# PCFormer: Accelerating Privacy-preserving Transformer Inference by Partition and Combination

**Bo Zeng[1], Zhi Pang[1], Yuyang Zhang[1], Kai Zhao[1], Tian Wu[2], Geying Yang[3], Lina Wang[1*], Run Wang[1*]**

[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China
[2]Digital Literacy and Skills Enhancement Research Center, Jiangxi Province Philosophy and Social Science Key Research Base, Nanchang University, Nanchang, 330031, China
[3]School of Cyber Science and Engineering, Tianjin University, China
{bobozen,zhipang,yuwanz,kizhao}@whu.edu.cn, wutian@ncu.edu.cn, yanggeying@tju.edu.cn, {lnwang,wangrun}@whu.edu.cn

## Abstract

In recent years, transformer-based models have achieved remarkable success in sensitive domains, including healthcare, finance and personalized services, but their deployment raises significant privacy concerns. Existing secure inference studies have introduced cryptographic techniques such as Homomorphic Encryption (HE) and Secure Multi-Party Computation (MPC). However, these approaches either target isolated model components or incur prohibitive computational and communication overheads, failing to support latency-sensitive or resource-limited environments. In our investigation, we identify substantial redundancy in the nonlinear operations and their alternation with linear layers in deep learning. Motivated by this observation, we propose PCFormer, a universal optimization methodology tailored for sequences of linear and nonlinear computations in the Transformer. PCFormer introduces structure-aware partition and combination techniques specially designed for Multi-Head Attention (MHA) and Feed-Forward Network (FFN). Specifically, we reveal the discrete sources of redundancy in the Softmax and GeLU functions during inference, implementing partitions at the token and channel levels, respectively. Subsequently, these reductions are then combined with the preceding and succeeding linear operations, thereby enhancing both computational and communication efficiency. Experimental results on GLUE benchmarks demonstrate that PCFormer achieves a 1.9× speedup in both computation and communication without compromising accuracy, compared to existing privacy-preserving Transformer frameworks. Furthermore, we demonstrate that PCFormer generalizes effectively to other deep learning architectures involving structured linear-nonlinear compositions under cryptographic constraints.

## Introduction

In recent years, Transformer-based models (Vaswani et al. 2017; Singla et al. 2024) have been the driving force behind significant advances in natural language processing (NLP). As these models are increasingly deployed in real-world services, particularly under the Transformer-as-a-Service (TaaS) paradigm (Radford et al. 2018), they have become integral to a wide range of sectors, including healthcare (Mayer, Cabrio, and Villata 2020), finance (Yang et al. 2022), and legal services (Yuan 2024), where applications often involve processing highly sensitive user data (Hahn and Rofin 2024). This has raised growing concerns about privacy, and driven the demand for secure AI deployment, bringing privacy-preserving computation techniques, such as Homomorphic Encryption (HE) (Brakerski, Gentry, and Vaikuntanathan 2014; Brakerski and Vaikuntanathan 2014; Fan and Vercauteren 2012; Cheon et al. 2017) and Secure Multi-Party Computation (MPC) (Yao 1986; Shamir 1979; Brassard, Crépeau, and Robert 1986), to the forefront. These techniques offer promising solutions to enable secure inference without compromising data confidentiality (Pang et al. 2024), and have demonstrated feasibility on toy models (He et al. 2024; Zeng et al. 2026).

However, applying these techniques directly to more complex architectures with nonlinear operations, such as Transformers, poses significant challenges (Devlin et al. 2019;

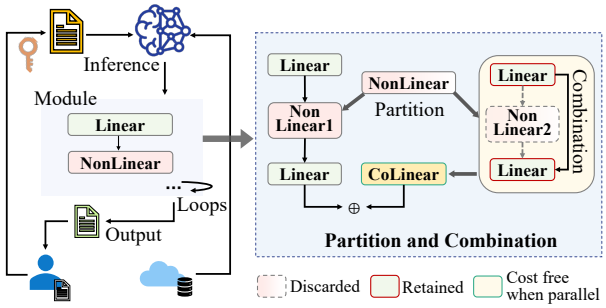



Figure 1: Conceptual illustration of the core principle of PCFormer: accelerating private-preserving inference by partitioning nonlinear operations and combining the linear operations that surround the redundant pathways.

*Lina Wang and Run Wang are corresponding authors.

Radford et al. 2019). Transformers feature deep, layered structures with frequent alternation between linear and nonlinear operations, such as the Softmax in attention mechanisms and GeLU in feed-forward networks, which significantly increase computational and communication costs under secure protocols (Hao et al. 2022). The prevailing methodology (Pang et al. 2024) is the hybrid protocol that judiciously integrates both HE and MPC. While this paradigm achieves a favorable trade-off in simpler models by assigning linear computations to HE and nonlinear operations to MPC (Lu et al. 2023), the intricate structure of Transformers still renders hybrid protocols impractical for private inference (Zhang et al. 2025b). A major bottleneck lies in the dense alternation between linear and nonlinear layers, where functions like Softmax and GeLU incur high computational and communication costs (Zhang et al. 2025a).

Prior works have explored various solutions. A body of research, including Iron (Hao et al. 2022), Bolt (Pang et al. 2024), and BumbleBee (Lu et al. 2023), focuses on accelerating matrix multiplication under homomorphic encryption to improve computational protocols. However, this direction faces a core limitation: the high baseline overhead of homomorphic operations remains unresolved, restricting overall performance gains. Another line of work, represented by CipherPrune (Zhang et al. 2025b) and Comet (Yan et al. 2025), tackles the problem architecturally by leveraging Transformer sparsity. Yet, these structural modifications risk compromising both information flow and model integrity, making it difficult to preserve the original functionality. Some studies instead pursue efficiency through non-homomorphic approximations (Kei and Chow 2025) or new security assumptions (Li et al. 2024). However, these alternatives have brought only limited practical improvements.

