\left[y\right]\!\right]_i + \left[\!\left[r\right]\!\right]_i\right)$

2)for $j = 1$ to $n-1$：并行计算 $\left[\!\left[d_j\right]\!\right]_i^{\text{B}} = \text{DReLU}\left(\hat{y} - 2^j\right)$

3)for $j = 1$ to $n-1$：并行计算 $\left[\!\left[d_j\right]\!\right]_i = \prod^{\text{B2A}}\left(\left[\!\left[d_j\right]\!\right]_i^{\text{B}}\right)$

4)$\left[\!\left[k\right]\!\right]_i = \sum_1^{n-1}\left[\!\left[d_j\right]\!\right]_i$

5)$\left[\!\left[t\right]\!\right]_i = \prod^{\text{LUT}}\left(\left[\!\left[k\right]\!\right]_i + 1, \boldsymbol{T}_{\exp 2}\right)$

6)$\left[\!\left[a\right]\!\right]_i = \left[\!\left[x\right]\!\right]_i \cdot \left[\!\left[t\right]\!\right]_i$，$\left[\!\left[b\right]\!\right]_i = \left[\!\left[y\right]\!\right]_i \cdot \left[\!\left[t\right]\!\right]_i$

7)$\left[\!\left[w\right]\!\right]_i = 2.914\,2 - 2\left[\!\left[b\right]\!\right]_i$

8)$\left[\!\left[\varepsilon_0\right]\!\right]_i = 1 - \left[\!\left[b\right]\!\right]_i \cdot \left[\!\left[w\right]\!\right]_i$，$\left[\!\left[\varepsilon_1\right]\!\right]_i = \left[\!\left[\varepsilon_0\right]\!\right]_i \cdot \left[\!\left[\varepsilon_0\right]\!\right]_i$

9)$\left[\!\left[z\right]\!\right]_i = \left[\!\left[a\right]\!\right]_i \cdot \left[\!\left[w\right]\!\right]_i \cdot \left(1 + \left[\!\left[\varepsilon_0\right]\!\right]_i\right) \cdot \left(1 + \left[\!\left[\varepsilon_1\right]\!\right]_i\right)$

3.2 安全非线性层计算协议

3.2.1 安全 GELU 函数

令 $\delta(x) = \text{ReLU}(x) - \text{GELU}(x)$，$\delta(x)$ 在 [−4, 4]区间外取值为 0，因此，通过以下近似公式求解 GELU 函数

$$\text{GELU}(x) = \begin{cases} \text{ReLU}(x), x \notin [-4, 4] \\ \text{ReLU}(x) - \delta(x), x \in [-4, 4] \end{cases} \tag{7}$$

由于 $\delta(x)$ 是偶函数，可通过求 x 的绝对值将查找表范围进一步缩减一半，从而提高计算效率。此外，由于数据表示为定点数，为限制查找表范围，表中元素使用 6 位表示小数部分即可保证查表数据的精度，从而可将查找表范围限制在[0, 256)范围内，可构建查找表 $\boldsymbol{T}_{\text{GELU}} = \left\lfloor \delta\left(\frac{i}{2}^6\right) \cdot 2^f \right\rfloor, i \in \mathbb{Z}_{256}$。基于上述过程的明文 GELU($x$)如算法 4 所示。

算法 4 明文 GELU(x)

输入 数据 x

输出 $z = \text{GELU}(x)$

1)$p = \mathrm{ReLU}(x)$

2)$c = \mathrm{Clip}_{A,B}(x)$

3)$a = \mathrm{Abs}(x)$

4)$t = \dfrac{a}{2^{f-6}}$

5)$z = \mathrm{LUT}\left(t, \boldsymbol{T}_{\mathrm{GELU}}\right)$

Clip函数用于限制x的范围，定义为

$$\mathrm{Clip}_{A,B}(x) = \begin{cases} A, x < A \\ x, x \in [A,B] \\ B, x > B \end{cases} \tag{8}$$

其中，$A = -2^{f+2} + 1$，$B = 2^{f+2} - 1$，f表示输入数据在定点数表示下的小数位数。算法3步骤4)的除法实质是将数据的低$f-6$位舍弃，而舍弃低位并不影响步骤1)~步骤3)中ReLU激活函数计算、绝对值求解Abs和Clip操作。因此，可将步骤4)提前以缩短数据位长、减少比较大小操作中密钥大小，从而降低计算开销。计算ReLU(x)需要先计算$d = 1\{x \geqslant 0\}$，再结合输入x通过选择协议Select求解，当$d = 1$时，Select输出〚x〛，否则输出〚0〛，该协议实现参考文献[29]。额外计算一个区间校验位$\mathrm{cw} = 1\{-255 \leqslant x \leqslant 255\}$，即$\mathrm{cw} = \mathrm{DReLU}(x-256) - \mathrm{DReLU}(x+255)$，从而将绝对值Abs和Clip操作结合起来，该过程中所有比较大小操作都是针对x的偏移量执行，因此可使用相同的密钥进行计算。对于给定的cw和d，Abs(Clip(x))可表示为

$$\mathrm{Abs}(\mathrm{Clip}(x)) = \begin{cases} 255, & \mathrm{cw} = 0 \text{且} d = 0 \\ 255, & \mathrm{cw} = 0 \text{且} d = 1 \\ -y, & \mathrm{cw} = 1 \text{且} d = 0 \\ y, & \mathrm{cw} = 1 \text{且} d = 1 \end{cases} \tag{9}$$

文献[30]提供了一个密钥大小为$8n$的非交互式算法$\mathrm{SelectLin}_{\gamma}$可实现$\mathrm{Abs}\left(\mathrm{Clip}(x)\right)$，对于四元向量$\boldsymbol{\gamma} = \left\{(\alpha_0,\beta_0),(\alpha_1,\beta_1),(\alpha_2,\beta_2),(\alpha_3,\beta_3)\right\}$，其中$\alpha_i,\beta_i \in \mathbb{Z}_N$对$\forall i \in \mathbb{Z}_4$成立，输入2个选择位$s_0$，$s_1$和数据$x$，$\mathrm{SelectLin}_{\gamma}$输出为$\alpha_{2s_0+s_1}x + \beta_{2s_0+s_1}$，安全GELU函数中$\boldsymbol{\gamma} = \left\{(0,255),(0,255),(-1,0),(-1,0)\right\}$。基于上述过程，安全GELU函数如算法5所示，先盲化恢复$\hat{x}$，再本地进行除法计算得到$\hat{y}$，可减少密文截断的计算开销，同时盲化后的$\hat{x}$可作为Select协议的输入，减少一轮通信开销。输入数据x范围受限，在$\mathbb{Z}_{2^{n-1}}$环上生成盲化因子r，从而避免产生$x + r$超环导致$\hat{y}$计算出错的问题，保证协议正确性，盲化因子r仍在$\mathbb{Z}_{2^n}$环上分享，并不影响安全性。

