applied sciences

MDPI

Article

Enhancing Privacy in Large Language Model with Homomorphic Encryption and Sparse Attention

Lexin Zhang †, Changxiang Li †, Qi Hu †, Jingjing Lang †, Sirui Huang, Linyue Hu, Jingwen Leng, Qiuhan Chen and Chunli Lv *

China Agricultural University, Beijing 100083, China
* Correspondence: lvcl@cau.edu.cn
† These authors contributed equally to this work.

Abstract: In response to the challenges of personal privacy protection in the dialogue models of the information era, this study introduces an innovative privacy-preserving dialogue model framework. This framework seamlessly incorporates Fully Homomorphic Encryption (FHE) technology with dynamic sparse attention (DSA) mechanisms, aiming to enhance the response efficiency and accuracy of dialogue systems without compromising user privacy. Experimental comparative analyses have confirmed the advantages of the proposed framework in terms of precision, recall, accuracy, and latency, with values of 0.92, 0.91, 0.92, and 15 ms, respectively. In particular, the newly proposed DSA module, while ensuring data security, significantly improves performance by up to 100 times compared to traditional multi-head attention mechanisms.

Keywords: privacy-preserving dialogue systems; Fully Homomorphic Encryption; dynamic sparse attention mechanism; data security in artificial intelligence; large language model

Citation: Zhang, L.; Li, C.; Hu, Q.; Lang, J.; Huang, S.; Hu, L.; Leng, J.; Chen, Q.; Lv, C. Enhancing Privacy in Large Language Model with Homomorphic Encryption and Sparse Attention. *Appl. Sci.* **2023**, *13*, 13146. https://doi.org/10.3390/app132413146

Academic Editor: Gianluca Lax

Received: 7 November 2023
Revised: 27 November 2023
Accepted: 7 December 2023
Published: 11 December 2023

1. Introduction

As artificial intelligence technology rapidly evolves [1–3], conversational large models have become a significant product of the information age, playing an increasingly critical role in customer service, personal assistance, health consultation, and various other domains [4,5]. However, as these models deepen their reliance on personal data, issues of data privacy protection have progressively come to the fore.

It was discovered by Jain Naman et al. that conversational large models enhance programmer productivity but struggle to ensure code quality [6]. Kurstjens Steef et al. validated the practicality of large models in the hemoglobin domain by querying ChatGPT about hemoglobinopathies [7]. Tanisha Jowsey et al. deployed conversational large models in the education sector of medicine and health sciences, using them for student teaching evaluations, which significantly boosted learning efficiency, although the safety of their use must be considered [8]. Leippold Markus discussed climate change issues and their impact on the economy and finance using conversational large models, noting that while the models' responses were correct, they also included repetitive or unrealistic outputs [9]. Zhong et al. analyzed transparency, accountability, and ethical issues brought by AI large language models, proposing solutions and discussing their application in psychiatric research and practice, amidst concerns over privacy violations [10]. Sorin Vera et al. discussed the application of conversational large models in oncology, which could enhance the accuracy of cancer research and care, reminding oncologists to be cognizant of their limitations and privacy issues [11].

In this context, researching and developing conversational large models that provide efficient services while protecting user privacy has become a significant challenge and hot topic in the field of artificial intelligence. Privacy security is critically important in today's digital age, especially when dealing with conversational systems that need to

handle sensitive personal information. These systems must be capable of understanding and responding to user queries while ensuring that personal data are not misused or leaked to unauthorized third parties. Therefore, providing privacy-safe conversational large models is crucial for the personal privacy rights of users and directly impacts the compliance and reputation of enterprises.

Hua et al. proposed a big data (BD) privacy protection model based on image encryption algorithms to address privacy leaks in the era of big data and compared it with traditional BD models, demonstrating that their model better protected user privacy with protection rates of 82.25% and 82.41% against tuple and attribute attacks, respectively [12]. Wei et al. introduced a privacy-aware information security risk assessment model—pISRA—that calculates privacy impact based on data recognizability, context, volume, and sensitivity [13]. Duy-Hien Vu et al. proposed a secure multi-party summation protocol based on the multi-party summation function, verifying its efficiency through privacy analysis and efficiency evaluation, though the method has a high computational complexity [14]. Zhou et al. addressed blockchain privacy issues using SMC, designing a new protocol based on Beaver's randomization technique to perform multiplication on encrypted secret shares, integrating the SMC protocol into Hyperledger Fabric, eliminating communication interactions between participants and the blockchain. However, most of the method's time was spent on communication, with performance not well guaranteed [15]. Oladayo Olufemi Olakanmi et al. formulated a secure privacy-preserving offloading scheme based on modified secret sharing and developed a morphology-based method to protect privacy data on IoT platforms, solving privacy leakage issues and the detection of worker honesty with low overhead [16].

Fully Homomorphic Encryption (FHE) technology provides a potential solution. FHE allows computations to be directly performed on encrypted data without the need for decryption, meaning conversational systems can process user requests without accessing any sensitive plaintext data. Thus, even if data leaks occur during processing or transmission, attackers cannot obtain any useful information, as they only see encrypted data.

Kim Jeongsu et al. proposed a new security concept—Fully Homomorphic Authenticated Encryption (FHAE)—employing an encryption-before-authentication pattern, but it was not very efficient [17]. To address this, they further introduced multi-dataset fully homomorphic authenticated encryption (MDFHAE), achieving privacy protection with high efficiency. Xu et al. proposed a general framework for constructing multi-key FHE, evaluating ciphertexts through appropriate computational protocols, but the practicality of the model remains an issue [18]. Zhang et al. constructed Quantum Fully Homomorphic Encryption (QFHE) schemes against quantum computing, proposing two QFHE schemes: single-qubit point obfuscation and multi-qubit point obfuscation. Both of these were proved to be highly secure, but future needs for secure multi-party computation must be considered [19]. Yagisawa Masahiro introduced an indistinguishability under chosen plaintext attacks (IND-CPA) secure FHE algorithm based on the difficulty of factoring in cloud computing, with the computational overhead of homomorphic evaluation at O(1) [20]. Cai et al. presented a new private set intersection protocol based on the Gao tree red FHE scheme, which is simple and practical, but limited to two-party protocols and not very efficient [21]. Peng et al. proved the insecurity of FHE applications—the Brakerski/Fan–Vercauteren (BFV) scheme—in IoT, confirming the potential security risks in the practical application of FHE; while FHE is secure, its protocols are prone to issues [22].

The introduction of this technology provides new possibilities for protecting privacy in conversational systems. However, applying FHE technology to conversational large models is not without challenges. Firstly, traditional FHE schemes have significant limitations in computational efficiency, making them difficult to apply directly to online conversational systems requiring quick responses. Secondly, the complexity of conversational large models themselves and the high-dimensional data processing requirements further exacerbate this challenge. Against this backdrop, the present study proposes a privacy-preserving conversational large model framework based on FHE and attention mechanisms. The at-

tention mechanism, a technique that allows models to focus on important parts of the input sequence, has proven its utility in the field of natural language processing, particularly within Transformer models [4,5]. It enables the identification of information segments most relevant to responding to user requests by allocating different weights. Our study not only explores how to combine the attention mechanism with FHE to process encrypted data but also proposes a dynamic sparse attention module aimed at enhancing the model's computational efficiency and processing speed.

