LATE-FUSION TRANSFORMER WITH SELECTIVE HOMOMORPHIC TOKEN ENCRYPTION FOR EFFICIENT PRIVATE LANGUAGE PROCESSING

Anonymous authors
Paper under double-blind review

ABSTRACT

To protect sensitive information when processing text with an external language model, it is necessary to safely protect the data throughout the entire inference pipeline. Token redaction can safely remove sensitive information trading accuracy for protection, while encrypting entire tokens significantly increases the inference cost. To overcome these limitations, we present Late-Fusion Transformer (LFT) that merges homomorphically encrypted tokens after the final attention block via cross-attention to minimize the computation overhead while keeping the sensitive data in the inference path. The client encrypts under CKKS only the embeddings of the tokens it designates for protection, and the redacted text runs through a plaintext transformer backbone. Then, the *merger* module combines the sensitive information (ciphertext) with the backbone's plaintext representation through a single encrypted cross-attention layer. This allows only a small portion of computations to be done under homomorphic encryption, efficiently enabling private language processing. Finally, the server returns an encrypted output that can only be decrypted by the client using the secret key. We evaluate LFT with 13 globally deduplicated text classification datasets on RoBERTa-base and ModernBERT-base. The mean gain over the redacted-input baseline was positive or at par across twelve datasets where larger redaction losses typically yielded greater gains. A GPU implementation of LFT on FIDESlib reproduces its plaintext-based implementation on 99.9% of the evaluation inputs, verifying reliable CKKS implementation. Against full-layer encrypted RoBERTa-base and ModernBERT-base built on the same GPU and CKKS library, the proposed LFT with selective token encryption is 247$\times$ faster on a single text input and 2,392$\times$ faster per input under batching, on average, demonstrating its efficiency.

1 INTRODUCTION

Clients who use an external model on the server to process clinical notes, financial filings, or support tickets must provide the information needed for the task while limiting the exposure of sensitive content. One effective safeguard is to completely mask sensitive tokens before they leave the client device. However, redacting these words can remove task-relevant information from the text and thereby reduce prediction accuracy (Lothritz et al., 2023; Deußer et al., 2025). Another way of protecting private information is to homomorphically encrypt the entire data and allow third-party servers to run computations directly on ciphertext, without any decryption in the computation path (Moon et al., 2025; De Castro et al., 2025). Running an entire transformer under homomorphic encryption (HE), however, incurs a significant computational overhead due to encrypted matrix multiplication and additional bootstrapping steps that restore the levels a ciphertext consumes (§2).

Therefore, in this work, we present Late-Fusion Transformer (LFT) that combines redaction with selective token encryption. Figure 1 shows the entire protocol on the input "My name is Alex, and I live in Chicago". The server receives "My name is [REDACTED], and I live in [REDACTED]" in plaintext and the two designated tokens only as their embeddings encrypted under CKKS. The plaintext backbone processes the redacted text and one attention layer after it, computed under encryption, lets the encrypted tokens become a part of the inference. We call the module that combines the plaintext and ciphertext paths the *merger*. Then, the server returns the final encrypted vector at

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

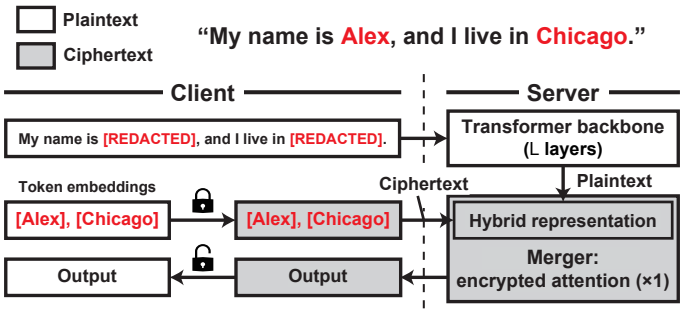

Figure 1: The Late-Fusion Transformer protocol. The redacted text travels to the server as plaintext while the selected sensitive token embeddings are passed as CKKS-encrypted ciphertext. Only the client decrypts the output representation the server returns.

the first output token position to the client. The client safely decrypts the vector at the edge node and passes the vector to the task head, e.g., classification head. The selected private tokens never get decrypted on the server and the server only sees the redacted text and mask positions as plaintext information (§3). Late-Fusion Transformer transmits the designated content only in encrypted form, but does not prevent the server from inferring that content from the context that remains, and §4.3 measures how much can be re-identified that way.

We implemented the encrypted merger (*proposed*) and two fully encrypted baselines, 12-layer RoBERTa-base and 22-layer ModernBERT-base, on the same GPU and CKKS library. Each reproduces its own plaintext predictions on 96.9% to 99.9% of the evaluated inputs. For batched inputs, LFT's encrypted merger is 1,788× and 2,997× faster per input than the RoBERTa-base and ModernBERT-base baselines (§5.2). To evaluate the quality of LFT, we compare the accuracy with the redacted baseline that only sees the masked input text. Another comparison is made to check whether the accuracy improvement comes from keeping the sensitive tokens not from the architectural modification. Thus, we simply train the LFT model while its encrypted input is held at zero to validate how accuracy changes by adding an attention layer. On two evaluated backbones, the average gain over the redacted baseline is positive on 11 of the 13 datasets for each, and larger redaction losses generally accompany higher gains (§4). Against the zeroed LFT baseline, LFT is ahead or at par on 12 datasets with RoBERTa and 10 datasets with ModernBERT.

Our main contributions can be summarized as:

- **Novel architecture for selective token encryption.** The redacted text runs through a plaintext backbone, and one additional encrypted attention layer (i.e., merger) expands the capability of selective token protection. With the query taken from the plaintext representation, while encrypted embeddings of client-selected tokens are projected into keys and values by the server, we use the selected encrypted tokens in prediction without decrypting them on the server (§3).
- **Controlled evaluation of encryption path and risk of re-identification.** The gain over the redacted baseline is measured with a zero-input control that separates the added structure from the content to solely evaluate the quality of an encryption path (§4). A separate evaluation measures how much of the masked content can be reconstructed from what the server sees, the redacted text and the mask positions (§4.3).
- **GPU-based LFT implementation with per-stage batching.** To improve the performance of LFT on GPU, we present a stage-specific batching strategy, which packs multiple inputs into a single set of 32,768 available CKKS ciphertext slots (§5).

2 Background and Related Work

Text de-identification removes sensitive words/phrases or replaces them with more general expressions (Lison et al., 2021; Deußer et al., 2025). Deciding what to protect depends on the context and remains difficult for existing models (Brynjólfsson et al., 2026), and even a general replacement can leak. Pilán et al. (2025) choose how general each replacement must be by testing whether an attacker

can still infer the original. Re-identification research asks the converse question, how much of the masked content can be inferred from the remaining context and background knowledge (Charpentier & Lison, 2025a;b).

PrivacyRestore (Zeng et al., 2025) removes sensitive spans from a query to a large language model and restores their utility through an auxiliary path. With the language model frozen, the server learns restoration vectors for predefined span types, which can include specific symptoms such as black stools. The client combines the vectors corresponding to its removed spans using importance weights and adds noise before transmission. The server uses the resulting vector to steer its attention-head activations during generation. This path carries a learned representation of the removed information, protected by noise, rather than encrypted token embeddings. It avoids homomorphic computation and targets decoder-only LLMs, so we do not compare it directly here.

Encrypted inference instead computes on the protected content without removing it. CKKS (Cheon et al., 2017; 2018), a fully homomorphic scheme, supports approximate arithmetic on vectors of real and complex numbers and is used in encrypted neural network inference (Ng & Chow, 2023). Running a transformer under encryption, however, requires large matrix operations and polynomial approximations of the non-linear functions, and the depth of that computation brings a bootstrapping cost with it. Figure 2b shows where the cost arises, comparing the encrypted cost of the primitive operations against plaintext execution in the environment used in this work. The scale for a whole model comes from a different environment. THOR (Moon et al., 2025) runs full-layer CKKS inference of a BERT-base encoder on one A100 and reports about ten minutes per 128-token input.

Prior work explores cryptographic inference and ways to reduce its cost. MPCFormer (Li et al., 2023) distills the model onto MPC-friendly approximations. CipherPrune (Zhang et al., 2025) securely prunes unimportant tokens, while disclosing the number of retained tokens and a mask that selects their polynomial approximation degrees. EncryptedLLM (De Castro et al., 2025) runs a GPT-2 forward pass on GPU-accelerated FHE and Ng & Chow (2023) systematize the design space.

The proposed Late-Fusion Transformer, in this work, combines redaction with selective encrypted computation. It brings the redacted content back into the prediction, which redaction alone gives up. The redacted text remains in plaintext, while encrypted computation is limited to the merger's cross-attention, plaintext–ciphertext addition, and layer normalization (§3). Rather than a learned restoration signal, the client sends CKKS-encrypted embeddings of the designated tokens to server. The merger uses those embeddings for prediction, and the server sees only their ciphertexts. What the public context still reveals is not assumed away. §4.3 measures it with re-identification attacks.

3 Late-Fusion Transformer: Proposed Architecture

3.1 Dual Plaintext and Ciphertext Input Paths

Figure 2a shows the proposed Late-Fusion Transformer comparing it to a fully encrypted baseline. There are two input paths, i.e., private ciphertext and redacted plaintext, and it is completely up to the client which tokens to designate for the encryption. Those selected tokens are replaced with the special token `[REDACTED]` in the plaintext, keeping the token positions and the sequence length. Then embeddings of the designated tokens are gathered and encrypted under CKKS (Cheon et al., 2017), an HE scheme (Figure 3b), and sent to the server. Throughout the experiment, we assume the client gathers the embeddings of up to 64 designated tokens in sequence order into a fixed-size matrix $\mathbf{Z}$ and fills unused entries with zeros. If more than 64 tokens are designated, only the first 64 enter the encrypted path, and the remaining designated tokens stay masked in the plaintext input. The server therefore receives the redacted text and the encrypted embeddings. We assume a semi-honest server, one that follows the prescribed protocol but may analyze everything it receives (Evans et al., 2018). The secret key stays with the client, and nothing decrypted goes to the server. About the protected content, the server learns nothing beyond what the redacted text and the mask positions reveal or information that can be inferred from them (§4.3).

