Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2026.073657

ARTICLE

IG-3D: Integrated-Gradients 3D Optimization for Private Transformer Inference

Lei Sun[1,2], Jingwen Wang[2,*], Peng Hu[2], Xiuqing Mao[1,2], Cuiyun Hu[1,2] and Zhihong Wang[2]

[1]Henan Key Laboratory of Information Security, Zhengzhou, 450004, China

[2]Cryptographic Engineering School, Information Engineering University, Zhengzhou, 450004, China

*Corresponding Author: Jingwen Wang. Email: wang_jw2023@163.com

Received: 23 September 2025; Accepted: 22 December 2025; Published: 12 March 2026

ABSTRACT: Transformer models face significant computational challenges in private inference (PI). Existing optimization methods often rely on isolated techniques, neglecting joint structural and operational improvements. We propose IG-3D, a unified framework that integrates structured compression and operator approximation through accurate importance assessment. Our approach first evaluates attention head importance using Integrated Gradients (IG), offering greater stability and theoretical soundness than gradient-based methods. We then apply a three-dimensional optimization: (1) structurally pruning redundant attention heads; (2) replacing Softmax with adaptive polynomial approximation to avoid exponential computations; (3) implementing layer-wise GELU substitution to accommodate different layer characteristics. A joint threshold mechanism coordinates compression across dimensions under accuracy constraints. Experimental results on the GLUE benchmark show that our method achieves an average 2.9× speedup in inference latency and a 50% reduction in communication cost, while controlling the accuracy loss within 2.3%, demonstrating significant synergistic effects and a superior accuracy-efficiency trade-off compared to single-technique optimization strategies.

KEYWORDS: Private inference; transformer; attention-head pruning; integrated gradients; transformer model optimization

1 Introduction

In recent years, large-scale Transformer-based models have achieved breakthrough results in fields such as natural language processing (NLP) [1] and computer vision (CV) [2]. From BERT to the GPT series, these models exhibit strong performance across diverse tasks. The growing use of cloud services has made outsourcing inference increasingly common, improving efficiency but also introducing privacy risks: sensitive data may be exposed, and model parameters are vulnerable to theft [3].

Private inference using fully homomorphic encryption (FHE) [4] or secure multi-party computation (MPC) [5] provides formal security guarantees, but incurs substantial computational and communication overhead—especially for non-linear operators like Softmax and GELU, which are orders of magnitude more expensive in encrypted form. Existing PI solutions can increase latency by over 60× or significantly reduce inference quality [6].

To mitigate this, research has pursued two paths: protocol-level optimizations (e.g., improved ciphertext packing and matrix multiplication in THOR [7] and SecFormer [8]) that mainly benefit linear layers; and model-level strategies such as operator substitution (e.g., polynomial approximations in THE-X [9] and MPCFormer [10]) or attention-head pruning [11]. However, these often rely on heuristics, ignore component-wise importance, or introduce accuracy loss due to uniform approximation.

We propose a unified framework based on Integrated Gradients to address these limitations. Our approach offers theoretical grounding and numerical stability in attributing importance to attention heads. We introduce a coordinated three-dimensional strategy—pruning redundant heads (structure), approximating Softmax with polynomials (operator), and replacing GELU with ReLU per layer (activation)—all governed by a joint threshold mechanism to control accuracy loss. On the GLUE benchmark, IG-3D achieves an average 2.9× speedup in inference latency while keeping the accuracy loss under 2.3%, outperforming current baselines. The contributions of this paper are summarized as follows:

- An Integrated Gradients–based attention-head importance evaluation method with stable and theoretically grounded attribution.
- A unified three-dimensional collaborative optimization framework, integrating pruning, approximation, and replacement into a holistic lightweighting strategy.
- Extensive experiments validating superior efficiency–accuracy trade-offs compared to existing methods.

The remainder of this paper is organized as follows: Section 2 reviews the relevant background and related work. Section 3 details the Integrated Gradients-based importance evaluation method and elaborates on the unified three-dimensional optimization strategy. Section 4 presents the experimental results and analysis. Finally, Section 5 concludes the paper and outlines future work.

2 Background and Related Work

2.1 Transformer Architecture and Computational Bottlenecks

The Transformer architecture, powered by its self-attention mechanism, has fundamentally reshaped the research paradigm in NLP [12]. Encoder models such as BERT [13] are composed of multiple stacked layers, each containing two primary sublayers: multi-head self-attention (MHSA) and a feed-forward network (FFN). The MHSA mechanism generates query (Q), key (K), and value (V) vectors via linear projections, and its central operation is the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{1}$$

where the Softmax function constitutes the first nonlinear computational bottleneck.

The FFN typically adopts the Gaussian Error Linear Unit (GELU) as its activation function. Its approximate form involves complex high-order polynomials and special functions, forming the second major nonlinear bottleneck:

$$\text{GELU}(x) = 0.5x\left[1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.047715x^3\right)\right)\right] \tag{2}$$

In plaintext environments, these operations are computationally efficient. However, in privacy-preserving contexts such as FHE or MPC, nonlinear functions incur extremely high costs. The evaluation of exponentials, reciprocals, and high-degree polynomials requires sophisticated cryptographic protocols and a large number of communication rounds, making them the dominant performance bottlenecks in the entire inference pipeline [6,13].

2.2 Optimization Techniques in Private Inference Protocols

To mitigate the aforementioned overheads, researchers have extensively explored optimizations at the cryptographic protocol level. These works primarily focus on executing linear and nonlinear computations more efficiently under ciphertext or secret-sharing settings. In the fully homomorphic encryption (FHE) domain, the main research emphasis lies in ciphertext packing and linear algebra computation. Iron [14] introduces an innovative packing strategy that encodes multiple matrix rows into a single ciphertext, significantly reducing the computational and communication overhead of linear layers. THOR [7] leverages vectorized matrix multiplication and batch processing to greatly reduce the number of key-switching operations. Primer [15] proposes a "tokens-first" packing scheme, minimizing homomorphic rotation operations and further optimizing Transformer inference latency. The BOLT framework [16] enables matrix multiplication within Transformers using the leveled FHE scheme BFV, employing column packing for ciphertext matrices and diagonal packing for plaintext matrices. These techniques substantially improve the efficiency of linear operations but provide only limited benefits for nonlinear computations. In the secure multi-party computation (MPC) domain, research has focused on designing specialized protocols that reduce interaction rounds and communication volume. SecureTLM [17] develops hybrid secret-sharing protocols for Softmax, GELU, and LayerNorm, improving efficiency of nonlinear functions. SecFormer [8] strategically decomposes computations across parties to reduce the communication cost associated with matrix factorization and recomposition. BumbleBee [18] introduces oblivious linear transformation (OLT) and dynamic compression strategies to effectively balance computation and communication. Other hybrid approaches combine CKKS-based homomorphic encryption with MPC [19] to break conventional hierarchical limitations and provide more efficient frameworks for private Transformer inference. Despite these advances, protocol-level optimizations remain fundamentally bound by cryptographic frameworks. They do not alter the inherent inefficiency of the model computation graph itself, leaving the high cost of nonlinear operations as the central unresolved challenge.

2.3 Model Optimization Techniques for Plaintext and Ciphertext

Another line of research focuses on directly optimizing the model to produce a "cryptography-friendly" architecture that reduces computational costs at the source. These approaches can be broadly classified into general acceleration methods for plaintext inference and specialized optimizations for ciphertext settings.

