SEND THE BRAIN, KEEP THE BODY: HYBRID PRIVATE TRANSFORMER INFERENCE WITH MINIMAL SECRET CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid private inference combines the computational strengths of homomorphic encryption (HE) and secure multiparty computation (MPC), but repeatedly converting intermediate states between them incurs substantial communication. Prior work optimizes operator partitioning and evaluation, but still organizes HE–MPC execution around transfers of intermediate states. We observe that the secret decisions involved in nonlinear evaluation can be separated from the high-dimensional numerical state. Based on this observation, we introduce **SHRINK**, a hybrid Transformer inference architecture with decoupled data and control paths. SHRINK sends MPC only decision-critical information, while HE continues the main computation on retained ciphertexts. MPC generates secret controls that HE applies to existing ciphertexts, enabling continued HE evaluation across nonlinear boundaries. We exploit Transformer structure to design compact control protocols for GELU, attention, and LayerNorm. We further tailor ciphertext representations and feedback to the HE operations that use these controls, and reuse control computations across data blocks, reducing communication and redundant homomorphic computation. On BERT-base and BERT-large, SHRINK reduces communication by 8.4×–1048.6× compared with hybrid inference baselines, while maintaining accuracy close to that of plaintext inference without retraining.

1 INTRODUCTION

Transformers are central to text understanding and generation, but their deployment as cloud services raises privacy concerns: users want to protect their inputs, while service providers need to protect proprietary model parameters. Private inference uses cryptographic protocols to protect both parties' data. Hybrid approaches combine homomorphic encryption, whose packed arithmetic efficiently supports linear operations, with secure multiparty computation, which provides efficient protocols for nonlinear operations. Their complementary strengths reduce the cost of either primitive alone.

Communication remains a major bottleneck in hybrid Transformer inference. Nonlinear processing commonly requires intermediate values to be converted from HE ciphertexts into secret shares, followed by a return conversion after MPC evaluation. Prior work reduces communication through efficient linear and nonlinear protocols, finer-grained HE–MPC partitioning, and operator fusion to reduce conversions (Pang et al., 2024; Lu et al., 2025; Xu et al., 2025). Even after these optimizations, conversion can account for about two thirds of total communication (Xu et al., 2025).

This cost motivates us to reconsider the purpose of conversion. Moving computation into MPC exploits its strength in nonlinear evaluation, but conventional conversion also transfers the numerical state and assigns MPC the task of materializing the result. We instead ask: **Can we send MPC only the minimal, decision-critical information, while letting HE continue the main computation?** This idea is inspired by the Staircase Project in Cixin Liu's *Death's End*, the final novel in the science fiction trilogy *The Three-Body Problem* (Liu, 2016): rather than sending the whole body, only Tianming Yun's brain is sent, with the expectation that the Trisolarans can put it to use. Our design follows the same principle. The wide numerical state remains encrypted in HE, while only a compact secret control crosses the boundary. This control supplies the branch, scale, or refinement information that HE needs to carry out the remaining computation on the ciphertexts it already holds.

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.

Our key observation is that nonlinear evaluation combines two distinct tasks: selecting an evaluation rule based on a secret input, and performing the arithmetic specified by that rule. Through mathematical reformulations, we separate these tasks: MPC makes the input-dependent decisions, while HE performs the corresponding arithmetic on retained ciphertexts. For inverse square root, for example, MPC selects a power-of-two scale and maps the input to a fixed interval, enabling HE to use the same polynomial across different input magnitudes. We call the input-dependent values returned by MPC in encrypted form *secret controls*.

Based on this observation, we propose SHRINK, an HE–MPC execution architecture with **decoupled data and control paths** (Figure 1). The data path retains encrypted intermediate states and performs arithmetic directly on them. The control path generates secret controls and returns them as ciphertexts. Together, the paths support nonlinear evaluation while preserving the encrypted state for subsequent computation. We exploit Transformer structure to reduce communication in this architecture.

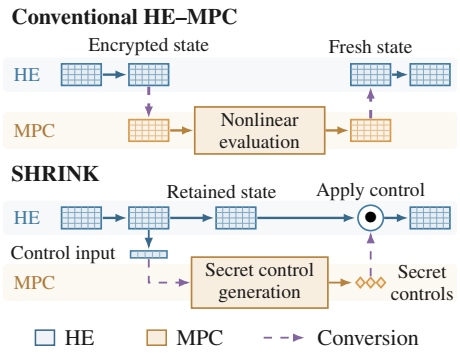

Figure 1: Decoupled HE–MPC execution.

To realize this division of work, we design secret control protocols that exploit Transformer structure. By combining model-dependent range analysis with mathematical decompositions of nonlinear functions, these protocols reduce the communication and computation required to generate controls, without retraining the model. Our design reduces the overall communication complexity of inference from $O(S^2)$ to $O(S)$ for sequence length S.

The communication and computation costs of controls also depend on how they are used in subsequent HE computation. We therefore develop homomorphic execution methods that exploit control structure. Shared ciphertext representations allow the same encrypted state to serve both the data and control paths. We compress feedback according to the components used by subsequent computation and reuse the results of homomorphic control evaluation, reducing both communication and repeated computation. We further schedule computation according to control dependencies and bootstrap normalized states to restore ciphertext depth, allowing HE computation to continue across Transformer layers.

Table 1: Logical values and control returns.

Boundary	Interface size		Return coefficients
	BLB	SHRINK	Full → compact
Attention	$\Theta(HS^2)$	$\Theta(HS)$	$N \to 2p_A$ ↓ **8×**
GELU	$\Theta(Sd)$	$\Theta(SK)$	$N \to N/2$ ↓ **2×**
LayerNorm	$\Theta(Sd)$	$\Theta(S)$	$N \to 2p_L$ ↓ **128×**

H: head count; S: sequence length; d: hidden width; K: padded number of GELU control channels.

N: HE polynomial degree; p_A, p_L: slots per repeated control block in Attention and LayerNorm, respectively.

At $S = 128$: $N = 2^{15}$, $p_A = 2^{11}$, $p_L = 2^7$.

We implement and evaluate the complete system on BERT-base and BERT-large. SHRINK reduces communication by 8.4×–1048.6× over prior hybrid private Transformer inference systems, while incurring negligible accuracy loss. Our contributions are as follows:

(1) An HE–MPC architecture with decoupled data and control. We reorganize nonlinear evaluation so that MPC generates secret controls and HE computes on retained ciphertexts. The architecture separates control generation, numerical evaluation, and state refresh, allowing encrypted states to persist across nonlinear boundaries.

(2) Secret control protocols from Transformer structure. We combine model-dependent range analysis, shared statistics, and secret scale reduction to construct control protocols for GELU, attention, and LayerNorm. These protocols reduce the data and computation involved in interaction while retaining the main numerical evaluation in HE.

(3) Homomorphic execution with structured controls. We design ciphertext representations shared by the data and control paths and compress control feedback according to the needs of subsequent HE computation. This design reuses existing ciphertexts and the results of control evaluation, reducing both communication and homomorphic computation.

2 BACKGROUND AND RELATED WORK

2.1 PROBLEM SETTING AND PRELIMINARIES

The client $\mathcal{C}$ holds a private input X and token-validity mask m; the server $\mathcal{S}$ holds model parameters W. The two parties jointly evaluate the Transformer, with the inference result returned to the client. We use S for sequence length, d for hidden width, and H for the number of attention heads, with head width $d_h = d/H$ and FFN width $d_f = 4d$.

Threat model. We consider a static semi-honest adversary corrupting at most one party (Huang et al., 2022; Lu et al., 2025; Xu et al., 2025). The corrupted party follows the protocol but may analyze its inputs, randomness, and received messages to infer the other party's private information. Our security analysis covers the conversion and control interfaces and the preservation of their guarantees under compact feedback (Appendices B and A.4).

Cryptographic background. Homomorphic encryption (HE) lets the server compute directly on ciphertexts. CKKS supports approximate real arithmetic and packs multiple values into a ciphertext, enabling matrix operations and polynomial evaluation. Multiplication and rescaling consume the ciphertext's remaining depth; bootstrapping can restore its capacity for further computation. Secure multiparty computation (MPC) evaluates functions through secret sharing and interactive protocols, including comparisons, selections, and other nonlinear operations. We write $\langle x \rangle$ for a CKKS ciphertext and $[x]$ for additive shares of the fixed-point encoding of x. Hybrid inference connects these representations through conversion protocols: HE-to-MPC conversion produces secret shares, and MPC-to-HE conversion produces ciphertexts for subsequent evaluation. The fresh ciphertexts also supply depth for that evaluation, so conversion affects both data layout and depth management.

