Encrypted Large Language Model Inference

contact@zionfhe.ai

May 2025

Contents

1 Introduction **3**

2 Research Background and Related Work **4**

3 Technical Background **5**

3.1 Large Language Models and Llama3.2-1B 5

3.2 Fully Homomorphic Encryption 5

3.3 Ciphertext Inference Requirements 6

4 FHE Operator Transformation and Inference **7**

4.1 Operator Transformation 7

4.2 Inference Workflow 8

4.2.1 Ideal End-to-End Ciphertext Inference Workflow 8

4.2.2 Current Simplified Workflow 9

5 Experiments and Results **10**

5.1 Experimental Setup 10

5.1.1 Server Configuration 10

5.1.2 Operator Latency Testing 11

5.1.3 Model Inference Testing 11

5.1.4 Baseline 11

5.1.5 Evaluation Metrics 11

5.2 Experimental Results 11

5.2.1 Operator Latency 11

5.2.2 Inference Efficiency 12

6 Discussion **12**

6.1 Key Findings 12

6.2 Limitations 13
6.3 Future Work 14

7 Conclusion **14**

Abstract

We propose FHE-Llama3.2-1B, a privacy-preserving large language model inference framework that integrates ZFHE Fully Homomorphic Encryption (FHE) based on coefficient mapping transformation with the Llama3.2-1B model to enable encrypted dialogue and ciphertext inference. The current approach reuses plaintext tokenization, encrypting plaintext vectors for ciphertext inference. Experiments demonstrate an inference latency of approximately 7.23 tokens per second, about 2.38 times that of the plaintext model. The core innovation lies in the comprehensive ciphertext adaptation of Transformer operators, including arithmetic operations, layer normalization, matrix multiplication, softmax, silu, and relu, and using plaintext weights to accelerate reasoning. This report details the research background, operator transformation, inference workflow, experimental comparisons (operator latency and inference efficiency), and technical significance, validating the potential of ciphertext inference in privacy-sensitive scenarios and outlining future work on end-to-end ciphertext processing and GPU optimization.

1 Introduction

Large language models (LLMs), leveraging the powerful Transformer architecture, have achieved remarkable success in tasks such as dialogue generation and question answering(1). However, their widespread adoption, particularly in privacy-sensitive domains like finance and healthcare, poses significant data privacy challenges. Traditional plaintext inference risks exposing input and output data, necessitating end-to-end encryption solutions. Fully Homomorphic Encryption (FHE) offers a theoretical foundation for privacy-preserving dialogue and inference by enabling arbitrary computations on encrypted data. However, the high computational complexity of FHE and the adaptation challenges of diverse Transformer operators limit its application in lightweight models, especially when maintaining inference performance on resource-constrained devices.

This paper introduces FHE-Llama3.2-1B, an encrypted dialogue and ciphertext inference system combining ZFHE (based on polynomial encryption) with the Llama3.2-1B model. Llama3.2-1B, a lightweight decoder-only Transformer model with 1 billion parameters, is optimized for efficient performance and suitable for edge device deployment, but its plaintext inference cannot meet privacy requirements. FHE-Llama3.2-1B addresses this by comprehensively transforming Transformer operators, including arithmetic operations, layer normalization (sum, mean, var, sqrt, rsqrt), matrix multiplication (for embeddings and linear layers), softmax, silu, and relu, enabling efficient execution in an FHE environment while using plaintext weights to accelerate reasoning. Experiments show that ciphertext inference achieves a latency of approximately 7.23 tokens per second, about 2.38 times that of the plaintext model, demonstrating the framework's efficiency and privacy-preserving capabilities on lightweight devices.

Key contributions include: (1) FHE adaptation for all core operators of Llama3.2-1B, cov-

ering comprehensive implementation and optimization from basic operations to nonlinear activations; (2) implementation of a ciphertext inference workflow based on plaintext tokenization, supporting encrypted dialogue generation; (3) performance validation on lightweight hardware (Mac mini, M4 Pro chip, CPU-only), with latency only 2.38 times that of plaintext. The framework development comprises three phases: (1) FHE adaptation and optimization of Transformer operators; (2) implementation of the FHE inference workflow; (3) experimental evaluation of operator performance and inference efficiency. This report elaborates on the technical background, operator transformation, inference workflow, experimental results, discussion, and conclusions, providing a technical reference for privacy-preserving lightweight inference systems.

