SAL-ViT: Towards Latency Efficient Private Inference on ViT using Selective Attention Search with a Learnable Softmax Approximation

Anonymous ICCV submission

Paper ID 10438

Abstract

Recently, private inference (PI) has addressed the rising concern over data and model privacy in machine learning inference as a service. However, existing PI frameworks suffer from high computational and communication overheads due to the expensive multi-party computation (MPC) protocols, particularly for large models such as vision transformers (ViT). The majority of this overhead is due to the encrypted `softmax` *operation in each self-attention layer. In this work, we present SAL-ViT with two novel techniques to boost PI efficiency on ViTs. Our first technique is a learnable PI-efficient approximation to* `softmax`*, namely, learnable 2Quad (*`L2Q`*), that introduces learnable scaling and shifting parameters to the prior* `2Quad` *softmax approximation, enabling improvement in accuracy. Then, given our observation that external attention (EA) presents lower PI latency than widely-adopted self-attention (SA) at the cost of accuracy, we present a selective attention search (SAS) method to integrate the strength of EA and SA. Specifically, for a given lightweight EA ViT, we leverage a constrained optimization procedure to selectively search and replace EA modules with SA alternatives to maximize the accuracy. Our extensive experiments show that our SAL-ViT can averagely achieve* $1.60\times$*,* $1.56\times$*,* $1.12\times$ *lower PI latency with* 1.79%*,* 1.41%*, and* 2.90% *higher accuracy compared to the existing alternatives, on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively.*

1. Introduction

The past few years have seen the tremendous success of transformer-based models in natural language processing (NLP) [28], largely because of their self-attention (SA) modules' ability to effectively capture long-range dependencies. Recently, vision transformers (ViTs) extended this success to computer vision tasks, including image classification [5, 10], object detection [1, 19], and semantic segmentation [19, 34, 31], by outperforming convolutional network architectures due to their lower inductive bias.

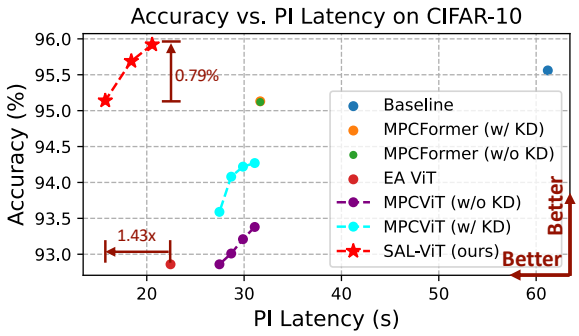

Figure 1. Performance comparison between prior works (Baseline, MPCFormer [18], EA ViT [8], MPCViT [32]) and ours on CIFAR-10. Note that the MPCFormer [18] was proposed on BERT that we have adapted to ViT.

The success of ViT and other deep neural network models have motivated emerging machine learning inference as a service (MLaaS), where a service provider trains the model and commercializes the inference service for various tasks including performing online diagnoses and financial product recommendations [17, 21]. However, growing privacy concerns have impeded such commercialization. In particular, clients may not wish to reveal their personal data to the service provider while the service providers wish to protect the details of their proprietary trained models [17]. In general, neither party wants to send sensitive unencrypted information to the other party. To mitigate these rising concerns, various private inference (PI) methods [23, 13, 22, 27, 11, 26] have been proposed that leverage techniques such as Homomorphic encryption (HE) and secure multi-party computation (MPC) protocols to preserve both the privacy of the client's data and the inference model's intellectual property (IP).

While some existing works explored efficient PI on convolutional neural networks (CNNs) [16, 2], the study of PI for transformers has been less explored. A direct implementation of existing PI methods on ViTs incurs dramatically higher latency and communication overhead than standard

inference, creating a significant roadblock in their wide-range adaptation, especially in resource-constrained applications [32, 18]. The high latency can be largely attributed to the softmax function, due to its high compute demand in PI [32, 18]. Interestingly, a recent work on BERT models [18] addresses this challenge by replacing the softmax with its 2^{nd} order polynomial approximation, 2Quad [3]. Also, for ViTs, [32] formulates a neural architecture search (NAS) algorithm to substitute the softmax with either the 2ReLU [23] or the scaling function [30].

However, these softmax approximations use a fixed constant to re-weight the attention maps, limiting the representation and thus costing accuracy. Because the heads at different layers aim at capturing diverse relations between patches, we hypothesize that a softmax approximation may need adaptable parameters to freely re-weight the attention map. With this motivation, we present a novel softmax approximation, namely, learnable 2Quad (L2Q), that has two different types of learnable parameters, shifting and scaling, enabling a fine-grained approximation of the softmax. More specifically, we provide three granularities of L2Q (global, head-wise, and element-wise) that differ in the degree of sharing of the learnable parameters across various instances of the approximation.

We further observe that, independent of the softmax approximation, the architecture of attention also plays a key role in both PI latency and accuracy. We compare recent attention architectures [29, 24, 6, 20, 8] and find that external attention (EA) yields the lowest PI latency due to the reduced involvement of softmax but at the cost of a significant drop in accuracy compared to an all SA ViT baseline. With this motivation, we propose to use a judicious hybrid of EA and SA modules with our L2Q approximation to achieve high accuracy while keeping the PI latency low. In particular, for a given SA module budget B and an initial ViT Model that uses all EA with L2Q, we present a selective attention search (SAS) that identifies the B EA modules to be replaced with SA to maximize accuracy. Thus, SAS provides a PI-friendly ViT architecture with a configurable hybrid of SA and EA, where each attention variant uses the L2Q approximation. We refer to the result as SAL-ViT, a PI-friendly ViT obtained through a selective attention search with a learnable softmax approximation.

We summarize our contributions as follows.