算法5　安全GELU函数$\prod^{\mathrm{GELU}}$

输入　P_i持有〚x〛$_i$，盲化因子〚r〛$_i$，查找表$\boldsymbol{T}_{\mathrm{GELU}}$

输出　〚z〛$_i$ = 〚GELU(x)〛$_i$

1)P_i计算$\hat{x} = \mathrm{restore}\left(〚x〛_i + 〚r〛_i\right)$

2)P_i计算$\hat{y} = \dfrac{\hat{x}}{2^{f-6}}$

3)P_i计算〚x〛$_i^{\mathrm{B}} = \mathrm{DReLU}\left(\hat{y}\right)$

4)〚cw〛$_i^{\mathrm{B}} = \mathrm{DReLU}\left(\hat{y} - 255\right) \oplus \mathrm{DReLU}\left(\hat{y} + 256\right)$

5)〚d〛$_i$‖〚cw〛$_i = \prod^{\mathrm{B2A}}\left(〚d〛_i^{\mathrm{B}} \| 〚\mathrm{cw}〛_i^{\mathrm{B}}\right)$

6)恢复盲化后的$\hat{d}$，$\widehat{\mathrm{cw}}$

7)P_i计算〚p〛$_i = \mathrm{Select}\left(\hat{d},\hat{x}\right)$

8)P_i计算〚c〛$_i = \mathrm{SelectLin}_{\gamma}\left(\widehat{\mathrm{cw}},\hat{d},\hat{y}\right)$

9)〚z〛$_i$ = 〚p〛$_i - \prod^{\mathrm{LUT}}\left(〚c〛_i, \boldsymbol{T}_{\mathrm{GELU}}\right)$

3.2.2　安全Softmax函数

对于输入向量$\boldsymbol{x} \in \mathbb{R}^k$，令$x_{\max} = \max\left(x_0,x_1,\cdots,x_{k-1}\right)$，则Softmax函数的输出向量$\boldsymbol{y} \in \mathbb{R}^k$为

$$y_i = \frac{\mathrm{e}^{x_i - x_{\max}}}{\sum_{j=0}^{k-1} \mathrm{e}^{x_j - x_{\max}}} \tag{10}$$

安全Softmax函数的输入为向量$\boldsymbol{x}$的分享值〚$\boldsymbol{x}$〛，需要执行最大值求解（max）、指数计算（exp）和除法。安全最大值求解可通过二分法调用$\lceil \mathrm{lb}k \rceil$次DReLU函数并选出每次比较的最大值。安全指数计算exp参考CrypTen[4]实现。基于max函数、exp和$\prod^{\mathrm{Div}}$即可实现安全Softmax函数。

3.2.3　安全LayerNorm函数

Transformer模型使用LayerNorm函数对数据标准化处理，对于输入向量$\boldsymbol{x} \in \mathbb{R}^k$，令$m = \dfrac{\sum x_i}{k}$和$v = \dfrac{\sum \left(x_i - m\right)^2}{k}$表示均值和方差。令$z_i = x_i - m$，则层归一化的输出向量$\boldsymbol{y} \in \mathbb{R}^k$为

$$y_i=\gamma\cdot\frac{x_i-m}{\sqrt{v}}+\beta=\gamma\cdot\frac{z_i}{\sqrt{\frac{\sum {z_i}^2}{k}}}+\beta \tag{11}$$

其中，γ和β是2个可学习的参数，表示权重和偏移量。安全LayerNorm函数的实现主要包含2个耗时操作：秘密分享数据上的除法Div和倒数平方根运算$\prod^{\text{Rsqrt}}$，具体功能及实现如下。

Div：对于输入$[\![x]\!]$和公开的除数k，输出$\left[\!\left[\frac{x}{k}\right]\!\right]$，参考CrypTen[4]的截断算法Truncation实现。

$\prod^{\text{Rsqrt}}$：对于倒数平方根$\frac{1}{\sqrt{x}}$，当$x>2^{33}$时，$\frac{1}{\sqrt{x}}<10^{-5}$，直接使用查找表算法进行求解则需要大小为$2^{33}$的查找表，内存占用和计算开销较大；采用近似拟合需要大量乘法，所需的通信开销较大，且存在一定的精度损失，因此本文将环$\mathbb{Z}_{2^n}$上定点数转为浮点数表示，以较小的位长表示较大的数据范围。选取6位指数，7位尾数，设尾数为$m\in\mathbb{Z}_{128}$，指数为$e\in\mathbb{Z}_{64}$，则有$x=2^e\cdot\left(1+\frac{m}{128}\right)$，$m=\frac{(x\cdot 128)}{2^e}-128$，指数$e$和$\frac{x}{2^e}$的求解与除法协议中标准化除数的过程相同。根据尾数$m$，指数$e$可构建13位查找表，对于$\forall p\in\mathbb{Z}_{2^{13}}$，$p=m\|e=m\cdot 2^6+e$，查找表$\boldsymbol{T}_{\text{Rsqrt}}[p]=\left\lfloor\sqrt{\frac{2^f}{q}}\cdot 2^f\right\rfloor$，其中$q=2^e\cdot\left(1+\frac{m}{128}\right)$，$f$为定点数表示下的精度位数。安全倒数平方根协议如算法6所示。

算法6　安全倒数平方根$\prod^{\text{Rsqrt}}$

输入　P_i持有$[\![x]\!]_i$，盲化因子$[\![r]\!]_i$，查找表$\boldsymbol{T}_{\exp 2}$，$\boldsymbol{T}_{\text{Rsqrt}}$

输出　$[\![z]\!]_i=\left[\!\left[\frac{1}{\sqrt{x}}\right]\!\right]_i$

1)P_i计算$\hat{x}=\text{restore}\left([\![x]\!]_i+[\![r]\!]_i\right)$

2) for $j=1$ to $n-1$：并行计算$\left[\!\left[d_j\right]\!\right]_i^{\text{B}}=\text{DReLU}\left(\hat{x}-2^j\right)$

3) for $j=1$ to $n-1$：并行计算$\left[\!\left[d_j\right]\!\right]_i=\prod^{\text{B2A}}\left(\left[\!\left[d_j\right]\!\right]_i^{\text{B}}\right)$

4)$[\![e]\!]_i=\sum_1^{n-1}[\![d_j]\!]_i$

5)$[\![t]\!]_i=\prod^{\text{LUT}}\left([\![e]\!]_i,\boldsymbol{T}_{\exp 2}\right)$

