MYGO: CROSS-MODULE CO-DESIGN FOR COMPACT NON-INTERACTIVE TRANSFORMER INFERENCE WITH FULLY HOMOMORPHIC ENCRYPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fully homomorphic encryption (FHE) enables non-interactive secure Transformer inference, but fitting the complete computation into a compact configuration while hiding request shape remains challenging. We present MYGO, which addresses this challenge by co-designing three components under a fixed deployment contract: (i) a *compact FHE configuration*, using $N = 2^{15}$ RNS-CKKS with a common level-4 post-bootstrap interface; (ii) *fixed execution for request-shape privacy*, designing Standardized Interleaved Batching (SIB) with encrypted validity metadata to hide within-container occupancy and sequence lengths for 1–128 variable-length inputs; and (iii) *cross-module numerical compilation*, jointly distilling FFN affine layers and cubic activations, selecting site-specific minimax circuits, and folding representation factors into affine maps to avoid separate online scaling operations. MYGO's BERT-base achieves 92.09% plaintext accuracy on SST-2. Its full encrypted graph processes 128 inputs in 2,788 s with 22.37 GiB peak GPU memory on one RTX 5090, taking only 2.1% longer than with one active input. Against a matched $N = 2^{16}$ control, evaluation-key payload, post-bootstrap resident GPU memory, and synchronized bootstrapping latency decrease by 50.3%, 35.1%, and 22.9%, respectively. The same configuration executes LLaMA-3-8B's complete first decoder layer with median output MAE of 4.65×10^{-4} against FP32. These results demonstrate the feasibility of resource-efficient, non-interactive FHE Transformer inference on a single consumer GPU.[1]

1 INTRODUCTION

Cloud-hosted Transformer inference raises privacy concerns when clients submit sensitive inputs to a cloud service provider (CSP) (Chen et al., 2022). Secure multiparty computation (MPC) and hybrid homomorphic encryption (HE)–MPC protocols protect input confidentiality, but typically require repeated client–server interaction, making communication a potential bottleneck (Luo et al., 2024; Pang et al., 2024; Lu et al., 2025; Xu et al., 2025; Zhang et al., 2025c). Fully homomorphic encryption (FHE) enables the CSP to evaluate encrypted inputs without further client interaction, but FHE-based secure Transformer inference remains computationally expensive (Zhang et al., 2025a; Moon et al., 2025; Park et al., 2025; Yang et al., 2025; Zhang et al., 2026b; Gao et al., 2026).

Reducing this cost requires reconciling cryptographic, numerical, and packing constraints. In the Cheon–Kim–Kim–Song (CKKS) scheme (Cheon et al., 2017; 2019), ring dimension determines slot capacity and strongly affects ciphertext and key storage (Choi et al., 2026). At fixed security, a smaller ring tightens the admissible modulus budget and leaves less multiplicative depth between costly bootstraps (Albrecht et al., 2018), motivating low-depth numerical designs (Park et al., 2025; Xie et al., 2026b). Batching improves amortized throughput, but shape-dependent packing can make layouts and rotations vary with batch and sequence dimensions (Zhang et al., 2026b; Wang et al., 2026). Even with a fixed interleaving stride, plaintext validity masks reveal occupancy and per-input sequence lengths (Zhang et al., 2025b).

Prior work addresses individual aspects: NEXUS (Zhang et al., 2025a) and THOR (Moon et al., 2025) establish non-interactive FHE Transformer inference; Powerformer (Park et al., 2025) adapts models for low-depth nonlinearities; Rho et al. (2025) modify the Transformer architecture with Gaussian kernels and LoRA to reduce HE overhead; MOAI (Zhang et al., 2026b) and ARION (Yang

[1] Code is available for review at `https://anonymous.4open.science/r/MYGO-B24D/`, and it will be publicly released upon acceptance.

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.

et al., 2025) optimize layout-aware attention; and Euston (Gao et al., 2026) and STIP (Wang et al., 2026) improve packing and encrypted matrix computation. However, mismatches in ciphertext layouts, numerical ranges, and level requirements can require extra conversions or bootstrapping, increasing latency and memory use. The central challenge is therefore to *coordinate these interfaces across the complete Transformer graph within a compact FHE configuration, while keeping server execution fixed across batch occupancies and sequence lengths*.

Our Contributions. MYGO jointly designs cryptographic parameters, packed execution, and numerical circuits for resource-efficient, non-interactive FHE Transformer inference under a fixed deployment contract. Figure 1 summarizes the design.

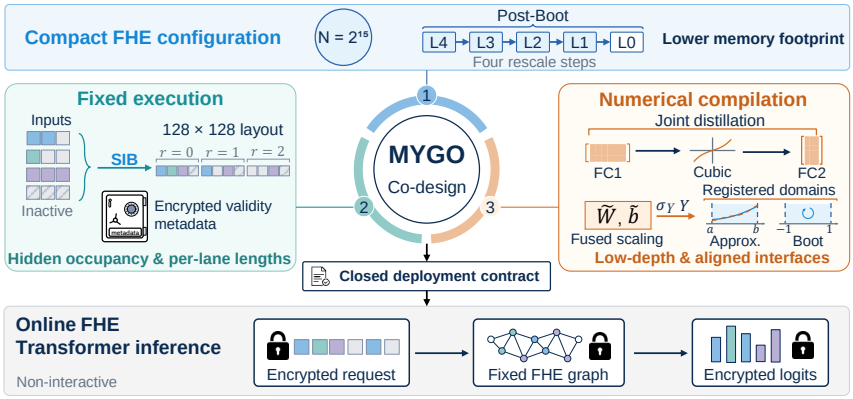


Figure 1: Overview of MYGO.

- **Compact FHE configuration.** MYGO co-designs RNS-CKKS parameters and circuit depth around a common level-4 post-bootstrap interface, enabling secure Transformer inference at $N = 2^{15}$. Computation between refreshes fits within four Rescales. Compared with a matched $N = 2^{16}$ control, this configuration reduces evaluation-key payload, post-bootstrap resident GPU memory, and synchronized bootstrapping latency by 50.3%, 35.1%, and 22.9%, respectively.
- **Fixed execution for request-shape privacy.** MYGO makes encrypted execution independent of within-container occupancy and sequence lengths by combining Standardized Interleaved Batching (SIB) with encrypted validity metadata. SIB packs 1–128 variable-length inputs into a fixed 128×128 lane–token grid; encrypted validity metadata excludes padded contributions and supplies safe inputs to nonlinear circuits. Both preserve batch amortization and hide metadata from the CSP while keeping ciphertext topology, evaluation keys, and the server graph fixed.
- **Cross-module numerical compilation.** MYGO compiles compatible numerical interfaces across attention, layer normalization, and FFNs. It jointly distills each FFN's two affine layers and cubic activation to meet the four-Rescale budget without bootstrapping the wide intermediate representation. Other nonlinear sites use site-specific minimax circuits with denominator guards and checked bootstrapping input domains. Public representation factors $\boldsymbol{\sigma}$ are folded into affine maps to align module amplitudes without separate online scaling operations. MYGO's BERT-base model achieves 92.09%/90.93%/70.49% accuracy on SST-2/QNLI/RTE.

MYGO evaluates a full 128-input BERT-base container in 2,788 s (21.78 s/input at full occupancy), with 22.37 GiB peak GPU memory on one RTX 5090. Under the same configuration, MYGO also executes LLaMA-3-8B's complete first decoder layer, with no observed domain violations and median output MAE of 4.65×10^{-4} against FP32.

2 SYSTEM MODEL

Deployment contract. MYGO assumes a trusted client and an honest-but-curious CSP. Figure 2 outlines the two-phase protocol. The client retains its secret key and registers public and evaluation keys once per key domain. The CSP caches these keys and holds the compiled model.

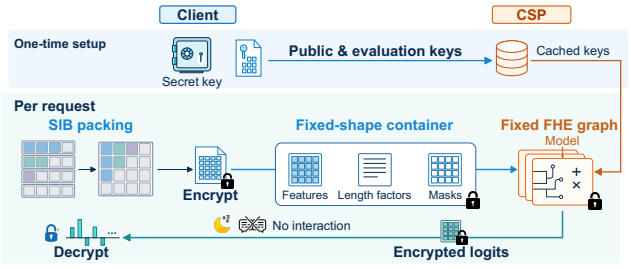


Figure 2: MYGO's two-phase client–server protocol.

For each request, the client uses SIB to pack up to $B_{\max}$ inputs, each of at most $M_{\max}$ tokens. It encrypts input features, length features, and validity masks into a *container* whose ciphertext counts and layouts remain fixed across supported occupancies and lengths. The CSP evaluates the fixed graph and returns encrypted logits for client decryption without intermediate interaction.

Definition 2.1 (Closed deployment contract)**.** Let $\mathcal{G}$ be a compiled server evaluation graph and $\mathfrak{R}$ its supported request family. A *closed deployment contract* is defined by

$$\mathcal{C} = (\Theta_{\mathrm{FHE}}, \Theta_{\mathrm{num}}, \Theta_{\mathrm{exec}}) \quad \longrightarrow \quad \begin{cases} \Theta_{\mathrm{FHE}} = (N, \mathbf{Q}, P), \\ \Theta_{\mathrm{num}} = (\ell_{\mathrm{post}}, \Delta_{\mathrm{ct}}, \boldsymbol{\sigma}, \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}_{\mathrm{Boot}}), \\ \Theta_{\mathrm{exec}} = (\mathcal{G}, B_{\max}, M_{\max}, \Pi, \mathcal{V}, \mathcal{K}). \end{cases}$$

These fix the cryptographic parameters, numerical interfaces, and execution structure. Here Π, $\mathcal{V}$, and $\mathcal{K}$ specify ciphertext topology, encrypted-validity semantics, and evaluation keys. Closure requires compatible module interfaces and a fixed public execution structure throughout $\mathfrak{R}$.

Leakage and privacy. Request-shape privacy hides occupancy and lengths within each container; container counts, submission times, and key reuse remain observable. For a request sequence $\mathbf{R}$, let n_{ctr} and $\boldsymbol{\tau}$ denote container count and submission times. The public setup is $\mathsf{pub} = (\mathcal{M}, \mathsf{keys}_{\mathrm{kd}}, \mathcal{C})$, where $\mathcal{M}$ is the compiled model and $\mathsf{keys}_{\mathrm{kd}}$ the reused public and evaluation-key bundle. Permitted leakage is $\mathcal{L}_{\mathcal{C}}(\mathbf{R}; \mathsf{pub}) = (\mathcal{M}, \mathsf{keys}_{\mathrm{kd}}, \mathcal{C}, n_{\mathrm{ctr}}, \boldsymbol{\tau})$. We suppress pub when clear. The contract fixes capacity, ciphertext serialization, public metadata, and the server graph.

Definition 2.2 (Request-shape privacy)**.** For every probabilistic polynomial-time adversary choosing two supported, polynomial-size request sequences $\mathbf{R}_0, \mathbf{R}_1$ with equal permitted leakage, the corresponding CSP views are computationally indistinguishable:

$$\mathsf{View}^{\mathsf{Mygo}}_{\mathrm{CSP}}(\mathbf{R}_0) \approx_c \mathsf{View}^{\mathsf{Mygo}}_{\mathrm{CSP}}(\mathbf{R}_1).$$

3 Compact FHE Configuration

Mygo co-designs RNS-CKKS parameters and circuit depth for Transformer evaluation at $N = 2^{15}$, halving the polynomial coefficient count and slot capacity relative to MOAI and ARION configurations (Zhang et al., 2026b; Yang et al., 2025). The compiled circuits share a common level-4 post-bootstrap interface and keep computation between refreshes within four Rescales, enabling their composition across the Transformer graph.

3.1 RNS-CKKS Parameterization

We use the residue-number-system (RNS) form of CKKS at scale $\Delta_{\mathrm{ct}} = 2^{31}$. Above a 41-bit level-0 modulus, levels 1–4 append four 31-bit primes and levels 5–6 append two 30-bit primes, yielding $\log_2 Q_4 \approx 165$ and $\log_2 Q_6 \approx 225$. Fresh ciphertexts enter at level 6; bootstrapping (Boot) returns to level 4, leaving four Rescales before level 0. Levels 7–16 add eight 55-bit and two 52-bit logical factors reserved for internal bootstrapping and key switching.

With $\log_2 Q_{16} = 769.0117$ and a 111.9870-bit auxiliary modulus P, the largest evaluation-key basis satisfies $\log_2(Q_{16}P) = 880.9988 < 881$, within the reference modulus budget adopted for the 128-bit classical-security target at $N = 2^{15}$ (Albrecht et al., 2018). This full-basis domain uses a dense secret of Hamming weight $h = N/2$. Boot uses sparse-secret encapsulation (Bossuat et al., 2022), temporarily switching ciphertexts to a weight-32 secret. In our configuration, evaluation-key samples under this secret use only the 96.983-bit short basis qp, never the full 881-bit basis. The 128-bit figure is a reference target, not a verified lower bound for the deployed fixed-weight secret: the archived full-basis MATZOV/default estimate is 125.90 bits. Appendices D.2 and D.3 report estimator sensitivity and the complete modulus accounting.

3.2 Meeting the Four-Rescale Budget

The FFN makes intermediate bootstrapping costly. For example, in BERT-base, column-packed FC1 expands 768 ciphertexts to 3,072, so refreshing this intermediate state involves four times as many ciphertexts as refreshing the 768-ciphertext output after FC2.

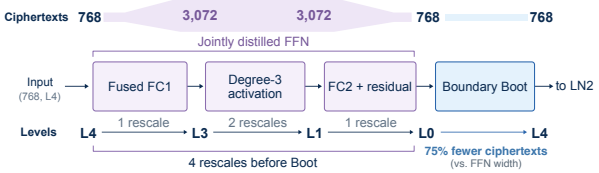


Figure 3: Four-rescale FFN schedule.

To avoid refreshing the wide intermediate representation, MYGO jointly distills FC1, a layer-specific cubic activation p_ℓ, and FC2. Training combines task loss with teacher matching at the logits, encoder outputs, and FFN outputs, allowing the surrounding affine layers to adapt to the cubic activation. The resulting FFN follows the $4 \to 3 \to 1 \to 0$ level path, meeting the four-Rescale budget and deferring Boot until after FC2, as shown in Figure 3. Both plaintext and FHE evaluations use the same cubic activation. Section 5 integrates this FFN schedule with the remaining nonlinear operators under the same level-4 interface. Appendix F.1 describes distillation.

4 FIXED EXECUTION FOR REQUEST-SHAPE PRIVACY

MYGO combines Standardized Interleaved Batching (SIB) with encrypted validity metadata to hide batch occupancy and per-input sequence lengths. SIB fixes slot placement, while encrypted masks handle unused positions under a fixed server evaluation graph.

4.1 STANDARDIZED INTERLEAVED BATCHING

Interleaved batching (IB) packs corresponding entries from multiple inputs into adjacent ciphertext slots for parallel evaluation (Jiang et al., 2018). MOAI (Zhang et al., 2026b) adopts IB with a fixed interleaving stride, but its released implementation derives validity masks from plaintext sequence lengths, exposing batch occupancy and per-input lengths (Zhang et al., 2025b).

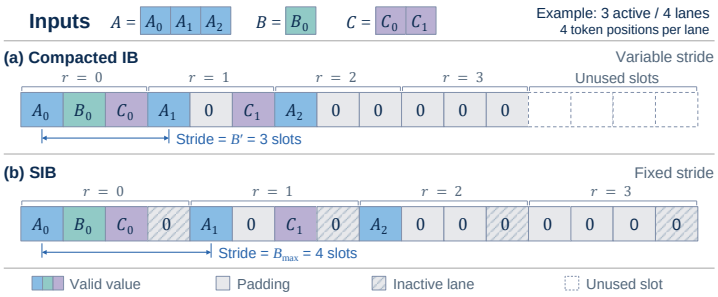


Figure 4: SIB versus occupancy-adaptive compaction.

To keep packed execution independent of batch occupancy, MYGO introduces Standardized Interleaved Batching (SIB), which fixes both lane capacity and interleaving stride at $B_{\max}$. A t-token shift therefore always uses $\mathrm{Rot}_{tB_{\max}}$. Figure 4 contrasts SIB with occupancy-adaptive compaction: SIB reserves inactive lanes and preserves slot positions and rotation offsets across occupancies.

Formally, let $M_{\max}$ divide $N/2$ and define $B_{\max} = N/(2M_{\max})$. Let $m_b \in \{0, \dots, M_{\max}\}$ denote the private length of lane $0 \le b < B_{\max}$, with $m_b = 0$ indicating inactivity. Zero-pad each input to $M_{\max}$ positions and define $q_b(r) = \mathbf{1}[r < m_b]$. For feature a, SIB constructs

$$\widetilde{x}_a[B_{\max} r + b] = q_b(r) x_{b,r,a}, \qquad 0 \le r < M_{\max}. \tag{1}$$

With $N = 2^{15}$ and $M_{\max} = 128$, each feature uses a fixed 128×128 lane–token grid, amortizing the common workload over 128 inputs at full occupancy.

4.2 ENCRYPTED VALIDITY METADATA

SIB reserves padded token positions and inactive input lanes, but their numerical effects still require masking. Nonlinear operations can turn padded zeros into nonzero contributions (e.g., $\exp(0) = 1$) that bias RowSum, while fully invalid rows can produce zero denominators outside the reciprocal approximation interval. MYGO uses encrypted masks to exclude these contributions and substitute safe nonlinear inputs without exposing occupancy or per-input lengths.

Specifically, the client encrypts $q_b(r)$ as one row-validity ciphertext ct_M^{row} and, for each attention-diagonal offset δ, encrypts the packed mask $M^C_{b,\delta}(r) = q_b(r)\, q_b((r+\delta) \bmod M_{\max})$. The attention range-alignment circuit also receives four encrypted length-derived features (Section 5.2), packed in a fixed layout with zeros for inactive lanes.

Before exponential approximation, invalid affine-mapped scores $\bar{U}$ are replaced by a fixed value t_{pad} inside the approximation interval:

$$Z_{b,\delta}(r) = M^C_{b,\delta}(r) \bar{U}_{b,\delta}(r) + \big(1 - M^C_{b,\delta}(r)\big) t_{\mathrm{pad}}. \tag{2}$$

Reapplying the mask after exponential reconstruction removes padded weights from CV and RowSum. The zero-offset diagonal mask M_0^C identifies invalid query rows for safe denominator substitution before reciprocal approximation. The row-validity mask replaces invalid LN variance inputs before reciprocal-square-root approximation. Only encrypted contents vary with request shape; ciphertext topology, rotations, evaluation keys, and the server graph remain fixed. Under the assumptions of Theorem C.1, MYGO satisfies request-shape privacy as defined in Definition 2.2.

5 CROSS-MODULE NUMERICAL COMPILATION

MYGO compiles compatible layouts, levels, and numerical ranges within the fixed cryptographic budget. The offline compiler profiles the frozen student on calibration inputs and fits site-specific circuits within selected numerical intervals. It accepts a circuit only after approximation-error, bootstrapping-domain, level, and scale checks, then freezes its coefficients, guards, and interface maps. Affine fusion carries the required scaling across module boundaries without extra online conversions. Figure 5 summarizes MYGO's compiled encoder and classification workflow.

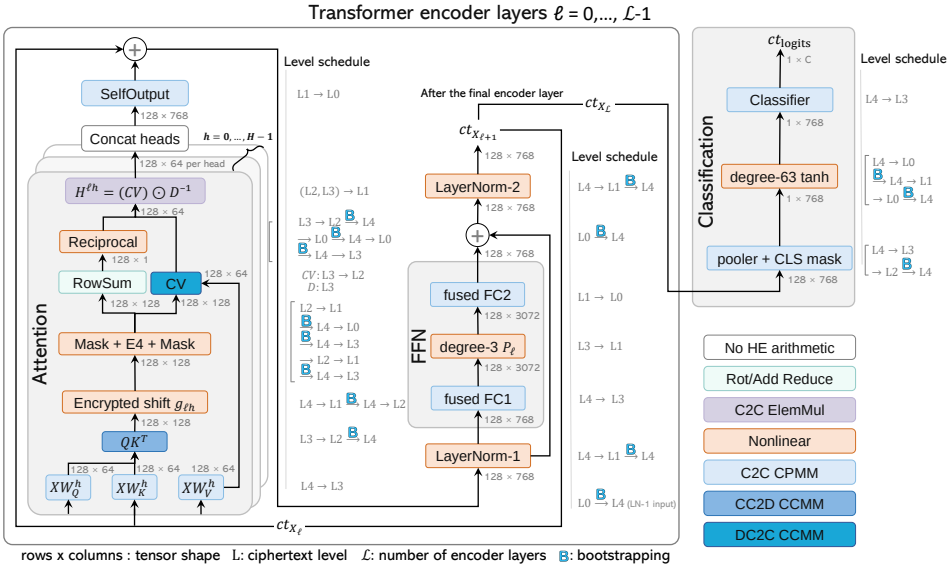


Figure 5: Compiled FHE workflow for MYGO's encoder and classification head. Attention levels are shown for $\ell \geq 1$; Layer 0 uses $6/5/4$ for the input, Q/K/V, and scores, without the score refresh.

5.1 COMPILED INTER-MODULE INTERFACES

Let $\mathrm{Dec}(ct[Z;\sigma_Z]) \simeq \sigma_Z Z$, where the *public representation factor* σ_Z controls amplitude independently of the CKKS scale Δ_{ct}. For $Y = ZW + b$, the compiler folds the factor transition into $\widetilde{W} = (\sigma_Y/\sigma_Z)W$ and $\widetilde{b} = \sigma_Y b$, giving $\mathrm{Dec}\Big(ct[Z;\sigma_Z]\widetilde{W} + \widetilde{b}\Big) \simeq \sigma_Y Y$ without an extra scaling multiplication or level.