3.2 The Merger Module

The main challenge with compute-under-encryption is the significant per-operation slowdown compared to the plaintext-based computing (Figure 2b). Thus, our primary goal was to minimize the

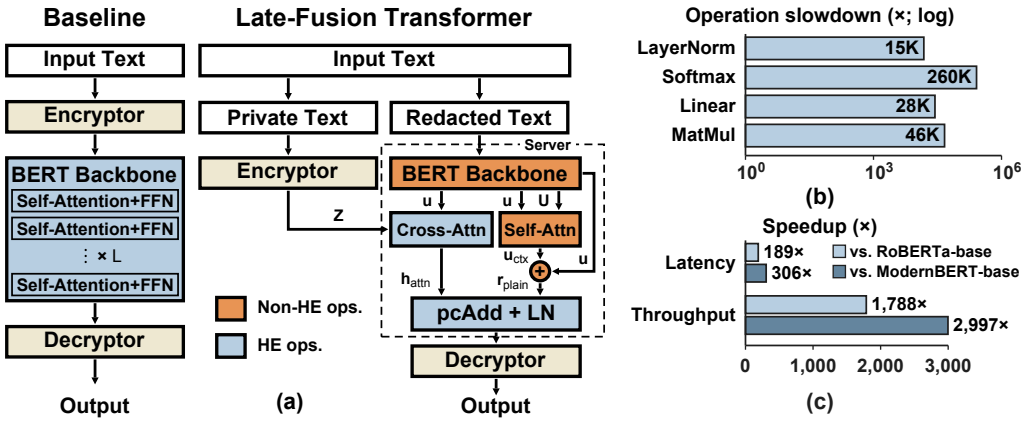

Figure 2: Proposed LFT architecture and cost of selective encrypted inference. (a) Full-layer HE vs. Late-Fusion Transformer. (b) Per-operation slowdown of HE-based computing against plaintext execution on the same GPU. (c) Speedup of the merger over the two full-layer HE baselines.

portion of HE operations leading to adding a single merger module right after the main backbone. The server's plaintext backbone processes the redacted text understanding the global context with public tokens and redacted positions. At each token position, we concatenate the hidden states from the backbone's final two layers and feed that to one learned linear projection. We denote the projected representations at all token positions by $\mathbf{U}$, and the vector at the first, classification-token (CLS) position by $\mathbf{u} = \mathbf{U}[0, :]$. The merger consists of two attention layers: (i) self-attention that focuses on interactions between all pairs of public tokens, and (ii) cross-attention that finds a relation between public and private tokens (Figure 2a). Appendix E compares the cross-attention with simpler alternatives. The query vectors $\mathbf{q}$ for both attention layers are generated using $\mathbf{u}$, while the key $\mathbf{K}$ and the value $\mathbf{V}$ are generated from $\mathbf{U}$ for the self-attention and $\mathbf{Z}$ for the cross-attention. The self-attention yields the context vector $\mathbf{u}_{\text{ctx}}$ as a plaintext and does not need encrypted operations. The private embeddings $\mathbf{Z}$ remain encrypted during cross-attention, which outputs $\mathbf{h}_{\text{attn}}$ as a ciphertext. As we generate the $\mathbf{q}$ vector from $\mathbf{u}$ at the CLS position, i.e., the position where the task head reads at the client, $\mathbf{h}_{\text{attn}}$ is the result by attending to at most 64 designated tokens. The CLS-focused $\mathbf{q}\mathbf{K}^T$ and $\mathbf{a}\mathbf{V}$ computations involve the minimum number of HE operations providing significant speedup (Figure 2c), where $\mathbf{a}$ denotes the attention weights.

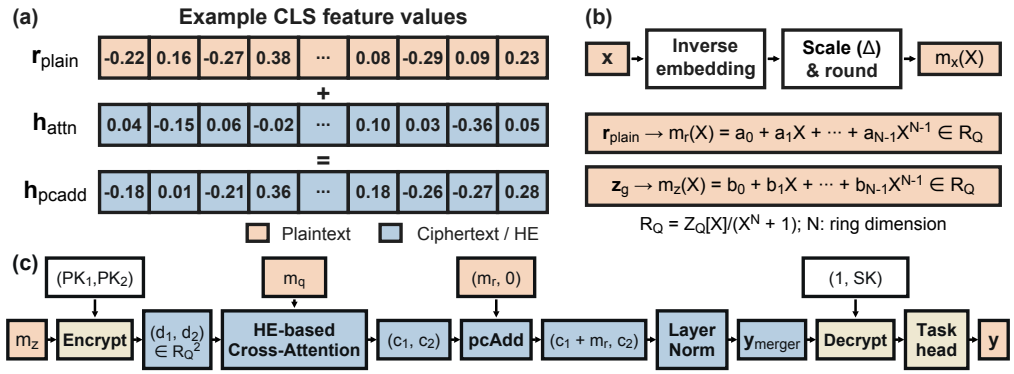

Figure 3: The merger's encrypted path: (a) Plaintext–ciphertext addition (pcAdd), (b) CKKS encoding of an input vector, e.g., $\mathbf{z}_g \to m_z(X)$, where a slot vector $\mathbf{z}_g$ is packed and partitioned from $\mathbf{Z}$, and (c) the path from the client's encryption of m_z through the encrypted merger output $\mathbf{y}_{\text{merger}}$ to the client's prediction $\mathbf{y}$.

Then, the outputs from the two attention layers are merged. The plaintext path yields the plaintext sum $\mathbf{r}_{\text{plain}} = \mathbf{u} + \mathbf{u}_{\text{ctx}}$ which is added to $\mathbf{h}_{\text{attn}}$ without encrypting $\mathbf{r}_{\text{plain}}$ via plaintext–ciphertext addition (pcAdd; Figure 3a). The pcAdd is followed by layer normalization under encryption:

$$\mathbf{y}_{\text{merger}} = \text{LayerNorm}\big(\mathbf{r}_{\text{plain}} + \mathbf{h}_{\text{attn}}\big).$$

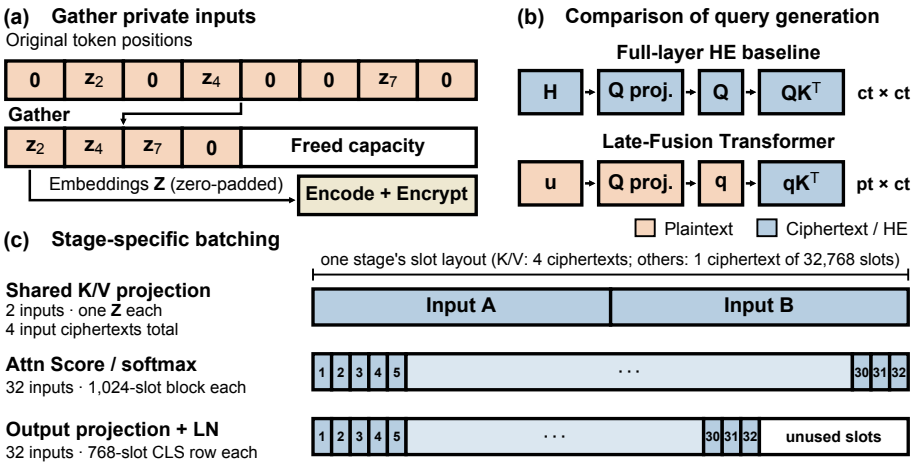

Figure 4: CKKS execution of the merger: (a) The gathered token embeddings $\mathbf{Z}$, zero-filled where unused. (b) The query stays in plaintext for LFT, so the attention scores are plaintext–ciphertext products. (c) Per-stage batching in LFT. The number of inputs processed together depends on the slot layout required by each stage.

Since $\mathbf{r}_{\text{plain}}$ contains the $\mathbf{u}_{\text{ctx}}$ term, it is not merely an output from the backbone but rather a learned plaintext representation that blends better with $\mathbf{h}_{\text{attn}}$ by training a self-attention layer to minimize the task loss. The server finally returns the ciphertext $\mathbf{y}_{\text{merger}}$, and the client decrypts it and applies the task head (Figure 3c). The transformer blocks and the merger parameters stay on the server. When no tokens are designated, the prediction depends on the plaintext path (Appendix G.1).

3.3 CKKS Execution and Our Stage-Specific Batching Strategy

Figure 4 shows how our merger benefits from the limited number of encrypted computations by processing multiple inputs in parallel. The fixed-size matrix of token embeddings $\mathbf{Z}$ in Figure 4a gives every input the same encrypted layout, regardless of the number of designated tokens. With the plaintext query $\mathbf{q}$ in Figure 4b, attention scores are computed by plaintext–ciphertext products ($\mathbf{qK}^T$). Since LFT focuses on the CLS position, the score stage produces a 64-d vector per head while the output projection and layer normalization (LN) process one 768-d vector.

These intermediate representations occupy only part of the available ciphertext slots, leaving room for other inputs (*batched CKKS*). Each CKKS operation acts on all 32,768 slots at once, so sharing those slots by multiple inputs allows parallel execution. Figure 4c shows our stage-specific batching strategy: K/V projection carries two inputs, attention score and softmax carry 32 inputs, and output projection and LN carry 32 inputs together. All inputs in a batch must be encrypted under the same client's keys. The score stage fills all slots with distinct inputs in a full batch and controls bootstrap error by bootstrapping twice after the exponential and at two points of every squaring step (Bae et al., 2022). Appendix G.1 gives the packing and bootstrap details and execution costs are measured in §5.2.

4 Experimental Results and Analysis

We evaluate accuracy of LFT on two encoders fine-tuned per task, RoBERTa-base (Liu et al., 2019) and ModernBERT-base (Warner et al., 2025). The accuracies in this section are those of the plaintext implementations for fast evaluation; §5 separately tests how closely the CKKS-encrypted merger on full test splits and the full-layer baselines on subsets reproduce their own plaintext predictions.