6)$[\![m]\!]_i=[\![x]\!]_i\cdot[\![t]\!]_i\cdot 128-128$

7)$[\![p]\!]_i=[\![m]\!]_i\cdot 2^6+[\![e]\!]_i$

8)$[\![z]\!]_i=\prod^{\text{LUT}}\left([\![p]\!]_i,\boldsymbol{T}_{\text{Rsqrt}}\right)$

基于安全倒数平方根协议，安全LayerNorm协议如算法7所示。

算法7　安全LayerNorm函数$\prod^{\text{LayerNorm}}$

输入　P_i持有$[\![\boldsymbol{x}]\!]_i$，参数$[\![\gamma]\!]_i$，$[\![\beta]\!]_i$

输出　$[\![\boldsymbol{y}]\!]_i=[\![\text{LayerNorm}(\boldsymbol{x})]\!]_i$

1)各参与方共同计算$[\![m]\!]_i=\text{Div}\left(\sum_{j=0}^{k}[\![x_j]\!]_i,k\right)$

2) for $j=1$ to k：计算$[\![z_j]\!]_i=[\![x_j]\!]_i-[\![m]\!]_i$

3)各参与方共同计算$[\![v]\!]_i=\text{Div}\left(\sum_{j=0}^{k}[\![z_j]\!]_i[\![z_j]\!]_i,k\right)$

4)$[\![t]\!]_i=\prod^{\text{Rsqrt}}\left([\![v]\!]_i\right)$

5)$[\![\boldsymbol{y}]\!]_i=[\![\gamma]\!]_i\cdot[\![z]\!]_i\cdot[\![t]\!]_i+[\![\beta]\!]_i$

3.3　理论分析

1)复杂度分析

本节对上述安全计算协议进行复杂度分析。

$\prod^{\text{LUT}}$所需密钥大小为$(M+1)\lceil\text{lb}M\rceil$ bit，在线阶段需要1次通信，通信量为n bit。

$\prod^{\text{B2A}}$所需密钥大小为n bit，在线阶段需要1次通信，通信量为1 bit。

$\prod^{\text{DIV}}$中比较大小操作可在本地并行执行，需要调用1次DReLU，该过程密钥大小为KeySize(DPF)+1；B2A协议可并行执行，但需要$n-1$个密钥，故调用1次$\prod^{\text{B2A}}$，密钥大小为$(n-1)n$ bit，通信开销为$n-1$ bit；调用1次$\prod^{\text{LUT}}$，由于$k\in[1,n-1]$，故查找表密钥大小为$n\lceil\text{lb}(n-1)\rceil$；在线阶段执行7次安全乘法，需要7个Beaver三元组，在线阶段需要14轮通信，通信开销21n bit。安全除法协议共需要17轮通信，通信开销为$(n-1)+23n$ bit。

$\prod^{\text{GELU}}$调用3次DReLU，需要1个DPF密钥，调用1次$\prod^{\text{B2A}}$、1次$\prod^{\text{LUT}}$、1次Select和1次Select-

Lin，在线阶段需要4轮通信，通信开销为$4n+2$ bit。

$\prod^{\text{Rsqrt}}$ 中比较大小操作可在本地并行执行，需要调用1次DReLU，该过程密钥大小为KeySize(DPF)+1；B2A协议可并行执行，但需要$n-1$个密钥，故调用1次$\prod^{\text{B2A}}$，密钥大小为$(n-1)n$ bit，通信开销为$n-1$ bit；调用2次$\prod^{\text{LUT}}$，所需密钥大小分别为$n\lceil \text{lb}(n-1)\rceil$和$\left(2^{13}+1\right)\cdot 13$ bit；调用1次安全乘法，需要2轮通信，通信开销$3n$ bit；在线阶段共需要6轮通信，通信开销为$7n-1$ bit。

$\prod^{\text{LayerNorm}}$ 需要调用2次Div、3次乘法和1次$\prod^{\text{Rsqrt}}$，在线阶段需要14轮通信，通信开销为$9n-1+9nk$ bit。

2)安全性证明

在半诚实敌手模型中，攻击者被假设为会严格按照协议规定的步骤执行操作，但在执行过程中会记录所接收到的全部中间数据和最终输出，并试图通过这些信息推测其他参与方的私有输入。该模型下的攻击者是被动的而非主动的，不能篡改消息、偏离协议、提交恶意输入或中断通信，也无法制造拒绝服务攻击。尽管攻击者的能力受到限制，但该模型依然能够揭示协议在正常执行过程中的隐私泄露风险，适用于计算环境中各参与方彼此信任度较高但仍需保护输入数据不被泄露的场景。

本节给出半诚实模型下上述协议的安全性定义及证明，所涉及的算术秘密分享及其基础计算协议在半诚实模型下是安全的，其安全性证明参考文献[19]。上述协议使用的DReLU协议、Select协议、SelectLin协议和Div协议在半诚实模型下是安全的，安全性证明分别参考文献[27]、文献[29]、文献[30]和文献[4]。本文计算所需的辅助参数均由可信第三方生成，其所产生的辅助参数是可靠的。接下来，采用标准的模拟范式[31]对所有安全基础计算协议和安全非线性层协议进行安全性证明，两方安全计算协议安全性定义如下。

定义1　两方安全计算协议。对于概率多项式时间（PPT, probabilistic polynomial time）函数$f=(f_0,f_1)$，记$\prod$为f的两方协议实现。在输入为(x,y)和安全参数为n的情况下，第i方执行协议$\prod$的视图记为$V_i^{\Pi}(x,y,n)$，输出记为$O_i^{\Pi}(x,y)$，其中$i\in\{0,1\}$。协议$\prod$的联合输出记为$O^{\Pi}(x,y)=(O_0^{\Pi}(x,y),O_1^{\Pi}(x,y))$。若存在概率多项式时间算法模拟器$\text{Sim}_0$和$\text{Sim}_1$满足

$$(\text{Sim}_0(x,f_0(x,y)),f(x,y))\overset{c}{\equiv}(V_0^{\Pi}(x,y),O^{\Pi}(x,y))$$

$$(\text{Sim}_1(x,f_1(x,y)),f(x,y))\overset{c}{\equiv}(V_1^{\Pi}(x,y),O^{\Pi}(x,y))$$