- We present a novel softmax alternative L2Q with fine-grained learnability that presents higher accuracy than existing softmax approximations, and a quadratic form that presents low PI latency.
- We present a detailed analysis of the various attention methods and their impact on PI latency and show that, compared to the SA baseline, EA [8] presents more than 2.7× lower PI latency at the cost of lower accuracy.
- We present a selective attention search (SAS) method to yield PI-friendly hybrid ViT models with a judicious mix of SA and EA, both leveraging our proposed L2Q.

Our experimental results show that our method outperforms the SOTA scheme MPCViT [32] by generating models with averagely 2.47%, 2.82% and 5.23% higher accuracy on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, and with averagely 1.61×, 1.58×, 1.09× lower PI latency on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. From Figure 1, our method produces ViTs with 1.43× lower PI latency compared to the lowest-PI-latency technique, i.e., EA ViT [8], and with 0.79% higher accuracy compared to the highest-accuracy PI ViT, i.e., MPCFormer [18] on CIFAR-10.

2. Background

2.1. Notations

In this paper, we use $\boldsymbol{X} \in \mathbb{R}^{N\times m}$ to denote an input sequence of N tokens with each token represented as a m-dimensional feature vector. There are three major components for the input feature, i.e., Query ($\boldsymbol{Q} \in \mathbb{R}^{N\times d_e}$), Key ($\boldsymbol{K} \in \mathbb{R}^{N\times d_e}$), and Value ($\boldsymbol{V} \in \mathbb{R}^{N\times d_e}$), obtained from three learnable linear matrices $\boldsymbol{W_Q} \in \mathbb{R}^{m\times d_e}$, $\boldsymbol{W_K} \in \mathbb{R}^{m\times d_e}$, and $\boldsymbol{W_V} \in \mathbb{R}^{m\times d_e}$ through $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W_Q}$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W_K}$, and $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W_V}$, where d_e is the embedding dimension of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. We use $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ to denote the attention map, which is obtained by performing $\boldsymbol{Q}\boldsymbol{K}^T$. The re-weighted normalized attention map is denoted as $\boldsymbol{A_N} \in \mathbb{R}^{N\times N}$.

2.2. Private Inference

Several PI frameworks using secret sharing (SS), MPC protocols, Garbled Circuits (GC), and oblivious transfer (OB) have been proposed on convolutional neural networks (CNNs) [13, 25, 22, 27, 11, 26, 33]. They observed that the computation and communication bottleneck for PI on CNNs is nonlinear functions like ReLU [7]. Two approaches have been studied to address this challenge. The first approach focuses on developing more efficient cryptographic protocols for ReLU [7, 11]. The second approach is to adapt the architecture of neural network models by either replacing ReLU with a less-costly quadratic function [22] or aggressively pruning ReLU [12, 2, 16].

For transformers, in contrast, the major bottleneck is the softmax function [18], which is seldom addressed in efforts focusing on CNN. One solution is to improve the cryptographic protocols for transformers. For example, Iron [9] develops efficient protocols for softmax, GELU, and LayerNorm, and proposes a customized HE-based

Function	Mathematical Description
`softmax`	$\texttt{softmax}(a_{ij}) = \frac{e^{a_{ij}}}{\sum_{j=1}^{N} e^{a_{ij}}+\epsilon}$ *
`2Quad` [18]	$\texttt{2Quad}(a_{ij}) = \frac{(a_{ij}+c)^2}{\sum_{j=1}^{N} (a_{ij}+c)^2}$ *
`2ReLU` [23]	$\texttt{2ReLU}(a_{ij}) = \frac{\text{ReLU}(a_{ij})}{\sum_{j=1}^{N} \text{ReLU}(a_{ij})+\epsilon}$ *
`scaling` [30]	$\texttt{Scale}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \frac{\boldsymbol{Q}}{\sqrt{n}}(\frac{\boldsymbol{K}^T}{\sqrt{n}}\boldsymbol{V})$

* ϵ is a small value to avoid a zero denominator.
* c is a constant.

Table 1. Softmax approximations

protocol to speed up high-dimensional matrix multiplication in transformers. An alternative approach is to develop PI-friendly transformer architectures and softmax approximations, such as MPCViT [32] and MPCFormer [18].

2.3. Softmax Approximations

For PI on transformers, the `softmax` operation corresponds to more than 67% and ∼80% of PI latency of a BERT [18] and a ViT model [32], respectively. Several approximations for the `softmax` function to mitigate the high latency of PI on transformers have been proposed. Given the Query, Key, Value matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$, and the attention map $\boldsymbol{A}$, Table 1 summarizes the softmax approximations where a_{ij} represents the element located at row i and in column j in $\boldsymbol{A}$. MPCViT [32] systematically compares the effects of these softmax alternatives in SA on vision tasks and concludes that 2ReLU provides the highest accuracy, and the scaling function yields the lowest PI latency.

2.4. Attention Variants

While SA yields the benefits of capturing long-distance dependencies, its computational and storage overheads increase quadratically ($O(N^2)$) with the size of the feature map [29]. To reduce these costs, attention variants with linear complexity ($O(N)$) have been widely studied [24, 29, 6, 20, 8].

SA leverages the scaled dot-product with `softmax` normalization to measure the similarity among $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. To reduce complexity, Linformer [29] learns to shrink the length of Key and Value matrices via projections. CosFormer [24] replaces SA with a linear projection kernel and a *cosine*-based re-weighting mechanism. Hamburger [6] reformulates learning the global context as a low-rank completion problem and solves it via matrix decomposition. SOFT [20] uses Gaussian kernel and exponential function to replace SA and solves it via Newton-Raphson iteration [4]. EA [8] leverages two lightweight external memories $\boldsymbol{W_k^{EA}}$ and $\boldsymbol{W_v^{EA}}$ to learn the most discriminative features across the entire dataset, and substitutes SA with two linear layers and a normalization layer.