The factors $\sigma_{0,\ell}$, $\sigma_{1,\ell}$, $\sigma_{2,\ell}$, and $\sigma_{3,\ell}$ align the pre-LN1 residual, LN1 normalized core, pre-LN2 residual, and post-LN2 output, respectively. Each encoder layer returns 768 column-packed ciphertexts at factor $\sigma_{3,\ell}$ and level 4, with $\sigma_{\mathrm{in},\ell} = \sigma_{3,\ell-1}$ for $\ell = 1, \ldots, 11$.

For attention, MYGO uses Double-BSGS multiplication (Yang et al., 2025) to produce rotated-diagonal scores and column-packed CV products before head normalization. At the LN1–FFN boundary, the LN1 affine map is folded into FC1 and evaluated separately on the residual branch. Appendix Table 5A and Appendix F.3 give the exact maps and schedules, respectively.

5.2 ENCRYPTED ATTENTION WITH RANGE ALIGNMENT

MYGO compiles each attention head into four stages with fixed public depth budgets and approximation intervals. Encrypted range alignment adapts the numerical inputs while keeping this execution schedule unchanged:

(i) Scaled scores. For one SIB lane and head, the compiler folds $d_h^{-1/4}$ and representation transitions into the query/key projections; the value path uses $\sigma_V = 1$. The score ciphertext directly represents $S_{ij}^{\ell h} = Q_i^{\ell h}\big(K_j^{\ell h}\big)^{\mathsf{T}}/\sqrt{d_h}$ without a separate factor conversion.

(ii) Encrypted row-common range alignment. A shift shared by all scores in a query row aligns them with the head's fixed exponential interval. Let $M_{ij} = q_i q_j$ be the encrypted pair-validity mask. For an active lane of length m, the client supplies $\rho_m = 1/m$, ω_m, $\bar{\ell}_m$, and $\bar{c}_m$ under encryption (Appendix F.3). The circuit computes $\mu_i = \rho_m \sum_{j<M_{\max}} M_{ij} S_{ij}$, $t_i = \omega_m \sum_{j<M_{\max}} M_{ij}(S_{ij} - \mu_i)^2$, and $\xi_i = 0.02734375 t_i - 0.875$, followed by the *row-common shift*

$$g_{\ell h,i} = \mu_i + P_{\ell h,2}(\xi_i) + a_{\ell h}\bar{\ell}_m + b_{\ell h}\bar{c}_m + d_{\ell h}(1-\rho_m) - s_\ell. \tag{3}$$

The polynomial and length terms fit $\mathrm{LSE}(S_i) - \mu_i$ offline. Here $\mathrm{LSE}(S_i) = \ln \sum_{j:q_j=1} e^{S_{ij}}$ sums over valid keys, and s_ℓ aligns $U_{ij} = S_{ij} - g_{\ell h,i}$ with the exponential interval. The shared shift preserves Softmax in exact arithmetic without a pseudocount.

(iii) Mask-aware exponential. On $[-W_{\ell h}, 0]$, valid coordinates use $Z_{ij} = 1 + 2U_{ij}/W_{\ell h}$. A degree-15 minimax polynomial in the Chebyshev basis and two squarings reconstruct the exponential:

$$P_{15,\ell h}^E(Z_{ij}) \simeq \beta_{\ell h} e^{U_{ij}/4}, \widehat{E}_{\ell h}(U_{ij}) \quad = \frac{c_\ell^E}{\beta_{\ell h}^4}\left[P_{15,\ell h}^E(Z_{ij})\right]^4 \simeq c_\ell^E e^{U_{ij}}. \tag{4}$$

After masking and pseudocount addition, the target weights in source units are $A_{ij}^{\ell h} = M_{ij}\big(e^{U_{ij}} + \bar{\tau}_{\ell h}\rho_m\big)$, where $\bar{\tau}_{\ell h} = \tau_{\ell h}/c_\ell^E$. Their packed representation C uses factor $\sigma_{C,\ell h} = c_\ell^E$. The pseudocount supplies a positive denominator margin but perturbs the normalized weights; its value is fixed and checked offline.

(iv) Fixed-bound normalization. All reductions use the public bound $M_{\max}$. In source units,

$$\begin{aligned} D_i^{\ell h} &= \sum_{j<M_{\max}} A_{ij}^{\ell h}, \qquad D_i^{\ell h,\mathrm{safe}} = D_i^{\ell h} + d_{\mathrm{fill}}(1-q_i), \\ H_{ia}^{\ell h} &= \left(\sum_{j<M_{\max}} A_{ij}^{\ell h} V_{ja}^{\ell h}\right)\big(\widehat{D_i^{\ell h,\mathrm{safe}}}\big)^{-1}. \end{aligned} \tag{5}$$

The zero-offset diagonal mask supplies q_i. Safe fill is encoded as $\sigma_{D,\ell h} d_{\mathrm{fill}}(1-q_i)$. The denominator and numerator factors are $\sigma_{D,\ell h} = \sigma_{C,\ell h}$ and $\sigma_{N,\ell h} = \sigma_{C,\ell h}\sigma_V$, respectively. Output restoration uses $\sigma_{H,\ell h}/\sigma_{N,\ell h}$. Each head returns 64 column-packed ciphertexts at factor $\sigma_{H,\ell h}$ and level 1.

5.3 Closing the Encoder and Classification Graph

The remaining paths align residuals and nonlinear modules, restore level 4 after each encoder layer, and produce encrypted logits.

(i) Output projection/residual/LN1. Once per layer, Attention is converted to its real component: $\overline{ct}_H = ct_H + \mathrm{Conj}(ct_H)$, with $\mathrm{Dec}(\overline{ct}_H) \simeq 2\sigma_{H,\ell}\,\mathrm{Re}(H_\ell)$. The output projection uses $\widetilde{W}_O = \sigma_{0,\ell} W_O/(2\sigma_{H,\ell})$ and $\widetilde{b}_O = \sigma_{0,\ell} b_O$, while the residual branch absorbs $\sigma_{0,\ell}/\sigma_{\mathrm{in},\ell}$. Both branches are reduced to level 0 before addition, yielding $\mathrm{Dec}(ct_{R_1}) \simeq \sigma_{0,\ell}\big(\mathrm{Re}(H_\ell)W_O + b_O + X_\ell\big)$. Boot refreshes the 768 ciphertexts to level 4 for LN1. LN1 encodes ϵ and invalid-row fill at the variance factor $\sigma_{0,\ell}^2$, removes $\sigma_{0,\ell}^2$ in the reciprocal-square-root input map, and returns normalized activations at $\sigma_{1,\ell}$.

(ii) Fused polynomial FFN. For $z_\ell = ((\gamma_1 \odot n_1 + \beta_1)W_1 + b_1 - c_\ell)/s_\ell^{\mathrm{in}}$, where n_1 is the LN1 normalized core and $c_\ell, s_\ell^{\mathrm{in}}$ are the input center and scale, the compiler folds the LN1 affine map, FC1, input normalization, and representation-factor conversion into

$$ct_z = ct[n_1; \sigma_{1,\ell}]\frac{\mathrm{Diag}(\gamma_1)W_1}{\sigma_{1,\ell} s_\ell^{\mathrm{in}}} + \frac{\beta_1 W_1 + b_1 - c_\ell}{s_\ell^{\mathrm{in}}}, \tag{6}$$

giving $\mathrm{Dec}(ct_z) \simeq z_\ell$. The cubic p_ℓ consumes two Rescales. FC2 absorbs the affine output restoration $\mathcal{R}_\ell^{\mathrm{out}}$ and returns ciphertexts at factor $\sigma_{2,\ell}$, matching the LN1 residual branch. This realizes the four-Rescale schedule in Figure 3, with Boot applied after FC2 reduces the ciphertext count to 768.

(iii) LN2 and classification. LN2 returns 768 column-packed ciphertexts at $\sigma_{3,\ell}$ and level 4. After the final BERT-base layer, the pooler absorbs $1/\sigma_{3,11}$ and maps to the registered tanh interval. A fixed public row-0 selector acts identically in every lane to extract classification-token positions, requiring no additional ciphertext. Polynomial tanh and the affine classifier produce encrypted logits; the client discards inactive-lane outputs.

6 EVALUATION

We evaluate MYGO's BERT-base task accuracy, layer-local FHE fidelity, complete-graph cost, and deployment-contract ablations, then assess whether the same design supports a complete LLaMA-3-8B decoder layer.

6.1 BERT-BASE RESULTS

Accuracy and FHE fidelity. With cubic activations in all 12 FFNs, MYGO's BERT-base model achieves 92.09%, 90.93%, and 70.49% accuracy on SST-2, QNLI, and RTE, respectively—0.34, 0.61, and 2.87 percentage points below the teacher (Table 1A). These results show that the four-Rescale FFN remains effective across three classification tasks.

Table 1: Task accuracy and layer-balanced FHE fidelity.

A. Task accuracy (%)

Task (n)	Teacher	MYGO	Δ (pp)
SST-2 (872)	92.43	92.09	-0.34
QNLI (5,438)	91.54	90.93	-0.61
RTE (244)	73.36	70.49	-2.87

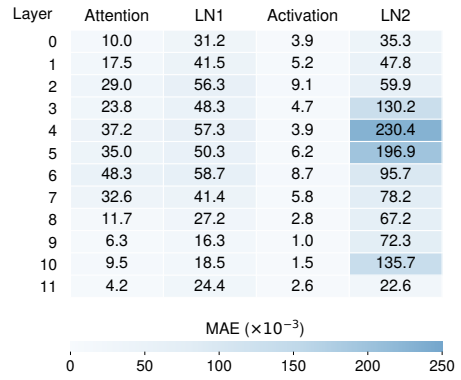
B. Layer-balanced FHE summaries (SST-2)

Checkpoint	Median $[Q_{25}, Q_{75}]$	P_{95}/P_{99}
Attention	0.0068 [0.0028, 0.0163]	0.0980/0.2128
LN1	0.0337 [0.0158, 0.0585]	0.1056/0.1502
Activation	0.0022 [0.0008, 0.0054]	0.0154/0.0300
LN2	0.0605 [0.0278, 0.1080]	0.1946/0.2652

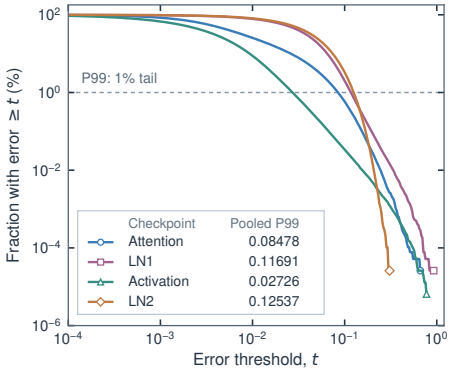
Panel A: Inputs longer than 128 tokens are excluded, not truncated.

To assess FHE fidelity, we evaluate 12 independent SST-2 encoder-layer probes, each initialized from its corresponding reference checkpoint and repeated in three fresh FHE processes. Errors are *layer-local*, not cumulative across layers. For each checkpoint, we take the median of the 12 per-layer median errors; the largest is approximately 0.0605 (Table 1B). Figure 6a shows per-layer MAE below 0.01 for Activation and below 0.06 for Attention and LN1; LN2 reaches 0.2304 at Layer 4. Figure 6b pools valid-coordinate Layer-0 errors from three runs on 100 SST-2 inputs (1,675 valid tokens per run). Pooled P99 ranges from 0.02726 to 0.12537 across checkpoints.

Together, these probes demonstrate numerical fidelity across the encoder's main modules under the compact $N = 2^{15}$ configuration and common level-4 interface, with the largest layer-local errors observed at LN2. Appendix Tables 12A and 12B give the layer-level statistics; Appendix Figure 11 shows coordinate-wise Layer-0 profiles.



(a) Per-layer MAE ($\times 10^{-3}$).



(b) Pooled Layer-0 error tails.

Figure 6: Layer-local FHE fidelity.

Complete-graph cost. MYGO evaluates the complete encrypted BERT-base graph—12 encoder layers, pooler, and classifier—for 128 inputs in 2,788 s (21.78 s/input at full occupancy) on one RTX 5090, with 22.37 GiB peak GPU memory, as shown in Table 2. One active input takes 2730 $\pm$ 21 s (mean $\pm$ sample standard deviation, three runs), so filling all 128 lanes increases measured

batch time by only 2.1%. Across all 12 layers, cubic activations cost only 0.073 amortized s/input, whereas the four 768-ciphertext refresh stages per layer collectively cost 5.606 s/input (Appendix Table 10). This breakdown motivates deferring FFN refreshes until after width contraction.

Table 2: Reported BERT-base capacity-amortized runtime under native configurations.

System	Hardware	Capacity	Timed scope	Retrain	s/input ↓
THOR (Moon et al., 2025)	A100	1	Enc.+Pool.+Class.	No	602.26
Powerformer (Park et al., 2025)	A100	1	Enc.+Pool.+Class.	Yes	344.52
MOAI (Zhang et al., 2026b)	H200	256	Enc. only	No	141.30
ARION (Yang et al., 2025)	Xeon 6530	256	Enc. only	No	224.91
Euston (Gao et al., 2026)	RTX 6000 Ada	32	Enc.+ArgMax	No	≈ 46.00
STIP (Wang et al., 2026)	A100	32	Enc.+Class.	Yes	64.23
MYGO	**RTX 5090**	**128**	**Enc.+Pool.+Class.**	**Yes**	**21.78**

Notes. Enc. denotes all 12 encoder layers, Pool. the encrypted pooler, and Class. the encrypted classifier. MYGO's runtime is based on a single 128-input run.

Among the systems in Table 2, MYGO has the lowest reported capacity-amortized runtime, including encrypted pooler and classifier evaluation. Euston and STIP have 2.11× and 2.95× its capacity-amortized runtime, respectively.[2] The reported MOAI and ARION timings end at the final encoder bootstrap. Following the end-to-end scope concerns raised by Zhang et al. (2026b), we omit NEXUS (Zhang et al., 2025a) from this comparison. Appendix Tables 10 and 11 detail MYGO's costs.

6.2 DEPLOYMENT-CONTRACT ABLATIONS

Compact FHE configuration. Against a matched ring-only $N = 2^{16}$ control with the same extended-modulus budget, post-Boot level, and operator topology, the deployed $N = 2^{15}$ ring in MYGO reduces raw evaluation-key payload, synchronized Boot latency, and post-Boot resident GPU memory by 50.3%, 22.9%, and 35.1%, as shown in Figure 7. Warm denotes a request upload with cached keys; Cold adds key registration. Their raw payloads fall by 50.0% and 50.3%, extending savings to client uploads. Costs are normalized to the control, labels give absolute values, and Memory† denotes post-Boot resident memory. Appendix Table 13A details these matched resource microbenchmarks.

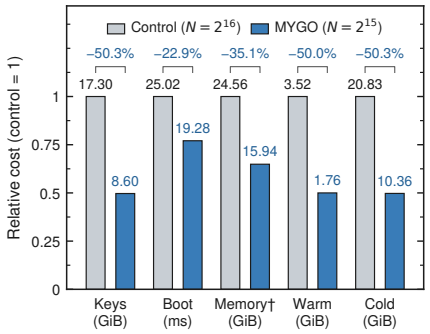


Figure 7: Compact-ring resource savings.

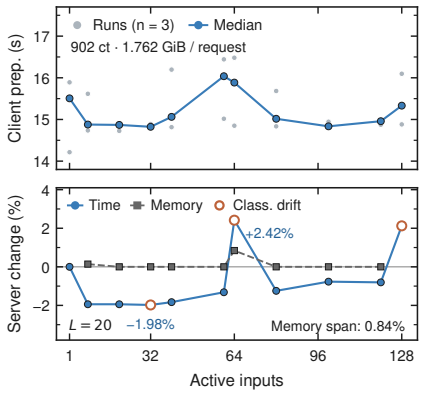


Figure 8: Resources across request shapes.

Fixed execution across request shapes. Across 11 heterogeneous client profiles (1–128 active inputs, 5–8,256 aggregate valid tokens), each encrypted request container contains 902 ciphertexts with a raw payload of 1.762 GiB. Figure 8 separates client preparation (top) from an independent server occupancy sweep at sequence length 20 (bottom). Server time varies by -1.98% to $+2.42\%$ relative to the three-run one-input mean. Across the nine profiles with memory telemetry, the maximum–minimum spread is 192 MiB, or 0.84% of their median. This supports stable resource use across occupancies. Orange circles indicate observed classification drift, so the affected runs support resource measurements rather than multi-input correctness (Appendix H.2).

[2]These ratios compare native hardware, configurations, and timed scopes; they are not controlled speedups.

Cross-module numerical alignment. We ablate σ_0 (pre-LN1 residual), σ_1 (LN1 normalized core), σ_2 (pre-LN2 residual), and σ_3 (post-LN2 output), omitting the layer index for this Layer-0 experiment. Selected factors are set to one and the corresponding affine compensation is recompiled. Figure 9(a) shows increased pre-Boot P99 magnitudes under single-factor ablations. Figure 9(b) shows up to 39.50% of valid coordinates outside $[-1, 1]$ under single and joint ablations, versus none observed with BASE. Even $\sigma_3 = 1$ yields 0.36% violations despite P99 $= 0.845 < 1$, demonstrating the need to audit tail coverage. Thus, MYGO's scaling, folded into affine maps, keeps the valid coordinates at all four audited interfaces within their Boot domains on this workload without additional online conversions or bootstraps, preserving the compact depth budget. Appendix Tables 14B and 14C provide detailed results.

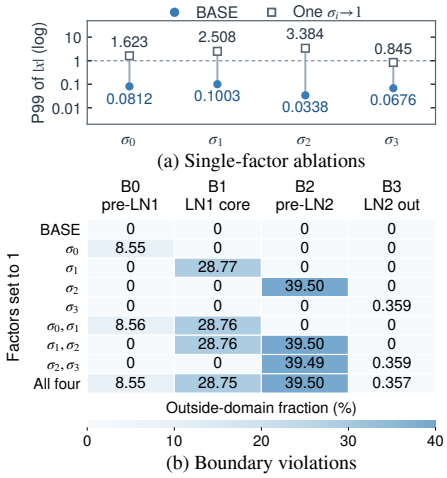


Figure 9: Layer-0 numerical alignment.

6.3 LLAMA-3-8B EVALUATION

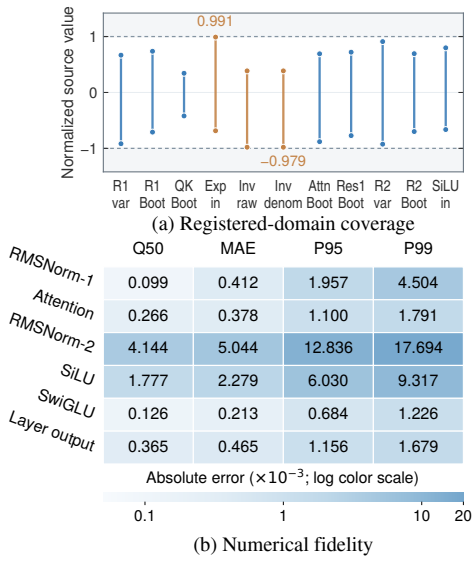


Figure 10: LLaMA-3-8B Layer-0 validation.

Using the original weights and the same $N = 2^{15}$ configuration, MYGO executes LLaMA-3-8B's complete first decoder layer. We freeze the numerical settings and evaluate 2,553 eight-token WikiText-2 test prefixes excluded from calibration, with three independent FHE runs per batch. Figure 10(a) shows no observed domain violations at the 11 audited boundaries. Figure 10(b) reports errors at six checkpoints, with median pooled output MAE and P99 of 4.652×10^{-4} and 1.679×10^{-3} against FP32. These results support numerical compatibility across attention, normalization, and the full 14,336-dimensional FFN. The median batch runtime is 3,758.09 s on one RTX 5090, with 21.717 GiB sampled peak GPU memory and 73.756 GiB median peak host RAM. Runtime details appear in Appendix I.3.

Separately, we combine operator-level measurements and fitted costs from an earlier timing configuration to estimate the cost of 32 LLaMA-3-8B decoder layers and the output head. The capacity-amortized estimate is 38.796 s/input, compared with MOAI's reported 224.530 s/input (Zhang et al., 2026b), yielding a 5.79× cost ratio under the respective native configurations. Estimation details and assumptions appear in Appendix I.6.

7 CONCLUSION

MYGO co-designs cryptographic parameters, numerical circuits, and packed execution for non-interactive FHE Transformer inference at $N = 2^{15}$. Joint FFN distillation meets the four-Rescale budget without refreshing the wide intermediate representation, while affine fusion aligns module ranges without separate online scaling. SIB and encrypted validity metadata maintain fixed execution and hide within-container occupancy and lengths under the stated security assumptions. The complete 128-input BERT-base graph takes 2,788 s with 22.37 GiB peak GPU memory on one RTX 5090. Under the same configuration, MYGO also executes LLaMA-3-8B's complete first decoder layer, achieving median output MAE of 4.65×10^{-4} against FP32 with no observed domain violations on the test inputs. Full-model encrypted autoregressive generation remains future work.

Artificial Intelligence (AI) Use Statement

We used ChatGPT solely to improve the grammar and clarity of the manuscript. We also used the Paper Assistant Tool (PAT) provided through ICLR 2027 and incorporated its feedback where appropriate after careful evaluation. The authors reviewed all AI-assisted content and checked that editorial revisions preserved the intended scientific meaning. The authors take full responsibility for the final manuscript.

Ethics Statement

To the best of our knowledge, this work does not raise any specific ethical concerns.

Reproducibility Statement

Code availability and release details for MYGO's encrypted inference implementation are provided at `https://anonymous.4open.science/r/MYGO-B24D/`. Our experiments use public benchmarks, including GLUE and WikiText-2. Sections 3–5 and Appendices D–F describe the cryptographic parameters, packing and masking, task-specific distillation, and circuit algorithms. Section 6 and Appendices G–I specify preprocessing, input selection, hardware and software settings, error metrics, and aggregation across runs. We report task accuracy, layer-local FHE fidelity, and complete-graph execution costs separately. Appendix I distinguishes continuous first-layer LLaMA evaluation from independent output-head experiments and operator-cost estimates. Baseline runtimes are taken from the cited publications; Table 2 reports their hardware, packing capacities, and timed scopes.