2.2 RELATED WORK

HE-only inference. HE-only systems retain model computation in the encrypted domain, largely limiting online communication to inputs and outputs (Zhang et al., 2025a; Moon et al., 2025; Park et al., 2025). They approximate nonlinear functions with homomorphic arithmetic and use modulus scheduling or bootstrapping to sustain evaluation. This avoids intermediate interaction at the cost of homomorphic nonlinear evaluation and ciphertext depth management.

Hybrid HE–MPC inference. Hybrid systems combine the two primitives to improve secure evaluation. BOLT improves encrypted matrix multiplication, low-degree nonlinear approximations, and polynomial evaluation (Pang et al., 2024). BumbleBee develops communication-efficient multiplication protocols and optimizes piecewise approximation and branch selection (Lu et al., 2025). Nimbus uses input distributions to simplify nonlinear approximation while improving matrix multiplication (Li et al., 2024). BLB decomposes layers into operators and fuses consecutive arithmetic in CKKS to reduce conversion and truncation (Xu et al., 2025). These approaches primarily optimize operator efficiency, approximation complexity, and protocol partitioning and fusion, reducing the cost of securely evaluating numerical operators and their compositions.

3 DECOUPLED HE–MPC EXECUTION

3.1 MOTIVATION: RETHINKING CROSS-PROTOCOL COOPERATION

Hybrid private inference typically partitions operators between HE and MPC, then reduces execution cost through specialized protocols and operator fusion. This operator-centric partitioning does not address the fundamental source of cross-domain interaction. We make three observations.

Observation 1: Using MPC for nonlinear evaluation does not require transferring the full numerical state. In conventional hybrid execution, assigning a nonlinear operator to MPC also moves its numerical inputs and results across domains. Yet *the size of the numerical state need not match the amount of information MPC needs*. Secret comparisons or selections can operate on row statistics or selected channels while the remaining arithmetic acts on the full tensor. Interaction therefore need not follow the operator's full input and output interfaces. **Observation 2: MPC can only provide secret controls that let HE continue computing on the retained state.** Mathematical

reformulations allow MPC to select branches, scales, or initial approximations while HE performs the subsequent arithmetic on retained ciphertexts. MPC thus supplies the controls needed to continue HE evaluation without computing and returning the full nonlinear result. **Observation 3: The cost of these controls depends on their representation and use in subsequent HE computation.** Few secret control values can still be expensive to communicate or evaluate. Returning a full ciphertext, reconstructing components that subsequent computation discards, or broadcasting a control to a wide state before it is needed can retain much of the cost of state conversion. Communication and HE work thus depend on the control's representation, reuse across data blocks, and point of application.

Based on these observations, we build a hybrid execution architecture with **decoupled data and control paths**. High-dimensional numerical states remain on the HE data path, while a narrow control path generates the secret controls needed to continue their evaluation. Controls return in representations matched to subsequent HE computation and join the retained state where they are needed. Cross-domain interaction is thus organized around the required controls and their use in HE, rather than the size of nonlinear intermediate states.

3.2 Execution architecture

Table 2 summarizes the main changes to nonlinear evaluation, detailed in Section 4.

The data path retains the numerical state in HE; the control path supplies secret controls for its evaluation. The server retains $\langle X \rangle$ and reusable encrypted intermediate results, converting only aggregates or selected values needed by the control path into shares. The parties generate controls in MPC and return them as ciphertexts, which the server applies to the retained state (Figure 1). The same encrypted state thus supports both numerical computation and control extraction. The paths join where a control is consumed, allowing independent HE work to proceed while feedback is being generated. State refresh follows the available depth and numerical range, rather than the location of control generation. Appendix A.6 specifies the conversion and return interfaces.

Table 2: Nonlinear computations, their control decomposition, and retained HE state.

Operator	Original	Reformulation	Control / HE state
GELU	$\mathrm{GELU}(x)$	$\approx F_v(x)$	v / x
Attention	$\mathrm{softmax}_m(s)V$	$\approx r(EV)$	(ρ, r_0) / (A, V)
LayerNorm	$C/\sqrt{w}$	$\approx C\, a\, p_{\mathrm{LN}}(u)$	(a, u) / C

$F_v(x) \in \{0, P(x), x\}$; $A_j = m_j q((s_j - \mu)/B)$, $E_j = (\rho A_j)^8$; $u = a^2 w \in [1/4, 1)$. Protected inputs map to controls as $\widetilde{x} \to v$, $\widetilde{D}_1 \to \rho$, $\widetilde{D} \to r_0$, and $\widetilde{v} \to (a, u)$. Here $D_1 = \sum_j A_j^2$, $D = \sum_j E_j$; HE refines r_0 to r. GELU control uses selected channels.

3.3 Design challenges

(1) Control construction. Nonlinear protocols do not directly provide inexpensive control interfaces. Mathematical and protocol reformulation must produce information sufficient for subsequent HE evaluation without moving the cost of processing the full state into control generation. **(2) Shared state.** Numerical computation and control extraction can require different ciphertext layouts; organizing them independently duplicates rearrangements, rotations, and reductions. **(3) Control feedback.** Few logical control values do not imply small ciphertext messages. The layout, precision, and depth required by subsequent evaluation can force even a small control to occupy a large ciphertext representation. **(4) Continued execution.** Fresh control ciphertexts do not restore the retained state's depth, and feedback dependencies can stall the data path. Control application and state refresh must satisfy both computational dependencies and numerical constraints. Section 4 develops control protocols from Transformer structure. Section 5 develops homomorphic execution methods that exploit control structure to address shared state, feedback, and continued evaluation.

4 Secret Control Protocols

We introduce secret control protocols for GELU, attention, and LayerNorm. The protocols use Transformer structure to reduce both the data sent to MPC and the cost of generating controls. HE evaluates the remaining arithmetic on retained ciphertexts (Table 2). Tildes denote protected numerical estimates. Section 5 describes how the controls are represented and used in HE.

4.1 GELU CONTROL

(1) Separate branch selection from numerical evaluation. Let $h \in \mathbb{R}^{S\times d_f}$ be the FFN activations before GELU. We approximate GELU by zero in the negative tail, a polynomial $P(x)$ on $[-\tau, \tau]$, and the identity in the positive tail. MPC returns an encrypted branch encoding v; HE evaluates the corresponding expression on the retained activation (Figure 2). **Selecting a branch for every activation would still require interaction on all Sd_f values.**

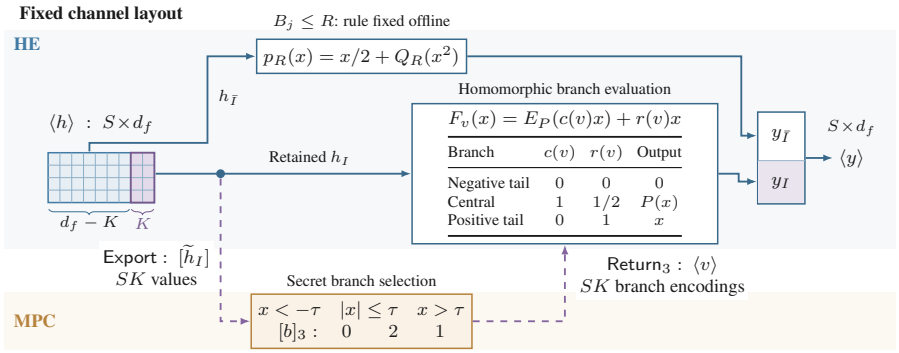

Figure 2: GELU control. Model bounds restrict branch selection to K padded channels; HE evaluates all activations on ciphertexts. Dashed arrows in Figures 2–4 denote protected exchanges.

(2) Eliminate comparisons using model bounds. For a normalized state satisfying $\|z\|_2 \le g\sqrt{d}$, the subsequent FFN channel j obeys $|h_j| \le B_j := |\mu_j| + g\sqrt{d}\,\|a_j\|_2 + \eta_j$, where $a_j = \gamma \odot W_1[:, j]$, $\mu_j = \beta^\top W_1[:, j] + b_{1j}$, and η_j bounds numerical error. Appendix A.3 derives sufficient local conditions for these bounds. The server uses these bounds to assign channels to two groups before inference: **(i) $B_j \le R$: no online decision.** HE uses a common polynomial p_R on $[-R, R]$. **(ii) $B_j > R$: secret branch selection.** MPC determines the evaluation rule from the protected activation. Padding to public capacity K and a private channel permutation hide the selected channels' identities and count while preserving the FFN function.