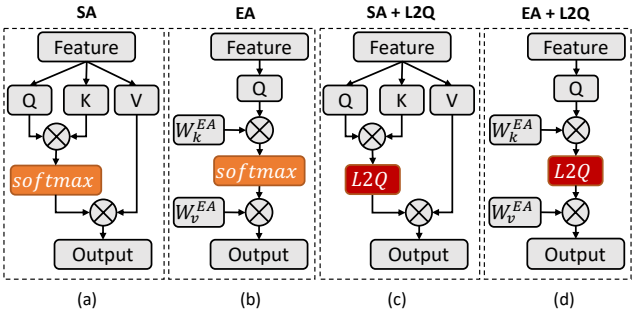

Figure 2. Brief overview of SA and EA. (a) SA with `softmax`, (b) EA with `softmax`, (c) SA with `L2Q`, (d) EA with `L2Q`

3. ViT Design for Latency Efficient PI

3.1. Motivation

While researchers have been aware of the high cost of the `softmax` function in PI on transformers, the effect of the attention architecture on PI latency has, to our understanding, yet to be explored.

Case study 1: We compare the PI latency of SA, illustrated in Figure 2(a), and its recent variants, including Linformer [29], CosFormer [24], Hamburger [6], SOFT [20] and EA [8], to find the most PI-friendly attention architecture. In all cases, we embedded the attention scheme in a 7-layer, 4-head ViT model [10] and measured the PI latency for 10 inference queries on CIFAR-10 using CrypTen [15]. The results, shown in Table 2, indicate that EA [8], illustrated in Figure 2(b), yields the lowest PI latency among all variants.

The high PI latency of CosFormer [24] is due to the introduction of *sin* and *cos* functions, which are expensive in PI [15]. Attention variants based on low-rank matrix decomposition and Gaussian kernel, i.e., Hamburger [6], and SOFT [20], rely on iterations to solve the matrix decomposition problem, which adds additional cost even if they have linear complexity. While Linformer [29] contains only projection and `softmax`, it still contains ∼2× more `softmax` computations than EA. In EA, due to the reduced size of tensor $\mathbf{W_k}$, the size of the `softmax` input is ∼4× smaller than SA, leading to the lowest PI latency. *We thus conclude that not all linear complexity approximations are suitable for latency-efficient PI.*

Case study 2: Although EA presents an advantage in reducing PI latency, it requires the `softmax` function to perform normalization. Ideally, replacing the `softmax` function with its PI-friendly approximation will further reduce the PI latency overhead. We compare the PI latency and accuracy on CIFAR-10 for different attention-softmax-

Attention	Complexity	PI Bottleneck	PI Latency (s)
Self-attention	$O(n^2)$	`softmax`	61.20
Linformer [29]	$O(n)$	`softmax`	48.39
CosFormer [24]	$O(n)$	cos/sin	$>40.29^*$
Hamburger [6]	$O(n)$	matrix decomposition	$>41.06^*$
SOFT [20]	$O(n)$	Gaussian kernel	$>45.17^*$
External attention [8]	$O(n)$	`softmax`	22.44

* Operations not supported by CrypTen [15] are not counted.

Table 2. Comparison of attention variants on CIFAR-10

Attention	Complex.	Softmax App.	Accuracy (%)	PI Latency (s)
Self-attention	$O(n^2)$	`softmax`	95.56	61.20
		`2Quad`	95.12	31.27
		`2ReLU`	95.24	32.90
		`scaling`	94.79	26.86
External attention	$O(n)$	`softmax`	92.86	22.44
		`2Quad`	92.20	14.08
		`2ReLU`	92.73	14.41
		`scaling`	33.45	13.30

Table 3. Performance comparison of SA and EA with various softmax approximations on CIFAR-10

approximation combinations in Table 3[1]. Table 3 shows EA achieves significantly lower latency than SA but at the cost of a significant drop in accuracy. *Therefore, a naive combination of EA with PI-friendly softmax approximations is not an ideal solution when high accuracy is desired.*

Based on these observations, we propose to optimize PI on ViT by adopting a PI-friendly softmax approximation and selectively using SA. In particular, we present a selective attention search (SAS) algorithm that begins with an all-EA ViT model and judiciously replaces some EA components with SA to improve accuracy. Below, we first describe an improved PI-friendly softmax approximation `L2Q`. Then, we detail our SAS for the ViT architecture with a mix of SA and EA, both of which use our proposed `L2Q` approximation.

3.2. Learnable 2Quad (`L2Q`)

For attention modules, a general form of re-weighted normalization of the attention map is as follows,

$$\frac{R(a_{ij})}{\sum_j R(a_{ij}) + \epsilon}, \tag{1}$$

where R is a re-weighting function and ϵ is a small positive value to avoid a zero denominator. While the widely-adopted `softmax` normalization function uses an exponential function for the re-weighting, the corresponding high PI latency has inspired the research community to find faster alternatives.

[1]The training hyperparameters are provided in Section 4

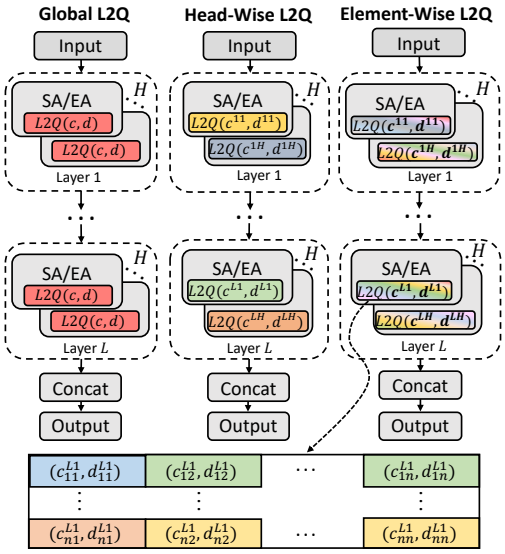

Figure 3. The proposed `L2Q` with various levels of granularity

To ensure better convergence, the normalized attention map is suggested to be positive, which implies the re-weighting function should produce non-negative outputs, and the re-weighting function should be differentiable [24, 14]. Moreover, a well-shaped non-linear re-weighting function stabilizes and amplifies the difference among the attention map elements more than a linear one [24]. Due to the nature of normalization, the summation of each row in $\boldsymbol{A_N}$ equals one. While `2Quad` satisfies all these requirements, the linear part of the re-weighting function in `2ReLU` and the linear `scaling` function make their performance unstable. For example, EA ViT only achieves an accuracy of 33.45% on CIFAR-10 with the `scaling` function as shown in Table 3. A similar performance deterioration is observed in our experiment on `2ReLU` in Section 4.2 (see Figure 6).