2 Research Background and Related Work

Fully Homomorphic Encryption (FHE), introduced by Gentry(2), has sparked extensive research in privacy-preserving computation, particularly for its potential in neural network inference. Early FHE schemes, such as BFV and CKKS(1), support integer and approximate floating-point operations, providing a foundation for encrypted matrix computations. However, they face challenges with high latency and memory overhead when handling nonlinear operators (e.g., softmax, relu) and high-dimensional Transformer models. CryptoNets(3) pioneered FHE application in convolutional neural networks, demonstrating the feasibility of encrypted inference, but its performance was insufficient for complex language models.

In recent years, privacy-preserving large language model (LLM) research has gained traction, focusing on inference over encrypted data. nGraph-HE(4) proposed an FHE-based deep learning framework supporting simple neural network operators, but it exhibits low efficiency for Transformer models' multi-head self-attention and nonlinear activations. Secure Multi-Party Computation (MPC) combined with FHE(5) attempts to reduce single-node overhead through distributed computing, but high communication costs make it unsuitable for edge devices. Additionally, some studies explore differential privacy and federated learning for LLM data protection, but these cannot achieve end-to-end encrypted inference.

FHE application in LLMs faces three major challenges: (1) high computational complexity of nonlinear operators; (2) memory demands of high-dimensional matrix operations; (3) performance bottlenecks on lightweight devices. Existing solutions primarily target large-scale servers, overlooking edge device deployment needs. In contrast, the FHE-Llama3.2-1B framework leverages ZFHE's coefficient mapping transformation, chaotic randomness, and operation dictionary mechanisms to significantly reduce ciphertext computation overhead. It optimizes operators and inference workflows for the lightweight Llama3.2-1B model (1 billion parameters). The current approach, reusing plaintext tokenization, simplifies implementation, with plans to develop ciphertext dictionaries and B+ tree lookups for end-to-end ciphertext processing, further enhancing privacy protection.

3 Technical Background

3.1 Large Language Models and Llama3.2-1B

Large language models rely on the Transformer architecture, processing natural language tasks through multi-head self-attention and feed-forward networks (FFNs) (6). Llama3.2-1B(7), a lightweight decoder-only Transformer model with 1 billion parameters, is designed for autoregressive tasks (e.g., dialogue generation) and is well-suited for resource-constrained deployment scenarios. Its architecture consists of Transformer decoder layers, each comprising:

- **Multi-Head Self-Attention**: Computes query, key, and value matrices via matrix multiplication, with softmax normalization to generate attention weights, capturing contextual dependencies in input sequences.
- **Feed-Forward Network (FFN)**: Performs feature transformations through matrix multiplication and silu or relu activations.
- **Layer Normalization and Residual Connections**: Uses arithmetic operations and normalization (sum, mean, var, sqrt, rsqrt) to stabilize training and inference.
- **Embedding Layer**: Maps token IDs to embedding vectors.

Llama3.2-1B's core operators include:

- **Arithmetic and Layer Normalization**: Addition, subtraction, multiplication, and division for residual connections and weight updates; sum, mean, var, sqrt, and rsqrt for layer normalization.
- **Matrix Multiplication**: Supports multi-head self-attention (including scaled dot-product), FFN linear layers, and embedding layers.
- **Nonlinear Operators**: Softmax for attention probability normalization, silu and relu for FFN activations.

These operators, using plaintext weights, perform efficiently in plaintext computation but require FHE adaptation in privacy-sensitive scenarios, encrypting only data flows (e.g., inputs, token IDs, intermediate inference results) to protect data privacy.