4.1 Experimental Setup

Compared models. We compare the proposed LFT with three baselines: *ORIG* is the original backbone with unmasked text, *RED* uses the same backbone with redacted text, and *ZEROZ* uses the same architecture as LFT but with only redacted text as input. ZEROZ keeps the architecture but

is trained without the private text to show that the accuracy improvement of LFT mainly comes from resurrecting the redacted tokens. All four models share the same data splits and designation policy. The gain is $\mathrm{Acc}(\mathrm{LFT}) - \mathrm{Acc}(\mathrm{RED})$, where LFT adds two things to RED at once, i.e., access to designated content through the ciphertext path and the learnable merger module.

Designating what to protect. LFT assumes that the client designates the tokens to protect. In the experiments, we automate the designation with a fixed pipeline, so that it is consistent across conditions and the results are reproducible. The pipeline targets the confidential attributes and identifiers of a document rather than an arbitrary selection of words, and it has two layers. The first is a rule layer that matches dictionaries of confidential attributes and patterns for identifiers. A token it matches is always designated. The second is a named-entity layer that selects tokens the rule layer misses, using two pretrained detectors, i.e., spaCy's general-domain model and GLiNER prompted with medical categories (Zaratiana et al., 2024). The pipeline fully covers 74% of the spans that human annotators marked in the Text Anonymization Benchmark (Pilán et al., 2022) (Appendix B).

Datasets. We evaluate on 13 text classification datasets in three domains, i.e., medical, mental health, and general, covering tasks such as diagnosis from symptoms, medical entailment, mental-health status, banking intent, hate speech, and sentence-pair paraphrase, with label counts from 2 to 77. The six medical and mental-health datasets are relevant to the setting the LFT targets, because what a user would want to hide, e.g., symptoms, medications, and mental state, is also the task-sensitive signal. In many of the seven general-domain datasets, what is masked is closer to incidental information such as names and places, so we expected both the loss and the recovery to be small. The sources and preprocessing of the datasets are detailed in Appendix A.

Training and reporting. Each model is trained with five seeds, and the reported accuracy in this section is the mean over those seeds. The gain is seed-paired, i.e., $\mathrm{gain}_s = \mathrm{LFT}_s - \mathrm{RED}_s$ with the seed s. We report the mean and standard deviation of the five paired gains to describe variation across training seeds. Within a dataset, every model trains at one base learning rate, and ModernBERT-base inherits the RoBERTa-base rates with one exception, on Symptom2Dx (Appendix C). Experiments under additional masking conditions are in Appendix D.

4.2 Accuracy Evaluation

Table 1 reports accuracy on thirteen datasets for both backbones, grouped by task domain. Six medical and mental-health datasets are the main LFT targets, where private data directly affect the accuracy. On these datasets, simple redaction degrades 5.6 to 18.0 pp on RoBERTa and 4.9 to 15.2 pp on ModernBERT. The proposed LFT recovers a mean of 70% and 67% of the loss on RoBERTa and ModernBERT, respectively. In the general domain, the loss is mostly 2 pp or less, leaving less room to recover; Bias in Bios and HateXplain have larger losses and gains. Across all thirteen datasets, the mean gain on RoBERTa is positive on eleven datasets, ranging from 0.1 to 15.8 pp, and negative on MRPC and GoEmotions. The LFT surpasses ZEROZ, i.e., the model with the same architecture trained without the ciphertext path, on every dataset but MRPC. On ModernBERT, the mean gain is positive on eleven datasets from 0.4 to 14.5 pp with a similar trend to RoBERTa.

Figure 5 plots the gain of LFT against the accuracy loss due to simple masking (RED). On both backbones, larger masking losses generally accompany larger gains, and the distance below the diagonal is what LFT could not recover. The medical and mental-health datasets (blue and orange) gather at the upper right and the general-domain ones (grey) at the lower left for both backbones. The per-dataset gains of the two differ by at most 1.3 pp on eleven of the thirteen datasets, demonstrating the backbone-agnostic impact of the merger. The datasets that sit away from the trend are MRPC and MedNLI, which compare a pair of sentences, on both backbones. On MRPC, masking loses only 1.3 pp/0.3 pp while LFT ends 3.9 pp/6.3 pp below RED on RoBERTa/ModernBERT. The ZEROZ, which only modifies the architecture, shows higher accuracy than the LFT, implying that the general sentence-pair dataset relies more on understanding global context than focusing on specific tokens. On MedNLI, i.e., another sentence-pair dataset with reliance on designated tokens, masking loses 13.0 pp/13.1 pp and LFT recovers 1.3 pp/2.2 pp on RoBERTa/ModernBERT, the smallest recovery among the datasets with a large loss. We conjecture that LFT may need more than one layer to allow designated tokens to be fused earlier for better global context understanding.

Table 1: Classification accuracy on the thirteen datasets averaged over five seeds. Gain (LFT−RED) is computed per seed and given as mean ± the standard deviation of five paired differences.

	RoBERTa-base					ModernBERT-base				
Dataset	ORIG	RED	ZEROZ	LFT	Gain (LFT−RED)	ORIG	RED	ZEROZ	LFT	Gain (LFT−RED)
Medical										
ADR	92.3	82.6	78.5	87.9	+5.3 ±1.7	90.4	82.7	79.3	86.8	+4.1 ±2.2
Drug Reviews	85.7	69.8	71.2	85.1	+15.3 ±0.6	85.6	70.4	70.8	84.9	+14.5 ±0.4
MedNLI	80.2	67.1	67.1	68.5	+1.3 ±1.4	79.7	66.7	66.2	68.8	+2.2 ±0.8
Symptom2Dx	95.7	77.6	81.8	93.5	+15.8 ±3.4	92.1	78.3	67.6	87.0	+8.7 ±2.0
Mental health										
MentalHealth	82.6	77.1	78.0	81.4	+4.3 ±0.6	82.3	77.5	77.6	81.5	+4.1 ±0.9
RedditMH	84.6	72.4	73.0	83.6	+11.2 ±0.4	84.6	73.1	73.3	83.5	+10.4 ±0.5
General										
BANKING77	91.8	89.9	91.4	93.2	+3.3 ±0.4	92.3	89.9	90.7	92.3	+2.4 ±0.8
Bias in Bios	84.4	78.4	78.3	82.5	+4.1 ±0.2	84.1	78.4	78.4	82.8	+4.3 ±0.3
FinPhraseBank	97.6	98.0	98.0	98.1	+0.1 ±0.6	97.8	97.8	97.7	97.5	−0.3 ±1.1
GoEmotions	62.4	61.3	60.7	60.9	−0.4 ±0.6	63.0	61.4	61.8	61.8	+0.4 ±0.4
HateXplain	69.9	62.9	62.2	67.1	+4.2 ±0.4	68.2	61.7	62.6	67.2	+5.5 ±1.1
MDCC	88.4	87.0	87.3	87.8	+0.7 ±0.6	86.6	86.4	87.5	87.4	+1.0 ±0.6
MRPC	86.3	85.0	84.2	81.1	−3.9 ±0.8	82.7	82.4	80.0	76.0	−6.3 ±1.4

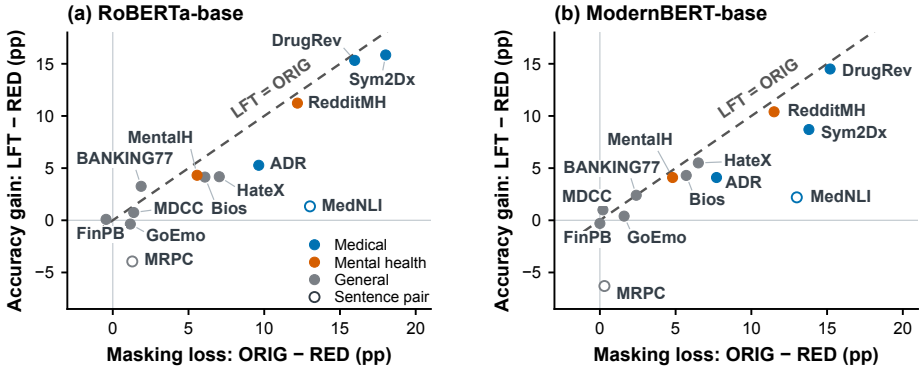

Figure 5: Correlation between the mean accuracy lost to masking (ORIG − RED) and the gain (LFT − RED) on the full test split of (a) RoBERTa and (b) ModernBERT. The dashed line represents the full recovery. Colors mark the domain, i.e., blue for medical, orange for mental health, and grey for general, and open markers are the two sentence-pair datasets (MRPC, MedNLI).

4.3 Re-identification Test: What the Public Context Reveals

In this subsection, we evaluate how much of the designated content can be reconstructed from the public context available to the server, the redacted text and mask positions, without using the ciphertexts. A span is a contiguous run of masked tokens, and it counts as reconstructed only when the attacker's prediction matches the original text exactly. The two fixed attackers, ModernBERT-large and RoBERTa-large, are the two strongest backbones of a wider search (Appendix F) and are applied identically to every dataset. Table 2 summarizes the reconstruction rates for all thirteen datasets. The designated tokens would be exposed in full unless they are sent encrypted. The attackers recover 1.2 to 28.8% of the masked spans from the public context alone, highest where the spans are common symptom words (Symptom2Dx) and lowest where they are drug and condition names (ADR).

Table 2: Percentage of masked spans reconstructed exactly from the redacted text.

	ADR	DrugRev	MedNLI	Sym2Dx	MentalH	RedditMH	BANK77	Bios	FinPB	GoEmo	HateX	MDCC	MRPC
ModernBERT-large	1.2	22.8	5.8	28.8	23.5	24.9	14.0	20.0	11.7	8.5	5.8	18.8	21.4
RoBERTa-large	1.4	18.5	3.9	23.8	20.4	20.6	14.0	18.1	10.3	8.0	4.6	18.3	22.8

5 Encrypted Inference: Fidelity and Cost Analysis

5.1 CKKS Implementation

We implemented the merger in LFT, the full-layer encrypted RoBERTa-base (12 layers), and the full-layer encrypted ModernBERT-base (22 layers) on FIDESlib (Agulló-Domingo et al., 2025), a GPU CKKS library built on OpenFHE (Al Badawi et al., 2022), and run each on one NVIDIA B200 GPU. All three use ring dimension $N = 2^{16}$, i.e., 32,768 slots per ciphertext, and a 23-level modulus chain with bootstrapping. For the main uniform ternary secret key, $\log QP = 1{,}732$ bits lie within the 128-bit classical bound for $N = 2^{16}$ in the guidelines of Bossuat et al. (2025). Bootstrapping uses the sparse-secret encapsulation method of Bossuat et al. (2022) (Appendix G).