References

Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018. URL `https://homomorphicencryption.org/wp-content/uploads/2018/11/HomomorphicEncryptionStandardv1.1.pdf`.

Jean-Philippe Bossuat, Juan Ramón 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*, pp. 521–541. Springer, 2022. doi: 10.1007/978-3-031-09234-3_26. URL `https://eprint.iacr.org/2022/024`.

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. THE-X: Privacy-preserving transformer inference with homomorphic encryption. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3510–3520. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.277. URL `https://aclanthology.org/2022.findings-acl.277/`.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, pp. 409–437. Springer, 2017. doi: 10.1007/978-3-319-70694-8_15. URL `https://doi.org/10.1007/978-3-319-70694-8_15`.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A full RNS variant of approximate homomorphic encryption. In *Selected Areas in Cryptography – SAC 2018*, pp. 347–368. Springer, 2019. doi: 10.1007/978-3-030-10970-7_16. URL `https://doi.org/10.1007/978-3-030-10970-7_16`.

Wonseok Choi, Jongmin Kim, and Jung Ho Ahn. Cheddar: A swift fully homomorphic encryption library designed for GPU architectures. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pp. 35–49. ACM, 2026. doi: 10.1145/3760250.3762223. URL `https://doi.org/10.1145/3760250.3762223`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Xinwen Gao, Shaojing Fu, Lin Liu, Zhuotao Liu, Yuchuan Luo, and Yongjun Wang. Euston: Efficient and user-friendly secure transformer inference with non-interactivity. In *2026 IEEE Symposium on Security and Privacy (SP)*, pp. 1443–1462. IEEE, 2026. doi: 10.1109/SP63933.2026.00048. URL https://doi.org/10.1109/SP63933.2026.00048.

hangenba. ARION: Privacy-preserving transformer inference framework. GitHub repository, 2026. URL https://github.com/hangenba/Arion/tree/bd5fcd7. Commit bd5fcd7.

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*, pp. 1209–1222. Association for Computing Machinery, 2018. doi: 10.1145/3243734.3243837. URL https://doi.org/10.1145/3243734.3243837.

Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and Wenguang Chen. BumbleBee: Secure two-party inference framework for large transformers. In *32nd Annual Network and Distributed System Security Symposium (NDSS 2025)*. The Internet Society, 2025. doi: 10.14722/ndss.2025.230057. URL https://www.ndss-symposium.org/wp-content/uploads/2025-57-paper.pdf.

Jinglong Luo, Yehong Zhang, Zhuo Zhang, Jiaqi Zhang, Xin Mu, Hui Wang, Yue Yu, and Zenglin Xu. SecFormer: Fast and accurate privacy-preserving inference for transformer models via SMPC. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13333–13348. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.790. URL https://aclanthology.org/2024.findings-acl.790/.

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. Association for Computing Machinery, 2025. doi: 10.1145/3719027.3765150. URL https://doi.org/10.1145/3719027.3765150.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. BOLT: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 4753–4771. IEEE, 2024. doi: 10.1109/SP54263.2024.00130. URL https://doi.org/10.1109/SP54263.2024.00130.

Dongjin Park, Eunsang Lee, and Joon-Woo Lee. Powerformer: Efficient and high-accuracy privacy-preserving language model with homomorphic encryption. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11090–11111. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.543. URL https://aclanthology.org/2025.acl-long.543/.

Donghwan Rho, Taeseong Kim, Minje Park, Jung Woo Kim, Hyunsik Chae, Ernest K. Ryu, and Jung Hee Cheon. Encryption-friendly LLM architecture. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6715b4e97be055687c1ecaf33913d358-Paper-Conference.pdf.

Zihao Wang, Rongmao Chen, Xinwen Gao, Yi Wang, Lin Liu, Zixin Lan, Zhaoyu Wang, Shaojing Fu, Qiong Wang, and Xinyi Huang. STIP: Efficient and secure non-interactive transformer inference via compact packing. Cryptology ePrint Archive, Paper 2026/174, 2026. URL https://eprint.iacr.org/2026/174.

Jianhang Xie, Sicheng Tan, Vishnu Naresh Boddeti, and Zhichao Lu. ATLAS: Automated approximation of transformers for efficient homomorphic inference in one hour. arXiv preprint arXiv:2607.23478, 2026a. URL https://arxiv.org/abs/2607.23478.

Xi Xie, Ran Ran, Jiahui Zhao, Bin Lei, Zhijie Jerry Shi, Wujie Wen, and Caiwen Ding. ULD-Net: Enabling ultra-low-degree fully polynomial networks for homomorphically encrypted inference. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0feb70f08090d165e6bbfa9c85d05e87-Abstract-Conference.html.

Tianshi Xu, Wen-jie Lu, Jiangrui Yu, Yi Chen, Chenqi Lin, Runsheng Wang, and Meng Li. Breaking the layer barrier: Remodeling private transformer inference with hybrid CKKS and MPC. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2653–2672. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/xu-tianshi.

Linhan Yang, Jingwei Chen, Wangchen Dai, Shuai Wang, Wenyuan Wu, and Yong Feng. ARION: Attention-optimized transformer inference on encrypted data. Cryptology ePrint Archive, Paper 2025/2271, 2025. URL https://eprint.iacr.org/2025/2271.

Jiawen Zhang, Xinpeng Yang, Lipeng He, Kejia Chen, Wen-jie Lu, Yinghao Wang, Xiaoyang Hou, Jian Liu, Kui Ren, and Xiaohu Yang. Secure transformer inference made non-interactive. In *32nd Annual Network and Distributed System Security Symposium (NDSS 2025)*. The Internet Society, 2025a. doi: 10.14722/ndss.2025.230868. URL https://www.ndss-symposium.org/ndss-paper/secure-transformer-inference-made-non-interactive/.

Linru Zhang, Xiangning Wang, Jun Jie Sim, Zhicong Huang, Jiahao Zhong, Huaxiong Wang, Pu Duan, and Kwok-Yan Lam. MOAI CPU reference implementation. GitHub repository, 2025b. URL https://github.com/dtc2025ag/MOAI/tree/8bcb0ea. Commit 8bcb0ea.

Linru Zhang, Xiangning Wang, Jun Jie Sim, Zhicong Huang, Jiahao Zhong, Huaxiong Wang, Pu Duan, and Kwok-Yan Lam. MOAI GPU reference implementation. GitHub repository, 2026a. URL https://github.com/dtc2025ag/MOAI_GPU/tree/3418055. Commit 3418055.

Linru Zhang, Xiangning Wang, Jun Jie Sim, Zhicong Huang, Jiahao Zhong, Huaxiong Wang, Pu Duan, and Kwok-Yan Lam. MOAI: Module-optimizing architecture for non-interactive secure transformer inference. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2f2b1d6bbd50865eca40e2774a057eef-Abstract-Conference.html.

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)*, 2025c. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/83a14a36de4502bac5b580db36e81858-Abstract-Conference.html.

APPENDIX

A RELATED WORK

Research on FHE Transformer inference reduces computational cost through ciphertext packing, operator optimization, and model adaptation. We review these approaches and numerical approximation methods, then discuss the additional requirements for hiding batch occupancy and sequence lengths.

A.1 NON-INTERACTIVE FHE TRANSFORMER INFERENCE

NEXUS (Zhang et al., 2025a) develops FHE primitives for ciphertext compression, slot folding, and encrypted output selection to support non-interactive Transformer inference. THOR (Moon et al., 2025) combines diagonal-major encoding and compact packing with matrix multiplication algorithms and nonlinear approximations.

MOAI (Zhang et al., 2026b) connects QK^{T}, Softmax, and CV through compatible column and diagonal layouts, avoiding explicit repacking along the attention path. Its Softmax and layer normalization algorithms reduce rotations, with further savings from scalar fusion and bootstrapping placement. ARION (Yang et al., 2025) introduces Double-BSGS ciphertext multiplication and fuses Softmax substeps with surrounding linear operations. MYGO builds on these compatible attention layouts and multiplication algorithms.

Packing exposes parallelism across matrices, attention heads, and independent inputs. Jiang et al. (2018) develop matrix encodings that allow multiple matrices to share a ciphertext for batched homomorphic computation. MOAI uses interleaved batching to evaluate independent Transformer inputs in parallel. Euston (Gao et al., 2026) combines SVD-based preprocessing with batched homomorphic matrix multiplication to reduce client preprocessing, ciphertext storage, and rotations, and coordinates packing with nonlinear evaluation depth. STIP (Wang et al., 2026) uses real and imaginary components for parallel matrix outputs or attention heads, while Adaptive Multi-Column Packing places multiple heads within a ciphertext. Head packing reduces the ciphertext count within an attention computation; input batching amortizes computation across independent inputs.

A.2 LOW-DEPTH MODELS AND NUMERICAL APPROXIMATION

Powerformer (Park et al., 2025) uses distillation to replace Softmax and LN with Batch Power-Max and Batch LN. Their inference-time normalization uses precomputed constants, reducing input-dependent division. Rho et al. (2025) replace Softmax attention with Gaussian kernels and use LoRA to reduce ciphertext–ciphertext computation in private personalized fine-tuning and subsequent inference. STIP (Wang et al., 2026) also uses Gaussian-kernel attention and an inverse-free reformulation of LN. ULD-Net (Xie et al., 2026b) trains fully polynomial networks from scratch, combining polynomial normalization and activations with linear attention. Its normalization and training procedures control activation growth through deep polynomial compositions.

ATLAS (Xie et al., 2026a) instead searches nonlinear approximation configurations without retraining. Its two-stage search and surrogate models balance multiplicative depth against model-output error across layers, while keeping encryption parameters and packing fixed. MYGO jointly distills the FFN's two affine layers and cubic activation while retaining Softmax and LN as approximation targets. Its numerical compilation also constrains module interfaces by the compact cryptographic budget and the levels, scales, and input domains required by subsequent operations.

A.3 REQUEST-SHAPE PRIVACY

Fixed packing does not conceal request shape if the evaluator still receives plaintext lengths or validity masks. MOAI uses a fixed interleaving stride, but its released CPU and GPU implementations construct plaintext validity operands from input lengths and pass them to padding-sensitive operators (Zhang et al., 2025b; 2026a). These operands expose within-container occupancy and per-input sequence lengths. Validity handling cannot simply be omitted: padded coordinates can contribute nonzero values after nonlinear operations, and fully invalid rows can produce unsafe reciprocal in-

puts. MYGO combines fixed-capacity interleaving with encrypted validity metadata to preserve the numerical role of masking while hiding occupancy and lengths under a fixed evaluation graph.

B PRELIMINARIES

B.1 NOTATION GUIDE

Table 3 summarizes the notation used in this paper. The layer index ℓ is distinct from the ciphertext level j, and the representation factor σ_Z from the CKKS scale Δ_{ct}. Packing tables use $B = B_{\max}$ and $M = M_{\max}$; proof-local symbols are defined where used.

Table 3: Notation index.

Symbol	Meaning	Symbol	Meaning
A. Cryptographic configuration (Section 3)			
N	Ring dimension (2^{15}).	$N/2$	Slot capacity (16,384).
j, Lev j	Logical ciphertext level.	ℓ_{post}	Post-Boot level (4).
Q_j	Ciphertext modulus at level j.	P	Auxiliary key-switching modulus.
q_j	Logical factor Q_j/Q_{j-1} ($j \geq 1$).	L_j	Physical RNS-prime count at j.
Δ_{ct}	Nominal CKKS scale (2^{31}).	δ	Arithmetic scale (RNS appendix).
B. Packing and encrypted validity (Section 4)			
$B_{\max}$	Input capacity; fixed stride.	$M_{\max}$	Token capacity per input.
b	Input-lane index.	r, a	Token-row and feature indices.
m_b	Private length (0: inactive).	$q_b(r)$	Validity bit $\mathbf{1}[r < m_b]$.
$M^C_{b,\delta}(r)$	Pair-validity mask at offset δ.	ct^{row}_M	Encrypted row-validity mask.
δ	Attention-diagonal offset.	$\mathrm{Rot}_{tB_{\max}}$	Rotation for a t-token shift.
M^{CLS}	Public row-0 selector per lane.	Π	Fixed ciphertext/message topology.
C. Transformer and attention quantities (Section 5)			
ℓ, $\mathcal{L}$	Layer index and layer count.	h, d_h	Attention-head index and width.
d	Hidden-state width.	Q, K, V	Query, key, and value projections.
S_{ij}	Scaled query–key score.	U_{ij}	Shifted score $S_{ij} - g_{\ell h,i}$.
$g_{\ell h,i}$	Encrypted row-common shift.	μ_i, t_i	Masked mean and scaled variance.
ρ_m, ω_m, $\bar{\ell}_m$, $\bar{c}_m$	Four encrypted length features.	$A^{\ell h}$, C	Target and packed weights.
D_i, D_i^{safe}	Denominator and safe-filled value.	$H^{\ell h}$	Normalized attention-head output.
D. Numerical compilation (Section 5)			
σ_Z	Public representation factor for Z.	$ct[Z;\sigma_Z]$	Decoded value $\simeq \sigma_Z Z$.
$\sigma_{0,\ell}$	Pre-LN1 residual factor.	$\sigma_{1,\ell}$	LN1 normalized-core factor.
$\sigma_{2,\ell}$	Pre-LN2 residual factor.	$\sigma_{3,\ell}$	Post-LN2 output factor.
$\mathcal{I}_s$	Site-s approximation interval.	$\mathcal{D}_{\mathrm{Boot},s}$	Site-s Boot input domain.
p_ℓ	Learned FFN cubic polynomial.	c_ℓ, s^{in}_ℓ	FFN input center and scale.
$\mathcal{R}^{\mathrm{out}}_\ell$	Output restoration folded into FC2.	t_{pad}	Safe exponential padding value.
c^E_ℓ, $\beta_{\ell h}$	Output/inner-polynomial gains.	$\tau_{\ell h}$, $\bar{\tau}_{\ell h}$	Pseudocounts: repr./source units.
d_{fill}	Safe fill for invalid denominators.	ϵ_{LN}	LN variance stabilizer.
E. Protocol and evidence (Section 2; Appendix H.3)			
$\mathcal{C}$	Closed deployment contract.	$\mathcal{G}$	Compiled server evaluation graph.
Θ_{FHE}, Θ_{num}, Θ_{exec}	Three contract interfaces.	$\mathcal{V}$, $\mathcal{K}$	Validity semantics; eval.-key set.
$\mathcal{R}$, $\mathbf{R}$	One container; request sequence.	n_{ctr}, $\boldsymbol{\tau}$	Container count; submission times.
$\mathcal{L}_{\mathcal{C}}$	Permitted leakage.	$\approx_c$	Computational indistinguishability.
n_{valid}	Valid-coordinate count.	r_j	Outside-domain % at boundary j.

Reading conventions. Boot denotes bootstrapping; a Rescale consumes one logical level. MAE averages absolute errors. P99 denotes an absolute-error percentile in the fidelity plots, but an absolute-

input magnitude percentile in the domain ablations. The context also distinguishes head index h from secret Hamming weight h, and attention projection Q from ciphertext modulus Q_j.

B.2 Transformer Model Configurations

BERT encoder. BERT-base is a post-LN Transformer encoder with 12 layers, hidden dimension 768, and 12 Attention heads (Devlin et al., 2019). Each layer contains multi-head attention (MHA) and a feed-forward network (FFN), with residual addition followed by layer normalization (LN) after each sublayer. Its classification graph applies a pooler and classifier to the final `[CLS]` representation. Appendix B.3 gives the encoder and polynomial-FFN equations; Appendix F.1 details training, and Appendix C defines the protocol.

LLaMA decoder. LLaMA-3-8B is a 32-layer decoder with causal grouped-query Attention (GQA), rotary positional embeddings (RoPE), root mean square layer normalization (RMSNorm), and a SwiGLU FFN. Each decoder block applies RMSNorm before its Attention and FFN sublayers, with residual addition after each sublayer. RoPE acts on the query and key vectors, and the SwiGLU FFN comprises gate, up, and down projections.

The evaluated configuration has hidden dimension 4,096, 32 query heads, eight key/value heads, head dimension 128, SwiGLU intermediate dimension 14,336, and vocabulary size 128,256. Appendices I.4 and I.5 describe the decoder circuits and the separate vocabulary-head experiment.

B.3 BERT-base Encoder Computation

The reference BERT-base encoder contains multi-head self-attention and an FFN, each followed by residual addition and layer normalization. The distilled student preserves this topology and uses a layer-specific cubic FFN activation. Layer indices are omitted from hidden states and affine parameters when unambiguous.

For a hidden state $X \in \mathbb{R}^{m\times 768}$, the query, key, and value projections for Attention head $h \in \{0,\dots,11\}$ are

$$\begin{aligned} Q_h &= XW_h^Q + b_h^Q, \\ K_h &= XW_h^K + b_h^K, \\ V_h &= XW_h^V + b_h^V, \end{aligned} \tag{7}$$

where $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{768\times 64}$, the biases have dimension 64 and are broadcast over token rows, and $Q_h, K_h, V_h \in \mathbb{R}^{m\times 64}$.

Let $M^{\mathrm{add}} \in \{0,-\infty\}^{m\times m}$ denote the additive attention mask, with $M_{ij}^{\mathrm{add}} = 0$ for permitted query–key pairs and $M_{ij}^{\mathrm{add}} = -\infty$ for excluded pairs. For rows with at least one permitted key, the head output is

$$C_h = \mathrm{softmax}\left(\frac{Q_h K_h^{\mathsf{T}}}{\sqrt{64}} + M^{\mathrm{add}}\right) V_h, \qquad C_h \in \mathbb{R}^{m\times 64}, \tag{8}$$

where Softmax is applied rowwise. The mask specifies the reference Attention semantics; its encrypted realization uses the validity metadata and safe-fill operations described in the main text.

Concatenating the head outputs along the feature dimension gives the combined Attention output

$$A = \mathrm{Concat}(C_0,\dots,C_{11})W^O + b^O, \qquad A \in \mathbb{R}^{m\times 768}, \tag{9}$$

where $W^O \in \mathbb{R}^{768\times 768}$ and b^O has dimension 768. Residual addition and the first LN then produce

$$Y = \mathrm{LN}_1(X + A). \tag{10}$$

For the reference encoder, the FFN hidden state and layer output are

$$\begin{aligned} H &= \mathrm{GELU}(YW_1 + b_1), \\ Z &= \mathrm{LN}_2(Y + HW_2 + b_2), \end{aligned} \tag{11}$$

where $W_1 \in \mathbb{R}^{768\times 3072}$, $W_2 \in \mathbb{R}^{3072\times 768}$, and b_1 and b_2 have dimensions 3,072 and 768, respectively. Thus, $H \in \mathbb{R}^{m\times 3072}$ and $Y, Z \in \mathbb{R}^{m\times 768}$. Both LN operators normalize each token row over its hidden features.

At layer ℓ, the deployed student uses the same normalized cubic path as the compiled FFN:

$$\begin{aligned} z_\ell &= (YW_1 + b_1 - c_\ell)/s_\ell^{\text{in}}, \\ H &= \mathcal{R}_\ell^{\text{out}}(p_\ell(z_\ell)), \qquad p_\ell(z) = \sum_{k=0}^{3} a_{\ell,k} z^k. \end{aligned} \tag{12}$$

Here c_ℓ and s_ℓ^{in} specify input centering and scaling, and $\mathcal{R}_\ell^{\text{out}}$ denotes the affine output restoration. The compiler absorbs these maps into FC1 and FC2, respectively, as in Section 5.3; they require no separate online stages. The student then applies FC2, residual addition, and the second LN as in Equation (11). Both model evaluation and its encrypted realization use this cubic path.

B.4 RNS-CKKS Overview

RNS-CKKS supports approximate SIMD arithmetic over real or complex values (Cheon et al., 2017; 2019). At *logical level* j, it operates over $R_{Q_j} = \mathbb{Z}_{Q_j}[X]/(X^N + 1)$ and encodes up to $N/2$ values in one ciphertext. The ciphertext modulus is $Q_j = \prod_{i=0}^{j} q_i$, where each logical factor q_i is implemented by one or more machine-word RNS primes. Thus, j indexes logical levels rather than individual *physical primes*. For a slot vector $\mathbf{u}$, encryption and decryption satisfy $ct \leftarrow \text{Enc}_{pk}(\mathbf{u})$ and $\text{Dec}_{sk}(ct) = \mathbf{u} + \mathbf{e}$, where $\mathbf{e}$ includes encoding and cryptographic approximation error.

Let ct and ct' encrypt slot vectors $\mathbf{u}$ and $\mathbf{u}'$, and let $\mathbf{v}$ be plaintext. The four elementary operations satisfy the following decoded-value relations:

$$\begin{aligned} \text{Dec}_{sk}(\text{Add}(ct, ct')) &\simeq \mathbf{u} + \mathbf{u}', \quad \text{Dec}_{sk}(\text{Mult}(ct, ct')) \simeq \mathbf{u} \odot \mathbf{u}', \\ \text{Dec}_{sk}(\text{Mult}(ct, \mathbf{v})) &\simeq \mathbf{u} \odot \mathbf{v}, \qquad \text{Dec}_{sk}(\text{Rot}_r(ct)) \simeq \rho_r(\mathbf{u}), \end{aligned}$$

where $\odot$ denotes slotwise multiplication, ρ_r denotes the slot permutation induced by rotation, and $\simeq$ allows accumulated CKKS approximation error. Complex conjugation satisfies the analogous relation with $\overline{\mathbf{u}}$.

Binary ciphertext operands are aligned to a common modulus basis; addition also requires compatible arithmetic scales. Plaintext operands are encoded over the current ciphertext basis with an operation-compatible scale. Under our logical-level convention, multiplication followed by one Rescale consumes one level, even when the removed logical factor contains several physical primes. Once operands are aligned, addition, rotation, and conjugation preserve the level.

Bootstrapping refreshes an admissible ciphertext to a configured output level while introducing additional approximation error. Its registered input domain and input/output level and scale requirements are therefore part of the deployment contract. Appendix D.1 specifies the scale, level, key, input-domain, and memory details for MYGO.