3.2 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) is an advanced cryptographic technique that allows arbitrary computations (e.g., addition and multiplication) on encrypted data without decryption, producing results equivalent to plaintext computations. Since Gentry's introduction [2], FHE's

potential in privacy-preserving computation has been widely studied, but its high computational complexity and memory overhead limit its application in complex neural network inference, particularly for Transformer models' high-dimensional matrix operations and nonlinear activations.

ZFHE is a polynomial-based FHE algorithm leveraging coefficient mapping transformation, with security rooted in the Multivariate Quadratic Problem. It incorporates chaotic cryptography principles and introduces an operation dictionary mechanism for efficiency.

The ZFHE algorithm is defined by a five-tuple, with its core principles as follows:

$$M = \sum_{i=1}^{n} a_i \cdot f(x_i) \cdot y_i$$

Where M is the plaintext, f is the function key, which is composed of the pattern code and analytical expression parameters of the analytical function; Y is the polynomial key, which is composed of a real number vector. These two parts together constitute the key of the algorithm:$K = \{f, Y\}$, $Y = \{y_i | i \in I\}$. The ciphertext C is composed of the coefficients in the formula and the function independent variables.

ZFHE's technical advantages include:

1. **Privacy Protection**: Chaotic functions introduce randomness, ensuring high unpredictability of ciphertexts, preventing key recovery through numerical fitting even if ciphertexts are intercepted.

2. **Computational Flexibility**: Supports addition and multiplication for floating-point data, seamlessly handling complex computations in Transformer inference.

3. **Performance Optimization**: The operation dictionary mechanism employs lookup tables to accelerate ciphertext computations, significantly reducing overhead for large-scale neural network inference.

These characteristics make ZFHE particularly suitable for Llama3.2-1B ciphertext inference, overcoming traditional FHE limitations in floating-point operations and memory efficiency.

3.3 Ciphertext Inference Requirements

Privacy-preserving dialogue systems require inputs and outputs to remain encrypted throughout, with inference results consistent with the plaintext model. The complexity of Transformer models, including high-dimensional matrix operations (multi-head self-attention, FFN), nonlinear activations (softmax, silu, relu), and layer normalization (sum, mean, var, sqrt, rsqrt), imposes stringent demands on FHE's computational capabilities. Traditional FHE schemes

face high latency and memory usage challenges, especially on lightweight devices like the Mac mini.

FHE-Llama3.2-1B addresses these through a hybrid architecture combining plaintext weights and ciphertext data flows, meeting the following requirements:

- **Functional Equivalence**: Ciphertext inference results match those of plaintext Llama3.2-1B, ensuring dialogue quality.
- **Performance Optimization**: Inference performance approaches plaintext levels, suitable for resource-constrained edge devices.
- **Controlled Computational Overhead**: ZFHE's operation dictionary and operator optimizations reduce FHE complexity.
- **Compliance**: Meets regulatory requirements (e.g., GDPR) for privacy-sensitive scenarios, safeguarding user data.

These requirements drive the design of FHE-Llama3.2-1B's operator transformation and inference workflow, providing an efficient solution for lightweight privacy-preserving inference.

4 FHE Operator Transformation and Inference

FHE-Llama3.2-1B optimizes all Transformer operators of Llama3.2-1B using ZFHE to enable ciphertext inference while maintaining plaintext weights. The goal is to minimize ciphertext computation latency, approaching plaintext model performance.

4.1 Operator Transformation

We designed ZFHE-compatible optimized versions of Llama3.2-1B's Transformer operators, covering all core operators, categorized into three types:

- **Arithmetic and Layer Normalization**
  Technical Challenge: Basic operations (addition, subtraction, multiplication, division) and layer normalization (sum, mean, var, sqrt, rsqrt) in polynomial encryption environments introduce significant overhead, especially in frequent residual connections and normalization.
  Solution: Inspired by efficient Transformer implementations (e.g., Hugging Face Transformers), we adopt vectorization techniques, batch-processing polynomial coefficients to optimize encrypted representations of addition, subtraction, multiplication, division, sum, mean, and var. For sqrt and rsqrt, linear approximation (e.g., Newton iteration method) are used to convert them into FHE-friendly forms, reducing computational steps.