All three implementations follow the diagonal packing of THOR (Moon et al., 2025), with weights encoded in a diagonal-major form (Halevi & Shoup, 2014; Jiang et al., 2018). Their encrypted workloads differ in the operations and token positions they process. The RoBERTa baseline evaluates all 12 layers, each with a softmax, two layer normalizations and a GELU, and carries all 128 token positions through every layer. The ModernBERT baseline also carries all 128 token positions through its 22 layers; its blocks use attention softmax, pre-layer normalization, and GeGLU feed-forward units. The LFT merger evaluates one cross-attention, adds the result with the plaintext sum, and applies one layer normalization, with no feed-forward block or GELU. It uses the plaintext query and per-stage batching as explained in §3.3. For both full-layer implementations, the 16-input run processes inputs one by one at each layer before moving to the next.

The LFT merger and both full-layer baselines are trained with the original non-linear functions. For encrypted evaluation, all three models use the normalized-and-squared softmax proposed by Cho et al. (2024), along with polynomial approximation for layer normalization; the full-layer models additionally apply polynomial approximation for GELU. The RoBERTa baseline is fine-tuned separately with activation clipping and a variance penalty to reduce intermediate activation ranges. Its plaintext reference is the unclipped export of that same checkpoint. The training settings and their effect on task accuracy are reported in Appendix G.2. The ModernBERT baseline is fine-tuned separately with pre-activation and variance-ratio penalties, without activation clipping.

5.2 Fidelity and Cost Evaluation Results

We measured prediction agreement between each encrypted implementation and its own plaintext realization on the same inputs. The LFT merger and full-layer RoBERTa evaluations use RoBERTa-base checkpoints. The merger agrees with its plaintext realization on 99.9% of the 39,497 inputs in thirteen full test splits. The RoBERTa baseline agrees on 99.1% of 1,664 inputs, with 128 inputs per dataset. The 22-layer ModernBERT baseline agrees with its plaintext realization on 96.9% of 416 inputs, with 32 inputs per dataset. The limit of 64 designated tokens per input excludes 0.14% of designated tokens in the merger evaluation. These tokens remain masked and are omitted from both the plaintext and encrypted merger inputs. The fidelity test includes the effects of polynomial approximation and the approximate arithmetic of CKKS. Per-dataset prediction agreement and calibration details for the merger and both full-layer baselines are in Appendices G.1 and G.2.

Table 3 reports the costs of all implementations, separating one-time server model preparation from per-input work. Server times cover the encrypted computation with a warm weight cache. They exclude plaintext server computation, communication, client-side preparation, one-time setup and the task head, which runs on the client after decryption. Separately measured single-input plaintext runtimes for RoBERTa-based LFT are 5.31 ms for the backbone and concatenation–projection, and 0.109 ms for public self-attention and the plaintext sum, averaged over per-dataset medians across the thirteen datasets on the same GPU. These are excluded from Table 3 as we focus on dominant runtime of encrypted computations. The merger's encrypted server segment is 189× and 306× faster than the RoBERTa and ModernBERT baselines for a single input, and 1,788× and 2,997× faster under batching. The ModernBERT baseline takes 392.6 s for a single input and 332.3 s per input under batching, i.e., $1.62\times$ and $1.68\times$ the 12-layer RoBERTa baseline. Batched inference times are amortized per input, using sixteen inputs for each full-layer implementation and the stage-specific batches of §3.3 for the merger. For a single input, softmax accounts for most of the merger's encrypted computation time, and its squaring count K ranges from 2 to 5 across datasets. The merger requires 20 to 35 bootstraps for a single input.

Table 3: Full-layer encryption and the encrypted merger on one NVIDIA B200 GPU. Server times use a warm cache; one-time model preparation excludes key generation and bootstrap setup.

	Full-layer HE		
	RoBERTa	ModernBERT	Merger
Time (s)			
Server model preparation, once	1,143.5	1,569.7	128.0
Client encoding and encryption (CPU), per input	0.28	0.27	0.25
Encrypted server time, single input	242.5	392.6	1.285
Encrypted server time, batched, per input	198.2	332.3	0.111
Communication			
Upload / download, single input (MB)	38 / 4.2	38 / 4.2	38 / 1.1
Key material, once per client (GB)	32.7	11.6	21.2

In the single-input setting, all three implementations have the same upload size, while the merger reduces the download size. Batching further reduces the merger's amortized upload to 19 MB per input and its download to 0.03 MB per input. Appendix G.3 gives the measurement procedure, per-dataset runtimes and communication details.

6 LIMITATIONS

The experiments cover classification with inputs limited to 128 tokens and do not evaluate long documents in full. The merger updates only the classification position and cannot be extended unchanged to autoregressive generation. It also excludes designated tokens beyond the first 64 in each input. LFT lowers accuracy relative to RED on MRPC and provides limited recovery on MedNLI. To overcome some of these limitations, we may extend the merger's encrypted path beyond the classification position to explore how a multi-position design adapts to sentence-pair datasets or autoregressive generation.

The reported speedups compare our own full-layer implementations with the LFT merger on the same GPU and CKKS library. Full-layer encryption protects all input tokens, while LFT reveals the redacted text and mask positions, so the comparison involves different disclosure. Our full-layer baselines also compute unused output positions in their final layers. The ratios are therefore specific to the evaluated implementation scenarios.

All three implementations use initial calibration statistics from pools disjoint from the evaluated inputs, but their final approximation settings also use evaluation-input numerical ranges, without evaluation labels. Fidelity therefore measures agreement on inputs that informed the numerical configuration. We have not measured how often previously unused inputs fall outside the fixed approximation domains.

7 CONCLUSION

In this paper, we proposed Late-Fusion Transformer that uses the embeddings of the tokens designated for protection under encryption, and thus recovers accuracy that simple redaction sacrifices. On two encoders, it is more accurate than RED, i.e., the model given only the redacted text, on eleven of thirteen datasets, and larger masking losses generally accompany larger gains. On most medical and mental-health datasets, where the designated content is the task-sensitive signal, it recovers most of what masking loses. Against ZEROZ, i.e., the same LFT architecture but trained separately without the sensitive input, the proposed LFT is ahead or at par on every dataset but MRPC on RoBERTa-base and on ten datasets on ModernBERT-base. GPU-based CKKS implementation of LFT reproduces the plaintext predictions on 99.9% of the evaluated inputs, and against fully encrypted baselines on the same hardware it reduces the execution time of the encrypted server segment by 247$\times$ for a single input and 2,392$\times$ for batched inputs, on average. The LFT confines encrypted computations to the merger regardless of the number of backbone layers. These results show that using selective homomorphic token encryption for prediction, without handing designated data to the server in plaintext, is a feasible approach that efficiently protects private data.

Reproducibility Statement

Dataset sources and preprocessing are described in Appendix A, training settings and seeds in Appendix C, and CKKS parameters and calibration procedures in Appendix G.1. The supplementary code includes data preparation, masking, training and evaluation scripts, dataset manifests, numerical reference results and table generators, and per-dataset CKKS calibration plans. Its README links to instructions for obtaining external data and models, preparing the licensed masking dictionary, and running the experiments. The table generators render the supplied measurements without rerunning the experiments. Dataset texts, trained checkpoints, large input/output dumps and full experiment logs are not bundled. Newly trained checkpoints require fresh export and calibration before encrypted evaluation.

AI Use Statement

The authors formulated the research question, set the research direction, and made the major design decisions and final judgments on the interpretation of the results without AI support. Under the authors' direction and review, we used LLM assistants in four ways. (i) We obtained feedback on the methodology, experimental design and interpretation of the results, and explored implementation alternatives and candidate numerical settings. (ii) We used them to debug code and to aggregate and analyze results. This included cleaning and reformatting public datasets, screening candidate terms for the masking dictionaries, table and figure generators, and the numerical error analysis of the merger and the full-layer baselines. (iii) We used them to develop the wording of the manuscript based on content and structure specified by the authors, and to assist with translation and sentence correction. (iv) We used them to find and analyze candidate related work. We did not use them to generate new data samples or to formulate the research question. The authors are responsible for decisions on adopting proposed alternatives, validation of the reported results, and the content and claims of the final manuscript.

References

Carlos Agulló-Domingo, Óscar Vera-López, Seyda Guzelhan, Lohit Daksha, Aymane El Jerari, Kaustubh Shivdikar, Rashmi Agrawal, David Kaeli, Ajay Joshi, and José L. Abellán. FIDESlib: A fully-fledged open-source FHE library for efficient CKKS on GPUs. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 1–3. IEEE, 2025. doi: 10.1109/ISPASS64960.2025.00045.

Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. OpenFHE: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pp. 53–63. ACM, 2022. doi: 10.1145/3560827.3563379.

Youngjin Bae, Jung Hee Cheon, Wonhee Cho, Jaehyung Kim, and Taekyung Kim. META-BTS: Bootstrapping precision beyond the limit. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 223–234. ACM, 2022. doi: 10.1145/3548606.3560696.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.148.

Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270, 2004. doi: 10.1093/nar/gkh061.

Jean-Philippe Bossuat, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. Bootstrapping for approximate homomorphic encryption with negligible failure-probability by using sparse-secret encapsulation. In *Applied Cryptography and Network Security (ACNS 2022)*, volume 13269 of *Lecture Notes in Computer Science*, pp. 521–541. Springer, 2022. doi: 10.1007/978-3-031-09234-3_26.

