SELECTIVE HOMOMORPHIC INFERENCE VIA ENCRYPTION BOUNDARY SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Homomorphic encryption (HE) can protect data during inference, but encrypted computation remains costly for transformer models. We study how the choice of encryption boundary affects representation exposure, task performance, and latency. We compare three strategies across DistilBERT, Mistral-7B, and Qwen-2.5-3B on multiple NLP tasks: encrypting only the classifier head, encrypting a representation for transfer to an assumed trusted component, and performing projection and prediction over ciphertext. Head-only encryption leaves informative plaintext representations exposed, while deeper encrypted computation incurs substantial latency and, in our experiments, often reduces accuracy. Encrypting a pooled representation for transfer, then decrypting it for downstream plaintext computation inside the trusted component, offers a favorable tradeoff among the evaluated strategies. On Mistral-7B with GoEmotions, this approach achieves 55.81% Micro-F1 versus 58.30% for plaintext inference, with 26.60 ms measured latency. These results suggest that pooled and moderately projected representations are promising boundary choices under the stated trust model.

1 INTRODUCTION

The growing deployment of machine learning (ML) models in cloud-based services has raised critical concerns about data privacy during inference (Rigaki & Garcia, 2023). In many real-world applications, such as sentiment analysis (Fei et al., 2022) and emotion recognition (Hu et al., 2023), user inputs contain sensitive information that must not be exposed to untrusted servers (Feng et al., 2024; Jaiswal & Provost, 2020). Homomorphic encryption (HE) offers a principled solution by enabling computation directly on encrypted data (De Castro et al., 2025; Yuan et al., 2025). However, its practical adoption remains limited due to the substantial computational overhead associated with encrypted operations (Juvekar et al., 2018).

Recent work has explored applying HE to neural network inference, typically by encrypting model components such as classifier heads or larger portions of the inference pipeline (Juvekar et al., 2018; Lee et al., 2022; Hashimoto et al., 2025; Schneider et al., 2025). While these approaches provide strong privacy guarantees under their respective threat models, they often incur substantial latency, especially for modern transformer architectures. Also, prior work in this area has largely focused on fully encrypted training/inference or improving low-level HE primitives such as efficient matrix multiplication (Dathathri et al., 2019) (e.g., diagonal packing, PC-MM, and CC-MM strategies). However, existing methods often treat the encrypted region as a fixed design choice, without explicitly analyzing how alternative encryption boundaries affect privacy leakage, task utility, and cryptographic cost. As a result, there remains limited guidance for selecting a practical boundary for modern transformer inference.

In this work, we argue that an important challenge in practical HE-based inference is not only how to encrypt computation, but also *where* to place the encryption boundary. We formalize this problem as a boundary optimization objective that jointly captures privacy leakage, task performance, and homomorphic computation cost. We introduce a *selective encryption* framework that systematically evaluates different points in the inference pipeline, including classifier heads, intermediate representations, and deeper computation layers. Through this lens, we study how different boundary choices expose distinct privacy–utility–efficiency tradeoffs and identify pooled and moderately projected representations as practical Pareto-efficient choices among the evaluated candidates.

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.

We instantiate this framework on both compact (DistilBERT) and large-scale transformer models, including Mistral-7B and Qwen-2.5-3B, across multiple datasets such as SST-2, IMDB, GoEmotions, and MNLI. We evaluate three configurations: **HE-Linear** (head-only encryption), **Repr-HE** (representation-level encrypted transmission with trusted downstream execution), and **Proj-Poly-HE** (deeper staged encrypted projection and prediction). These configurations represent different points in the selective-encryption design space rather than equivalent security models.

Our results reveal that head-only encryption provides limited protection for exposed intermediate representations despite additional cryptographic cost, while deeper encrypted computation incurs substantially higher latency and may degrade task performance. In contrast, Repr-HE provides a practical hybrid operating point: the intermediate representation is encrypted while crossing the untrusted boundary and is decrypted only inside an assumed logical trusted component, where downstream computation proceeds in plaintext. Thus, Repr-HE is not an end-to-end fully homomorphic inference pipeline; its confidentiality depends jointly on the encryption scheme and the integrity of the trusted component.

For example, on Mistral-7B, Repr-HE maintains 55.81% Micro-F1 compared to 58.30% for plaintext and achieves an end-to-end latency of 26.60 ms. Our representation-level probing and recovery analyses further show that boundary location substantially affects the intrinsic information recoverable from intermediate representations. These findings demonstrate that carefully selecting the encryption boundary can provide a favorable tradeoff between privacy, task utility, and cryptographic cost. Rather than implying a universally optimal split point, our results suggest that pooled and moderately projected representations form a practical Pareto region under balanced deployment objectives.

Our contributions are summarized as follows:

- We introduce a selective encryption framework and formulate encryption boundary selection as a joint optimization problem over privacy, utility, and homomorphic computation cost.
- We provide a systematic empirical evaluation across models and datasets, showing how head-level, representation-level, and deeper encrypted boundaries expose different privacy–utility–efficiency tradeoffs.
- We identify representation-level encryption as a practical Pareto-efficient solution among the evaluated candidates, achieving limited accuracy degradation and low cryptographic overhead under the hybrid trust model considered in this work.

Privacy is not achieved simply by encrypting more computation, but by selecting an appropriate protection boundary for the deployment setting.

2 Related Work

Homomorphic encryption (HE) has been widely studied for privacy-preserving inference, enabling computation directly on encrypted data (De Castro et al., 2025; Yuan et al., 2025; Fu et al., 2025). Early systems such as CryptoNets (Gilad-Bachrach et al., 2016) and its extensions demonstrated the feasibility of encrypted neural network inference by replacing nonlinear activations with low-degree polynomials and limiting model depth. Subsequent works improved the efficiency of HE primitives, including optimized matrix multiplication (Dathathri et al., 2019) and ciphertext packing schemes (Peng et al., 2025; Boemer et al., 2019). However, these approaches typically focus on fully encrypted pipelines, which incur substantial computational overhead and scale poorly to modern deep architectures.

Recent efforts have extended HE to transformer-based models, including BERT (Lee et al., 2022; Hashimoto et al., 2025; Moon et al., 2025) and related architectures (Juvekar et al., 2018), by encrypting selected components such as classifier heads or intermediate layers. To mitigate the cost of full encryption, hybrid approaches (Juvekar et al., 2018; Schneider et al., 2025) combine plaintext and encrypted computation, applying HE only to specific parts of the model. While these methods reduce latency, they often rely on heuristic choices of what to encrypt and do not explicitly analyze how different components contribute to privacy leakage. As a result, they provide limited guidance for balancing privacy, accuracy, and efficiency, particularly in large language models where both representation dimensionality and computational cost are high.

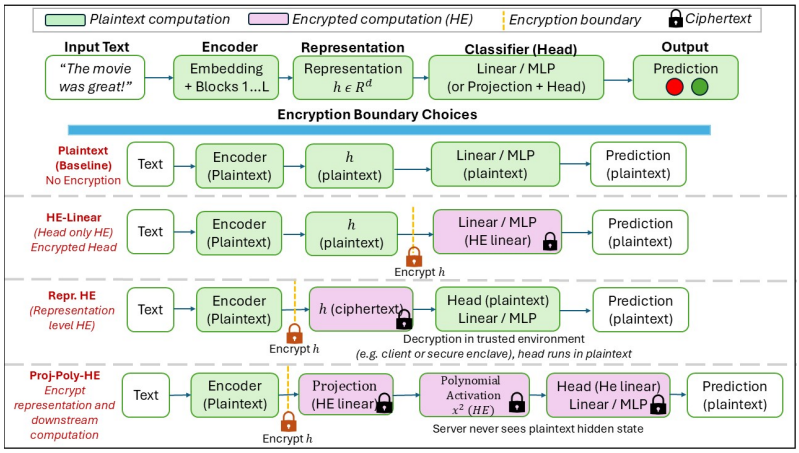


Figure 1: Comparison of encryption boundary choices in transformer inference. Different strategies (Plaintext, HE-Linear, Repr-HE, Proj-Poly-HE) apply encryption at different stages, illustrating tradeoffs between privacy and computation.

Closely related work has studied privacy-aware split inference and representation-level protection. Maeng et al. (Maeng et al., 2022; 2023) use Fisher-information-based analysis to quantify and control the invertibility of split-layer activations and privacy-preserving instance encodings. Their focus is on constructing or evaluating representations that resist reconstruction, whereas our goal is not to design a new representation-encoding mechanism, but to determine *where encryption should be introduced and how computation beyond that boundary should be executed*. Split HE (Pereteanu et al., 2022) combines split inference with homomorphic encryption and studies the runtime and security implications of a client–server partition. In contrast, our formulation jointly considers privacy leakage, task utility, and HE-specific cost, including encryption/decryption overhead, ciphertext operations, and multiplicative depth, while evaluating multiple post-boundary execution strategies. C^2PI (Zhang et al., 2023) is particularly related in its use of privacy attacks to identify a crypto–clear partition; however, it focuses on MPC-based, layer-wise partitioning of CNNs. Our work instead studies HE execution at and beyond the selected boundary for transformer and LLM inference, including token-level, pooled, projected, and encrypted representations. Thus, our contribution is not the general concept of selecting a split layer, but an *HE-aware boundary-selection framework* that explicitly characterizes the privacy–utility–efficiency trade-off across alternative transformer representation boundaries (more in Appendix A.2).