A drawback of `2Quad` is that it applies a universal constant c, as shown in Table 1, for all heads and layers in ViT. In general, multi-head transformer models can capture different relations between tokens in $\boldsymbol{X}$ [28]. Similarly, in ViT, the heads at different layers have varying attention distances. The average attention distance generally increases as the information passes through the model [5], and the distance can vary across heads within and across layers.

Therefore, we present a learnable re-weighting of the attention maps, referred to as learnable `2Quad` (`L2Q`), that introduces a learnable shifting parameter matrix $\boldsymbol{C}$ and a learnable scaling parameter matrix $\boldsymbol{D}$ for each head to improve the re-weighting ability of `2Quad`. We use $\boldsymbol{D}^{lh}$ and $\boldsymbol{C}^{lh}$ to represent the introduced learnable matrices at layer l, head h. The formal description of our `L2Q` on attention

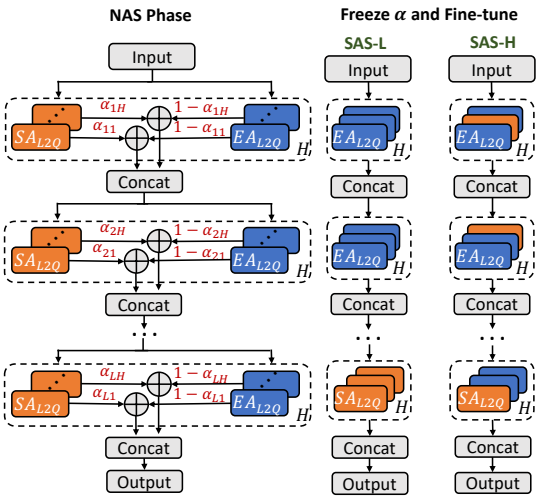

Figure 4. Overview of the proposed SAS

map $\boldsymbol{A}$ at layer l head h is as follows,

$$\texttt{L2Q}(a_{ij}) = \frac{(d_{ij}^{lh} \times a_{ij} + c_{ij}^{lh})^2}{\sum_j (d_{ij}^{lh} \times a_{ij} + c_{ij}^{lh})^2 + \epsilon}, \quad (2)$$

where d_{ij}^{lh} and c_{ij}^{lh} are learnable elements in $\boldsymbol{D}^{lh}$ and $\boldsymbol{C}^{lh}$, respectively. With this learnability, L2Q can find a better re-weighted attention distribution for each attention map. The implementation of L2Q in SA and EA is shown in Figure 2(c) and (d).

We present three levels of granularity for L2Q, namely, global, head-wise, and element-wise, as shown in Figure 3. Global L2Q applies the same pair of learnable shifting and scaling parameters for all attention maps across all the layers. Namely, $d_{ij}^{lh} = d, c_{ij}^{lh} = c, \forall i, j, l, h$, where d and c are two scalars. Head-wise L2Q shares the same pair of learnable parameters for all elements in head h at layer l, i.e., $d^{lh} = d_{ij}^{lh}, c^{lh} = c_{ij}^{lh}, \forall i, j$, where d^{lh} and c^{lh} are the learnable scalars for layer l head h. The most fine-grained, element-wise L2Q, has a d_{ij}^{lh} and a c_{ij}^{lh} for each element a_{ij} in the attention map at head h of layer l. In SAL-ViT, we use element-wise L2Q unless otherwise stated.

3.3. Selective Attention Search (SAS)

The overview of our SAS approach is shown in Figure 4. We initialize a ViT model with EA at each transformer layer and introduce a matrix $\boldsymbol{\alpha} \in \mathbb{R}^{L \times H}$, where L is the number of layers and H is the number of heads per layer, of learnable attention selection parameters to help decide between SA and EA for each layer. By replacing softmax in SA and EA modules with element-wise L2Q we obtain a ViT model with a mix of SA_{L2Q} and EA_{L2Q} modules. The SA_{L2Q} modules help achieve high classification accuracy at the cost of high PI latency, and EA_{L2Q} modules yield moderate accuracy with relatively low PI latency. Our SAS supports two levels of granularity, *layer-wise,* referred to as SAS-L, and *head-wise*, referred to as SAS-H. SAS-L defines a coarse-grained search space where each transformer layer uses either SA_{L2Q} or EA_{L2Q}. SAS-H makes a finer-grained replacement of EA_{L2Q} by deciding whether to use SA_{L2Q} at every attention head.

We denote a ViT model $\boldsymbol{\mathcal{M}}$ parameterized by $\boldsymbol{\Theta}$ as $\boldsymbol{\mathcal{M}}_{\boldsymbol{\Theta}}$, the input feature of a transformer layer l as $\boldsymbol{x}_l$, the parametric SA function as f^s, and the parametric EA function as f^e. For SAS, the output of l^{th} transformer layer can be formally expressed as:

$$z_l = \Big\|_{h=1}^{H} (\alpha_{lh} f_{lh}^s(\boldsymbol{x}_l) + (1 - \alpha_{lh}) f_{lh}^e(\boldsymbol{x}_l)), \quad (3)$$

where $\Big\|_{k=1}^{K}(a_k) = a_1 \| a_2 \| \cdots \| a_k$ represents the concatenation of all a_ks.