Jean-Philippe Bossuat, Rosario Cammarota, Ilaria Chillotti, Benjamin R. Curtis, Wei Dai, Huijing Gong, Erin Hales, Duhyeong Kim, Bryan Kumara, Changmin Lee, Xianhui Lu, Carsten Maple, Alberto Pedrouzo-Ulloa, Rachel Player, Yuriy Polyakov, Luis Antonio Ruiz Lopez, Yongsoo Song, and Donggeon Yhee. Security guidelines for implementing homomorphic encryption. *IACR Communications in Cryptology*, 1(4), 2025. doi: 10.62056/anxra69p1.

Sean Brynjólfsson, Shashvat Jayakrishnan, Esha Sali, Diptanshu Purwar, and Madhav Aggarwal. RedactionBench, 2026. URL https://arxiv.org/abs/2606.18782. arXiv preprint.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.nlp4convai-1.5.

Lucas Georges Gabriel Charpentier and Pierre Lison. Re-identification of de-identified documents with autoregressive infilling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1192–1209. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.60.

Lucas Georges Gabriel Charpentier and Pierre Lison. Stronger re-identification attacks through reasoning and aggregation, 2025b. URL https://arxiv.org/abs/2510.09184. arXiv preprint.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pp. 409–437. Springer, 2017. doi: 10.1007/978-3-319-70694-8_15.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pp. 360–384. Springer, 2018. doi: 10.1007/978-3-319-78381-9_14.

Wonhee Cho, Guillaume Hanrot, Taeseong Kim, Minje Park, and Damien Stehlé. Fast and accurate homomorphic softmax evaluation. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 4391–4404. ACM, 2024. doi: 10.1145/3658644.3670369.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in Bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 120–128. ACM, 2019. doi: 10.1145/3287560.3287572.

Leo De Castro, Daniel Escudero, Adya Agrawal, Antigoni Polychroniadou, and Manuela Veloso. EncryptedLLM: Privacy-preserving large language model inference via GPU-accelerated fully homomorphic encryption. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12677–12688. PMLR, 2025. URL https://proceedings.mlr.press/v267/de-castro25a.html.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4040–4054. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.372.

Tobias Deußer, Lorenz Sparrenberg, Armin Berger, Max Hahnbück, Christian Bauckhage, and Rafet Sifa. A survey on current trends and recent advances in text anonymization. In *2025 IEEE 12th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9. IEEE, 2025. doi: 10.1109/DSAA65442.2025.11247969.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP)*, 2005. URL https://aclanthology.org/I05-5002/.

David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security*, 2(2–3):70–246, 2018. doi: 10.1561/3300000019.

Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health (DH)*, pp. 121–125. ACM, 2018. doi: 10.1145/3194658.3194677.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45 (5):885–892, 2012. doi: 10.1016/j.jbi.2012.04.008.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740.

Shai Halevi and Victor Shoup. Algorithms in HElib. In *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pp. 554–571. Springer, 2014. doi: 10.1007/978-3-662-44371-2_31.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. Secure outsourced matrix computation and application to neural networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1209–1222. ACM, 2018. doi: 10.1145/3243734.3243837.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Hao Zhang. MPCFormer: Fast, performant and private transformer inference with MPC. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=CWmvjOEhgH-.

Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4188–4203. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.323.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692. arXiv preprint.

Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Saad Ezzini, Tegawendé Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. Evaluating the impact of text de-identification on downstream NLP tasks. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 10–16. University of Tartu Library, 2023. URL https://aclanthology.org/2023.nodalida-1.2/.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014. doi: 10.1002/asi.23062.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. HateXplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875, 2021. doi: 10.1609/aaai.v35i17.17745.

Jungho Moon, Dongwoo Yoo, Xiaoqian Jiang, and Miran Kim. THOR: Secure transformer inference with homomorphic encryption. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 3765–3779. ACM, 2025. doi: 10.1145/3719027.3765150.

Lucien K. L. Ng and Sherman S. M. Chow. SoK: Cryptographic neural-network computation. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 497–514. IEEE, 2023. doi: 10.1109/SP46215.2023.10179483.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, 2022. doi: 10.1162/coli_a_00458.

Ildikó Pilán, Benet Manzanares-Salor, David Sánchez, and Pierre Lison. Truthful text sanitization guided by inference attacks. *Applied Soft Computing*, 185:114013, 2025. doi: 10.1016/j.asoc.2025.114013.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388. arXiv preprint.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1586–1596. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1187.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019. URL https://arxiv.org/abs/1910.01108. arXiv preprint.

Chaitanya Shivade. MedNLI – a natural language inference dataset for the clinical domain. *PhysioNet*, 2019. doi: 10.13026/C2RS98. Version 1.0.0.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.127.

Xovee Xu, Jiayang Li, and Fan Zhou. MDCC: A multimodal dynamic dataset for donation-based crowdfunding campaigns. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 5417–5421. ACM, 2023. doi: 10.1145/3583780.3615124.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5364–5376. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.300.

Ziqian Zeng, Jianwei Wang, Junyao Yang, Zhengdong Lu, Haoran Li, Huiping Zhuang, and Cen Chen. PrivacyRestore: Privacy-preserving inference in large language models via privacy removal and restoration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10821–10855. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.532.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. CipherPrune: Efficient and scalable private transformer inference. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 52967–52985, 2025. URL https://openreview.net/forum?id=mUMvr33FTu.

Itamar Zimerman, Moran Baruch, Nir Drucker, Gilad Ezov, Omri Soceanu, and Lior Wolf. Converting transformers to polynomial form for secure inference over homomorphic encryption. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62803–62814. PMLR, 2024. URL https://proceedings.mlr.press/v235/zimerman24a.html.

A Datasets and Preprocessing

Sources. MedNLI (Romanov & Shivade, 2018) is the PhysioNet v1.0.0 release (Shivade, 2019). Bias in Bios (De-Arteaga et al., 2019) is loaded from `LabHC/bias_in_bios`. FinPhraseBank is the Financial PhraseBank (Malo et al., 2014), loaded from `takala/financial_phrasebank`, configuration `sentences_allagree`. GoEmotions (Demszky et al., 2020) is loaded from the `simplified` configuration and MRPC (Dolan & Brockett, 2005) through GLUE (Wang et al., 2019). MDCC (Xu et al., 2023) comes from the authors' GoFundMe release, BANKING77 (Casanueva et al., 2020) from the authors' repository (`PolyAI-LDN/task-specific-datasets`), and HateXplain (Mathew et al., 2021) from `Hate-speech-CNERG/hatexplain`. Drug Reviews is `lewtun/drug-reviews`, a redistribution of the UCI Drug Review data introduced by Gräßer et al. (2018). ADR is `vic35get/adverse_drug_reaction`, a class-balanced subset of the ADE-Corpus-V2 classification data (Gurulingappa et al., 2012). The names ADR, Symptom2Dx, MentalHealth and RedditMH are ours. MentalHealth and RedditMH have no published description, so we name the distributions we loaded, `btwitssayan/sentiment-analysis-for-mental-health` and `solomonk/reddit_mental_health_posts`. Neither label set is a clinical annotation. RedditMH's labels identify the subreddit a post came from, and MentalHealth's are the distribution's own seven status categories. Symptom2Dx is `gretelai/symptom_to_diagnosis`, an LLM-rewritten version of the Symptom2Disease corpus.

Splits. Duplicate texts are removed within each dataset across its splits. Datasets released with training, validation and test splits (ADR, MedNLI, Bias in Bios, GoEmotions, HateXplain, MRPC) keep them. The others are pooled across their published splits, if any, and split 80/10/10 with the label proportions preserved, except Symptom2Dx, released with training and test splits only. An 80/10/10 split would leave it 97 test inputs, fewer than the 128 per dataset that the RoBERTa-base full-layer encryption evaluation uses, so it keeps its published splits and holds out 10% of its training split for validation. The training splits of Bias in Bios and RedditMH are capped at 50,000 inputs, and the validation and test splits of Bias in Bios at 5,000 and 10,000, each a fixed-seed sample (Table 4).

Filtering and truncation. Drug Reviews, RedditMH and MDCC drop texts outside a length range, and MDCC drops appeals not written in English. Inputs are then truncated to 128 tokens, which shortens mainly MDCC, RedditMH and MentalHealth.

B Designation Pipeline and Human-Annotation Check

The designation pipeline of §4 has two layers with different roles. The rule layer is deterministic. Every position it matches is designated, so what it covers can be checked list by list. The named-entity layer adds positions the rules miss, such as names, organizations and medical terms that no

Table 4: The thirteen datasets with test-split statistics under the RoBERTa tokenizer. *Mean tokens*: tokens the model reads (at most 128, special tokens included). *Kept*: percentage of an input's tokens left after truncation. *Masked*: percentage of tokens masked under the sensitive masking condition. *Masked inputs*: percentage of inputs with at least one masked token.

Dataset	Task	Train / valid / test	Labels	Mean tokens	Kept	Masked	Masked inputs
Medical							
ADR	adverse-drug-reaction sentence	1,630 / 219 / 220	2	17.2	100.0	51.7	96.4
Drug Reviews	review → condition (top 10)	27,133 / 3,385 / 3,385	10	82.7	99.3	17.0	99.3
MedNLI	sentence pair → entailment	11,218 / 1,393 / 1,421	3	35.4	99.9	31.3	98.9
Symptom2Dx	symptoms → diagnosis	764 / 84 / 212	22	37.9	100.0	20.9	100.0
Mental health							
MentalHealth	text → mental-health status	40,463 / 5,056 / 5,056	7	73.7	85.1	9.9	80.5
RedditMH	post → subreddit	50,000 / 8,475 / 8,475	5	106.0	73.1	8.2	93.5
General							
BANKING77	query → banking intent	10,479 / 1,271 / 1,271	77	15.5	100.0	2.3	22.5
Bias in Bios	biography → profession	50,000 / 5,000 / 10,000	28	79.1	98.0	25.8	97.6
FinPhraseBank	sentence → sentiment	1,806 / 225 / 225	3	29.5	100.0	24.9	87.1
GoEmotions	comment → emotion (single-label)	35,263 / 4,444 / 4,489	28	18.7	100.0	6.3	44.9
HateXplain	post → hate, offensive, normal (majority)	15,327 / 1,921 / 1,920	3	30.2	100.0	14.9	83.1
MDCC	appeal → hardship type (Medical excluded)	8,844 / 1,104 / 1,099	4	116.8	66.8	15.2	98.9
MRPC	sentence pair → paraphrase	3,655 / 408 / 1,724	2	53.6	100.0	15.7	92.1

list contains, but gives no such guarantee. The pipeline is *fixed*. Its procedure and settings are frozen across every experiment reported in this paper, and every masked condition records the rule version, the detector names and their thresholds in the dataset manifest of the supplementary package. The parts that came from inspecting data are noted below.