For plaintext inference, model compression techniques offer valuable insights. Attention-head pruning, for instance, has been shown to effectively reduce Transformer complexity. Michel et al. [11] demonstrated that many attention heads are redundant and can be removed without significant performance loss. Voita et al. [20] introduced a gradient-based importance scoring mechanism for head pruning. Nevertheless, while pruning reduces attention load, it does not alleviate computational costs in feed-forward networks [21]. Recent methods like automatic channel pruning aim to preserve more informative heads by merging similar channels [22]. Nonlinear operator approximation is another common strategy, particularly in edge computing, where GELU is replaced with ReLU or piecewise linear functions, and Softmax is approximated via polynomials [23,24]. However, these techniques often rely on heuristic importance measures lacking theoretical rigor, and are designed for plaintext settings without considering encrypted computation.

For ciphertext inference, methods such as THE-X [9] replace all non-polynomial functions with low-degree polynomials to maintain FHE compatibility. Zhang et al. [25] introduce CipherPrune, a method that integrates encrypted token pruning with low-degree polynomial approximations, designed specifically for private Transformer inference; this approach improves both computational efficiency and reduces communication costs while preserving model privacy. Similarly, MPCFormer [10] designs quadratic approximations tailored for MPC. Ghazvinian et al. [26] develop MOFHEI, a model optimization framework tailored for homomorphic encryption, which significantly reduces computational overhead in private inference by pruning blocks and optimizing model structures for HE compatibility. Despite their utility, most existing ciphertext-inference methods still exhibit two major limitations: (1) they often rely on uniform approximation strategies that ignore component-specific importance, which can lead to accuracy loss; and (2) they are not fully integrated with fine-grained structural compression techniques (e.g., attention-head or FFN pruning), leaving room for more holistic optimization of the entire Transformer computation graph.

3 Method

In this section, we first specify the private inference threat model and security guarantees under which IG-3D is designed, and then present the method details: an Integrated Gradients–based attention-head importance evaluation scheme and a three-dimensional collaborative optimization framework.

3.1 Threat Model and Security Guarantees

We consider a standard private inference setting in which a model owner and a data owner rely on cryptographic protocols such as fully homomorphic encryption or secure multi-party computation to protect model parameters and user inputs [4,5]. The goal is to obtain model predictions on private data without revealing the plaintext model or inputs to any single party.

3.1.1 Adversary Model

Unless otherwise stated, we assume semi-honest (honest-but-curious) adversaries: corrupted parties follow the prescribed protocol but may try to infer additional information from their local view, following common assumptions in recent PI systems for Transformer models [7–10,14–19]. In MPC-style deployments, we assume that a small subset of computing servers can be corrupted but do not collude (e.g., one corrupted server in typical three-party protocols), as in prior work on secure Transformer inference with MPC [8,10,17,19]. In FHE-style deployments, the evaluator does not possess the decryption key, which is held by the client or a separate key holder, as in FHE-based PI systems [4,7,9,15]. Network-level side channels (e.g., traffic analysis or denial-of-service) and micro-architectural attacks (e.g., cache timing, speculative execution) are out of scope and are assumed to be mitigated by standard system-level defenses, consistent with the threat models of prior PI frameworks [7–10,14,16–18].

3.1.2 Security Guarantees of the Underlying PI Framework

We target state-of-the-art PI frameworks for Transformer models such as THE-X [9], MPCFormer [10], IRON [14], Primer [15], BumbleBee [18], Bolt [16], SecureTLM [17], SecFormer [8], THOR [7], and the hybrid CKKS+MPC design of Xu et al. [19]. These systems combine linear layers over encrypted or secret-shared values with secure evaluation of non-linear operations, and provide provable security against semi-honest adversaries at a prescribed security level (typically 128-bit), in the standard simulation-based sense: intuitively, the adversary's view (ciphertexts, secret shares, and protocol transcripts) can be simulated given only the prescribed leakage (e.g., tensor dimensions and the public model architecture) and does not reveal additional information about private inputs or model parameters [4,5].

In our experiments, we instantiate this abstract PI framework using CrypTen [27], a PyTorch-based secure multi-party computation library that implements two-party additive secret sharing. CrypTen follows the above semi-honest adversary model and provides 128-bit security by operating over a 64-bit ring with standard cryptographic primitives. All tensor operations in PI are executed as CrypTen secure protocols between two non-colluding parties connected over a reliable network channel. The model owner and data owner each hold additive shares of the inputs and model parameters; no single party ever observes the plaintext values during the protocol. This instantiation is representative of MPC-style PI backends and fits within the generic framework described above.

3.1.3 Effect of IG-3D on Security

Our proposed IG-3D framework (Fig. 1) is a *model-level* optimization that rewires the computation graph of the Transformer but does not change the cryptographic backend, the number or roles of parties, or the key management procedures. Specifically, attention-head pruning modifies only which heads are evaluated, similar in spirit to structural optimizations explored in prior PI systems; PolySoftmax replaces Softmax with polynomial approximations that are executed by the same secure arithmetic circuits as other polynomial operations; and GELU→ReLU or GELU→mixing changes the non-linear activation function inside existing secure activation routines, analogous to activation approximations used in FHE/MPC-based neural inference [6,9,14,15,28]. As a result, IG-3D does not expand the information available to any party beyond what is already revealed by the baseline PI protocol. The access patterns (e.g., sequence length, number of layers) and communication structure remain within the leakage profile of the underlying framework [7–10,14,16,17,19]. In particular, in our CrypTen-based instantiation IG-3D only modifies the plaintext computation graph that is compiled into secure protocols and leaves the CrypTen backend, security level, and threat model unchanged. Consequently, IG-3D preserves the protocol-level security guarantees of the chosen FHE/MPC backend; our contributions focus on reducing arithmetic complexity and communication overhead under these existing guarantees rather than redefining the security model.

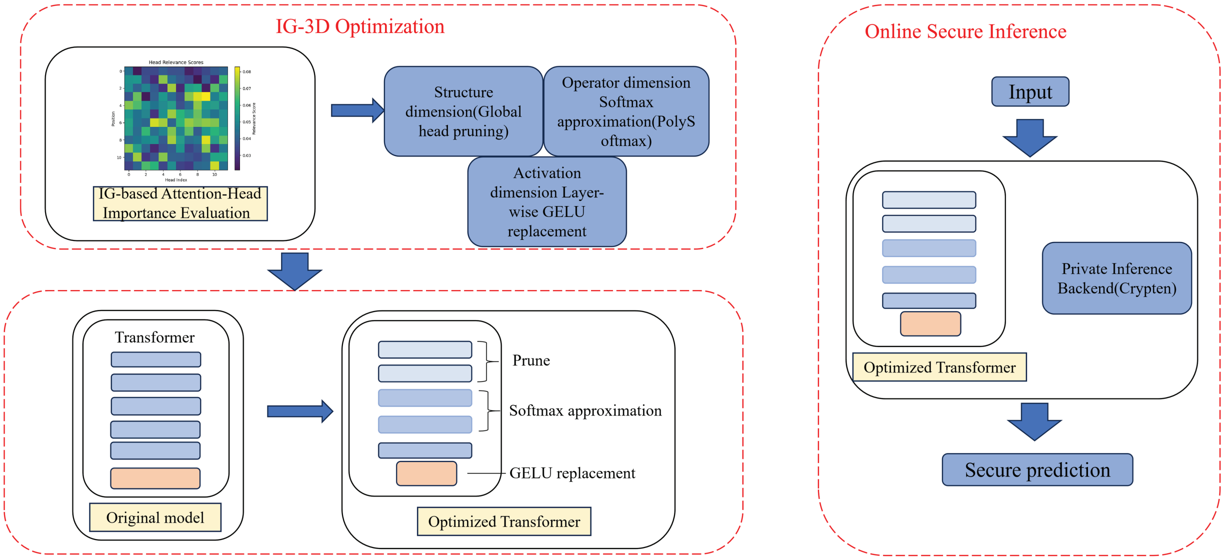


Figure 1: Overall workflow of IG-3D optimization and online secure inference. The left dashed box (*IG-3D Optimization*) depicts the offline pipeline: IG-based attention-head importance evaluation followed by three-dimensional optimizations in the structure (global head pruning), operator (Softmax approximation via PolySoftmax), and activation (layer-wise GELU replacement) dimensions, yielding an optimized Transformer. The right dashed box (*Online Secure Inference*) shows how the optimized Transformer is deployed inside a generic MPC/FHE-based private inference backend (instantiated as CrypTen in our experiments) to process encrypted or secret-shared inputs and produce secure predictions

3.2 Integrated Gradients-Based Attention Head Importance Evaluation

3.2.1 Problem Analysis and Methodological Motivation

In modern Transformer models, attention heads often contain substantial redundancy. While multi-head attention captures diverse features, many heads contribute minimally to final outputs. To improve efficiency, particularly under resource constraints, identifying and removing redundant attention heads requires precise evaluation mechanisms.