Motivated by advances in Model Compression (Zhu et al. 2024) and Diffusion Step Skipping techniques (Ye et al. 2025), we investigate the inherent computational sparsity within Transformer architectures. As detailed in **Section Motivation**, our analysis reveals that certain computational pathways contribute negligibly to the final output, indicating significant redundancy in Transformer inference. However, key challenges arise in accurately estimating sparsity, effectively managing the bypassed redundant data to maintain representational integrity, and ensuring the approach generalizes across various Transformer variants and their differing nonlinear operations.

Building on our observation, we propose PCFormer, a universal optimization framework grounded in a "partition and combination" paradigm, as depicted in **Figure 1**. PCFormer is tailored to sequential linear-nonlinear structures, markedly enhancing the efficiency of private inference. In essence, our methodology leverages nonlinear sparsity analysis to identify and circumvent computational pathways whose impact is negligible. This allows critical data to proceed immediately to nonlinear computation, while redundant elements bypass this stage and are routed directly to the subsequent linear layer. Moreover, as the disparate linear operations can be consolidated into a single execution, their time cost is substantially lower than the interactive latency inherent in nonlinear computations; when executed in parallel, their overhead becomes negligible. This methodology is universally applicable to any continuous linear-nonlinear structure. Additionally, to maintain inference accuracy despite aggressive optimization, we employ an iterative, architecture-aware fine-tuning technique that progressively adapts the model to the modified computation pathways. Experimental results on the GLUE benchmark demonstrate that our proposed PCFormer achieves significant improvements in both computational efficiency and communication overhead. Compared to state-of-the-art (SOTA) methods, PCFormer delivers a **1.9x** enhancement in both computational performance and communication efficiency, thereby enabling more potent and impactful applications in real-world scenarios. In summary, our principal contributions are as follows:

- We propose a universal optimization framework in private inference that leverages nonlinear sparsity to fuse adjacent linear operations and bypass redundant computation, fundamentally mitigating the efficiency bottlenecks in privacy-preserving deep learning.
- We implement the framework as PCFormer, a universal optimization technique for Transformers, which enhances the efficiency of MHA (Softmax) and FFN (GeLU) by leveraging token-level and channel-level sparsity, respectively.
- We deploy PCFormer on BERT and GPT2, and our experiments show that compared to the previous SOTA model, PCFormer achieved a **1.9x** reduction in end-to-end computation and communication costs without compromising accuracy. PCFormer also generalizes to other activation functions such as ReLU and SiLU, demonstrating broad applicability under linear-nonlinear cryptographic constraints.

## Related Work

Privacy-preserving Transformers primarily rely on HE (Brakerski, Gentry, and Vaikuntanathan 2014; Brakerski and Vaikuntanathan 2014; Fan and Vercauteren 2012; Cheon et al. 2017) and MPC (Yao 1986; Shamir 1979; Brassard, Crépeau, and Robert 1986) to secure data transmission. Research in this area focuses on reducing the significant overhead of these protocols to enable practical deployment (Yan et al. 2024). Existing approaches fall into two main categories: optimizing cryptographic protocols and removing redundant computations at the Transformer architecture level (Zeng et al. 2025).

### Efficient Computations at Protocol

Frameworks such as Iron (Hao et al. 2022) and BumbleBee (Lu et al. 2023) have concentrated on accelerating the linear operations performed under homomorphic encryption, enhancing overall efficiency by speeding up homomorphic matrix multiplication in MHA. Concurrently, Bolt (Pang et al. 2024) demonstrates that it is possible to further curtail the number of time-intensive homomorphic rotation operations. On another front, SecFormer (Luo et al. 2024) and SHAFT (Kei and Chow 2025) have turned

their attention to enhancing the computational performance of nonlinear functions within the privacy-preserving context. NEXUS (Zhang et al. 2025a) employs a purely homomorphic encryption-based approach to optimize both linear and nonlinear computations, thereby constructing a non-interactive Transformer inference architecture.

### Elimination of Redundancies at Architecture

Existing studies have observed that certain operations within the Transformer's computational graph are of negligible consequence to the final output. Comet (Yan et al. 2025), for instance, identifies the phenomenon of activation sparsity, leveraging it to bypass inconsequential computations in nonlinear activation functions and thereby enhance overall efficiency. CipherPrune (Zhang et al. 2025b) approaches the problem from a token-centric viewpoint, analyzing the relationship between tokens and Softmax redundancy to construct its corresponding pruning strategy. PrivCirNet (Xu et al. 2024), in a similar vein, employs structured sparse matrices to construct circulant blocks that approximate the original weight matrices. While these schemes can indeed directly curtail computational workload, their channel-level redundancy elimination strategies fail to fundamentally reduce the communication overhead, which remains proportional to the number of layers. Although Nimbus (Li et al. 2024) does take this bottleneck into consideration, its implementation alters the computational protocol by offloading a greater share of the computation to the client. This approach is inherently flawed within the typical TaaS paradigm, where clients are often computationally constrained, rendering any performance gains under such resource-limited conditions marginal at best.

## Preliminaries

In this section, we provide a concise overview of the foundational concepts pertinent to this work, encompassing the Transformer architecture and fundamental cryptographic primitives.