3 System Overview

We consider a client–server inference setting with a pretrained model f_θ. Given input x, prediction is computed as $\hat{y} = f_\theta(x)$. We introduce a *selective homomorphic encryption (HE)* framework in which protection is introduced at a chosen boundary of the inference pipeline rather than uniformly across the entire model. The backbone computation remains in plaintext, while the location and extent of encrypted computation depend on the selected strategy.

We study three configurations: **HE-Linear**, **Repr-HE**, and **Proj-Poly-HE**, implemented using the CKKS scheme. These configurations represent different points in the privacy–utility–efficiency design space and do not assume identical trust models. In particular, Repr-HE is a hybrid encrypted-transmission and trusted-execution configuration rather than an end-to-end HE pipeline. A detailed description of the notation is provided in Appendix E.

Representation Extraction. We extract a fixed-dimensional representation h from the backbone. For encoder models (DistilBERT):

$$h = \mathrm{CLS}(f_\theta(x)) \in \mathbb{R}^{768}. \tag{1}$$

For decoder-only models (Mistral-7B, Qwen):

$$h = \text{LastToken}(f_\theta(x)) \in \mathbb{R}^{4096}. \tag{2}$$

To reduce HE cost, we apply a projection:

$$z = W_{\text{proj}} h, \quad z \in \mathbb{R}^d, \quad d \in \{128, 256\}. \tag{3}$$

We additionally evaluate $d = 512$ in the ablation studies.

CKKS Setup. We use the CKKS scheme (Yuan et al., 2025), an approximate homomorphic encryption method that supports efficient arithmetic over real-valued data. Encryption and decryption are denoted by $\text{Enc}(\cdot)$ and $\text{Dec}(\cdot)$. Parameters: polynomial degree 16384, modulus $[60, 40, 40, 40, 60]$, and scale 2^{40}, supporting up to three multiplicative levels. For polynomial-head configurations, plaintext training applies clipping to $[-5, 5]$ before squaring for numerical stability:

$$\phi_{\text{train}}(x) = \left(\text{clip}(x, -5, 5)\right)^2. \tag{4}$$

Because comparison-based clipping is not directly evaluated in our CKKS implementation, encrypted inference uses the quadratic polynomial

$$\phi_{\text{HE}}(x) = x^2, \tag{5}$$

which requires one ciphertext multiplication.

3.1 Selective Encryption Strategies

HE-Linear. The backbone and projection are evaluated in plaintext, after which the projected representation is encrypted and the classifier is evaluated under HE:

$$\hat{y} = \text{Dec}(W_{\text{head}}\, \text{Enc}(z)). \tag{6}$$

Because z is produced before encryption, it may be observable to the server and therefore remains relevant to our representation-leakage analysis. **Repr-HE.** Repr-HE protects the selected representation while it crosses an untrusted boundary:

$$\tilde{h} = \text{Enc}(h), \qquad h = \text{Dec}(\tilde{h}), \qquad z = W_{\text{proj}} h, \qquad \hat{y} = W_{\text{head}} z.$$

The untrusted component observes only the encrypted representation, while downstream computation proceeds in plaintext inside the logical trusted component. The corresponding threat and deployment assumptions are detailed in Section 3.2 and Appendix A.5.

Proj-Poly-HE. Projection, polynomial activation, and classification are evaluated under HE:

$$\hat{y} = \text{Dec}(W_{\text{head}}\, \phi(W_{\text{proj}}\, \text{Enc}(h))). \tag{7}$$

This increases multiplicative depth and computational cost, but performs a larger fraction of the downstream computation directly over ciphertexts.

See Appendix D for further details on these design scenarios.

3.2 Threat Model

We assume an *honest-but-curious* adversary that follows the inference protocol but attempts to infer sensitive information from data available outside the trusted boundary. Depending on the strategy, the adversary may observe exposed plaintext representations, CKKS ciphertexts, public cryptographic parameters, representation dimensions, message sizes, and timing metadata. The adversary does not have access to the secret key or to plaintext values protected inside the trusted component.

Deployment interpretation. The trusted boundary denotes a logical trusted component, not a hardware Trusted Execution Environment (TEE) instantiated in our prototype. This component is assumed to hold the secret key and protect plaintext representations and trusted plaintext computation from the untrusted host. A hardware TEE is one possible realization of this component, but evaluating a specific TEE is outside the scope of this work. In our prototype, the boundary is simulated within a

single process: neural-network computation runs on an NVIDIA A800 GPU, while CKKS operations run on CPU. Reported latency therefore excludes network communication, hardware-TEE transitions, and remote-attestation overhead. For Repr-HE, the representation is encrypted before crossing the untrusted boundary, and the untrusted host observes only CKKS ciphertext. Decryption and downstream plaintext computation occur inside the logical trusted component. Compromise of the trusted component, secret-key leakage, and side-channel attacks are outside our threat model.

Representation leakage. We distinguish *intrinsic representation leakage* from *protocol-level confidentiality*. To estimate the information contained in a candidate representation, we train a probing classifier $g(\cdot)$ on plaintext representation–label pairs:

$$\hat{y}_{\text{attack}} = g(r), \qquad r \in \{h, z\}. \tag{8}$$

We report probe performance using ROC-AUC, where 0.5 corresponds to random guessing and values closer to 1.0 indicate greater recoverable information. Probe AUC therefore characterizes how informative a representation would be *if available in plaintext*; it is not a protocol-level attack-success metric when the deployed adversary observes only ciphertext. Accordingly, protocol-level probe AUC is reported as N/A for encrypted-only representations such as Repr-HE. Additional details are provided in Appendix A.5.

3.3 Boundary Optimization Formulation

We now formalize selective HE inference as an optimization problem over the encryption boundary. Let the inference pipeline be decomposed into a sequence of stages:

$$x \xrightarrow{f_1} r_1 \xrightarrow{f_2} r_2 \xrightarrow{\cdots} r_B \xrightarrow{f_{B+1}} \hat{y}, \tag{9}$$

where r_b denotes the intermediate representation at boundary b. For example, in our setting, r_b may correspond to an intermediate token representation, the pooled representation h, the projected representation z, or a representation processed by an encrypted projection and polynomial activation.

For a candidate boundary b, the representation is protected before crossing the corresponding trust boundary, and computation after b may be performed either in plaintext within a trusted environment or under HE, depending on the selected strategy. We define boundary selection as minimizing a joint privacy–utility–cost objective:

$$b^* = \arg\min_{b \in \mathcal{B}} \mathcal{J}(b), \tag{10}$$

where

$$\mathcal{J}(b) = \lambda_{\text{priv}} \mathcal{L}_{\text{priv}}(b) + \lambda_{\text{acc}} \mathcal{L}_{\text{task}}(b) + \lambda_{\text{he}} \mathcal{C}_{\text{he}}(b). \tag{11}$$

Here, $\lambda_{\text{priv}}, \lambda_{\text{acc}}, \lambda_{\text{he}} \geq 0$ control the relative importance of representation privacy, predictive utility, and cryptographic cost. For empirical boundary selection, we normalize each objective term to $[0, 1]$ across the candidate boundaries before applying the weights.

Privacy term. The privacy term measures the intrinsic sensitive information recoverable from the representation associated with boundary b. Since exact mutual information is intractable for high-dimensional neural representations, we use attacker performance on the corresponding plaintext representation as a surrogate:

$$\mathcal{L}_{\text{priv}}(b) = \text{AUC}(g(r_b)). \tag{12}$$

Lower values indicate lower intrinsic representation leakage, with AUC $= 0.5$ corresponding to random guessing. Importantly, this term is used to compare the information content of candidate boundaries; for encrypted representations such as Repr-HE, it should not be interpreted as the actual attack success of the untrusted server.

More generally, the privacy objective is threat-model dependent. Our main instantiation uses label-inference probe AUC; sensitive-attribute inference and input reconstruction capture different leakage notions and may rank candidate boundaries differently. Therefore, the selected boundary should be interpreted relative to the specified adversarial objective rather than as universally optimal for privacy.

Task loss. The utility term measures predictive degradation relative to the plaintext baseline:

$$\mathcal{L}_{\text{task}}(b) = \max\big(0,\ F_{\text{plain}} - F(b)\big), \tag{13}$$

where $F(b)$ denotes the task metric at boundary b (Micro-F1 in our main experiments) and F_{plain} is the corresponding plaintext performance. This term is zero when the selected configuration matches or exceeds the plaintext baseline.