Existing methods for evaluating attention-head importance can be grouped into three main categories, each with notable drawbacks.

Weight-norm methods assess importance by computing norms of attention weight matrices [29]:

$$\text{Importance}_{l,h}^{\text{norm}} = \|\boldsymbol{W}_{l,h}^{Q}\| \cdot \|\boldsymbol{W}_{l,h}^{K}\| \cdot \|\boldsymbol{W}_{l,h}^{V}\| \tag{3}$$

where $\boldsymbol{W}_{l,h}^{Q} \in \mathbb{R}^{d \times d_k}$, $\boldsymbol{W}_{l,h}^{K} \in \mathbb{R}^{d \times d_k}$, and $\boldsymbol{W}_{l,h}^{V} \in \mathbb{R}^{d \times d_v}$ are query, key, and value projection matrices. These methods ignore input dependence and contextual information—identical weights may produce different effects under varying inputs. As Kobayashi et al. [30] note, analyses considering only attention weights without vector norms are biased.

Gradient-sensitivity methods use first-order gradients of loss with respect to attention outputs [31]:

$$\text{Importance}_{l,h}^{\text{grad}} = \left\| \frac{\partial \mathcal{L}}{\partial \boldsymbol{A}_{l,h}} \right\| \tag{4}$$

where $\boldsymbol{A}_{l,h} \in \mathbb{R}^{N \times N}$ is the attention matrix. Limitations include gradient saturation in activation functions and local approximation that lacks global consistency.

Attention-weight analysis uses attention score entropy as importance indicator [32]:

$$\text{Importance}_{l,h}^{\text{attn}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \text{Entropy}(\boldsymbol{A}_{l,h}^{(i)}) \tag{5}$$

This method ignores the complex nonlinear mapping between attention weights and final predictions.

3.2.2 Attribution Computation for Attention Heads via Integrated Gradients

Integrated Gradients [33] is an attribution method that addresses limitations of traditional gradient-based techniques. Let the Transformer model be $f : \mathbb{R}^{N \times d} \to \mathbb{R}^{C}$, where C is the number of output classes. For input $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, the output is $\boldsymbol{y} = f(\boldsymbol{X}) \in \mathbb{R}^{C}$. For the (l, h)-th head, scaled dot-product attention is given by:

$$\boldsymbol{A}_{l,h} = \text{Softmax}\left(\frac{\boldsymbol{Q}_{l,h} \boldsymbol{K}_{l,h}^{\top}}{\sqrt{d_k}} \right) \tag{6}$$

$$\boldsymbol{O}_{l,h} = \boldsymbol{A}_{l,h} \boldsymbol{V}_{l,h} \tag{7}$$

where $\boldsymbol{Q}_{l,h} = \boldsymbol{X}_l \boldsymbol{W}_{l,h}^{Q} \in \mathbb{R}^{N \times d_k}$, $\boldsymbol{K}_{l,h} = \boldsymbol{X}_l \boldsymbol{W}_{l,h}^{K} \in \mathbb{R}^{N \times d_k}$ and $\boldsymbol{V}_{l,h} = \boldsymbol{X}_l \boldsymbol{W}_{l,h}^{V} \in \mathbb{R}^{N \times d_v}$ denote the query, key and value matrices of the (l, h)-th attention head, respectively. $\boldsymbol{X}_l \in \mathbb{R}^{N \times d}$ is the input representation to layer l, and $\boldsymbol{O}_{l,h} \in \mathbb{R}^{N \times d_v}$ is the corresponding head output. Our objective is to quantify the contribution of $\boldsymbol{O}_{l,h}$ to the final model output $\boldsymbol{y}$.

We adopt a zero-attention baseline $\boldsymbol{O}_{l,h}^{\text{baseline}} = \boldsymbol{0}_{N\times d_v}$, which corresponds to complete removal of the head. This choice is well justified: a zero baseline naturally represents the absence of the head and thus aligns with the practical pruning scenario, avoids biases introduced by arbitrary baseline selections, and is computationally simple to evaluate at scale. The straight-line interpolation path from the baseline to the actual head output is defined as

$$\boldsymbol{O}_{l,h}^{(\alpha)} = \boldsymbol{O}_{l,h}^{\text{baseline}} + \alpha\left(\boldsymbol{O}_{l,h} - \boldsymbol{O}_{l,h}^{\text{baseline}}\right) = \alpha\, \boldsymbol{O}_{l,h}, \qquad \alpha \in [0,1] \tag{8}$$

Consequently, $\boldsymbol{O}_{l,h}^{(0)} = \boldsymbol{0}$ corresponds to the head being fully removed, whereas $\boldsymbol{O}_{l,h}^{(1)} = \boldsymbol{O}_{l,h}$ recovers the original head output.

Let $f_{l,h}(\boldsymbol{O}_{l,h})$ denote the model output as a function of the (l,h)-th head output with other heads fixed. For classification tasks we focus on the target-class logit $f_{l,h}^{(c)}(\boldsymbol{O}_{l,h})$. The Integrated Gradients attribution for this head is

$$\boldsymbol{IG}_{l,h} = (\boldsymbol{O}_{l,h} - \boldsymbol{O}_{l,h}^{\text{baseline}}) \odot \int_0^1 \frac{\partial f_{l,h}^{(c)}(\boldsymbol{O}_{l,h}^{(\alpha)})}{\partial \boldsymbol{O}_{l,h}^{(\alpha)}}\, d\alpha \tag{9}$$

where $\odot$ denotes element-wise multiplication. Numerically, we approximate the integral using a Riemann sum with m steps (in practice we use $m = 16$):

$$\boldsymbol{IG}_{l,h} \approx \frac{\boldsymbol{O}_{l,h}}{m} \sum_{k=1}^{m} \frac{\partial f_{l,h}^{(c)}\left(\boldsymbol{O}_{l,h}^{(k/m)}\right)}{\partial \boldsymbol{O}_{l,h}^{(k/m)}} \tag{10}$$

We aggregate attributions into a scalar score using the Frobenius norm:

$$S_{l,h}^{\text{raw}} = \|\boldsymbol{IG}_{l,h}\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{d_v} (\boldsymbol{IG}_{l,h})_{i,j}^2} \tag{11}$$

For an evaluation dataset $\mathcal{D} = \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(|\mathcal{D}|)}\}$, the final importance score is:

$$\bar{S}_{l,h} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} S_{l,h}^{(i)} \tag{12}$$

This produces a global ranking across all attention heads, directly quantifying each head's contribution to predictions. Compared with norm- or gradient-based heuristics, IG attribution more accurately captures true impact, enabling more reliable pruning and optimization decisions.