- **Matrix Multiplication**
  Technical Challenge: Matrix multiplication, central to multi-head self-attention (including scaled dot-product), FFN linear layers, and embedding layers, involves high-dimensional computations, leading to increased latency and memory usage in FHE due to high-order polynomial representations.
  Solution: Decompose matrix multiplication into sub-matrix operations to reduce single-operation complexity, with pre-computed intermediate results minimizing redundant computations. This strategy aligns with ZFHE, ensuring efficient execution of self-attention, FFN, and embedding layers.

- **Nonlinear Operators (softmax, silu, relu)**
  Technical Challenge: Nonlinear operators are difficult to implement in FHE, as FHE does not support complex non-polynomial operations. Softmax normalizes attention probabilities, while silu and relu involve exponential and nonlinear computations, increasing complexity.
  Solution: Polynomial approximation techniques transform softmax, silu, and relu into FHE-friendly polynomial forms. Taylor expansion and least-squares methods optimize approximation accuracy, ensuring correct attention weights and activation functions while reducing computational complexity.

All operators are adapted to ZFHE, ensuring consistency with plaintext computations and significantly reducing ciphertext overhead. Plaintext weights are directly applied to encrypted data flows, maintaining model behavior consistency.

4.2 Inference Workflow

FHE-Llama3.2-1B aims to enable end-to-end encrypted dialogue generation, mirroring the autoregressive inference of plaintext Llama3.2-1B, using plaintext weights to process encrypted data flows for functional equivalence. The workflow leverages ZFHE's chaotic randomness and operation dictionary mechanisms for efficient ciphertext computation. Below, we first describe the ideal end-to-end ciphertext inference workflow, followed by the simplified workflow currently adopted due to unimplemented tokenization steps.

4.2.1 Ideal End-to-End Ciphertext Inference Workflow

The ideal workflow achieves full encryption from input to output, with the following steps:

- **Input Encryption**: Input text is encrypted as ciphertext characters using ZFHE, with chaotic functions generating unpredictable ciphertexts to ensure privacy.

- **Ciphertext Dictionary**: Ciphertext characters are mapped to a ciphertext dictionary, constructing encrypted vocabulary representations consistent with plaintext tokenization logic.

- **Ciphertext B+ Tree**: The ciphertext dictionary is organized in a B+ tree structure, supporting efficient ciphertext lookup to optimize tokenization performance.
- **Ciphertext Lookup (Tokenizer)**: The tokenizer performs lookups in the ciphertext B+ tree, generating encrypted token ID sequences representing the input text's semantic units.
- **Model Generator Interface**: Encrypted token IDs are fed into the model's generator interface, initializing the autoregressive inference process, with plaintext weights processing encrypted inputs.
- **Llama Model Inference**: Llama3.2-1B performs inference on encrypted token IDs using optimized FHE operators (arithmetic, layer normalization, matrix multiplication, softmax, silu, relu), generating encrypted next-token probability distributions via multi-head self-attention, FFN, and layer normalization.
- **Sampling**: Sampling from the encrypted probability distribution yields the next ciphertext token ID, added to the output stream.
- **Iterative Inference**: New ciphertext token IDs are fed back to the generator interface, repeating Llama model inference and sampling until a stopping condition (e.g., maximum sequence length) is met.
- **Ciphertext Output**: The output stream's ciphertext token ID sequence forms the ciphertext response, decrypted by the client into plaintext text, completing dialogue generation.

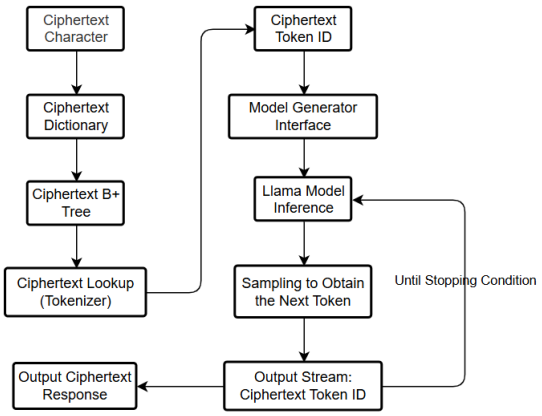

Figure 1: FHE-Llama3.2-1B ciphertext inference workflow.

