基于秘密共享的轻量级隐私保护ViT推理框架

马敏[1,2]，付钰[1]，黄凯[3]，贾潇风[4]
（1.海军工程大学信息安全系，湖北 武汉 430033；2.湖北开放大学软件工程学院，湖北 武汉 430074；
3.国防大学联合作战学院，河北 石家庄 050084；4.浙江工商大学计算机科学与技术学院，浙江 杭州 310018）

摘　要：针对广泛应用于图像处理的ViT推理框架存在泄露用户隐私数据的风险，而已有隐私保护推理框架存在计算效率较低、在线通信量较大等问题，提出了一种高效隐私保护推理框架SViT。该框架由2个边缘服务器协作执行基于秘密共享设计的安全计算协议SSoftmax、SLayerNorm、SGeLU，在保持ViT-B/16原始框架结构的情况下，解决了隐私保护框架推理开销大的问题。理论分析与实验表明，相比CrypTen，SViT在计算效率和在线通信开销方面分别提升了2~6倍和4~14倍。

关键词：隐私保护；秘密共享；图像分类；安全计算协议
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000−436x.2024025

Light weighted privacy protection ViT inference framework based on secret sharing

MA Min[1,2], FU Yu[1], HUANG Kai[3], JIA Xiaofeng[4]
1. Department of Information Security, Naval University of Engineering, Wuhan 430033, China
2. School of Software Engineering, Hubei Open University, Wuhan 430074, China
3. College of Joint Operation, National Defense University, Shijiazhuang 050084, China
4. School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018, China

Abstract: The ViT (vision transformer) inference framework, which was widely used in image processing, was found to have a risk of leaking user privacy data. However, existing privacy protection inference frameworks had problems such as low computational efficiency and high online communication volume. To address this issue, a highly efficient privacy protection inference framework SViT was proposed. Two edge servers collaborated to execute secure computing protocols based on secret sharing design, such as SSoftmax, SLayerNorm, SGeLU, etc. While maintaining the original framework structure of ViT-B/16, the problem of large inference overhead in privacy protection framework was solved. Theoretical analysis and experiments show that compared to Crypton, SViT has improved computational efficiency by 2~6 times and online communication overhead by 4~14 times, respectively.

Keywords: privacy protection, secret sharing, image classification, secure computing protocol

0　引言

计算机视觉领域中，ViT（vision transformer）框架备受关注，它基于Transformer架构，常用于图像特征精准提取和分类[1]。ViT框架的广泛使用推动了其推理服务的发展，即服务商将训练好的ViT框架部署为在线推理服务，而用户将查询请求发送至服务器以获得推理结果[2]。然而，查询使用的图像往往包含面部特征、个人身份、位置、喜好

收稿日期：2023−11−02；**修回日期：**2023−12−14
通信作者：付钰，fuyu0219@163.com
基金项目：国家自然科学基金资助项目（No.62102422）
Foundation Item：The National Natural Science Foundation of China (No.62102422)

等敏感信息，这些信息可能被窃取和滥用，带来不良后果[3]。因此，设计隐私保护ViT推理框架成为一个亟须解决的问题。

隐私保护ViT推理框架的设计面临多方面的挑战，一方面是类似于ViT、Bert的大模型架构复杂，包含大量计算密集型函数（Softmax、GeLU、LayerNorm等），这导致计算开销很大，从而增加了推理时延；另一方面是在提供推理服务时，在线通信量也会成为限制用户查询吞吐量的因素[4]。虽然可以通过调整大模型的网络架构降低推理开销，但这往往会牺牲推理的精度[5]，因此有必要设计一种安全高效且在线通信量低的ViT推理框架，以应对这些挑战。

本文提出了一种轻量级隐私保护推理框架SViT。该框架基于秘密共享技术，实现了一系列对应于ViT模型结构的安全计算协议；使用了2个独立的边缘服务器和一个可信任的第三方服务器。离线阶段，可信任的第三方服务器负责生成随机值；在线阶段，2个边缘服务器通过一系列安全交互协议对接收到的加密图像执行分类任务。用户可以从2个边缘服务器生成的加密结果中获得输入图像的分类结果。本文的主要贡献如下。1) 基于秘密共享设计了一系列安全计算协议，包括安全指数（SExp）协议、安全除法（SDiv）协议、安全平方根（SSqrt）协议、安全方差（SVar）协议、SLayerNorm、SSoftmax、SGeLU，用于实现安全ViT框架的各层。相较于已有的相关安全协议，本文提出的协议计算复杂度更低，在线通信开销更小，同时精度更高。2) 提出了基于ViT-B/16模型结构的轻量级隐私保护推理框架SViT，2个边缘服务器交互执行相关安全计算协议，实现图像分类任务，同时避免泄露用户数据、中间数据和推理结果。3) 通过理论分析和实验，证明了本文提出的协议的安全性、高效性和正确性。在实验评估中，与现有的CrypTen协议对比，SViT在计算效率和在线通信开销方面分别提升了2~6倍和4~14倍。

1　相关工作

近年来，隐私保护的神经网络推理逐渐成为研究热点。在早期的研究中，学者主要关注卷积神经网络（CNN）的隐私保护推理，并提出一系列方案。

SecureML[6]设计了面向秘密共享的定点数算术计算，提出Sigmoid的分段计算和Softmax的近似计算方法。ABY[7]考虑到秘密共享和乱码电路在线性和非线性计算的不同优势，设计了混合协议框架以提升安全计算性能。与ABY相比，ABY2.0[8]提出新的分享语义，大大降低了安全计算的在线通信量，通过取消在线阶段的不经意传输（OT），提升了混合协议的转换效率。与前面的工作不同，FALCON[9]完全使用秘密共享以避免协议转换开销，并且降低整数环的大小以改进通信量。Delphi[10]采用神经网络搜索（NAS）技术，用二次多项式替换ReLU以降低计算开销。SIRNN[11]使用了查找表（LUT）和混合位宽的方法构造函数，大大减少了数据通信量。CrypTFlow2[12]基于OT设计新的比较协议，并将此协议应用到ReLU、Maxpool等非线性函数上，降低非线性计算的开销。

随着大模型如Bert[13]和ViT的兴起，学者也开始研究如何在这些模型中实现隐私保护推理。然而，Bert、ViT相对于CNN模型，采用的自注意力机制计算复杂度更高，模型参数量更大。因此，直接将以前的方案应用到大模型可能会带来推理速度过慢的问题。例如，构成自注意力机制的SExp、SDiv、SSqrt在MP-SPDZ[14]、CrypTen[15]中均基于迭代的方法设计，将导致较大的通信和计算开销。

为了解决这个问题，学者提出了多种方案以降低推理开销。针对Bert模型，MPCFormer[16]引入了蒸馏学习压缩推理模型，降低了模型的计算和通信负荷。THE-X[17]用ReLU激活函数替代复杂的GeLU函数，并设计了一个近似网络逼近Softmax。Wang等[18]选择Transformer变体改进多头注意力层的结构，降低了Softmax的计算复杂度，并利用tensor-train分解优化嵌入表查找。以上的工作修改了原有的网络结构，以推理精度为代价降低了推理开销。Iron[2]设计了同态加密矩阵乘法协议，并对Softmax、GeLU、LayerNorm函数进行了优化，但在计算开销方面仍有提升空间。