3.2.3 Computational Cost and Stability of Integrated Gradients

Although vanilla Integrated Gradients (IG) requires m integration steps, IG-3D uses IG only once as an *offline structure-search* stage before private inference. To quantify this cost and evaluate reduced-step approximations, we measured actual runtimes on an NVIDIA V100 using 50 randomly sampled MNLI development examples. Table 1 reports the measured wall-clock cost and the Spearman correlation with the full m=50 attribution.

Table 1: Integrated Gradients (IG) offline cost and approximation accuracy. Spearman correlation is computed against the full $m = 50$ IG attribution

Method	Steps (m)	GPU cost (V100)	Spearman vs. m=50
Full IG (reference)	50	1.76 GPU-hours	1.00
Ours (default)	**16**	**0.83 GPU-hours**	**0.95**
Path-sampling IG	8	0.55 GPU-hours	0.92
Stochastic IG	16	0.35 GPU-hours	0.90

Note: Bold indicates the results of our method.

These findings show that IG-3D remains stable even under significantly reduced integration steps. All experiments in this paper adopt the m=16 configuration, which achieves reliable importance rankings while keeping the offline attribution cost modest. Since IG is never executed during training or during private inference, this preprocessing stage does not affect any communication, latency, or cryptographic overhead reported in the private-inference results.

3.3 Three-Dimensional Unified Collaborative Optimization

3.3.1 Overall Framework of the 3D Optimization Strategy

Our optimization strategy applies differentiated optimizations based on the heterogeneous importance and optimization potential of model components, achieving precise cost reduction with minimal accuracy loss through three dimensions. In the structure dimension, we identify and prune redundant attention heads using Integrated Gradients importance scores to reduce parameters and computational cost. In the operator dimension, we replace Softmax with adaptive low-order polynomial approximations to reduce exponentiation costs in cryptographic settings. In the activation dimension, we design layer-wise GELU replacement that selects appropriate activation functions for different layer roles, trading activation complexity for efficiency while preserving necessary expressive capacity. Formally, given the original Transformer $F(\boldsymbol{x};\Theta)$ with input sequence $\boldsymbol{x}$ and parameter set Θ, the optimized model is expressed as a composition of three optimization modules:

$$F'(\boldsymbol{x};\Theta') = F_{\mathrm{act}}\Big(F_{\mathrm{op}}\Big(F_{\mathrm{struct}}(\boldsymbol{x};\Theta_{\mathrm{struct}});\Theta_{\mathrm{op}}\Big);\Theta_{\mathrm{act}}\Big) \tag{13}$$

where F_{struct} denotes the structure-pruned model with $\Theta_{\mathrm{struct}} \subset \Theta$, F_{op} denotes the operator-approximation module parameterized by Θ_{op}, and F_{act} denotes the activation-replacement module with parameters Θ_{act}. A unified threshold coordinator selects thresholds for the three modules to satisfy a global accuracy constraint while maximizing inference speedup and reducing communication.

3.3.2 Structure Dimension: Importance-Score Based Attention-Head Pruning

Using the attention-head importance scores $S_{l,h}$ derived in Section 3, we design a global threshold pruning policy that employs a cross-layer global ranking to avoid uneven pruning caused by inter-layer importance disparities. First, we sort all attention heads globally by their importance scores:

$$\{S_{(1)}, S_{(2)}, \ldots, S_{(H)}\} \quad \text{where} \quad S_{(1)} \geq S_{(2)} \geq \cdots \geq S_{(H)} \tag{14}$$

where $H = L \times N_h$ is the total number of attention heads, L denotes the number of layers and N_h denotes the number of heads per layer. We define a global pruning threshold τ_s and make pruning decisions via the indicator function

$$\mathrm{Keep}(l,h) = \begin{cases} 1, & \text{if } S_{l,h} \geq \tau_{\mathrm{s}} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $\mathrm{Keep}(l,h) = 1$ indicates the (l,h)-th head is retained. To preserve basic model functionality, we enforce a minimum retention constraint:

$$\sum_{l=1}^{L}\sum_{h=1}^{N_h}\mathrm{Keep}(l,h) \;\geq\; \alpha \cdot H \tag{16}$$

where $\alpha \in [0.1, 0.5]$ is the minimum retention ratio that prevents excessive pruning.

3.3.3 Operator Dimension: Differentiated Softmax Approximation

The dominant cost of the Softmax operator in private inference stems from the exponential function, which is particularly expensive in FHE and MPC settings. We propose an importance-aware, differentiated approximation strategy: keep exact Softmax for important heads while applying polynomial approximations for less important heads.

Using a truncated Taylor expansion, we construct polynomial approximations to the exponential. Given an input vector $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]$, we first apply a numerically stable shift:

$$\widetilde{\boldsymbol{z}} = \boldsymbol{z} - \max(\boldsymbol{z}) \tag{17}$$

and then approximate the exponential by its truncated Taylor series

$$\exp(x) \approx \sum_{k=0}^{m}\frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^m}{m!} \tag{18}$$

For computational efficiency we adopt a quadratic approximation ($m = 2$):

$$\exp(x) \approx 1 + x + \frac{x^2}{2} \tag{19}$$

Guided by attention-head importance scores, we define a differentiated policy:

$$\mathrm{SoftmaxApprox}_{l,h}(\boldsymbol{z}) = \begin{cases} \mathrm{Softmax}(\boldsymbol{z}), & \text{if } S_{l,h} \geq \tau_a \\ \mathrm{PolySoftmax}(\boldsymbol{z}), & \text{otherwise} \end{cases} \tag{20}$$

where τ_{soft} is the Softmax-approximation threshold and PolySoftmax denotes the polynomial approximation:

$$\mathrm{PolySoftmax}(\boldsymbol{z})_i = \frac{1 + \tilde{z}_i + \frac{1}{2}\tilde{z}_i^2}{\sum_{j=1}^{n}\left(1 + \tilde{z}_j + \frac{1}{2}\tilde{z}_j^2\right)} \tag{21}$$

To justify the choice of using a quadratic expansion in Eq. (19), we analyze how the polynomial degree affects both the approximation of the exponential function and the resulting softmax probabilities. Let the clipped logit range be $|z| \leq B$, with the calibrated value on MNLI being $B \approx 1.55$. Over this compact interval, we approximate e^z using a degree-m least-squares polynomial $p_m(z)$ and characterize the approximation error through $e^z = p_m(z) + r_m(z), |r_m(z)| \leq \|r_m\|_\infty$.

Standard perturbation analysis of the softmax operator shows that probability deviations scale proportionally to $|r_m|_\infty$, and the argmax remains unchanged as long as class margins exceed this perturbation. Thus, the magnitude of the polynomial approximation error directly governs both numerical stability and prediction consistency.

We empirically validate this analysis using BERT-base fine-tuned on MNLI. For polynomial degrees $m \in 2, 3, 4$, we fit least-squares approximations on $[-B, B]$ and measure both the exponential approximation error on a dense grid and the resulting softmax deviation on collected logits. As shown in Table 2, higher-order polynomials naturally reduce approximation error. However, all three polynomial degrees yield identical predictions on the MNLI development set, demonstrating that the quadratic expansion is already sufficiently accurate for classification.