(3) Apply controls to retained activations. MPC compares the selected activations with $-\tau, \tau$ and returns v. Writing $P(x) = x/2 + E_P(x)$, with E_P even and $E_P(0) = 0$, HE evaluates $F_v(x) = E_P(c(v)x) + r(v)x$ using the selectors in Figure 2. **In both tails, the selectors set the polynomial input to zero, keeping it inside the approximation interval.** At tail encodings, $\partial F_v(x)/\partial v = 0$, suppressing first-order error in the returned branch encoding. Appendix A.1 gives the construction and proof.

Interaction. MPC processes SK positions; all Sd_f activations remain in HE. Protocol 1 specifies the complete construction.

4.2 ATTENTION CONTROL

(1) Separate amplitude from normalized weights. We use two controls per attention row: a shared amplitude and a reciprocal seed. **A common amplitude can bound intermediate magnitudes without changing normalized weights.** For scores s, private validity mask m, and public radius B, let μ be the mean of the valid scores and $A_j = m_j q((s_j - \mu)/B)$, where $q = p/p(0)$ and $p(u) \approx e^{Bu/8}$ on $[-1, 1]$. For a shared amplitude $\rho > 0$, set $E_j = (\rho A_j)^8$ and $P_j = \frac{E_j}{\sum_\ell E_\ell} = \frac{A_j^8}{\sum_\ell A_\ell^8}$. Centering and the common amplitude preserve the Softmax temperature; ρ bounds the energy before the eighth powers (Figure 3).

(2) Generate controls from row statistics. HE computes and exports protected row statistics in two sequential feedback stages; MPC uses leading-bit selection: **(i) Shared amplitude.** From $\widetilde{D}_1$, an estimate of $D_1 = \sum_j A_j^2$, MPC selects and returns encrypted ρ to bound intermediate energy before the eighth powers. **(ii) Reciprocal seed.** From $\widetilde{D}$, an estimate of $D = \sum_j E_j$, MPC selects

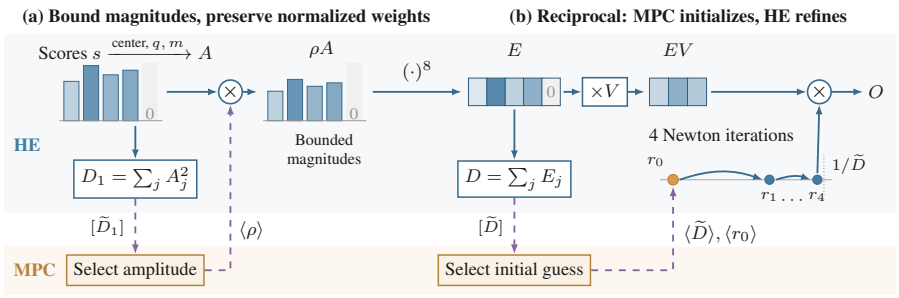

Figure 3: Attention control. (a) A shared amplitude bounds intermediate magnitudes without changing normalized weights. (b) MPC selects an initial reciprocal estimate; HE refines it and normalizes EV. Values are schematic; gray entries are masked.

$r_0 \approx 1/\widetilde{D}$ and returns encryptions of $(\widetilde{D}, r_0)$ for HE refinement. The mask makes invalid entries zero in both statistics and in the weighted-value computation.

(3) Complete attention in HE. HE inserts ρ into the retained polynomial factors, computes EV, and performs four Newton iterations $r_{i+1} = r_i(2 - \widetilde{D} r_i)$. The output is $O = \mathrm{diag}(r_4)(EV)$. Within the specified approximation interval, the normalized polynomial error bound is independent of ρ and the number of valid keys (Appendix A.2). Section 5 describes factor reuse and feedback scheduling.

Interaction. The two feedback stages exchange $O(HS)$ logical values while HS^2 scores remain in HE. Protocol 2 gives the complete sequence.

4.3 LayerNorm control

(1) Separate scale selection from inverse-square-root evaluation. LayerNorm requires one inverse square root per row, but MPC evaluation requires high-precision secret arithmetic. For a centered row C and guarded variance w, a secret power-of-two scale a gives $u = a^2 w \in [1/4, 1)$ and $w^{-1/2} = a\, u^{-1/2}$. **MPC selects the scale; HE approximates the inverse square root on a fixed interval** (Figure 4).

(2) Reduce variances to one interval. HE centers each input row to obtain C, computes $v = \|C\|_2^2/d$, and exports its protected estimate $\widetilde{v}$. MPC forms $w = \max(\widetilde{v}, 0) + \varepsilon$, with public variance guard ε, then uses leading-bit selection and secret shifts to obtain a and a quantized copy $\bar{u}$ of $u = a^2 w$. Both are returned as ciphertexts. The same polynomial can therefore be used across rows, layers, and hidden widths.

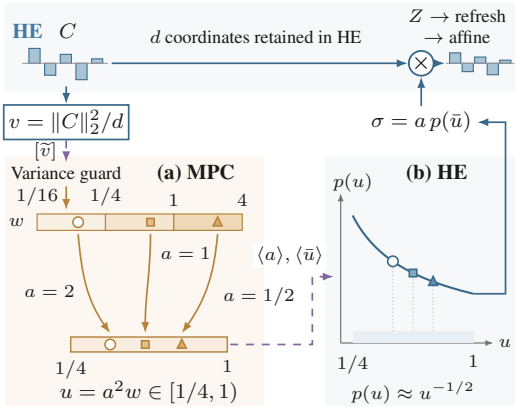

Figure 4: LayerNorm control. (a) MPC maps variances to one interval. (b) HE uses one polynomial to normalize retained rows. Markers denote illustrative inputs; the curve shows the inverse-square-root target.

(3) Normalize the retained row. HE computes one factor $\sigma = a p(\bar{u})$, where $p(u) \approx u^{-1/2}$, and applies it to all coordinates: $Z = \sigma C$. **The resulting norm bound also supports the GELU channel analysis in Section 4.1.** Appendix A.3 derives relative error and norm bounds. When a refresh is needed, GPU bootstrapping produces Z^+ before the learned affine transform $Y = Z^+ \odot \gamma + \beta$ (Section 5.3).

Interaction. Each row exports one variance and receives $(a, \bar{u})$; its d coordinates remain in HE. Protocol 3 gives the complete construction.

5 HOMOMORPHIC EXECUTION WITH STRUCTURED CONTROLS

We reduce feedback and HE computation by reusing control calculations and transmitting only the coefficients needed after projection. We schedule control application and state refresh to overlap independent work and preserve the depth needed by subsequent layers.

5.1 SHARED REPRESENTATIONS FOR DATA AND CONTROL

(1) Extract statistics in the data layout. The same encrypted matrix E is used for both EV and $D = \mathsf{RowSum}(E)$ (Figure 5). We align heads and rows so that a rotate-and-sum reduction produces repeated row statistics in the layout used by matrix computation. This avoids extracting each head and repacking its row statistics. The amplitude statistic D_1 uses the same reduction.

(2) Refine once, then reuse. Pointwise evaluation preserves slot repetition. For controls $\mathbf{u}$, let Rep repeat each input across slots. If the polynomial coefficients follow the same pattern, $p(\mathrm{Rep}(\mathbf{u})) = \mathrm{Rep}(p(\mathbf{u}))$. We refine each shared control ciphertext once and reuse it across data blocks. In Figure 5, one reciprocal r normalizes every output block $O_k = \mathrm{diag}(r)U_k$, where $U_k = (EV)_k$. The LN normalization factor is also shared (Appendix A.4).

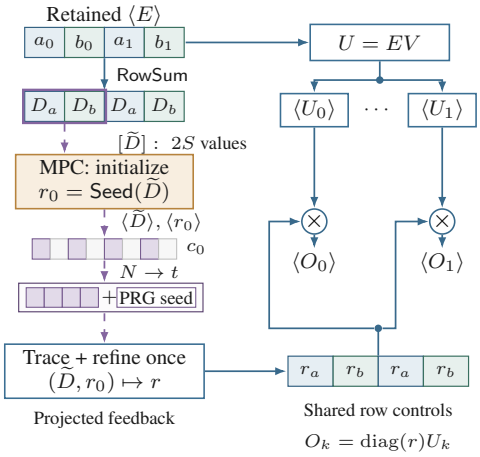

Figure 5: Shared state and compact feedback. Blue/green cells hold S rows per head; purple cells show selected coefficients of c_0, sent with a PRG seed for c_1.