对于ViT模型，CrypTen使用了高度优化的PyTorch张量库实现高性能计算，具体来说，各参与方基于高效的PyTorch API实现本地计算，提高计算效率。然而CrypTen没有设计SGeLU、SLayerNorm等协议，并且CrypTen仅支持网络加密，不适用于强调推理效率的场景。本文提出的SViT框

架设计了 GeLU 等非线性函数的安全计算协议，保持了 ViT 模型的原始网络结构，在保证推理精度的基础上降低了微调成本和推理开销。

2　预备知识

2.1　ViT

ViT 是一种基于 Transformer 结构的神经网络模型，主要分为 3 个部分。

嵌入层。由于 Transformer 主要应用于 NLP 领域，而 NLP 主要处理的是序列数据，因此 ViT 需要将图像也转换成序列数据。具体做法是将输入的图像先分成多个块，展平后添加位置编码信息和类别 token，方便后续的特征提取工作。

Transformer 编码器层。该层的主要作用是提取图像的特征，在 ViT-B/16 结构中，该层由编码器模块堆叠 12 次构成。编码器模块由 LayerNorm 子层、多头注意力子层和前馈神经网络子层组成。首先，数据通过 LayerNorm 子层进行归一化。然后，经过处理的数据进入多头注意力子层，实现图像块特征的加权汇总，该层的关键是注意力函数，如式(1)所示。

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

其中，d_K为向量维数。最后，数据经过由全连接和 GeLU 构成的前馈神经网络子层进一步处理，提取出图像的更深层特征。

分类层。该层实现最终分类。首先，数据通过 LayerNorm 进行归一化。然后，通过 Linear 进行线性变换。最后，进入 Softmax 实现分类。

2.2　秘密共享

秘密共享是一种安全多方计算技术，用于在多个参与方之间进行计算，同时保护各方的数据隐私。秘密共享允许参与方共同计算结果，而不暴露他们各自的输入值。其中，加法秘密共享是一种较为常用的技术。

以两方加法秘密共享为例，秘密$x \in \mathbb{Z}_{2^\ell}$被随机分割成 2 个分量$x_0,x_1 \in \mathbb{Z}_{2^\ell}$，满足$x = x_0 + x_1$，参与方$P_0$和$P_1$分别持有分量$x_0$和$x_1$，本文用[·]表示加法秘密共享。

基于加法秘密共享，文献[19]提出了安全加法−乘法转换（STAM）协议和安全乘法−加法转换（STMA）协议支持基本模块的构建。STAM 协议的输入是$[x]_0$和$[x]_1$，输出是u和v，满足$[x]_0 + [x]_1 = uv$。STMA 协议满足$[x]_0[x]_1 = u + v$。

为了减少在线通信开销，ABY2.0 中引入了一种新的秘密共享，本文用〈·〉秘密共享表示，秘密$x \in \mathbb{Z}_{2^\ell}$被拆分为 3 个分量$\Delta_x,[\delta_x]_0,[\delta_x]_1 \in \mathbb{Z}_{2^\ell}$，满足$\Delta_x = x + [\delta_x]_0 + [\delta_x]_1$，参与方$P_0$和$P_1$分别持有分量$\langle x\rangle_0 = \{\Delta_x,[\delta_x]_0\}$和$\langle x\rangle_1 = \{\Delta_x,[\delta_x]_1\}$。

ABY2.0 提出的安全乘法协议中，参与方P_i计算并发送给$P_{1-i}$$[\Delta_z]_i = i\Delta_{xy} - \Delta_x[\delta_y]_i - \Delta_y[\delta_x]_i + [\delta_{xy}]_i + [\delta_z]_i$，其中，$[\delta_z]_i$为参与方$P_i$在离线阶段获取的随机数，在线通信量为 2 个环元素，与加法秘密共享的乘法相比，该协议在线通信量减少一半。对于矩阵$\boldsymbol{A}^{p\times q}$和$\boldsymbol{B}^{q\times r}$，ABY2.0 提出的安全矩阵乘法协议的在线通信量为$O(pr)$，而加法秘密共享在线通信量为$O(pqr)$。

3　模型建立

3.1　系统模型

与文献[20-22]类似，SViT 采用 2 个边缘服务器进行计算。这是由于参与计算的服务器每增加一方（三方或者多方），通信量和被攻击的风险都会随之增长，并且三方和多方计算通常基于诚实多数的设定，因此 SViT 采用两方计算模型，增强安全性，降低通信量。

在 SViT 系统模型中，包含 3 类主要的实体：用户U、2 个边缘服务器（P_0和P_1）和一个可信第三方服务器T，如图 1 所示。

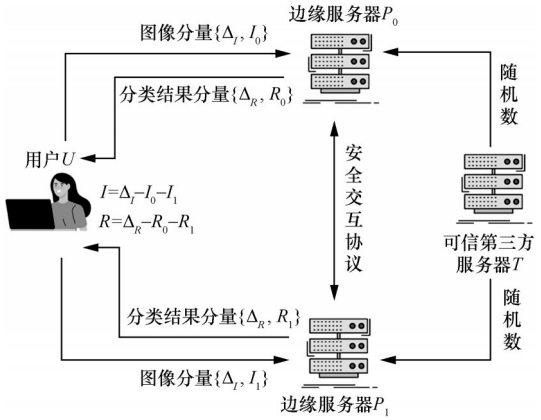


图 1　系统模型

用户U基于秘密共享技术将查询图像拆分成两组分量，分别提交给 2 个边缘服务器P_0和P_1。

可信第三方服务器T负责在离线阶段生成随机数，发送给2个边缘服务器P_0和P_1，确保安全协议的执行。

2个边缘服务器P_0和P_1收到用户发送的秘密共享分量后，通过一系列安全计算协议执行特征提取和分类任务，然后将各自的结果分量发送给用户U。

用户U仅需执行减法操作即可恢复出图像分类结果。

3.2 威胁模型

与文献[2,8,11]假设相同，本文方案采用半诚实的威胁模型，每个边缘服务器将诚实地遵循协议规定，但是试图在执行计算协议的过程中猜测隐私或者敏感信息。此外，2个边缘服务器相互独立且不共谋，这意味着它们不会将协议交互之外的信息透露给对方。上述假设在实际应用中是合理且可行的，例如，它们可以分别属于不同的大型服务提供商，共谋成本高，任何不诚实的行为都会损害其商业信誉。可信第三方服务器T仅负责产生随机数，轻量级服务器或者用户客户端均可胜任，由于不参与具体计算过程，因此对这个模型的安全性没有影响。另外，本文假定用户是诚实的，提交的数据是可靠的。

4 基础安全计算协议

4.1 SExp协议

给定$\langle X\rangle$，SExp协议输出$\langle \mathrm{e}^X\rangle$。推导过程为$\mathrm{e}^X=\mathrm{e}^{\Delta_X-[\delta_X]_0-[\delta_X]_1}=\mathrm{e}^{\Delta_X-[\delta_X]_0}\mathrm{e}^{-[\delta_X]_1}$，$P_0$和$P_1$分别计算$\mathrm{e}^{\Delta_X-[\delta_X]_0}$和$\mathrm{e}^{-[\delta_X]_1}$。这里使用STMA求得

$$[V]_0+[V]_1=\mathrm{e}^{\Delta_X-[\delta_X]_0}\mathrm{e}^{-[\delta_X]_1}=\mathrm{e}^X$$