4.2.2 Current Simplified Workflow

Due to the unimplemented ciphertext tokenization steps (ciphertext characters, dictionary, B+ tree, and lookup), the current approach reuses plaintext tokenization and embedding models, encrypting plaintext vectors for inference. The simplified workflow is as follows:

- **Plaintext Tokenization**: Input text is processed by a plaintext tokenizer, generating plaintext token ID sequences representing semantic units.
- **Token ID Encryption**: Encrypt the plaintext token ID sequences with ZFHE.
- **Ciphertext Embedding**: Ciphertext token IDs are fed into Llama3.2-1B's embedding layer, producing ciphertext vector representations to ensure data privacy.
- **Llama Model Inference**: Llama3.2-1B performs inference on ciphertext vectors using optimized FHE operators (arithmetic, layer normalization, matrix multiplication, softmax, silu, relu), generating encrypted next-token probability distributions via multi-head self-attention, FFN, and layer normalization.
- **Decode and Output**: Sampling from the encrypted probability distribution yields a ciphertext token ID, and added to the output stream.
- **Iterative Inference**: New ciphertext token IDs are fed back to the embedding layer, repeating embedding, inference, and decode until a stopping condition (e.g., maximum sequence length) is met.

The current workflow relies on plaintext tokenization, simplifying implementation but involving plaintext processing at input stages. Privacy protection depends on vector encryption and ciphertext inference. No additional optimizations are applied, with performance relying on operator transformations and ZFHE's operation dictionary mechanism.

5 Experiments and Results

To evaluate FHE-Llama3.2-1B's performance, we conducted experiments comparing ciphertext and plaintext model operator latency and inference efficiency. Experiments were performed under unified hardware configurations, with plaintext Llama3.2-1B as the baseline.

5.1 Experimental Setup

5.1.1 Server Configuration

Experiments were conducted on the following server:

- **Processor**: Mac mini (M4 Pro chip, 14 CPU cores + 20 GPU cores, GPU cores unused).
- **Memory**: 64GB.

5.1.2 Operator Latency Testing

We tested the following operators' single-operation latency, comparing ciphertext and plaintext execution times:

- Arithmetic and layer normalization (addition, subtraction, multiplication, division, sum, mean, var, sqrt, rsqrt, 100×100 matrix).
- Matrix multiplication (100×100 matrix, including embedding and linear layers).
- Nonlinear operators (softmax, silu, relu, 100×100 matrix).

Tests focused on single-operation latency to evaluate FHE operator transformation efficiency.

5.1.3 Model Inference Testing

Llama3.2-1B inference latency tests used the following configuration:

- **Maximum Sequence Length**: 256.
- **Batch Size**: 1 (single inference).
- **Inference Mode**: top-k, top-p sampling.

Tests measured complete inference latency (tokens per second) to assess overall efficiency.

5.1.4 Baseline

- **Plaintext Llama3.2-1B**: Original model, serving as the performance benchmark.

5.1.5 Evaluation Metrics

- **Operator Performance**: Single operator latency (milliseconds).
- **Inference Efficiency**: Inference latency (tokens per second).

5.2 Experimental Results

5.2.1 Operator Latency

Operator latency results are categorized into three types, presented in the following tables:

Arithmetic and Layer Normalization

Matrix Multiplication

Nonlinear Operators

Results indicate that FHE operator transformations introduce additional latency in ciphertext environments, but vectorization, decomposition, and polynomial approximations keep latency within reasonable bounds.