5.2 COMPACT CONTROL FEEDBACK

Feedback need only preserve the ciphertext used after the prescribed HE projection. Repeated row controls use a periodic projection; real-valued GELU controls use a real projection. For the full return $c = (c_0, c_1)$ and recovered compact return $\widehat{c}$, both after local share addition, the homomorphic projection $\mathsf{EvalProj}$ satisfies $\mathsf{EvalProj}(\widehat{c}) = \mathsf{EvalProj}(c)$ under the same evaluation keys and coins. Thus subsequent HE computation receives identical ciphertexts (Appendix A.5). We use two formats: **(i) Periodic row controls: $N \to 2p$ coefficients.** For ring degree N and slot period p with $2p \mid N$ and $2p \geq 4$, we send $2p$ selected coefficients of c_0. After zero-filled recovery, a relative trace removes the omitted components. **(ii) Real GELU controls: $N \to N/2$ coefficients.** The real projection depends on $N/2$ coefficient combinations of c_0, recovered through a conjugate sum. Both formats send a PRG seed for c_1. **Feedback size depends on both coefficient count and refinement depth:** the return modulus must support projection and subsequent HE refinement. Protocol 4 gives the constructions; Table 1 lists coefficient reductions; Appendix C.5 reports byte costs.

5.3 CONTROL APPLICATION AND STATE REFRESH

(1) Delay control application. HE computation need only wait where it depends on feedback. For $R = \mathrm{diag}(r)$, delayed normalization (Dao, 2023) uses $(RE)V = R(EV)$ to move normalization from S^2 numerators to Sd_h output values per head. The product EV proceeds during reciprocal feedback and refinement; V projection proceeds during amplitude feedback (Figure 6). The row normalizer is applied before mixing heads. For GELU, HE evaluates channels covered by model bounds while MPC selects the remaining branches; their outputs join before FFN2.

(2) Reuse retained factors after feedback. While evaluating $q(u)$ for the amplitude statistic D_1, HE retains intermediate products from its factorization. Once ρ returns, HE multiplies it into a retained factor and completes the products and eighth powers to form E. Reusing these shallower ciphertexts leaves enough depth for EV without bootstrapping the score matrix. Appendix C.6 gives the factorization and schedule.

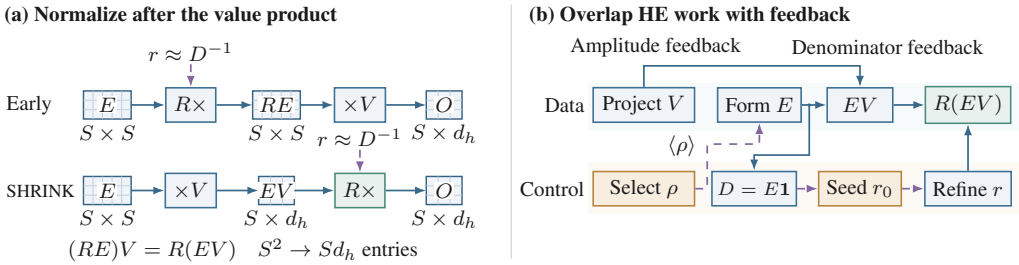

Figure 6: Control dependencies in HE. (a) Normalize after EV. (b) V projection and EV overlap their respective control exchanges.

(3) Refresh after normalization. At each nonterminal LN, GPU bootstrapping (Choi et al., 2025) refreshes $Z = \sigma C$ at hidden width, before the learned affine transform and expansion. Errors propagate as $\sigma\mathsf{Bootstrap}(C) = \sigma C + \sigma e_{\text{pre}}$ and $\mathsf{Bootstrap}(\sigma C) = \sigma C + e_{\text{post}}$. Refreshing Z avoids error amplification by σ. The terminal LN feeds the task head. Appendices A.3 and C.6 give bounds and implementation details.

6 Evaluation

We evaluate SHRINK on BERT-base and BERT-large using an NVIDIA A100 GPU. Our primary baselines are the hybrid HE–MPC frameworks IRON, BOLT, MPCFormer, BumbleBee, BLB, CipherPrune, and PCFormer (Hao et al., 2022; Pang et al., 2024; Li et al., 2023; Lu et al., 2025; Xu et al., 2025; Zhang et al., 2025b; Zeng et al., 2026); we also compare with the pure-HE systems THOR and Powerformer (Moon et al., 2025; Park et al., 2025).

6.1 End-to-end efficiency

At 128 tokens, SHRINK reduces communication by $8.4\times/11.0\times$ over BLB, with LAN latencies of 53/123 s for Base/Large (Table 3). WAN speedups over BLB range from 3.2 to $5.1\times$ (Figure 7(a)). Reducing wide-state traffic makes secure control extraction and encoding the dominant communication cost (Figure 7(b)). Communication scales linearly with sequence length while retaining dense attention (Figure 7(c)). Increasing the sequence length from 128 to 512 raises communication by only $3.55\times/3.79\times$ for Base/Large, despite a $16\times$ increase in the number of attention scores.

Homomorphic computation. SHRINK keeps rotation and ciphertext-multiplication counts comparable to BLB while reducing communication by 8.4–$11.0\times$. It requires 3–$7.5\times$ fewer bootstraps than the pure-HE systems NEXUS and THOR (Table 4).

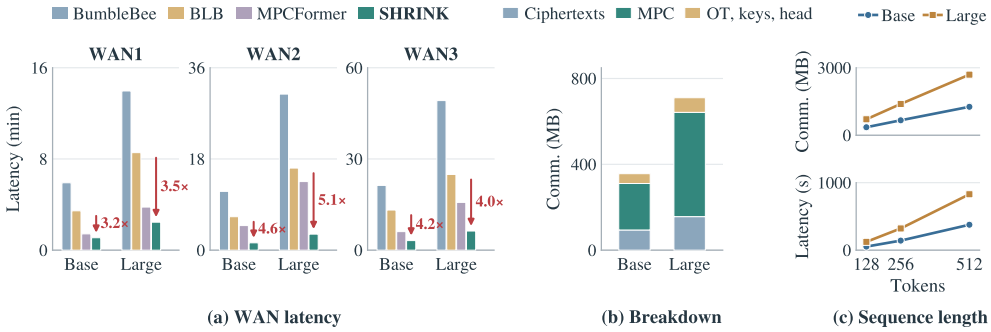

Figure 7: (a) WAN latency: WAN1, 400 Mbps and 4 ms RTT; WAN2, 100 Mbps and 4 ms RTT; WAN3, 100 Mbps and 80 ms RTT. Red arrows show speedup over BLB. (b) Communication breakdown (Appendix Table 12). (c) Sequence-length scaling with dense attention.

6.2 Ablation studies

Figure 8 shows complementary savings from selective control and state retention. State refresh, projected feedback, and HE LN refinement further cut traffic by 60.4%/65.3% for Base/Large. Protocol reductions exceed 96% for GELU and attention and reach 85.7%/88.2% for LN (Appendix C.5).

Table 4: HE operation counts (128 tokens).

Method	Paradigm	Bootstrapping	Rotation/ Conjugation	Ct×Ct
BERT-base				
NEXUS	HE-only	12	8,128†	8,192†
THOR	HE-only	25–30	2,059–2,077	1,409–1,443
Rho et al.	HE-only	≥ 60	–	–
BOLT	Hybrid	0	22,584	6,144
SHRINK	Hybrid	**4**	**2,731**	**2,056**
BERT-large				
BOLT	Hybrid	0	$\geq 28{,}634$	–
BLB	Hybrid	0	$\geq 1{,}696$	$\geq 3{,}072$
SHRINK	Hybrid	**4**	**2,848**	**2,729**

Per block; † NEXUS $QK^\top$ per head (total/12).

Table 3: Communication (GB) and LAN latency (s).

Method	BERT-base Comm.	BERT-base Latency	BERT-large Comm.	BERT-large Latency
IRON (NeurIPS'22)	281.0 (**788.6**×)	1088 (CPU)	744.8 (**1048.6**×)	2874 (CPU)
BOLT (S&P'24)	63.6 (**178.5**×)	930	158.9 (**223.7**×)	2628
BumbleBee (NDSS'25)	6.9 (**19.3**×)	153 (CPU)	17.6 (**24.7**×)	371 (CPU)
BLB (Security'25)	3.0 (**8.4**×)	150	7.8 (**11.0**×)	396
CipherPrune[a] (ICLR'25)	9.7 (**27.2**×)	79 (CPU)	18.4 (**25.9**×)	158 (CPU)
PCFormer[b] (AAAI'26)	7.1 (**19.9**×)	62 (CPU)	13.2 (**18.6**×)	111 (CPU)
THOR (CCS'25)	–	602	–	–
Powerformer (ACL'25)	–	345	–	–
SHRINK	**0.4** (**1.0**×)	**53**	**0.7** (**1.0**×)	**123**

[a] Token pruning and lower polynomial degrees. [b] Fine-tuned linear bypasses for selected attention tokens and FFN channels.