为了盲化$[V]_0$和$[V]_1$，SExp协议需要加入随机数。离线阶段，各方分别收到可信第三方服务器T产生的随机值。在线阶段，SExp协议只需要2轮通信，共4个元素。

协议1 SExp协议

输入 P_i拥有$\langle X\rangle_i$

输出 $\langle Z\rangle=\mathrm{Exp}(X)$

离线阶段

可信第三方服务器T发送随机数$[\delta_Z]_i$给边缘服务器P_i

在线阶段

1) P_i计算

$$U_0=\mathrm{e}^{\Delta_X-[\delta_X]_0}$$

$$U_1=\mathrm{e}^{-[\delta_X]_1}$$

2) P_i计算$[\Delta_Z]_i$并将其发送给P_{1-i}

$$[V]=\mathrm{STMA}(U_0,U_1)$$

$$[\Delta_Z]_i=[V]_i+[\delta_Z]_i$$

3) P_i计算

$$\Delta_Z=[\Delta_Z]_0+[\Delta_Z]_1$$

4.2 SDiv协议

给定$\langle X\rangle$和$\langle Y\rangle$，SDiv协议输出$\left\langle\frac{X}{Y}\right\rangle$。首先，用函数STAM做加法到乘法关系的转换

$$U_0U_1=\Delta_X-[\delta_X]_0-[\delta_X]_1$$

$$V_0V_1=\Delta_Y-[\delta_Y]_0-[\delta_Y]_1$$

则$\langle X\rangle=U_0U_1$，$\langle Y\rangle=V_0V_1$。各参与方本地计算$K_i=\frac{U_i}{V_i}$，然后用函数STMA做乘法到加法关系的转换$K_1K_2=[W]_0+[W]_1$，P_i分别得到$[W]_i$，为保持ABY2.0的语义，$[W]_i$需要分别加上随机数进行盲化。在离线阶段，各方取一个随机值。

协议2 SDiv协议

输入 P_i拥有$\langle X\rangle_i$，$\langle Y\rangle_i$

输出 $\langle Z\rangle=\mathrm{Div}\left(\frac{X}{Y}\right)$

离线阶段

可信第三方服务器T发送随机数$[\delta_Z]_i$给边缘服务器P_i

在线阶段

1) P_i交互计算

$$V_0,V_1=\mathrm{STAM}\left(\Delta_X-[\delta_X]_0,[\delta_X]_1\right)$$

$$V_0,V_1=\mathrm{STAM}\left(\Delta_Y-[\delta_Y]_0,[\delta_Y]_1\right)$$

2) P_i计算

$$K_i=\frac{U_i}{V_i}$$

3) P_i计算$[\Delta_Z]_i$并将其发送给

$$P_{1-i}[W]=\mathrm{STMA}\left(K_1,K_2\right)$$

$$[\Delta_Z]_i=[W]_i+[\delta_Z]_i$$

4) P_i计算

$$\Delta_Z=[\Delta_Z]_0+[\Delta_Z]_1$$

4.3 SSqrt协议

给定$\langle X\rangle$，SSqrt协议输出$\left\langle\sqrt{X}\right\rangle$。首先，调用函数STAM做加法到乘法的转换$U_0U_1=\Delta_X-[\delta_X]_0-$

$[\delta_X]_1$，各参与方本地计算 $V_i=\sqrt{U_i}$，然后用函数 STMA 做乘法到加法关系的转换，$V_0V_1=[W]_0+[W]_1$，各参与方 P_i 分别得到 $[W]_i$，为保持 ABY2.0 的语义，$[W]_i$ 需要分别加上随机数进行盲化。

协议 3　SSqrt 协议

输入　P_i 拥有 $\langle X\rangle_i$

输出　$\langle Z\rangle=\mathrm{SSqrt}(X)$

离线阶段

可信第三方服务器 T 发送随机数 $[\delta_Z]_i$ 给边缘服务器 P_i

在线阶段

1)　P_i 交互计算

$$U_0,U_1=\mathrm{STAM}\left(\Delta_X-[\delta_X]_0,-[\delta_X]_1\right)$$

2)　P_i 计算

$$V_i=\sqrt{U_i}$$

3)　P_i 计算并将 $[\Delta_Z]_i$ 发送给 P_{1-i}

$$[W]=\mathrm{STMA}(V_0,V_1)$$
$$[\Delta_Z]_i=[W]_i+[\delta_Z]_i$$

4)　P_i 计算

$$\Delta_Z=[\Delta_Z]_0+[\Delta_Z]_1$$

5　构造 SViT

SViT 模型结构如图 2 所示，相应地，边缘服务器 P_0 和 P_1 顺序执行安全嵌入层、SLayerNorm、安全全连接层、SMatMul、SSoftmax、SGeLU 等相关安全计算协议。其中安全嵌入层和安全全连接层只需在边缘服务器上进行计算，SLayerNorm、SMatMul、SSoftmax、SGeLU 则需要进行交互计算。SMatMul 采用 ABY2.0 提出的协议，SViT 设计了 SLayerNorm、SSoftmax 和 SGeLU 安全计算协议。

5.1　SLayerNorm

LayerNorm 功能是对每个图像块的指定维度进行归一化处理。计算 LayerNorm 之前，需要先计算方差，方差的计算式为

$$\mathrm{Var}=\frac{1}{N-1}\sum_{i=1}^{N}(X_i-\bar{X})^2 \tag{2}$$

其中，$\bar{X}$ 是 X_i 的均值，N 是向量 X_i 中的元素个数。首先设计安全方差协议 SVar，已知边缘服务器 P_0 和 P_1 分别拥有输入 $\langle X\rangle_0$ 和 $\langle X\rangle_1$，这里 X 代表张量，$\langle X\rangle_i=\{\Delta_X,[\delta_X]_i\}$，$[\delta_X]_i$ 代表加法秘密共享。首先边缘服务器各自在计算 $\langle X\rangle_i$ 的均值 mean($\langle X\rangle_i$)，再计算 $\langle X\rangle_i$ 与 mean($\langle X\rangle_i$) 的差值 $\langle Y\rangle_i$，然后求 $\langle Y\rangle_i$ 的平方值 $\langle U\rangle$，这里需要用到 ABY2.0 中的乘法计算协议 SMul。边缘服务器继续在 $\langle U\rangle$ 最后一个维度求和，结果为 $\langle \mathrm{SUM}\rangle$。最后边缘服务器计算 $\langle \mathrm{SUM}\rangle$ 除以 $N-1$，此处为本地计算。具体过程如协议 4 所示。

协议 4　SVar 协议

输入　P_i 拥有 $\langle X\rangle_i$

输出　$\langle Z\rangle=\mathrm{Var}(X)$

在线阶段

1)　P_i 计算

$$\Delta_Y=\Delta_X-\mathrm{mean}(\Delta_X)$$
$$[\delta_Y]_i=[\delta_Y]_i-\mathrm{mean}([\delta_Y]_i)$$