HE cost term. The HE cost term captures the cryptographic burden associated with boundary b:

$$\mathcal{C}_{\text{he}}(b) = \alpha\, N_{\text{op}}(b) + \beta\, D(b) + \gamma\, T_{\text{enc/dec}}(b), \tag{14}$$

where $N_{\text{op}}(b)$ is the number of ciphertext operations, $D(b)$ is the multiplicative depth consumed after boundary b, and $T_{\text{enc/dec}}(b)$ is the encryption/decryption overhead. This formulation aligns with CKKS behavior: ciphertext operations dominate latency, while multiplicative depth governs noise accumulation and numerical degradation.

Constrained form. Equivalently, boundary selection can be written as:

$$\min_{b \in \mathcal{B}} \quad \mathcal{C}_{\text{he}}(b) \tag{15}$$

$$\text{s.t.} \quad \mathcal{L}_{\text{priv}}(b) \le \tau_{\text{priv}}, \tag{16}$$

$$\mathcal{L}_{\text{task}}(b) \le \tau_{\text{task}}, \tag{17}$$

where τ_{priv} specifies an acceptable intrinsic representation-leakage threshold and τ_{task} specifies a maximum tolerable accuracy drop. This constrained view identifies the least expensive candidate boundary satisfying the desired privacy and utility requirements under the selected deployment model.

Additional analysis and computational details are provided in Appendix A.6. Full experimental and dataset details are provided in Appendices A.4 and A.1.

4 RESULTS

We analyze all configurations through the boundary optimization objective (Eq. 11), which captures the tradeoff between privacy, task performance, and homomorphic cost. Across datasets and model scales, each strategy corresponds to a different boundary and execution choice, allowing us to evaluate how these factors evolve in practice (see Table 1). We instantiate Eq. 11 using plaintext probe AUC as a measure of intrinsic representation leakage for $\mathcal{L}_{\text{priv}}$, Micro-F1 for $\mathcal{L}_{\text{task}}$, and latency for $\mathcal{C}_{\text{he}}$.

For compact models (DistilBERT on SST-2, IMDB, and MNLI), HE-Linear closely matches plaintext performance, indicating that head-only encryption preserves accuracy but introduces substantial latency overhead (e.g., $\sim$45$\times$ slowdown). Polynomial heads (HE-PolyMLP) slightly improve performance due to increased capacity, but incur significantly higher latency. These results show that HE can be applied to small models with minimal accuracy degradation, although efficiency remains a limiting factor. On MNLI, similar trends hold: performance remains stable under shallow encryption, while deeper encrypted pipelines exhibit greater performance and computational costs.

For large models (Mistral-7B and Qwen-2.5-3B on GoEmotions), the tradeoff becomes more pronounced. HE-Linear scales poorly due to high-dimensional ciphertext operations. In contrast, Repr-HE provides a favorable privacy–utility–efficiency operating point, preserving near-plaintext performance while maintaining low cryptographic overhead (e.g., 55.81% vs. 58.30% Micro-F1 on Mistral). Under Repr-HE, the untrusted server observes only the CKKS-encrypted representation; therefore, plaintext probe AUC is not interpreted as a protocol-level leakage metric. Deeper encrypted pipelines (Proj-Poly-HE) incur substantially higher latency and greater performance degradation across the evaluated models.

Latency Analysis Latency is primarily driven by ciphertext operations and circuit depth rather than model size. For compact models (DistilBERT), HE-Linear introduces moderate overhead ($\sim$45$\times$, 75 ms vs. 1.67 ms), while HE-PolyMLP further increases latency to $\sim$224 ms due to additional polynomial computation. Despite this, latency remains within a practical range for small models (Table 2). For large models (Mistral-7B), the cost of HE becomes prohibitive. HE-Linear incurs

Table 1: Main results across privacy-preserving strategies on SST-2, IMDB, GoEmotions, and MNLI. Values are averaged over 3 runs and reported as mean ± standard deviation. Latency and throughput are measured separately over the prototype evaluation procedure and therefore need not be exact reciprocals.

Method	Enc. Region	Mic-F1(%)	Mac-F1(%)	EM/Acc(%)	Lat. (ms)	Thr. (samples/s)
DistilBERT (SST-2)						
(Plain)	None	$89.77_{\pm 0.10}$	$89.77_{\pm 0.10}$	$89.77_{\pm 0.10}$	$1.67_{\pm 0.02}$	–
(HE-Linear)	Head	$89.68_{\pm 0.12}$	$89.68_{\pm 0.12}$	$89.68_{\pm 0.12}$	$75.03_{\pm 0.50}$	$13.32_{\pm 0.10}$
(Repr-HE)	Repr.	$89.61_{\pm 0.11}$	$89.60_{\pm 0.11}$	$89.61_{\pm 0.11}$	$1.82_{\pm 0.03}$	$549.45_{\pm 9.00}$
(HE-PolyMLP$_{h=128}$)	Head (Poly.)	$\mathbf{90.48}_{\pm 0.15}$	$\mathbf{90.48}_{\pm 0.15}$	$\mathbf{90.48}_{\pm 0.15}$	$223.95_{\pm 1.20}$	$4.46_{\pm 0.05}$
DistilBERT (IMDB)						
(Plain)	None	$92.15_{\pm 0.08}$	$92.12_{\pm 0.09}$	$92.15_{\pm 0.08}$	$1.72_{\pm 0.02}$	–
(HE-Linear)	Head	$91.98_{\pm 0.10}$	$91.95_{\pm 0.11}$	$91.98_{\pm 0.10}$	$75.40_{\pm 0.60}$	$13.25_{\pm 0.12}$
(Repr-HE)	Repr.	$92.08_{\pm 0.09}$	$92.05_{\pm 0.10}$	$92.08_{\pm 0.09}$	$\mathbf{26.40}_{\pm 0.20}$	$\mathbf{37.84}_{\pm 0.25}$
(HE-PolyMLP$_{h=128}$)	Head (Poly.)	$\mathbf{92.63}_{\pm 0.12}$	$\mathbf{92.60}_{\pm 0.12}$	$\mathbf{92.63}_{\pm 0.12}$	$224.30_{\pm 1.30}$	$4.43_{\pm 0.06}$
DistilBERT (MNLI)						
(Plain)	None	$81.69_{\pm 0.30}$	$81.69_{\pm 0.30}$	$81.69_{\pm 0.30}$	$3.56_{\pm 0.05}$	–
(HE-Linear)	Head	$80.00_{\pm 0.50}$	$80.00_{\pm 0.50}$	$80.00_{\pm 0.50}$	$164.38_{\pm 5.00}$	$6.08_{\pm 0.20}$
(Repr-HE)	Repr.	$80.00_{\pm 0.50}$	$80.00_{\pm 0.50}$	$80.00_{\pm 0.50}$	$\mathbf{17.39}_{\pm 1.50}$	$\mathbf{57.49}_{\pm 2.00}$
(Proj-Poly-HE)	Proj. + Head	$30.20_{\pm 1.00}$	$30.20_{\pm 1.00}$	$30.20_{\pm 1.00}$	$44057.90_{\pm 500.00}$	$0.02_{\pm 0.001}$
Mistral-7B (GoEmotions, $p = 256$)						
(Plain)	None	$\mathbf{58.30}_{\pm 0.40}$	$42.09_{\pm 0.50}$	$\mathbf{47.18}_{\pm 0.45}$	$19.68_{\pm 0.20}$	–
(HE-Linear)	Head	$43.33_{\pm 0.60}$	$20.34_{\pm 0.70}$	$31.80_{\pm 0.65}$	$16{,}353_{\pm 80}$	$0.061_{\pm 0.002}$
(Repr-HE)	Repr.	$55.81_{\pm 0.50}$	$\mathbf{45.06}_{\pm 0.55}$	$39.40_{\pm 0.60}$	$\mathbf{26.60}_{\pm 0.10}$	$\mathbf{37.63}_{\pm 0.40}$
(Proj-Poly-HE)	Proj. + Head	$7.18_{\pm 0.30}$	$3.10_{\pm 0.25}$	$1.00_{\pm 0.10}$	$37{,}177_{\pm 120}$	$0.027_{\pm 0.001}$
Qwen-2.5-3B (GoEmotions, $p = 256$)						
(Plain)	None	$\mathbf{59.68}_{\pm 0.45}$	$47.61_{\pm 0.50}$	$\mathbf{46.41}_{\pm 0.40}$	$17.49_{\pm 0.20}$	–
(HE-Linear)	Head	$55.59_{\pm 0.55}$	$46.51_{\pm 0.60}$	$39.40_{\pm 0.50}$	$1{,}453_{\pm 20}$	$0.68_{\pm 0.02}$
(Repr-HE)	Repr.	$57.27_{\pm 0.50}$	$\mathbf{45.02}_{\pm 0.55}$	$41.40_{\pm 0.60}$	$\mathbf{15.17}_{\pm 0.10}$	$\mathbf{64.18}_{\pm 0.50}$
(Proj-Poly-HE)	Proj. + Head	$0.31_{\pm 0.10}$	$0.38_{\pm 0.10}$	$0.00_{\pm 0.00}$	$35{,}912_{\pm 150}$	$0.03_{\pm 0.001}$

extreme latency (16–17 s per sample), driven by high-dimensional encrypted dot products, while Proj-Poly-HE further increases latency to ∼37 s due to additional homomorphic layers and depth consumption. These results confirm that deep HE pipelines do not scale to large transformer models. In contrast, Repr-HE avoids homomorphic computation entirely, resulting in low end-to-end latency (26.6 ms). This efficiency arises because only encryption, decryption, and lightweight projection are applied, while the backbone computation remains unchanged. Overall, latency scales with the number of encrypted operations and circuit depth, making representation-level encryption significantly more efficient than fully encrypted pipelines.