### Transformer Architecture

The architecture of the Transformer, as depicted in the left part of **Figure 2**, derives its principal computational workload from the MHA and FFN modules. The MHA module processes Query (Q), Key (K), and Value (V) matrices through matrix-matrix multiplication, after which the Softmax is applied to the intermediate result to introduce nonlinearity. This sequence of operations can be formally expressed by the following equation:

$$\mathrm{MHA}(x) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{k}}\right)V, \tag{1}$$

where $k$ is the dimension of $K$. The FFN module, in turn, primarily involves matrix-vector multiplication, and the resultant vector is subsequently processed by GeLU. This operation is formulated as follows:

$$\mathrm{FFN}(x) = W_2 \cdot \mathrm{GeLU}\left(W_1 \cdot x + b_1\right) + b_2, \tag{2}$$

where $b_1$ and $b_2$ are the biases.

### Cryptographic Primitives

Our architecture employs a hybrid protocol, mainly integrating HE based on Ring Learning with Errors (RLWE) problem and Additive Secret Sharing (ASS). These are strategically deployed to leverage their respective strengths in computing linear and nonlinear operations.

**Homomorphic Encryption.** In HE schemes based on the RLWE problem (BFV used in this work), a plaintext vector $m$ is first encoded into a polynomial representation, $\hat{m}$, with respect to a plaintext modulus $p$. This polynomial is subsequently encrypted, yielding a ciphertext $[m] = (\hat{b}, \hat{a})$ under a distinct ciphertext modulus $q$. The homomorphic property of the scheme permits both additions and multiplications to be performed directly on these ciphertexts, and the decrypted result is the same as it directly on the plaintext.

**Additive Secret Sharing.** For a given secret value $x$ to be shared between two parties, each participant is allocated a specific share. One party receives $x_1 = r$ and the other receives $x_2 = x - r$, where $r$ is a randomly generated value. A key property of this additive secret sharing construction is that each share, when viewed in isolation, is statistically indistinguishable from a random number. The original secret, $x$, can be correctly reconstructed if, and only if, both parties combine their respective, untampered shares.

### Threat Model

In this study, analogous to the frameworks presented in (Hao et al. 2022) and (Pang et al. 2024), we address a canonical two-party Private Inference (PI) scenario, a setting prevalent in protocols such as HE and MPC. This scenario involves two participating entities: the server possesses a pre-trained Transformer model and the client holds the sensitive input data (e.g., images, text, or audio). We operate under the semi-honest security model, wherein each party adheres to the protocol's procedures but may attempt to infer private information from the exchanged data. This paradigm entails dual, symmetrical privacy goals: the client seeks to protect its data throughout inference, while the server aims to safeguard its model parameters from disclosure.

## Method

This section provides a detailed exposition of the PCFormer, with its core framework illustrated in **Figure 2**. A foundational principle of our design is that the formation of any structural component invariably adheres to a two-phase process: *Partition* and *Combination*. Within this section, we will further elaborate on how this methodology is specifically applied to the Transformer's MHA and FFN modules.

### Motivation

To lucidly articulate our objective, we formalize the typical Transformer inference process as follows:

$$\mathrm{Output} = \mathrm{Transformer}(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x), \tag{3}$$

where each submodule $f_i$ comprises a linear mapping and a nonlinear activation, which can be deconstructed as $f_i(x) = \phi_i(W_i x)$, where $W_i$ represents the linear weights and $\phi_i(\cdot)$ is the nonlinear activation function (e.g., Softmax, GeLU). It

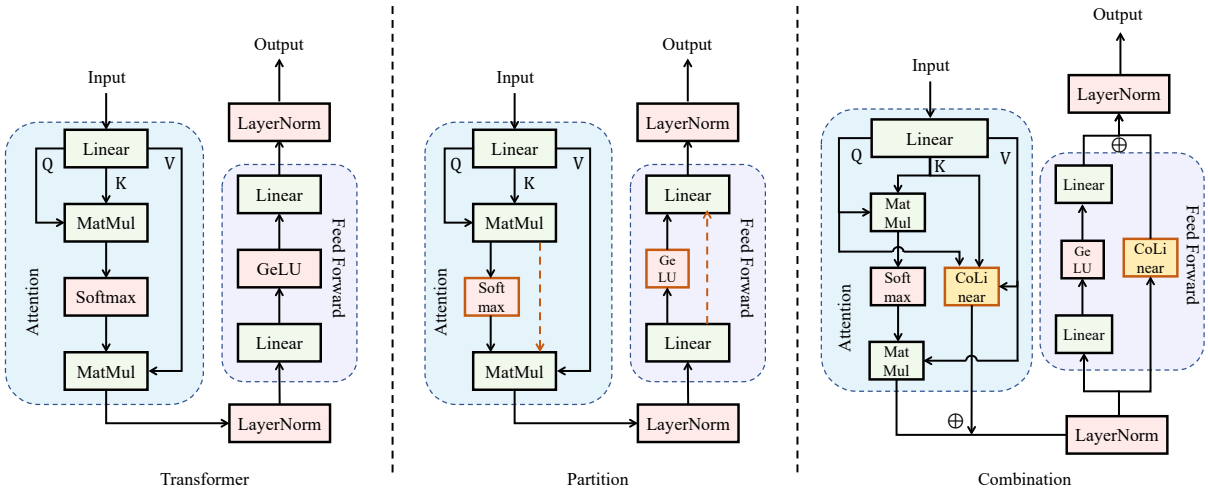



Figure 2: The framework of PCFormer. The left part delineates the foundational MHA and FFN modules, the central part illustrates the partition of the nonlinear Softmax and GeLU functions, while the right part reveals the combination of preceding and subsequent linear computations that address redundant data.

is thus evident that nonlinear operations are an inescapable component of every layer.