In this paper, we first provide a detailed review of the fundamental principles of FHE and its applications in privacy protection. We then discuss the attention mechanism, specifically its application in transformer models, and how it can contribute to performance improvements in conversational systems. Next, we introduce our proposed FHE-based privacy-preserving conversational large model framework, explaining how complex dialogues can be processed without sacrificing user privacy. Additionally, we propose a dynamic sparse attention module, an important enhancement to existing technology, which significantly improves the model's computational efficiency while maintaining privacy protection. The significance of this research lies in that it not only proposes a new model for privacy-safe conversational systems but also offers practical guidance on effectively integrating FHE technology into real-time, efficient conversational systems. Through the studies presented in this paper, we aim to provide a more robust theoretical foundation and technical support for privacy security in conversational systems, offer stronger data security for users, and provide direction for future research in this domain.

2. Related Work

2.1. Fully Homomorphic Encryption

The fundamental principle of FHE enables arbitrary computations to be performed on ciphertexts, with the result of such computations, when decrypted, remaining correct [23]. The essence of this principle is to process and analyze data without exposing the original data content, thus accomplishing complex data processing tasks while protecting privacy [24].

Mathematically, assume a plaintext message m is encrypted using a public key pk to produce a ciphertext c, as shown in Figure 1:

$$c = \mathrm{Enc}_{pk}(m) \tag{1}$$

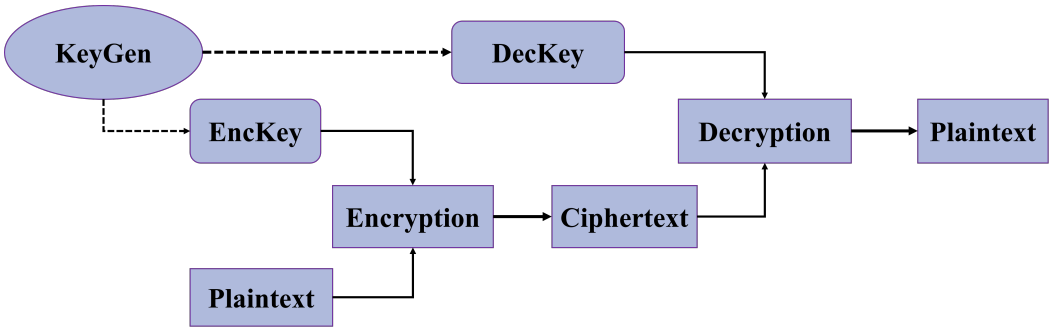

Figure 1. Flowchart of FHE.

FHE supports two basic operations: homomorphic addition and multiplication. Given two ciphertexts $c_1 = \mathrm{Enc}_{pk}(m_1)$ and $c_2 = \mathrm{Enc}_{pk}(m_2)$, homomorphic addition and multiplication can be represented as:

$$c_{\mathrm{add}} = c_1 \oplus c_2 = \mathrm{Enc}_{pk}(m_1 + m_2) \tag{2}$$

$$c_{\mathrm{mul}} = c_1 \otimes c_2 = \mathrm{Enc}_{pk}(m_1 \cdot m_2) \tag{3}$$

Here, $\oplus$ and $\otimes$ denote the homomorphic addition and multiplication operations, respectively. In the context of large-scale dialogue models, for example, the intention might be to compute the language model scores of a user without revealing the user's input.

If the user's input is x and the language model parameters are θ, the desired score to compute is $f(x;\theta)$. With FHE, the input x can be encrypted on the user's device to obtain $c_x = \mathrm{Enc}_{pk}(x)$, then the computation under encryption is performed on the server to obtain the encrypted score $c_f = \mathrm{Enc}_{pk}(f(x;\theta))$, and finally, c_f is sent back to the user's device for decryption. The application of FHE in natural language processing (NLP) tasks can extend to various machine learning models, such as sentiment analysis and text classification. For instance, in sentiment analysis, the model might need to encode each word in a sentence and compute sentiment predisposition. For each word, w_i, encoded as v_i, the sentiment predisposition calculation can be viewed as the dot product of a weight matrix W and the encodings:

$$s = \sum_i W_i \cdot v_i \tag{4}$$

If each v_i is encrypted, then the aforementioned computation needs to be completed under homomorphic encryption:

$$c_s = \bigoplus_i (W_i \otimes c_{v_i}) \tag{5}$$

In the task of this paper, a privacy-protected dialogue large model framework based on FHE and attention mechanisms, the computation of attention weights can also be implemented using homomorphic encryption. Normally, a simple attention weight computation can be expressed as:

$$\alpha_{ij} = \frac{\exp(\mathrm{score}(h_i, h_j))}{\sum_k \exp(\mathrm{score}(h_i, h_k))} \tag{6}$$

where *score* is a scoring function, and h_i and h_j are different states in a sequence. Under the framework of homomorphic encryption, this computation can be transformed into:

$$c_{\alpha_{ij}} = \frac{\exp(\mathrm{Enc}_{pk}(\mathrm{score}(h_i, h_j)))}{\bigoplus_k \exp(\mathrm{Enc}_{pk}(\mathrm{score}(h_i, h_k)))} \tag{7}$$

However, as exponential operations and divisions are not directly supported in homomorphic operations, as shown in Figure 1, approximate algorithms or special encoding methods need to be designed to implement these operations. In summary, FHE technology offers significant potential in protecting privacy within large dialogue models. Although FHE currently faces challenges in computational efficiency, with advancements in algorithms and hardware, it is expected to be widely applied in the near future. In this paper, we will explore in detail how to apply FHE to large dialogue models, particularly, how to ensure the correctness and efficiency of computations in natural language tasks by designing effective homomorphic algorithms. Through such research, it is hoped to advance the application of privacy protection technologies in dialogue systems, providing a stronger technical guarantee for the security of user data.

2.2. Attention Mechanism

The attention mechanism is a significant innovation in the field of deep learning in recent years, simulating the mechanism of human visual attention: rather than passively receiving all visual information when observing the surrounding world, attention is selectively focused on certain parts according to the current task [4,25,26]. In computer vision and NLP tasks, the attention mechanism allows the model to dynamically focus on important parts of the input data, significantly improving the efficiency and effectiveness of information processing.

Mathematically, the attention mechanism can be viewed as a mapping function that maps a set of queries, keys, and values to an output. Queries are usually related to the current task or target state, while keys and values correspond to different parts of the input data. The attention function computes the similarity between each key and the query,

generating a weight distribution that is used to weight the corresponding values to form the output. The mathematical formula can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

where Q, K, and V are the matrix representations of queries, keys, and values, respectively; d_k is the dimension of the key vectors, used to scale the dot product to prevent it from becoming too large. In computer vision, the attention mechanism can be used for image classification, object detection, and image segmentation tasks. For example, in image classification, attention can enable the network to focus on the most informative regions of the image for the classification task. For a convolutional neural network (CNN), attention weights α can be applied to the feature maps to enhance or suppress features in certain areas, as follows:

$$F' = \alpha \odot F \tag{9}$$

Here, F is the original feature map, $\odot$ represents element-wise multiplication, and F' is the feature map after attention weighting. In NLP, the attention mechanism is widely used in machine translation, text summarization, question-answering systems, and other tasks. In machine translation, sequence-to-sequence (Seq2Seq) models often use an encoder–decoder architecture, where the attention mechanism helps the decoder focus on the parts of the encoder that are most relevant to the word currently being generated. If h_t represents the hidden state of the decoder at time step t, and s_i represents the hidden state sequence of the encoder, attention weights can be calculated as:

$$\alpha_{t,i} = \frac{\exp(\text{score}(h_t, s_i))}{\sum_j \exp(\text{score}(h_t, s_j))} \tag{10}$$

Then, the current hidden state of the decoder can be updated using the weighted sum of the encoder's hidden states:

$$h'_t = \sum_i \alpha_{t,i} s_i \tag{11}$$

In the task of this paper, namely a privacy-protected dialogue large model framework based on FHE and attention mechanisms, the attention mechanism can help the model focus computational resources on the most critical information for the current task when processing encrypted data. If c_{h_i} and c_{h_j} are the homomorphically encrypted hidden states, it is necessary to calculate their homomorphic encrypted attention weights $c_{\alpha_{ij}}$, which can be achieved through the homomorphic encryption version of the dot product and normalization functions. In a homomorphic environment, this computation process requires special design because the traditional softmax function involves exponentiation and division operations, which are usually not directly feasible on homomorphic encryption. Therefore, it may be necessary to apply polynomial approximation or other numerical techniques to implement this:

$$c_{\alpha_{ij}} = \text{HomoSoftmax}\Big(\text{Enc}_{pk}(\text{score}(c_{h_i}, c_{h_j}))\Big) \tag{12}$$

Here, HomoSoftmax represents the homomorphic version of the softmax function that has been specially designed to operate on homomorphically encrypted data.