Privacy–Efficiency Tradeoff. The evaluated strategies occupy distinct privacy–utility–efficiency operating points (Tables 1 and 2). Shallow encrypted computation preserves utility but can leave informative representations exposed, whereas deeper encrypted computation incurs substantially higher cryptographic cost. Repr-HE provides a favorable hybrid operating point by protecting the representation across the untrusted boundary while relying on trusted plaintext execution downstream. Overall, the preferred boundary depends on the joint privacy, utility, cost, and trust requirements.

Threat Model Evaluation The threat model analysis shows that exposing intermediate representations can lead to substantial information leakage regardless of downstream encryption. In both plaintext and HE-Linear settings, the attacker can observe the 4096-dimensional representation, resulting in near-perfect inference (AUC ≈ 0.97–0.98; Table 3). This demonstrates that encrypting only the classifier head does not protect an already exposed representation. We use probe AUC as a proxy for *intrinsic representation leakage*, where higher values indicate more recoverable information from a plaintext representation.

Table 2: Latency breakdown across strategies. Component timings are separately profiled and are not additive. Reported latency excludes network communication and hardware-TEE overheads.

Strategy	Enc. (ms)	HE Comp. (ms)	Dec. (ms)	Lat. (ms)	Thr. (samples/s)
DistilBERT (SST-2)					
(Plain)	–	–	–	1.67	–
(HE-Linear)	≈35	≈40	< 1	75.03	13.32
(HE-PolyMLP)	≈100	≈120	< 1	≈224	4.46
Mistral-7B (GoEmotions)					
(Plain)	–	–	–	19.68	–
(Repr-HE)	11.25	0	4.04	26.60	37.63
(HE-Linear, $p = 256$)	≈8,000	≈8,000	< 1	16,353	0.061
(HE-Linear, $p = 512$)	≈9,000	≈9,000	< 1	17,691	0.057
(Proj-Poly-HE)	≈18,000	≈19,000	< 1	37,177	0.027

In contrast, under Repr-HE the untrusted server observes only the CKKS-encrypted representation. Therefore, plaintext probing is not applicable as a protocol-level attack metric, and we report its probe AUC as N/A. Repr-HE maintains strong task performance (Micro-F1 = 0.558) with low latency (26.60 ms) because it performs encryption and decryption of the representation while downstream computation proceeds in plaintext inside the trusted environment. Proj-Poly-HE similarly prevents plaintext access to the protected representation, but performs substantially more computation under HE, leading to severe performance degradation and much higher latency. Overall, these results highlight the distinction between intrinsic representation leakage and protocol-level confidentiality, and show that protecting the pooled representation provides a practical hybrid operating point under the stated trust model.

Table 3: Threat model and protocol-level attacker visibility across strategies. Probe AUC is reported only when the attacker observes a plaintext representation.

Strategy	Mic-F1	Lat. (ms)	Attacker Access	Probe-AUC	Protection
Mistral-7B (GoEmotions)					
(Plain)	0.583	19.68	Full 4096-dim repr.	0.97	None
(HE-Linear, $p = 256$)	0.433	16,353	Full 4096-dim repr.	0.98	Head enc.; Repr. exposed
(Repr-HE)	**0.558**	**26.60**	CKKS ciphertext only	N/A	Enc. transfer + trusted execution
(Proj-Poly-HE)	0.072	37,177	CKKS ciphertext only	N/A	Enc. downstream computation

Role of HE in Repr-HE. Repr-HE performs no arithmetic while the representation is encrypted. Accordingly, its confidentiality during transmission does not inherently require the homomorphic property of CKKS; conventional authenticated encryption could also protect the representation between trusted endpoints. We include Repr-HE as a boundary and trust-model operating point within the broader selective-encryption design space, rather than as evidence that HE is necessary for representation transport. HE becomes essential when computation must be performed while the representation remains encrypted, as in HE-Linear and Proj-Poly-HE.

Comparison with Full-HE Inference. We additionally compare against THOR (Moon et al., 2025), a full-HE transformer inference baseline, using BERT-base on MRPC with a maximum sequence length of 128 tokens (Table 6 in Appendix G). THOR protects the entire transformer computation and therefore provides stronger end-to-end cryptographic protection, but incurs substantially higher latency. Repr-HE achieves near-plaintext performance at much lower cost by protecting representation transmission and relying on trusted downstream execution, while Proj-Poly-HE performs a larger portion of downstream computation over ciphertexts. Thus, the comparison illustrates different security–efficiency operating points rather than equivalent security guarantees or a direct speedup over full HE. We do not include GAZELLE (Juvekar et al., 2018) and HE-SecureNet (Schneider et al., 2025) as direct quantitative baselines because they target different settings, CNN inference with

Table 4: Ablation study across architectural and encryption design factors.

Model	Factor	Setting	Micro-F1 (%)	Lat. (ms)	Observation
DistilBERT	Head Type	HE-Linear	89.68	75.03	Lower latency
DistilBERT	Head Type	HE-PolyMLP	**90.48**	223.95	Higher F1, higher cost
DistilBERT	Poly. Head Dim.	64	90.37	224.06	Slightly lower F1
DistilBERT	Poly. Head Dim.	128	**90.48**	223.95	Highest F1
DistilBERT	Poly. Head Dim.	256	90.25	224.57	No added benefit
Mistral-7B	HE-Linear Proj. Dim.	256	43.33	16,353	Lower HE cost
Mistral-7B	HE-Linear Proj. Dim.	512	45.39	17,691	Modest F1 gain
Mistral-7B	Boundary / Execution	HE-Linear	43.33	16,353	High HE latency
Mistral-7B	Boundary / Execution	Repr-HE	**55.81**	**26.60**	Favorable hybrid point
Mistral-7B	Boundary / Execution	Proj-Poly-HE	7.18	37,177	High depth and cost

hybrid HE/garbled circuits and privacy-preserving training, respectively whereas THOR provides the closest full-HE transformer inference comparison.

Ablation Study Across Models and Design Factors Table 4 shows that architectural choices affect both utility and HE cost. For DistilBERT, the polynomial MLP improves Micro-F1 but increases latency, with $h = 128$ performing best among the tested widths. For Mistral-7B, increasing the HE-Linear projection dimension from 256 to 512 yields a modest accuracy gain at higher cost. Across strategies, boundary placement has the largest effect: Repr-HE offers a favorable hybrid utility–latency tradeoff, whereas Proj-Poly-HE incurs substantially higher cost and utility degradation.

Ablation on Additional Boundary Choices. To examine whether the observed tradeoff depends on the three primary configurations, we additionally evaluate intermediate token-level, final-token, pooled, and projected representation boundaries on Mistral-7B/GoEmotions (Appendix G.1). Later token-level boundaries improve task utility, but protecting the full 128×4096 token representation incurs approximately 745 ms latency. Pooling reduces this cost to approximately 27 ms with only a small utility change, while projection further decreases latency and intrinsic probe AUC at the cost of increasing task degradation. Overall, the results indicate that no single boundary is universally optimal; final pooled and moderately projected representations form a practical Pareto-efficient region among the evaluated candidates.

Additional privacy analyses, including layer-wise attack recovery, matched plaintext ablation, authenticated-encryption baselines and sensitive-attribute inference, are provided in Appendix G.

Limitations. Repr-HE relies on an assumed logical trusted component and performs no computation over ciphertext. Our AES-GCM baseline shows that conventional authenticated encryption is more efficient for representation transfer under this trust model. Privacy estimates also depend on the chosen attack: task-label inference, sensitive-attribute inference, and input reconstruction capture different forms of leakage and may favor different boundaries. Matched plaintext controls show that the severe Proj-Poly-HE accuracy loss arises along the encrypted execution path, but we cannot isolate its cause without stage-level diagnostics. Finally, our single-process latency measurements exclude network communication and trusted-environment overhead and depend on the implementation and hardware.