During our exploration shown in **Figure 3**, we discovere that Transformers contain a significant number of redundant (optimizable) channels and tokens, which contribute minimally to the final prediction. Consequently, if one could accurately identify this redundancy and circumvent its corresponding nonlinear pathways, it would be possible to drastically reduce the nonlinear computation and communication overhead while preserving model accuracy.

The logical structure of nonlinear operations inherently contains significant redundancy, which is directly correlated with the dimensionality of the data, while changes in data dimensionality primarily take place within linear operations. Crucially, server-side linear weights are maintained in plaintext. By merging the linear weights preceding and succeeding redundant nonlinear operations, new combined weights can be precomputed directly in plaintext. This precomputation enables the combined weights to be processed in full parallelism with the remaining computational path, without interference. Moreover, this resultant linear layer can be made trainable during the fine-tuning phase, presenting further augment the model's expressive capacity.

To this end, we propose a universal framework for nonlinear partition and linear combination. During the fine-tuning phases, our framework actively guides the model to concentrate its critical representations within a minority of "important channels" and "important tokens". This strategic concentration allows a vast number of redundant channels and positions to be transformed into purely linear pathways during inference, rendering them amenable to subsequent merging and acceleration.

## Token-wise Partition and Combination in MHA

Within the Transformer architecture, the MHA module serves as the cornerstone for modeling long-range dependencies. The Softmax function operates row-wise upon the attention scores to yield a probability distribution for each token. In PI scenario, the Softmax operation poses a particular challenge due to its reliance on exponentiation and normalization, making it significantly more costly than preceding linear computations. To address this, we propose a novel token-wise partitioning strategy, which selectively applies the expensive Softmax function only to a subset of critical tokens.

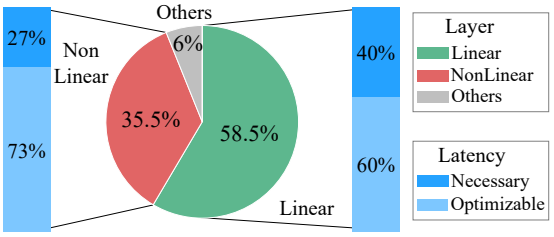



Figure 3: Statistics of Latency. Latency breakdown of the privacy-preserving inference on BERT-Base.

We utilize a structural reconstruction of the attention pathway within the MHA. Given a token importance mask $m \in \{0,1\}^N$, learned during the training phase, we partition the full set of tokens into two disjoint subsets: 1) The selected token set, $\mathcal{T}_s = \{i \mid m_i = 1\}$, comprising the important tokens; 2) The redundant token set, $\mathcal{T}_r = \{i \mid m_i = 0\}$. Accordingly, we reformulate the attention calculation into two parallel paths, i.e., *Activated Path* and *CoLinear Path*.

$$y_{active} = \text{Softmax}\left(\frac{Q_s K_s^\top}{\sqrt{k_s}}\right) V_s,\ y_{colinear} = W_{colinear} \cdot x_r, \quad (4)$$

For *Activated Path*, the standard attention mechanism is applied exclusively to the subset of important tokens, where $Q_s, K_s$, and $V_s$ are the representations corresponding to the

Algorithm 1: **MHA inference protocol in PCFormer**

**Input:** ASS shares $(x^S, x^C)$ of client and server
**Output:** ASS shares $(y^S, y^C)$ after MHA

$x = \text{ss\_to\_he}(x^S + x^C)$
$(x_s, x_r) = \text{Partition\_token}(x)$
$y = \text{MHA}(x_s) + \text{colinear}(x_r)$
**return** $(y^S + y^C) = \text{he\_to\_ss}(y)$

selected tokens $x_s \in \mathcal{T}_s$. For *CoLinear Path*, the attention path for redundant tokens is approximated by a single, direct linear transformation where we simplify this path to a unified linear mapping to the redundant token $x_r \in \mathcal{T}_r$ and the matrix $W_{colinear}$ is learnable during fine-tuning.

This design preserves the expressive capacity of models for the most salient tokens during training, while concurrently leveraging linear consolidation at inference time to transform the computational pathways of non-essential tokens into highly efficient linear operations, thereby augmenting the inference process.

Operationally, the *Activated Path* benefits from substantially enhanced processing speed and communication efficiency when executed under MPC, owing to the reduced scale of its input data. Concurrently, *CoLinear Path*, which consists of the consolidated linear operation, requiring no communications. And due to parallel execution, it introduces no additional latency to the critical path; its output is simply computed and held pending combination with the result from the *Activated Path*. Finally, our inference protocol in MHA is shown in **Algorithm 1**, and for details of cryptographic functions such as $\text{ss\_to\_he}$ and $\text{he\_to\_ss}$ are described in the Supplementary Materials.

## Channel-wise Partition and Combination in FFN

The FFN within the Transformer architecture is typically composed of two linear layers separated by GeLU. Within this module, the nonlinear activation is applied exclusively to the intermediate, or hidden, channel dimension. In the context of PI, the GeLU function requisites high-order polynomial approximations or interactive protocols drastically increase latency and resource consumption.

Notably, whereas the redundancy in the MHA's Softmax function is intrinsically linked to the sequence length (i.e., at the token level), the redundancy associated with the GeLU activation is predominantly channel-centric due to its computational form.