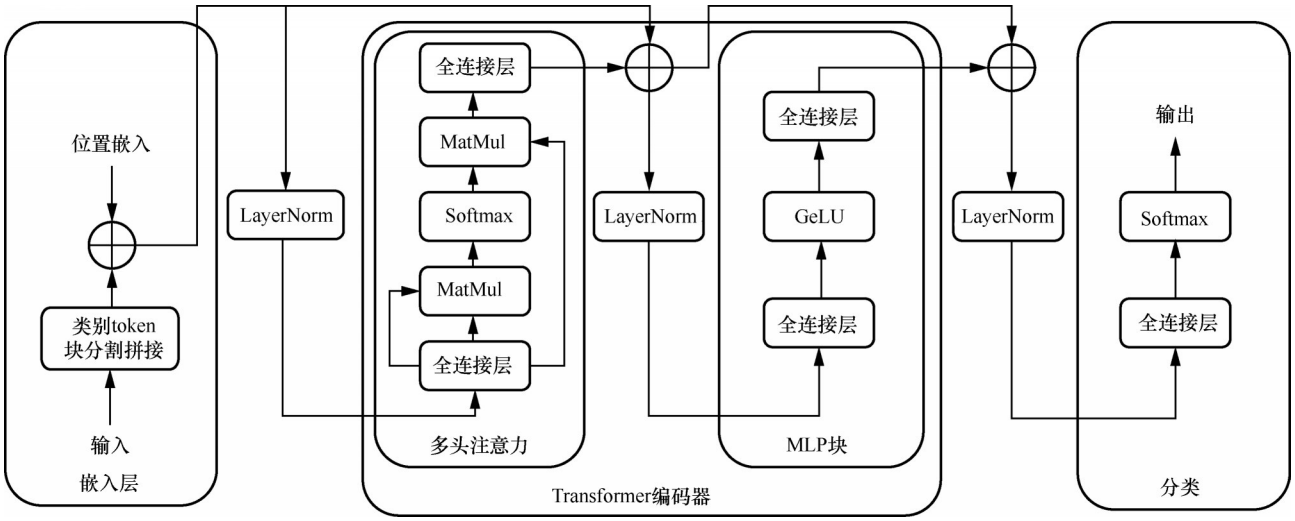


图 2　SViT 模型结构

2) P_i交互计算

$$\langle U\rangle = \mathrm{SMul}(\langle Y\rangle,\langle Y\rangle)$$

3) P_i计算

$$\langle \mathrm{SUM}\rangle = \mathrm{sum}(\langle U\rangle)$$

$$\langle Z\rangle = \frac{\langle \mathrm{SUM}\rangle}{N-1}$$

求得方差之后，可以进一步求出LayerNorm。LayerNorm计算式为

$$Y = \frac{X - \mathrm{E}(X)}{\sqrt{\mathrm{Var}(X)+\varepsilon}}\gamma + \beta \tag{3}$$

其中，$\mathrm{E}(X)$是均值，$\mathrm{Var}(X)$是方差，ε是添加到分母的值，防止分母为0，这里ε=10^{-6}，γ和β是通过训练得到的权重和偏差。根据式(3)，SViT设计了SLayerNorm协议。已知边缘服务器P_i拥有输入$\langle X\rangle_i$、公开权重w和公开偏差b。首先边缘服务器各自计算$\langle X\rangle_i$的均值$\langle Y\rangle_i$。然后执行安全方差协议SVar求得$\langle X\rangle$的方差$\langle U\rangle$后，继续计算$\langle U\rangle+\varepsilon$，注意这里$\langle U\rangle = \left\{\Delta_U,[\delta_U]_0,[\delta_U]_1\right\}$，$\langle U\rangle+\varepsilon$等价于$\Delta_U+\varepsilon$，将输出的值进行安全平方根计算得到$\langle V\rangle$之后，执行安全除法计算$\langle X\rangle-\langle Y\rangle$除以$\langle V\rangle$得到$\langle M\rangle$。最后，边缘服务器计算$\langle M\rangle$乘以公开权重$w$之后与偏差$b$求和。具体过程如协议5所示。

协议5 SLayerNorm协议

输入 P_i拥有$\langle X\rangle_i$、公开权重w和公开偏差b

输出 $\langle Z\rangle = \mathrm{LayerNorm}(X)$

在线阶段

1) P_i计算

$$\Delta_Y = \Delta_X - \mathrm{mean}(\Delta_X)$$

$$[\delta_Y]_i = [\delta_Y]_i - \mathrm{mean}([\delta_Y]_i)$$

2) P_i交互计算

$$\langle U\rangle = \mathrm{SVar}(\langle X\rangle)$$

$$\langle V\rangle = \mathrm{SSqrt}(\langle U\rangle+\varepsilon)$$

$$\langle M\rangle = \mathrm{SDiv}((\langle X\rangle-\langle Y\rangle),\langle V\rangle)$$

3) P_i计算

$$\Delta_Z = w\Delta_M + b$$

$$[\delta_Z]_i = w[\delta_M]_i$$

5.2 SSoftmax

Softmax为每个输出分类的结果赋予一个概率值，表示每个类别的可能性。通过Softmax函数可以将多分类的输出值转化为范围在[0,1]且和为1的概率分布。SViT采用式(4)计算。

$$\mathrm{Softmax}(X_i) = \frac{\exp(X_i - \max(X_i))}{\sum_{i=1}^{n}\exp(X_i - \max(X_i))} \tag{4}$$

其中，X_i代表向量。为了防止输入过大引起溢出，输入值需要先减去最大值。SSoftmax协议计算时，边缘服务器拥有输入$\langle X\rangle$，首先求出$\langle X\rangle$与最大值的差值，这里求最大值用的是ABY2.0中的Maxpool协议，再求出差值对应的指数函数值$\langle U\rangle$。然后边缘服务器对$\langle U\rangle$求和得到$\langle Y\rangle$，其中$\langle Y\rangle = \left\{\Delta_Y,[\delta_Y]_0,[\delta_Y]_1\right\}$。最后2个边缘服务器交互计算$\langle Y\rangle$和$\langle U\rangle$的商。具体过程如协议6所示。

协议6 SSoftmax协议

输入 P_i拥有$\langle X\rangle_i$

输出 $\langle Z\rangle = \mathrm{Softmax}(X)$

在线阶段

1) P_i交互计算

$$\langle U\rangle = \mathrm{SExp}(\langle X\rangle - \mathrm{Max}(\langle X\rangle))$$

2) P_i计算

$$\Delta_Y = \mathrm{sum}(\Delta_U)$$

$$[\delta_Y]_i = \mathrm{sum}([\delta_U]_i)$$

3) P_i计算

$$\langle Z\rangle = \mathrm{SDiv}(\langle U\rangle,\langle Y\rangle)$$

5.3 SGeLU

激活函数GeLU相较于ReLU更加平滑，优点是在处理负数时不会将输入截断为0，而ReLU会导致梯度消失的问题。当输入大于0，GeLU基本是线性输出；当输入接近0，输出为非线性；当输入小于0且不接近0，输出为0。