2.3. Transformer

The transformer model, introduced by Vaswani et al. [4] in 2017, has precipitated a revolutionary shift in the field of NLP. This architecture abandons the traditional recurrent neural network (RNN) [27,28] and convolutional neural network (CNN) [29–31] frameworks, being constructed entirely on the attention mechanism, which facilitates more efficient processing of sequence data and better captures long-distance dependencies, as shown in Figure 2.

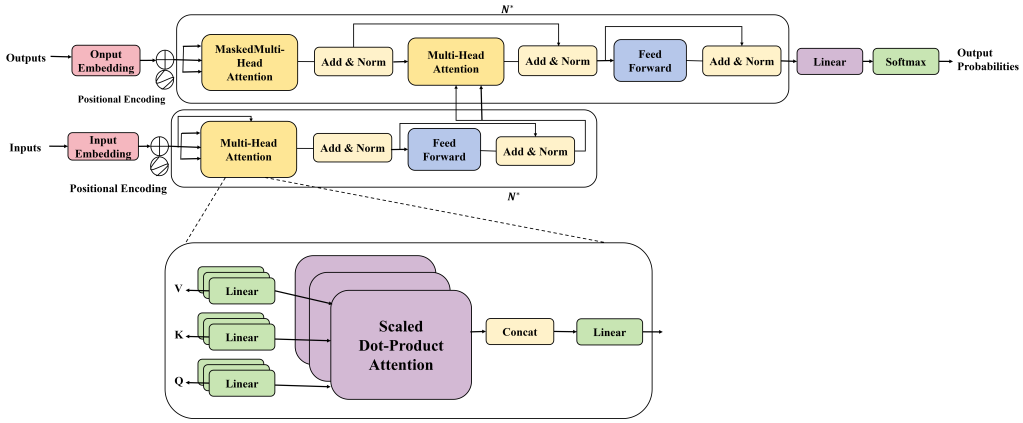

Figure 2. Illustration of the transformer architecture.

The core concept of the transformer is the parallel processing of all elements in a sequence, a stark departure from the sequential processing required by RNNs. It consists of an encoder and a decoder, each comprising multiple identical layers, with each layer containing two primary sub-layers: the multi-head attention mechanism and the position-wise feed-forward networks. Mathematically, the overall structure of the transformer can be described by the following formula:

$$\begin{aligned}\text{Transformer}(Q,K,V) = \text{LayerNorm}(\text{FFN}(\text{LayerNorm}(\text{Attention}(Q,K,V)+Q)) \\ + \text{Attention}(Q,K,V))\end{aligned} \tag{13}$$

Here, $\text{Attention}(Q,K,V)$ denotes the output of the multi-head attention mechanism, FFN represents the position-wise feed-forward networks, and LayerNorm refers to the layer normalization operation.

The encoder is composed of N identical layers, each with two sub-layers. The first sub-layer is the multi-head attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. For each sub-layer, the transformer employs a residual connection, followed by layer normalization. This means that the output of each sub-layer is $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ is the operation of the sub-layer itself. The multi-head attention mechanism allows the model to focus on different parts of the input at different positions. Specifically, multi-head attention maps Q, K, and V to the dimensions d_k, d_k, and d_v, respectively, h times, then performs independent attention function computations on each mapped triplet, concatenates the outputs of all heads, and performs a final linear mapping to produce the final output.

Due to its superior performance and flexibility, the transformer model has achieved remarkable success in NLP tasks and has been applied to various language models and tasks, such as BERT and GPT. In this work, the exploration focuses on integrating the transformer architecture with Homomorphic Encryption technology to provide high-quality conversational generation services while preserving user privacy. The challenge lies in designing an effective encryption-compatible transformer model that ensures computational efficiency and model performance do not suffer significantly due to encryption operations.

3. Materials and Methods

3.1. Dataset Collection

In the pursuit of developing a privacy-preserving dialogue model framework based on FHE and attention mechanisms, the initial step involves the aggregation of one or more datasets for the purpose of training and evaluation. The selection of datasets must be closely aligned with the research objectives, necessitating not only a sufficient volume of data to ensure the learning of complex linguistic patterns but also high-quality annotations to guarantee effective model training. The sources of datasets in this study are primarily categorized as follows:

1. Public datasets: **Cornell Movie-Dialogs Corpus**: Comprising over 200,000 conversations extracted from 617 movies, this dataset includes a range of topics from everyday dialogues to emotional expressions and context-specific communications. It is utilized to test the model's capability in processing everyday conversations and emotional understanding. **Ubuntu Dialogue Corpus**: Focused on technical support dialogues, this dataset contains approximately one million conversations from the Ubuntu forums. Its technical nature makes it ideal for assessing the model's performance in handling technical terminologies and complex queries. **Stanford Question Answering Dataset (SQuAD)**: A reading comprehension-based question-answering dataset, SQuAD's questions are formulated based on Wikipedia articles. It tests the model's ability to understand and respond to specific questions based on given text passages. **Twitter Customer Support Dataset**: Comprising customer service dialogues on Twitter across various industries and topics, this dataset is used to evaluate the model's performance in handling real-time, informal interactions and customer service inquiries. **Medical Dialogue Dataset**: Focused on medical consultations, this dataset contains dialogues between doctors and patients, allowing for an assessment of the model's accuracy and privacy protection capabilities in handling professional medical information and sensitive health data.
2. Synthetic data: To enhance the diversity of data, it might also be necessary to generate synthetic data using natural language processing techniques. Such data can assist the model in learning dialogue patterns specific to certain scenarios.

3.2. Dataset Preprocessing

Data preprocessing represents a crucial step within the machine learning workflow, involving the transformation of raw data into a format comprehensible by models. In this study, preprocessing encompasses the following steps: (1) Data cleaning involves the removal of noise and irrelevant information from the dataset, such as eliminating HTML tags, URLs, user tags, and extraneous punctuation marks. (2) Text normalization converts the text into a uniform format, including transforming all letters to lowercase, expanding abbreviations to their full forms, and replacing special characters. (3) Tokenization processes the text by breaking down continuous strings into meaningful units (words, punctuation). (4) Vocabulary construction entails building a vocabulary from the tokenized results, where each word is assigned a unique index. (5) Text encoding transforms the text into a numerical form that can be processed by the model, usually by replacing each word with its corresponding index or word vector. (6) Sequence padding or truncation ensures that all sequences are of consistent length to enable the model to process inputs of varying lengths. (7) Construction of homomorphic encryption-compatible formats: given the study's involvement with homomorphic encryption, the numerically formatted text must be further transformed into a format suitable for homomorphic encryption operations.