Table 1: Arithmetic and Layer Normalization Latency Comparison

Operator	Ciphertext Latency (ms)	Plaintext Latency (ms)
Addition	0.025702	0.001264
Subtraction	0.022578	0.001097
Multiplication	0.022626	0.000978
Division	0.020599	0.001097
Sum	0.115013	0.001073
Mean	0.109196	0.002503
Variance	0.507808	0.011611
Sqrt	0.209498	0.017905
Rsqrt	0.275493	0.002503

Table 2: Matrix Multiplication Latency Comparison

Operator	Ciphertext Latency (ms)	Plaintext Latency (ms)
Matrix Multiplication	0.042605	0.004196

5.2.2 Inference Efficiency

FHE-Llama3.2-1B's single inference latency is 7.23 tokens per second (147 tokens in 20.34 seconds), about 2.38 times that of plaintext Llama3.2-1B:

- **Ciphertext Inference Latency**: approximately 7.23 tokens/s.
- **Plaintext Inference Latency**: approximately 17.18 tokens/s, 247 tokens in 14.38 seconds.

On a Mac mini (M4 Pro chip, using 14 CPU cores, 20 GPU cores unused, 64GB memory), FHE-Llama3.2-1B achieves an inference speed of approximately 7.23 tokens per second, demonstrating its ability to balance privacy protection and computational efficiency on lightweight devices. This performance is attributed to ZFHE's operation dictionary mechanism and comprehensive operator optimizations, providing a practical solution for privacy-sensitive scenarios.

6 Discussion

6.1 Key Findings

FHE-Llama3.2-1B, leveraging ZFHE, achieves ciphertext transformation of all Llama3.2-1B core operators, including arithmetic, layer normalization (sum, mean, var, sqrt, rsqrt), matrix multiplication (for embeddings and linear layers), softmax, silu, and relu, enabling encrypted dialogue and ciphertext inference. The current approach reuses plaintext tokenization and embedding models, encrypting plaintext vectors for inference. Experiments show a ciphertext

Table 3: Nonlinear Operator Latency Comparison

Operator	Ciphertext Latency (ms)	Plaintext Latency (ms)
Softmax	2.349830	0.013900
Silu	2.288866	0.010920
Relu	0.634599	0.000882

inference latency of 7.23 tokens per second, about 2.38 times that of plaintext, demonstrating high efficiency on lightweight devices (Mac mini, M4 Pro chip, 14 CPU cores, 20 GPU cores unused, 64GB memory). The CPU-only approach ensures inference stability but limits further optimization due to unused GPU cores.

Notably, the 7.23 tokens per second inference speed holds significant practical value in privacy-preserving scenarios. For sensitive data processing in finance, healthcare, and similar domains, FHE-Llama3.2-1B provides end-to-end encrypted inference with acceptable latency on lightweight devices, meeting real-time dialogue and regulatory requirements (e.g., GDPR). Although ciphertext inference is 2.38 times slower than plaintext, reflecting FHE's polynomial encryption and complex operator overhead, this performance benefits from ZFHE's operation dictionary mechanism, which significantly reduces computational complexity, and comprehensive operator optimizations: arithmetic and layer normalization leverage vectorization and sqrt/rsqrt polynomial approximations to reduce encryption overhead; matrix multiplication is adapted to FHE via decomposition; nonlinear operators achieve high-precision computation through polynomial approximations. Using plaintext weights simplifies FHE adaptation, encrypting only data flows to ensure model behavior consistency.

ZFHE's unique design is key to the framework's success. Its security, based on the Multivariate Quadratic Problem and chaotic cryptography, ensures ciphertext unpredictability through chaotic function randomness, thwarting numerical fitting attacks. The operation dictionary mechanism significantly reduces ciphertext computation overhead, enabling efficient matrix multiplication and nonlinear operator execution. Compared to plaintext optimizations in the Hugging Face Transformers ecosystem (e.g., dialogue generation, question answering), FHE-Llama3.2-1B introduces ciphertext inference, opening new avenues for privacy-preserving scenarios.

6.2 Limitations

The current approach has the following limitations:

- **Plaintext Tokenization Dependency**: Reusing plaintext tokenization involves plaintext processing at input stages, limiting end-to-end privacy protection.
- **Inference Latency**: Ciphertext inference latency of approximately 7.23 tokens per second, 2.38 times that of plaintext, reflects FHE's inherent overhead from polynomial encryption

and additional operator computation steps.