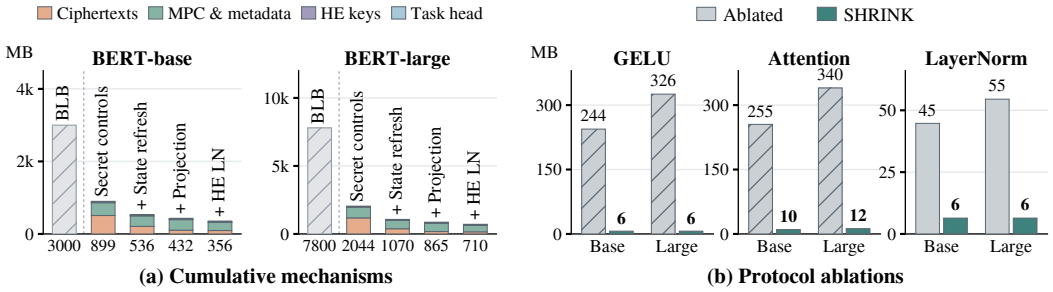

Figure 8: Ablation study.

6.3 Task accuracy

Table 5 reports accuracy from our encrypted executions for both model sizes (Wang et al., 2018). SHRINK retains the original weights without retraining. Relative to plaintext, the maximum accuracy drops are 0.10 and 0.09 percentage points for Base and Large, respectively, with no change on MRPC or RTE. These results show that the same approximation configuration maintains near-plaintext task accuracy as model size increases.

Table 5: Task accuracy (%).

Task	BERT-base Plain	BERT-base SHRINK	BERT-large Plain	BERT-large SHRINK
MRPC	88.24	88.24	87.99	87.99
SST-2	91.86	91.86	93.46	93.58
RTE	66.06	66.06	74.01	74.01
QNLI	91.58	91.56	92.24	92.22
CoLA	83.70	83.60	84.85	84.76

7 Conclusion

SHRINK decouples data and control in HE–MPC inference. Interaction supplies secret controls to retained ciphertexts. Model constraints and shared statistics reduce the control computation; joint representations and delayed control application make execution efficient. State refresh at normalized states sustains computation across Transformer blocks. The implementation supports BERT-base and BERT-large with common control rules and no retraining.

Reproducibility statement

Appendix C specifies the common model parameters and measurement conventions. The accompanying artifact records source and binary hashes and both parties' communication counters.

AI use statement

AI assistance was used in protocol analysis, prototype development, checking recorded results, and manuscript drafting and translation.

References

Jean-Philippe Bossuat, Rosario Cammarota, Ilaria Chillotti, Benjamin R. Curtis, Wei Dai, Huijing Gong, Erin Hales, Duhyeong Kim, Bryan Kumara, Changmin Lee, Xianhui Lu, Carsten Maple, Alberto Pedrouzo-Ulloa, Rachel Player, Yuriy Polyakov, Luis Antonio Ruiz Lopez, Yongsoo Song, and Donggeon Yhee. Security guidelines for implementing homomorphic encryption. *IACR Communications in Cryptology*, 2025. URL https://cic.iacr.org/p/1/4/26.

Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000. doi: 10.1007/s001459910006.

Wonhee Cho, Guillaume Hanrot, Taeseong Kim, Minje Park, and Damien Stehlé. Fast and accurate homomorphic Softmax evaluation. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4391–4404. Association for Computing Machinery, 2024. doi: 10.1145/3658644.3670369. URL https://doi.org/10.1145/3658644.3670369.

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*, 2025. doi: 10.1145/3760250.3762223. URL https://doi.org/10.1145/3760250.3762223.

Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. In *Selected Areas in Cryptography – SAC 2023*, volume 14201 of *Lecture Notes in Computer Science*, pp. 325–345. Springer, 2024. doi: 10.1007/978-3-031-53368-6_16. URL https://doi.org/10.1007/978-3-031-53368-6_16.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. arXiv:2307.08691, 2023. URL https://arxiv.org/abs/2307.08691.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15718–15731, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/64e2449d74f84e5b1a5c96ba7b3d308e-Abstract-Conference.html.

Zhicong Huang, Wen-jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 809–826. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric Xing, and Hao Zhang. MPCFormer: Fast, performant and private transformer inference with MPC. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.01452.

Zhengyi Li, Kang Yang, Jin Tan, Wen-jie Lu, Haoqi Wu, Xiao Wang, Yu Yu, Derun Zhao, Yancheng Zheng, Minyi Guo, and Jingwen Leng. Nimbus: Secure and efficient two-party inference for transformers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/264a9b3ce46abdf572dcfe0401141989-Abstract-Conference.html.

Cixin Liu. *Death's End*. Tor Books, 2016. ISBN 9780765377104. URL https://us.macmillan.com/books/9780765377104/deathsend/. Translated by Ken Liu.

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 *Network and Distributed System Security Symposium (NDSS)*, 2025.

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*, 2025. doi: 10.1145/3719027.3765150. URL https://eprint.iacr.org/2024/1881.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. BOLT: Privacy-preserving, accurate and efficient inference for transformers. In *45th IEEE Symposium on Security and Privacy*, 2024. URL https://eprint.iacr.org/2023/1893.

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, 2025. doi: 10.18653/v1/2025.acl-long.543. URL https://aclanthology.org/2025.acl-long.543/.

Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. CrypTFlow2: Practical 2-party secure inference. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020. URL https://eprint.iacr.org/2020/1002.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

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.

Bo Zeng, Zhi Pang, Yuyang Zhang, Kai Zhao, Tian Wu, Geying Yang, Lina Wang, and Run Wang. PCFormer: Accelerating privacy-preserving transformer inference by partition and combination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28076–28084, 2026. doi: 10.1609/aaai.v40i33.40033. URL https://ojs.aaai.org/index.php/AAAI/article/view/40033.

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 *Network and Distributed System Security Symposium (NDSS)*, 2025a. doi: 10.14722/ndss.2025.230868. URL https://www.ndss-symposium.org/ndss-paper/secure-transformer-inference-made-non-interactive/.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. CipherPrune: Efficient and scalable private transformer inference. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/83a14a36de4502bac5b580db36e81858-Paper-Conference.pdf.

A CORRECTNESS AND APPROXIMATION ANALYSIS

We establish the control identities, approximation bounds, and representation properties used in Sections 4 and 5. Concrete parameters appear in Appendix C.

Numerical specification. The numerical program $\widehat{f}_{\theta,W}(X, m; \xi)$ fixes the stored polynomials, fixed-point operations, control tables, and normalization guards. Here ξ denotes the secret randomness prescribed by the integer interfaces. This program provides the reference for the approximation bounds and numerical comparisons in this appendix. The ciphertext-level interface functionalities are defined separately in Appendix B.

A.1 GELU CONTROL

Protocol 1 GELU control Π_{GELU}

Input: $\mathcal{S}$: $\langle X \rangle$ (LayerNorm output), $W_1, b_1, W_2, b_2, \gamma, \beta$.
Output: $\mathcal{S}$: $\langle Y \rangle$, the FFN output.
Public parameters: $R, R_{\mathrm{ctl}}, K, \tau, g, \{\eta_j\}, p_R, E_P, c, r$; fixed positions I.
Offline
$\mathcal{S}$: $B_j \leftarrow |\beta^\top W_1[:, j] + b_{1j}| + g\sqrt{d}\,\|\gamma \odot W_1[:, j]\|_2 + \eta_j$.
$\mathcal{S}$: $J \leftarrow \{j : B_j > R\}$; $\mathsf{assert}(|J| \le K)$.
$\mathcal{S}$: $J \leftarrow \mathsf{Pad}_K(J)$; $\mathsf{assert}(\max_{j \in J} B_j \le R_{\mathrm{ctl}})$.
$\mathcal{S}$: $\Pi \leftarrow \mathsf{Permute}(J \mapsto I)$; $W_1' \leftarrow W_1\Pi$, $b_1' \leftarrow b_1\Pi$, $W_2' \leftarrow \Pi^\top W_2$.
Online
$\mathcal{S}$: $\langle h \rangle \leftarrow \langle XW_1' + b_1' \rangle$; $\langle y_{\bar{I}} \rangle \leftarrow \langle p_R(h_{\bar{I}}) \rangle$.
$\mathcal{S}, \mathcal{C}$: $[\widetilde{x}] \leftarrow \mathsf{Export}_I(\langle h \rangle)$.
$\mathcal{S}, \mathcal{C}$: $[b]_3 \leftarrow \mathcal{F}_{\mathrm{Branch},\tau}([\widetilde{x}])$.
$\mathcal{S}, \mathcal{C}$: $\langle v \rangle \leftarrow \mathsf{Return}_3([b]_3)$.
$\mathcal{S}$: $\langle y_I \rangle \leftarrow \langle E_P(c(v)h_I) + r(v)h_I \rangle$.
$\mathcal{S}$: $\langle Y \rangle \leftarrow \langle yW_2' + b_2 \rangle$.
return $\langle Y \rangle$.