Table 2: Effect of polynomial degree on the approximation of e^z over $[-B, B]$ ($B \approx 1.55$) and the resulting softmax probabilities on MNLI dev

$\boldsymbol{m}$	**Exponential approximation error**		**Softmax error**	
	max $\lvert e^z - p_m(z)\rvert$	**mean**	**max** $\lvert \sigma - \tilde{\sigma}_m \rvert$	**mean**
2	3.56×10^{-1}	9.26×10^{-2}	5.39×10^{-2}	2.73×10^{-2}
3	7.32×10^{-2}	1.76×10^{-2}	1.15×10^{-2}	3.53×10^{-3}
4	1.20×10^{-2}	2.70×10^{-3}	4.57×10^{-3}	7.90×10^{-4}
Prediction change	None for all $m \in \{2, 3, 4\}$			

In addition to its numerical stability, the quadratic approximation also offers a practical advantage: its evaluation requires only a small number of fused multiply-add operations, making it substantially cheaper than cubic or quartic polynomials on modern hardware. Since $m = 2$ provides stable probabilities, preserves all predictions, and minimizes computational overhead, it represents the best trade-off between accuracy and efficiency.

3.3.4 Activation Dimension: Layer-Wise GELU Replacement Strategy

Considering the high polynomial cost of GELU in encrypted inference and the functional heterogeneity across network layers, we propose a layer-wise progressive replacement mechanism:

$$\mathrm{Act}_l(x) = \begin{cases} \mathrm{ReLU}(x), & \text{if } l \le L/3 \\ \lambda_l\,\mathrm{GELU}(x) + (1 - \lambda_l)\,\mathrm{ReLU}(x), & \text{if } L/3 < l \le 2L/3 \\ \mathrm{GELU}(x), & \text{if } l > 2L/3 \end{cases} \tag{22}$$

where the mixing coefficient increases with layer depth in the middle region:

$$\lambda_l = \frac{l - L/3}{L/3}, \qquad l \in (L/3,\, 2L/3] \tag{23}$$

This design is motivated by observations on layer-wise functionality: shallow layers ($l \le L/3$) primarily perform feature extraction and can benefit from the sparsity and simplicity of ReLU; intermediate layers ($L/3 < l \le 2L/3$) are in a feature-transformation stage and therefore require a balance between computational efficiency and expressive power, achieved by a gradual interpolation between ReLU and GELU; deep layers ($l > 2L/3$) focus on abstract representation learning and thus retain GELU's smoothness to preserve representation quality.

3.3.5 Joint-Threshold Optimization Mechanism

To coordinate the three optimization dimensions and avoid conflicting decisions, we design a joint-threshold optimization mechanism that formulates a constrained multi-objective search over threshold parameters to maximize inference speedup under user-specified accuracy constraints.

Let the threshold vector be $\boldsymbol{\tau} = \{\tau_s, \tau_a, \{\lambda_l\}_{l=1}^{L}\}$, where τ_s is the structural-pruning threshold, τ_a is the Softmax-approximation threshold, and λ_l is the activation-mixing coefficient at layer l. The three-dimensional collaborative optimization is formulated as:

$$\max_{\boldsymbol{\tau}} R(\boldsymbol{\tau}), \quad \text{s.t.} \quad A(\boldsymbol{\tau}) \le \epsilon_{\text{acc}}, \quad \boldsymbol{\tau} \in \Omega \tag{24}$$

where the objective $R(\boldsymbol{\tau}) = T_{\text{original}}/T_{\text{optimized}}(\boldsymbol{\tau})$ denotes the achieved speedup, and the accuracy-loss constraint is $A(\boldsymbol{\tau}) = A_{\text{original}} - A_{\text{optimized}}(\boldsymbol{\tau})$. ϵ_{acc} is the user-specified maximum allowable relative accuracy loss, and Ω denotes feasible parameter bounds.

We employ a serialized heuristic decomposition following the priority order *structure* → *operator* → *activation*, splitting the joint optimization into three sequential subproblems. A progressively allocated accuracy-loss budget is assigned to each subproblem to satisfy the global constraint. Algorithm 1 details this workflow. The approach isolates single-variable effects by fixing previously optimized parameters, transforming the complex coupled search into sequential single- or low-dimensional searches.

Algorithm 1: Serialized heuristic optimization.

```
Input: importance scores $S_{l,h}$, accuracy budget $\epsilon_{\text{acc}}$, layers $L$, validation set $\mathcal{D}_{\text{val}}$, steps $\Delta\tau_s, \Delta\tau_a, \Delta\beta$
Output: optimal thresholds $\boldsymbol{\tau}^* = (\tau_s^*, \tau_a^*, \{\lambda_l^*\}_{l=1}^{L})$
Initialize stage budgets: $\epsilon_1 = \epsilon_{\text{acc}}/3$, $\epsilon_2 = 2\,\epsilon_{\text{acc}}/3$, $\epsilon_3 = \epsilon_{\text{acc}}$
Stage I – Structure:
$\mathcal{C}_s \leftarrow \varnothing$
for $\tau_s = 0$ to $S_{\max}$ step $\Delta\tau_s$ do
    Apply pruning: $\text{Keep}(l, h) \leftarrow [S_{l,h} \ge \tau_s]$ (Eq.(15))
    Enforce minimum retention (Eq.(16))
    Temporarily set $\tau_a \leftarrow 1$, $\lambda_l \leftarrow 1\ \forall l$
    Evaluate $A(\tau_s)$ and $\text{Speedup}(\tau_s)$ on $\mathcal{D}_{\text{val}}$
    if $A(\tau_s) \le \epsilon_1$ then
        $\mathcal{C}_s \leftarrow \mathcal{C}_s \cup \{(\tau_s, A, \text{Speedup})\}$
    end if
end for
Select $\tau_s^* \leftarrow \arg\max_{(\tau_s, A, R) \in \mathcal{C}_s} R$ s.t. $A \le \epsilon_1$
Stage II – Operator:
$\mathcal{C}_a \leftarrow \varnothing$
for $\tau_a = 0$ to $S_{\max}$ step $\Delta\tau_a$ do
    Apply pruning using $\tau_s^*$
    Apply differentiated Softmax (Eq.(20)): Softmax if $S_{l,h} \ge \tau_a$, else PolySoftmax (Eq.(21))
    Fix $\lambda_l \leftarrow 1\ \forall l$
    Evaluate $A(\tau_s^*, \tau_a)$ and $\text{Speedup}(\tau_s^*, \tau_a)$
```

(Continued)

Algorithm 1 (continued)

```
    if $A(\tau_s^*, \tau_a) \le \epsilon_2$ then
        $\mathcal{C}_a \leftarrow \mathcal{C}_a \cup \{(\tau_a, A, \text{Speedup})\}$
    end if
end for
Select $\tau_a^* \leftarrow \arg\max_{(\tau_a, A, R) \in \mathcal{C}_a} R \quad \text{s.t. } A \le \epsilon_2$
Stage III – Activation:
$\mathcal{C}_\lambda \leftarrow \varnothing$
for each candidate layer-wise configuration $\{\lambda_l\}$ (per Eq. (22)) do
    Apply pruning with $\tau_s^*$ and operator approx with $\tau_a^*$
    Apply activation mixing per Eq.(22)
    Evaluate $A(\tau_s^*, \tau_a^*, \{\lambda_l\})$ and $\text{Speedup}(\tau_s^*, \tau_a^*, \{\lambda_l\})$
    if $A \le \epsilon_3$ then
        $\mathcal{C}_\lambda \leftarrow \mathcal{C}_\lambda \cup \{(\{\lambda_l\}, \Delta A, \text{Speedup})\}$
    end if
end for
Select $\{\lambda_l^*\} \leftarrow \arg\max_{(\{\lambda_l\}, A, R) \in \mathcal{C}_\lambda} R \quad \text{s.t. } A \le \epsilon_3$
return $\boldsymbol{\tau}^* = (\tau_s^*, \tau_a^*, \{\lambda_l^*\})$
```

A brute-force joint grid search over $G_s \times G_a \times G_\lambda^L$ exhibits exponential complexity in L, but our serialized strategy reduces this to $O(G_s + G_a + G_\lambda)$, achieving linear scalability. When any stage fails to find feasible candidates within its budget, we relax the previous stage's budget ϵ_{i-1} and re-run the search to restore feasibility through backtracking.