Our objective is therefore to identify critical channels during fine-tune, partitioning the activation function exclusively to this subset while combing a purely linear path for all other channels. Drawing an analogy to the token-level branching structure previously described, we introduce a channel importance mask, $m \in \{0,1\}^{d_h}$, to partition the intermediate channels into two distinct sets: 1) The selected channel set, $\mathcal{C}_s = \{i \mid m_i = 1\}$, designating those channels that will pass through the GeLU activation; 2) The redundant channel set, $\mathcal{C}_r = \{i \mid m_i = 0\}$, containing the channels that will bypass the nonlinear pathway.

Algorithm 2: **Iterative and Structure-aware fine-tuning**

**Input:** Fine-tuned model $f_\theta$, training data $\mathcal{D}$, epochs $T$, sparsity weights $\lambda_{\text{tok}}, \lambda_{\text{chn}}$
**Output:** Lightweight Transformer $f_{\theta'}$

*// Phase 1: Add learnable gates and bypass modules*
**for** each Transformer block **do**
  Add MHA token gate $G_{\text{tok}}$, colinear layer $W^{\text{colin}}_{\text{tok}}$
  Add FFN channel gate $G_{\text{chn}}$, colinear layer $W^{\text{colin}}_{\text{chn}}$
*// Phase 2: Joint training with sparsity-aware loss*
**for** epoch $t = 1$ to $T$ **do**
  Sample batch $(x, y) \sim \mathcal{D}$
  Generate soft masks:
    $M_{\text{tok}}, M_{\text{chn}} \leftarrow \text{GumbelSoftmax}(G_{\text{tok}}, G_{\text{chn}})$
  Forward pass with split paths:
    Attention output:
      $h_{\text{attn}} = M_{\text{tok}} \cdot \text{MHA}(x) + (1 - M_{\text{tok}}) \cdot W^{\text{colin}}_{\text{tok}}(x)$
    FFN output:
      $h_{\text{ffn}} = M_{\text{chn}} \cdot \text{FFN}(x) + (1 - M_{\text{chn}}) \cdot W^{\text{colin}}_{\text{ffn}}(x)$
  Combine and predict:
    $y_{\text{pred}} = f_\theta(x; h_{\text{attn}}, h_{\text{ffn}})$
  Loss:
    $\mathcal{L}_{\text{task}} \leftarrow \text{CrossEntropy}(y_{\text{pred}}, y)$
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{task}} + \lambda_{\text{tok}} \cdot \|M_{\text{tok}}\|_1 + \lambda_{\text{chn}} \cdot \|M_{\text{chn}}\|_1$
  Backpropagate and update $\theta$, $G$, and $W^{\text{colin}}$
*// Phase 3: Finalize structure*
Threshold $G_{\text{tok}}, G_{\text{chn}}$ to obtain binary masks $M'_{\text{tok}}, M'_{\text{chn}}$
For redundant units ($m = 0$), retain their learned linear bypass $W^{\text{colin}}$
Reparameterize $f_\theta \to f_{\theta'}$ with fixed structure
**return** $f_{\theta'}$

We can therefore rewrite the FFN module as the sum of a nonlinear path and a linear path:

$$\text{FFN}(x) = W_2^{(s)} \cdot \text{GeLU}\left(W_1^{(s)} \cdot x_s\right) + W_{colinear} \cdot x_r, \quad (5)$$

where $x_s \in \mathcal{C}_s$, $x_r \in \mathcal{C}_r$, and $W_1^{(s)}$ and $W_2^{(s)}$ are the submatrices of the weight tensors corresponding to the activated channels. The term $W_{colinear}$, which incorporates biases, is learnable and represents the consolidated linear mapping for the redundant path.

Building on the same core idea, we deconstruct the nonlinear function, employing a reasoning protocol similar to that used in the MHA. The complete protocol is detailed in the Supplementary Materials.

## Iterative and Structure-aware Fine-tuning

To implement the "partition and combination" strategy within the MHA and FFN modules during Transformer inference, it is imperative to fine-tune the model to adapt to this new architectural paradigm. This undertaking is nontrivial, as our modifications to the Transformer architecture introduce considerable challenges, principally for two reasons. On the one hand, fine-tuning is predicated upon a pre-trained model wherein the informational inputs to each

| Method | BERT-Medium | | BERT-Base | | BERT-Large | |
|---|---|---|---|---|---|---|
| | Time | Comm. | Time | Comm. | Time | Comm. |
| Iron | 350.4 | 124.9 | 609.9 | 282.6 | 1888.3 | 746.6 |
| Bolt | 102.2 | 14.5 | 260.5 | 26.0 | 648.4 | 78.4 |
| CipherPrune | 50.7 | 7.8 | 104.9 | 12.1 | 210.9 | 25.3 |
| PCFormer | **40.5** | **5.9** | **62.2** | **7.1** | **110.6** | **13.2** |

Table 1: End-to-end efficiency of PCFormer and baselines on BERT. Time is measured in seconds (s) while the communication overhead (Comm.) is in gigabytes (GB).

nonlinear function have been meticulously allocated. Consequently, effecting structural alterations to this established configuration presents significant difficulty. On the other hand, our modifications are comprehensive, encompassing both the MHA and FFN modules, and are therefore substantial in their scope.

To surmount these obstacles, we propose an iterative, structure-aware fine-tuning methodology, as delineated in **Algorithm 2**. Our approach decouples the processes of structural modification and model training. We employ a simple yet efficacious heuristic, the magnitude of the weights, to govern the partitioning decisions. Subsequently, a brief period of retraining is conducted to allow the model to assimilate these structural changes. The model is concurrently engaged in two learning objectives via $\mathcal{L}_{\text{total}}$: (1) refining its classificatory accuracy by updating the weights $\theta$; (2) dynamically learning which tokens and channels are to be partitioned by updating the masks $G_{\text{tok}}, G_{\text{chn}}$.