Rule layer. The rule layer consists of dictionaries and pattern rules. The dictionaries hold UMLS (Bodenreider, 2004) surface forms (release 2022AB) of 14 clinical semantic types, restricted with `wordfreq`[1] to words rarer than about six occurrences per million words (Zipf frequency 3.8), and nine category lists: 128 common sensitive words that this frequency cut would drop, such as *depression*, of which 19 were added after inspecting data; occupations; four closed lists for sexual orientation and gender identity, religion, political opinion and trade-union membership; family relations; nationality and ethnicity terms; and 30 phrases of suicidal ideation. The occupation list combines O*NET Database 29.1[2] occupation titles and alternate titles with the 28 Bias in Bios professions and their synonyms. The pattern rules match e-mail addresses, URLs and bare domains, user handles, phone and social-security numbers, ZIP codes, ages, dates and weekdays, case numbers (a rule added after inspecting court documents), alphanumeric codes such as tickers, doses and currency amounts, numbers with thousands separators, names introduced with "my name is", and strings of seven or more digits such as provider numbers. Year ranges are exempt from the digit rule, since they mark dates such as birth and death years rather than identifiers.

Named-entity layer. Two pretrained detectors add positions the rules miss. The spaCy model `en_core_web_md`[3] covers eight entity types (person, organization, geopolitical entity, location, facility, nationality or group, money and percent) plus dates that contain a year, and the GLiNER checkpoint `gliner-community/gliner_large-v2.5` (Zaratiana et al., 2024) is prompted with eight medical categories (disease, symptom, medication, medical treatment, mental health condition, substance use, injury and reproductive health) at a fixed threshold of 0.5. On MRPC, the numeric parts of both layers (the rules for ages, dates, codes, amounts and other numbers, as well as

[1] `https://github.com/rspeer/wordfreq`

[2] `https://www.onetcenter.org/dictionary/29.1/text/`

[3] `https://spacy.io/models/en#en_core_web_md`

money, percent and date entity types) are turned off so that paraphrase pairs keep the numbers the task depends on. Phone and social-security numbers stay masked.

Coverage of human annotations. We compare the designations with the Text Anonymization Benchmark (TAB; Pilán et al., 2022), 1,268 human-annotated court documents (Table 5). A span counts as annotated when a majority of its document's annotators marked it as a direct or quasi-identifier. The pipeline covers 74.0% of these spans in full (complete recall) and 92.2% at least in part (partial recall), while masking 1.64 times as much text as the annotators. Of 4,177 direct identifiers, 48.8% are fully covered and another 49.2% partially covered. Thus, broad overall coverage does not ensure that each identifier is fully hidden. This is a limitation of automatic designation, not of the merger. The training split of TAB (1,014 of the 1,268 documents) was also used to compare detector options, and that comparison kept the existing detector.

Table 5: Coverage of human-annotated spans in the Text Anonymization Benchmark (1,268 documents, all splits) by the designation pipeline. n: number of annotated spans.

Entity type	n	Complete recall (%)	Partial recall (%)
DATETIME	31,503	94.6	96.4
PERSON	12,783	33.6	96.1
ORG	9,518	77.7	93.3
LOC	4,524	85.8	95.0
CODE	2,727	85.8	88.1
DEM	2,408	64.2	82.1
MISC	2,392	21.6	45.4
QUANTITY	2,067	25.7	64.9
By identifier type			
Quasi-identifiers	63,745	75.7	91.8
Direct identifiers	4,177	48.8	98.0
All	67,922	74.0	92.2

C Training Configuration

Within each dataset and backbone, all models share the splits, the 128-token input limit, the base learning rate, batch size, epoch cap, early-stopping rule and LoRA configuration (Table 6), and the masked baseline models also share their masks. Settings are not tuned separately for each model.

Table 6: Per-dataset batch size, epoch cap and base learning rate for both backbones.

			Base learning rate	
Dataset	Batch	Epoch cap	RoBERTa	ModernBERT
Medical				
ADR	128	50	2×10^{-4}	2×10^{-4}
Drug Reviews	256	24	2×10^{-4}	2×10^{-4}
MedNLI	256	60	5×10^{-4}	5×10^{-4}
Symptom2Dx	64	60	2×10^{-4}	2×10^{-3}
Mental health				
MentalHealth	256	24	2×10^{-4}	2×10^{-4}
RedditMH	256	24	2×10^{-4}	2×10^{-4}
General				
BANKING77	256	30	2×10^{-4}	2×10^{-4}
Bias in Bios	256	20	2×10^{-4}	2×10^{-4}
FinPhraseBank	256	60	5×10^{-4}	5×10^{-4}
GoEmotions	256	30	2×10^{-4}	2×10^{-4}
HateXplain	256	24	2×10^{-4}	2×10^{-4}
MDCC	256	24	2×10^{-4}	2×10^{-4}
MRPC	256	40	2×10^{-4}	2×10^{-4}

Trained parameters. All models train LoRA adapters (Hu et al., 2022) (rank 8, $\alpha = 16$, dropout 0.1) and the task head, and LFT and ZEROZ also train the merger. On RoBERTa-base, LoRA attaches to the query and value projections and adds $294{,}912$ parameters (0.237% of the backbone). On ModernBERT-base, it attaches to the fused query/key/value projection, the attention, and feed-forward output projections, and adds $1{,}148{,}928$ parameters (0.771%). RED, LFT and ZEROZ also train the embedding of `[REDACTED]`. A gradient mask isolates that row and all other embedding rows receive neither gradients nor weight decay.

Optimization. Every trainable parameter takes the same per-dataset base rate in every model. RoBERTa uses 2×10^{-4}, except 5×10^{-4} on MedNLI and FinPhraseBank, and ModernBERT inherits these rates. The one exception is ModernBERT on Symptom2Dx, which uses 2×10^{-3}, chosen on validation accuracy. We use AdamW with weight decay 0.1 and a linear schedule with warmup over the first 6% of steps. Training runs for at most the epoch cap and stops after ten epochs without validation improvement. The checkpoint with the best validation accuracy is restored and evaluated on the test split. Each benchmark uses seeds 0 through 4.

D Ablation on Masking Conditions

The main results mask the sensitive content of each input (i.e., the *sens* condition). Two further conditions, run on RoBERTa-base, ask what the gain of LFT depends on. The *chaff20* asks whether the gain persists when more of the input is designated and sent through the encrypted path. The *rnd* asks whether the gain depends on which positions are masked or only on how many.

Masking conditions. The *chaff20* keeps the masks of *sens* and, on inputs where masked tokens make up less than 20% of the tokens (special tokens excluded), masks randomly chosen words until that share is reached. Inputs already at or above it are left unchanged. The added words enter the same encrypted input path as the sensitive ones. The *rnd* is the random control of *sens*: it masks as many tokens as *sens* did at random positions. We performed two random mask selections, each trained under five seeds. Thus, the gain comes from ten runs for *rnd*.

Results. Table 7 gives, for each condition, the accuracy that masking removes (ORIG $-$ RED) and the gain of LFT (LFT $-$ RED). Designating more input tokens does not remove the gain. Under *chaff20* the gain stays positive on every dataset where it is positive under *sens*, and where *chaff20* masks much more than *sens*, most visibly on BANKING77, there is more to recover and the gain grows with the loss. On most datasets, the gain depends on which positions are masked rather than on how many. Masking as many random tokens as *sens* costs RED little, so there is little for LFT to recover. The large *sens* gains therefore come from masking tokens that carry the task-relevant signal. Across the three conditions, the gain follows the loss and what differs between them is how much each masking removes. The exceptions are ADR, MedNLI and FinPhraseBank, whose inputs are short and heavily masked; masking even random tokens removes task-relevant signal, and LFT recovers part of it.

E Ablations on Merger Design

The merger combines the designated tokens through softmax cross-attention whose query comes from the plaintext context. Table 8 replaces this cross-attention with three alternatives, with the rest of the model, the training procedure and the seeds unchanged, to test whether cross-attention is the best of these ways of combining the designated tokens. All runs are on RoBERTa-base.

The alternatives. *Mean pooling* projects designated and gathered embeddings to values and averages them along the token dimension, without attention weights. Then, the resulting vector goes through the output projection layer. *Matched pooling* adds a two-layer MLP on the averaged value vector prior to the output projection layer of the mean pooling so that its parameter count equals the cross-attention's. *Linear* keeps the cross-attention but removes its softmax, using the scaled dot-product scores divided by the number of designated tokens directly as weights.