Channel bounds and model preparation. For FFN channel j, write $h_j = \mu_j + z^\top a_j + e_j$, with $a_j = \gamma \odot W_1[:, j]$ and $\mu_j = \beta^\top W_1[:, j] + b_{1j}$. Suppressing the channel index, the conditions $\|z\|_2 \le g\sqrt{d}$ and $|e| \le \eta$ give

$$|h| \le |\mu| + g\sqrt{d}\,\|a\|_2 + \eta$$

by Cauchy–Schwarz and the triangle inequality. This proves the channel bound in Section 4.1. In Protocol 1, Pad_K fills the selected set to public capacity K, and $\mathsf{Permute}(J \mapsto I)$ places it at fixed positions I. The permutation acts jointly on weights, biases, and activation assignments; $\Pi\Pi^\top$ cancels before FFN2. Thus model preparation preserves the FFN function. Unused physical positions contain zero activation and the public central-branch control.

Even polynomial evaluation. The common approximation exploits the evenness of $\mathrm{GELU}(x) - x/2$: $p_R(x) = x/2 + Q_R(x^2)$. HE evaluates the even component in powers of x^2 and adds the linear term. For branch-controlled evaluation, write $P(x) = x/2 + E_P(x)$, where E_P is even and $E_P(0) = 0$, and set

$$F_v(x) = E_P(c(v)x) + r(v)x. \tag{A.1}$$

MPC compares the protected activation with $-\tau, \tau$ to determine the branch. The selectors below preserve the polynomial input in the central branch and set it to zero in both tails.

Branch selectors. Let canonical shares $a, b \in \{0, 1, 2\}$ encode the negative tail, positive tail, and center as residues $0, 1, 2$ modulo three. The ordinary sum gives $v = a + b - 2 \in \{-2, -1, 0, 1, 2\}$. Define

$$c(v) = \frac{(v^2 - 1)(v^2 - 4)}{4}, \qquad r(v) = \frac{1}{2} - \frac{v(v^2 - 5)(v^4 - v^2 - 4)}{32}. \tag{A.2}$$

Lemma A.1 (GELU branch evaluation)**.** *The expression $E_P(c(v)x) + r(v)x$ realizes the prescribed negative, central, and positive branches. Its derivative with respect to v vanishes at the tail encodings.*

Proof. At $v = -2, -1, 0, 1, 2$, the pairs $(c(v), r(v))$ are $(0,0), (0,1), (1,1/2), (0,0), (0,1)$. Since $P(x) = E_P(x) + x/2$ and $E_P(0) = 0$, the outputs are $0, x, P(x), 0, x$. The control derivative is $x\{E'_P(c(v)x)c'(v) + r'(v)\}$. At a tail encoding, $c(v) = 0$, $E'_P(0) = 0$, and $r'(v) = 0$. □

A.2 Attention control

Protocol 2 Attention control Π_{Attn}

Input: $\mathcal{S}$: $\langle s\rangle, \langle V\rangle, \langle m\rangle, \langle m/n\rangle$.
Output: $\mathcal{S}$: $\langle O\rangle$, the attention head output.
Public parameters: radius B, root polynomial q, power $r = 8$.
1: $\mathcal{S}$: $\langle \mu\rangle \leftarrow \mathsf{RowSum}(\langle (m/n) \odot s\rangle)$; $\langle u\rangle \leftarrow \langle (s-\mu)/B\rangle$.
2: $\mathcal{S}$: Evaluate reusable factors of $q(u)$; $\langle D_1\rangle \leftarrow \mathsf{RowSum}(\langle (m \odot q(u))^2\rangle)$.
3: $\mathcal{S}, \mathcal{C}$: $[\widetilde{D}_1] \leftarrow \mathsf{Export}(\langle D_1\rangle)$; $[\rho] \leftarrow \mathcal{F}_{\mathrm{Amp}}([\widetilde{D}_1])$.
4: $\mathcal{S}, \mathcal{C}$: $\langle \rho\rangle \leftarrow \mathsf{Return}([\rho])$.
5: $\mathcal{S}$: Insert $\langle m\rho\rangle$ into the retained factor tree; $\langle E\rangle \leftarrow \langle (m\rho \odot q(u))^8\rangle$; $\langle D\rangle \leftarrow \mathsf{RowSum}(\langle E\rangle)$.
6: $\mathcal{S}, \mathcal{C}$: $[\widetilde{D}] \leftarrow \mathsf{Export}(\langle D\rangle)$.
7: $\mathcal{S}$: $\langle U\rangle \leftarrow \langle EV\rangle$ ▷ parallel with lines 8–12
8: $\mathcal{S}, \mathcal{C}$: $[r_0] \leftarrow \mathcal{F}_{\mathrm{Seed}}([\widetilde{D}])$.
9: $\mathcal{S}, \mathcal{C}$: $(\langle \widetilde{D}\rangle, \langle r_0\rangle) \leftarrow \mathsf{Return}([\widetilde{D}], [r_0])$.
10: **for** $i = 0, \ldots, 3$ **do**
11: $\quad \mathcal{S}$: $\langle r_{i+1}\rangle \leftarrow \langle r_i(2 - \widetilde{D} r_i)\rangle$.
12: **end for**
13: $\mathcal{S}$: $\langle O\rangle \leftarrow \langle \mathrm{diag}(r_4)U\rangle$.
14: return $\langle O\rangle$.

Control statistics. Let $m_j \in \{0,1\}$, $n = \sum_j m_j \geq 1$, $\mu = \sum_j m_j s_j / n$, and $u_j = (s_j - \mu)/B$. With $q = p/p(0)$, set $A_j = m_j q(u_j)$ and $E_j = (\rho A_j)^8$. The two feedback stages export

$$D_1 = \sum_j A_j^2, \qquad D = \sum_j E_j. \tag{A.3}$$

The client supplies encrypted m and m/n once per request. The mask zeros invalid entries before either statistic and in EV. For public radius $B > 0$, define the approximation domain

$$\mathcal{D}_B = \left\{ (s, m) : m \in \{0,1\}^S,\ n \geq 1, \quad \max_{j : m_j = 1} |s_j - \mu| \leq B \right\}. \tag{A.4}$$

Common-amplitude invariance. For $m_j \in \{0,1\}$, $\rho > 0$, and $\sum_j m_j q_j^8 > 0$,

$$\frac{(m_j \rho q_j)^8}{\sum_\ell (m_\ell \rho q_\ell)^8} = \frac{m_j q_j^8}{\sum_\ell m_\ell q_\ell^8}.$$

This identity holds for every selected amplitude, including amplitudes computed from a randomized statistic. Thus amplitude selection leaves the normalized polynomial weights unchanged.

Root-polynomial construction. For $B = 64$ and $r = 8$, truncate the Chebyshev expansion $e^{8u} = I_0(8) + 2\sum_{k \geq 1} I_k(8) T_k(u)$ at degree 15, where $I_k(8) = \sum_{j \geq 0} 4^{2j+k}/(j!(j+k)!)$. Termwise comparison gives $I_{k+1}(8) \leq 4 I_k(8)/(k+1)$. Since $|T_k(u)| \leq 1$ on $[-1, 1]$, the uniform error is at most

$$\delta \leq \frac{2 I_{16}(8)}{1 - 4/17} + \epsilon_{\mathrm{coef}},$$

where ϵ_{coef} includes Q40 coefficient rounding and the stored factorization error. Rational bounds on the positive series certify the value in Appendix C; a monomial coefficient norm bounds the factorization residual on $[-1, 1]$.

Proposition A.2 (Normalized exponential approximation)**.** *Let $(s, m) \in \mathcal{D}_B$. If $\sup_{|u| \leq 1} |p(u) - e^{Bu/8}| \leq \delta < 1$, then the weights in Section 4.2 are well-defined and satisfy*

$$\|P - \mathrm{softmax}_m(s)\|_1 \leq 2\big((1+\delta)^8 - 1\big). \tag{A.5}$$

The bound is independent of ρ and the number of valid keys.

Proof. Restrict sums to valid entries and put $x = s - \mu$ and $Z = \sum_j e^{x_j}$. The valid entries of x have mean zero. For $0 \le t \le 1$, convexity gives $\sum_j e^{tx_j} \le (1-t)n + tZ \le Z$, since Jensen gives $Z \ge n$. The binomial theorem and the uniform error imply

$$\sum_j |p(x_j/B)^8 - e^{x_j}| \le \sum_{k=1}^{8} \binom{8}{k} \delta^k \sum_j e^{(8-k)x_j/8} \le \big((1+\delta)^8 - 1\big) Z.$$

At least one $x_j \ge 0$, so its polynomial value is positive and the denominator is nonzero. For nonnegative vectors a, b with positive sums A, B', the triangle inequality gives $\|a/A - b/B'\|_1 \le 2\|a-b\|_1/A$. Applying it to the exponential and polynomial weights proves the result, since $\mathrm{softmax}_m(s-\mu) = \mathrm{softmax}_m(s)$. □

The public normalization $q = p/p(0)$ cancels from the weights, and $x = s - \mu$ gives the original masked Softmax target.