- **Unused GPU Cores**: Experiments use only the Mac mini M4 Pro's 14 CPU cores, with 20 GPU cores unused, constraining performance optimization.
- **Model Specificity**: Operator transformations target Llama3.2-1B's decoder-only architecture, with generalizability to other Transformer models (e.g., encoder-decoder structures) requiring further validation.

6.3 Future Work

Future efforts will focus on the following directions:

- **End-to-End Ciphertext Inference**: Implement ciphertext character processing, ciphertext dictionaries, B+ tree lookups, and ciphertext token ID generation to eliminate plaintext tokenization dependency, achieving fully encrypted inference.
- **GPU Acceleration Optimization**: Leverage the GPU cores to develop parallelized FHE operators and inference workflows, further reducing latency.
- **Operator Efficiency Optimization**: Explore more efficient FHE operator implementations, such as improved vectorization algorithms, to narrow the latency gap.
- **Multimodal Encrypted Inference**: Extend the framework to support encrypted inference for text, images, and other modalities, addressing diverse privacy-preserving needs.
- **Cross-Model Generalizability**: Validate operator transformations' applicability to other Transformer models, developing a universal FHE inference framework.

7 Conclusion

We propose FHE-Llama3.2-1B, an encrypted dialogue and ciphertext inference system based on ZFHE and Llama3.2-1B, achieving privacy-preserving inference through comprehensive operator transformation. The current approach reuses plaintext tokenization, encrypting plaintext vectors for inference, supporting FHE-adapted operators (arithmetic, layer normalization, matrix multiplication, softmax, silu, relu) to ensure functional equivalence. Experiments on lightweight devices (Mac mini, M4 Pro chip, 14 CPU cores, 64GB memory) demonstrate an inference performance of approximately 7.23 tokens per second, about 2.38 times that of plaintext, validating its operator latency and inference efficiency.

ZFHE's chaotic randomness and operation dictionary mechanisms provide robust privacy protection and performance optimization. FHE-Llama3.2-1B balances privacy and computational efficiency through vectorization, decomposition, and polynomial approximations, with

its 7.23 tokens per second inference speed offering practical value in privacy-sensitive scenarios like finance and healthcare. Despite reliance on plaintext tokenization and unused GPU cores, future implementation of ciphertext dictionaries, B+ tree lookups, ciphertext token IDs, and GPU acceleration will enable end-to-end ciphertext inference and further enhance performance. The operator transformations' generalizability lays a foundation for FHE adaptation in other models, with broad technical potential. Future work will optimize operator efficiency, explore GPU acceleration, and support multimodal inference, providing stronger support for privacy-preserving scenarios and advancing secure LLM applications in sensitive domains.

References

[1] J. H. Cheon, A. Kim, M. Kim, and Y. Song, ""homomorphic encryption for arithmetic of approximate numbers"," in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT), vol. 10624, 2017, pp. 409–437.

[2] C. Gentry, ""fully homomorphic encryption using ideal lattices"," in Proc. 41st Annu. ACM Symp. Theory Comput. (STOC), 2009, pp. 169–178.

[3] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, ""cryptonets: Applying neural networks to encrypted data with high throughput and accuracy"," in Proc. Int. Conf. Mach. Learn. (ICML), vol. 48, 2016, pp. 201–210.

[4] R. Dathathri, O. Saarikivi, H. Chen, K. Lauter, S. Maleki, M. Musuvathi, and T. Mytkowicz, ""ngraph-he: A graph compiler for deep learning on homomorphically encrypted data"," in Proc. 17th ACM Int. Conf. Comput. Front. (CF), 2020, pp. 80–89.

[5] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, ""gazelle: A low latency framework for secure neural network inference"," in Proc. USENIX Secur. Symp., 2018, pp. 1651–1668.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, ""attention is all you need"," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 5998–6008.

[7] Meta AI, ""llama 3.2: Efficient and high-performing language models"," Meta AI, Tech. Rep., Sep. 2024, accessed: May 21, 2025. [Online]. Available: https://ai.meta.com/research/llama