The mathematical principles underlying preprocessing are primarily concerned with text encoding and sequence handling. During text encoding, word embeddings transform discrete words into points in a continuous vector space, capturing semantic relationships between words. Mathematically, word embeddings can be realized through an embedding matrix E, where each row of E corresponds to a word in the vocabulary. For a word w, its encoded vector is obtained by looking up the embedding matrix:

$$v_w = E_{\text{index}(w)} \tag{14}$$

Sequence padding or truncation involves adjusting all text sequences to the same length L. For sequences shorter than L, padding tokens (such as 0) are added to the end to reach length L; for sequences longer than L, the excess is truncated. Let S be the original sequence and S' the processed sequence, then this is mathematically expressed as:

$$S' = \begin{cases} S \oplus [0]_{L-|S|}, & \text{if } |S| < L \\ S[:L], & \text{if } |S| \geq L \end{cases} \tag{15}$$

where $\oplus$ represents the sequence concatenation operation, and $[0]_{L-|S|}$ denotes a padding sequence of length $L - |S|$.

Preprocessing is essential for the tasks addressed in this study, aimed at enhancing model performance and generalizability. The importance of preprocessing is elucidated as follows:

1. Data Quality: High-quality data are pivotal to the performance of models. By cleaning and normalizing text, noise that could confuse the model is removed, thus improving data quality.
2. Computational Efficiency: By padding and truncating sequences, a uniform shape for all data during computation is ensured, which is beneficial for parallel processing by hardware such as GPUs and simplifies model design.
3. Model Training: Proper preprocessing can enhance the efficiency and stability of model training. For instance, uniform text encoding and length can accelerate the convergence of gradient descent.
4. Homomorphic Encryption Compatibility: In the application of homomorphic encryption, preprocessing data into a compatible format is a prerequisite for encrypted computations. Due to the constraints typically associated with homomorphic encryption operations, preprocessing steps must be specially designed to accommodate these constraints, thereby ensuring the model functions correctly on encrypted data.

In summary, data preprocessing not only impacts the training efficiency and final performance of the model but also plays a key role in protecting user privacy. In subsequent sections, a detailed exposition on how the preprocessed data are applied to the homomorphic encryption-based dialogue model will be provided, exploring its performance and potential in practical applications.

3.3. Proposed Method

3.3.1. Overview

A privacy-preserving dialogue model framework that integrates FHE with a dynamic sparse attention mechanism is presented in this paper. This framework, based on the transformer architecture, is specifically designed to perform complex natural language processing tasks on encrypted data without revealing the content of user data, as shown in Figure 3. Due to the reliance of traditional transformer models on linear and nonlinear operations over real numbers, they are not directly applicable to homomorphically encrypted data. Consequently, modifications have been made to the internal operators of the model. Specifically, linear transformations in the model are replaced with homomorphic linear transformations, employing integer approximation and quantization techniques to convert weights into a format operable within the homomorphic encryption environment. Activation functions are substituted with homomorphic-compatible versions that can be approximated by polynomials or piecewise linear functions. Additionally, special homomorphic batch normalization and integer optimization algorithms are introduced to accommodate the training and inference of the model on encrypted data.

To enhance computational efficiency under FHE, a dynamic sparse attention mechanism is adopted. Unlike traditional attention mechanisms, which compute attention scores for all elements in a sequence, the dynamic sparse attention mechanism dynamically selects a subset of elements deemed most important for the current task based on predefined strategies. This approach significantly reduces the amount of computation in an encrypted environment and aids the model in focusing on processing information, particularly in the handling of long sequences, effectively mitigating the degradation of model performance. Within the overall workflow, user input data are first homomorphically encrypted locally and then sent to a server for encrypted inference computation. The server-side model executes inference tasks that include homomorphic operator replacements and returns encrypted results, which are then sent back to the client and decrypted locally, allowing the user to receive plaintext output results without concerns of privacy breach.

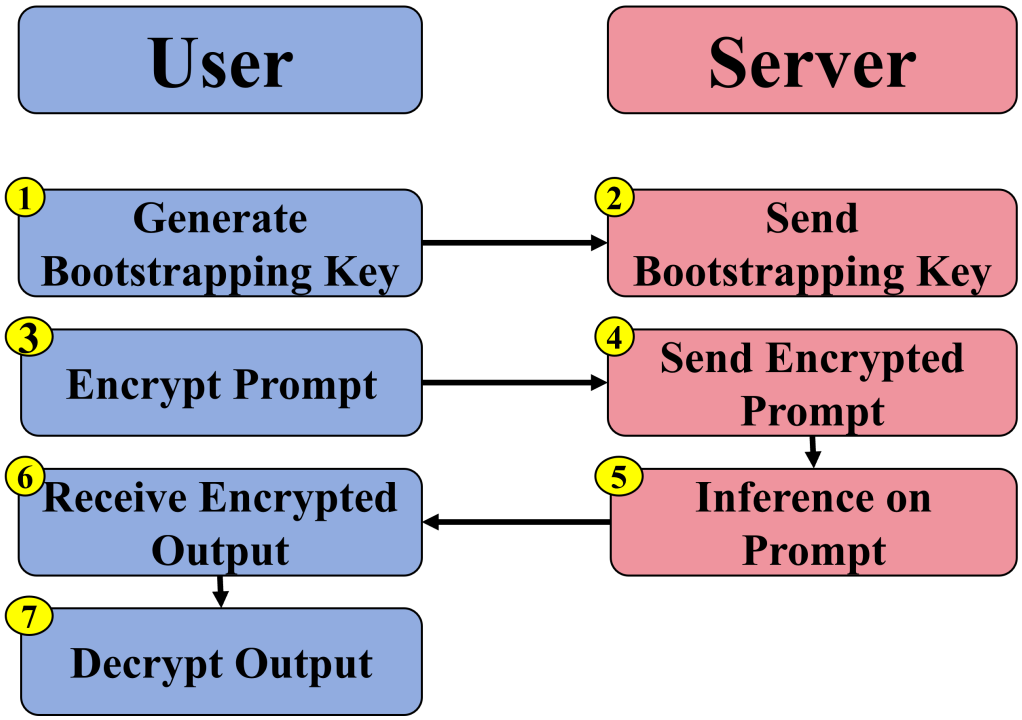

Figure 3. Overview of the privacy-preserving framework proposed in this paper. The steps are as follows: (1) A bootstrapping key is generated by the user. (2) The bootstrapping key is received by the server, which is utilized for subsequent encrypted data processing. (3) The data are encrypted by the user utilizing the bootstrapping key, resulting in an encrypted prompt. (4) The encrypted data are transmitted to the server. (5) Inference on the prompt is conducted by the server with the encrypted data. (6) The server encrypts the outcome of the processing before sending it back to the user. (7) The encrypted result is received by the user, who then decrypts it using the bootstrapping key to obtain the final plaintext output.

In summary, the method described in this paper offers a novel solution to the privacy issues faced by large-scale dialogue models when processing sensitive data. By incorporating operator replacements and a dynamic sparse attention mechanism on the foundation of homomorphic encryption technology, our model maintains privacy protection while minimizing computational costs and sustaining performance, demonstrating potential for complex natural language processing tasks with secure privacy guarantees.

3.3.2. Operator Replacement Technique

The capability of homomorphic encryption to perform computations on encrypted data without decryption enables models to calculate while preserving data privacy. The detailed implementation of operator replacement is expounded herein, providing corresponding mathematical proofs and formulas, and elucidating the advantages of this design. Traditional transformer models contain several key operators, such as linear transformations, activation functions, and layer normalization, typically operating in the real number domain. To realize a transformer model in an FHE environment, these operators must be replaced with versions compatible with homomorphic operations.