则称协议$\prod$在半诚实敌手模型下安全计算f，其中，$\overset{c}{\equiv}$表示计算不可区分性。

根据定义1，各安全基础计算协议和安全非线性层协议的安全性证明如下。

定理1　如果算术秘密分享上的基础运算协议在半诚实模型下是安全的，则$\prod^{\text{LUT}}$在半诚实模型下是安全的。

证明　密钥生成阶段由可信第三方执行，因此认为该阶段是安全的，接下来给出在线评估阶段的安全性证明。$\prod^{\text{LUT}}$将$[\![x]\!]$和公共查找表$\boldsymbol{T}$作为输入，输出$[\![\boldsymbol{T}[x]]\!]$。考虑$P_0$被攻击者俘获的情况，构造模拟器$\text{Sim}_0$模拟$P_0$，对于$P_0$持有的密钥$k_0$，$\text{Sim}_0$产生随机值$k_{\text{sim}}=r_{\text{sim}}\|v_{\text{sim}}$进行模拟，其中$r_{\text{sim}}\in\mathbb{Z}_{2^n}$，$v_{\text{sim}}\in\mathbb{Z}_{2^n}$，根据算术秘密分享定义，$[\![r]\!]_0$和$[\![v]\!]_0$均为$\mathbb{Z}_{2^n}$上的随机数，因此$k_{\text{sim}}$与$k_0$不可区分。对于$P_0$接收到的中间值$[\![x]\!]_1-[\![r]\!]_1$，$\text{Sim}_0$随机产生$\alpha\in\mathbb{Z}_{2^n}$进行模拟，$[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，则中间值$[\![x]\!]_1-[\![r]\!]_1$为随机数，因此$\alpha$与$[\![x]\!]_1-[\![r]\!]_1$不可区分。对于乘法计算中的中间值，因为其由算术秘密分享数据上的乘法计算得到，可通过类似的方式进行模拟，由安全乘法定义可知，计算中间值均为$\mathbb{Z}_{2^n}$上的随机数。进一步地，P_0和P_1在执行过程中是对称的，通过上述类似的情况可对P_1被攻击者俘获的情况进行模拟。因此，$\prod^{\text{LUT}}$在半诚实模型下是安全的。证毕。

定理2　如果算术秘密分享上的基础运算协议在半诚实模型下是安全的，则$\prod^{\text{B2A}}$在半诚实模型下是安全的。

证明　密钥生成阶段由可信第三方执行，因此认为该阶段是安全的，接下来给出在线评估阶段的安全性证明。$\prod^{\text{B2A}}$将$[\![x]\!]^{\text{B}}$作为输入，输出$[\![x]\!]$。考虑P_0被攻击者俘获的情况，构造模拟器Sim_0模拟P_0，对于P_0持有的密钥$[\![r]\!]_0$，Sim_0产生随机值$r_{\text{sim}}\in\mathbb{Z}_{2^n}$进行模拟，根据算术秘密分享定义，$[\![r]\!]_0$

为$\mathbb{Z}_{2^n}$上的随机数，因此r_{sim}与$[\![r]\!]_0$不可区分。对于P_0接收到的中间值$\left([\![x]\!]_1^{\text{B}}+[\![r]\!]_1\right)\bmod 2$，$\text{Sim}_0$随机产生$\alpha\in\mathbb{Z}_2$进行模拟，$[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，则中间值$\left([\![x]\!]_1^{\text{B}}+[\![r]\!]_1\right)\bmod 2$为$\mathbb{Z}_2$上的随机数，因此$\alpha$与$\left([\![x]\!]_1^{\text{B}}+[\![r]\!]_1\right)\bmod 2$不可区分。对于其余计算，$\text{Sim}_0$可通过类似的方式进行模拟，$[\![r]\!]_0$为$\mathbb{Z}_{2^n}$上的随机数，因此计算中间值为$\mathbb{Z}_{2^n}$上的随机数。进一步地，$P_0$和$P_1$在执行过程中是对称的，通过上述类似的情况可对$P_1$被攻击者俘获的情况进行模拟。因此，$\prod^{\text{B2A}}$在半诚实模型下是安全的。证毕。

定理3 如果$\prod^{\text{LUT}}$、$\prod^{\text{B2A}}$、DReLU和算术秘密分享上的基础运算协议在半诚实模型下是安全的，则$\prod^{\text{DIV}}$在半诚实模型下是安全的。

证明 $\prod^{\text{DIV}}$将$[\![x]\!]$、$[\![y]\!]$、随机数$[\![r]\!]$和查找表$\boldsymbol{T}_{\exp 2}$作为输入，输出$\left[\!\!\left[\frac{x}{y}\right]\!\!\right]$。考虑$P_0$被攻击者俘获的情况，构造模拟器$\text{Sim}_0$模拟$P_0$，对于随机数$[\![r]\!]_0$，$\text{Sim}_0$产生随机值$r_{\text{sim}}\in\mathbb{Z}_{2^n}$进行模拟，根据算术秘密分享定义，$[\![r]\!]_0$为$\mathbb{Z}_{2^n}$上的随机数，因此$r_{\text{sim}}$与$[\![r]\!]_0$不可区分。对于$P_0$接收到的中间值$[\![y]\!]_1+[\![r]\!]_1$，$\text{Sim}_0$随机产生$\alpha\in\mathbb{Z}_{2^n}$进行模拟，$[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，则中间值$[\![y]\!]_1+[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，因此$\alpha$与$[\![y]\!]_1+[\![r]\!]_1$不可区分。计算中间值$[\![d]\!]_0^{\text{B}}$可通过DReLU的模拟器模拟，$[\![d]\!]_0$通过$\prod^{\text{B2A}}$的模拟器模拟，$[\![t]\!]_0$通过$\prod^{\text{LUT}}$的模拟器模拟，其余计算中间值均可通过产生随机数的方式模拟，由算术秘密分享基础协议的定义可知，计算中间值均为$\mathbb{Z}_{2^n}$上的随机数。进一步地，P_0和P_1在执行过程中是对称的，通过上述类似的情况可对P_1被攻击者俘获的情况进行模拟。因此，$\prod^{\text{DIV}}$在半诚实模型下是安全的。证毕。

定理4 如果$\prod^{\text{LUT}}$、$\prod^{\text{B2A}}$、DReLU、Select、SelectLin和算术秘密分享基础协议在半诚实模型下是安全的，则$\prod^{\text{GELU}}$在半诚实模型下是安全的。