Specially, in SAS-L, all heads in the same transformer layer share the same attention selection parameter, i.e., $\alpha_{lh} = \alpha_l, \forall h$, where α_l is a scalar of the attention selection parameter at layer l. SAL-ViT adopts SAS-L unless otherwise stated.

SAS aims to obtain a ViT model with high accuracy and as few SA_{L2Q} modules as possible. We set an upper bound B for the number of SA_{L2Q} modules, referred to as the SA_{L2Q} budget, and formulate the training process as a constrained optimization problem:

$$\min_{\Theta, \alpha} \mathcal{L}(\boldsymbol{\mathcal{M}}_{\boldsymbol{\Theta}, \boldsymbol{\alpha}}(\boldsymbol{X}), y) \quad s.t. \sum_{l,h} \mathbb{1}(\alpha_{lh} \geq \alpha') \leq B, \quad (4)$$

where $\boldsymbol{X}$ and y are the input and corresponding label, $\mathbb{1}$ refers to the indicator function, and α' is the threshold for binarizing α_{lh}. With an SA_{L2Q} budget B, the threshold α' equals the B-th highest α_{lh}.

To solve this optimization problem, we separate the procedure of binarization and simplify the optimization problem as follows:

$$\min_{\Theta, \alpha} \mathcal{L}(\boldsymbol{\mathcal{M}}_{\boldsymbol{\Theta}, \boldsymbol{\alpha}}(\boldsymbol{X}), y) + \lambda \sum_{l,h} (||\alpha_{lh}||_1), \quad (5)$$

where λ is a hyper-parameter that helps balance the two loss terms. Here, the l_1-regularization term tries to minimize the number of SA_{L2Q} modules while the first term tries to maximize accuracy.

Our SAS algorithm is shown in Algorithm 1. In the NAS phase (line 2 - line 6), the network parameter $\boldsymbol{\Theta}$ and attention selection parameter $\boldsymbol{\alpha}$ are updated simultaneously with the loss function in Equation 5. $\boldsymbol{\alpha}$ thus learns the importance of replacing an instance of EA_{L2Q} with SA_{L2Q} to lower the accuracy loss, such that replacing an EA_{L2Q} at the location with a higher α_{lh} yields higher accuracy than replacing one with a lower α_{lh}. The NAS loop terminates when the number of NAS epochs hits the predefined limit E_{NAS}. Then, the top B α_{lh} are frozen to 1 and the remaining ones are frozen to 0 (line 7). After this step, SAS

Algorithm 1 Selective attention search

Inputs: An untrained ViT model $\mathcal{M}_{\Theta,\alpha}$, SA_{L2Q} budget B, the number of epoch for NAS phase E_{NAS}, the number of fine-tune epoch E_{FT}.
Output: A trained ViT model with hybrid self-attention and external attention.

1: $\mathcal{M}_{\Theta,\alpha}$.train()
2: $epoch = 0$
3: **while** $epoch < E_{NAS}$ **do**
4: Update (Θ, α) via ADAM optimizer for one epoch
5: $epoch += 1$
6: **end while**
7: Freeze(α) // set top B of α_{lh} to 1 and all others to 0.
8: $epoch = 0$
9: **while** $epoch < E_{FT}$ **do**
10: Update Θ via ADAM optimizer for one epoch
11: $epoch += 1$
12: **end while**
13: **return** $\mathcal{M}$

obtains a ViT containing both SA_{L2Q} and EA_{L2Q} modules. Finally, SAS fine-tunes the hybrid model's network parameters Θ (line 8-line 12).

4. Experiments

Experimental setup. We conduct our experiments on two types of CCT [10] ViT architecture on three datasets, CIFAR-10, CIFAR-100, and Tiny-ImageNet. The ViT depth, the number of heads, hidden dimension, and patch size, are set to 7, 4, 256, and 8, respectively, for the CIFAR-10 and CIFAR-100 datasets, and 9, 4, 196, and 16, respectively, for the Tiny-ImageNet dataset. The batch size for CIFAR-10 and CIFAR-100 is 512, and the batch size for Tiny-ImageNet is 256. We apply the same image augmentations as [10]. The training procedures are conducted on an Nvidia A100 GPU. PI latency is measured via CrypTen [15] under the semi-honest threat model [22] on a 2.3 GHz 8-Core Intel Core i9 CPU with 16 GB RAM. We keep in line with [22, 11] and use the LAN mode for network settings where the bandwidth and round-trip time between the two parties are 384 MBps and 0.3 ms, respectively.

For the experiments on SAS, the NAS phase is trained for $E_{NAS} = 600$ epochs for CIFAR-10/100 datasets, and $E_{NAS} = 100$ epochs for the Tiny-ImageNet dataset. The hyper-parameter λ is set to 0.1, and all attention selection coefficients in α are initialized to 0.1. The fine-tuning phase has $E_{FT} = 600$ epochs, uses the Adam optimizer, has a learning rate of 0.0006 and a weight decay of 0.06, for all three datasets.

The experiments on L2Q do not contain the NAS phase. Corresponding training parameters are the same as the ones in the fine-tuning phase of SAS.

4.1. Comparison of SAL-ViT with Prior-Art

In this section, we quantify the advantages of our proposed SAL-ViT. We measure the performance of SAL-ViT under three SA_{L2Q} budgets 4, 8, and 12, each specifying the number of heads in a ViT that are implemented with SA_{L2Q}. Note that SAL-ViT adopts SAS-L and element-wise `L2Q`, given the number of head per layer is 4, the SA_{L2Q} budgets of 4, 8, and 12 indicate that SAS-L searches for 1, 2, and 3 entire layers to be implemented by SA_{L2Q}, respectively. The attention-softmax-approximation combinations of the baseline and prior PI-efficient ViT frameworks, namely, MPCFormer [18] (with and without knowledge distillation (KD)), MPCViT [32] (with and without KD), and EA ViT [8], are presented in Table 4.