Homomorphic Linear Transformation: In the transformer model, linear transformations are executed by multiplying a weight matrix with an input vector. In the homomorphic encryption setting, weight matrices and input vectors are converted to integers via quantization techniques, followed by the application of addition and multiplication from the homomorphic encryption algorithm to effectuate linear transformations. The specific replacement technique is mathematically expressed as:

$$\mathrm{Enc}(Wx + b) = \mathrm{Enc}(W) \cdot \mathrm{Enc}(x) + \mathrm{Enc}(b) \tag{16}$$

where W represents the weight matrix, x represents the input vector, b represents the bias term, and Enc() denotes the homomorphic encryption function. The homomorphic encryption's additive and multiplicative operations ensure that the outcome's encrypted form is equivalent to directly computing $Wx + b$ in plaintext.

Homomorphic Activation Function: Activation functions are typically nonlinear, such as the ReLU or Sigmoid functions. In a homomorphic encryption context, implementing nonlinear functions presents challenges, thus, piecewise linear functions are employed as approximations. For instance, the ReLU function can be approximated by the following piecewise linear function:

$$\mathrm{ReLU}(x) \approx \max(0, x) \approx \begin{cases} 0 & \text{if } \mathrm{Enc}(x) < 0 \\ \mathrm{Enc}(x) & \text{if } \mathrm{Enc}(x) \geq 0 \end{cases} \tag{17}$$

The comparison operation here can be realized through comparison protocols supported by FHE.

Homomorphic Normalization: Layer normalization plays a pivotal role in stabilizing training and accelerating convergence in the transformer model. Implementing layer normalization in an FHE context necessitates the computation of mean and variance that is supported by homomorphic encryption. Mean and variance of encrypted data can be calculated using homomorphic encryption algorithms, which are then utilized for normalization. Mathematically, homomorphic layer normalization is depicted as:

$$\mathrm{Enc}(\hat{x}) = \frac{\mathrm{Enc}(x) - \mathrm{Enc}(\mu)}{\sqrt{\mathrm{Enc}(\sigma^2) + \epsilon}} \tag{18}$$

where μ and σ^2 are the mean and variance of the encrypted data x, respectively, ϵ is a small constant added for numerical stability, and $\hat{x}$ is the normalized data.

Advantages and Applications of the Design: The design of the above operator replacement techniques enables the transformer model to be trained and inferred without any knowledge of the data content, thereby offering robust privacy protection for users, as shown in Figure 4.

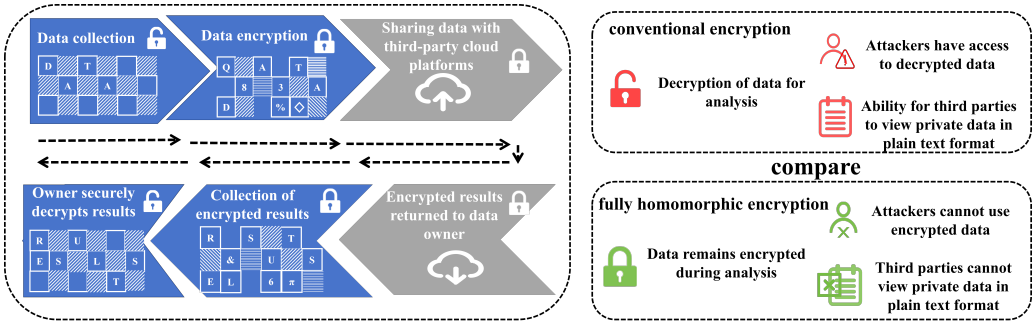

Figure 4. Advantages of the FHE-based encryption framework.

Mathematically, the homomorphic encryption algorithm ensures that the computations on encrypted data are logically equivalent to those performed in plaintext, meaning that the model's output will not be biased due to encryption. This operator replacement technique has the following advantages:

1. Privacy Protection: Since all computations are conducted in the encrypted domain, user data are never exposed to the model provider in any form, ensuring the privacy of user data.
2. Security: The security of the homomorphic encryption algorithm is based on mathematically hard problems, such as factoring large numbers or learning with errors, making it difficult for attackers to decrypt the data with current computing capabilities.
3. Model Performance: Through carefully designed operator replacement and approximation techniques, privacy is ensured while minimizing the impact on model performance.

4. Compatibility: The operator replacement technique is compatible with existing homomorphic encryption algorithms and can be integrated with the latest FHE optimization technologies to further enhance computational efficiency.
5. Scalability: This method is not only applicable to dialogue systems but can also be extended to other natural language processing tasks that require privacy protection.

With rigorous mathematical design and proof, the feasibility of performing complex natural language processing tasks on encrypted data are demonstrated, providing robust technical support for protecting user privacy.

3.3.3. Privacy-Preserving Transformer Framework

A novel transformer framework has been designed to provide privacy-preserving capabilities, primarily through the integration of FHE technology, as shown in Figure 5. The core idea of this framework is to ensure the security of data throughout the processing chain, from input to model processing, and finally to output, with every step completed in an encrypted state. This section details the main differences between our privacy-preserving transformer framework and the original transformer model, as well as the specific details of the implementation.

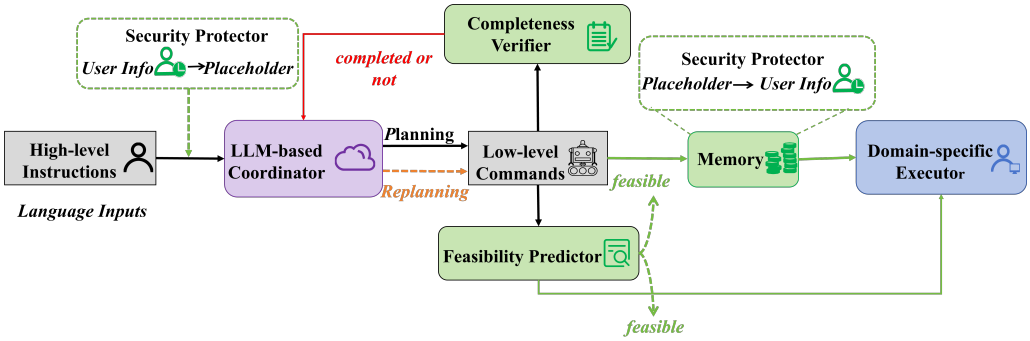

Figure 5. The privacy-preserving transformer framework proposed in this paper.

Key Differences in Model Architecture: The original transformer model is based on the self-attention mechanism, capturing internal dependencies by computing attention scores across all pairs of words in the input sequence. This process involves extensive real-number computations, which are infeasible in the context of FHE. Consequently, in our privacy-preserving transformer framework, the original computational flow has been adjusted to only use operations supported by homomorphic encryption algorithms.

1. Adjustment of Self-Attention Mechanism: Within the homomorphic encryption environment, it is not possible to directly compute the softmax function, a key component of the self-attention mechanism in the original transformer model. Therefore, a method for calculating attention scores based on FHE has been designed, utilizing algorithms compatible with homomorphic encryption to approximate the softmax function.
2. Encryption of Weights and Biases: In the privacy-preserving transformer framework, all model weights and biases are pre-encrypted. Hence, the model does not need to access any plaintext information before performing any calculations.
3. Protection of Intermediate States: At every layer of the model, all intermediate states exist in encrypted form. This ensures that data privacy is protected, even within a multi-layer network structure.

Input, Output, and Processing Flow: Within our privacy-preserving transformer framework, a user's input is first encrypted using a homomorphic encryption algorithm and then fed into the model in an encrypted state. The model performs encrypted linear transformations, encrypted activation functions, and encrypted attention mechanisms on the encrypted input and produces an encrypted output. Finally, the encrypted output is returned to the user, who can decrypt it locally with a private key to obtain the final plaintext result.