证明 $\prod^{\text{GELU}}$将$[\![x]\!]$、随机数$[\![r]\!]$和查找表$\boldsymbol{T}_{\text{GELU}}$作为输入，输出$[\![\text{GELU}(x)]\!]$。考虑$P_0$被攻击者俘获的情况，构造模拟器$\text{Sim}_0$模拟$P_0$，对于随机数$[\![r]\!]_0$，$\text{Sim}_0$产生随机值$r_{\text{sim}}\in\mathbb{Z}_{2^n}$进行模拟，根据算术秘密分享定义，$[\![r]\!]_0$为$\mathbb{Z}_{2^n}$上的随机数，因此$r_{\text{sim}}$与$[\![r]\!]_0$不可区分。对于$P_0$接收到的中间值$[\![x]\!]_1+[\![r]\!]_1$，$\text{Sim}_0$随机产生$\alpha\in\mathbb{Z}_{2^n}$进行模拟，$[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，则中间值$[\![x]\!]_1+[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，因此$\alpha$与$[\![x]\!]_1+[\![r]\!]_1$不可区分。计算中间值$[\![d]\!]_0^{\text{B}}$、$[\![\text{cw}]\!]_0^{\text{B}}$通过DReLU的模拟器模拟，$[\![d]\!]_0$、$[\![\text{cw}]\!]_0$通过$\prod^{\text{B2A}}$的模拟器模拟，$[\![p]\!]_0$、$[\![c]\!]_0$分别通过Select的模拟器和SelectLin的模拟器模拟，查表结果通过$\prod^{\text{LUT}}$的模拟器模拟，其余计算中间值均可通过产生随机数的方式模拟，由算术秘密分享基础协议的定义可知，计算中间值均为$\mathbb{Z}_{2^n}$上的随机数。进一步地，P_0和P_1在执行过程中是对称的，通过上述类似的情况可对P_1被攻击者俘获的情况进行模拟。因此，$\prod^{\text{GELU}}$在半诚实模型下是安全的。证毕。

定理5 如果$\prod^{\text{LUT}}$、$\prod^{\text{B2A}}$、DReLU和算术秘密分享上的算术运算协议在半诚实模型下是安全的，则$\prod^{\text{Rsqrt}}$在半诚实模型下是安全的。

证明 $\prod^{\text{Rsqrt}}$将$[\![x]\!]$、随机数$[\![r]\!]$和查找表$\boldsymbol{T}_{\exp 2}$、$\boldsymbol{T}_{\text{Rsqrt}}$作为输入，输出$\left[\!\!\left[\frac{1}{\sqrt{x}}\right]\!\!\right]$。考虑$P_0$被攻击者俘获的情况，构造模拟器$\text{Sim}_0$模拟$P_0$，对于随机数$[\![r]\!]_0$，$\text{Sim}_0$产生随机值$r_{\text{sim}}\in\mathbb{Z}_{2^n}$进行模拟，根据算术秘密分享定义，$[\![r]\!]_0$为$\mathbb{Z}_{2^n}$上的随机数，因此$r_{\text{sim}}$与$[\![r]\!]_0$不可区分。对于$P_0$接收到的中间值$[\![x]\!]_1+[\![r]\!]_1$，$\text{Sim}_0$随机产生$\alpha\in\mathbb{Z}_{2^n}$进行模拟，$[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，则中间值$[\![x]\!]_1+[\![r]\!]_1$为$\mathbb{Z}_{2^n}$上的随机数，因此$\alpha$与$[\![x]\!]_1+[\![r]\!]_1$不可区分。计算中间值$[\![d]\!]_0^{\text{B}}$通过DReLU的模拟器模拟，$[\![d]\!]_0$通过$\prod^{\text{B2A}}$的模拟器模拟，$[\![t]\!]_0$、$[\![z]\!]_0$通过$\prod^{\text{LUT}}$的模拟器模拟，其余计算中间值均可通过产生随机数的方式模拟，由算术秘密分享基础协议的定义可知，计算中间值均为$\mathbb{Z}_{2^n}$上的随机数。进一步地，P_0和

P_1在执行过程中是对称的，通过上述类似的情况可对P_1被攻击者俘获的情况进行模拟。因此，$\prod^{\text{Rsqrt}}$在半诚实模型下是安全的。证毕。

定理 6　如果$\prod^{\text{Rsqrt}}$、Div和算术秘密分享上的算术运算协议在半诚实模型下是安全的，则$\Pi^{\text{LayerNorm}}$在半诚实模型下是安全的。

证明　$\Pi^{\text{LayerNorm}}$将$[\![x]\!]$、$[\![\gamma]\!]$、$[\![\beta]\!]$作为输入，输出$[\![\text{LayerNorm}(x)]\!]$。考虑$P_0$被攻击者俘获的情况，构造模拟器$\text{Sim}_0$模拟$P_0$。计算中间值$[\![m]\!]_0$、$[\![v]\!]_0$可通过Div的模拟器模拟，$[\![t]\!]_0$通过$\prod^{\text{Rsqrt}}$的模拟器模拟，其余计算中间值均可通过产生随机数的方式模拟，由算术秘密分享基础协议的定义可知，计算中间值均为$\mathbb{Z}_{2^n}$上的随机数。进一步地，P_0和P_1在执行过程中是对称的，通过上述类似的情况可对P_1被攻击者俘获的情况进行模拟。因此，$\Pi^{\text{LayerNorm}}$在半诚实模型下是安全的。证毕。

综上所述，本文提出的协议在半诚实敌手模型下是安全的。另一种安全性更高的模型是恶意敌手模型，攻击者可以偏离协议执行，主动篡改消息或提交不一致的输入等。为了在此模型下保证协议的正确性与隐私性，通常需要引入额外的机制，例如零知识证明以证明计算正确性，一致性检测机制以确保各方输入与行为的一致性。这些机制虽会带来额外的计算与通信开销，但可将协议扩展至更强的安全设定下运行[32]。本文将面向恶意敌手模型的协议拓展作为未来工作的重要方向。

4　实验分析

本节对安全基础计算协议和大语言模型密态推理方案进行了计算和通信开销的评估，并基于GLUE基准数据集测试了密态推理的准确率。

4.1　实验设置

安全计算协议基于Python实现，在一台服务器上进行评估，该服务器使用NVIDIA GeForce RTX 4090用于GPU计算，具有24 GB显存，CPU为Inter® Core™ i9-14900K CPU @ 3.20 GHz，操作系统为Ubuntu 20.04。所有计算均在环上进行，环的大小设置为$\mathbb{Z}_{2^{64}}$，函数秘密分享的安全参数$\lambda = 128$。选取Bert-Base、Bert-Large和GPT-2模型，模型参数如表3所示，其中，B表示Transformer模块的数量，D表示模型维度，T表示输入记号的数量，模型中注意力头的数量为$\frac{D}{64}$，前馈特征大小为$4D$。

表3　大语言模型及其参数

模型	参数量	超参数		
		B	D	T
Bert-Base	110×10^6	12	768	128
Bert-Large	340×10^6	24	1 024	128
GPT-2	124×10^6	12	768	128

对于Bert-Base模型，用于训练和推理的GLUE基准数据集为语言可接受性语料库（CoLA, corpus of linguistic acceptability）、问答自然语言推断数据集（QNLI, question-answering nature language inference）和语义文本相似性基准测试数据集（STS-B, semantic textual similarity benchmark）[33]，评价指标分别为马修斯相关系数（MCC, Matthews correlation coefficient）、准确率和皮尔森–斯皮尔曼相关系数（PSC, Person-Spearman correlation）。Bert-Large模型使用的数据集为QNLI。对于GPT-2模型，用于训练和推理的数据集为WikiText103[34]，评价指标为困惑度。