## Experiments

This section empirically validates PCFormer, evaluating its *Computational and Communication Overhead*, *Accuracy Changes* and *Ablation Studies*. We begin with the experimental setup, followed by detailed results. More experiments about GPT2 and other details are provided in the Supplementary Materials.

### Experiment Setup

We evaluate our solution on three BERT (Devlin et al. 2019) variants (Medium, Base, Large) and GPT2-Base (Radford et al. 2019). Following prior work (Pang et al. 2024; Zhang et al. 2025b), BERT models are fine-tuned on four GLUE (Wang et al. 2018) tasks: MNLI, QNLI, SST-2, and MRPC. We use the EzPC (Chandran et al. 2017) framework and SEAL (Microsoft Research 2023) HE library, under two simulated network conditions: LAN (3 Gbps, 0.8 ms) and WAN (20 Mbps, 40 ms). All tests run on a server with Intel Xeon Gold 6430 (2.10 GHz, 120 GB RAM) and an NVIDIA RTX 4090 GPU for fine-tuning.

### Main Result

**End-to-end performance.** We conduct a series of end-to-end benchmarks under a LAN setting, evaluating PCFormer against the antecedent SOTA methodologies: Iron (Hao et al. 2022), Bolt (Pang et al. 2024), and CipherPrune (Zhang et al. 2025b). The resultant execution times and communication overheads are systematically documented in **Table 1**. The results reveal that PCFormer unequivocally achieves superior performance in private inference across all three BERT model scales, demonstrating substantial enhancements in both temporal efficiency and communication economy. Notably, when deployed on the BERT-Large architecture, PCFormer delivers a 1.9x improvement in execution speed and communication when compared to CipherPrune, the most competitive prior method.

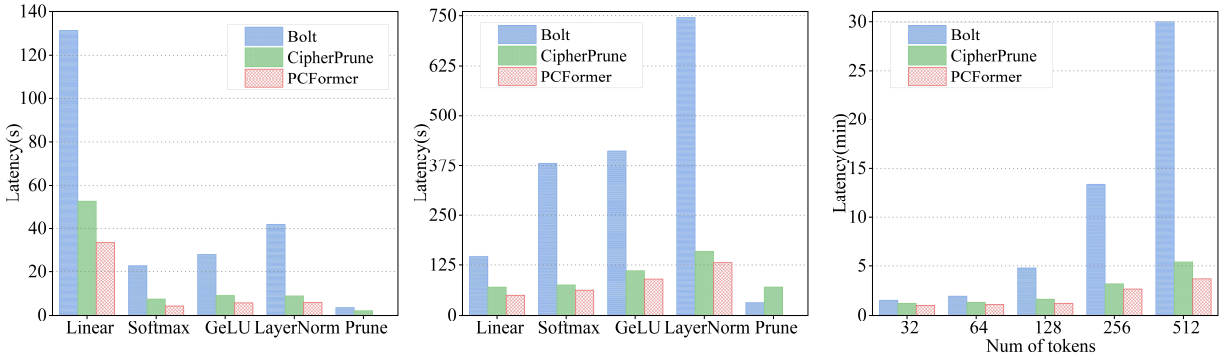
**Computational overhead breakdown.** To gain a more granular understanding, we conduct an in-depth analysis of the computation time required by each functional block during inference. We extend this analysis to a WAN scenario to assess performance variations across different network environments, with the results depicted in **Figure 4(a)&(b)**.

A primary observation is that, under both LAN and WAN conditions, PCFormer consistently exhibits lower computation times for every functional block when compared to Bolt and CipherPrune. Notably, PCFormer incurs no overhead for pruning at inference time, as its architecture is definitively established during the fine-tuning phase. Furthermore, our analysis reveals that in the LAN environment, where computational resources are ample, the primary overhead stems from the linear layers due to the introduction of HE operations at deployment. Conversely, in the WAN setting, the latency associated with nonlinear operations, namely Softmax, GeLU, and LayerNorm, escalates dramatically. This increase is attributable to their implementation via MPC, where reduced network bandwidth directly impacts communication-intensive protocols. Regardless of the network conditions, however, PCFormer's ability to reduce the computational dimensionality of both linear and nonlinear operations results in marked performance gains across all functional modules. Furthermore, as illustrated in **Figure 4(c)**, our analysis extends to the impact of token sequence length on the GPT2-Base model. While execution times predictably scale with the number of tokens for all methods, PCFormer still maintains performance advantages.

### Accuracy Changes

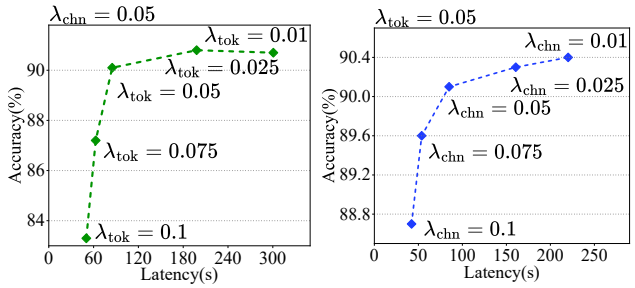
A key distinction among Bolt, CipherPrune, and PCFormer lies in their approach to data dimensionality. While all three methodologies involve dimensional alterations, Bolt and CipherPrune rely on conventional pruning strategies. PCFormer, in contrast, re-engineers the Transformer architecture through its "partition and combine" paradigm.