Lemma A.3 (Amplitude range)**.** *Let $D_1 = \sum_j (m_j q_j)^2 \ge d_* > 0$, $0 \le \epsilon_1 < 1$, and $\Delta \ge 0$. Suppose $(1-\epsilon_1)D_1 \le \overline{D}_1 \le (1+\epsilon_1)D_1 + \Delta$. Set $k = \max(0, \lfloor \log_4 \overline{D}_1 \rfloor)$, $\rho = 6 \cdot 2^{-k}$, and $D = \sum_j (m_j \rho q_j)^8$. If $(1+\epsilon_1)d_* + \Delta \ge 1$, then*

$$\frac{1}{n^3} \left(\frac{36 d_*}{(1+\epsilon_1)d_* + \Delta} \right)^4 \le D \le \left(\frac{144}{1-\epsilon_1} \right)^4. \tag{A.6}$$

Proof. When $\overline{D}_1 \ge 1$, $4^k \le \overline{D}_1 < 4^{k+1}$ gives $36 d_* / \{(1+\epsilon_1)d_* + \Delta\} \le \rho^2 D_1 \le 144/(1-\epsilon_1)$. If $\overline{D}_1 < 1$, then $k = 0$ and $\rho^2 D_1 = 36 D_1$. The lower bound follows from $D_1 \ge d_*$ and $(1+\epsilon_1)d_* + \Delta \ge 1$; the upper bound follows from $(1-\epsilon_1)D_1 < 1$. For $b_j = (m_j \rho q_j)^2 \ge 0$, Hölder gives $(\sum_j b_j)^4 / n^3 \le \sum_j b_j^4 \le (\sum_j b_j)^4$. □

For the stored polynomial, reverse triangle inequality and Jensen give $D_1 \ge n(1-\delta)^2 / p(0)^2$. The controller adds $G = 1/16$ to its Q20 release. If the release obeys $|\widetilde{D}_1 - D_1| \le \epsilon_1 D_1 + \xi$ with $\xi \le G$, the lemma applies with $\Delta = G + \xi$. Equivalently, it selects the bin of $\lfloor 16 \widetilde{D}_1 \rfloor / 16 + 1/16$. Only powers of four are thresholds, so both comparisons give the same bin.

Private validity mask.

Proposition A.4 (Invariance to masked scores)**.** *Fix a binary mask with at least one valid key and the random tapes of the scalar functionalities. If $m \odot s = m \odot s'$, then the prescribed numerical computation produces identical μ, D_1, ρ, E, D and normalized weights.*

Proof. The valid means and valid polynomial arguments coincide. Both the statistic branch $mq(u)$ and the retained branch $m\rho q(u)$ are zero at invalid positions. Thus D_1 and its coupled control coincide, and so do E, D and the final weights. The valid values alone determine the attention output. □

For finite-precision execution, let $a_j = m_j \rho q(u_j)$ and let ξ_j include input, mask, control, and polynomial-evaluation errors. The final powers, including their fresh error η_j, satisfy

$$|\widehat{E}_j - E_j| \le (|a_j| + |\xi_j|)^8 - |a_j|^8 + |\eta_j|. \tag{A.7}$$

This follows by expanding the eighth power and using the triangle inequality; real projection cannot increase the modulus of the error. At an invalid position $a_j = 0$, the incoming root error contributes only $|\xi_j|^8$. The local HE contract bounds ξ_j, including the finite centered values at padding positions.

Reciprocal and numerical error. The seed $r_0 = (21/32)2^{-\lfloor \log_2 \widetilde{D} \rfloor}$ gives $21/32 \le \widetilde{D} r_0 < 21/16$ and hence $|1 - \widetilde{D} r_0| \le 11/32$. Starting from this seed, HE performs four Newton updates. In exact arithmetic, they satisfy

$$r_{i+1} = r_i(2 - \widetilde{D} r_i), \qquad |1 - \widetilde{D} r_i| \le (11/32)^{2^i}. \tag{A.8}$$

The recurrence squares the residual at each step. Suppose the evaluated numerator has aggregate error $\nu = \sum_j |\widehat{E}_j - E_j|$, the exported denominator satisfies $|\widetilde{D} - D| \le \eta < D$, and $|1 - \widetilde{D}\widehat{r}| \le \epsilon_{\mathrm{inv}}$. Then

$$\|\widehat{r}\widehat{E} - E/D\|_1 \le \frac{\nu + \eta}{D - \eta} + \epsilon_{\mathrm{inv}} \frac{D + \nu}{D - \eta}. \tag{A.9}$$

Add Eq. (A.5) to obtain the total error relative to $\mathrm{softmax}_m(s)$. The quantities ν, η include the HE, mask, aggregation, and protected-interface errors in their own units.

Deferred normalization. For diagonal R, each entry of both products in

$$(RE)V = R(EV) \tag{A.10}$$

equals $R_{ii} \sum_j E_{ij} V_{jk}$. Both products use the same masked E. Nonnegative E additionally gives $|(EV)_{ik}| \le D_i \max_j |V_{jk}|$, where $D = E\mathbf{1}$. For $\widehat{U} = EV + \Delta_U$, $\widehat{R} = R + \Delta_R$, and final multiplication error Δ_O, any submultiplicative norm gives

$$\|\widehat{R}\widehat{U} + \Delta_O - REV\| \le \|R\|\|\Delta_U\| + \|\Delta_R\|\|EV\| + \|\Delta_R\|\|\Delta_U\| + \|\Delta_O\|. \tag{A.11}$$

This follows by expanding the product and applying the triangle inequality. The row controls are applied before the projection mixing heads.

A.3 LayerNorm control

Protocol 3 LayerNorm control Π_{LN}

Input: $\mathcal{S}$: $\langle X \rangle, \gamma, \beta$.
Output: $\mathcal{S}$: $\langle Y \rangle$, the refreshed LayerNorm output.
Public parameters: d, variance guard ε from Eq. (A.16), p, grid h, layout $\mathcal{L}$.
1: $\mathcal{S}$: $\langle \mu \rangle \leftarrow d^{-1}\mathsf{RowSum}(\langle X \rangle)$; $\langle C \rangle \leftarrow \langle X - \mu \mathbf{1}^\top \rangle$.
2: $\mathcal{S}$: $\langle v \rangle \leftarrow d^{-1}\mathsf{RowSum}(\langle C \odot C \rangle)$.
3: $\mathcal{S}, \mathcal{C}$: $[\widetilde{v}] \leftarrow \mathsf{Export}(\langle v \rangle)$.
4: $\mathcal{S}, \mathcal{C}$: $[w] \leftarrow [\max(\widetilde{v}, 0) + \varepsilon]$.
5: $\mathcal{S}, \mathcal{C}$: $[k] \leftarrow [\lfloor \mathsf{msb}(w)/2 \rfloor]$; $[a] \leftarrow [2^{-k-1}]$; $[\bar{u}] \leftarrow [\lfloor a^2 w \rfloor_h]$.
6: $\mathcal{S}, \mathcal{C}$: $(\langle a \rangle, \langle \bar{u} \rangle) \leftarrow \mathsf{Return}([a], [\bar{u}])$.
7: $\mathcal{S}$: $\langle \sigma \rangle \leftarrow \langle ap(\bar{u}) \rangle$; $\langle Z \rangle \leftarrow \langle \sigma C \rangle$.
8: $\mathcal{S}$: $\langle Z^+ \rangle \leftarrow \mathsf{Adapt}_{\mathcal{L}}(\mathsf{Bootstrap}(\langle Z \rangle))$.
9: $\mathcal{S}$: $\langle Y \rangle \leftarrow \langle Z^+ \odot \gamma + \beta \rangle$.
10: return $\langle Y \rangle$.

Scale selection. For a centered row C, HE computes $v = \|C\|_2^2/d$ and exports $\widetilde{v}$. MPC forms $w = \max(\widetilde{v}, 0) + \varepsilon$ and selects

$$k = \left\lfloor \frac{\lfloor \log_2 w \rfloor}{2} \right\rfloor, \qquad a = 2^{-k-1}, \qquad u = a^2 w \in [1/4, 1). \tag{A.12}$$

Leading-bit selection and secret shifts implement this rule, with $\mathsf{msb}(w) = \lfloor \log_2 w \rfloor$. For the fixed-point interface, MPC returns encryptions of a and $\bar{u} = \lfloor u \rfloor_h$, where $\lfloor u \rfloor_h = h\lfloor u/h \rfloor$ on a binary grid containing $1/4$. HE evaluates $\sigma = ap(\bar{u})$ and $Z = \sigma C$.