4.2　安全基础计算协议实验评估

为评估本文提出的安全基础计算协议的优越性，本节在相同环境下复现了现有最先进的各项具体功能方案，包括Pika[18]的查找表协议、Liu等[2]的安全分享模式转换协议以及CrypTen[4]的除法协议。其中，Pika的查找表协议基于分布式点函数实现，安全分享模式转换协议基于算术秘密分享的乘法实现，CrypTen的除法协议基于Newton-Raphson迭代思想实现。表4列出了不同数据规模下在线阶段各安全基础计算协议的执行时间和通信开销，其中，安全查找表算法中，查找表的大小为2^{12}。

Pika的查找表协议在线执行阶段需要执行一次全域的DPF评估，以求解离线密钥生成阶段产生的随机数的one-hot向量；本文方案将该one-hot向量作为密钥分发给各个计算参与方，因此在线阶段执行时间相比Pika的查找表协议缩短了90%~95%，在线阶段仅在恢复盲化的输入数据时需要通信，故本文与Pika的通信开销相同。对于数量级，DPF评估产生的计算中间值大小超过显存限制，无法评估Pika的执行时间。

表4 不同数据规模下在线阶段各安全基础计算协议的执行时间和通信开销

协议	方案	数据规模为10^3		数据规模为10^4		数据规模为10^5		通信轮次
		执行时间/s	通信开销/KB	执行时间/s	通信开销/KB	执行时间/s	通信开销/KB	
查找表协议	Pika[18]	0.172 1	7.81	1.778 2	78.13	—	—	1
	本文方案	0.015 1	7.81	0.077 7	78.13	0.789 5	781.25	1
分享模式转换协议	Liu等[2]	0.000 9	1.95	0.001 0	19.53	0.001 8	195.31	1
	本文方案	0.000 6	0.98	0.000 6	9.77	0.001 2	97.66	1
安全除法协议	CrypTen[4]	0.043 8	687.50	0.043 8	6 875.00	0.066 2	68 750.00	59
	本文方案	0.045 6	210.94	0.126 9	2 109.38	0.779 4	21 093.75	17

安全分享模式转换协议由本地基础算术运算构成，协议性能瓶颈为计算参与方的通信，相比Liu等[2]的工作，本文方案将通信开销减少了50%。

针对安全除法协议，CrypTen[4]首先利用Newton-Raphson迭代法求解除数的倒数，再结合乘法求解除法结果，将非线性计算转换为多次乘法运算，迭代初始值为$3e^{1-2x}+0.003$，迭代次数为10，由于指数函数需要利用大量乘法迭代求解，基于Beaver三元组的安全乘法需要大量通信开销[24]，因此，CrypTen的安全除法协议在线阶段需要59轮通信。本文方案结合比较大小协议和查找表算法限制除数范围，其余计算仅需要7次乘法，由于比较大小协议无须通信，查找表算法仅需1轮通信，共需17轮通信。在大语言模型推理任务中，除法所需计算量仅在10^2~10^3数量级，因此，相比CrypTen，本文方案的安全除法协议在相似计算开销的同时，通信轮次和通信开销减少了70%。

密文计算均在环上进行，由于环结构不支持浮点数表示，需将浮点数映射为环上的定点数表示，映射过程为$x=\left\lfloor \frac{x_{\text{real}}}{2^f} \right\rfloor$，其中，$x_{\text{real}}$为实数域上的浮点数，$f$为定点数小数位数，通常取值为16。该转换引入的最大量化误差为$\frac{1}{2^f}$，在f=16的情况下，误差上界为1.5×10^{-5}，属于可接受范围。在此基础上，结合安全查找表协议、安全分享模式转换协议等安全基础协议可实现安全除法、安全GELU、安全倒数平方根等安全非线性函数计算协议，各非线性函数计算协议的计算误差主要受定点表示的浮点位数限制。其中，安全除法协议和安全倒数平方根协议使用的浮点数位数为16，对应的计算误差不超过$1\times10^{-4}\left(\text{即}\frac{1}{65\,536}\right)$；安全GELU协议基于查找表实现，为控制查表空间开销，选取的定点数小数位数为6，其计算误差不超过$1\times10^{-2}\left(\text{即}\frac{1}{64}\right)$。在大语言模型推理等机器学习任务中，通常保留2位小数，即计算精度约为1×10^{-2}。因此，本文各类安全非线性函数计算协议在满足隐私保护的同时，能够保证符合机器学习任务中对计算精度的要求。

4.3 大语言模型密态推理实验评估

本节基于本文安全基础计算协议和安全非线性计算协议构建大语言模型密态推理系统，并将其与现有先进工作CrypTen[4]、SecFormer[17]、BOLT[16]和BumbleBee[21]进行对比，各非线性层及总体密态推理时间和通信开销如表5所示。其中，CrypTen[4]是现有较为完善的隐私保护机器学习框架，可支持大语言模型密态推理，SecFormer[17]是最新的基于秘密分享的隐私保护大语言模型推理工作，BOLT[16]和BumbleBee[21]则是最新基于同态加密与秘密分享相结合的大模型密态推理方案，SecFormer和BOLT只支持Bert模型的密态推理，数据均来自论文，其中BOLT只给出了Bert-base模型的推理时间和通信开销，BumbleBee未测评Bert模型中各非线性函数的密态计算开销。

表5中数据显示，相比现有方案，本文方案在通信开销优化显著。本文方案基于安全查找表协议求解GELU激活函数，在线阶段仅需4轮通信，与现有基于分段函数和高次多项式拟合的求解方法相比，通信开销减少了85%~95%。对于Softmax函数，SecFormer通过二次逼近求解指数函数，只需

表5　各非线性层及总体密态推理时间和通信开销

模型	方案	GELU		Softmax		LayerNorm		密态推理总体	
		推理时间/s	通信开销/KB	推理时间/s	通信开销/KB	推理时间/s	通信开销/KB	推理时间/s	通信开销/KB
Bert-Base	CrypTen[4]	4.59	3.45	7.53	3.27	4.31	0.11	21.55	8.34
	SecFormer[15]	8.13	17.82	1.36	1.84	2.52	0.47	19.51	23.59
	BOLT[14]	14.4	1.43	16.4	1.41	14.3	0.58	185.00	59.61
	BumbleBee[19]	—	—	—	—	—	—	153.00	6.40
	本文方案	5.01	0.15	5.01	0.56	1.55	0.17	15.71	1.97
Bert-Large	CrypTen[4]	11.50	9.19	17.35	8.72	8.75	0.29	54.53	23.36
	SecFormer[15]	14.53	35.63	3.12	4.92	3.12	1.25	39.09	50.36
	本文方案	13.33	0.40	10.73	1.51	3.40	0.42	41.63	5.51
GPT-2	CrypTen[4]	4.47	3.45	6.89	3.27	3.94	0.11	20.45	8.34
	BumbleBee[19]	20.55	1.06	19.48	0.78	4.93	0.42	183.60	3.31
	本文方案	6.76	0.15	5.98	0.56	1.90	0.17	19.91	2.43