The results show that SAL-ViT always presents lower latency with similar or better accuracy. More precisely, our SAL-ViT models yield up to 3.89×, 3.83×, and 1.72× lower latency, and up to 0.36%, 0.5%, 2.54% higher accuracy than even the baseline ViT on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. Our SAL-ViT achieves up to 2.01×, 1.90×, and 1.22× lower latency than MPCFormer [18] on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, and increases accuracy by 0.79%, 0.79%, and 3.30% on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. Moreover, SAL-ViT presents lower PI latency than EA ViT [8] while increasing accuracy by more than 3% on all three datasets. As MPCViT [32] can configure the trade-off between PI latency and accuracy, we compare SAL-ViT to the MPCViT variants with the lowest latency and highest accuracy, respectively. For the lowest latency, SAL-ViT further lowers latency by 1.75×, 1.74×, and 1.03× while increasing accuracy by 2.33%, 1.46%, 1.49% on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. On the other hand, compared to the highest accuracy MPCViT model, SAL-ViT achieves an average increase of 0.87% in accuracy with 1.37× lower PI latency.

Figures 1 and 5 show that SAL-ViT provides a better accuracy-latency trade-off than the existing alternatives. Note that SAL-ViT, which is without KD, outperforms MPCViT and MPCFormer with KD.

4.2. Ablation Study on SAS

The effects of SA_{L2Q} location selection. Recall that the NAS phase in our proposed SAS determines which layers or heads are important and should be replaced with SA to achieve higher accuracy. This section illustrates the value of this phase. In particular, after our proposed SAS-L terminates with a specific α_l for each layer l, we select the B layers with the highest α_l. We compare this to an alternative algorithm that takes the B layers with the lowest α_l, referred to as SAS-L-WC. As detailed in Table 5, SAS-L outperforms SAS-L-WC for all tested SA_{L2Q} budgets on all datasets. This result shows that α_l of each layer indeed

Work	Attention	Softmax Approx.	CIFAR-10			CIFAR-100			Tiny-ImageNet		
			$\#SA_{L2Q}$[3] /$\#EA_{L2Q}$	PI Lat. (s)	Acc. (%)	$\#SA_{L2Q}$[3] /$\#EA_{L2Q}$	PI Lat. (s)	Acc. (%)	$\#SA_{L2Q}$[3] /$\#EA_{L2Q}$	PI Lat. (s)	Acc. (%)
Baseline	SA	`softmax`	7/0	61.20	95.56	7/0	61.34	77.36	9/0	75.12	61.60
MPCFormer [18] (w/o KD)	SA	`2Quad`	7/0	31.66	95.12	7/0	30.53	76.70	9/0	53.46	59.37
MPCFormer [18] (w/ KD)	SA	`2Quad`	7/0	31.66	95.13	7/0	30.53	77.07	9/0	53.46	60.84
EA [8]	EA	`softmax`	0/7	22.44	92.86	0/7	22.23	74.39	0/9	53.19	59.27
MPCViT [32] (w/o KD)	SA	Hybrid[1]	7/0	31.09 29.88 28.67 27.47	93.38 93.21 93.01 92.86	7/0	30.36 29.52 28.67 27.83	75.38 74.45 74.51 73.17	9/0	51.64 49.71 47.77 45.84	59.02 58.39 58.05 56.75
MPCViT [32] (w/ KD)	SA	Hybrid[1]	7/0	31.09 29.88 28.67 27.47	94.27 94.22 94.08 93.59	7/0	30.36 29.52 28.67 27.83	77.76 76.92 76.93 76.40	9/0	51.64 49.71 47.77 45.84	63.03 63.45 63.38 62.65
SAL-ViT (Ours)	Hybrid[2]	`L2Q`	3/4 2/5 1/6	20.56 18.41 **15.74**	**95.92** 95.69 95.14	3/4 2/5 1/6	20.72 18.42 **16.03**	77.62 **77.86** 76.12	3/6 2/7 1/8	45.38 44.53 **43.65**	**64.14** 63.18 62.53

1 A mix of `2ReLU` and `scaling`. 2 A mix of SA_{L2Q} and EA_{L2Q}.
3 The number of layers implemented by SA_{LAQ} / the number of layers implemented by EA_{L2Q}.

Table 4. Performance comparison on CIFAR-10, CIFAR-100, and Tiny-ImageNet

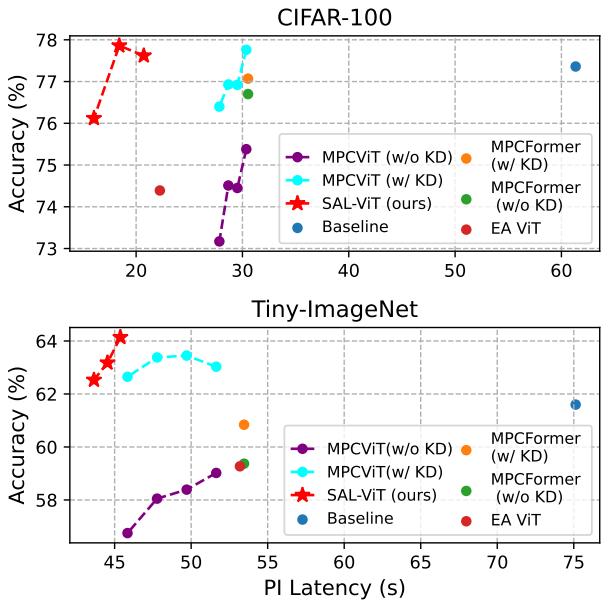

Figure 5. Performance comparison between prior works (Baseline, MPCFormer [18], EA ViT [8], MPCViT [32]) and ours (SAS-L + element-wise `L2Q`) on CIFAR-100 and Tiny-ImageNet.

reflects the value of replacing the encoding with SA_{L2Q}.