C Protocol and Security Analysis

C.1 Client–Server Roles and Permitted Leakage

Roles and public setup. The trusted client generates and retains all secret keys, uploads the public and evaluation-key bundle, encrypts request containers, and locally decrypts the returned ciphertexts. The honest-but-curious CSP holds the compiled model and follows the prescribed evaluation algorithm. It receives neither decrypted results nor feedback derived from their decryption. For security parameter λ, let

$$(\mathsf{sk}, \mathsf{keys}_{\text{kd}}) \leftarrow \mathsf{KeyGen}(1^\lambda, \mathcal{C}), \qquad \mathsf{pub} = (\mathcal{M}, \mathsf{keys}_{\text{kd}}, \mathcal{C}).$$

Here sk includes the inference and auxiliary Boot secrets. The model and contract are public; key generation does not depend on the hidden challenge bit. Requests may be chosen after the public setup is known.

Requests and leakage. A container is the ordered tuple

$$\mathcal{R}^{(t)} = \big((X_b^{(t)}, m_b^{(t)})\big)_{b=0}^{B_{\max}-1}, \qquad 0 \le m_b^{(t)} \le M_{\max}, \tag{13}$$

where $X_b^{(t)} \in \mathbb{R}^{M_{\max} \times d}$ is represented at the finite precision prescribed by $\mathcal{C}$, and $m_b^{(t)} = 0$ denotes an inactive lane. A supported sequence is polynomial in λ and has a well-defined encoding under the contract. As in Section 2, the permitted leakage is

$$\mathcal{L}_{\mathcal{C}}(\mathbf{R}; \mathsf{pub}) = (\mathcal{M}, \mathsf{keys}_{\mathrm{kd}}, \mathcal{C}, n_{\mathrm{ctr}}, \boldsymbol{\tau}). \tag{14}$$

The quantities $B_{\max}, M_{\max}, \Pi$, and $\mathcal{G}$ are determined by the public contract and model. The topology Π specifies message order, ciphertext counts and serialized lengths, key-domain assignments, and all public ciphertext metadata, including levels and scales, for uploads and responses. Identifiers may label public objects but do not replace those objects in the leakage. Container count, submission times, and key reuse are observable. Occupancy, validity boundaries, and the individual $m_b^{(t)}$ are not additional plaintext fields of the view.

Fixed transcript structure. For fixed $\mathsf{pub}, n_{\mathrm{ctr}}, \boldsymbol{\tau}$, client preprocessing produces a typed list of plaintext polynomials

$$(\mu_1, \ldots, \mu_T) = \mathsf{Pack}_{\mathcal{C}}(\mathbf{R}).$$

Each container repeats the same public upload template fixed by $\mathcal{C}$ before request selection. Hence the domain $d(j)$, level, modulus, and public type at global position j are predetermined; choosing n_{ctr} only determines how many copies of this template are used. Features and all request-dependent lengths, factors, and masks occur only in these encrypted values. Inactive lanes also use fresh randomized encryptions. Let ρ_{S} be the server's internal random tape, sampled independently of the private request; it is empty for deterministic evaluation. With $\mathbf{c} = (c_1, \ldots, c_T)$ denoting the uploads, the modeled CSP view is

$$\mathsf{View}(\mathbf{R}) = \big(\mathsf{pub}, n_{\mathrm{ctr}}, \boldsymbol{\tau}, \rho_{\mathrm{S}}, \mathbf{c}, \Phi_{\mathcal{M},\mathcal{C}}(\mathsf{keys}_{\mathrm{kd}}, \mathbf{c}; \rho_{\mathrm{S}})\big). \tag{15}$$

The polynomial-time map Φ records the prescribed server computation, including intermediate and output ciphertexts. Its only request-dependent input is $\mathbf{c}$. There are no additional plaintext validity signals, decryption-dependent messages, or request-dependent protocol branches.

C.2 Request-Shape Privacy

Left/right experiment. For $b \in \{0, 1\}$, experiment $\mathsf{Shape}_b^{\mathcal{A}}(1^\lambda)$ samples an honest public setup and gives it to $\mathcal{A}_1$. The adversary returns two request sequences $\mathbf{R}_0, \mathbf{R}_1$, a common submission schedule, and state z. The experiment returns 0 if either sequence is unsupported or their permitted leakage differs. Otherwise, it honestly executes $\mathbf{R}_b$ and returns $\mathcal{A}_2(z, \mathsf{View}(\mathbf{R}_b))$. Define

$$\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A}) = \Big|\Pr[\mathsf{Shape}_0^{\mathcal{A}} = 1] - \Pr[\mathsf{Shape}_1^{\mathcal{A}} = 1]\Big|. \tag{16}$$

All probabilities include setup, encryption, server, and adversary randomness. Both experiments use the same honest setup distribution; security is averaged over setup rather than asserted separately for every fixed public key. The complete request sequences are selected after setup and before any challenge uploads are observed.

Encryption security with public evaluation keys. Let HE denote the deployed encryption scheme. The CPA-EK experiment gives an adversary the entire honestly generated public and evaluation-key bundle $\mathsf{keys}_{\mathrm{kd}}$, including all cross-domain switching keys. After receiving this bundle, the adversary selects a registered client encryption domain, a supported public ciphertext type, and two admissible plaintexts of that type. The type fixes the encryption modulus, level, and public format. The challenger returns a fresh encryption of one selected plaintext under a hidden bit b. Fresh encryptions of other chosen plaintexts may be generated publicly or obtained from the registered encryption interfaces; no decryption oracle is available. Write

$$\mathrm{Adv}^{\mathrm{CPA\text{-}EK}}_{\mathsf{HE}}(\mathcal{B}) = \Big|\Pr[\mathsf{CPA\text{-}EK}_0^{\mathcal{B}} = 1] - \Pr[\mathsf{CPA\text{-}EK}_1^{\mathcal{B}} = 1]\Big|.$$

This is the IND-CPA requirement for the encryption interface with its complete public auxiliary keys.

Theorem C.1 (Fixed-topology request privacy)**.** *Assume the fixed transcript structure above, fresh independent encryption randomness, and CPA-EK security of the deployed RNS-CKKS scheme. Then* Mygo *satisfies Definition 2.2 against an honest-but-curious CSP. For any PPT adversary*

$\mathcal{A}$ and positive polynomial upper bound $T = T(\lambda)$ on the total number of uploaded ciphertexts, there exists a PPT adversary $\mathcal{B}$ such that

$$\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A}) \le T \ \mathrm{Adv}^{\text{CPA-EK}}_{\mathsf{HE}}(\mathcal{B}). \tag{17}$$

Proof. Fix λ and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. Every game returns 0 on an unsupported or unequal-leakage pair. Each game samples the server random tape ρ_{S} according to the prescribed evaluation algorithm, independently of the private request and the client encryption coins.

Step 1: Define the transcript hybrids. For an accepted request pair, let

$$\eta = \mathcal{L}_{\mathcal{C}}(\mathbf{R}_0; \mathsf{pub}) = \mathcal{L}_{\mathcal{C}}(\mathbf{R}_1; \mathsf{pub}), \qquad \boldsymbol{\mu}^{(b)} = \mathsf{Pack}_{\mathcal{C}}(\mathbf{R}_b).$$

Let $t \le T$ be the common actual upload count. The fixed upload template specifies each position's domain, modulus, and public type before request selection. Write $\mathsf{E}_k(\mu)$ for fresh encryption at position k, and $\mathsf{V}_\eta(\mathbf{c}; \rho_{\mathrm{S}})$ for the complete view in Equation (15).

For $j \in \{0, \dots, T\}$, game H_j runs honest setup and $\mathcal{A}_1$, then independently generates

$$c_k^{(j)} \leftarrow \begin{cases} \mathsf{E}_k(\mu_k^{(1)}), & k \le j, \\ \mathsf{E}_k(\mu_k^{(0)}), & k > j, \end{cases} \qquad 1 \le k \le t.$$

It returns $\mathcal{A}_2(z, \mathsf{V}_\eta(\mathbf{c}^{(j)}; \rho_{\mathrm{S}}))$. Mixed hybrids need not encode a valid request; the fixed public evaluation remains well-defined on their ciphertexts. With $h_j = \Pr[H_j = 1]$, the endpoints satisfy

$$h_0 = \Pr[\mathsf{Shape}_0^{\mathcal{A}} = 1], \qquad h_T = \Pr[\mathsf{Shape}_1^{\mathcal{A}} = 1].$$

Step 2: Embed one encryption challenge. The reduction $\mathcal{B}$ receives the complete key bundle from its CPA-EK challenger, independently samples J uniformly from $\{1, \dots, T\}$, and runs $\mathcal{A}_1$. On an unsupported or unequal-leakage pair, it ignores a dummy challenge on an admissible equal-message pair and returns 0. If $J \le t$, it submits $(\mu_J^{(0)}, \mu_J^{(1)})$ with the domain and public type fixed for position J. Given the challenge ciphertext $c^\star$, it forms

$$\widehat{c}_k \leftarrow \begin{cases} \mathsf{E}_k(\mu_k^{(1)}), & k < J, \\ c^\star, & k = J, \\ \mathsf{E}_k(\mu_k^{(0)}), & k > J, \end{cases} \qquad 1 \le k \le t,$$

and returns $\mathcal{A}_2(z, \mathsf{V}_\eta(\widehat{\mathbf{c}}; \rho_{\mathrm{S}}))$. Nonchallenge encryptions are computed publicly or obtained from the allowed encryption interfaces. If $J > t$, it ignores a dummy challenge and generates all uploads from $\boldsymbol{\mu}^{(1)}$; the two adjacent hybrids then coincide. The reduction needs no secret keys: intermediate and output ciphertexts are computed by the public server evaluation.

Step 3: Bound the distinguishing advantage. Conditioned on $J = j$, challenge bits 0 and 1 simulate H_{j-1} and H_j, respectively. Therefore,

$$\Pr[\mathsf{CPA\text{-}EK}_0^{\mathcal{B}} = 1] = \frac{1}{T} \sum_{j=1}^{T} h_{j-1},$$

$$\Pr[\mathsf{CPA\text{-}EK}_1^{\mathcal{B}} = 1] = \frac{1}{T} \sum_{j=1}^{T} h_j.$$

Subtracting and taking absolute values gives

$$\mathrm{Adv}^{\text{CPA-EK}}_{\mathsf{HE}}(\mathcal{B}) = \frac{|h_0 - h_T|}{T} = \frac{\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A})}{T}.$$

The single reduction $\mathcal{B}$ runs in polynomial time, including simulation of the server graph. CPA-EK security and polynomial T therefore make the request-shape advantage negligible. □

C.3 Cryptographic Assumptions and RLWE Reduction

Theorem C.1 is a protocol reduction and applies to the actual encryption interface. Concrete parameter accounting appears in Appendix D.2. Here we state the Ring-LWE distributions and the additional assumption needed for the complete related-secret evaluation-key bundle. Under the explicit public-key instantiation conditions in Lemma C.2, the following conditional bound holds:

$$\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A}) \le 2\epsilon_{\mathrm{key}} + 2T\epsilon_{\mathrm{enc}}. \tag{18}$$

Here ϵ_{enc} bounds decisional Ring-LWE advantage, and ϵ_{key} bounds the pseudorandomness advantage for the complete bundle of circular and related-secret keys. This requires an additional assumption; it is not asserted to follow from ordinary Ring-LWE.

Decisional Ring-LWE. Let $R = \mathbb{Z}[X]/(X^N + 1)$ and $R_q = R/qR$. For a specified secret distribution χ_s, error distribution χ_e, and sample count r, define

$$\mathsf{RLWE}^{(r)}_{q,\chi_s,\chi_e} = \big((a_i, a_i s + e_i)\big)_{i=1}^{r}, \qquad s \leftarrow \chi_s, \quad a_i \leftarrow R_q, \quad e_i \leftarrow \chi_e, \tag{19}$$

$$\mathsf{U}^{(r)}_{q} = \big((a_i, u_i)\big)_{i=1}^{r}, \qquad a_i, u_i \leftarrow R_q. \tag{20}$$

Following the distribution-parameterized formulation in Cheon et al. (2017), the decisional Ring-LWE advantage is the absolute difference between a distinguisher's output probabilities on these two distributions. All uniform samples and errors are independently drawn, except for the shared secret s. The assumption is made for the actual modulus and sampling distributions, including fixed-weight secrets where used; it is not a claim that every sparse-secret parameter choice follows from a generic worst-case lattice reduction.

For the conditional instantiation below, consider the coefficient-form public-key encryption algorithm of CKKS (Cheon et al., 2017):

$$\mathsf{pk} = (\beta, a) = (-as + e, a), \qquad \mathsf{Enc}_{\mathsf{pk}}(\mu) = (v\beta + e_0 + \mu,\ va + e_1) \pmod q, \tag{21}$$

where $v \leftarrow \chi_{\mathrm{enc}}$ and $e_0, e_1 \leftarrow \chi_e$ are sampled independently for each encryption, and v is independent of key generation. The polynomial μ is the finite-precision encoded message, and q is the modulus of this fresh encryption. The ephemeral distribution χ_{enc} must be specified separately from the inference and sparse Boot secret distributions. At a fixed modulus, RNS and NTT are public invertible representations of ring elements. Additional modulus switching or rounding is not automatically covered by this representation equivalence.

Evaluation-key distributions. Fix a public contract $\mathcal{C} = \mathcal{C}(\lambda)$. Define

$$\mathsf{K}^{\mathrm{real}}_{\mathcal{C}} \leftarrow \mathsf{KeyGen}^{\mathrm{pub}}_{\mathcal{C}}(1^\lambda), \qquad \mathsf{K}^{\mathrm{rnd}}_{\mathcal{C}} \leftarrow \mathsf{Rand}_{\mathcal{C}}(1^\lambda).$$

Here $\mathsf{KeyGen}^{\mathrm{pub}}_{\mathcal{C}}$ returns the complete native public and evaluation-key bundle, excluding secret keys. Its rows have the schematic form

$$(a_i,\ -a_i s_{\kappa(i)} + e_i + f_i(\mathbf{s})) \pmod{q_i}, \qquad i \in \mathcal{I}_{\mathcal{C}}, \tag{22}$$

where $\mathbf{s}$ collects the inference and auxiliary secrets, $s_{\kappa(i)}$ is the target secret for row i, and q_i is its native modulus, including auxiliary moduli when present. The functions f_i encode gadget factors, relinearization, Galois automorphisms, and both dense/sparse switching directions. All sampling and randomness-reuse distributions are those of native key generation.

The reference distribution is specified by a public PPT sampler $\mathsf{Rand}_{\mathcal{C}}$ that preserves public labels, component reuse, modulus projections, and serialization relations. Its public-key marginals and conditional sampling behavior must satisfy condition (ii) of the following assumption.

Assumption C.1 (Full-bundle pseudorandomness with key embedding)**.** The distributions above satisfy the following two conditions.

(i) Computational indistinguishability. For every PPT distinguisher $\mathcal{D}$, there exists a negligible function $\mathrm{negl}_{\mathcal{D}}$ such that

$$\mathrm{Adv}^{\mathrm{key}}_{\mathcal{C}}(\mathcal{D}; \lambda) := \big|\ \Pr[\mathcal{D}(\mathcal{C}, \mathsf{K}^{\mathrm{real}}_{\mathcal{C}}) = 1] - \Pr[\mathcal{D}(\mathcal{C}, \mathsf{K}^{\mathrm{rnd}}_{\mathcal{C}}) = 1]\big| \le \mathrm{negl}_{\mathcal{D}}(\lambda). \tag{23}$$

Let ϵ_{key} bound this advantage at the reductions' running times.

(ii) Exact reference-key embedding. For every supported client encryption domain d and modulus q, let $\mathsf{PK}_{d,q}(K)$ extract its public key from K. We require

$$\mathsf{Law}\big(\mathsf{PK}_{d,q}(\mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}})\big) = \mathsf{Unif}(R_q^2).$$

There exists an efficient conditional sampler $\mathsf{Extend}_{\mathcal{C},d,q}$ satisfying, for every $p \in R_q^2$,

$$\mathsf{Law}(\mathsf{Extend}_{\mathcal{C},d,q}(p)) = \mathsf{Law}\big(\mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}} \;\big|\; \mathsf{PK}_{d,q}(\mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}}) = p\big)\,.$$

The sampler runs in polynomial time uniformly over supported (d, q) and uses only p, public parameters, and fresh randomness; the conditional law preserves all public projections and reuse relations.

Condition (i) is an additional assumption on the complete related-secret key bundle, beyond ordinary Ring-LWE. Condition (ii) specifies the exact reference sampler required by Lemma C.2.

Lemma C.2 (Conditional Ring-LWE instantiation)**.** *Suppose client uploads use Equation* (21) *and Assumption C.1 holds. Let ϵ_{enc} bound decisional $\mathsf{RLWE}^{(2)}_{q,\chi_{\mathrm{enc}},\chi_e}$ distinguishing advantage uniformly over supported client encryption domains and levels, at the running times of the reductions below. For the protocol experiment of Appendix C.2 with at most T uploaded ciphertexts,*

$$\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A}) \le 2\epsilon_{\mathrm{key}} + 2T\epsilon_{\mathrm{enc}}.$$

The advantages are evaluated for the reduction algorithms' running times, including simulation of the public server evaluation.

Proof. Fix λ and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. Every game below rejects unsupported or unequal-leakage request pairs by returning 0. Let $T = T(\lambda)$ be a polynomial upper bound; $t \le T$ denotes the actual upload count in each experiment. Server randomness is sampled as in Appendix C.2.

Step 1: Replace the complete key bundle. For $b \in \{0, 1\}$, let $\mathsf{Exec}_b(K)$ run request selection, client encryption, the complete server computation, and $\mathcal{A}_2$, using public setup $(\mathcal{M}, K, \mathcal{C})$. Define the two key games

$$\begin{aligned} G_b^{\mathrm{real}}: &\quad K \leftarrow \mathsf{K}_{\mathcal{C}}^{\mathrm{real}}; \quad \text{return } \mathsf{Exec}_b(K), \\ G_b^{\mathrm{rnd}}: &\quad K \leftarrow \mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}}; \quad \text{return } \mathsf{Exec}_b(K). \end{aligned}$$

Set $p_b = \Pr[G_b^{\mathrm{real}} = 1]$ and $r_b = \Pr[G_b^{\mathrm{rnd}} = 1]$. The real game is exactly $\mathsf{Shape}_b^{\mathcal{A}}$. The PPT distinguisher $\mathcal{D}_b(K) = \mathsf{Exec}_b(K)$ therefore satisfies

$$|p_b - r_b| = \big|\Pr[\mathcal{D}_b(\mathsf{K}_{\mathcal{C}}^{\mathrm{real}}) = 1] - \Pr[\mathcal{D}_b(\mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}}) = 1]\big| \le \epsilon_{\mathrm{key}}.$$

This transition changes the entire bundle once in each request world. The reference keys need not decrypt correctly: all operations simulated here are public algorithms on ring elements.

Step 2: Define hybrids under the reference keys. For $j \in \{0, \ldots, T\}$, game H_j samples $K \leftarrow \mathsf{K}_{\mathcal{C}}^{\mathrm{rnd}}$, runs $\mathcal{A}_1$, and sets $\boldsymbol{\mu}^{(b)} = \mathsf{Pack}_{\mathcal{C}}(\mathbf{R}_b)$. At each position $1 \le k \le t$, independently generate

$$c_k^{(j)} \leftarrow \begin{cases} \mathsf{Enc}_{\mathsf{pk}_{d(k)}}(\mu_k^{(1)}), & k \le j, \\ \mathsf{Enc}_{\mathsf{pk}_{d(k)}}(\mu_k^{(0)}), & k > j, \end{cases}$$

using the fixed level and public type of that position. The game returns $\mathcal{A}_2$'s decision on the complete resulting view. With $h_j = \Pr[H_j = 1]$, we have

$$h_0 = r_0, \qquad h_T = r_1.$$

Step 3: Embed a two-sample Ring-LWE challenge. Fix a position j and its predetermined encryption modulus q. For $b \in \{0, 1\}$, construct $\mathcal{E}_{j,b}$ given

$$Z = ((\beta, Y_0), (a, Y_1)), \qquad Z \leftarrow \begin{cases} \mathsf{RLWE}^{(2)}_{q,\chi_{\mathrm{enc}},\chi_e}, \\ \mathsf{U}_q^{(2)}. \end{cases}$$

Using the conditional sampler assumed in the lemma, it generates

$$K \leftarrow \mathsf{Extend}_{\mathcal{C},j}(\beta, a), \qquad \mathsf{pk}_{d(j)} = (\beta, a) \text{ at modulus } q.$$

Here $\mathsf{Extend}_{\mathcal{C},j}$ abbreviates $\mathsf{Extend}_{\mathcal{C},d(j),q}$ from Assumption C.1. It samples the remaining reference bundle conditional on the embedded public key, preserving public projections and reuse relations. It uses only (β, a) and fresh coins, not (Y_0, Y_1). The reduction then runs $\mathcal{A}_1$ on this setup and inserts

$$c_j^\star = (Y_0 + \mu_j^{(b)},\ Y_1) \pmod q.$$

It encrypts positions $k < j$ from $\mathbf{R}_1$ and positions $k > j$ from $\mathbf{R}_0$, evaluates the public server graph, and returns $\mathcal{A}_2$'s decision. If $j > t$, it ignores (Y_0, Y_1) and simulates the common hybrid.