SViT采用下面的GeLU近似算法

$$\mathrm{GeLU} = 0.5X\cdot\left(1+\tanh\left(\mathrm{SQrt}\left(\frac{2}{\mathrm{pi}}\right)(X+0.044\,715\mathrm{pow}(X,3))\right)\right) \tag{5}$$

其中，$\tanh = \dfrac{e^X - e^{-X}}{e^X + e^{-X}} = \dfrac{2e^X}{e^X + e^{-X}} - 1$。

SGeLU协议计算时，离线阶段，可信第三方服务器分别给2个边缘服务器发送随机数$[B]_i$；在线阶段，边缘服务器首先调用STAM协议，实现加法关系到乘法关系的转换，即$A_0A_1 = \left(\Delta_X - [\delta_X]_0\right) + \left(-[\delta_X]_1\right)$。然后边缘服务器在本地各自计算$A_0^3$和$A_1^3$后，转换为加法秘密共享$[B]_0$和$[B]_1$。

接着计算$\langle C\rangle=\sqrt{\frac{2}{\mathrm{pi}}}\left(\langle X\rangle+0.044\,715\langle B\rangle\right)$。最后计算$\frac{2\mathrm{SExp}\left(\langle C\rangle\right)}{\mathrm{SExp}\left(\langle C\rangle\right)+\mathrm{SExp}\left(\langle -C\rangle\right)}-1$得到GELU($X$)的结果。具体过程如协议7所示。

协议7　SGeLU协议

输入　P_i拥有$\langle X\rangle_i$

输出　$\langle Z\rangle=\mathrm{GeLU}(X)$

离线阶段

可信第三方服务器T发送随机数$[B]_i$给边缘服务器P_i

在线阶段

1) P_i交互计算

$$A_0,A_1=\mathrm{STAM}((\Delta_X-[\delta_X]_0),-[\delta_X]_1)$$

2) P_i计算

$$A_i=(A_i)^3$$

3) P_i交互计算

$$[B]=\mathrm{STMA}\left(A_0,A_1\right)$$

4) P_i计算并发送给P_{1-i}

$$\left[\Delta_B\right]_i=[B]_i+\left[\delta_B\right]_i$$

5) P_i计算

$$\Delta_B=\left[\Delta_B\right]_0+\left[\Delta_B\right]_1$$

$$\langle B\rangle=\langle X\rangle+0.044\,715\langle B\rangle$$

$$\langle C\rangle=\sqrt{\frac{2}{\mathrm{pi}}}\langle B\rangle$$

6) P_i交互计算

$$\langle D\rangle=\mathrm{SExp}\left(\langle C\rangle\right)$$

$$\langle H\rangle=\mathrm{SExp}(-C)$$

$$\langle D\rangle=\mathrm{SDiv}\left(\langle D\rangle,\langle H\rangle\right)$$

$$\langle Z\rangle=\mathrm{SMul}\left(\langle X\rangle,2D\right)$$

$$\langle Z\rangle=0.5\langle Z\rangle$$

6　理论分析

6.1　安全性分析

SViT提供半诚实模型下的安全性。在半诚实模型下，允许敌手$\mathcal{A}$最多攻击2个边缘服务器中的一个。基于文献[23]的安全证明研究基础，SViT的安全性定义和引理如下。

定义1　对于任意概率多项式时间敌手$\mathcal{A}$，如果存在模拟器S可以构造模拟协议，使模拟视图与真实执行环境的视图计算不可区分，则多方计算协议就是可证明安全的[24]。

引理1　如果安全计算协议在概率多项式时间是可模拟的，那么依此为基础构成的系统必定存在一个对应的组合模拟，使敌手视图与真实执行环境的敌手视图计算不可区分。

引理2　如果r是敌手未知的均匀随机数且与x不相关，那么$x\pm r$必定是均匀随机的。

接下来，本节将基于以上2个引理，证明本文设计的协议的安全性。

定理1　SExp、SDiv、SSqrt协议在半诚实模型下是安全的。

证明　SExp协议的执行包括三次本地计算和一次交互计算STMA。本地计算没有信息交互，可以被完美模拟，关于交互计算，根据引理1，SExp协议可以直接使用STMA的模拟器[16]来模拟。可证SExp在半诚实模型下保证安全。SDiv包括四次本地计算和两次交互计算STAM、STMA，因此SDiv基于STAM和STMA模拟器来模拟。同理SSqrt使用STAM和STMA模拟器来模拟。因此SExp、SDiv、SSqrt协议在半诚实模型下是安全的。证毕。

定理2　SLayerNorm、SStd、SSoftmax、SGeLU协议在半诚实模型下是安全的。

证明　SStd协议在执行过程中的交互协议为SMul和SSqrt，因此该协议基于SMul和SSqrt模拟器来模拟。根据引理1，SStd协议可以直接使用SMul和SSqrt的模拟器来模拟。同理，SLayerNorm协议中的交互协议为SStd和SDiv，因此SLayerNorm协议基于SStd和SDiv模拟器进行模拟。SSoftmax协议中的交互协议为SExp和SDiv，因此该协议基于SExp和SDiv模拟器构造。SGeLU协议的安全性基于STMA、STAM、SDiv、SExp和SMul。因此SStd、SLayerNorm、SSoftmax、SGeLU协议在半诚实模型下是安全的。证毕。

定理3　SViT模型在半诚实模型下是安全的。

证明　SViT模型包含嵌入层、Transofrmer编码器、MLP分类等部分。其中嵌入层均为本地操作，没有信息交互。Transformer编码器中，主要包含的交互协议为SLayerNorm、SSfotmax、SGeLU，定理2中已证明其安全性。MLP分类调用SLayerNorm和SSfotmax协议。根据引理1，SViT模型在半诚实模型下是安全的。证毕。

6.2　复杂度分析

本节对SViT中设计的安全计算协议进行计算复杂度和通信复杂度分析。

安全计算协议的计算复杂度如表1所示。表1中，n表示输入向量长度，m表示子协议迭代次数。

表1 安全计算协议的计算复杂度

协议	CrypTen	SViT
SExp	$O(nm)$	$O(n)$
SDiv	$O(nm)$	$O(n)$
SSqrt	$O(nm)$	$O(n)$
SVar	$O(n)$	$O(n)$
SSoftmax	$O(nm)$	$O(n)$
SLayerNorm	—	$O(n)$
SGeLU	—	$O(n)$

对于SExp，CrypTen利用公式$\exp(x)=\lim\limits_{n\to\infty}\left(1+\frac{x}{n}\right)^n$近似求值，其中$n=2^m$，$m$为迭代次数。因此CrypTen的计算复杂度为$O(nm)$。而SViT只需执行STMA和本地计算，计算复杂度为$O(n)$。

对于SDiv，CrypTen利用Newton-Raphson迭代法近似求值，迭代次数为m，因此计算复杂度为$O(nm)$，SViT通过调用STMA、STAM和本地计算实现，计算复杂度为$O(n)$。

CrypTen在计算SVar的过程中调用了经过优化的平方计算，计算复杂度为$O(n)$，SViT中SVar计算复杂度也为$O(n)$。

SSoftmax在执行计算过程中调用了SDiv、SSqrt、SExp的一个或多个算子，因此CrypTen的计算复杂度为$O(nm)$，而SViT计算复杂度为$O(n)$。