SAS-L vs. SAS-H. Table 5 also compares the performance of SAS-L and SAS-H. The results show that SAS-L presents higher accuracy than SAS-H despite SAS-H having a finer-grained search space. This is potentially because

Method	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	$\#SA_{L2Q}$[1] /$\#EA_{L2Q}$	Acc. (%)	$\#SA_{L2Q}$[1] /$\#EA_{L2Q}$	Acc. (%)	$\#SA_{L2Q}$[1] /$\#EA_{L2Q}$	Acc. (%)
SAS-L-WC	12/16	95.16	12/16	74.53	12/24	61.91
	8/20	94.47	8/20	75.12	8/28	60.21
	4/24	94.05	4/24	75.38	4/32	58.81
SAS-L	12/16	95.92	12/16	77.62	12/24	64.14
	8/20	95.69	8/20	77.86	8/28	63.18
	4/24	95.14	4/24	76.12	4/32	62.53
SAS-H	12/16	95.77	12/16	77.15	12/24	63.15
	8/20	95.57	8/20	76.72	8/28	62.33
	4/24	95.11	4/24	76.26	4/32	61.53

1 The number of heads implemented with SA_{L2Q} modules / the number of heads implemented with EA_{L2Q} modules.

Table 5. Performance comparison of SAS-L-WC, SAS-L, and SAS-H

the fine-grained search space makes training more difficult. A similar phenomenon was observed in [32] in which a finer-grained search space of softmax approximations led to lower accuracy. Although not shown in Table 5, it is important to note that the PI latency of SAS-L and SAS-H are equal because the latency is only a function of the number of SA_{L2Q}s.

Contribution of `L2Q` in SAS. In this experiment, we show that the softmax approximation impacts the performance of our SAS-L, and more specifically, that the proposed element-wise `L2Q` outperforms its counterparts. We com-

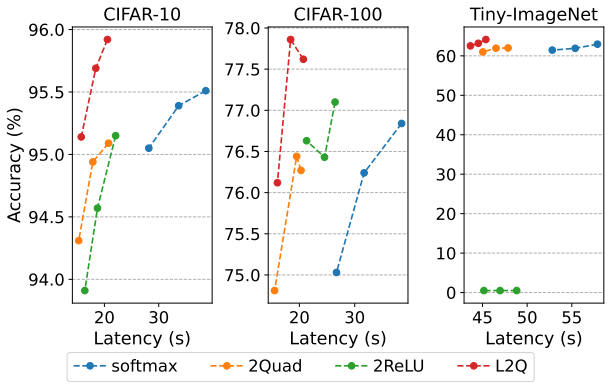

Figure 6. Performance comparison of SAS with `softmax`, `2Quad`, `2ReLU` and the proposed `L2Q`.

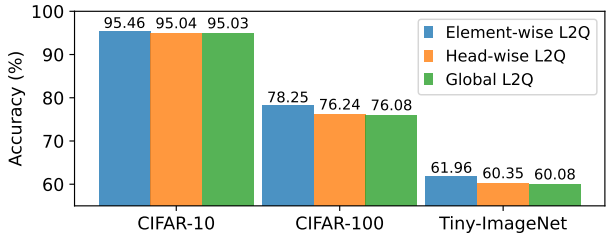

Figure 7. Accuracy comparison between `L2Q`s with different levels of granularity in normal ViT on CIFAR10, CIFAR-100, and TinyImagenet

pare the latency and accuracy of SAS-L variants that use `softmax`, `2Quad`, `2ReLU`, and our element-wise `L2Q` in both SA and EA in Figure 6, where our SAS-L with element-wise `L2Q` presents the highest average accuracy on all three datasets. While our element-wise `L2Q` shows obviously lower PI-latency than `softmax` and `2ReLU` , it presents competitive PI-latency with `2Quad`. This experiment also proves that `2ReLU` is not a stable re-weighted normalization, as mentioned in Section 3.2, because it collapses on Tiny-ImageNet.

4.3. Ablation Study on `L2Q`

In this section, we show the impact of our proposed `L2Q`. We embed various `L2Q` variants in the all-SA ViT and compare their accuracies.

Comparison of `L2Q` with different granularities. We present and compare the results of three `L2Q` granularities, i.e., element-wise, head-wise, and global, in Figure 7. The results show that as the granularity becomes finer-grained, the corresponding accuracy increases in all datasets, which suggests that finer-grained learnability yields better optimized attention re-weighting, thus improving accuracy.

Effect of shifting and scaling parameters in `L2Q`. To understand the importance of shifting and scaling parameters, we conducted three different experiments on element-wise `L2Q`. The first one uses both learnable shifting parameters

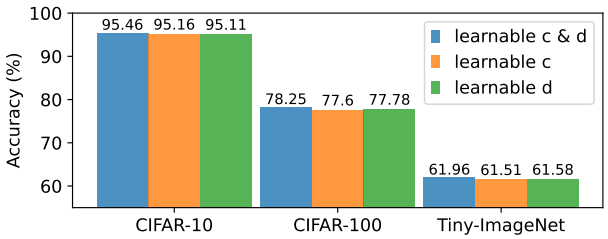

Figure 8. Accuracy of `L2Q` with various learnable parameters in normal ViT on CIFAR10, CIFAR-100, and TinyImagenet.