Table 7: Masking conditions on RoBERTa-base with *sens* for reference (accuracy points, five-seed means). Loss: ORIG − RED. Gain: LFT − RED, mean ± standard deviation of the seed-paired differences.

		sens		chaff20		rnd	
Dataset	ORIG	Loss	Gain	Loss	Gain	Loss	Gain
Medical							
ADR	92.3	9.6	+5.3 ±1.7	9.2	+4.6 ±0.9	15.3	+7.7 ±2.6
Drug Reviews	85.7	16.0	+15.3 ±0.6	16.3	+15.3 ±0.4	2.2	+1.4 ±0.4
MedNLI	80.2	13.0	+1.3 ±1.4	14.3	+3.1 ±1.7	18.2	+4.6 ±1.0
Symptom2Dx	95.7	18.0	+15.8 ±3.4	18.2	+16.3 ±2.4	4.9	−0.4 ±2.1
Mental health							
MentalHealth	82.6	5.6	+4.3 ±0.6	6.7	+4.9 ±0.3	1.5	+0.2 ±0.5
RedditMH	84.6	12.2	+11.2 ±0.4	14.4	+12.7 ±0.3	0.9	+0.1 ±0.3
General							
BANKING77	91.8	1.9	+3.3 ±0.4	12.6	+12.0 ±0.6	1.0	+1.9 ±0.3
Bias in Bios	84.4	6.1	+4.1 ±0.2	6.6	+4.2 ±0.2	2.9	+0.6 ±0.2
FinPhraseBank	97.6	−0.4	+0.1 ±0.6	0.7	+0.4 ±0.8	6.8	+4.5 ±1.1
GoEmotions	62.4	1.2	−0.4 ±0.6	5.8	+2.7 ±0.7	1.8	−0.3 ±0.7
HateXplain	69.9	7.0	+4.2 ±0.4	9.5	+7.2 ±0.9	3.7	+1.0 ±0.7
MDCC	88.4	1.4	+0.7 ±0.6	1.7	+0.5 ±0.6	0.4	−0.8 ±0.7
MRPC	86.3	1.3	−3.9 ±0.8	3.1	−2.6 ±1.4	2.8	−3.6 ±0.7

Table 8: Three replacements for the merger's cross-attention on the thirteen datasets (accuracy in %, mean of five seeds). The last three columns give cross-attention minus each variant. The cross-attention baseline is the RoBERTa-base run of Table 1.

	Variant accuracy			Cross-attention − variant		
Dataset	Mean	Matched	Linear	Mean	Matched	Linear
Medical						
ADR	85.4	85.7	87.6	+2.5	+2.2	+0.3
Drug Reviews	84.1	83.8	84.7	+1.0	+1.3	+0.3
MedNLI	68.4	71.1	69.7	+0.1	−2.6	−1.2
Symptom2Dx	92.5	92.6	92.7	+0.9	+0.8	+0.8
Mental health						
MentalHealth	81.5	81.3	81.3	−0.1	+0.1	+0.1
RedditMH	82.6	82.8	83.4	+1.0	+0.8	+0.2
General						
BANKING77	92.8	92.6	92.8	+0.3	+0.6	+0.4
Bias in Bios	81.6	81.7	82.1	+0.9	+0.8	+0.4
FinPhraseBank	97.9	98.1	98.0	+0.3	+0.0	+0.2
GoEmotions	60.9	60.8	60.5	+0.1	+0.1	+0.4
HateXplain	66.2	66.5	66.7	+0.8	+0.5	+0.4
MDCC	87.6	88.2	87.2	+0.2	−0.5	+0.6
MRPC	82.9	80.9	80.9	−1.8	+0.2	+0.2

Results. Cross-attention is more accurate than each alternative on most datasets: on 11 of the 13 against mean pooling, on 10 against matched pooling and on 12 against linear. The margin over the two pooling variants is largest on ADR, about two percent points. Matched pooling, which has as many parameters as the cross-attention, does not close the gap, so the margin does not come from capacity. All three alternatives also remain below RED on MRPC, so the MRPC deficit of LFT is not specific to cross-attention.

F Additional Re-identification Results

The §4.3 reports two fixed attackers. This appendix lists the ten attacks we ran on every dataset and all of their reconstruction rates (Table 9). The sequential, voting and LLM attacks follow Charpentier & Lison (2025b). Each attack reads only the redacted text and guesses every masked span.

Table 9: Percentage of masked spans reconstructed exactly from the redacted text by each attack, under the sensitive token masking. Bold marks the strongest attack on each dataset. *Spans* is the number of evaluated spans.

	Medical				Mental health		General						
Attack	ADR	DrugRev	MedNLI	Sym2Dx	MentalH	RedditMH	BANK77	Bios	FinPB	GoEmo	HateX	MDCC	MRPC
RoBERTa-base													
Single	1.1	13.1	3.2	16.1	14.2	14.0	9.5	14.0	9.3	5.9	3.0	13.5	17.0
Sequential	1.4	14.3	3.2	18.9	15.6	15.6	9.2	15.0	7.9	6.0	3.3	14.8	18.6
Vote	1.4	14.2	3.3	18.0	15.6	15.4	9.0	15.1	8.2	6.2	3.1	14.8	18.2
Sequential filling, other backbones													
ModernBERT-large	1.2	**22.8**	**5.8**	**28.8**	**23.5**	**24.9**	**14.0**	**20.0**	**11.7**	**8.5**	**5.8**	**18.8**	21.4
BioMed-RoBERTa	**3.5**	10.2	4.9	15.7	9.3	9.3	7.6	11.4	6.8	2.9	1.1	8.9	10.6
Twitter-RoBERTa	0.9	10.3	2.2	15.8	11.1	10.5	10.4	9.4	6.1	6.2	3.2	9.9	10.8
RoBERTa-large	1.4	18.5	3.9	23.8	20.4	20.6	**14.0**	18.1	10.3	8.0	4.6	18.3	**22.8**
DistilRoBERTa	1.2	11.3	3.2	17.7	12.4	13.7	10.1	12.2	6.1	6.0	2.8	11.5	16.9
LLM													
Qwen3-4B-Instruct	0.4	8.1	3.0	9.7	7.3	7.4	5.9	6.5	2.0	3.1	1.4	5.6	4.8
Qwen3.5-9B	0.7	11.3	3.1	16.4	10.2	10.2	5.9	7.9	2.8	3.9	2.8	6.9	6.5
Spans	564	25,606	6,465	1,064	18,281	44,229	357	101,883	897	3,013	3,698	10,860	7,788

Attack	How the masked spans are filled
RoBERTa-base	
Single	all at once, so none sees the guesses for the others
Sequential	one at a time, most confident first, reusing earlier guesses
Vote	sequential in five orders, then a probability-weighted vote
Sequential filling, other backbones	
ModernBERT-large	a larger and newer masked language model (Warner et al., 2025)
BioMed-RoBERTa	RoBERTa adapted to biomedical papers (Gururangan et al., 2020)
Twitter-RoBERTa	RoBERTa adapted to tweets (Barbieri et al., 2020)
RoBERTa-large	the larger RoBERTa (Liu et al., 2019)
DistilRoBERTa	a distilled, smaller RoBERTa (Sanh et al., 2019)
LLM	
Qwen3-4B-Instruct	one span per prompt with the rest left masked, filled by an LLM (Qwen Team, 2025)
Qwen3.5-9B	the same prompt with a newer and larger LLM (Qwen Team, 2026)

One of the two fixed attackers, ModernBERT-large or RoBERTa-large, is the strongest attack on twelve of the thirteen datasets. The exception is ADR, where BioMed-RoBERTa recovers 3.5% of the spans against at most 1.4% for the fixed attackers.

A span counts as recovered only when the guess matches it exactly. Because the attackers use different tokenizers, guesses are compared as decoded strings: both gold and predicted spans are NFKC-normalized, lowercased and stripped of non-word characters, so spelling and diacritics affect a match but case, spacing and punctuation do not.

G Encrypted Implementation and Evaluation

This appendix details the security parameters, merger packing, full-layer numerical plans, and cost measurements of §5.

Parameters and security. The LFT merger and both full-layer baselines use the CKKS parameters in §5. For the main uniform-ternary secret key, $N = 2^{16}$ and $\log QP = 1{,}732$ bits fall within the 128-bit classical-security bound of Bossuat et al. (2025). This parameter bound only relates to the main-key setting; it does not by itself assess the temporary sparse key used during bootstrapping. FIDESlib's `ENCAPS` procedure switches a ciphertext at the smallest, 56-bit modulus to a ternary key of Hamming weight 32, raises the modulus, and switches it back to the main key before the remaining bootstrap operations. This temporary-key and key-switching procedure follows the sparse-secret encapsulation construction of Bossuat et al. (2022). The uniform-ternary guideline does not cover the sparse-key setting, so we do not apply its 128-bit bound to the intermediate ciphertext.

G.1 Encrypted Merger: Packing and Fidelity

Figure 4c summarizes the three layouts of our merger with stage-specific batching. A ciphertext has 32,768 slots, and one input uses a different share of them in each stage: the key and value projections carry two inputs, while the attention score, softmax, output projection, and layer norm carry 32 inputs.

Key and value projections (2 inputs). Each input supplies at most 64 designated-token embeddings, with unused rows set to zero. Two 64-row matrices occupy the first and second halves of a 128-row layout spread over four ciphertexts. The diagonal method multiplies weight diagonals with rotated copies of the input ciphertexts (shifts by multiples of 1,024 slots). These copies are computed once and shared by the key and value projections and by both inputs. Masks then split each projected ciphertext into the rows of the two inputs, and the second input's rows are rotated into the positions of the first so that both inputs continue in the same layout.

Attention scores and softmax (32 inputs). One input's twelve heads over 64 keys occupy a 1,024-slot block, so 32 distinct inputs fill the 32,768 slots of one ciphertext. A final partial batch fills its unused blocks with copies of its first input. To control bootstrap error in this full-slot layout, we bootstrap twice at three points of the softmax: after the exponential, and in each squaring step at the input of the inverse square root and at the normalized square. Following Bae et al. (2022), the second bootstrap acts on the error of the first, scaled by 2^6, and its scaled-back result is subtracted. To keep the level budget, the scaling applied before the bootstrap of the inverse square root's output is folded into that polynomial's coefficients. In a synthetic test, the two-pass bootstrap lowers the error of a full-slot bootstrap about 60-fold. After softmax, each input's block is extracted, multiplied by its values, and reduced to one 768-wide CLS row.

Output projection and layer norm (32 inputs). The CLS rows use a stride of 768 slots. Although 42 rows fit in 32,768 slots, the implementation uses 32, one row per input of a score batch. The server applies output projection, plaintext–ciphertext addition, and layer norm to this batch; the client decrypts the resulting outputs. Like the attention score stage, this stage uses the full 32,768-slot bootstrap configuration.

Padding, calibration, and fidelity. Padding keys are excluded from attention in both realizations. The plaintext path masks their scores before softmax; the encrypted path multiplies their exponentials by a plaintext binary mask. If no token is designated, the zero-filled private input yields an input-independent cross-attention term and the prediction depends on the plaintext path.