For a Ring-LWE challenge, the shared secret is the ephemeral encryption randomness $v \leftarrow \chi_{\mathrm{enc}}$, not a long-term inference or Boot secret, and

$$(Y_0, Y_1) = (v\beta + e_0,\ va + e_1).$$

Thus Equation (21) gives the exact identities

$$\Pr[\mathcal{E}_{j,0}(\mathsf{RLWE}^{(2)}_{q,\chi_{\mathrm{enc}},\chi_e}) = 1] = h_{j-1}, \Pr[\mathcal{E}_{j,1}(\mathsf{RLWE}^{(2)}_{q,\chi_{\mathrm{enc}},\chi_e}) = 1] = h_j.$$

For a uniform challenge, denote (Y_0, Y_1) by (U_0, U_1). On an accepted pair with $j \le t$, condition on the public setup, request-selection coins and outputs, and all nonchallenge upload coins; denote this information by $\mathcal{F}$. The selected messages are determined by $\mathcal{F}$, while (U_0, U_1) remains independent and uniform. Hence

$$\mathsf{Law}\big((U_0 + \mu_j^{(0)}, U_1) \mid \mathcal{F}\big) = \mathsf{Unif}(R_q^2), \mathsf{Law}\big((U_0 + \mu_j^{(1)}, U_1) \mid \mathcal{F}\big) = \mathsf{Unif}(R_q^2).$$

The two uniform simulations therefore have the same output distribution, including after public server evaluation. Let

$$u_j = \Pr[\mathcal{E}_{j,0}(\mathsf{U}_q^{(2)}) = 1] = \Pr[\mathcal{E}_{j,1}(\mathsf{U}_q^{(2)}) = 1].$$

The common rejection and unused-position rules preserve this equality. By the assumed Ring-LWE bound,

$$|h_{j-1} - u_j| \le \epsilon_{\mathrm{enc}}, \qquad |h_j - u_j| \le \epsilon_{\mathrm{enc}},$$

and consequently $|h_{j-1} - h_j| \le 2\epsilon_{\mathrm{enc}}$.

Step 4: Account for the total security loss. Combining the key transitions with the ciphertext hybrids yields

$$\begin{aligned}\mathrm{Adv}^{\mathrm{shape}}_{\mathsf{Mygo}}(\mathcal{A}) &= |p_0 - p_1| \\ &\le |p_0 - r_0| + \sum_{j=1}^{T} |h_{j-1} - h_j| + |r_1 - p_1| \\ &\le 2\epsilon_{\mathrm{key}} + 2T\epsilon_{\mathrm{enc}}.\end{aligned}$$

All distinguishers are PPT, with running times including the complete public evaluation. This proves the conditional bound. □

This lemma is conditional on the stated encryption form and reference-key assumptions. Applying it to a concrete implementation requires verifying the client encryption procedure and the reference samplers for its native key representation. The protocol theorem itself only requires CPA-EK security of the actual encryption interface. Concrete attack-cost estimates are separate from these asymptotic statements.

D Compact FHE Configuration Details

D.1 RNS-CKKS Instantiation and Key Domains

Ciphertext representation. Let $R = \mathbb{Z}[X]/(X^N + 1)$ and $R_{Q_j} = R/Q_j R$, where Q_j is the ciphertext modulus at logical level j. A standard two-component ciphertext is $\mathrm{ct} = (c_0, c_1) \in R_{Q_j}^2$,

together with its level j and arithmetic scale $\delta > 0$. For a slot vector $\mathbf{u} \in \mathbb{C}^{N/2}$, let $m_\delta(\mathbf{u}) \in R$ denote its CKKS encoding at scale δ. Under secret s, the decryption invariant is

$$c_0 + c_1 s \equiv m_\delta(\mathbf{u}) + e_{\mathrm{poly}} \pmod{Q_j}, \qquad \mathrm{Dec}_s(\mathrm{ct}) = \mathbf{u} + \mathbf{e}, \tag{24}$$

where e_{poly} is the polynomial error and $\mathbf{e}$ includes encoding and decryption error after decoding and scale removal. Correct decoding requires the encoded value and accumulated error to remain within the applicable modulus and numerical range. The nominal arithmetic scale is $\Delta_{\mathrm{ct}} = 2^{31}$; intermediate scales are tracked by the compiler. This scale is distinct from the public representation factor σ_Z in the main text: $\mathrm{ct}[Z; \sigma_Z]$ represents $\mathbf{u} = \sigma_Z Z$, whereas δ determines the polynomial encoding scale.

Key generation and key requirements. Given the configured secret and error distributions χ_s and χ_e and the compiled graph $\mathcal{G}$, client setup generates

$$(\mathrm{sk}, \mathrm{pk}, \mathcal{K}) \leftarrow \mathrm{KeyGen}(\Theta_{\mathrm{FHE}}, \chi_s, \chi_e, \mathcal{G}), \qquad \mathrm{sk} = s, \qquad \mathrm{KeyReq}(\mathcal{G}) \subseteq \mathcal{K}.$$

Here $\mathrm{sk} = s$ denotes the inference-domain secret; complete client setup also retains the auxiliary Boot secret specified in Appendix D.2. The evaluation-key set in the closed deployment contract is

$$\mathcal{K} = \mathcal{K}_{\mathrm{relin}} \cup \mathcal{K}_{\mathrm{Gal}} \cup \mathcal{K}_{\mathrm{Boot}},$$

where the three sets provide relinearization, Galois operations, and bootstrapping, respectively; shared keys are included only once. A switching key $\mathrm{ek}_{a \to b}$ encrypts the required decomposition-dependent multiples of the source secret a under the target secret b. Relinearization uses the source relation $s^2 \to s$, and a Galois operation uses $\varphi(s) \to s$ for its automorphism φ. Key switching uses an extended modulus basis containing an auxiliary modulus and returns a ciphertext over the ordinary ciphertext basis. In particular, P is not part of a resting ciphertext's level. Keys are generated once for each configured secret-key domain and reused across requests. Section 3.1 specifies the deployed secret distributions, and Appendix E describes the rotation-key reuse policy.

Operand alignment and multiplication. For ciphertext operands at levels j_1 and j_2, the common level is $j = \min\{j_1, j_2\}$. The higher-level operand is reduced to R_{Q_j} without changing its arithmetic scale, subject to the decryption-range condition. Addition also requires compatible scales; multiplication produces scale $\delta_1 \delta_2$. A plaintext operand is encoded over the current ciphertext basis: its scale matches the ciphertext for addition and is selected according to the compiled output scale for multiplication. For aligned ciphertexts $\mathrm{ct} = (c_0, c_1)$ and $\mathrm{ct}' = (c_0', c_1')$, their raw product is

$$(d_0, d_1, d_2) = (c_0 c_0',\ c_0 c_1' + c_1 c_0',\ c_1 c_1') \in R_{Q_j}^3, \tag{25}$$

with decryption relation $d_0 + d_1 s + d_2 s^2 = (c_0 + c_1 s)(c_0' + c_1' s)$ modulo Q_j. Relinearization converts this relation to two components, adding key-switching error while preserving the level and arithmetic scale.

Rescale and Galois operations. Let $q_j = Q_j / Q_{j-1}$ denote the exact logical factor removed at level j. A Rescale divides the ciphertext components by q_j with rounding and reduces them modulo Q_{j-1}, giving the state transition

$$(j, \delta) \xrightarrow{\mathrm{Rescale}} (j - 1, \delta / q_j). \tag{26}$$

The compiler selects operand scales and Rescale positions so that the resulting scale lies in the prescribed range. In particular, a product followed by one Rescale has scale $\delta_1 \delta_2 / q_j$; proximity to Δ_{ct} is a compilation condition, rather than an automatic property of multiplication. One logical factor q_j can contain several physical RNS primes, as formalized below.

For an odd integer a modulo $2N$, the map $\varphi_a : f(X) \mapsto f(X^a)$ is a Galois automorphism of R_{Q_j}. A slot rotation ρ_r is implemented by the corresponding automorphism followed by key switching:

$$\mathrm{Rot}_r(\mathrm{ct}) = \mathrm{KS}_{\varphi_a(s) \to s}\big(\varphi_a(c_0), \varphi_a(c_1)\big).$$

Complex conjugation uses $a = -1 \bmod 2N$. Both operations preserve (j, δ) and add key-switching error. They do not invoke Rescale and hence do not consume a logical level.

Bootstrapping interface and memory. Fresh ciphertexts enter the inference graph at level 6, whereas Boot returns to the common interface

$$\ell_{\mathrm{post}} = 4, \qquad \delta_{\mathrm{out}} \simeq \Delta_{\mathrm{ct}}.$$

For an admissible input ciphertext at a registered bootstrapping site, the decoded values satisfy

$$\mathrm{Dec}_s\big(\mathrm{Boot}(\mathrm{ct})\big) = \mathrm{Dec}_s(\mathrm{ct}) + \boldsymbol{\eta}_{\mathrm{B}}. \tag{27}$$

Admissibility includes the compiled input level, scale, coordinate map, and domain $\mathcal{D}_{\mathrm{Boot}}$. The modular-reduction approximation is calibrated on a bounded, usually symmetric interval; outside the registered domain, the intended approximation may fail. Equation (27) explicitly retains the incoming approximation error: Boot restores computational capacity and introduces an additional perturbation $\boldsymbol{\eta}_{\mathrm{B}}$.

If L_j physical primes represent Q_j and each residue occupies a w-bit word, the raw two-component ciphertext payload is

$$M_{\mathrm{ct}}(j) = \frac{2NL_jw}{8} \text{ bytes},$$

excluding metadata and allocation overhead. This identifies the linear dependence of ciphertext storage on N and the remaining physical basis. Persistent memory is dominated by the evaluation-key set $\mathcal{K}$ and public precomputation tables. Live ciphertexts and temporary operator buffers, including bootstrapping workspaces, determine the additional peak allocation.

D.2 Concrete Security Accounting

Secret distributions and exposed key bases. For a fixed weight h, define

$$\mathcal{S}_h = \left\{ \sum_{i=0}^{N-1} s_i X^i \in R : s_i \in \{-1, 0, 1\},\ \sum_{i=0}^{N-1} \mathbf{1}[s_i \neq 0] = h \right\},$$

and let χ_h denote the configured sampling distribution on $\mathcal{S}_h$. At $N = 2^{15}$, MYGO uses

$$s \leftarrow \chi_h, \quad h = N/2 = 16{,}384, \qquad \widetilde{s} \leftarrow \chi_{\widetilde{h}}, \quad \widetilde{h} = 32. \tag{28}$$

The inference secret s protects fresh ciphertexts and ordinary evaluation keys. The sparse secret $\widetilde{s}$ is used internally by sparse-secret encapsulation during Boot.

Let $\mathcal{B}(a)$ denote the largest modulus basis of the public or evaluation-key samples encrypted under target secret a. The two accounting regimes are

$$\begin{aligned} &\mathcal{B}(s) = Q_{16}P, && \log_2 \mathcal{B}(s) \simeq 880.9988 < 881, \\ &\mathcal{B}(\widetilde{s}) = qp, && \log_2 \mathcal{B}(\widetilde{s}) \simeq 96.9830. \end{aligned} \tag{29}$$

The dense-to-sparse key $\mathrm{ek}_{s\to\widetilde{s}}$ is generated only over Cheddar's short basis qp, before modulus raising, with

$$\log_2 q \simeq 40.9773, \qquad \log_2 p \simeq 56.0057, \qquad \log_2(qp) \simeq 96.9830.$$

After modulus raising, $\mathrm{ek}_{\widetilde{s}\to s}$ switches the ciphertext back to s using the full extended basis $Q_{16}P$. This return key encrypts source-secret material derived from $\widetilde{s}$ under the dense target secret s; it is therefore accounted for in $\mathcal{B}(s)$. The raised sparse-secret ciphertext is an intermediate value derived from the short-basis ciphertext, rather than an independently generated full-basis sparse-secret evaluation-key sample. Thus, sparse-secret encapsulation does not publish an 881-bit evaluation key encrypted under $\widetilde{s}$ (Bossuat et al., 2022).

Security target and assumptions. The dense-domain modulus $Q_{16}P$ is below the 881-bit reference limit for $N = 2^{15}$ and 128-bit classical security in the 2018 HomomorphicEncryption.org standard (Albrecht et al., 2018). This limit assumes i.i.d. uniform ternary coefficients and serves only as a reference for our fixed-weight secret. Concrete security estimates must account for the deployed secret and error distributions, sample count, and attack cost model. Ordinary dense-domain ciphertexts use $Q_j \mid Q_{16}$ without P, so their moduli remain below the full evaluation-key modulus $Q_{16}P$.

The archived fixed-weight modeling audit uses $N = 32768$, a balanced ternary secret of weight 16,384, and discrete Gaussian error with standard deviation 3.2. With lattice-estimator revision `53da598`, Table 4A reports the minimum estimated attack cost returned by each recorded estimator configuration. The $m = N$ and unbounded-sample runs agree at the reported precision. The full-QP result is 125.90 bits under the recorded MATZOV/default configuration; the 128-bit reference

target is therefore not a verified lower bound from this audit. The secret sampler and estimator distribution must also be matched when interpreting these model-dependent estimates.

For the sparse domain, the published encapsulation instance with $N = 2^{15}$, $\widetilde{h} = 32$, and $\log_2(qp) = 111$ remains above 128 bits under the sparse-secret attacks reported in Bossuat et al. (2022). MYGO matches its ring dimension and sparse-secret weight and uses a smaller short-basis modulus. This comparison supports the sparse-domain parameter choice under the reported attack model; it does not expose the sparse secret under the dense-domain modulus. The archive does not contain a matching estimator run for the actual short basis qp. Its sparse-secret runs at the full Q and QP bases do not represent this deployed sparse target domain.

Concrete parameter assessment must cover both pairs

$$(\chi_h, Q_{16}P) \quad \text{and} \quad (\chi_{\widetilde{h}}, qp),$$

with their configured error distributions. The protocol theorem in Appendix C assumes IND-CPA security in the presence of the complete public and evaluation-key bundle. The conditional Ring-LWE instantiation in Appendix C.3 states a stronger, explicit full-bundle pseudorandomness assumption covering the related-secret material. The exact parameter file, prime products, fixed weights, and short-basis split are frozen with the experimental artifacts, enabling independent verification of the two regimes.

Client-randomness integration check. An independent adapter reuses Cheddar's client algorithms while routing random words through OpenSSL's `RAND_bytes`, without modifying the cryptographic kernels. Three fresh-process runs each generated the 160-key evaluation bundle, produced distinct ciphertexts for repeated encryption of one plaintext, completed a decryption round trip, and returned a bootstrapped ciphertext at level 4. These results establish compatibility of the exercised adapter with setup, encryption, and Boot; they do not constitute a security audit of the complete deployed client.

D.3 MODULUS ACCOUNTING AND RESTRICTED-DEPTH COMPARISON

Logical levels and physical RNS primes. To make the logical schedule in Section 3.1 explicit, let $\widehat{q}_1, \ldots, \widehat{q}_{28}$ denote the physical ciphertext primes and let $\widehat{p}_1, \ldots, \widehat{p}_4$ denote the physical key-switching auxiliary primes. Partition the ciphertext-prime indices into the configured nonempty groups $\mathcal{I}_0, \ldots, \mathcal{I}_{16}$. Then

$$q_j = \prod_{i \in \mathcal{I}_j} \widehat{q}_i, \qquad Q_j = \prod_{k=0}^{j} q_k, \qquad P = \prod_{i=1}^{4} \widehat{p}_i. \tag{30}$$

Thus, j counts logical levels, whereas $L_j = \sum_{k=0}^{j} |\mathcal{I}_k|$ counts physical ciphertext primes. In particular, $L_{16} = 28$, even though the chain has 17 logical factors including q_0. The logarithms of the exact prime products, rounded to four decimal places, are

$$\log_2 Q_{16} \simeq 769.0117, \qquad \log_2 P \simeq 111.9870, \qquad \log_2(Q_{16}P) \simeq 880.9988.$$

Each quantity is computed from its unrounded product; independently rounded logarithms need not sum to the last displayed digit. Table 4A reports the archived dense-secret attack-cost sensitivity discussed above. Table 4B summarizes the rounded logical modulus accounting.

Following Cheddar's convention, a nominal scale divisor Δ_j satisfies $q_j \simeq \Delta_j$, so that $Q_{j-1} = Q_j/q_j \simeq Q_j/\Delta_j$. A logical 55-bit divisor can therefore be implemented by several smaller machine-word primes in a 32-bit RNS backend while representing one approximately 55-bit Rescale (Choi et al., 2026). The ten logical factors above level 6 are reserved for internal bootstrapping and key switching. They are not additional inference levels available after Boot.

Restricted-depth feasibility. Let $\mathcal{G}_f$ be a compiled approximation graph between two Boot boundaries, and let $d_{\mathrm{lev}}(\mathcal{G}_f)$ denote the maximum cumulative logical-level consumption along any input–output path, including scheduled Rescales and explicit level alignment. The common post-bootstrap interface requires

$$d_{\mathrm{lev}}(\mathcal{G}_f) \le \ell_{\mathrm{post}} = 4, \tag{31}$$

together with the compiled scale, approximation-error, and input-domain checks. The size of the full extended basis does not relax this post-bootstrap constraint.

Table 4: Security and modulus accounting.

A. Attack-cost sensitivity

Exposed basis	$\log_2$ modulus	ADPS16/rough	MATZOV/default
Q	769.012	113.30	145.51
QP	880.999	92.86	125.90

B. Selected logical levels

Basis	Logical composition	Rounded bit length
Q_0	41	41
Q_4	$41 + 4 \times 31$	165
Q_6	$41 + 4 \times 31 + 2 \times 30$	225
Q_{16}	$225 + 8 \times 55 + 2 \times 52$	769
P	2×56	112
$Q_{16}P$	$769 + 112$	881

Note. Panel A reports $\min \log_2 \mathrm{rop}$ over the attacks returned by each recorded estimator configuration. In Panel B, P appears only in the extended key-switching basis.

ARION's released implementation (Yang et al., 2025; hangenba, 2026) uses an initial degree-255 Chebyshev approximation followed by two reciprocal-refinement iterations. Starting from an encrypted degree-one input, a polynomial circuit with binary ciphertext multiplications and multiplicative depth $d_\times$ has degree at most $2^{d_\times}$. Consequently, evaluation of a polynomial of degree 255 requires

$$d_\times \geq \lceil \log_2 255 \rceil = 8.$$

Under the multiplication-followed-by-Rescale convention used here, this already exceeds the four-level budget before any additional cost from the input map, refinement iterations, or numerator multiplication.

Refinement under bootstrap perturbation. Let $D > 0$ be a Softmax denominator, let y approximate D^{-1}, and define the reciprocal residual $r = 1 - Dy$. In exact arithmetic, the reciprocal refinement satisfies

$$y^+ = y(1 + r), \qquad r^+ = 1 - Dy^+ = r^2. \tag{32}$$

If an intermediate Boot changes y to $y_\mathrm{B} = y + \eta_\mathrm{B}$, then, holding D fixed to isolate this perturbation,

$$r_\mathrm{B} = r - D\eta_\mathrm{B}, \qquad |r_\mathrm{B}| \leq |r| + D\varepsilon_\mathrm{B} \quad \text{when } |\eta_\mathrm{B}| \leq \varepsilon_\mathrm{B}. \tag{33}$$

For a Boot immediately after the refinement, the corresponding bound is $|r_\mathrm{B}^+| \leq |r|^2 + D\varepsilon_\mathrm{B}$. Thus, retaining the predicted quadratic improvement in the error budget requires the bootstrap contribution to be small relative to $|r|^2$. When it is comparable or larger, it can dominate the refined residual; the actual outcome depends on the perturbation's sign and magnitude. These relations explain why inserting Boot into a deeper reciprocal graph requires joint depth and error accounting. MYGO instead synthesizes its approximation graphs directly for the common level-4 post-bootstrap interface.

E FIXED EXECUTION AND PRIVATE PACKING

Non-Interleaved Coordinate Map. Let $X \in \mathbb{R}^{m \times m}$, with row r and column c. Its kth cyclic diagonal is $d_k(X)$, whose rth entry is $X_{r,(r+k) \bmod m}$. The rotation convention is $\rho_t(v)[s] = v[(s + t) \bmod m]$. In rotated-diagonal packing, index k is decomposed as $k = iw + j$, where w is the BSGS block width and $0 \leq j < w$. Under these conventions, Table 5A maps every matrix entry to one ciphertext and one logical slot.

SIB Coordinate Lift. For compact table notation, let $B = B_{\max}$ and $M = M_{\max}$. SIB lifts a logical slot s to physical slot $Bs + b$, where b is the input-lane index; $B = N/(2M) = 128$ for the BERT configuration. Let $X^{(b)}$ denote the matrix in lane b, let $m_b \leq M = 128$ be its private length, and let $q_b(r) = \mathbf{1}[r < m_b]$. Table 5B applies the same lift to all four layouts and records the two mask coordinates required by Attention. Inactive lanes set $m_b = 0$, but retain the same ciphertext and slot indices.

Table 5: Fixed-capacity packing maps.

A. Non-interleaved maps

Layout	Ciphertext index	Logical slot s	Value in the slot
Row	r	c	$X_{r,c}$
Column	c	r	$X_{r,c}$
Cyclic diagonal	$k = (c - r) \bmod m$	r	$d_k(X)[r] = X_{r,c}$
Rotated diagonal	(i, j), $k = iw + j$	$(r + iw) \bmod m$	$\rho_{-iw}(d_{iw+j})[s] = X_{r,c}$

B. SIB maps and masks

Object	Ciphertext index	Physical slot	Stored value
Row-packed $X^{(b)}$	r	$Bc + b$	$X^{(b)}_{r,c}$
Column-packed $X^{(b)}$	c	$Br + b$	$X^{(b)}_{r,c}$
Diagonal $d_k(X^{(b)})$	$k = (c - r) \bmod M$	$Br + b$	$X^{(b)}_{r,c}$
Rotated diagonal	(i, j), $k = iw + j$	$B((r + iw) \bmod M) + b$	$X^{(b)}_{r,c}$
Row mask	one fixed ciphertext	$Br + b$	$q_b(r)$
Rotated full-C mask	(i, j), $k = iw + j$	$B((r + iw) \bmod M) + b$	$q_b(r)q_b((r + k) \bmod M)$

Note. In Panel B, the base logical slot s is the corresponding slot defined in Panel A before applying the fixed physical lift $s \mapsto Bs + b$.

The non-interleaved map specifies the Double-BSGS matrix representation; the SIB lift embeds independent inputs and encrypted validity without altering it.