5 Conclusion

We present a selective encryption framework for transformer inference that formulates protection placement as a joint privacy–utility–cost problem. Our experiments show that head-only, representation-level, and deeper encrypted computation occupy substantially different operating points. Under the hybrid trust model considered here, representation-level encrypted transmission preserves near-plaintext utility with low cryptographic overhead, while deeper HE computation incurs substantially greater cost. Fine-grained boundary and sensitivity analyses further identify pooled and moderately projected representations as practical Pareto-efficient choices rather than a universally optimal boundary.

AI USE STATEMENT

Generative AI tools were used only for language polishing, including grammar, clarity, and stylistic refinement. They were not used to generate experimental results, analyses, or scientific conclusions. The authors reviewed and take responsibility for all final content.

ETHICS STATEMENT

This work studies privacy-preserving inference and does not involve human-subject experiments or the collection of new personal data. We evaluate publicly available datasets and focus on reducing exposure of intermediate representations during model inference. As with other privacy-preserving systems, the guarantees depend on the stated threat model and trusted execution assumptions.

REPRODUCIBILITY STATEMENT

We provide implementation details, training settings, CKKS parameters, threat-model assumptions, and pseudocode in Sections 3–4 and Appendices A and C.

REFERENCES

Fabian Boemer, Anamaria Costache, Rosario Cammarota, and Casimir Wierzynski. ngraph-he2: A high-throughput framework for neural network inference on encrypted data. In *Proceedings of the 7th ACM workshop on encrypted computing & applied homomorphic cryptography*, pp. 45–56, 2019.

Roshan Dathathri, Olli Saarikivi, Hao Chen, Kim Laine, Kristin Lauter, Saeed Maleki, Madanlal Musuvathi, and Todd Mytkowicz. Chet: an optimizing compiler for fully-homomorphic neural-network inferencing. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pp. 142–156, 2019.

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 *Forty-second International Conference on Machine Learning*, 2025.

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*, pp. 4040–4054, 2020.

Hao Fei, Tat-Seng Chua, Chenliang Li, Donghong Ji, Meishan Zhang, and Yafeng Ren. On the robustness of aspect-based sentiment analysis: Rethinking model, data, and training. *ACM Transactions on Information Systems*, 41(2):1–32, 2022.

Tiantian Feng, Rajat Hebbar, and Shrikanth Narayanan. Trust-ser: On the trustworthiness of fine-tuning pre-trained speech embeddings for speech emotion recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11201–11205. IEEE, 2024.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Yu Fu, Yu Tong, Yijing Ning, Tianshi Xu, Meng Li, Jingqiang Lin, and Dengguo Feng. Swift: Fast secure neural network inference with fully homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 2025.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pp. 201–210. PMLR, 2016.

Futoshi Hashimoto, Takuya Suzuki, Shunta Sakai, and Hayato Yamana. Accurate and low-latency secure bert inference leveraging homomorphic encryption and trusted execution environment. In *2025 IEEE International Conference on Big Data (BigData)*, pp. 7340–7346. IEEE, 2025.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, 2019.

Dou Hu, Yinan Bao, Lingwei Wei, Wei Zhou, and Songlin Hu. Supervised adversarial contrastive learning for emotion recognition in conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10835–10852, 2023.

Mimansa Jaiswal and Emily Mower Provost. Privacy enhanced multimodal neural representations for emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7985–7993, 2020.

Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th USENIX security symposium (USENIX security 18)*, pp. 1651–1669, 2018.

Garam Lee, Minsoo Kim, Jai Hyun Park, Seung-won Hwang, and Jung Hee Cheon. Privacy-preserving text classification on bert embeddings with homomorphic encryption. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3169–3175, 2022.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Kiwan Maeng, Chuan Guo, Sanjay Kariyappa, and Edward Suh. Measuring and controlling split layer privacy leakage using fisher information. *arXiv preprint arXiv:2209.10119*, 2022.

Kiwan Maeng, Chuan Guo, Sanjay Kariyappa, and G Edward Suh. Bounding the invertibility of privacy-preserving instance encoding using fisher information. *Advances in Neural Information Processing Systems*, 36:51904–51925, 2023.

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*, pp. 3765–3779, 2025.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, volume 2018, pp. 1–15, 2018.

Yiteng Peng, Dongwei Xiao, Zhibo Liu, Zhenlan Ji, Daoyuan Wu, Shuai Wang, and Juergen Rahmel. The phantom menace in crypto-based pet-hardened deep learning models: Invisible configuration-induced attacks. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4379–4393, 2025.

George-Liviu Pereteanu, Amir Alansary, and Jonathan Passerat-Palmbach. Split he: Fast secure inference combining split learning and homomorphic encryption. *arXiv preprint arXiv:2202.13351*, 2022.

Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4):1–34, 2023.

Thomas Schneider, Huan-Chih Wang, and Hossein Yalame. He-securenet: An efficient and usable framework for model training via homomorphic encryption. In *Proceedings of the 24th Workshop on Privacy in the Electronic Society*, pp. 104–115, 2025.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–206, 2019.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Jiangjun Yuan, Weinan Liu, Jiawen Shi, and Qingqing Li. Approximate homomorphic encryption based privacy-preserving machine learning: a survey. *Artificial Intelligence Review*, 58(3):82, 2025.

Yuke Zhang, Dake Chen, Souvik Kundu, Haomei Liu, Ruiheng Peng, and Peter A Beerel. C 2 pi: an efficient crypto-clear two-party neural network private inference. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6. IEEE, 2023.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A Technical appendices and supplementary material

A.1 Datasets

We evaluate our framework on four standard text classification datasets.

SST-2. The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) is a binary sentiment classification dataset consisting of short movie review snippets labeled as positive or negative. It contains approximately 67k training samples and 1.8k validation samples.

IMDB. The IMDB dataset (Maas et al., 2011) is a binary sentiment classification benchmark composed of long-form movie reviews. It includes 25k training and 25k test samples. Compared to SST-2, the longer sequences introduce greater contextual complexity, making it a more challenging setting for evaluating model robustness and efficiency.

MNLI. The Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) is a three-class classification benchmark for natural language inference, where each example is labeled as entailment, contradiction, or neutral. It contains approximately 393k training samples and 20k validation samples (matched and mismatched splits).

GoEmotions. GoEmotions (Demszky et al., 2020) is a multi-label emotion classification dataset containing 58,009 English Reddit comments annotated across 28 fine-grained emotion categories. Each sample may express multiple simultaneous emotions. We use a standard train/validation/test split (43k/5k/10k). Its large label space and label co-occurrence make it a challenging benchmark for evaluating privacy–utility tradeoffs in inference.

A.2 Related Works

In parallel, prior work on representation analysis (Zhu et al., 2019; Fredrikson et al., 2015; Nasr et al., 2018) has shown that intermediate activations in neural networks can leak sensitive information, even when model parameters are protected. Probing and inference attacks demonstrate that representations encode both task-relevant and private attributes, making them a key target for secure inference (Hewitt & Manning, 2019; Song & Shmatikov, 2019). Building on these insights, we reinterpret HE-based inference as a problem of *encryption boundary selection*. Unlike prior approaches that apply encryption uniformly or heuristically, we provide a principled formulation for selecting the boundary that optimizes the privacy–utility–efficiency tradeoff. Our results show that representation-level protection provides a favorable hybrid operating point relative to head-only and deeper encrypted computation under the evaluated deployment settings.

A.3 Summary and Design Guidelines

Across all experiments, the observed trends align with the proposed boundary-optimization formulation. HE-Linear leaves informative intermediate representations exposed despite substantial computational cost. Proj-Poly-HE protects a larger portion of downstream computation but incurs high homomorphic cost and utility degradation due to ciphertext operations and depth constraints. In contrast, Repr-HE provides a favorable hybrid operating point by encrypting the representation across the untrusted boundary while relying on trusted plaintext execution downstream. Under this deployment, the server observes only ciphertext; therefore, plaintext probe AUC is not interpreted as a protocol-level privacy metric. Instead, probing is used to characterize the intrinsic leakage of candidate representations. Overall, our results identify pooled and moderately projected representations as practical Pareto-efficient boundaries under balanced privacy, utility, and cost requirements.

Design Guidelines.

- **Select the appropriate protection boundary.** Protecting informative intermediate representations can provide a favorable tradeoff compared with head-only or substantially deeper encrypted computation.
- **Limit multiplicative depth.** Polynomial activations and stacked HE layers consume the CKKS depth budget, increasing latency and numerical approximation error.
- **Control representation dimensionality.** Projection can reduce cryptographic cost and intrinsic recoverability, although aggressive compression may reduce task utility.
- **Match encrypted computation to the trust model.** When trusted execution is available, encrypted representation transmission can avoid costly downstream homomorphic arithmetic; stronger end-to-end protection requires correspondingly greater encrypted computation.