Mathematically, suppose the user's original input is a vector x, the model's encrypted weight matrix is $\text{Enc}(W)$, the encrypted bias is $\text{Enc}(b)$, and the encrypted output is $\text{Enc}(y)$, then the processing flow can be expressed as:

$$\text{Enc}(y) = \text{Model Processing}(\text{Enc}(W), \text{Enc}(x), \text{Enc}(b)) \tag{19}$$

where "Model Processing" includes encrypted linear transformations, encrypted activation functions, and encrypted attention mechanisms.

3.3.4. Dynamic Sparse Attention Module

In natural language processing, the attention mechanism has become a key component for enhancing model performance, especially within the transformer model's multi-head attention mechanism. However, the computational cost becomes exceedingly high when dealing with large datasets or long sequences, which is particularly pronounced in the context of homomorphic encryption. To address this, a dynamic sparse attention (DSA) module is introduced to reduce computational demands while maintaining or even improving model performance. This chapter will detail the differences between the DSA module and the original multi-head attention mechanism, its design details, and its advantages when applied to the tasks addressed in this paper.

The core idea of the DSA module is to reduce the consumption of computational resources while preserving the essential functions of the attention mechanism by dynamically selecting key parts of the sequence. In the traditional multi-head attention mechanism, attention scores are computed at every position in the sequence, leading to a computational complexity of $O(n^2)$. The DSA module, through a predefined strategy such as gradient-based or task-relevance metrics, selects the most critical information to reduce computational volume. To illustrate this, suppose there is a sequence of length n, where the computational complexity is $O(n^2)$ in the traditional attention mechanism. In our dynamic sparse model, if only k most important positions are selected for attention score computation, the computational complexity can be reduced to $O(nk)$, where $k \ll n$. This process can be mathematically formulated as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{20}$$

In the dynamic sparse model, the above formula is modified to:

$$\text{SparseAttention}(Q, K, V) = \text{Softmax}\left(\frac{Q\tilde{K}^T}{\sqrt{d_k}}\right)\tilde{V} \tag{21}$$

where $\tilde{K}$ and $\tilde{V}$ are the k most important keys and values selected from the original K and V based on some strategy. The primary reason for designing the DSA module is to address the computational challenges faced when processing large-scale natural language tasks in a homomorphic encryption environment. Homomorphic encryption itself introduces additional computational overhead, and using the computationally expensive standard multi-head attention mechanism would make the model impractical for real-world scenarios. By implementing sparsity, our model significantly reduces computational costs while maintaining focus on the most important information in the sequence.

3.4. Experiment Configuration

3.4.1. Experimental Evaluation Metrics

When evaluating the privacy-preserving transformer framework based on FHE and attention mechanisms, key performance indicators include accuracy, response time, and computational efficiency. These metrics are crucial for understanding the practicality, efficiency, and security of the model. Below, each evaluation metric's mathematical definition and its importance are described in detail. Accuracy is typically measured by three metrics: precision, recall, and accuracy.

Precision: Precision refers to the proportion of true positives among the samples predicted as positive by the model. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

where TP represents true positives, the number of samples correctly predicted as positive by the model; FP represents false positives, the number of samples incorrectly predicted as positive.

Recall: Recall indicates the proportion of actual positive samples that are correctly predicted as positive by the model. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

where FN represents false negatives, the number of samples incorrectly predicted as negative by the model.

Accuracy: Accuracy refers to the proportion of samples correctly predicted by the model (regardless of positive or negative) out of the total number of samples. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{24}$$

where TN represents true negatives, the number of samples correctly predicted as negative by the model.

Latency: Latency measures the time taken for the model to respond to a request, which is vital for user experience. In a homomorphic encryption environment, maintaining reasonable latency is challenging due to increased computational complexity. Latency is typically expressed as the average processing time for individual requests:

$$\text{Latency} = \frac{1}{N} \sum_{i=1}^{N} t_i \tag{25}$$

where N is the total number of requests, and t_i is the time taken to process the i-th request.

In experiments, these evaluation metrics will be used to compare the performance of our privacy-preserving model framework with other baseline models. Through the comparison of these quantitative metrics, the advantages and potential areas for improvement of the proposed model in this paper can be objectively demonstrated. Furthermore, they assist in understanding the model's performance in real-world applications and how the model can be further optimized to meet practical needs.

3.4.2. Experiment Design

In this study, the experimental design plays a critical role in verifying the effectiveness of the proposed model. The research commences with a widely-recognized large dialogue dataset, from which data are randomly sampled and divided into three parts: 70% for training the model, which serve as the foundation for the model's learning; 15% for model validation, which are used to adjust hyperparameters and prevent overfitting during the model development process; and the remaining 15% for the test set, which are utilized to assess the model's final performance. This division strategy is intended to ensure that the model can make reasonable predictions on unseen data, thereby verifying the model's generalization capability.

Five different baseline models were selected for comparison in the experiments: original transformer model, the traditional trusted execution environment (TEE) model, SOTER as a secure multi-party computation framework, the differential privacy (DP) model for measuring the strength of privacy protection, and a baseline model based on FHE. These models were chosen as baselines because they each represent different directions and the latest advancements in privacy protection technology. TEE provides an isolated execution environment but relies on the security of the hardware; SOTER protects the security of

data in use through multi-party computation but may introduce additional communication overhead; DP is a commonly used technology in current privacy protection, it protects personal information by adding noise, but may sacrifice the utility of the data in some cases; the FHE model offers a powerful way to keep data encrypted while in use but comes with high computational costs. The comparison with these models allows for the demonstration of the relative advantages of the proposed model in terms of privacy protection and performance.

The choice of optimizer is crucial for the model's convergence speed and final performance during the training process. The Adam optimizer was selected because it combines the benefits of momentum and adaptive learning rates, enabling the model to automatically adjust the learning rate at different stages of training, thus accelerating the convergence process and enhancing the stability of the model. After a series of preliminary experiments and parameter tuning, it was found that setting the learning rate to 10^{-4}, batch size to 32, and weight decay to 10^{-5} allowed the model to achieve optimal performance. The selection of these hyperparameters strikes a balance between training speed and convergence quality, ensuring that the model can learn effectively at different stages.

4. Results and Discussion

4.1. Comparison Results

The design of this experiment aims to assess the impact of various models on the performance of natural language processing tasks while maintaining privacy protection. Through the evaluation of four key metrics—precision, recall, accuracy, and latency—a comprehensive understanding of the advantages and disadvantages of each technology in terms of data security and processing efficiency is ascertained. The experimental results are presented in Table 1.

Table 1. Model performance comparison.

Model	Security	Precision	Recall	Accuracy	Latency (ms)
Transformer [4] (baseline)	✗	0.92	0.91	0.92	0.13
FHE (baseline) [32]	✓	0.92	0.91	0.92	127.33
Proposed Method	✓	0.92	0.91	0.92	0.15
SOTER [33]	✓	0.91	0.91	0.90	0.27
TEE [34]	✓	0.92	0.91	0.92	0.33
DP [35]	◯ *	0.83	0.81	0.82	0.18

* DP can only protect partial privacy while training. DP can be adjusted by tuning parameters for the proportion of Gaussian noise, but it is still possible to restore the original data through methods such as denoising.

The transformer model, serving as a baseline, exhibits the best performance metrics, yet lacks privacy protection capabilities. The FHE model, while impeccable in terms of security, incurs a significant increase in latency due to high computational complexity, affecting its practicality. The proposed method retains the performance of the transformer while achieving comparable security to FHE with minimal latency, indicating the adoption of efficient encryption techniques. SOTER and TEE, while providing secure computation, experience performance constraints due to communication and hardware limitations. DP, although capable of protecting privacy to a certain extent, compromises a portion of performance.