4 Experiments

4.1 Experimental Setup

We evaluate IG-3D on five GLUE benchmark tasks: natural language inference, question paraphrasing, sentiment analysis, and linguistic acceptability. We adopt BERT-Base as model backbones, initialized from pretrained checkpoints. Dataset statistics and model configurations are in Table 3.

Table 3: Datasets and models used in our experiments

	Datasets		
Dataset	**Task**	**Train**	**Dev**
MNLI	Multi-Genre NLI	392,702	9815
QQP	Question Pair Paraphrase	363,846	40,430
QNLI	Question–Answer NLI	104,743	5463
SST-2	Sentiment (Binary)	67,349	872
CoLA	Linguistic Acceptability	8551	1043
	Models		
Model	Layers	Hidden size	Attention heads
BERT-Base	12	768	144

We adopt multi-dimensional evaluation metrics: accuracy metrics (Accuracy, F1-score, Matthews Correlation Coefficient for CoLA), efficiency metrics (inference latency, communication cost, computational complexity in FLOPs), and optimization metrics (speedup ratio, accuracy retention). Experiments use 2 NVIDIA V100 GPUs (32 GB each), 2 Intel Xeon Gold 6248R CPUs (3.0 GHz), 512 GB DDR4 RAM, and 10 Gbps Ethernet. Software includes PyTorch 1.12.0, HuggingFace Transformers 4.21.0, CUDA 11.3, and Python 3.8.10. Key hyperparameters are integration steps $m = 16$, minimum head retention ratio $\alpha = 0.3$, batch size = 32, learning rate 2×10^{-5}, and 3 training epochs.

For private inference experiments, we run both the baseline and IG-3D–optimized models inside CrypTen's two-party MPC backend. The two parties are placed on the two V100 machines connected via the 10 Gbps Ethernet link. All reported latency numbers correspond to end-to-end secure inference time per batch as measured by CrypTen, and the communication cost metric is the total amount of data transferred between parties during a forward pass, obtained from CrypTen's runtime statistics. We do not apply additional numerical quantization or ciphertext packing beyond CrypTen's default fixed-point representation, so the observed trends mainly reflect the effect of IG-3D on activation sizes and nonlinear operation counts rather than backend-specific packing optimizations.

4.2 Experimental Results and Analysis

4.2.1 Performance of IG-3D across Datasets

We evaluate the proposed IG-3D framework against the standard BERT-Base baseline on selected GLUE tasks. Table 4 reports detailed results per task, where the baseline denotes the standard BERT-Base model fine-tuned on each dataset.

Table 4: Comparison between IG-3D and the BERT-Base on multiple GLUE tasks

Dataset	Metric	BERT-Base	IG-3D	Accuracy Loss	Speedup
MNLI	Acc	84.3	82.8	1.5%	2.8×
QQP	F1	88.2	87.1	1.1%	2.9×
QNLI	Acc	91.5	89.7	1.8%	2.7×
SST-2	Acc	91.9	91.2	0.7%	3.2×
CoLA	MCC	57.8	55.5	2.3%	3.0×
Average	—	—	—	1.48%	2.92×

As shown in Table 4, IG-3D achieves significant efficiency improvements with minimal performance loss. Across all evaluated tasks, IG-3D delivers an average 2.9× speedup in inference latency, while limiting accuracy degradation to 1.5% on average. These results demonstrate that IG-3D's unified three-dimensional optimization effectively reduces inference costs while maintaining acceptable performance levels.

4.2.2 Detailed Comparison with Representative Methods

To validate the effectiveness of the proposed Integrated-Gradients three-dimensional optimization framework (IG-3D) for secure Transformer inference, we compare it against three representative baselines adapted to the BERT backbone to ensure a fair comparison. PriViT [15] was originally designed for Vision Transformers and employs selective Taylor expansions for private inference; we adapt its approximation strategy for BERT-based text tasks. MPCFormer [10] designs quadratic approximations for GELU and

Softmax with knowledge distillation for MPC settings. Traditional Pruning ranks attention heads by gradient sensitivity and removes low-ranked heads.

Fig. 2 shows the aggregated performance of these methods on the MNLI task. IG-3D achieves a 2.8× speedup, substantially outperforming competing approaches through coordinated three-dimensional optimization: structural pruning reduces arithmetic workload, operator-level approximations lower nonlinear primitive costs, and activation replacements simplify feed-forward computations. Communication cost decreases approximately 50% reduction, as attention-head pruning reduces intermediate activation volume and polynomial approximations avoid multi-round interactions for complex exponential evaluations. IG-3D's accuracy degradation on MNLI is modest (1.5%), comparable to other methods, demonstrating effective accuracy-preservation mechanisms.

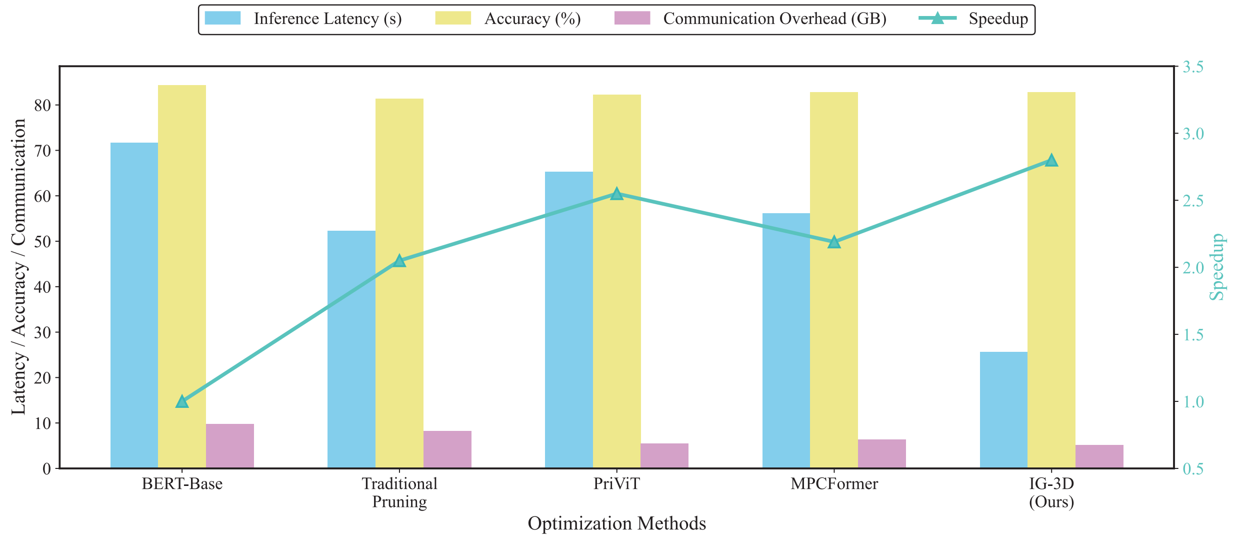


Figure 2: Integrated gradient-based transformer secure inference optimization multi-dimensional performance comparison

4.2.3 Performance across Model Scales

To validate the generality of IG-3D, we conduct additional experiments on a larger backbone (BERT-Large) and compare results with the BERT-Base configuration. As shown in Table 5, IG-3D consistently improves efficiency across model scales on MNLI. In terms of accuracy, IG-3D incurs only 1.5% and 2.6% absolute loss on BERT-Base and BERT-Large, respectively, while achieving 2.8×–2.9× speedup and 46.9%–48.2% reduction in communication volume. These results indicate that IG-3D scales favorably to larger models.