A.4 HE-Compatible Training

To maintain consistency between plaintext training and encrypted inference, we design models that operate with HE-compatible components. Algorithms 1–3 summarize the overall selective HE inference pipeline, boundary optimization procedure, and privacy leakage evaluation used throughout this work.

a) Polynomial Activation: Standard activations (e.g., ReLU) are not directly compatible with CKKS. For polynomial-head configurations, plaintext training applies clipping to $[-5, 5]$ before squaring for numerical stability:

$$\phi_{\text{train}}(x) = \left(\text{clip}(x, -5, 5)\right)^2 .$$

Because comparison-based clipping is not directly evaluated in our CKKS implementation, encrypted inference uses the quadratic polynomial

$$\phi_{\text{HE}}(x) = x^2,$$

which requires one ciphertext multiplication.

b) DistilBERT: For compact models, we extract the pooled representation from the `[CLS]` token and train lightweight classifier heads. We consider a linear head and a two-layer polynomial MLP. Training is performed using standard cross-entropy loss for classification tasks.

c) Large Models (Mistral, Qwen): For large models, we use last-token pooling to obtain a representation vector. To reduce HE cost, we project this representation using a linear layer ($z = W_{\text{proj}} h$) to a lower dimension ($d \in \{128, 256\}$) before downstream processing. We additionally evaluate $d = 512$ in the ablation studies. Models are fine-tuned using LoRA, which introduces low-rank updates while keeping the backbone frozen and training only the projection and classifier layers.

Training and System Setup. DistilBERT models are trained for 2 epochs with batch size 16 and learning rate 2×10^{-5}, while Mistral-7B and Qwen-2.5-3B are trained for 3 epochs using LoRA with batch size 8 and learning rate 1×10^{-4}. Gradient clipping with norm 1.0 is applied. All reported results are averaged over three runs. Gradient clipping with norm 1.0 is applied. Neural network training and inference are performed on an NVIDIA A800 80GB GPU, while CKKS operations are implemented on CPU using TenSEAL. We report Micro-F1, Macro-F1, Exact Match (EM), accuracy, probe AUC, latency, and throughput.

Algorithm 1 Selective HE Inference

Require: Input text x, backbone f_θ, projection matrix W_{proj}, classifier head W_{head}, strategy $\in \{\texttt{HE-Linear}, \texttt{Repr-HE}, \texttt{Proj-Poly-HE}\}$
Require: CKKS encryption/decryption $\text{Enc}(\cdot), \text{Dec}(\cdot)$
Require: Polynomial activation $\phi(\cdot)$
Ensure: Prediction $\hat{y}$

$h \leftarrow \text{ExtractRepresentation}(f_\theta(x))$ ▷ CLS for encoder models, LastToken for decoder-only models
if strategy == `HE-Linear` **then**
  $z \leftarrow W_{\text{proj}} h$
  $z_{\text{enc}} \leftarrow \text{Enc}(z)$
  $\hat{y} \leftarrow \text{Dec}(W_{\text{head}} z_{\text{enc}})$
else if strategy == `Repr-HE` **then**
  $h_{\text{enc}} \leftarrow \text{Enc}(h)$
  $h \leftarrow \text{Dec}(h_{\text{enc}})$
  $z \leftarrow W_{\text{proj}} h$
  $\hat{y} \leftarrow W_{\text{head}} z$
else if strategy == `Proj-Poly-HE` **then**
  $h_{\text{enc}} \leftarrow \text{Enc}(h)$
  $z_{\text{enc}} \leftarrow W_{\text{proj}} h_{\text{enc}}$
  $z_{\text{poly}} \leftarrow \phi(z_{\text{enc}})$
  $\hat{y} \leftarrow \text{Dec}(W_{\text{head}} z_{\text{poly}})$
end if
return $\hat{y}$

Algorithm 2 Boundary Optimization

Require: Candidate boundaries $\mathcal{B}$
Require: Weights $\lambda_{\text{priv}}, \lambda_{\text{acc}}, \lambda_{\text{he}}$
Ensure: Selected boundary b^*

for each boundary $b \in \mathcal{B}$ **do**
  $\mathcal{L}_{\text{priv}}(b) \leftarrow \text{AUC}(g(r_b))$
  $\mathcal{L}_{\text{task}}(b) \leftarrow \max(0, F_{\text{plain}} - F(b))$
  $\mathcal{C}_{\text{he}}(b) \leftarrow \alpha N_{\text{op}}(b) + \beta D(b) + \gamma T_{\text{enc/dec}}(b)$
end for
Normalize $\mathcal{L}_{\text{priv}}$, $\mathcal{L}_{\text{task}}$, and $\mathcal{C}_{\text{he}}$ to $[0, 1]$ across $\mathcal{B}$
for each boundary $b \in \mathcal{B}$ **do**
  $\mathcal{J}(b) \leftarrow \lambda_{\text{priv}} \widetilde{\mathcal{L}}_{\text{priv}}(b) + \lambda_{\text{acc}} \widetilde{\mathcal{L}}_{\text{task}}(b) + \lambda_{\text{he}} \widetilde{\mathcal{C}}_{\text{he}}(b)$
end for
$b^* \leftarrow \arg\min_{b \in \mathcal{B}} \mathcal{J}(b)$
return b^*

A.5 Threat Model Details

We formalize the threat model used to evaluate privacy leakage in selective HE inference.

The trusted boundary is simulated logically within the prototype; no hardware TEE or physical network communication is instantiated, and their associated overheads are therefore excluded from the reported latency.

Attacker Assumption. We consider an *honest-but-curious* server that follows the prescribed inference protocol but attempts to infer sensitive information from data available outside the trusted boundary. The attacker has full knowledge of the model architecture and parameters, consistent with standard white-box assumptions in representation-leakage analysis.

Attacker Access. The attacker observes any plaintext intermediate representation exposed by the selected strategy. In plaintext and HE-Linear settings, this includes the pooled representation $h \in \mathbb{R}^D$ (e.g., $D = 4096$ for Mistral-7B). In Repr-HE and Proj-Poly-HE, the protected representation is

Algorithm 3 Privacy Leakage Evaluation

Require: Intermediate representation r
Require: Probe attacker $g(\cdot)$
Ensure: Leakage score
1: Train probing classifier $g(r)$
2: Predict sensitive labels using g
3: Compute ROC-AUC score
4: **return** AUC

available to the untrusted server only as CKKS ciphertext. The attacker may observe ciphertexts, representation dimensions, public cryptographic parameters, message sizes, and timing metadata, but does not have access to the plaintext input, secret key, or plaintext values inside the trusted environment. Compromise of the trusted environment, key leakage, and side-channel attacks are outside our current threat model.

Attack Model. To quantify the intrinsic information contained in a candidate representation, we train a probing classifier $g(\cdot)$ on plaintext representation–label pairs (r_b, y), where $r_b \in \{h, z, \ldots\}$ denotes the representation at boundary b and y is the ground-truth label. The probe is implemented as a lightweight linear classifier. This analysis characterizes representation leakage if r_b were available in plaintext and is used to compare candidate boundaries.

Evaluation Metric. We use the Area Under the ROC Curve (AUC) to measure probe performance. An AUC of 0.5 corresponds to random guessing, while values approaching 1.0 indicate greater recoverable label information. Compared with accuracy, AUC is less sensitive to class imbalance and provides a consistent measure of representation informativeness across datasets.

Interpretation. Probe AUC measures *intrinsic representation leakage*, not protocol-level attacker success. Lower AUC indicates that less information is recoverable when the corresponding representation is available in plaintext. For Repr-HE, however, the deployed untrusted server observes only CKKS ciphertext; therefore, plaintext probe AUC is not applicable as a protocol-level privacy metric. Its confidentiality instead follows from encrypted transmission and trusted decryption. Accordingly, we use probing primarily as a diagnostic for comparing candidate boundary locations.

Rather than competing directly with end-to-end fully homomorphic approaches, Repr-HE illustrates a different security–efficiency operating point. When a trusted execution environment is available, protecting the representation while it crosses an untrusted boundary can substantially reduce cryptographic overhead. Stronger protection with fewer trust assumptions requires correspondingly more computation to be performed under encryption.

Positioning of Repr-HE. Repr-HE is not an end-to-end homomorphic inference pipeline. It protects the intermediate representation while crossing the untrusted boundary, after which decryption and downstream plaintext computation occur inside the trusted environment. Thus, Repr-HE represents a hybrid encrypted-transmission and trusted-execution configuration. Within our boundary-optimization framework, it occupies a different security–efficiency operating point from both head-only encryption and deeper homomorphic computation. Our experiments identify pooled and moderately projected representations as practical Pareto-efficient choices under balanced deployment requirements.

A.6 Extended Analysis of Boundary Optimization

This section provides additional interpretation of the boundary-optimization formulation introduced in Section 3.3.

Representation space. We consider candidate boundaries $\{r_b\}_{b \in \mathcal{B}}$, where r_b may correspond to an intermediate token representation, pooled representation h, or projected representation z. Each boundary induces a different combination of intrinsic representation leakage, task utility, and cryptographic cost.