Rotation-Key Reuse under Partial IB Occupancy A CKKS slot rotation combines a Galois automorphism with key switching. At a fixed configured token capacity M, MOAI's reported IB uses stride $B = N/(2M)$ and a corresponding fixed rotation-key set. Compacting a container to B' active lanes, however, maps a j-token shift to rotation by jB'. Supporting variable B' would require a larger key union, an occupancy-specific key subset, or additional automorphisms and key switches from a fixed generating set. SIB retains stride B, direct rotations, and cached keys at every occupancy. Reserving the full IB lane width provides the same structural invariance, but encrypted validity remains necessary for private, padding-safe execution.

Structural invariance does not imply metadata privacy. In MOAI's released central processing unit (CPU) and GPU implementations, a plaintext vector of per-input sequence lengths is expanded into `bias_vec`; the evaluator derives plaintext masks inside Attention, Softmax, and LN from `bias_vec` and `input_num` (Zhang et al., 2025b; 2026a). Its fixed stride avoids occupancy-specific rotation keys, but the plaintext interface reveals the valid-slot map, occupancy, and sequence lengths.

AMCP Scaling with the Batch Size STIP's AMCP (Wang et al., 2026) allocates the same slot budget to the batch and to packed heads/columns. Let S be the number of CKKS slots, t the batch size, m the sequence length, and H the number of Attention heads. Its admissible merging factor is $k = \max\{\kappa \in \text{Divisors}(H) : \kappa tm \le S\}$. For the reported BERT-base dimensions $S = 32768$, $m = 128$, and $H = 12$, the resulting factors are:

Table 6: STIP AMCP scaling with batch size.

Batch t	32	64	128	256
Merging factor k	6	4	2	1
Head groups H/k	2	3	6	12
Occupied slots ktm	24576	32768	32768	32768

Increasing the batch size from 32 to 256 raises the head-group count from two to twelve. This relation is specific to STIP's AMCP; Euston's column packing allocates slots differently.

F Cross-Module Numerical Compilation Details

F.1 Distillation Objective and Schedule

Models, data, and parameterization. We initialize from frozen task-fine-tuned teachers `textattack/bert-base-uncased-{SST-2,QNLI,RTE}`. The student retains 12 BERT-base layers, hidden width 768, and FFN width 3,072. Each task uses its teacher's tokenizer, retaining examples with at most 128 tokens including special tokens; longer examples are excluded, not truncated. Table 7 gives the resulting sets. Using the notation of Equation (12), the trained activation is

$$z_\ell = \frac{x - c_\ell}{s_\ell^{\text{in}}}, \qquad P_\ell(x) = s_\ell^{\text{out}} \sum_{k=1}^{3} a_{\ell,k} z_\ell^k + b_\ell^{\text{poly}}. \tag{34}$$

The three coefficients, center, two log-scales, and output bias are trainable: seven scalars per layer, 84 overall. For each log-scale parameter θ, $s = \exp(\text{clamp}(\theta, -4, 6))$; this does not clip activations. The constant term is represented by b_ℓ^{poly} ($a_{\ell,0} = 0$ in the training parameterization). These scales are distinct from the compiled representation factors $\sigma_{0:3,\ell}$. For SST-2/QNLI, initialization least-squares fits $\text{GELU}(4z)/4$ using z, z^2, z^3 at 8,193 uniformly spaced points in $[-1, 1]$, with $c_\ell = b_\ell^{\text{poly}} = 0$ and both scales set to 4.

Objectives. Let a_T, a_S denote teacher/student logits and $p_J^{(T)} = \text{softmax}(a_J/T)$ for $J \in \{T, S\}$. With temperature $T = 2$, define

$$\begin{aligned} \mathcal{L}_D &= \text{CE}(y, \text{softmax}(a_S)) + T^2 \,\text{KL}(p_T^{(T)} \| p_S^{(T)}), \\ \text{NMSE}(A, B) &= \frac{\sum_{i \in \mathcal{V}} (A_i - B_i)^2}{\max(\sum_{i \in \mathcal{V}} B_i^2, 10^{-8})}, \\ \mathcal{L}_{\text{base}} &= \mathcal{L}_D + 0.2\mathcal{L}_H + 0.01\mathcal{L}_F + \lambda_R \mathcal{L}_R + \lambda_{\text{tail}} \mathcal{L}_{\text{tail}} + 10^{-6} \mathcal{L}_{\text{coef}}. \end{aligned} \tag{35}$$

KL is teacher-to-student with batch-mean reduction. $\mathcal{V}$ contains all feature coordinates of valid tokens. $\mathcal{L}_H$ and $\mathcal{L}_F$ average layerwise NMSE over the 12 encoder outputs (excluding embeddings) and FC2 outputs, respectively; FC2 matching is before dropout, residual addition, and LN2. Teacher targets come from its own full-model forward pass. We combine task supervision with logit and intermediate matching, following the distillation approach of Park et al. (2025). With $[u]_+ = \max(u, 0)$ and $\mathcal{V}_\ell^{\pm} = \{i \in \mathcal{V}_\ell : \pm z_{\ell i} > 1\}$, define

$$\begin{aligned} \mathcal{L}_R &= \frac{1}{12} \sum_{\ell=0}^{11} \text{mean}_{i \in \mathcal{V}_\ell} [|z_{\ell i}| - 1]_+^2, \qquad \mathcal{L}_{\text{coef}} = \frac{1}{36} \sum_{\ell=0}^{11} \sum_{k=1}^{3} a_{\ell,k}^2, \\ \mathcal{L}_{\text{tail}} &= \frac{1}{24} \sum_{\ell=0}^{11} \left[\frac{\sum_{i \in \mathcal{V}_\ell^-} P_\ell(x_i)^2}{\max(|\mathcal{V}_\ell^-|, 1)} + \frac{\sum_{i \in \mathcal{V}_\ell^+} (P_\ell(x_i) - \text{sg}(x_i))^2}{\max(|\mathcal{V}_\ell^+|, 1)} \right]. \end{aligned} \tag{36}$$

Here sg stops gradients; tail masks also carry no gradient. The two tails are normalized separately and an empty tail contributes zero. Only the three power coefficients enter $\mathcal{L}_{\text{coef}}$.

SST-2/QNLI optimization and selection. We train FC1, all cubic parameters, FC2, LN2 affine parameters, and the classifier; embeddings, Attention, LN1, and pooler weights are frozen. The student uses training mode (including dropout), and the teacher uses evaluation mode. During S0/Q0 all layers use $(1 - \alpha(t))\,\text{GELU}(x) + \alpha(t) P_\ell(x)$, where normalized stage progress t gives $\alpha = 0$ through 10%, a half-cosine rise to 1 from 10% to 90%, and $\alpha = 1$ thereafter. Subsequent stages and every validation checkpoint use pure cubics. AdamW is reset at every stage, with weight decay 0.01 for all non-polynomial parameters (including biases/LN) and zero for polynomial parameters; each group is clipped to gradient norm 1. The schedule uses 10% warmup, rounded down to whole updates, then linear decay. Microbatch size 2, accumulation 8, and evaluation batch size 4 give a usual effective training batch of 16. SST-2 uses BF16 AMP. The seed is 20260721; QNLI uses the global torch generator, whereas SST-2 replay uses an explicitly seeded generator.

QNLI performs 6,523 updates per epoch: Q0 has 19,569 updates/1,956 warmup updates; Q1 and Q2 each have 13,046/1,304. Its selected Q2 epoch-1 model inherits the *two-epoch* learning-rate horizon; rerunning a one-epoch schedule is different. The selected parameter path thus contains

Table 7: Data and selected distillation paths. Correct counts are on the same length-filtered validation set. Stage learning rates are non-polynomial/polynomial; epoch selections refer to model parameters. SST-2 and QNLI share stages 0–1, then use the stated task-specific endpoints.

Task	Train	Validation	Teacher/student correct	Student accuracy
SST-2	67,349	872	806/803	92.09%
QNLI	104,366	5,438	4,978/4,945	90.93%
RTE	2,186	244	179/172	70.49%

Stage	Epochs	Learning rates	λ_R	λ_{tail}	Selected endpoint
S0/Q0: conversion	3	$2\times10^{-5}/5\times10^{-4}$	1	0	Epoch 3 $\rightarrow$ S1/Q1
S1/Q1: pure cubic	2	$10^{-5}/10^{-4}$	10	0	Epoch 2 $\rightarrow$ S2/Q2
S2: SST-2 tail	2	$10^{-5}/10^{-4}$	10	5×10^{-4}	Epoch 2 $\rightarrow$ S3
Q2: QNLI tail	2	$10^{-5}/10^{-4}$	10	5×10^{-4}	Epoch 1 reported
S3: SST-2 LN2/replay	1	$10^{-5}/10^{-4}$	10	5×10^{-4}	Final model reported

$3 + 2 + 1$ epochs, while the executed candidate run contains $3 + 2 + 2$. QNLI uses neither LN2 variance regularization nor sample replay.

SST-2 variance refinement and replay. S3 starts from S2 epoch 2. Let $r_{\ell t}$ be the actual pre-LN2 residual vector after FC2-branch dropout and residual addition. Define

$$v_{\ell t} = \text{mean}_d(r_{\ell td} - \text{mean}_{d'}\, r_{\ell td'})^2 + 10^{-12},$$

$$e_{\ell t} = [\log(v_{\ell t}/64)]_+^2 + [\log(0.5/v_{\ell t})]_+^2, \qquad \mathcal{L}_{\text{LN2}} = \frac{1}{12}\left(\sum_{\ell=0}^{11} C_\ell + C_9\right),$$

$$\mathcal{L}_{S3} = \mathcal{L}_{\text{base}} + 0.1 w(t) \mathcal{L}_{\text{LN2}}. \tag{37}$$

C_ℓ averages the largest $\max(1, \lceil 0.01n \rceil)$ values of $e_{\ell t}$ among the current microbatch's n valid tokens. $w(t)$ is zero through 10% of S3, rises linearly to 1 by 30%, then stays at 1. Layer indices start at zero; the additional Layer-9 term retains the denominator 12. The target $[0.5, 64]$ concerns variance, and the soft penalty does not guarantee coverage of this interval or a Boot domain.

The fixed replay manifest contains 4,096 distinct training examples and 4,288 positions: 46 severe examples each occur four additional times, plus eight neutral positions. All 54 replayed identities belong to the base 4,096. Severe candidates are correctly classified by the source model, have Layer-9 maximum variance above 512, and attain their cross-layer maximum in Layer 9. Exact membership/order follows the frozen manifest. S3 uses 268 updates, including 26 warmup updates. The selected SST-2 path is seven full-data epochs plus this single small-set pass; validation accuracy, loss, and range diagnostics informed branch selection.

Layerwise conversion and replay for RTE. A stratified 1,749/437 split of the training set is used to choose the per-layer schedule by internal-holdout correct count, cross-entropy, and fixed tie-breaking rules. After freezing Table 8, we restart from the original teacher and replay on all 2,186 training examples. At layer ℓ, the converted prefix is purely cubic and the unconverted suffix uses GELU. Current-student FC1 activations on valid training tokens initialize the cubic by streaming-moment least squares: fit $\text{GELU}(4z)/4 \simeq \sum_{k=0}^{3} q_k z^k$ with $z = x/4$, then set $a_{\ell,k} = q_k$ ($k = 1, 2, 3$), $b_\ell^{\text{poly}} = 4q_0$, $c_\ell = 0$, and both scales to 4.

Layers 5–11 first receive one epoch of local pre-fitting. With F_ℓ denoting FC1–activation–FC2 before residual/LN, and u_ℓ^S the current student FFN input,

$$\mathcal{L}_{\text{local}} = \text{NMSE}\big(F_\ell^S(\text{sg}(u_\ell^S)), F_\ell^T(\text{sg}(u_\ell^S))\big),$$

$$\mathcal{L}_{\text{RTE}} = \mathcal{L}_D + 0.2\mathcal{L}_H^{0:\ell} + 0.01\mathcal{L}_F^{0:\ell}. \tag{38}$$

Local pre-fitting updates only the current FC1/cubic/FC2, feeding the same detached student input to both FFNs. Main/recovery stages instead use $\mathcal{L}_{\text{RTE}}$, average matching only over layers $0, \ldots, \ell$, and obtain teacher targets from the teacher's own full forward pass. They train converted FC1/cubic/FC2 modules and the classifier, with both LNs, embeddings, Attention, and pooler frozen. Dropout is disabled. There is no added local, range, tail, coefficient, or variance loss in these joint stages.

Table 8: Frozen RTE replay: executed epochs per layer. Seeds are reference order seeds; the implementation further derives stage/epoch orders. The 41-epoch total excludes schedule exploration and activation initialization.

Layer	Local pre-fit	Main	Recovery	Reference seed
0	0	3	0	20260907
1	0	1	0	20260907
2	0	4	0	20260907
3	0	3	0	20260907
4	0	4	1	20260907
5	1	3	2	20260908
6	1	3	1	20260908
7	1	0	0	20260908
8	1	1	2	20260908
9	1	0	2	20260908
10	1	0	1	20260908
11	1	2	1	20260908
Total	7	24	10	41 stage epochs

If main training is scheduled, only its first epoch linearly blends the current GELU into its cubic; subsequent main and recovery epochs use pure cubics; validation also makes the current layer purely cubic. Zero-main stages receive no blend epoch.

RTE uses FP32 with TF32 disabled, microbatch size 8 and accumulation 2; the final accumulation window is normalized by its actual sample count. AdamW decays linear weights by 0.01 and excludes biases/polynomial parameters; FFN, polynomial, and classifier groups are each clipped to norm 1. Local/main learning rates are 2×10^{-5} for FC1/FC2, 10^{-4} for polynomial modules in layers 0–3, 10^{-5} for polynomial modules in layers 4–11, and 10^{-5} for the classifier (frozen locally). Recovery halves each rate. Each phase resets AdamW and uses 10% warmup, rounded *up*, then constant rates. Warmup retains its historical reference horizon: four main epochs for layers 0–4, three for layers 5–11, one local epoch, and two recovery epochs, even when fewer are executed. Replay evaluates the frozen final endpoint, without selecting intermediate replay epochs on the 244-example validation set; validation was nevertheless used to compare complete protocols during development.

Evaluation scope. All reported endpoints have pure cubics in all 12 FFNs. Their accuracies are plaintext classification results on the filtered validation sets, using standard Attention, LN, and pooler operators. They do not measure full-circuit FHE classification accuracy. The schedules have different budgets; the RTE internal holdout is drawn from the teacher's fine-tuning training set. For encrypted realization, the frozen activation's affine maps and representation factors are folded into adjacent linear maps under the compiled level/scale interfaces.

F.2 Offline Numerical Circuit Synthesis

The offline synthesizer accepts a frozen distilled model, a calibration set, and a library of level-compatible polynomial graphs. A site is indexed by (ℓ, h, f) for Attention exponential or reciprocal and by (ℓ, s, f) for the two LN positions. Synthesis proceeds in four steps.

Step 1: register feasible circuit templates. For each nonlinearity f, candidates are $\mathcal{C}_f = (d_f, \mathcal{G}_f, \bar{I}_f, \rho_f, \varepsilon_f)$, where d_f is the degree, $\mathcal{G}_f$ the evaluation graph, $\bar{I}_f$ the initial working interval, $\rho_f < 1$ the reserved bootstrap radius, and ε_f the plaintext error target. Dense-grid and Remez tests reject candidates whose compiled depth exceeds the level budget or whose error exceeds ε_f before model traces are inspected.

Step 2: collect site-wise plaintext traces. The profiler executes the frozen student and records the shifted exponential input $u = QK^{\mathsf{T}}/\sqrt{d_h} - g$, the valid-row Softmax denominator $D = \sum_{k \in \mathcal{V}_s} \exp(u_k)$, and the LN1/LN2 variance denominator $V = d^{-1} \sum_i (x_i - \mu)^2 + \epsilon_{\mathrm{LN}}$. Padding is excluded. For each site, the trace summary retains extrema, quantiles, tail density, and every lower-bound witness needed by reciprocal or reciprocal square root.

Step 3: align coordinates and tighten the interval. For each site, the search selects a public map $\widetilde{x} = \phi^{(f)}_{\ell,h}(x;\theta)$ and solves for the narrowest $I = [a,b]$ that satisfies a fixed coverage gate, contains the trace envelope plus asymmetric margins, maps the protected pre-bootstrap envelope into $[-\rho_f, \rho_f]$, and obeys $a \geq a^{(f)}_{\min}$ for singular functions. The search first considers semantics-preserving transformations. A row-common Softmax shift scales the exponential numerator and RowSum together and cancels after normalization; an LN scale is paired with restoration after reciprocal square root. These factors are folded into adjacent weights or constants. When additional lower-bound margin is required, the search extends this family with a pseudocount from a predefined public candidate set.

Step 4: fit, compile, and freeze. For each fixed interval and degree, Remez search in the Chebyshev basis minimizes $\max_{x\in I} |f(x)-p(x)|$ for regular functions, $\max_{x\in I} |1-xp(x)|$ for reciprocal, or $\max_{x\in I} |1-\sqrt{x}\,p(x)|$ for reciprocal square root. The selected polynomial is decomposed according to $\mathcal{G}_f$. A candidate is published only after exact-plaintext, polynomial-plaintext, bootstrapping-input-domain, compiled level, and scale checks. The artifact stores the physical interval, coordinate map, guard, coefficients, restoration factor, expected input/output levels and scales, and a configuration version. All guards and stabilizing constants are fixed before FHE execution; the server does not branch on an encrypted denominator.

F.3 Online FHE Evaluation Procedures

This appendix specifies the compiled online evaluation graph introduced in Section 5, from the protected Attention primitives to the complete 12-layer inference procedure.

Encrypted length interface. The current protected-Attention path consumes four client ciphertexts: ct_ρ (`inverse_m_by_lane`), ct_ω (`variance_factor_by_lane`), $ct_{\bar{\ell}}$ (`normalized_log_m`), and $ct_{\bar{c}}$ (`normalized_c_m`). For an active lane of length $m = m_b > 0$, their values are

$$\rho_m = 1/m, \qquad \omega_m = 0.1\sqrt{2\ln m}/m, \qquad \bar{\ell}_m = \ln m/\ln 128, \qquad \bar{c}_m = \sqrt{2\ln m}/\sqrt{2\ln 128}.$$

Each ciphertext is encoded in row-major SIB format, repeats its lane scalar over all row blocks, starts at level 6, and contains zero for an inactive lane. The server derives $ct_{1-\rho} = 1 - ct_\rho$. The shared request interface additionally contains ct_κ (`shift_offset_by_row`), a row-masked legacy shift field whose valid-slot plaintext is $\kappa_{b,r} = 1.25\sqrt{2\ln m_b} - \delta + \iota_{\text{fuse}}(\frac{1}{2}\ln m_b - \ln\lambda)$ when $m_b > 0$ and $r < m_b$, and zero otherwise. Here δ, λ, and $\iota_{\text{fuse}} \in \{0,1\}$ are public legacy-configuration values. This level-6 field is retained for compatibility but is not an input to the current row-common range-alignment circuit. The measured request fixture is encoded with the same row-major SIB packing with $\lambda = 1.5$, $\delta = \ln(0.90) - 0.85$, and $\iota_{\text{fuse}} = 1$.

External level contract. Table 9 records the interfaces between the major procedures. Layer 0 consumes fresh level-6 inputs and reaches levels 5 and 4 after Q/K/V and QK^T. Later layers start at level 4 and reach levels 3 and 2 before the score refresh. The table shows this common inter-layer path, the E4 exponential schedule, and the reciprocal and reciprocal-square-root interfaces fixed offline.

Table 9: Online packing and level contract.

Stage	Input packing/level	Output packing/level	Live width
Layer input	column/Lev 4	column/Lev 4	768
Q/K/V	column/Lev 4	column/Lev 3	3×768
QK^T	column/Lev 3	rotated diagonal/Lev 2 $\to 4^*$	128/head
Degree-2 row-common shift	rotated diagonal/Lev 4	row vector/Lev 2	1/head
masked E4 affine	rotated diagonal/Lev 2	rotated diagonal/Lev 1 $\to 4^*$	128/head
E4 inner	rotated diagonal/Lev 4	rotated diagonal/Lev 0	128/head
E4 reconstruction	rotated diagonal/Lev 4^*	Lev 3 $\to 2 \to 1 \to 4^*$	128/head
full-C mask	rotated diagonal/Lev 4	rotated diagonal/Lev 3	128/head

Continued on the next page.

Table 9 continued.

Stage	Input packing/level	Output packing/level	Live width
CV / RowSum	rotated diagonal/Lev 3	column/Lev 2 / row vector/Lev 3	64+1/head
head normalization	column/Lev 2	column/Lev 1	64/head
pre-LN1 boundary	column/Lev 0	column/Lev 4^*	768
LN1 scalar rsqrt	row vector/Lev 4	row vector/Lev 4^*	1
LN1 normalized-core Boot	column/Lev 1	column/Lev 4^*	768
FC1 / p_ℓ / FC2	column/Lev 4	Lev $3 \to 1 \to 0$	3072→768
pre-LN2 boundary	column/Lev 0	column/Lev 4^*	768
LN2 scalar rsqrt	row vector/Lev 4	row vector/Lev 4^*	1
post-LN2 boundary	column/Lev 1	column/Lev 4^*	768
public row-0 selection / pooler	column/Lev 4	column/Lev $2 \to 4^*$	768
D63/Mod-8 tanh	column/Lev 4	column/Lev 4^*	768
classifier	column/Lev 4	column/Lev 3	c

Note. Lev 4^* denotes a refresh to the compiled level-4 interface.

In the pseudocode, $\mathrm{MatchLevel}(a, b)$ returns a copy of a lowered to the level of b, without Boot or Rescale; its calls assume that a starts at least as high as b. Addition operands use compatible CKKS scales and representation units. Ciphertext products include relinearization; each explicit Rescale consumes one logical level. Public scalar and affine operations follow their compiled encodings and scheduled Rescales.