The softmax plan chooses a squaring count K and a score bound B for $y_0 = \exp((s - B)/2^K)$ for each dataset. For each candidate K, the calibration pool drawn from the training split fixes B; the smallest K whose score and normalization ranges cover the evaluated inputs is retained. This last check uses the inputs' numerical ranges, not their labels, as also explained in §6. Table 10 reports the chosen constants, prediction agreement, and decrypted-vector error on each full test split. Agreement is at least 99.8% in every dataset.

G.2 Full-Layer HE Baselines: Conversion and Fidelity

The full-layer baselines carry all 128 token positions through twelve RoBERTa-base or twenty-two ModernBERT-base layers. They retain the original nonlinear functions during fine-tuning and use polynomial approximations during encrypted evaluation. Table 11 records the different training controls and offline numerical-calibration rules. Both baselines multiply ciphertexts in the score product $\mathbf{QK}^T$ and in the product of the attention weights with $\mathbf{V}$. ModernBERT-base evaluates them with a baby-step giant-step decomposition of the diagonal rotations, the strategy THOR uses for ciphertext–ciphertext products (Moon et al., 2025), and folds the rotary position encoding of the queries into the giant-step masks. RoBERTa-base rotates by each diagonal directly. On one ADR input, the decomposition made the score product faster but the value product slower, and the two products took comparable time over twelve layers (35.4 s with the decomposition and 34.6 s without it).

Table 10: Encrypted merger fidelity on each full test split using the seed-0 LFT checkpoint. (K, B) are the per-dataset softmax constants. Agreement compares encrypted and plaintext predictions on the same inputs. Relative error, in units of 10^{-2}, is $\|\tilde{\mathbf{y}}_{\text{merger}} - \mathbf{y}_{\text{merger}}\|_2 / \|\mathbf{y}_{\text{merger}}\|_2$ for the decrypted merger output $\tilde{\mathbf{y}}_{\text{merger}}$ and its plaintext value $\mathbf{y}_{\text{merger}}$.

Dataset	n	(K, B)	Agreement (%)	Rel. err. (10^{-2})	
				Median	Maximum
ADR	220	(2, 6)	100.0	1.5	1.9
Drug Reviews	3,385	(3, 13)	99.8	2.4	5.9
MedNLI	1,421	(5, 41)	100.0	0.9	1.5
Symptom2Dx	212	(2, 7)	100.0	3.5	10.4
MentalHealth	5,056	(4, 24)	100.0	1.1	3.3
RedditMH	8,475	(4, 25)	100.0	1.2	3.2
BANKING77	1,271	(3, 11)	99.9	2.2	5.1
Bias in Bios	10,000	(4, 26)	99.9	1.3	3.5
FinPhraseBank	225	(3, 11)	100.0	0.9	1.3
GoEmotions	4,489	(3, 12)	99.8	1.1	1.7
HateXplain	1,920	(4, 24)	99.9	1.2	1.6
MDCC	1,099	(4, 27)	100.0	1.1	2.7
MRPC	1,724	(4, 21)	100.0	1.7	2.5

Table 11: Offline training and numerical-calibration rules for the two full-layer HE baselines. Margins expand observed numerical ranges.

	RoBERTa-base	ModernBERT-base
Training	Clip feed-forward pre-activations in layers 8–9 to $[-30, 30]$; penalize pre-LN variance above 3 ($\lambda = 0.02$).	No clipping; pre-activation cap 20 ($\lambda = 0.1$) and variance-ratio cap 256 ($\lambda = 0.5$) are penalty thresholds.
Softmax	Choose $K \geq 4$ and B to keep normalization sums inside the calibrated interval, widening its upper bound when necessary.	Choose per-layer softmax windows with $K \geq 4$ and expand the normalization window where needed.
Layer norm	Chebyshev inverse square root, degrees 8–32 and 0–4 Newton steps; widen the window while meeting relative error ≤ 0.005.	Lower margin $4\times$; upper margin the largest of $\{1.15, 1.3, 1.6, 2, 3, 4\}\times$ meeting relative error ≤ 0.005; degrees $12, 24, 32$, with the planned Newton steps plus three refinements.
GELU	Asymmetric window $1.35\times$ the observed range; degree at most 56, with a histogram-weighted least-squares fit.	Base window margin $1.4\times$; separately calibrated wide windows for selected high-activation rows.

Training and numerical plans. The RoBERTa-base baseline is fine-tuned separately with feed-forward pre-activation clipping in layers 8 and 9 (zero-based indices) and a penalty on excess pre-layer-norm variance. Its plaintext fidelity reference is the unclipped export of that same checkpoint. ModernBERT-base instead uses pre-activation and variance-ratio penalties across its layers, with no runtime clipping. These controls are specific to the full-layer baselines; the LFT merger's backbone runs in plaintext. Training to reduce activation ranges has precedent in HE-oriented Transformer conversion (Zimerman et al., 2024). For RoBERTa-base, a seed-0 comparison with and without activation control found a test-accuracy change below one point on ten of thirteen datasets; the largest changes were -4.7 points on Symptom2Dx and $+2.3$ on ADR.

Initial numerical ranges use 512 calibration inputs from the validation split, topped up from training for small datasets. The final window rules were refined after inspecting evaluation-input ranges without their labels. Consequently, the reported fidelity measures these evaluated inputs, not the rate at which future inputs will stay within the fixed windows (§6). The RoBERTa plan jointly selects inverse-square-root polynomial degree and Newton steps to meet relative error 0.005 with the fewest levels, then widens the layer-norm domain as far as that choice permits. Its GELU uses an asymmetric $1.35\times$ observed-range window and a histogram-weighted least-squares fit. The ModernBERT plan uses a $1.4\times$ base GELU window and selects the largest admissible upper margin separately

for each layer norm, while keeping a $4\times$ lower margin. Those margins are fixed in the plan before encrypted execution; they do not adapt to an input at runtime.

Fidelity. Before encrypted evaluation, a plaintext audit compares the selected inputs with the fixed approximation windows. Table 12 compares each encrypted prediction with the corresponding plaintext prediction of the same checkpoint. RoBERTa-base agrees on 1,649 of 1,664 inputs (128 per dataset), and ModernBERT-base on 403 of 416 (32 per dataset); neither run has an execution failure.

Table 12: Encrypted predictions that match the plaintext predictions of the same full-layer checkpoint, out of 128 inputs per dataset for RoBERTa-base and 32 for ModernBERT-base.

	Medical				Mental health		General						
Model	ADR	DrugRev	MedNLI	Sym2Dx	MentalH	RedditMH	BANK77	Bios	FinPB	GoEmo	HateX	MDCC	MRPC
RoBERTa	126	128	126	125	127	128	128	128	128	125	126	128	126
ModernBERT	31	30	30	31	32	32	32	30	32	32	29	32	30

G.3 Cost Measurement and Communication

What each Table 3 row measures. The one-time server model-preparation row includes cold encoding of distinct weights and model-dependent precomputation. For the merger, this is mainly the encoding of the THOR diagonal weights. It excludes CKKS context construction, key generation, and bootstrap setup, which are not silently included in the server-time rows. Client encoding and encryption, including serialization, is timed on the server host (Intel Xeon 6960P, eight OpenMP threads) and reported as a per-input median.

Warm encrypted server time covers only the encrypted computation. It includes per-input reads of encoded weights from the host cache; the full-layer weights do not fit on the GPU and are read again for each input. It excludes plaintext LFT computation, communication, client preparation, one-time model preparation, and the client-side task head. For each full-layer baseline, the sixteen-input run processes the inputs one by one within each layer, then moves to the next layer, and its reported time is divided by sixteen. The merger's amortized time uses the stage-specific batch sizes in §G.1. Table 13 gives the warm time for each dataset; Table 3 takes the equal-weight mean over the thirteen datasets. The merger's amortized server time increases by about 5.5 ms per input for each additional softmax squaring. Figure 2b uses the RoBERTa-base implementation for its corresponding plaintext and encrypted operator timings, with the encrypted median taken over 19,968 input–layer pairs.

Table 13: Dataset-level warm encrypted server times underlying Table 3. Latency is one input; amortized time divides a batch by its input count. K is the merger's softmax squaring count. Model preparation, client preparation and communication are excluded.

		Merger (s)		Full-layer RoBERTa (s)		Full-layer ModernBERT (s)	
Dataset	K	Latency	Amortized	Latency	Amortized	Latency	Amortized
ADR	2	1.010	0.101	236.8	191.8	372.4	331.6
Drug Reviews	3	1.206	0.111	242.6	202.8	382.4	334.9
MedNLI	5	1.571	0.115	251.0	199.0	406.8	332.5
Symptom2Dx	2	1.003	0.102	250.9	185.1	395.3	323.8
MentalHealth	4	1.376	0.116	276.8	210.9	399.2	332.6
RedditMH	4	1.394	0.114	250.7	206.0	382.0	330.6
BANKING77	3	1.197	0.105	244.9	208.2	398.4	340.9
Bias in Bios	4	1.386	0.117	242.3	191.2	389.9	328.1
FinPhraseBank	3	1.198	0.108	230.0	195.5	408.9	340.9
GoEmotions	3	1.212	0.110	221.9	193.6	393.3	324.0
HateXplain	4	1.381	0.115	245.8	202.0	397.9	347.3
MDCC	4	1.380	0.113	237.6	201.9	400.7	332.7
MRPC	4	1.387	0.115	220.9	189.1	376.3	320.4

Communication and keys. Upload and download sizes are the serialized ciphertext sizes at their respective transmission levels. Key material is measured by serializing the final public key, relinearization key, and rotation-key map, once per client. The merger has 210 rotation and bootstrapping keys and 21.2 GB of serialized key material, the RoBERTa baseline 325 and 32.7 GB, and the ModernBERT baseline 115 and 11.6 GB. Each implementation registers only the keys its execution path uses. The ModernBERT baseline needs the fewest because its baby-step giant-step decomposition replaces the per-diagonal rotations of the RoBERTa baseline with baby and giant steps. Every batched input uses the same client's key. In the merger's measured batches, two inputs share upload ciphertexts (19 MB per input), and 32 outputs share one ciphertext (0.03 MB per input).