We assessed the accuracy of these models on four downstream tasks, i.e., MNLI, QNLI, SST-2, and MPRC. The results, as delineated in **Table 2**, demonstrate that PCFormer maintains a comparably high level of accuracy across all four benchmarks. This sustained performance is attributable to its fundamental divergence from direct pruning. Instead of merely excising redundant components, PCFormer strategically recombines them to furnish auxiliary information for the final prediction. Crucially, as established by the preceding theoretical and experimental analyses, this recombination process, executed within a parallel computational framework, incurs no additional computational or communication overhead.




(a) BERT-Base in LAN (b) BERT-Base in WAN (c) GPT2-Base Changes as Token

Figure 4: Computational overhead for the BERT-Base and GPT2-Base models. Figures 4(a) and 4(b) delineate a detailed breakdown of the BERT-Base runtime under LAN and WAN conditions, respectively. Figure 4(c) illustrates the variation in runtime for GPT2-Base as the number of the token changes.




(a) Ablation study on $\lambda_{\text{tok}}$ (b) Ablation study on $\lambda_{\text{chn}}$

Figure 5: Impact of hyperparameters $\lambda_{\text{tok}}$ and $\lambda_{\text{chn}}$ on accuracy.

| Method | Accuracy(%) ↑ | | | | Time(s) ↓ |
|---|---|---|---|---|---|
| | MNLI | QNLI | SST2 | MRPC | |
| Bolt | 84.71 | 89.94 | 92.74 | 89.95 | 260.5 |
| CipherPrune | 84.68 | 90.11 | 92.66 | 90.18 | 104.8 |
| PCFormer | **84.73** | **90.22** | **92.75** | **90.21** | **62.2** |

Table 2: Accuracy changes of different methods on four downstream fine-tuning tasks of GLUE.

## Ablation Studies

**Impact of fine-tuning hyperparameters.** We proceed to investigate the impact of key hyperparameters on the final model performance during the fine-tuning phase. The hyperparameters in question are $\lambda_{\text{tok}}$ and $\lambda_{\text{chn}}$, which respectively govern the sparsity ratio of tokens processed by the Softmax and of channels processed by the GeLU. Our findings from experiments conduct on the BERT-Base model are illustrated in **Figure 5**. A key observation is that the model's performance exhibits greater sensitivity to variations in token sparsity ($\lambda_{\text{tok}}$) compared to channel sparsity ($\lambda_{\text{chn}}$). Nevertheless, when both $\lambda_{\text{tok}}$ and $\lambda_{\text{chn}}$ are maintained below a threshold of 0.05, the model's accuracy remains stable and robust. Beyond this threshold, however, a more pronounced increase in sparsity leads to a significant degradation in model accuracy.

| Type | Baseline | | | PCFormer | | | |
|---|---|---|---|---|---|---|---|
| | Linear | NonLin | Total | Linear | NonLin | CoLin | Total |
| ReLU | 723.5 | 50.5 | 774.3 | 228.7 | 28.4 | 89.7 | **257.1** |
| SiLU | 2673.3 | 1277.2 | 3950.5 | 816.8 | 216.5 | 283.4 | **1033.3** |

Table 3: Latency(ms) for ReLU and SiLU structures. "NonLin" and "CoLin" are abbreviations for NonLinear and CoLinear, respectively.

**Impact of layer type.** Our proposed "partition and combination" method is a universal framework designed for "linear-nonlinear" architectural patterns. To validate its versatility, we evaluate the efficiency gains when applying our method to two other prevalent nonlinear activation functions in deep learning: the Rectified Linear Unit (ReLU) and the Sigmoid Linear Unit (SiLU). As presented in **Table 3**, by deconstructing the nonlinear computation, the reduced dimensionality of the data entering the original activation pathway leads to substantial performance enhancements. Simultaneously, the partitioned linear component is processed in parallel, culminating in a significant overall acceleration of the entire composite linear-nonlinear operation.

## Conclusion

To reduce the substantial computational and communication overhead introduced in privacy-preserving deep learning, we propose a universal partition and combination framework specifically designed for common linear–nonlinear structures. Applying this framework to the Transformer, we propose PCFormer, a novel model that analyzes and reduces nonlinear redundancy within MHA and FFN modules. Experiments demonstrate that PCFormer achieves significant improvements in computational efficiency and communication overhead compared to prior methods. Furthermore, the underlying framework exhibits broad applicability across diverse deep learning architectures.

## Ethical Statement

This work focuses on enhancing the efficiency of privacy-preserving inference frameworks. Our contribution is to strengthen secure computing capabilities without introducing new risks to data security or user privacy. We exclusively use publicly available datasets and models, involving no proprietary, sensitive, or personal data. We strictly adhere to ethical standards, ensuring that our experiments and proposed methodology do not introduce any privacy violations. In short, this research is designed to accelerate privacy-preserving inference and aligns with the principles of responsible innovation.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 62372334, No. 62202340, No. 62576255, the National Key Research and Development Program of China under No.2023YFB3106900, the Fundamental Research Funds for the Central Universities under No. 2042025kf0054, the Natural Science Foundation of Hubei Province under No. 2025AFB455.

## References

Brakerski, Z.; Gentry, C.; and Vaikuntanathan, V. 2014. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3): 1–36.

Brakerski, Z.; and Vaikuntanathan, V. 2014. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on computing*, 43(2): 831–871.

Brassard, G.; Crépeau, C.; and Robert, J.-M. 1986. All-or-nothing disclosure of secrets. In *Conference on the Theory and Application of Cryptographic Techniques*, 234–238. Springer.