一次乘法，但其安全除法协议需要多轮迭代近似求解，因此通信开销较大，本文方案所需计算时间是 SecFormer 的 3.4 倍，但将通信开销减少了 70%。本文方案在 LayerNorm 函数上的通信开销略高于 CrypTen，但将计算效率提升了 2.5 倍，相比 SecFormer，通信开销减少了 63%~66%。相比 BOLT 和 BumbleBee 的 Softmax 和 LayerNorm 方案，本文方案计算效率提高了 3~10 倍。基于 Bert 和 GPT 模型进行单样本密态推理，与 CrypTen 相比，将计算效率提升了 1.3 倍，通信开销仅为该方案的 25%。由于 Softmax 所需时间多于 SecFormer，Bert-Large 的推理时间略长，但将通信开销减少了 90%，而 Bert-Base 的推理效率相比 SecFormer 有 20% 的提升。BOLT 和 BumbleBee 结合了同态加密与秘密分享设计了混合式协议，以提升隐私保护大模型推理性能。然而，受限于同态加密的高计算开销，这些方案在性能上仍存在瓶颈；相较之下，本文方案在保证相同性能精度的前提下，推理效率提升约 10 倍，通信开销减少了 20%~90%，进一步验证了所提方法在大模型推理场景下的计算效率和通信成本优势。

选取 CoLA、QNLI、STS-B 数据集对 Bert-Base 模型进行评估，选取 QNLI 数据集对 Bert-Large 模型进行评估，选取 WikiText103 数据集对 GPT-2 模型进行评估，准确率结果如表 6 所示。

表6　大语言模型密态推理准确率与明文准确率对比

模型	数据集	准确率	
		明文	本文
Bert-Base	CoLA	53.98%	53.91%
	QNLI	90.10%	89.11%
	STS-B	93.89%	93.94%
Bert-Large	QNLI	93.56%	93.07%
GPT-2	WikiText103(↓)	37.5%	37.6%

由于密文计算均以定点数形式执行，在浮点数与定点数的转换过程中不可避免地会产生一定的精度损失。此外，在使用近似公式计算指数等非线性函数时，也会引入计算误差。因此，密态推理评估结果与明文模型评估有一定差距，但最大精度损失小于 0.1%。综上，本文方案在实现有效数据隐私保护的同时，能够有效地保证推理精度。

此外，为证明本文方案在不同规模的大语言模型和数据量上的可扩展性，选取 Bert-tiny 和 Qwen2[35]模型进行测试，其中 Bert-Tiny 模型参数量为 4.4×10^6，Qwen2 模型参数量为 1.5×10^9。

表 7 列出了不同规模 Bert 模型的密态推理时间及通信开销，并与现有先进方案 CrypTen 进行对比。与 CrypTen 相比，本文方案的推理效率提升了 18%~27%，通信开销减少了 55%~76%。

表7 不同规模Bert模型的密态推理时间及通信开销

Bert模型及参数量	方案	时间/s	通信量/GB
Bert-Tiny (4.4×10^6)	CrypTen[4]	1.71	0.20
	本文方案	1.41	0.09
Bert-Base (110×10^6)	CrypTen[4]	21.55	8.34
	本文方案	15.71	1.97
Bert-Large (340×10^6)	CrypTen[4]	54.53	23.36
	本文方案	43.63	5.51

本文从相同输入记号数、不同batch大小和相同batch大小、不同记号数两方面对参数量更大的Qwen2模型[35]进行评估，受GPU显存大小限制，输入记号总数不超过32个，结果如表8所示，其中，T表示输入记号数。在固定输入记号数为4的情况下，随着batch大小增加，GPU加速效果显著提升。在batch大小为8时，均摊时间仅需12.95 s $\left(即\frac{103.56}{8}\right)$，相比batch大小为1时，计算效率提高了5.9倍。在固定batch大小的情况下，输入记号数的增加仅带来了16.1%的额外计算开销和9.3%的额外通信开销。因此本文方案在不同规模的模型和输入上均展现了较好的性能，可扩展性较强。

表8 不同输入规模下Qwen2模型密态推理评估

输入规模	Batch大小/输入记号数	时间/s	通信量/GB
不同batch大小（T=4）	1	90.08	9.88
	2	88.75	10.00
	4	94.63	10.25
	8	103.56	10.74
不同记号数（batch=1）	4	90.08	9.88
	8	93.40	10.01
	32	104.6	10.80

5 结束语

针对大语言模型密态推理方案计算效率低、通信开销大的问题，本文设计了高效精准的大语言模型密态推理方案。基于秘密分享技术提出了安全查找表算法、安全分享模式转换协议和安全除法协议。将复杂非线性函数的安全计算问题转换为查表问题，设计了GELU和LayerNorm的安全计算协议，与利用高阶多项式近似的方案相比，显著降低了复杂非线性函数安全计算的通信开销。基于上述安全计算协议，构建了大语言模型密态推理方案，并在半诚实模型下证明了方案的安全性。实验结果表明，与现有工作相比，本文方案在保证推理精度的前提下，密态推理效率得到显著提升。后续研究将进一步评估该方案在多模态任务中的适用性，扩展至图像分类、语音识别等非文本场景，并考虑将协议扩展至恶意安全模型，增强系统在现实威胁环境下的鲁棒性与实用性。

参考文献：

[1] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv Preprint, arXiv: 1706.03762, 2017.

[2] LIU X N, ZHENG Y F, YUAN X L, et al. Securely outsourcing neural network inference to the cloud with lightweight techniques[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 620-636.

[3] MOHASSEL P, RINDAL P. ABY3: a mixed protocol framework for machine learning[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2018: 35-52.

[4] KNOTT B, VENKATARAMAN S, HANNUN A, et al. CrypTen: secure multi-party computation meets machine learning[J]. arXiv Preprint, arXiv: 2109.00984, 2021.

[5] TAN S J, KNOTT B, TIAN Y, et al. CryptGPU: fast privacy-preserving machine learning on the GPU[C]//Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2021: 1021-1038.

[6] RATHEE D, RATHEE M, KUMAR N, et al. CrypTFlow2: practical 2-party secure inference[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2020: 325-342.