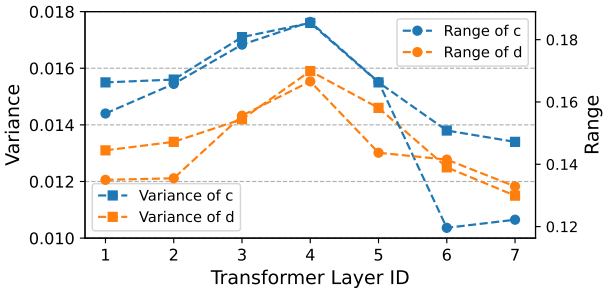

Figure 9. Variance and the range of learned shifting and scaling parameters on CIFAR-100

in $\boldsymbol{C}$ and scaling parameters in $\boldsymbol{D}$, the second one only uses the learnable shifting parameters in $\boldsymbol{C}$, and the third experiment only uses the learnable scaling parameters in $\boldsymbol{D}$. As shown in Figure 8, the model with both shifting and scaling parameters provides better accuracies on all three datasets than the models with only one set of learnable parameters, demonstrating the importance of using both learnable parameters.

Analysis of learned parameters. We present the variance and range of the learnable parameters for each layer in Figure 9. Notice that the range of the learned elements in $\boldsymbol{C}$ and $\boldsymbol{D}$ in all layers is significant, suggesting the benefit of learning specific attention scaling and shifting values in *L2Q*.

5. Summary and Conclusions

In this work, we present SAL-ViT, which leverages a novel softmax approximation and selective attention search to boost the efficiency of private inference on ViTs. Our extensive experiments show that the proposed SAL-ViT can effectively reduce PI latency and improve image classification accuracy. To the best of our knowledge, SAL-ViT sets a new state of the art in PI on ViT. Note that research on improving MPC protocols for ViT, e.g., Iron [9], is orthogonal to our work, and can be applied on top of SAL-ViT. Our future work includes applying knowledge distillation to SAL-ViT to achieve even higher accuracy, and extending SAL-ViT to more applications, such as object detection and semantic segmentation.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[2] Minsu Cho, Ameya Joshi, Brandon Reagen, Siddharth Garg, and Chinmay Hegde. Selective network linearization for efficient private inference. In *International Conference on Machine Learning*, pages 3947–3961. PMLR, 2022. 1, 2

[3] Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. Faster CryptoNets: Leveraging sparsity for real-world encrypted inference. *arXiv preprint arXiv:1811.09953*, 2018. 2

[4] MA Crisfield. A faster modified Newton-Raphson iteration. *Computer methods in applied mechanics and engineering*, 20(3):267–278, 1979. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 4

[6] Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? *International Conference on Learning Representations*, 2021. 2, 3, 4

[7] Zahra Ghodsi, Nandan Kumar Jha, Brandon Reagen, and Siddharth Garg. Circa: Stochastic ReLUs for private deep learning. *Advances in Neural Information Processing Systems*, 34:2241–2252, 2021. 2

[8] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 3, 4, 6, 7

[9] Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In *Advances in Neural Information Processing Systems*, 2022. 2, 8

[10] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. 2021. 1, 3, 6

[11] Zhicong Huang, Wen jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. Cryptology ePrint Archive, Paper 2022/207, 2022. 1, 2, 6

[12] Nandan Kumar Jha, Zahra Ghodsi, Siddharth Garg, and Brandon Reagen. DeepReDuce: ReLU reduction for fast private inference. In *International Conference on Machine Learning*, pages 4839–4849. PMLR, 2021. 2

[13] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669, 2018. 1, 2

[14] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 4

[15] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CRYPTEN: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:4961–4973, 2021. 3, 4, 6

[16] Souvik Kundu, Shunlin Lu, Yuke Zhang, Jacqueline Liu, and Peter A Beerel. Learning to linearize deep neural networks for secure and efficient private inference. *International Conference on Learning Representation*, 2023. 1, 2

[17] Souvik Kundu, Qirui Sun, Yao Fu, Massoud Pedram, and Peter A Beerel. Analyzing the confidentiality of undistillable teachers in knowledge distillation. *Advances in Neural Information Processing Systems*, 34:9181–9192, 2021. 1

[18] Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Hao Zhang. MPCFormer: fast, performant and private Transformer inference with MPC, 2022. 1, 2, 3, 6, 7

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[20] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021. 2, 3, 4

[21] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that cannot teach students. *arXiv preprint arXiv:2105.07381*, 2021. 1

[22] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. DELPHI: A cryptographic inference service for neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, Aug. 2020. 1, 2, 6

[23] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017. 1, 2, 3

[24] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosFormer: Rethinking softmax in attention. In *International Conference on Learning Representations*, 2022. 2, 3, 4

[25] M Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. XONN: XNOR-based oblivious deep neural network inference. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1501–1518, 2019. 2

[26] Liyan Shen, Ye Dong, Binxing Fang, Jinqiao Shi, Xuebin Wang, Shengli Pan, and Ruisheng Shi. ABNN2: secure two-party arbitrary-bitwidth quantized neural network predictions. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 361–366, 2022. 1, 2

[27] Sijun Tan, Brian Knott, Yuan Tian, and David J Wu. CryptGPU: Fast privacy-preserving machine learning on the GPU.

In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1021–1038. IEEE, 2021. 1, 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 4

[29] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2, 3, 4

[30] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2, 3

[31] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 1

[32] Wenxuan Zeng, Meng Li, Wenjie Xiong, Wenjie Lu, Jin Tan, Runsheng Wang, and Ru Huang. MPCViT: Searching for MPC-friendly vision transformer with heterogeneous attention. *arXiv preprint arXiv:2211.13955*, 2022. 1, 2, 3, 6, 7

[33] Yuke Zhang, Dake Chen, Souvik Kundu, Haomei Liu, Ruiheng Peng, and Peter A. Beerel. C2PI: An efficient crypto-clear two-party neural network private inference. In *Proceedings of the 60th ACM/IEEE Design Automation Conference*, 2023. 2

[34] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 1