The procedures share request-local encrypted weighted-mask caches $ct^C_{W_\rho,k}$ and $ct^C_{W_\omega,k}$, representing $M^C_k\rho$ and $M^C_k\omega$. The server derives them from the uploaded metadata and reuses them in masked moments and pseudocount addition; they are not additional uploads. Cache construction is performed once per request. The compiled public-scalar operation produces pseudocount terms at a level no lower than 3, with scales and representation units compatible with the post-mask output. Reusing the cache for this addition requires no new ciphertext–ciphertext multiplication or Boot.

Composition of the online graph. The six procedures progress from local encrypted primitives to complete inference. Algorithms 1 and 2 implement the masked Attention path; Algorithms 3 and 4 complete the normalization, residual, and FFN paths of one Encoder block. Algorithms 5 and 6 then compose these routines into one Encoder layer and the complete 12-layer inference and classification graph. Table 9 summarizes their packing and level interfaces.

Algorithm 1 Mask-aware site-specific exponential

Require: score diagonals ct^{diag}_S; ct_ρ, ct_ω, $ct_{\bar\ell}$, $ct_{\bar c}$; full-C masks; shared caches $ct^C_{W_\rho}, ct^C_{W_\omega}$; public $\tau_{\ell h}$; artifacts $\mathcal{A}^G_{\ell h}$ and $\mathcal{A}^E_{\ell h}$

Ensure: masked weight diagonals ct^{diag}_C at the compiled post-mask level

$(ct_\mu, ct_t) \leftarrow \textsc{MaskedMoments}(ct^{\mathrm{diag}}_S, ct^C_M, ct_\rho, ct_\omega; ct^C_{W_\rho}, ct^C_{W_\omega})$
$ct_{1-\rho} \leftarrow 1 - ct_\rho$
$ct_g \leftarrow \textsc{RowCommonShift}(ct_\mu, ct_t, ct_{\bar\ell}, ct_{\bar c}, ct_{1-\rho}; \mathcal{A}^G_{\ell h})$ ▷ one-ciphertext Boot; degree-2 output at Lev 2
$ct^{\mathrm{diag}}_U \leftarrow ct^{\mathrm{diag}}_S - ct_g$ ▷ level-align ct_S; output at Lev 2
$ct_Z \leftarrow \textsc{MaskedAffineSafeFill}(ct^{\mathrm{diag}}_U, ct^C_M, t_{\mathrm{pad}}; \mathcal{A}^E_{\ell h})$ ▷ invalid slots map to $t_{\mathrm{pad}} = -0.5$; Lev : $2 \to 1$
$ct_Z \leftarrow \textsc{Boot}(ct_Z)$ ▷ mapped E4 input at Lev 4
$ct_T \leftarrow \textsc{EvalD15}(ct_Z; \mathcal{A}^E_{\ell h})$ ▷ degree-15 inner, Lev : $4 \to 0$
$ct_T \leftarrow \textsc{Boot}(ct_T)$
$ct_E \leftarrow \textsc{Rescale}(ct_T \odot ct_T)$
$ct_E \leftarrow \textsc{Rescale}(ct_E \odot ct_E)$ ▷ fourth-power reconstruction
$ct_E \leftarrow \textsc{Restore}(ct_E; \mathcal{A}^E_{\ell h})$
$ct_E \leftarrow \textsc{Boot}(ct_E)$
for each rotated diagonal k **do**

$ct'_{M,k} \leftarrow \textsc{MatchLevel}(ct^C_{M,k}, ct_{E,k})$
$ct_{C,k} \leftarrow \textsc{Rescale}(ct'_{M,k} \odot ct_{E,k})$ ▷ full-C mask; level $4 \to 3$
if $\tau_{\ell h} > 0$ **then**
  $ct_{a,k} \leftarrow \textsc{MulPublicScalar}(ct^C_{W_\rho,k}, \tau_{\ell h})$
  $ct'_{a,k} \leftarrow \textsc{MatchLevel}(ct_{a,k}, ct_{C,k})$
  $ct_{C,k} \leftarrow ct_{C,k} + ct'_{a,k}$ ▷ reuse cached $M^C_k \rho$; output remains level 3
end if
end for
return ct_C^{diag}

Algorithm 2 One protected encrypted Attention head

Require: layer/head indices ℓ, h; public $\sigma_{\mathrm{in},\ell}$, $\sigma_{D,\ell h}$, d_{fill}, $\tau_{\ell h}$; column-packed ct_X; W_Q^h, W_K^h, W_V^h; ct_ρ, ct_ω, $ct_{\bar{\ell}}$, $ct_{\bar{c}}$; row and full-C masks; shared caches $ct^C_{W_\rho}, ct^C_{W_\omega}$; artifacts $\mathcal{A}^G_{\ell h}, \mathcal{A}^E_{\ell h}, \mathcal{A}^R_{\ell h}$
Ensure: 64 column-packed context ciphertexts ct_H^h at level 1
$(ct_Q, ct_K, ct_V) \leftarrow \textsc{ScaleAwareQKV}(ct_X, W_Q^h, W_K^h, W_V^h, \sigma_{\mathrm{in},\ell})$ ▷ $1/\sqrt{8}$ is folded into Q and K
$ct_S^{\mathrm{diag}} \leftarrow \textsc{DoubleBSGSScore}(ct_Q, ct_K)$
if $\ell > 0$ **then**
  $ct_S^{\mathrm{diag}} \leftarrow \textsc{RefreshToLevelFour}(ct_S^{\mathrm{diag}})$
end if
$ct_C^{\mathrm{diag}} \leftarrow \textsc{MaskAwareExp}(ct_S^{\mathrm{diag}}, ct_\rho, ct_\omega, ct_{\bar{\ell}}, ct_{\bar{c}}, ct^C_M, ct^C_{W_\rho}, ct^C_{W_\omega}, \tau_{\ell h}; \mathcal{A}^G_{\ell h}, \mathcal{A}^E_{\ell h})$
$ct_N[0{:}63] \leftarrow \textsc{DoubleBSGSCV}(ct_C^{\mathrm{diag}}, ct_V)$
$ct_D \leftarrow \textsc{RowSum}(ct_C^{\mathrm{diag}})$
$ct_q \leftarrow ct^C_{M,0}$ ▷ zero-offset diagonal gives query-row validity
$ct_D \leftarrow ct_D + \sigma_{D,\ell h} d_{\mathrm{fill}}(1 - ct_q)$ ▷ safe fill in the represented denominator units
$ct_D \leftarrow ct_D + \overline{ct_D}$ ▷ $2\,\mathrm{Re}(D)$; halve the inverse affine slope
$ct_Z \leftarrow \textsc{Affine}(ct_D; \mathcal{A}^R_{\ell h})$
$ct_Z \leftarrow \textsc{Boot}(ct_Z)$
$ct_r \leftarrow \textsc{MinimaxReciprocal}(ct_Z; \mathcal{A}^R_{\ell h})$
▷ D63/Mod-8: two internal Boots; D31/Mod-4: compiled schedule
for $j = 0, \ldots, 63$ **do**
  $ct'_r \leftarrow \textsc{MatchLevel}(ct_r, ct_N[j])$
  $ct^h_{H,j} \leftarrow \textsc{Rescale}(ct_N[j] \odot ct'_r)$
end for
return ct_H^h

Algorithm 3 One LayerNorm site

Require: scaled residual $ct_R[0{:}767]$ representing σR; encrypted row mask; LN artifact $\mathcal{A}^{\mathrm{LN}}_{\ell s}$
Ensure: normalized core $\sigma_{\mathrm{out}} n$, or materialized $\sigma_{\mathrm{out}}(\gamma n + \beta)$
$ct_\mu \leftarrow 768^{-1} \sum_j ct_R[j]$
for $j = 0, \ldots, 767$ **do**
  $ct_Z[j] \leftarrow ct_R[j] - ct_\mu$
end for
$ct_V \leftarrow 768^{-1} \sum_j ct_Z[j]^2 + \sigma^2(\epsilon_{\mathrm{LN}} + \tau^{\mathrm{LN}}_{\ell s})$
$ct_V \leftarrow \textsc{SafeFillInvalidRows}(ct_V, ct^{\mathrm{row}}_M, \mathcal{A}^{\mathrm{LN}}_{\ell s})$
$ct_U \leftarrow \textsc{Affine}(ct_V; \mathcal{A}^{\mathrm{LN}}_{\ell s})$
$ct_U \leftarrow \textsc{Boot}(ct_U)$
$ct_r \leftarrow \textsc{MinimaxRSqrt}(ct_U; \mathcal{A}^{\mathrm{LN}}_{\ell s})$
if the site artifact enables real projection **then**
  $ct_r \leftarrow ct_r + \overline{ct_r}$ ▷ fold $1/2$ into restoration
end if
for $j = 0, \ldots, 767$ **do**
  $ct_n[j] \leftarrow \textsc{Rescale}(ct_Z[j] \odot ct_r)$
  $ct_n[j] \leftarrow \textsc{Restore}(ct_n[j]; \mathcal{A}^{\mathrm{LN}}_{\ell s})$

16: **if** materialized affine output is requested **then**
17: $ct_n[j] \leftarrow \gamma_j\, ct_n[j] + \sigma_{\text{out}}\beta_j$
18: **end if**
19: **end for**
20: **return** ct_n

Algorithm 4 Fused post-Attention path

Require: ct_X, concatenated ct_H, encrypted row mask, boundary scales $\sigma_{0:3,\ell}$, distilled FFN, and LN1/LN2 artifacts
Ensure: next-layer interface $ct_{X'}$ representing $\sigma_{3,\ell}X_{\ell+1}$ at level 4
$ct_H \leftarrow ct_H + \text{Conj}(ct_H)$ ▷ once after concatenation; fused W_O absorbs $1/2$
$ct_{R_1} \leftarrow \textsc{FusedSelfOutputResidual}(ct_H, ct_X, W_O, b_O, \sigma_{0,\ell})$
$ct_{R_1} \leftarrow \textsc{Boot}(ct_{R_1})$
$ct_{n_1} \leftarrow \textsc{LayerNormSite}(ct_{R_1}, ct_M^{\text{row}}, \mathcal{A}_\ell^{\text{LN1}}, \text{core})$
$ct_{n_1} \leftarrow \textsc{Boot}(ct_{n_1})$
$ct_{\text{res}} \leftarrow \textsc{MaterializeLN1Affine}(ct_{n_1}, \gamma_1, \beta_1, \sigma_{2,\ell})$
$ct_z \leftarrow \textsc{FusedLN1FC1Map}(ct_{n_1}, W_1, b_1)$ ▷ Lev : $4 \to 3$
$ct_p \leftarrow \textsc{EvalDegree3}(ct_z; p_\ell)$ ▷ Lev : $3 \to 1$; no 3072-wide Boot
$ct_y \leftarrow \textsc{FusedActivationFC2Map}(ct_p, W_2, b_2)$
$ct'_{\text{res}} \leftarrow \textsc{MatchLevel}(ct_{\text{res}}, ct_y)$
$ct_{R_2} \leftarrow ct_y + ct'_{\text{res}}$ ▷ pre-LN2 residual at Lev 0
$ct_{R_2} \leftarrow \textsc{Boot}(ct_{R_2})$
$ct_{X'} \leftarrow \textsc{LayerNormSite}(ct_{R_2}, ct_M^{\text{row}}, \mathcal{A}_\ell^{\text{LN2}}, \text{affine})$
$ct_{X'} \leftarrow \textsc{Boot}(ct_{X'})$
$ct_{X'} \leftarrow ct_{X'} + \overline{ct_{X'}}$ ▷ γ_2, β_2 precompensate $1/2$
return $ct_{X'}$

Algorithm 5 One encrypted BERT-base Encoder layer

Require: layer input $ct_X[0{:}767]$, encrypted metadata, and compiled layer artifact $\mathcal{A}_\ell$
Ensure: next-layer interface $ct_{X'}[0{:}767]$
for $h = 0, \ldots, 11$ **do**
load $\mathcal{A}_{\ell h}^E$ and $\mathcal{A}_{\ell h}^R$
$ct_H^h \leftarrow \textsc{ProtectedAttentionHead}(ct_X, h, ct_{\text{meta}}, \mathcal{A}_\ell)$
release the head-local polynomial objects and temporary diagonals
end for
$ct_H \leftarrow \textsc{Concat}(ct_H^0, \ldots, ct_H^{11})$
$ct_{X'} \leftarrow \textsc{FusedPostAttention}(ct_X, ct_H, ct_M^{\text{row}}, \mathcal{A}_\ell)$
release ct_H and the layer-local FFN state
return $ct_{X'}$

Algorithm 6 Non-interactive 12-layer inference and classification

Require: SIB feature ciphertexts ct_X; encrypted lengths and masks ct_{meta}; model and compiled artifact manifest $\mathcal{A}$
Ensure: encrypted logits ct_{logits}
for $\ell = 0, \ldots, 11$ **do**
$ct_X \leftarrow \textsc{EncryptedEncoderLayer}(ct_X, ct_{\text{meta}}, \mathcal{A}_\ell)$
end for
$ct_P \leftarrow \textsc{ScaleAwarePooler}(ct_X, W_p, b_p, \sigma_{3,11})$
$ct_P \leftarrow M^{\text{CLS}} \odot ct_P$ ▷ fixed public plaintext selector for row zero in every lane
$ct_P \leftarrow \textsc{Boot}(ct_P)$
$ct_T \leftarrow \textsc{MinimaxTanhD63Mod8}(ct_P; \mathcal{A}^{\tanh})$
$ct_T \leftarrow ct_T + \overline{ct_T}$ ▷ classifier weights absorb $1/2$ and output restoration
$ct_{\text{logits}} \leftarrow \textsc{ClassifierLinear}(ct_T, W_c, b_c)$
return ct_{logits} to the client

F.3.1 Accumulation Scheduling

Dense ciphertext–plaintext projections accumulate all products for one output ciphertext before applying a single Rescale. The ciphertext–ciphertext products in QK^{T} and CV are likewise accumulated before a fused Relinearize–Rescale. This schedule avoids repeatedly rescaling or relinearizing partial sums while preserving the required common level and scale of every addition operand. Public constants are pre-encoded, and ciphertext views and accumulation scratch are reused within each layer. These choices are part of the production implementation measured in Table 10; they do not change the mathematical graph or its level schedule.

Table 10: Encrypted BERT-base runtime breakdown.

Component	Operation	Level	Packing	Tensor shape × calls	Time (s/input)
Attention	Q/K/V projections	$4 \to 3^{\dagger}$	Col→Col	$(\mathbb{R}^{128\times768} \times \mathbb{R}^{768\times64}) \times 432$	0.604
	Protected head circuit	$3 \to 1$ (7/8 Boots)	Col→ RotDiag→ Col	$(\mathbb{R}^{128\times64}, \mathbb{R}^{128\times128}) \times 144$	12.227
SelfOutput	Output projection + residual	$1 \to 0$	Col→Col	$(\mathbb{R}^{128\times768} \times \mathbb{R}^{768\times768}) \times 12$	0.196
	Boundary Boot	$0 \xrightarrow{\text{Boot}} 4$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	1.400
	LayerNorm-1 core	$4 \to 1$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	0.062
	Normalized-core Boot	$1 \xrightarrow{\text{Boot}} 4$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	1.402
Intermediate	FC1	$4 \to 3$	Col→Col	$(\mathbb{R}^{128\times768} \times \mathbb{R}^{768\times3072}) \times 12$	0.798
	Degree-3 activation	$3 \to 1$	Col→Col	$\mathbb{R}^{128\times3072} \times 12$	0.073
Final	FC2 + residual	$1 \to 0$	Col→Col	$(\mathbb{R}^{128\times3072} \times \mathbb{R}^{3072\times768}) \times 12$	0.752
	Boundary Boot	$0 \xrightarrow{\text{Boot}} 4$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	1.400
	LayerNorm-2 core	$4 \to 1$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	0.064
	Output Boot	$1 \xrightarrow{\text{Boot}} 4$	Col→Col	$\mathbb{R}^{128\times768} \times 12$	1.404
Classification	Public row-0 selection + pooler + Boot	$4 \to 2 \xrightarrow{\text{Boot}} 4$	Col→Col	$\mathbb{R}^{1\times768} \times \mathbb{R}^{768\times768}$	0.252
	Degree-63 tanh	$4 \to 4$	Col→Col	$\mathbb{R}^{1\times768}$	0.268
	Classifier	$4 \to 3$	Col→Col	$\mathbb{R}^{1\times768} \times \mathbb{R}^{768\times c}$	< 0.001
Runtime	Alignment, layout, and orchestration	–	–	–	0.879
Total	Complete encrypted graph	–	–	–	**21.781**

Note. Times are amortized over the measured 2,788-second full 128-lane run and sum to 21.781 seconds per input. Tensor dimensions are per input; all rows use the same fixed full-capacity graph. †The common inter-layer path is shown; layer 0 starts at level 6 and leaves Q/K/V at level 5. Its head path uses seven Boots; later layers use eight because QK^{T} is additionally refreshed. The protected-head row includes QK^{T}, encrypted range alignment, E4, masking, CV, RowSum, reciprocal, and normalization.

G BERT Evaluation Details

G.1 Dataset, Runtime, and Communication Details

Environment and protocol. FHE measurements use an NVIDIA RTX 5090 (32 GiB) and 25 allocated Intel Xeon Platinum 8470Q cores. Cheddar (Choi et al., 2026) is compiled in Release mode with CUDA 12.0.140 and GCC 13.3 on Ubuntu 22.04 (driver 580.95.05). Examples exceeding

128 BERT tokens are removed without truncation. Accuracy is measured on the complete filtered validation sets. Repeated FHE measurements use fresh processes and encryption randomness.

Table 11: BERT evaluation setup and costs.

A. Filtered GLUE split sizes

Task	Train	Validation	Test
SST-2	67,349	872	1,821
QNLI	104,366	5,438	5,430
RTE	2,186	244	2,672

B. End-to-end runtime and peak GPU memory

Measurement	Inputs	Repetitions	Server time (s)	Amortized (s/input)	Peak GPU
Detailed single-input profile	1	1	2716	–	22.00 GiB
Single-input production	1	3	2730 ± 21 (mean$\pm$s.d.)	–	22.37 GiB
Full-capacity graph	128	1	2788	21.781	22.37 GiB

C. Raw communication footprint

Scope	Objects	Raw payload
Evaluation keys (once)	160	8.597 GiB
Warm client-to-server batch	902 ciphertexts	1.762 GiB
Server-to-client logits	2 ciphertexts	0.0039 GiB
Cold setup plus first batch	–	10.359 GiB

Note. Panel A applies the 128-token filter; reported task accuracy uses the validation split. Panel B measures the complete 12-layer Encoder, encrypted pooler, and classifier. The three one-input times are 2706, 2744, and 2740 s; the single full-capacity run takes 2788 s. Peak memory is driver-observed GPU memory. Panel C reports raw, uncompressed Cheddar RNS words.

The teacher and degree-3 student respectively classify 806/872 and 803/872 SST-2 examples, 4,978/5,438 and 4,945/5,438 QNLI examples, and 179/244 and 172/244 RTE examples correctly.

The measured 902-ciphertext request comprises 768 feature ciphertexts, 128 diagonal-validity masks, one row-validity ciphertext, four length-derived ciphertexts consumed by the current path—ct_{ρ}, ct_{ω}, $ct_{\bar{\ell}}$, and $ct_{\bar{c}}$—and one legacy shift-offset ciphertext ct_{κ} retained for interface compatibility.

G.2 Layer-0 FHE Error Diagnostics

Using three fresh FHE processes on the same Layer-0 workload, we summarize absolute errors over valid tokens at each feature coordinate. Figure 11 reports the median, interquartile range, P99, and maximum. Activation has 3,072 coordinates; each of the other checkpoints has 768.

G.3 Layer-Level FHE Fidelity

Table 12B lists the per-layer MAEs; Table 12A summarizes mean errors and error quantiles across layers. Each layer starts from its corresponding reference checkpoint. We first take the median of each statistic over three fresh FHE processes, then aggregate across layers as specified in the table note; these are not statistics pooled over all layer coordinates.

LN2 has the largest layer-local error at layers 3–5 and 10; at layer 10, for example, the mean is 0.136 and P99 is 0.514. These independent probes measure locally introduced error rather than error accumulated through all 12 layers.

H Deployment-Contract Ablation Details

The experiments below follow the three deployment-contract properties tested in the main text: compact-ring resource use, fixed execution across request shapes, and cross-module numerical alignment.

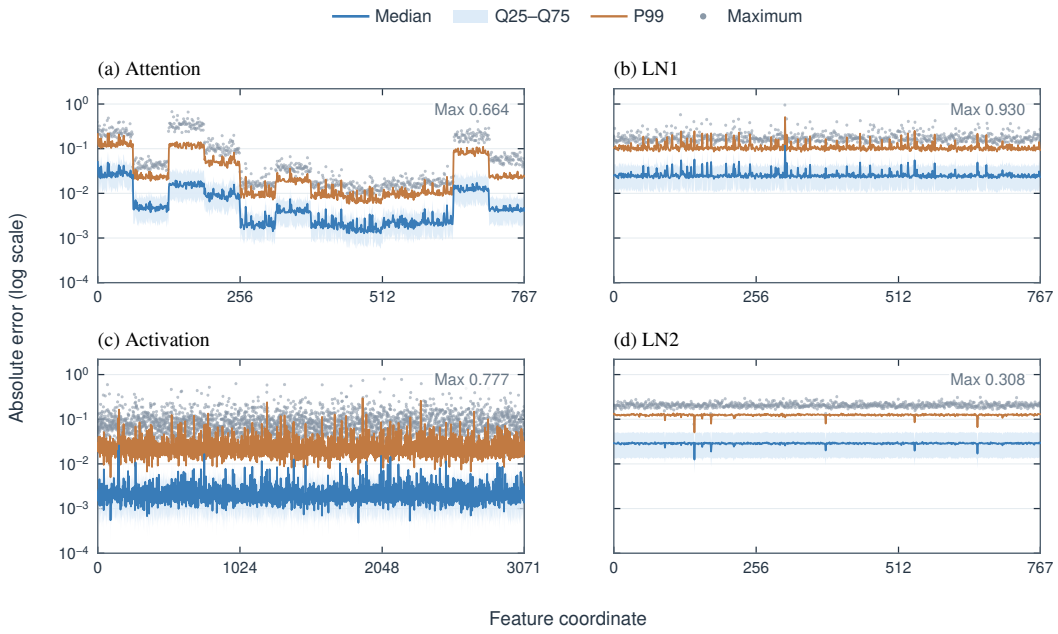


Figure 11: Coordinate-wise Layer-0 FHE errors for 100 SST-2 inputs (1,675 valid tokens) over three fresh FHE processes. Lines show the median and P99, shaded bands show Q_{25}–Q_{75}, and gray markers show maxima.