From a mathematical perspective, the transformer model leverages parallel computation and multi-head attention mechanisms to optimize processing speed and performance in an unencrypted state. FHE ensures data remain encrypted throughout the computation process but introduces increased latency due to its complex mathematical operations. The proposed method likely integrates lightweight homomorphic encryption and model optimization, effectively balancing security and efficiency. SOTER and TEE, in ensuring secure computation, must contend with additional communication and hardware overhead. Differential privacy protects privacy by adding noise to the data but this approach can compromise the accuracy of the data and the performance of the model. The design of each

model reflects a trade-off between security, performance, and efficiency. The transformer model excels in performance but falls short in privacy protection; FHE ensures security at the cost of increased latency; the proposed method finds a mathematical balance point, achieving efficient privacy protection and high-performance processing.

4.2. Latency Overhead Analysis

This section delves into the latency overheads present in the proposed method, focusing on the implications of computation, communication, and encryption/decryption, as shown in Table 2. The aim is to scrutinize the factors influencing latency via mathematical and empirical methods, and to ascertain the efficiency of the proposed privacy-preserving method in natural language processing tasks.

Table 2. Latency Detail.

Model	Computational Latency	Communication Latency	Encryption/Decryption Latencyn	Total
Transformer [4]	0.13	0	0	0.13
FHE [32]	13.55	0	113.78	127.33
TEE [34]	0.18	0.12	0.03	0.33
Proposed Method	0.05	0	0.08	0.15
SOTER [33]	0.11	0.08	0.08	0.27
DP [35]	0.18	0	0	0.18

4.2.1. Computational Latency

The proposed method aims for seamless integration with the transformer architecture, leveraging its capability for parallel computation. The latency related to computation is primarily affected by the complexity of operations that must be performed in an encrypted state. Mathematically, the computational latency L_c can be expressed as:

$$L_c = O(n \cdot d^2 + m \cdot d) \tag{26}$$

where n represents the sequence length, d the dimensionality of the model, and m the number of operations per layer. The method optimizes this aspect through effective algorithmic modifications that are compatible with the FHE scheme while maintaining the core functionality of the transformer model.

4.2.2. Communication Latency

In distributed systems, where privacy-preserving models are often deployed, communication latency L_{comm} is a pivotal factor. It quantifies the time required to transmit encrypted data between a client and a server. Given a bandwidth B and data size S, the communication latency is provided by the following formula:

$$L_{comm} = \frac{S}{B} \tag{27}$$

The proposed method minimizes L_{comm} by utilizing a high-efficiency communication protocol, which reduces the amount of data to be transmitted without compromising privacy guarantees.

4.2.3. Encryption/Decryption Latency

When assessing the overall latency of privacy-preserving methods, encryption latency L_{enc}, and decryption latency L_{dec} are crucial. For a given security parameter λ, the encryption and decryption latencies can be articulated as:

$$L_{enc} = f(\lambda, n, d) \tag{28}$$

$$L_{dec} = g(\lambda, n, d) \tag{29}$$

where f and g are the complexity functions of encryption and decryption algorithms, respectively. The proposed method employs a lightweight encryption scheme to reduce L_{enc} and L_{dec}, thus facilitating fast and secure operations in line with the efficiency goals of the model.

Empirical results from the performance comparison table indicate that the proposed method generates a minimal latency of 0.15 milliseconds, only slightly higher than the 0.13 milliseconds of the unencrypted baseline transformer model, and significantly lower than the 127.33 milliseconds of the FHE method. This substantiates the effectiveness of the proposed optimizations in mitigating the overhead introduced by encryption operations, while preserving the computational efficiency of the underlying transformer architecture.

In summary, the proposed method provides a comprehensive solution to the challenges of latency faced by privacy-preserving models in NLP tasks. By achieving a balance between computational efficiency and security considerations, a practical implementation has been realized, facilitating the development of real-time applications without compromising user privacy.

4.3. Ablation Study on DSA Mechanism

The primary objective of the experiment was to evaluate the performance of the DSA module within privacy-preserving dialogue large models. By contrasting the conventional Multi-head Attention mechanism with the newly proposed DSA module, a quantitative analysis of the differences in key performance metrics such as precision, recall, accuracy, and latency was facilitated. Table 3 demonstrated that the DSA module surpassed the multi-head attention mechanism across precision, recall, and accuracy metrics, with only a minimal increase in latency. This indicates that the DSA mechanism, by reducing computational load, is capable of enhancing model performance while ensuring security, particularly excelling in tasks where precision and recall are paramount.

Table 3. Performance comparison on different attention mechanisms.

Model	Security	Precision	Recall	Accuracy	Latency (ms)
Multi-head Attention [4]	✗	0.89	0.87	0.89	0.13
Dynamic Sparse Attention	✓	0.92	0.91	0.92	0.15

From a mathematical perspective, the computational complexity of the multi-head attention mechanism is quadratic in relation to the length of the input sequence, as it computes attention scores for every pair of elements. Specifically, for a sequence of length n, the complexity is $O(n^2d)$, where d represents the model's dimension. This results in significant latency when processing large sequences. In contrast, the module optimizes the computation process by only calculating attention scores among key elements that have the most significant impact on the output. Its complexity is contingent on the number of key elements selected, typically much less than the length of the sequence, thus the complexity can be denoted as $O(knd)$, where k is the number of key elements. Given that k is usually much smaller than n, this markedly reduces both computation and latency.

In summary, the design of the module adeptly balances computational efficiency and performance, while maintaining the security of data privacy, this novel attention mechanism enhances the processing speed and responsiveness of the dialogue large model by minimizing unnecessary computations, thereby showcasing its potent potential and practicality, especially in scenarios requiring real-time feedback.

5. Conclusions

In this study, in response to the challenges of protecting personal privacy in dialogue-based large models in the information era, an innovative framework for privacy-preserving dialogue-based large models is presented, effectively integrating FHE and mechanisms. The primary task of the proposed framework is to enhance the responsiveness and accuracy of dialogue systems without compromising user privacy. The novelties of this

paper are manifested in the combination of FHE technology with dialogue-based large models, enabling the processing of user data without decryption. Thus, even if data leakage occurs during processing and transmission, sensitive user information is not exposed. The introduced module optimizes processing efficiency and computational resources compared to traditional multi-head attention mechanisms by only calculating the attention scores between key elements that significantly affect the output, thereby reducing unnecessary computations.

In the experimental section, a comparative analysis of different models on the performance impact of natural language processing tasks while protecting privacy yielded the following results:

1. While maintaining the performance of transformers, the proposed method achieved security comparable to FHE with minimal latency of 0.15 ms, demonstrating efficient encryption technology.
2. In Section 4.2, the analysis of latency overhead, the proposed method effectively controlled and optimized the latency in computation, communication, and encryption/decryption, with results of 0.05 ms, 0 ms, and 0.08 ms, respectively.
3. The ablation study, in Section 4.3, on the DSA module showed that it outperforms the multi-head attention mechanism in terms of precision, recall, and accuracy which are 0.92, 0.91, and 0.92, respectively, with a negligible increase in latency of 0.15 ms.

However, there are still shortcomings in this research. For instance, although the mechanism reduces computational load, it may miss important information when selecting key elements. Future research plans will unfold in several directions: 1. Optimizing encryption algorithms to further reduce latency and enhance performance. 2. Improving the mechanism to increase accuracy in selecting key elements, ensuring no crucial information is missed. 3. Cross-domain applications to demonstrate the utility and scalability of the framework in various fields, such as online education and legal consultation.

In conclusion, this paper provides a new direction for research on privacy-preserving dialogue-based large models and has empirically demonstrated the effectiveness of the proposed methods. Future studies will continue to delve deeper, refining the model to provide stronger security for user privacy and to promote the healthy development of the artificial intelligence field.