Table 5: Performance and end-to-end CrypTen PI metrics across model scales on MNLI (two-party additive secret sharing, batch size 4, seq. length 128)

Model scale	Method	Acc.	Acc. loss	Speedup	Comm. red.	Rounds	Comm. (GB)	Time/batch (s)
BERT-Base	Original model	84.3%	–	1.0×	–	2419	9.741	71.7
BERT-Base	IG-3D	82.8%	1.5%	2.8×	46.9%	1366	5.174	25.6

(Continued)

Table 5 (continued)

Model scale	Method	Acc.	Acc. loss	Speedup	Comm. red.	Rounds	Comm. (GB)	Time/batch (s)
BERT-Large	Original model	86.7%	–	1.0×	–	4783	26.522	143.4
BERT-Large	IG-3D	84.1%	2.6%	2.9×	48.2%	2659	13.738	49.4

In addition, Table 5 reports the end-to-end CrypTen PI metrics under the same two-party additive secret-sharing backend and hardware configuration as described in Section 4.1. For BERT-Base, the original model requires 2419 communication rounds and transfers 9.741 GB of data per batch, resulting in 71.7 s wall-clock latency, whereas IG-3D reduces these to 1366 rounds, 5.174 GB, and 25.6 s, respectively. For BERT-Large, the original model requires 4783 rounds and 26.522 GB with 372.4 s latency, while IG-3D reduces this to 2659 rounds, 13.738 GB, and 49.4 s. These concrete CrypTen measurements confirm that IG-3D's model-level optimizations lead to significant end-to-end PI efficiency gains on a representative MPC stack.

4.3 Ablation Study

4.3.1 Validity of the Integrated Gradients Evaluation Method

To validate the Integrated Gradients method's advantage over conventional importance-evaluation techniques, we compared pruning outcomes of several importance metrics at 40% pruning rate. Results in Table 6 show that Integrated Gradients achieves the highest post-pruning accuracy retention, demonstrating clear advantage in identifying redundant attention heads. This corroborates the theoretical analysis in Section 3: by integrating along input paths and smoothing noise, Integrated Gradients provides more accurate and stable importance estimates.

Table 6: Comparison of different importance evaluation methods

Evaluation method	Accuracy retention
Weight norm	96.8%
Gradient sensitivity	97.1%
Attention weight analysis	96.9%
Integrated gradients (Ours)	**98.3%**

Note: Bold indicates the results of our method.

4.3.2 Contribution Analysis of Different Optimization Dimensions

To quantify the contribution of each dimension in the 3D optimization strategy, we conducted a stepwise ablation. Table 7 reports performance on the MNLI task under different combinations.

Table 7: Stepwise ablation of the 3D optimization strategy (MNLI)

Optimization combination	Acc.	Acc. loss	Speedup	Comm. red	FLOPs
Baseline	84.3%	0%	1.0×	0%	22.5 G
+ Structural pruning	83.4%	0.9%	1.8×	28.3%	17.2 G

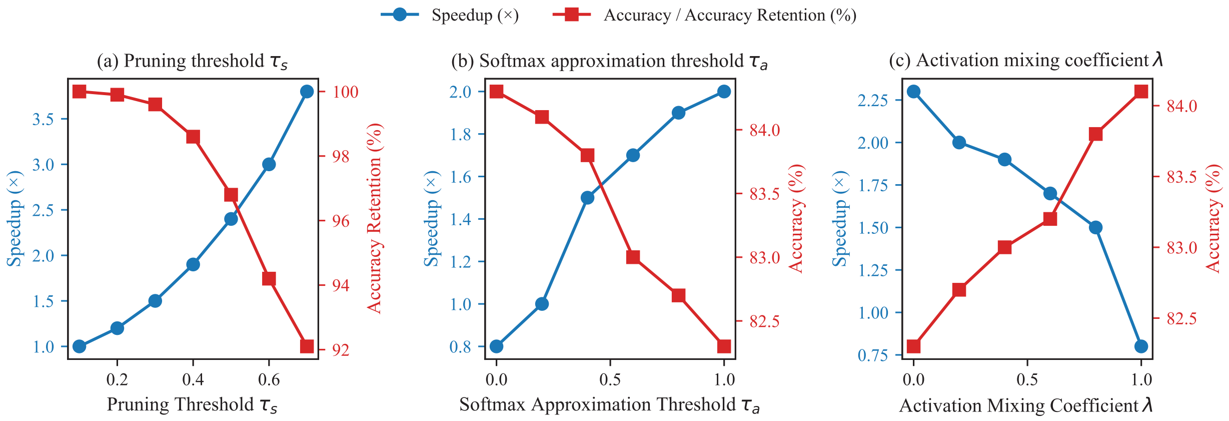
(Continued)

Table 7 (continued)

Optimization combination	Acc.	Acc. loss	Speedup	Comm. red	FLOPs
+ Operator approximation	83.1%	1.2%	2.4×	35.7%	17.2 G
+ Activation replacement	82.8%	1.5%	2.8×	46.9%	15.1 G

The ablation results reveal a clear complementary effect among the three optimization dimensions. Structural pruning provides the first-stage improvement, yielding a large parameter reduction and an initial 1.8× speedup with less than 1% accuracy loss. On top of this, operator approximation further increases the speedup to 2.4× and improves communication reduction, without introducing a substantial additional accuracy drop. Finally, activation replacement pushes the speedup to 2.8× and yields the largest reduction in both communication and FLOPs. In other words, each additional dimension still brings non-trivial gains when applied after the others, rather than saturating early. This monotonic improvement across all three stages indicates that the three axes act in a largely complementary manner and empirically supports our claim that a unified 3D optimization framework is more effective than applying pruning, polynomial approximation, or activation substitution in isolation.

We next analyze the sensitivity of IG-3D to its key hyperparameters. Fig. 3a shows the effect of the pruning threshold τ_s, which controls the strength of head pruning in the structure dimension. A moderate range $\tau_s \in [0.4, 0.6]$ achieves a good balance between speedup and accuracy, while larger values lead to more aggressive pruning, higher speedup, but noticeable accuracy degradation. Fig. 3b reports the sensitivity to the Softmax approximation threshold τ_a in the operator dimension. Larger τ_a increases the fraction of heads using the quadratic PolySoftmax approximation and thus yields higher speedup, but will hurt accuracy; values in $\tau_a \in [0.3, 0.5]$ preserve accuracy with reasonable acceleration. Fig. 3c evaluates the activation mixing coefficient λ in the activation dimension. Values around $\lambda \in [0.6, 0.8]$ provide a smooth transition between ReLU and GELU, maintaining accuracy while reducing computational overhead.


(a) Pruning threshold τ_s

(b) Softmax approximation threshold τ_a

(c) Activation mixing coefficient λ

Figure 3: Sensitivity of IG-3D performance to key hyperparameters

5 Conclusion

This paper addresses the high computational cost of Transformer models in private inference scenarios and proposes IG-3D, a unified optimization framework. IG-3D builds a precise attention-head importance estimation mechanism based on Integrated Gradients attribution and implements a coordinated "structure-operator-activation" optimization across attention and feed-forward components to achieve end-to-end model lightweighting. Experimental results on standard benchmarks show that IG-3D attains an average 2.9× speedup in inference latency while keeping accuracy loss within 2.3%.