Chandran, N.; Gupta, D.; Rastogi, A.; Sharma, R.; and Tripathi, S. 2017. EzPC: Programmable, Efficient, and Scalable Secure Two-Party Computation. *IACR Cryptol. ePrint Arch.*, 2017: 1109.

Cheon, J. H.; Kim, A.; Kim, M.; and Song, Y. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in cryptology–ASIACRYPT 2017: 23rd international conference on the theory and applications of cryptology and information security, Hong kong, China, December 3-7, 2017, proceedings, part i 23*, 409–437. Springer.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Fan, J.; and Vercauteren, F. 2012. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*.

Hahn, M.; and Rofin, M. 2024. Why are sensitive functions hard for transformers? *arXiv preprint arXiv:2402.09963*.

Hao, M.; Li, H.; Chen, H.; Xing, P.; Xu, G.; and Zhang, T. 2022. Iron: Private inference on transformers. *Advances in neural information processing systems*, 35: 15718–15731.

He, J.; Yang, K.; Tang, G.; Huang, Z.; Lin, L.; Wei, C.; Yan, Y.; and Wang, W. 2024. Rhombus: Fast homomorphic matrix-vector multiplication for secure two-party inference. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2490–2504.

Kei, A. Y.; and Chow, S. S. 2025. SHAFT: Secure, Handy, Accurate, and Fast Transformer Inference. In *Network and Distributed System Security Symposium, NDSS*, volume 2025.

Li, Z.; Yang, K.; Tan, J.; Lu, W.-j.; Wu, H.; Wang, X.; Yu, Y.; Zhao, D.; Zheng, Y.; Guo, M.; et al. 2024. Nimbus: Secure and efficient two-party inference for transformers. *Advances in Neural Information Processing Systems*, 37: 21572–21600.

Lu, W.-j.; Huang, Z.; Gu, Z.; Li, J.; Liu, J.; Hong, C.; Ren, K.; Wei, T.; and Chen, W. 2023. Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*.

Luo, J.; Zhang, Y.; Zhang, Z.; Zhang, J.; Mu, X.; Wang, H.; Yu, Y.; and Xu, Z. 2024. Secformer: Towards fast and accurate privacy-preserving inference for large language models. *arXiv preprint arXiv:2401.00793*.

Mayer, T.; Cabrio, E.; and Villata, S. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020*, 2108–2115. IOS Press.

Microsoft Research, W., Redmond. 2023. Microsoft SEAL (release 4.1). https://github.com/Microsoft/SEAL.

Pang, Q.; Zhu, J.; Möllering, H.; Zheng, W.; and Schneider, T. 2024. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, 4753–4771. IEEE.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Shamir, A. 1979. How to share a secret. *Communications of the ACM*, 22(11): 612–613.

Singla, S.; Thakur, A.; Swami, A.; Sawarn, U.; Singla, P.; et al. 2024. Advancements in Natural Language Processing: BERT and Transformer-Based Models for Text Understanding. In *2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech)*, 372–379. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Xu, T.; Wu, L.; Wang, R.; and Li, M. 2024. Privcirnet: Efficient private inference via block circulant transformation. *Advances in Neural Information Processing Systems*, 37: 111802–111831.

Yan, B.; Li, K.; Xu, M.; Dong, Y.; Zhang, Y.; Ren, Z.; and Cheng, X. 2024. On protecting the data privacy of large language models (llms): A survey. *arXiv preprint arXiv:2403.05156*.

Yan, G.; Zhang, Y.; Guo, Z.; Zhao, L.; Chen, X.; Wang, C.; Wang, W.; Meng, D.; and Hou, R. 2025. Comet: Accelerating Private Inference for Large Language Model by Predicting Activation Sparsity. In *2025 IEEE Symposium on Security and Privacy (SP)*, 2827–2845. IEEE.

Yang, L.; Li, J.; Dong, R.; Zhang, Y.; and Smyth, B. 2022. Numhtml: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11604–11612.

Yao, A. C.-C. 1986. How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (Sfcs 1986)*, 162–167. IEEE.

Ye, Z.; Chen, Z.; Li, T.; Huang, Z.; Luo, W.; and Qi, G.-J. 2025. Schedule on the fly: Diffusion time prediction for faster and better image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 23412–23422.

Yuan, J. 2024. Efficient techniques for processing medical texts in legal documents using transformer architecture. In *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, 990–993. IEEE.

Zeng, B.; Wu, T.; Peng, S.; Pang, Z.; Yu, R.; and Wang, L. 2026. Efficient privacy-preserving convolutional neural network inference with skip-perm. *Advanced Engineering Informatics*, 69: 104081.

Zeng, W.; Xu, T.; Chen, Y.; Zhou, Y.; Zhang, M.; Tan, J.; Hong, C.; and Li, M. 2025. Towards Efficient Privacy-Preserving Machine Learning: A Systematic Review from Protocol, Model, and System Perspectives. *arXiv preprint arXiv:2507.14519*.

Zhang, J.; Yang, X.; He, L.; Chen, K.; Lu, W.-j.; Wang, Y.; Hou, X.; Liu, J.; Ren, K.; and Yang, X. 2025a. Secure Transformer Inference Made Non-interactive. In *NDSS*.

Zhang, Y.; Xue, J.; Zheng, M.; Xie, M.; Zhang, M.; Jiang, L.; and Lou, Q. 2025b. CipherPrune: Efficient and Scalable Private Transformer Inference. In *The Thirteenth International Conference on Learning Representations*.

Zhu, X.; Li, J.; Liu, Y.; Ma, C.; and Wang, W. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577.