[7] RYFFEL T, THOLONIAT P, POINTCHEVAL D, et al. AriaNN: low-interaction privacy-preserving deep learning via function secret sharing[J]. Proceedings on Privacy Enhancing Technologies, 2022(1): 291-316.

[8] MOHASSEL P, ZHANG Y P. SecureML: a system for scalable privacy-preserving machine learning[C]//Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2017: 19-38.

[9] WAGH S, TOPLE S, BENHAMOUDA F, et al. Falcon: honest-majority maliciously secure framework for private deep learning[J]. Proceedings on Privacy Enhancing Technologies, 2021(1): 188-208.

[10] SONG A X, FU J X, MU X T, et al. L-SecNet: towards secure and lightweight deep neural network inference[J]. Journal of Networking and Network Applications, 2023, 3(4): 171-181.

[11] GUO C, CHENG K, FU J X, et al. GFS-CNN: a GPU-friendly secure computation platform for convolutional neural networks[J]. Journal of Networking and Network Applications, 2023, 3(2): 66-72.

[12] 任艳丽, 余凌赞, 何港, 等. 一种隐私保护的卷积神经网络预测方案[J]. 计算机学报, 2023, 46(8): 1606-1619.
REN Y L, YU L Z, HE G, et al. A scheme of privacy-preserving convolutional neural network prediction[J]. Chinese Journal of Computers, 2023, 46(8): 1606-1619.

[13] HAO M, LI H W, CHEN H X, et al. Iron: private inference on transformers[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: ACM Press, 2022: 15718-15731.

[14] CHEN T Y, BAO H B, HUANG S H, et al. THE-X: privacy-preserving transformer inference with homomorphic encryption[C]//Proceedings of the Findings of the Association for Computational Linguistics. Stroudsburg: ACL Press, 2022: 3510-3520.

[15] LI D, WANG H, SHAO R, et al. MPCFormer: fast, performant and private Transformer inference with MPC[C]//Proceedings of the Eleventh International Conference on Learning Representations. Piscataway: IEEE Press, 2022: 1-16.

[16] PANG Q, ZHU J H, MÖLLERING H, et al. BOLT: privacy-preserving, accurate and efficient inference for transformers[C]//Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2024: 4753-4771.

[17] LUO J L, ZHANG Y H, ZHANG Z, et al. SecFormer: fast and accurate privacy-preserving inference for transformer models via SMPC[C]//Proceedings of the Findings of the Association for Computational Linguistics. Stroudsburg: ACL Press, 2024: 13333-13348.

[18] WAGH S. Pika: secure computation using function secret sharing over rings[J]. Proceedings on Privacy Enhancing Technologies, 2022(4): 351-377.

[19] DEMMLER D, SCHNEIDER T, ZOHNER M. ABY - A framework for efficient mixed-protocol secure two-party computation[C]//Proceedings of 2015 Network and Distributed System Security Symposium. Rosten: Internet Society, 2015: 1-15.

[20] PATRA A, SCHNEIDER T, SURESH A, et al. ABY2.0: improved mixed-protocol secure two-party computation[C]//Proceedings of the 30th USENIX Security Symposium (USENIX Security 21). Berkeley: USENIX Association, 2021: 2165-2182.

[21] LU W J, HUANG Z C, GU Z, et al. BumbleBee: secure two-party inference framework for large transformers[C]//Proceedings of 2025 Network and Distributed System Security Symposium. Rosten: Internet Society, 2025: 1-18.

[22] ZHANG J, LIU J, YANG X, et al. Secure transformer inference made non-interactive[C]//Proceedings of the 39th International Conference on Neural Information Processing Systems. New York: ACM Press, 2025: 1-15.

[23] AGRAWAL N, SHAHIN SHAMSABADI A, KUSNER M J, et al. QUOTIENT: two-party secure neural network training and prediction[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2019: 1231-1247.

[24] BEAVER D. Efficient multiparty protocols using circuit randomization[C]//Advances in Cryptology — CRYPTO'91. Berlin: Springer, 2007: 420-432.

[25] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Stroudsburg: ACL Press, 2019: 4171-4186.

[26] ACHIAM J, ADLER S, AGARWAL S, et al. Gpt-4 technical report[J]. arXiv Preprint, arXiv: 2303.08774, 2023.

[27] GUPTA K, JAWALKAR N, MUKHERJEE A, et al. Sigma: secure GPT inference with function secret sharing[C]//Proceedings on Privacy Enhancing Technologies. Saarland: DBLP, 2024: 1-19.

[28] BOYLE E, GILBOA N, ISHAI Y. Function secret sharing: improvements and extensions[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2016: 1292-1303.

[29] JAWALKAR N, GUPTA K, BASU A, et al. Orca: FSS-based secure training and inference with GPUs[C]//Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2024: 597-616.

[30] DAMGÅRD I, NIELSEN J B, NIELSEN M, et al. The TinyTable protocol for 2-party secure computation, or: gate-scrambling revisited[C]//Advances in Cryptology – CRYPTO 2017. Berlin: Springer, 2017: 167-187.

[31] LINDELL Y. How to simulate it – A tutorial on the simulation proof technique[C]//Tutorials on the Foundations of Cryptography. Berlin: Springer, 2017: 277-346.

[32] LEHMKUHL R, MISHRA P, SRINIVASAN A, et al. Muse: secure inference resilient to malicious clients[C]//Proceedings of the 30th USENIX Security Symposium (USENIX Security 21). Berkeley: USENIX Association, 2021: 2201-2218.

[33] WANG A. Glue: a multi-task benchmark and analysis platform for natural language understanding[J]. arXiv Preprint, arXiv: 1804.07461, 2018.

[34] MERITY S, XIONG C, BRADBURY J, et al. Pointer sentinel mixture models[J]. arXiv Preprint, arXiv: 1609.07843, 2016.

[35] YANG A, YANG B, ZHANG B, et al. Qwen2.5 technical report[J]. arXiv Preprint, arXiv: 2412.15115, 2024.

[作者简介]

程珂（1993– ），男，安徽潜山人，博士，西安电子科技大学副教授，主要研究方向为隐私保护、机器学习、应用密码学等。

付家瑄（1997– ），男，陕西西安人，西安电子科技大学博士生，主要研究方向为物联网安全、隐私保护和机器学习。

夏昱珩（2003– ），男，江苏镇江人，西安电子科技大学硕士生，主要研究方向为隐私保护、大模型。

祝幸辉（1990– ），男，河北邢台人，博士，西安电子科技大学讲师，主要研究方向为云计算与数据安全、物联网安全等。

代川云（2003– ），男，四川达州人，西安电子科技大学硕士生，主要研究方向为隐私保护、机器学习。

沈玉龙（1978– ），男，江苏泗洪人，博士，西安电子科技大学教授，主要研究方向为云计算与数据安全、无线网络安全等。