An important direction for future work is to extend IG-3D to generative models such as GPT-2 and T5, whose autoregressive structures introduce different communication and nonlinearity bottlenecks in private inference (PI) settings. Another promising direction is to evaluate IG-3D under various cryptographic backends and measure energy consumption and latency in full end-to-end PI pipelines. These systems-level evaluations are beyond the scope of our current model-level focus but represent key next steps toward practical deployment. In addition, developing adaptive or task-aware optimization policies may further improve cross-task generalization and robustness. Finally, we plan to systematically evaluate IG-3D under distributional shifts, including out-of-domain and robustness benchmarks, to assess whether the efficiency–accuracy trade-offs observed on GLUE carry over to shifted data distributions in privacy-preserving settings.

In summary, IG-3D offers a theoretically grounded and practically effective solution for accelerating private Transformer inference, substantially improving efficiency under privacy constraints and helping pave the way for broader adoption of privacy-preserving NLP.

Acknowledgement: The authors declare no specific support to acknowledge in this section.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Lei Sun; methodology, Lei Sun and Jingwen Wang; software, Jingwen Wang; validation, Jingwen Wang, Peng Hu and Xiuqing Mao; formal analysis, Cuiyun Hu; investigation, Cuiyun Hu; resources, Cuiyun Hu; data curation, Jingwen Wang; writing—original draft preparation, Jingwen Wang; writing—review and editing, Zhihong Wang; visualization, Peng Hu; supervision, Cuiyun Hu; project administration, Peng Hu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Xue J, Zheng M, Hua T, Shen Y, Liu Y, Bölöni L, et al. Trojllm: a black-box trojan prompt attack on large language models. Adv Neural Inf Process Syst. 2023;36:65665–77.
2. Zheng M, Lou Q, Jiang L. Trojvit: trojan insertion in vision transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Piscataway, NJ, USA: IEEE; 2023. p. 4025–34.
3. Iqbal U, Kohno T, Roesner F. LLM platform security: applying a systematic evaluation framework to OpenAI's ChatGPT plugins. In: Proceedings of the AAAI/ACM conference on AI, ethics, and society. New York, NY, USA: ACM; 2024. p. 611–23.
4. Fan J, Vercauteren F. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive; 2012. Report No.: 2012/144.
5. Goldreich O. Secure multi-party computation. Manus Prelim Vers. 1998;78(110):1–108.
6. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In: International conference on machine learning. London, UK: PMLR; 2016. p. 201–10.

7. Moon J, Yoo D, Jiang X, Kim M. THOR: secure transformer inference with homomorphic encryption. IACR Cryptology ePrint Archive; 2024. Report No.: 2024/1881.
8. Luo J, Zhang Y, Zhang Z, Zhang J, Mu X, Wang H, et al. Secformer: fast and accurate privacy-preserving inference for transformer models via SMPC. In: Findings of the association for computational linguistics ACL 2024. Wierden, The Netherlands: ACL; 2024. p. 13333–48.
9. Chen T, Bao H, Huang S, Dong L, Jiao B, Jiang D, et al. The-x: privacy-preserving transformer inference with homomorphic encryption. arXiv:2206.00216. 2022.
10. Li D, Wang H, Shao R, Guo H, Xing E, Zhang H. MPCformer: fast, performant and private transformer inference with MPC. In: The Eleventh International Conference on Learning Representations [Internet]. OpenReview.net (Online); 2023 [cited 2025 Dec 20]. Available from: https://openreview.net/forum?id=CWmvjOEhgH-.
11. Michel P, Levy O, Neubig G. Are sixteen heads really better than one? In: Advances in neural information processing systems (NeurIPS). Red Hook, NY, USA: Curran Associates, Inc.; 2019. p. 14014–24.
12. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems (NeurIPS). Red Hook, NY, USA: Curran Associates, Inc.; 2017. p. 5998–6008.
13. Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies. Wierden, The Netherlands: ACL; 2019. p. 4171–86.
14. Hao M, Li H, Chen H, Xing P, Xu G, Zhang T. Iron: private inference on transformers. In: NIPS'22: proceedings of the 36th international conference on neural information processing syste. Red Hook, NY, USA: Curran Associates Inc; 2022. p. 15718–31.
15. Zheng M, Lou Q, Jiang L. Primer: fast private transformer inference on encrypted data. In: 2023 60th ACM/IEEE design automation conference (DAC). Piscataway, NJ, USA: IEEE; 2023. p. 1–6.
16. Pang Q, Zhu J, Möllering H, Zheng W, Schneider T. Bolt: privacy-preserving, accurate and efficient inference for transformers. In: 2024 IEEE symposiumon security and privacy (SP). Piscataway, NJ, USA: IEEE; 2024. p. 4753–71.
17. Chen Y, Meng X, Shi Z, Ning Z, Lin J. SecureTLM: private inference for transformer-based large model with MPC. Inf Sci. 2024;667(2):120429. doi:10.1016/j.ins.2024.120429.
18. Lu W, Huang Z, Gu Z, Li J, Liu J, Hong C, et al. Bumblebee: secure two-party inference framework for large transformers. IACR Cryptology ePrint Archive; 2023. Report No.: 2023/1678.
19. Xu T, Lu Lu, W J, Chen Y, Lin C, Wang R, et al. Breaking the layer barrier: remodeling private transformer inference with hybrid {CKKS} and {MPC}. In: 34th USENIX security symposium (USENIX Security 25). Berkeley, CA, USA: USENIX Association; 2025. p. 2653–72.
20. Voita E, Talbot D, Moiseev F, Sennrich R, Titov I. Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned. arXiv:1905.09418. 2019.
21. Shim K, Choi I, Sung W, Choi J. Layer-wise pruning of transformer attention heads for efficient language modeling. In: 2021 18th international SoC design conference (ISOCC). Piscataway, NJ, USA: IEEE; 2021. p. 357–8.
22. Lee E, Hwang Y. Automatic channel pruning for multi-head attention. arXiv:2405.20867. 2024.
23. Hendrycks D, Gimpel K. Gaussian error linear units (gelus). arXiv:1606.08415. 2016.
24. Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Netw. 2018;107(3):3–11. doi:10.1016/j.neunet.2017.12.012.
25. Zhang Y, Xue J, Zheng M, Xie M, Zhang M, Jiang L, et al. Cipherprune: efficient and scalable private transformer inference. arXiv:2502.16782. 2025.
26. Ghazvinian P, Podschwadt R, Panzade P, Rafiei MH, Takabi D. MOFHEI: model optimizing framework for fast and efficient homomorphically encrypted neural network inference. In: 2024 IEEE 6th international conference on trust, privacy and security in intelligent systems, and applications (TPS-ISA). Piscataway, NJ, USA: IEEE; 2024. p. 233–44.
27. Knott B, Venkataraman S, Hannun A, Sengupta S, Ibrahim M, van der Maaten L. Crypten: secure multi-party computation meets machine learning. Adv Neural Inf Process Syst. 2021;34:4961–73.

28. Dathathri R, Saarikivi O, Chen H, Laine K, Lauter K, Maleki S, et al. CHET: an optimizing compiler for fully-homomorphic neural-network inferencing. In: Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation. New York, NY, USA: ACM; 2019. p. 142–56.
29. Salimans T, Kingma DP. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In: NIPS'16: proceedings of the 30th international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc; 2016. p. 901–9.
30. Kobayashi G, Kuribayashi T, Yokoi S, Inui K. Attention is not only a weight: analyzing transformers with vector norms. arXiv:2004.10102. 2020.
31. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034. 2013.
32. Clark K, Khandelwal U, Levy O, Manning CD. What does bert look at? An analysis of bert's attention. arXiv:1906.04341. 2019.
33. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: International conference on machine learning. London, UK: PMLR; 2017. p. 3319–28.