Table 12: Layer-local FHE fidelity.

A. Checkpoint error distribution

Checkpoint	Mean	Q25	Median	Q75	P95	P99
Attention	0.02208	0.00283	0.00681	0.01632	0.09798	0.21285
LN1	0.03928	0.01580	0.03365	0.05851	0.10563	0.15025
Activation	0.00464	0.00078	0.00221	0.00539	0.01539	0.03002
LN2	0.09769	0.02781	0.06051	0.10801	0.19464	0.26517

B. Per-layer mean absolute error

Layer	Attention	LN1	Activation	LN2
0	0.01001	0.03118	0.00389	0.03526
1	0.01749	0.04146	0.00522	0.04780
2	0.02901	0.05632	0.00913	0.05991
3	0.02379	0.04830	0.00474	0.13024
4	0.03720	0.05730	0.00395	0.23041
5	0.03497	0.05033	0.00623	0.19689
6	0.04834	0.05872	0.00873	0.09568
7	0.03265	0.04142	0.00581	0.07820
8	0.01167	0.02715	0.00278	0.06715
9	0.00625	0.01631	0.00101	0.07233
10	0.00946	0.01845	0.00150	0.13574
11	0.00418	0.02441	0.00264	0.02263

Note. Errors use valid tokens in active lanes. Each per-layer statistic is the median over three fresh FHE runs. Panel A averages the 12 per-layer MAEs for Mean and takes the median across layers separately for each quantile. Panel B lists the per-layer MAEs. Values in this table and Table 1B are rounded directly from the unrounded summaries to five and four decimals, respectively.

H.1 Compact Configuration and Fixed-Execution Resources

Table 13 separates three measurements: a matched ring-dimension comparison (A), heterogeneous client shape profiles (B), and a fixed-length server occupancy sweep (C). They assess compact-ring savings, request-inventory invariance, and server resource use, respectively.

Table 13: Resource and fixed-execution ablations.

A. Compact-ring resources

Ring	Slots	Eval-key payload	Boot (ms)	Memory after Boot	Process wall (s)
$N = 2^{15}$	16,384	8.597 GiB	19.278	15.94 GiB	32.143
$N = 2^{16}$ control	32,768	17.304 GiB	25.018	24.56 GiB	86.188

B. Client shape profiles

Active	Valid tokens	Slot use	Prep. (s)
1	5	0.03%	15.507
8	501	3.06%	14.878
20	1,242	7.58%	14.869
32	2,128	12.99%	14.824
40	2,564	15.65%	15.062
60	3,838	23.43%	16.038
64	4,128	25.20%	15.885
80	5,192	31.69%	15.016
100	6,370	38.88%	14.835
120	7,756	47.34%	14.958
128	8,256	50.39%	15.332

C. Server occupancy sweep

Active	Wall time (s)	Peak GPU (MiB)
1	2730	–
8	2677	22,815
20	2677	22,783
32	2676	22,783
40	2680	22,783
60	2694	22,783
64	2796	22,975
80	2696	22,783
100	2709	22,783
120	2708	22,783
128	2788	–

Note. Panel A reports medians of three runs. Panel B varies client lengths and occupancy; every request contains 902 ciphertexts and 1.762 GiB of raw payload, and preparation times are medians of three fresh processes. Panel C independently times the complete 12-layer server graph at sequence length 20. Its one-input time is a three-run mean; most other occupancies have one run. Memory telemetry covers the nine intermediate occupancies. The client token counts in B do not describe the server workloads in C. Appendix H.2 discusses classification drift.

H.2 Evidence Scope

The client and server measurements in Table 13 use different inputs. Several multi-input server runs complete the graph but exhibit classification drift; these runs therefore support timing and memory claims, not multi-input classification correctness. The independent layer probes and pre-Boot domain audits provide separate evidence for local fidelity and numerical-range compliance.

H.3 Cross-Module Numerical Ablations

Table 14 separates attention-circuit fidelity (A) from pre-Boot domain coverage (B–C). In Table 14A, the compact scale-free exponential reduces relative ℓ_1 error from 7.359% to 0.978% (86.7%); independently narrowing the reciprocal interval reaches 0.937%. These circuit variants support the numerical design but are separate from the boundary audit. Table 14B and Table 14C audit four interfaces during a complete Layer-0 encoder evaluation on 100 fixed SST-2 inputs with 1,675 valid tokens.

Each boundary contributes $n_{\text{valid}} = 1{,}675 \times 768 = 1{,}286{,}400$ valid coordinates per run; padding and inactive slots are excluded. For decoded real values $x_{j,k}$ at boundary j, we report

$$P99_j = \text{Quantile}_{0.99}\big(\{|x_{j,k}|\}_{k=1}^{n_{\text{valid}}}\big), \qquad r_j = \frac{100}{n_{\text{valid}}} \sum_{k=1}^{n_{\text{valid}}} \mathbf{1}[|x_{j,k}| > 1].$$

Thus, $P99_j$ measures input magnitude, and r_j is the percentage outside the registered normalized Boot domain.

BASE uses $(\sigma_0, \sigma_1, \sigma_2, \sigma_3) = (0.05, 0.04, 0.01, 0.08)$. Each ablated arm sets selected factors to one and recompiles the corresponding affine compensation. Ring parameters, CKKS scales, packing, operator topology, and the Boot schedule remain fixed. No additional online conversion or refresh is introduced to repair an ablated boundary.

Table 14: Cross-module numerical ablations.

A. Attention-circuit fidelity

Circuit arm	Relative ℓ_1 error (%)	Reduction (%)
Legacy Exp restoration + D255 Mod-4 inverse	7.359 ± 0.531	–
Legacy Exp restoration + D31 Mod-8 inverse	6.527 ± 0.364	11.3
Compact Exp with output restoration	1.234 ± 0.117	83.2
Compact scale-free Exp	0.978 ± 0.057	86.7
Scale-free Exp + independently narrowed inverse	0.937 ± 0.041	87.3
Coupled RowSum/inverse, $\lambda = 1.0$	1.814 ± 0.099	75.3
Coupled RowSum/inverse, $\lambda = 1.4$	1.444 ± 0.060	80.4

B. Single-factor boundary effects

Factor	Prod.→Abl.	Boot boundary	P99		Outside-domain (%)	
			Prod.	Abl.	Prod.	Abl.
σ_0	$0.05 \to 1$	pre-LN1 residual	0.0812	1.6231	0.00	8.55
σ_1	$0.04 \to 1$	LN1 core	0.1003	2.5079	0.00	28.77
σ_2	$0.01 \to 1$	pre-LN2 residual	0.0338	3.3839	0.00	39.50
σ_3	$0.08 \to 1$	post-LN2 output	0.0676	0.8445	0.00	0.36

C. Complete Layer-0 contract: outside-domain (%)

Arm	Factors set to one	pre-LN1 residual	LN1 core	pre-LN2 residual	post-LN2 output
BASE	none	0.000	0.000	0.000	0.000
S0-ID	σ_0	8.554	0.000	0.000	0.000
S1-ID	σ_1	0.000	28.770	0.000	0.000
S2-ID	σ_2	0.000	0.000	39.503	0.000
S3-ID	σ_3	0.000	0.000	0.000	0.359
LN1-ID	σ_0, σ_1	8.558	28.756	0.000	0.000
LN2-ID	σ_2, σ_3	0.000	0.000	39.490	0.359
FFN-bridge-ID	σ_1, σ_2	0.000	28.762	39.497	0.000
ALL-ID	$\sigma_0, \sigma_1, \sigma_2, \sigma_3$	8.553	28.755	39.495	0.357

Note. **A:** Relative ℓ_1 errors are means $\pm$ sample standard deviations over three FHE processes; reductions are relative to the first row. These circuit-variant results are reported separately from the boundary-domain audit. **B:** Prod./Abl. denote production/ablated settings. P99 is the 99th percentile of pre-Boot absolute input magnitudes, not an error quantile. **B–C:** Boundary statistics are medians over three fresh FHE processes per arm and use valid coordinates only. Outside-domain means $|x| > 1$ at a registered pre-Boot boundary. Zero entries denote zero observed violations, not small nonzero fractions rounded to zero. **C:** FFN-bridge-ID uses its supplementary batch's matched BASE; other arms use the original batch's BASE. Both BASE batches have zero violations at all four boundaries and are not pooled. Every ablated arm exhibits violations in all three runs. Ciphertext topology, levels, CKKS scales, and the Boot schedule remain fixed.

The arms comprise four single-factor removals, three paired removals, and ALL-ID. FFN-bridge-ID removes (σ_1, σ_2), spanning the LN1 normalized core and the pre-LN2 residual across the FFN. It belongs to a supplementary measurement batch with its own three-run BASE control; the remaining arms use the original batch's three-run BASE. The two BASE batches are retained separately rather than pooled into six repetitions.

Simultaneous boundary compliance. Each paired removal produces violations at its two ablated boundaries, while unablated boundaries retain zero observed violations. FFN-bridge-ID, for example, yields 28.762% and 39.497% outside-domain coordinates at the LN1 core and pre-LN2 residual. ALL-ID produces violations at all four boundaries. With zero observed violations in both matched BASE controls, these results support simultaneous coverage of the four audited interfaces on this workload without extra online conversions or refreshes.

Percentile and execution scope. For S3-ID, the post-LN2 P99 magnitude is only 0.8445, yet 0.359% of valid coordinates exceed the registered domain. P99 must therefore be interpreted together with the outside-domain fraction. All runs complete and produce finite valid-boundary val-

ues, so successful execution alone does not establish domain compliance. The audit concerns these four pre-Boot boundaries on the fixed Layer-0 workload; it does not establish coverage of every polynomial approximation interval or end-to-end classification accuracy.

I LLaMA Evaluation Details

I.1 Experimental Setup

We evaluate Layer 0 of Meta-Llama-3-8B with its original weights, all 32 query heads, eight key/value heads, and the full $14{,}336 \to 4{,}096$ FFN down projection. The $N = 2^{15}$ configuration packs eight token rows into 2,048 lanes. Modules consume preceding ciphertext outputs; diagnostic copies are decrypted after execution and never fed back into the layer. Numerical coefficients, approximation intervals, packing, and refresh locations are frozen before test selection (Table 15).

Table 15: Frozen LLaMA evaluation setup.

Field	Setting
Model	Meta-Llama-3-8B (base); original weights
Model revision	`315b20096dc791d381d514deb5f8bd9c8d6d3061`
Dataset	`Salesforce/wikitext`, `wikitext-2-raw-v1`, official `test` split
Dataset revision	`b08601e04326c79dfdd32d625aee71d232d685c3`
Numerical configuration	`r3_a7p5_res_m15_n3_repair_v1`
Freeze / manifest	2026-09-14, 11:28:39 / 11:29:02
Packing	Eight token rows $\times$ 2,048 sequence lanes
Reference	HF CPU-FP32; PyTorch 2.11.0+cu128; Transformers 4.57.6
Hardware	RTX 5090; ciphertexts staged in host RAM
GPU memory sampling	10 ms interval

Each input is an eight-token prefix, including BOS, from WikiText-2's official test split. Of 4,358 lines, we exclude 1,467 blank lines, 241 short lines, 28 prefixes matching earlier project inputs, and 69 duplicate prefixes, leaving 2,553 inputs and 20,424 token positions. Selection precedes numerical evaluation. With shuffle seed `20260914`, batch sizes are $512, 512, 512, 512, 505$; three rounds use a fresh FHE process for each batch.

I.2 Fidelity and Domain Coverage

Errors compare outputs restored to their original units with the original-model HF CPU-FP32 reference. For checkpoint k in round r,

$$e_j^{(r,k)} = \left| \widehat{x}_j^{(r,k)} - x_j^{(k),\mathrm{FP32}} \right|, \qquad \mathrm{MAE}^{(r,k)} = \frac{1}{M_k} \sum_{j=1}^{M_k} e_j^{(r,k)}.$$

Each round pools all real coordinates across five batches, including BOS and excluding padding: $M_k = 83{,}656{,}704$ for hidden-width outputs and $292{,}798{,}464$ for SiLU/SwiGLU. P99 uses linear interpolation; Table 16 reports medians and sample SDs across three rounds. Errors include upstream propagation and diagnostic level reduction; RMSNorm and attention are measured after refresh.

Table 16: Checkpoint errors ($\times 10^{-3}$). Entries show median $\pm$ sample SD across three pooled rounds.

Checkpoint	MAE	P99
RMSNorm-1	0.411567 ± 0.000272	4.504482 ± 0.001721
Attention	0.378274 ± 0.000588	1.791135 ± 0.003166
RMSNorm-2	5.043844 ± 0.006320	17.694018 ± 0.019028
SiLU	2.279220 ± 0.003385	9.317488 ± 0.018081
SwiGLU	0.212814 ± 0.000275	1.226046 ± 0.001399
Layer output	$\mathbf{0.465217} \pm 0.000426$	$\mathbf{1.678629} \pm 0.003441$

The layer-output MAE's SD is 0.0917% of its median. Domain checks cover 11 boundaries (Table 17), using causal valid terms for the exponential and all computed QK scores. Padding is checked after safe substitution. All checks pass in every round, with no non-finite values. This coverage applies to the tested eight-token inputs; the exponential input reaches 0.990669 within its $[-1, 1]$ domain.

Table 17: Audited input domains. Extrema span three rounds; counts are per round. All boundaries have zero observed violations and non-finite values.

Source boundary	Allowed interval	Observed range	Count/round
RMSNorm-1 variance map	$[-1, 1]$	$[-0.916713, 0.666880]$	20,424
RMSNorm-1 output Boot	$[-1, 1]$	$[-0.709161, 0.736835]$	83,656,704
QK source Boot	$[-1, 1]$	$[-0.419685, 0.343264]$	5,228,544
Exponential input map	$[-1, 1]$	$[-0.685145, 0.990669]$	2,941,056
Reciprocal denominator	$[0.236452, 5.013026]$	$[0.286004, 3.547145]$	653,568
Reciprocal input map	$[-1, 1]$	$[-0.979244, 0.386220]$	653,568
Attention output Boot	$[-1, 1]$	$[-0.880306, 0.691480]$	83,656,704
First residual Boot	$[-1, 1]$	$[-0.771757, 0.720353]$	83,656,704
RMSNorm-2 variance map	$[-1, 1]$	$[-0.924612, 0.909813]$	20,424
RMSNorm-2 output Boot	$[-1, 1]$	$[-0.698991, 0.693880]$	83,656,704
SiLU gate map	$[-1, 1]$	$[-0.663412, 0.798338]$	292,798,464

I.3 Runtime and Memory

Table 18 summarizes all 15 test processes. Layer time includes ciphertext transfers and diagnostic snapshots (76.9–91.5 s), excluding key generation, input encryption, and final decryption. Full-process time also includes initialization and auditing. The complete down projection uses 128-column tiles; its timer excludes input staging, which remains in layer time. Capacity-normalized times divide by 2,048 lanes; actual batches contain 512 or 505 inputs.

Table 18: Runtime and memory across 15 processes. GPU peaks are sampled every 10 ms; capacity normalization divides by 2,048 lanes.

Metric	Median	SD	Range
Encrypted layer (s)	3758.093	129.995	3374.353–3891.577
Layer / configured lane (s)	1.835006	0.063474	1.647633–1.900184
Full down projection (s)	580.462	36.860	475.860–627.796
Down / configured lane (s)	0.283429	0.017998	0.232354–0.306541
Full process (s)	4997.548	132.528	4617.016–5142.462
Sampled GPU peak (GiB)	21.716980	0.000000	21.717
Peak host RSS (GiB)	73.756027	0.015898	73.723717–73.774960

I.4 Decoder Circuits

Margin settings $3/7.5/3/1.5$ determine per-channel representation factors for RMSNorm-1, attention, RMSNorm-2, and residuals. RMSNorm's γ is combined with these factors, and subsequent affine maps absorb their restoration, including the attention output projection. RoPE explicitly pairs dimensions i and $i + 64$ in each 128-dimensional head, for $0 \le i < 64$:

$$q'_{p,i} = q_{p,i} \cos\theta_{p,i} - q_{p,i+64} \sin\theta_{p,i},$$
$$q'_{p,i+64} = q_{p,i} \sin\theta_{p,i} + q_{p,i+64} \cos\theta_{p,i}.$$

The same map applies to K. Causal GQA uses encrypted occupancy masks, an encrypted maximum approximation, squared degree-63 exponential reconstruction, and a degree-63 reciprocal (Table 19).

For the first residual output Y and its RMSNorm-2 output R, the SiLU/SwiGLU circuit evaluates

$$g = RW_g, \qquad u = RW_u, \qquad p_{63}(t) \approx \tfrac{1}{2}\tanh(5t/2), \quad t \in [-1, 1],$$
$$\widehat{\mathrm{SiLU}}(g) = g \odot \big(\tfrac{1}{2} + p_{63}(g/5)\big), \qquad \widehat{X}_{\mathrm{out}} = Y + \big(\widehat{\mathrm{SiLU}}(g) \odot u\big)W_d.$$

Table 19: Frozen nonlinear settings; degree-63 polynomials use Mod-8 evaluation.

Setting	Value
RMSNorm-1 rsqrt source interval	$[2.400182513, 16.798800395] \times 10^{-5}$
RMSNorm-2 rsqrt source interval	$[3.492231329, 27.523940130] \times 10^{-5}$
Attention score factor	0.01676261966936708
Invalid normalized score for maximum	-0.45
Exponential source interval	$[-54, 0.75]$; squared reconstruction
Reciprocal denominator interval	$[0.2364519696537098, 5.013025897566642]$
Reciprocal restoration coefficient	4.975516126840756
First-residual restoration coefficient	0.45815902948379517
SiLU gate mapping radius	5; raw gate and raw up branches

Only the polynomial input branch is divided by five; gate and up outputs retain their original units. The measured layer ends at L0 (Table 20), without a next-layer refresh.

Table 20: Decoder module interfaces.

Path	Interface
Initial input / RMSNorm output	L4 / L4
Q/K/V projection	L4 $\rightarrow$ L3
Explicit RoPE	L3 $\rightarrow$ L2
QK product / score refresh	L2 $\rightarrow$ L1 / Boot $\rightarrow$ L4
Normalized attention / output refresh	L0 / Boot $\rightarrow$ L4
Gate / up projection	L4 $\rightarrow$ L3
SiLU / SwiGLU output	L2 / L1
Down projection / final residual output	L0 / L0

During development, increasing the attention margin from 6 to 7.5 removed an observed attention-output domain violation in three repeat runs. V representation coefficients were multiplied by 1.25, with restoration folded into the output projection; model weights and Q/P parameters were unchanged. On the same development inputs, MAE/P99 increased by 7.53%/7.72%. The 7.5 configuration was then frozen for the test set.

I.5 Independent Vocabulary-Head Evaluation

This separate experiment encrypts final hidden states from a plaintext LLaMA forward pass, then evaluates the full 128,256-class projection, repacking, and encrypted ArgMax on 512 positions in three FHE runs (Table 21). A selection is correct only if exactly one finite real component exceeds 0.5, matches the FP32-logit Top-1, and all finite-value checks pass; all other outcomes count as failures.

Table 21: Independent vocabulary-head results on 512 positions per run.

Metric	Run 1	Run 2	Run 3
Projection Top-1 agreement (count)	511/512	512/512	512/512
Encrypted ArgMax uniquely correct (count)	359/512	362/512	366/512
No selection (count)	27	24	26
Multiple selections (count)	124	126	119
Unique incorrect selection (count)	2	0	1
Projection MAE ($\times 10^{-3}$)	0.722949	0.726842	0.717335
Projection P99 ($\times 10^{-3}$)	2.975931	2.977952	2.938280
Full encrypted head (s)	11065.700	11137.519	11222.402
Vocabulary projection (s)	8188.996	8261.911	8321.563
Sampled GPU peak (GiB)	17.436768	17.436768	17.436768

Median unique-selection accuracy is 70.7031% (SD 0.6859 percentage points), versus 75% for the same-polynomial plaintext selector. Median head time is 11,137.519 s, with approximately

51.092 GiB peak host RAM. The projection largely preserves Top-1, but approximate ArgMax remains a source of selection errors.

I.6 Operator-Cost Estimates

Table 22 combines kernel measurements, matrix fits, and operator counts for 32 decoder layers and an output head under an earlier timing configuration. This historical estimate is not extrapolated from the final continuous Layer-0 measurements above and is not measured full-model latency. Costs are normalized to 2,048 configured sequence lanes; the total is rounded after summing unrounded values.

Table 22: Historical cost estimate per 2,048-lane capacity. The total uses unrounded component values.

Group	Estimated s/input
Degree-63 SiLU	14.197
Gate and up projections	7.727
Four hidden-state Boot groups	4.942
Down projections	3.844
Encrypted ArgMax, including repacking	3.710
Q/K/V and output projections	2.760
Vocabulary projection	1.080
Causal attention	0.488
RMSNorm, products, and residuals	0.049
Total	38.796

The 3.844 down-projection component extrapolates timings with input widths up to 4,096 to width 14,336; the 3.710 ArgMax component aggregates tile timings. These estimates are separate from the directly measured full-width down projection and continuous vocabulary head reported above. MOAI's reported total is 5.79 times the historical 38.796 estimate under the respective native hardware, parameters, and capacities; this ratio does not measure full-model speedup of the final implementation. The earlier 40.344 estimate assumes explicit RoPE, whereas 38.796 assumes RoPE fusion; their difference is a cost-model sensitivity comparison, not a paired measurement of the final layer.