Tradeoff structure. For empirical boundary selection, the privacy, task, and HE-cost terms are normalized to $[0, 1]$ across candidate boundaries before evaluating

$$\mathcal{J}(b) = \lambda_{\text{priv}}\mathcal{L}_{\text{priv}}(b) + \lambda_{\text{acc}}\mathcal{L}_{\text{task}}(b) + \lambda_{\text{he}}\mathcal{C}_{\text{he}}(b).$$

The preferred boundary therefore depends on both the measured properties of each representation and the deployment-specific weights. Moving the boundary changes not only representation informativeness but also the amount and dimensionality of encrypted computation, producing the observed privacy–utility–efficiency tradeoff.

Constrained interpretation. The constrained formulation identifies the least-cost boundary satisfying specified privacy and utility requirements:

$$b^* = \arg\min_{b \in \mathcal{B}} \mathcal{C}_{\text{he}}(b) \quad \text{s.t.} \quad \mathcal{L}_{\text{priv}}(b) \leq \tau_{\text{priv}}, \qquad \mathcal{L}_{\text{task}}(b) \leq \tau_{\text{task}}.$$

Thus, no particular boundary is universally optimal; the selected point depends on the desired privacy, utility, cost, and trust requirements.

Privacy interpretation. We use plaintext probe performance as an empirical measure of *intrinsic representation leakage*. Lower probe AUC indicates less recoverable information when the corresponding representation is available in plaintext. This quantity is used to compare candidate boundaries and should not be interpreted as protocol-level attacker success when the deployed adversary observes only ciphertext.

Connection to empirical results. Our experiments illustrate this multi-objective structure. HE-Linear leaves informative upstream representations exposed while encrypting only the downstream classifier. Proj-Poly-HE performs more downstream computation over ciphertexts but incurs substantially greater latency and utility degradation. Repr-HE instead protects the pooled representation while crossing the untrusted boundary and relies on trusted plaintext execution downstream. Across the broader boundary ablation and weight sensitivity analysis, pooled and moderately projected representations form a practical Pareto-efficient region rather than a universally optimal boundary.

Computational Complexity. Under CKKS, computational cost depends primarily on ciphertext operations, representation dimensionality and packing, and multiplicative depth. Repr-HE performs no downstream homomorphic arithmetic and incurs mainly encryption/decryption overhead, whereas HE-Linear adds an encrypted classifier operation. Proj-Poly-HE evaluates the projection, polynomial activation, and classifier over ciphertexts, requiring additional multiplications, rescaling, and rotations. Consequently, cryptographic cost increases with the amount and depth of computation performed after the encryption boundary.

B Leakage

B.1 Leakage Measurement

We quantify intrinsic representation leakage by training a linear probe on plaintext candidate representations. When a representation is exposed by the deployed strategy, this also reflects attacker-accessible leakage; for encrypted-only strategies, probing is used only as a boundary diagnostic. AUC values closer to 0.5 indicate lower intrinsic recoverability.

C Broader Impact

This work advances privacy-preserving inference for transformer models by identifying efficient encryption boundaries using homomorphic encryption (HE). It enables practical deployment of machine learning systems in sensitive domains such as healthcare, finance, and user-facing NLP applications by protecting informative representations without sacrificing performance. By demonstrating that representation-level encryption provides a strong tradeoff between utility, latency, and privacy, the approach lowers the barrier to adopting secure inference at scale. More broadly, this work contributes to the development of trustworthy AI systems by integrating privacy guarantees into modern deep learning pipelines while maintaining efficiency.

D Design Scenarios for Selective Encryption

Head-Only Encryption (HE-Linear). In HE-Linear, the backbone and projection are evaluated in plaintext, and the resulting projected representation is encrypted before the classifier is evaluated under HE. This requires relatively little encrypted computation, but the intermediate representation is produced before encryption and may therefore remain visible to the untrusted server. Consequently, head-only encryption provides limited protection when informative upstream representations are exposed.

Representation-Level Encryption (Repr-HE). Repr-HE places the protection boundary at the pooled representation. The backbone executes in plaintext inside a trusted environment, and the representation is encrypted before crossing into an untrusted component. The untrusted host or communication channel observes only CKKS ciphertext. Decryption and downstream prediction are performed in plaintext inside the trusted environment. Repr-HE therefore represents a hybrid encrypted-transmission and trusted-execution design rather than end-to-end homomorphic inference. This configuration provides a favorable utility–efficiency operating point under the stated trust model.

Projection + Polynomial Head (Proj-Poly-HE). Proj-Poly-HE performs a larger portion of downstream computation under homomorphic encryption. After the plaintext backbone produces the representation, it is encrypted, and the projection, degree-2 polynomial activation, and classifier are evaluated over ciphertexts. This protects the representation across the untrusted boundary and reduces reliance on plaintext downstream computation, but requires additional ciphertext operations and multiplicative depth. Consequently, it incurs substantially higher latency and greater numerical and utility degradation than the shallower configurations.

E Notation.

Let x denote the input text and $f_\theta(\cdot)$ the pretrained transformer backbone parameterized by θ. The model prediction is denoted by $\hat{y} = f_\theta(x)$. We extract an intermediate representation $h \in \mathbb{R}^{d_h}$ from the backbone, where $h = \mathrm{CLS}(f_\theta(x))$ for encoder models and $h = \mathrm{LastToken}(f_\theta(x))$ for decoder-only models. The dimensionality of h is denoted by d_h (e.g., 768 for DistilBERT and 4096 for Mistral/Qwen). To reduce homomorphic encryption (HE) cost, we project h using a learned matrix W_{proj}:

$$z = W_{\mathrm{proj}} h, \qquad z \in \mathbb{R}^d,$$

where $d \in \{128, 256\}$ is the projected dimension. The classifier head is represented by W_{head}. We additionally consider $d = 512$ in the ablation analysis.

We use the CKKS homomorphic encryption scheme, where $\mathrm{Enc}(\cdot)$ and $\mathrm{Dec}(\cdot)$ denote encryption and decryption, respectively. For polynomial-head configurations, plaintext training uses the stabilized activation

$$\phi_{\mathrm{train}}(x) = \left(\mathrm{clip}(x, -5, 5)\right)^2,$$

whereas encrypted inference evaluates the CKKS-compatible quadratic polynomial

$$\phi_{\mathrm{HE}}(x) = x^2.$$

In our threat model, $g(\cdot)$ denotes a probing attacker trained on intermediate representations to infer sensitive information. The representation exposed at a candidate encryption boundary is denoted by r_b, where $b \in \mathcal{B}$ indexes possible boundary locations in the inference pipeline. The privacy loss at boundary b is measured using $\mathcal{L}_{\mathrm{priv}}(b)$, the task degradation using $\mathcal{L}_{\mathrm{task}}(b)$, and the homomorphic computation cost using $\mathcal{C}_{\mathrm{he}}(b)$. The overall boundary optimization objective is denoted by $\mathcal{J}(b)$, while λ_{priv}, λ_{acc}, and λ_{he} control the relative weighting of privacy, utility, and computational cost.

F CKKS Noise and Multiplicative Depth Analysis

For the large-model CKKS configuration, the polynomial modulus degree is set to 16384 with modulus chain $[60, 40, 40, 40, 60]$ and scale 2^{40}, providing sufficient depth for the encrypted operations

evaluated in our prototype. The computational behavior of the strategies differs substantially with respect to ciphertext arithmetic and multiplicative depth.

Repr-HE performs no downstream homomorphic arithmetic. It encrypts the selected representation before the untrusted boundary and decrypts it inside the assumed logical trusted component, after which downstream computation proceeds in plaintext. Consequently, Repr-HE does not consume multiplicative depth; its cryptographic overhead is primarily associated with encryption and decryption. Physical network communication and hardware-TEE overheads are not included in our prototype measurements.

HE-Linear performs only the encrypted classifier operation and therefore requires relatively shallow encrypted computation. In contrast, Proj-Poly-HE performs encrypted projection, quadratic activation, and classification stages. The quadratic activation $\phi_{\mathrm{HE}}(x) = x^2$ requires one ciphertext multiplication. These stages introduce additional ciphertext–plaintext multiplications, rescaling, additions, and rotations, substantially increasing cryptographic cost.

Our prototype uses intermediate decrypt–re-encrypt steps for vector repacking between encrypted stages. These operations are modeled as occurring inside the logical trusted component; therefore, Proj-Poly-HE should be interpreted as staged encrypted computation rather than a continuously end-to-end homomorphic circuit. Accordingly, the reported depth analysis characterizes the encrypted arithmetic within each stage rather than a single uninterrupted ciphertext-depth path.

Approximate CKKS arithmetic and repeated encrypted operations may contribute numerical error, although our experiments do not isolate this effect from architecture- and training-related degradation. In addition to multiplicative depth, representation dimensionality, packing efficiency, and the number of ciphertext operations contribute substantially to runtime. These observations motivate our boundary-selection formulation, which jointly considers the location of the protected representation and the amount of encrypted downstream computation in the privacy–utility–efficiency tradeoff.