Author Contributions: Conceptualization, L.Z.; Methodology, L.Z. and S.H.; Software, L.Z., Q.H. and Q.C.; Validation, C.L. (Changxiang Li), S.H. and Q.C.; Formal analysis, C.L. (Changxiang Li), S.H. and L.H.; Investigation, L.H. and J.L. (Jingwen Leng); Resources, J.L. (Jingjing Lang) and Q.C.; Data curation, C.L. (Changxiang Li), Q.H., J.L. (Jingjing Lang), L.H. and J.L. (Jingwen Leng); Writing—original draft preparation, L.Z., C.L. (Changxiang Li), Q.H., J.L. (Jingjing Lang), S.H., L.H., J.L. (Jingwen Leng), Q.C. and C.L. (Chunli Lv); Writing—review and editing, C.L. (Chunli Lv); Visualization, Q.H. and J.L. (Jingjing Lang); Supervision, C.L. (Chunli Lv); Project administration, J.L. (Jingwen Leng) and C.L. (Chunli Lv); Funding acquisition, C.L. (Chunli Lv). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 61202479.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-accuracy detection of maize leaf diseases CNN based on multi-pathway activation function module. *Remote Sens.* **2021**, *13*, 4218 [CrossRef]
2. Lin, X.; Wa, S.; Zhang, Y.; Ma, Q. A dilated segmentation network with the morphological correction method in farming area image Series. *Remote Sens.* **2022**, *14*, 1771 [CrossRef]

3. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Lin, J.; Fan, D.; Fu, J.; Lv, C. Symmetry GAN Detection Network: An Automatic One-Stage High-Accuracy Detection Network for Various Types of Lesions on CT Images. *Symmetry* **2022**, *14*, 234 [CrossRef]
4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
6. Jain, N.; Vaidyanath, S.; Iyer, A.; Natarajan, N.; Parthasarathy, S.; Rajamani, S.; Sharma, R. Jigsaw: Large Language Models meet Program Synthesis. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering (ICSE), Pittsburgh, PA, USA, 22–27 May 2022; IEEE: Piscataway, NJ, USA ; Association for Computing Machinery: New York, NY, USA, 2022. [CrossRef]
7. Kurstjens, S.; Schipper, A.; Krabbe, J.; Kusters, R. Predicting hemoglobinopathies using ChatGPT. *Clin. Chem. Lab. Med.* **2023**, *103*, 9194–9218. [CrossRef]
8. Jowsey, T.; Stokes-Parish, J.; Singleton, R.; Todorovic, M. Medical education empowered by generative artificial intelligence large language models. *Trends Mol. Med.* **2023**, *29*, 971–973. [CrossRef]
9. Leippold, M. Thus, spoke GPT-3: Interviewing a large-language model on climate finance. *Financ. Res. Lett.* **2023**, *53*, 103617. [CrossRef]
10. Zhong, Y.; Chen, Y.; Zhou, Y.; Lyu, Y.A.H.; Yin, J.J.; Gao, Y.j. The Artificial intelligence large language models and neuropsychiatry practice and research ethic. *Asian J. Psychiatry* **2023**, *84*, 103577. [CrossRef]
11. Sorin, V.; Barash, Y.; Konen, E.; Klang, E. Large language models for oncological applications. *J. Cancer Res. Clin. Oncol.* **2023**, *149*, 9505–9508. [CrossRef]
12. Hua, B.; Wang, Z.; Meng, J.; Xi, H.; Qi, R. Big data security and privacy protection model based on image encryption algorithm. *Soft Comput.* **2023**, *45*, 829–845. [CrossRef]
13. Wei, Y.C.; Wu, W.C.; Lai, G.H.; Chu, Y.C. pISRA: Privacy considered information security risk assessment model. *J. Supercomput.* **2020**, *76*, 1468–1481. [CrossRef]
14. Vu, D.H.; Luong, T.D.; Ho, T.B. An efficient approach for secure multi-party computation without authenticated channel. *Inf. Sci.* **2020**, *527*, 356–368. [CrossRef]
15. Zhou, J.; Feng, Y.; Wang, Z.; Guo, D. Using Secure Multi-Party Computation to Protect Privacy on a Permissioned Blockchain. *Sensors* **2021**, *21*, 1540. [CrossRef]
16. Olakanmi, O.O.; Odeyemi, K.O. Trust-aware and incentive-based offloading scheme for secure multi-party computation in Internet of Things. *Internet Things* **2022**, *19*, 100527. [CrossRef]
17. Kim, J.; Yun, A. Secure Fully Homomorphic Authenticated Encryption. *IEEE Access* **2021**, *9*, 107279–107297. [CrossRef]
18. Xu, W.; Wang, B.; Hu, Y.; Duan, P.; Zhang, B.; Liu, M. Multi-key Fully Homomorphic Encryption from Additive Homomorphism. *Comput. J.* **2023**, *66*, 197–207. [CrossRef]
19. Zhang, Y.; Shang, T.; Liu, J. A multi-valued quantum fully homomorphic encryption scheme. *Quantum Inf. Process.* **2021**, *20*, 101. [CrossRef]
20. Yagisawa, M. IND-CCA1 Secure FHE on Non-Associative Ring. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2021**, *E104A*, 275–282. [CrossRef]
21. Cai, Y.; Tang, C.; Xu, Q. Two-Party Privacy-Preserving Set Intersection with FHE. *Entropy* **2020**, *22*, 1339. [CrossRef]
22. Peng, Z.; Zhou, W.; Zhu, X.; Wu, Y.; Wen, S. On the security of fully homomorphic encryption for data privacy in Internet of Things. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7330. [CrossRef]
23. Menon, S.J.; Wu, D.J. Spiral: Fast, high-rate single-server PIR via FHE composition. In Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 930–947.
24. Bonte, C.; Iliashenko, I.; Park, J.; Pereira, H.V.; Smart, N.P. Final: Faster fhe instantiated with ntru and lwe. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–9 December 2022; Springer: Cham, Switzerland, 2022; pp. 188–215.
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Wang, J.; Li, X.; Li, J.; Sun, Q.; Wang, H. NGCU: A new RNN model for time-series data prediction. *Big Data Res.* **2022**, *27*, 100296 [CrossRef]
28. Chen, J.; Zhang, Y.; Wu, J.; Cheng, W.; Zhu, Q. SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output. *Energy* **2023**, *262*, 125375 [CrossRef]
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105 [CrossRef]
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.

32. Chen, T.; Bao, H.; Huang, S.; Dong, L.; Jiao, B.; Jiang, D.; Zhou, H.; Li, J.; Wei, F. The-x: Privacy-preserving transformer inference with homomorphic encryption. *arXiv* **2022**, arXiv:2206.00216.
33. Shen, T.; Qi, J.; Jiang, J.; Wang, X.; Wen, S.; Chen, X.; Zhao, S.; Wang, S.; Chen, L.; Luo, X.; et al. SOTER: Guarding Black-box Inference for General Neural Networks at the Edge. In Proceedings of the 2022 USENIX Annual Technical Conference (USENIX ATC 22), Carlsbad, CA, USA, 11–13 July 2022; pp. 723–738.
34. Wang, Y.; Rajat, R.; Annavaram, M. MPC-Pipe: An Efficient Pipeline Scheme for Secure Multi-party Machine Learning Inference. *arXiv* **2022**, arXiv:2209.13643.
35. Zhang, Y.; Li, J.; Liu, D.; Chen, G.; Dou, J. DP-transformer: A distilling and probsparse self-attention rockburst prediction method. *Energies* **2022**, *15*, 3959 [CrossRef]