安全计算协议的在线通信复杂度如表2所示。

对于SExp协议，CrypTen默认执行8次安全平方操作，共8轮通信，通信复杂度为$16nl$。SViT中共2轮通信，通信复杂度为$4nl$。

对于SDiv协议，CrypTen首先判断分母的符号位，需要$\mathrm{lb}\,l$轮通信，通信复杂度为$n(12l-16)$。然后调用倒数计算和乘法计算，需要11轮通信，通信开销为$44nl$，因此CrypTen计算安全除法共需要$33+\mathrm{lb}\,l$轮通信，通信复杂度为$142nl$。SViT进行SDiv操作调用2次STAM，1次STMA，共通信6轮，通信复杂度为$10nl$。

对于SSqrt协议，CrypTen需要计算安全平方根倒数和1次乘法，需要18轮通信，通信复杂度为$50nl$。SViT中需要调用STMA和STAM各1次，共需要4轮通信，通信复杂度为$7nl$。

对于SVar协议，CrypTen调用一次安全平方操作，共通信1轮，通信复杂度为$2nl$。SViT中需要调用1次SMul，进行1轮通信，通信复杂度为$2nl$。

对于SLayerNorm协议，CrypTen没有实现。SViT需要调用SVar、SSqrt、SDiv各一次，共需11轮通信，通信复杂度为$19nl$。

对于SGeLU协议，CrypTen没有实现。SViT需要调用STAM、STMA、SDiv、SMul各1次，SExp 2次，共计15轮通信，通信复杂度为$27nl$。

7 实验结果

本节将从SViT安全计算协议性能和误差及SViT框架性能2个方面进行评估。

实验环境：实验设备为两台服务器和一台笔记本计算机，服务器配置为Intel(R) Xeon(R) Platinum 8255c CPU @2.5 GHz， 38 GB RAM，NVIDIA Tesla V100-SXM2 GPU，32G VRAM。笔

表2 安全计算协议的在线通信复杂度

协议	CrypTen		SViT	
	通信轮数	通信开销	通信轮数	通信开销
SExp	8	$16nl$	2	$4nl$
SDiv	$33+\mathrm{lb}\,l$	$142nl$	6	$10nl$
SSqrt	18	$50nl$	4	$7nl$
SVar	1	$2nl$	1	$2nl$
SLayerNorm	—	—	11	$19nl$
SGeLU	—	—	15	$27nl$

注：n表示输入向量长度，l表示整数环大小。

记本配备为 4 核 1.80 GHz Intel i7-10510U CPU 和 16 GB RAM。SViT 模型基于 Python3.7 实现，利用 PyTorch 库实现基于秘密共享的安全计算协议。

数据集：使用花朵数据集进行 SViT 模型实验。该数据集是由伯明翰大学计算机学院提供的开源库，包含 4 317 张图像，涵盖雏菊、蒲公英、玫瑰、向日葵、郁金香 5 种花卉。其中，3 817 张作为训练样本，其余作为测试样本。

预处理：本文采用迁移学习[25]的方法，对预训练模型在明文环境下用花卉数据集进行微调，微调的 epoch 为 5。输入的图像缩放至 256×265，然后裁剪为 224×224，并进行归一化处理。

7.1　安全计算协议性能和误差

本节首先从运行时间和在线通信量 2 个方面评估安全计算协议的性能，然后进行了协议的误差分析。

表 3 比较了 SViT、MP-SPDZ 和 CrypTen 的安全计算协议的性能，输入均为 100×100 的矩阵。对于安全除法 SDiv，MP-SPDZ 和 CrypTen 运行时间分别为 885.8 ms 和 9.1 ms，而 SViT 仅为 1.4 ms。对于在线通信量，前两者分别为 176 MB 和 10.833 7 MB，而 SViT 为 0.762 9 MB。SViT 在其他计算协议的运行时间和在线通信量也均为最优，主要有以下几个原因：1) MP-SPDZ 仅支持向量运算，因此需要通过循环的方式来实现张量计算，从而增加了计算的开销；2) CrypTen 中的 SExp、SDiv 和 SSqrt 等协议采用了 Newton-Raphson 等迭代法来实现，这导致计算效率较低。此外，MP-SPDZ 和 CrypTen 都没有实现 ViT 模型中的关键非线性函数 SGeLU 和 SLayerNorm。

由于 SDiv、SSqrt 和 SExp 是构成其他协议的基础，因此本文针对这 3 个协议进行误差分析。图 3 分别比较了 SViT 和 CrypTen 中 SExp、SDiv、SSqrt 协议的计算误差，横坐标是输入范围，纵坐标是用均方根误差[26]（RMSE）指标来衡量的计算误差。根据图 3(a)可知，当输入值范围为 $(10^{-4},10^{-1})$ 时，CrypTen 的 RMSE 范围为 $(10^{-4},10^{-3})$，而 SViT 的误差范围为 $(10^{-8},10^{-7})$，因此 SViT 的 RMSE 小于 CrypTen。当输入值大于 10^{-1} 时，CrypTen 的 RMSE 急剧上升至 10^{0}，而 SViT 的 RMSE 为 10^{-5}。当 CrypTen 超出特定范围时，RMSE 显著增加。类似地，如图 3(b)和图 3(c)所示，当 CrypTen 输入范围超过 $(10^{-3},10^{0})$ 和 $(10^{-2},10^{1})$ 时，RMSE 会迅速超出合理范围。这是由于 CrypTen 基于迭代法实现上述安全计算协议，而迭代法对输入数据的范围有一定限制，超出范围后精度会降低。相比之下，SViT 并没有采用迭代法，因此从图 3 中可以观察到 SViT 的安全计算协议 RMSE 基本保持稳定。