G Additional Results

G.1 Ablation on Additional Boundary Choices

We extend the boundary analysis on Mistral-7B/GoEmotions by varying the protected representation while keeping the remaining execution protocol fixed. We consider intermediate transformer layers, the final token-level representation, alternative pooling boundaries, and projected representations of different dimensions. Probe AUC is computed from the corresponding *plaintext* representation and therefore measures intrinsic representation leakage rather than protocol-level attacker success. For Repr-HE, the deployed adversary observes only ciphertext, as described in the threat model.

Table 5: Ablation over additional boundary choices for Mistral-7B on GoEmotions. Probe AUC measures intrinsic information recoverable from the corresponding plaintext representation and is used only as a boundary diagnostic.

Boundary	**Repr. Shape**	**Mic-F1 (%)**	**Lat. (ms)**	**Plain. Probe AUC**
Plain	–	58.30 ± 0.40	19.68 ± 0.20	0.97
HE-Linear	Projected head input	43.33 ± 0.60	$16{,}353 \pm 80$	0.98
Layer 8 token repr.	128×4096	49.12 ± 0.58	742.60 ± 8.30	0.74
Layer 16 token repr.	128×4096	52.84 ± 0.54	743.10 ± 8.20	0.83
Layer 24 token repr.	128×4096	55.73 ± 0.49	744.00 ± 8.40	0.91
Final-layer token repr.	128×4096	$\mathbf{57.64 \pm 0.46}$	744.80 ± 8.50	0.98
Final mean-pooled repr.	4096	56.42 ± 0.48	27.10 ± 0.15	0.95
Repr-HE: final last-token pooled	4096	55.81 ± 0.50	26.60 ± 0.10	0.97
Projected repr., $d = 512$	512	55.37 ± 0.51	23.84 ± 0.13	0.94
Projected repr., $d = 256$	256	54.91 ± 0.53	22.16 ± 0.12	0.90
Projected repr., $d = 128$	128	52.88 ± 0.56	$\mathbf{21.34 \pm 0.11}$	0.84
Proj-Poly-HE	Proj. + poly. head	7.18 ± 0.30	$37{,}177 \pm 120$	–

The ablation reveals three trends. First, moving the token-level boundary deeper improves task utility, with the final-layer representation reaching 57.64% Micro-F1, but the full 128×4096 representation remains expensive to protect, requiring approximately 745 ms across token-level boundaries. Second, pooling reduces latency by more than an order of magnitude while retaining strong task performance. Third, reducing the projection dimension from 512 to 128 further lowers latency and plaintext probe AUC, but increasingly reduces task utility. These results show that boundary selection is inherently multi-objective: final pooled representations favor utility, whereas moderate projections provide lower-cost alternatives with reduced intrinsic recoverability.

Table 6: Comparison with THOR on BERT-base/MRPC (sequence length 128). Methods provide different protection scopes; latencies should therefore be interpreted together with the corresponding trust model.

Method	**Protected Scope**	**Perf. (%)**	**Latency**
Plain	None	88.7	8.4 ms
Repr-HE	Encrypted repr. transfer	87.9	34.8 ms
HE-Linear	Encrypted classifier	87.6	182.4 ms
Proj-Poly-HE	Projection + poly. head	84.1	48.6 s
THOR	Full BERT-base under HE	88.1	10.43 min

Ablation on Objective Weights. We examine the sensitivity of boundary selection to λ_{priv}, λ_{acc}, and λ_{he} using Mistral-7B on GoEmotions. After normalizing the three objective terms to $[0, 1]$, we sweep the weight simplex in increments of 0.1 subject to $\lambda_{\text{priv}} + \lambda_{\text{acc}} + \lambda_{\text{he}} = 1$. The selected boundary changes with deployment priorities: privacy-focused settings favor earlier intermediate boundaries, utility-focused settings favor the final token representation, and HE-cost-focused settings favor compressed projections. The final pooled representation is selected in 58% of configurations, while pooled or moderately projected representations ($d \in \{256, 512\}$) are selected in 87%. This confirms that no single boundary is universally optimal, while pooled and moderately projected representations remain robust Pareto-efficient choices across a broad range of weights.

Table 7: Sensitivity of boundary selection to the objective weights on Mistral-7B/GoEmotions. The objective gap denotes the difference between the best and second-best normalized objective values.

$(\lambda_{\text{priv}}, \lambda_{\text{acc}}, \lambda_{\text{he}})$	**Preference**	**Selected Boundary**	**Second Best**	**Gap**
$(0.60, 0.20, 0.20)$	Privacy	Layer 16 token repr.	Layer 8 token repr.	0.013
$(0.45, 0.35, 0.20)$	Privacy–utility	Final pooled repr.	Projected repr. ($d = 512$)	0.009
$(0.33, 0.33, 0.34)$	Balanced	Final pooled repr.	Projected repr. ($d = 512$)	0.008
$(0.20, 0.60, 0.20)$	Utility	Final-layer token repr.	Final pooled repr.	0.016
$(0.20, 0.20, 0.60)$	HE cost	Projected repr. ($d = 256$)	Projected repr. ($d = 128$)	0.006

G.2 Sensitive-Attribute Leakage

To distinguish task-label leakage from leakage of an independent sensitive attribute, we conduct an additional attribute-inference experiment using Mistral-7B on Bias in Bios. Occupation is the downstream prediction task, while gender is treated as the sensitive attribute. A separate probe is trained to predict gender from the attacker-visible representation.

Table 8: Sensitive-attribute inference on Bias in Bios. Gender AUC is reported only when the attacker observes a plaintext representation.

Method	**Occupation Mic-F1 (%)**	**Gender AUC**	**Attacker Observation**
Plain	61.82 ± 0.46	0.84 ± 0.01	Plaintext repr.
HE-Linear	59.74 ± 0.53	0.85 ± 0.01	Plaintext repr.
Repr-HE	60.91 ± 0.49	N/A	CKKS ciphertext only
Proj-Poly-HE	12.63 ± 0.38	N/A	CKKS ciphertext only

The results show that plaintext representations expose information about a sensitive attribute distinct from the downstream task label, with gender AUCs of 0.84–0.85. Under Repr-HE and Proj-Poly-HE, the untrusted attacker observes only CKKS ciphertext, so plaintext attribute inference is not applicable under our threat model. This experiment confirms that our probing framework can evaluate attribute-specific leakage in addition to task-label leakage.

G.3 Layer-Wise Attack-Recovery Ablation

We additionally evaluate intrinsic input recoverability across selected DistilBERT/SST-2 boundaries. For each boundary, an inversion attacker is trained to reconstruct the original WordPiece tokens from the plaintext representation. Attack recovery is measured as the percentage of non-padding tokens correctly reconstructed.

Table 9: Attack-recovery ablation across representative DistilBERT boundaries on SST-2. Lower recovery indicates lower intrinsic representation leakage.

Boundary	**Recovery (%)**	**Mic-F1 (%)**
Embedding output	96.80 ± 0.40	89.73 ± 0.11
Layer 3	77.50 ± 1.00	89.68 ± 0.12
Layer 6	42.80 ± 1.10	89.62 ± 0.11
Pooled repr.	$\mathbf{21.60 \pm 0.90}$	$\mathbf{89.61 \pm 0.11}$
Projected ($d = 256$)	14.20 ± 0.70	89.31 ± 0.14
Projected ($d = 128$)	9.10 ± 0.60	88.94 ± 0.17

Recovery decreases substantially as the boundary moves deeper and becomes more compressed, while task performance remains close to plaintext up to the pooled representation. Further projection reduces recoverability but introduces greater utility loss. This supports the pooled representation as a favorable privacy–utility operating point among the evaluated boundaries. The analysis measures intrinsic plaintext-representation leakage; under deployed Repr-HE, the untrusted server observes only ciphertext.

G.4 Matched plaintext ablation.

We evaluated the saved Proj-Poly-HE checkpoints in plaintext using the same projection, classifier head, and unclipped square activation as in encrypted inference. Micro-F1 was 53.00% for Mistral-7B and 54.00% for Qwen-2.5-3B, compared with 7.18% and 0.31% under encrypted execution. These gaps show that the polynomial architecture and activation mismatch alone do not explain the collapse. The additional loss arises along the staged encrypted execution path, but we cannot distinguish CKKS numerical error from repacking or other implementation effects without stage-level diagnostics. We therefore report the measured Proj-Poly-HE latency without treating its accuracy collapse as an inherent limitation of deeper homomorphic computation.

G.5 Authenticated-encryption baseline.

Under the same trusted-component model, AES-GCM achieves comparable Micro-F1 to Repr-HE (55.80% vs. 55.81%) with lower encryption/decryption time (0.06 vs. 15.29 ms) and total latency (22.10 vs. 26.60 ms). CKKS enables computation on ciphertext, but Repr-HE does not use this capability. AES-GCM is therefore more efficient for representation transfer alone; Repr-HE's result should be interpreted as a boundary-selection comparison rather than a benefit of homomorphic computation.