7.2　SViT 框架性能

本节将从性能、误差和安全性 3 个方面对 SViT 框架进行分析。

实验从花卉数据集中随机抽取了 500 张图像作为样本。其中，雏菊、蒲公英、玫瑰、太阳花、郁金香每个类别各随机取 100 张图像作为测试集。

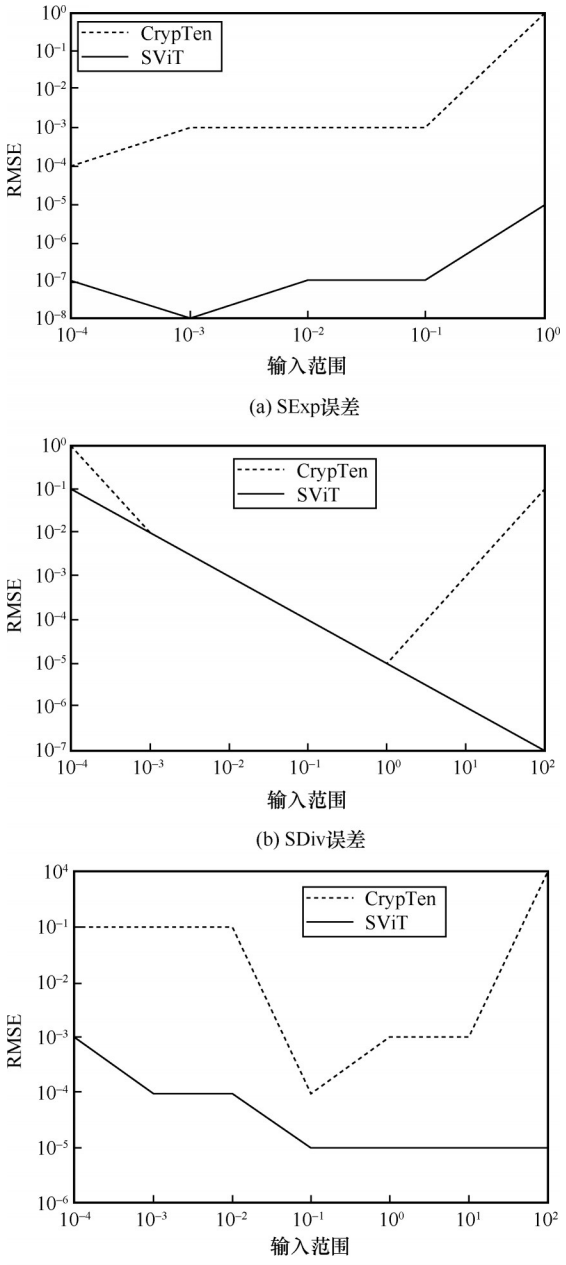
ViT-B/16 是一种经典的 ViT 图像分类明文框架，主要由嵌入层、Transformer 编码器层、分类层组成。其中，Transformer 编码器层由 12 个相同的模块组成，每个模块包含多头注意力子层。SViT 框架完全依据 ViT-B/16 架构实现，没有进行模型压缩或者其他结构的改变。

CrypTen 框架实现了隐私保护 ViT 推理模型，但不支持 GeLU、LayerNorm 等函数，同时由于仅

表 3　　**安全计算协议运行时间与在线通信量**

协议	运行时间/ms			在线通信量/MB		
	MP-SPDZ	CrypTen	SViT	MP-SPDZ	CrypTen	SViT
SExp	—	38.4	6.1	—	1.220 7	0.305 2
SDiv	885.8	9.1	1.4	176	10.833 7	0.762 9
SSqrt	1 982.9	266.4	116.8	358.08	3.814 6	0.534 1
SVar	158.12	9.4	2.5	0.662 4	0.152 6	0.152 6
SSoftmax	2 326.79	703.2	234.5	161.347	14.137 2	1.566 1
SLayerNorm	—	—	16.4	—	—	1.449 6
SGeLU	—	—	8.1	—	—	1.907 3

支持加密网络，计算效率较低，通信量较大，因此无法与SViT框架进行公平的比较。



(a) SExp误差

(b) SDiv误差

(c) SSqrt误差

图3　安全计算协议的误差比较

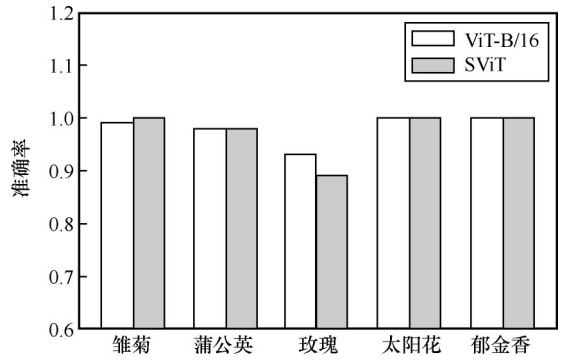
表4展示了SViT框架基于花卉数据集的运行时间和在线通信量。P_0和P_1首先执行嵌入层协议，主要为安全卷积协议，本地计算即可，不需要进行交互计算，运行时间为18.791 ms。接下来，通过Transformer编码器层提取图像特征，计算协议包括SMatmul、SSoftmax、SGeLU、SLayerNorm等，运行时间为313.551 ms，在线通信量为2.83 GB，这里指12个Transformer模块的平均运行时间和在线通信量。最后进入分类层，包括SSoftmax等计算协议，运行时间为18.34 ms，在线通信量为0.02 GB。综上所述，SViT中总运行时间为3 799.744 ms，总在线通信量为2.85 GB，90%的运行时间和80%的在线通信量是用于执行Transformer编码器层。这是由于Transformer编码器层由SSfotmax、SLayerNorm、SGeLU等复杂的非线性函数构成。虽然Transformer编码器带来了高昂的计算和通信开销，但它同时也为推理结果的精度提供了保证。

表4　SViT框架基于花卉数据集的运行时间和在线通信量

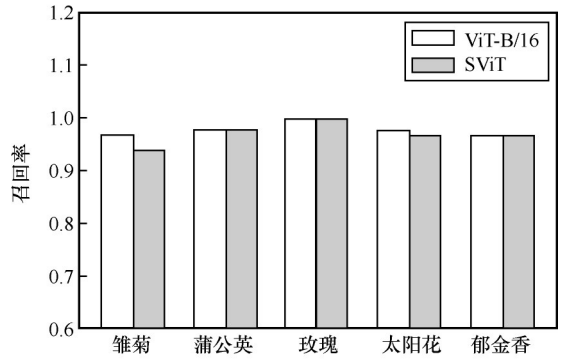
子层	运行时间/ms	在线通信量/GB
嵌入层	18.791	0
Transformer编码器层	313.551	2.83
分类层	18.34	0.02
总计	3 799.744	2.85

图4对比了ViT-B/16框架和SViT框架在图像分类任务上的推理准确率，使用了准确率、召回率和F1分数3个指标。准确率表示预测为正确的样本中实际为正确的比例；召回率表示被预测为错误的样本中被正确预测为错误的比例；F1分数综合考虑了准确率和召回率2个指标。通过观察图4(a)可以发现，对于玫瑰类别，SViT的准确率为0.89，稍低于ViT-B/16的准确率0.93，其他类别准确率相当。图4(b)中，对于雏菊和太阳花类别，SViT的召回率分别为0.94和0.97，稍低于ViT-B/16的0.97和0.98，其他类别召回率相当。图4(c)中，对于F1分数指标，SViT与ViT-B/16结果也非常接近。这是由于SViT框架基于高精度的安全计算协议，依据ViT-B/16架构组成，没有进行模型压缩等牺牲推理精度的措施。

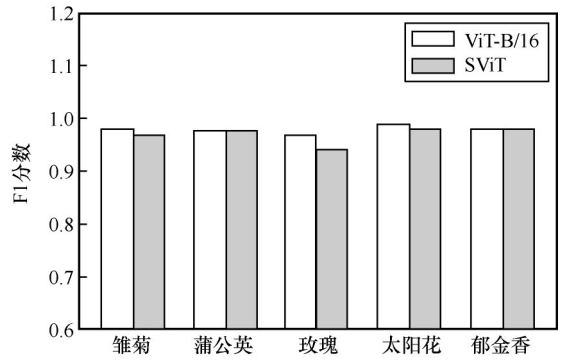
为了验证本文框架的安全性，实验从花朵数据集中随机抽取3张图像进行分类测试。如图5所示，给定原始图像I（第一列），用户随机生成图像分量I_0和I_1（第二列、第三列），原始图像与2个图像分量相加，生成新的图像分量Δ_I（第四列）。用户分别将$\{\Delta_I, I_0\}$和$\{\Delta_I, I_1\}$发送给2个边缘服务器，由于每个边缘服务器只有3个分量中的2个，因此无法获取原始图像的特征信息。



(a) 准确率

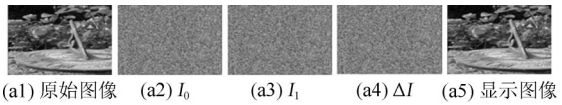


(b) 召回率



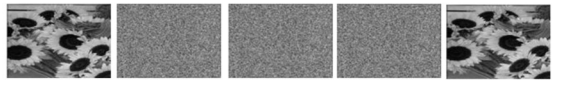
(c) F1分数

图4　ViT-B/16 框架和 SViT 框架的准确率、召回率和 F1 分数



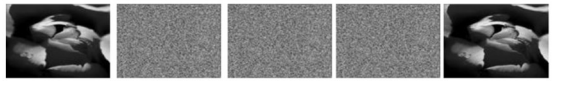
(a1) 原始图像　(a2) I_0　(a3) I_1　(a4) ΔI　(a5) 显示图像

(a) 玫瑰



(b1) 原始图像　(b2) I_0　(b3) I_1　(b4) ΔI　(b5) 显示图像

(b) 太阳花



(c1) 原始图像　(c2) I_0　(c3) I_1　(c4) ΔI　(c5) 显示图像

(c) 郁金香

图5　原始图像与秘密共享分量

8　结束语

针对图像处理过程中图像隐私泄露的风险，本文提出了隐私保护 ViT 推理框架 SViT。通过设计快速、在线通信量低、精度高的安全计算协议，使 2 个边缘服务器可以实现高效安全的推理，完成图像分类任务。实验结果表明，本文在计算效率和在线通信开销方面分别提升了 2~6 倍和 4~14 倍。未来的工作将继续研究进一步降低推理开销的方法，特别是对 Softmax、GeLU 等非线性函数的优化方法。

参考文献：

[1] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: transformers for image recognition at scale[J]. arXiv Preprint, arXiv: 2010.11929, 2020.

[2] HAO M, LI H, CHEN H, et al. Iron: private inference on transformers[J]. Advances in Neural Information Processing Systems, 2022, 35: 15718-15731.

[3] 李凤华, 李晖, 贾焰, 等. 隐私计算研究范畴及发展趋势[J]. 通信学报, 2016, 37(4): 1-11.
LI F H, LI H, JIA Y, et al. Privacy computing: concept, connotation and its research trend[J]. Journal on Communications, 2016, 37(4): 1-11.

[4] DONG Y, XIAOJUN C, JING W Z, et al. Meteor: improved secure 3-party neural network inference with reducing online communication costs[C]//Proceedings of the ACM Web Conference 2023. New York: ACM Press, 2023: 2087-2098.

[5] NG L K L, CHOW S S M. SoK: cryptographic neural-network computation[C]//Proceedings of 2023 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2023: 497-514.

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

[7] DEMMLER D, SCHNEIDER T, ZOHNER M. ABY - a framework for efficient mixed-protocol secure two-party computation[C]//Proceedings of the Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2015: 1-15.

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

[9] WAGH S, TOPLE S, BENHAMOUDA F, et al. FALCON: honest-majority maliciously secure framework for private deep learning[J]. arXiv Preprint, arXiv: 2004.02229, 2020.

[10] SRINIVASAN W Z, AKSHAYARAM P, ADA P R. Delphi: a cryptographic inference service for neural networks[C]//Proceedings of the 29th USENIX Conference on Security Symposium. Berkeley: USE, 2019: 2505-2522.

[11] RATHEE D, RATHEE M, GOLI R K K, et al. SIRNN: a math library

for secure rnn inference[C]//Proceedings of 2021 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2021: 1003-1020.

[12] 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.

[13] DEVLIN J, CHANG M W, LEE K, et al. Bert: pre-training of deep bidirectional transformers for language understanding[J]. arXiv Preprint, arXiv: 1810.04805, 2018.

[14] KELLER M. MP-SPDZ: a versatile framework for multi-party computation[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2020: 1575-1590.

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

[16] LI D C, WANG H Y, SHAO R L, et al. MPCFormer: fast, perfor-mant and private transformer inference with MPC[J]. arXiv Preprint, arXiv: 2211.01452, 2022.

[17] CHEN T Y, BAO H B, HUANG S H, et al. THE-X: privacy-preserving transformer inference with homomorphic encryption[J]. arXiv Preprint, arXiv: 2206.00216, 2022.

[18] WANG Y Q, SUH G E, XIONG W J, et al. Characterization of MPC-based private inference for transformer-based models[C]//Proceedings of the 2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). Piscataway: IEEE Press, 2022: 187-197.

[19] 熊金波, 周永洁, 毕仁万, 等. 边缘协同的轻量级隐私保护分类框架[J]. 通信学报, 2022, 43(1): 127-137.
XIONG J B, ZHOU Y J, BI R W, et al. Towards edge-collaborative, lightweight and privacy-preserving classification framework[J]. Journal on Communications, 2022, 43(1): 127-137.

[20] HUANG K, LIU X, FU S, et al. A lightweight privcy-preserving CNN feature extraction framework for mobile sensing[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1441-1455.

[21] 马敏, 付钰, 黄凯. 云环境下基于秘密共享的安全外包主成分分析方案[J]. 信息网络安全, 2023, 23(4): 61-71.
MA M, FU Y, HUANG K. A principal component analysis scheme for security outsourcing in cloud environment based on secret sharing[J]. Netinfo Security, 2023, 23(4): 61-71.

[22] 许春根, 薛少康, 徐磊, 等, 基于安全两方计算的高效神经网络推理协议[J]. 信息网络安全, 2023, 23(7): 22-30.
XU C G, XUE S K, XU L, et al. Efficient neural network inference protocol based on secure two-party computation[J]. Netinfo Security, 2023, 23(7): 22-30.

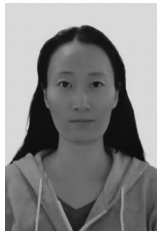
[23] RAN C. Security and composition of multiparty cryptographic protocols[J]. Journal of Cryptology, 2000, 13(1): 143-202.

[24] HAZAY C, LINDELL Y. Efficient secure two-party protocols: techniques and constructions[M]. Berlin: Springer Science & Business Media, 2010.

[25] PAN S J, YANG Q. A survey on transfer learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[26] WILLMOTT C J. Some comments on the evaluation of model performance[J]. Bulletin of the American Meteorological Society, 1982, 63(11): 1309-1313.

[作者简介]



马敏（1979– ），女，江苏扬州人，海军工程大学博士生，主要研究方向为信息安全、人工智能。

付钰（1982– ），女，湖北武汉人，博士，海军工程大学教授、博士生导师，主要研究方向为信息安全、人工智能。

黄凯（1986– ），男，安徽安庆人，博士，国防大学讲师，主要研究方向为人工智能、信息安全。

贾潇风（1998– ），男，河南许昌人，浙江工商大学硕士生，主要研究方向为应用密码学、区块链和隐私计算。