Lemma A.5 (Scale reduction and relative error)**.** *For $w > 0$, Eq. (A.12) gives $u \in [1/4, 1)$ and $w^{-1/2} = au^{-1/2}$. If $\sup_{[1/4,1]} |p(u) - u^{-1/2}| \le \eta$ and $\bar{u}$ is rounded downward on a binary grid of spacing h containing $1/4$, then $|\sqrt{w}\, ap(\bar{u}) - 1| \le \eta + 2h$.*

Proof. For $k = \lfloor \log_2 w/2 \rfloor$, $2^{2k} \le w < 2^{2k+2}$ and $a^2 = 2^{-2k-2}$, giving the interval. Using $\sqrt{w}\, a = \sqrt{u} \le 1$ and $0 \le u - \bar{u} < h$, the error is at most $\eta + \sqrt{u/\bar{u}} - 1 \le \eta + (u - \bar{u})/(2\bar{u}) \le \eta + 2h$. □

Refinement polynomial. With $t = (8u - 5)/3$, the Legendre generating function yields

$$u^{-1/2} = \frac{4}{3}\sum_{j=0}^{\infty}(-1/3)^j P_j(t), \qquad p_m(u) = \frac{4}{3}\sum_{j=0}^{m}(-1/3)^j P_j(t). \tag{A.13}$$

Since $|P_j(t)| \le 1$ on $[-1, 1]$, the tail is bounded by $\eta_m := \frac{4}{3}\sum_{j>m} 3^{-j} = 2 \cdot 3^{-(m+1)}$. Combining the preceding lemma with the guard relation $w \ge \theta\|C\|_2^2/d$, for public $\theta \in (0, 1]$, gives

$$|\sqrt{w}\, ap_m(\bar{u}) - 1| \le \eta_m + 2h, \qquad \|C\, ap_m(\bar{u})\|_2 \le \frac{1 + \eta_m + 2h}{\sqrt{\theta}}\sqrt{d}. \tag{A.14}$$

Both BERT sizes use $m = 16$. The guard construction below and the layer invariant account for the additional HE, control-return, and refresh errors.

Lemma A.6 (Normalization from a shared source)**.** *Let $V \ge 0$ be the source energy and suppose $\|\widehat{c}\|_2/\sqrt{d} \le \lambda\sqrt{V} + \delta$ and $\widehat{v} \ge V - a_v\sqrt{V} - b_v$, with nonnegative error parameters. Choose $\theta > 0$ with $\theta\lambda^2 < 1$. If*

$$\varepsilon \ge b_v + \theta\delta^2 + \frac{(a_v + 2\theta\lambda\delta)^2}{4(1 - \theta\lambda^2)},$$

then $w := \max(\widehat{v}, 0) + \varepsilon \ge \theta\|\widehat{c}\|_2^2/d$. Consequently, $\sigma^2 w \le K_\sigma$ implies $\|\sigma\widehat{c}\|_2^2 \le (K_\sigma/\theta)d$.

Proof. For $t = \sqrt{V}$, subtracting $\theta\|\widehat{c}\|_2^2/d$ from $\widehat{v} + \varepsilon$ gives the lower bound $(1 - \theta\lambda^2)t^2 - (a_v + 2\theta\lambda\delta)t + \varepsilon - b_v - \theta\delta^2$. The stated condition makes its minimum nonnegative. Replacing $\widehat{v}$ by $\max(\widehat{v}, 0)$ and multiplying by σ^2 proves the result. □

The parameters include the residuals of the common source, variance computation, and export. The Hermitian variance product measures the energy of the same packed state used by the data path.

A guard for the retained state. Writing $V = \|C\|_2^2/d$ avoids charging errors already contained in C a second time. Suppose the variance producer and protected export satisfy

$$\widetilde{v} \ge \alpha V - a_v\sqrt{V} - b_v, \qquad \alpha > \theta > 0. \tag{A.15}$$

Here α includes public scale gains, a_v accounts for the cross term in the Hermitian product, and b_v includes additive arithmetic error, quantization, and the bounded interface noise. Choose

$$\varepsilon \ge b_v + \frac{a_v^2}{4(\alpha - \theta)}. \tag{A.16}$$

Then $w = \max(\widetilde{v}, 0) + \varepsilon \ge \theta V$. Indeed, the minimum over $t = \sqrt{V} \ge 0$ of $(\alpha - \theta)t^2 - a_v t + \varepsilon - b_v$ is nonnegative. The inequality includes arbitrarily small variances.

Normalized energy and channel bounds. For the stored degree-16 coefficients, exact rational Bernstein subdivision on $u \in [1/4, 1]$ gives $|up(u)^2 - 1| \le 2^{-25}$. Downward Q22 quantization therefore gives

$$w\{ap(\bar{u})\}^2 \le K_p := (1 + 2^{-20})(1 + 2^{-25}). \tag{A.17}$$

Put $\sigma_0 = ap(\bar{u})$ and let Z^+ be the refreshed state in the original units, after all ring and scale adapters and before the learned affine transform. Define the aggregate local error by

$$\delta_{\mathrm{LN}} = Z^+ - \sigma_0 C, \qquad \frac{\|\delta_{\mathrm{LN}}\|_2}{\sqrt{d}} \le \rho_{\mathrm{LN}}. \tag{A.18}$$

The refreshed state then obeys

$$\frac{\|Z^+\|_2}{\sqrt{d}} \le g_* := \sqrt{K_p/\theta} + \rho_{\mathrm{LN}}. \tag{A.19}$$

This follows from Eq. (A.16), Eq. (A.17), and the triangle inequality. Separate row-RMS bounds for application and refresh give $\rho_{\mathrm{LN}} \le \rho_{\mathrm{app}} + \rho_{\mathrm{boot}}$, while Eq. (A.18) bounds their combined effect directly. At each LN boundary, the reference energy is that of the current retained state.

For a subsequent affine transform and FFN1, let e_{aff} denote the error after the learned LN affine transform and e_{lin} the fresh FFN1 error. With the stored weights, one can take

$$\eta_j = \|W_1[:, j]\|_1 \|e_{\mathrm{aff}}\|_\infty + |e_{\mathrm{lin},j}|. \tag{A.20}$$

Combining this bound with $g \geq g_*$ gives the channel bounds in Section 4.1, used for allocation during model preparation. The refresh error belongs to g_*; the affine and projection errors belong to η_j.

Proposition A.7 (Local normalization and channel bounds)**.** *Let C be a retained row with $V = \|C\|_2^2/d$. Suppose its variance estimate satisfies Eq. (A.15), and choose ε according to Eq. (A.16). Let $\sigma_0 = ap(\bar{u})$ satisfy Eq. (A.17), and let Z^+ satisfy Eq. (A.18). Then*

$$\|Z^+\|_2/\sqrt{d} \leq g_* := \sqrt{K_p/\theta} + \rho_{\mathrm{LN}}.$$

For a following affine–FFN projection satisfying Eq. (A.20), any $g \geq g_*$ gives $|h_j| \leq |\mu_j| + g\sqrt{d}\|a_j\|_2 + \eta_j$.

Proof. The guard gives $w \geq \theta V$, while $w\sigma_0^2 \leq K_p$. Hence $\|\sigma_0 C\|_2/\sqrt{d} = |\sigma_0|\sqrt{V} \leq \sqrt{K_p/\theta}$. Adding the local residual proves the norm bound. Cauchy–Schwarz applied to $h_j = \mu_j + a_j^\top Z^+ + e_j$, with $|e_j| \leq \eta_j$, gives the channel bound. □

The reference energy is computed from the retained row itself. Errors inherited by that row enter V; the local budgets cover the variance computation, control application, refresh, and following projection.

Bootstrap input range. Let Z^- denote the state before refresh and suppose $\|Z^- - \sigma_0 C\|_2/\sqrt{d} \leq \rho_{\mathrm{app}}$. Put $g_- = \sqrt{K_p/\theta} + \rho_{\mathrm{app}}$. Pairing two real components gives $|Z_j^- + iZ_k^-| \leq \sqrt{2d}\, g_-$. Thus a public normalizer $B_z \geq 2\sqrt{2d}$ bounds each complex component of a degree-32,768 input carrier by $g_-/2$. The bootstrap input-domain condition is applied after the degree-doubling pack and includes the preceding adapter error. Its output residual includes the return adapters in Eq. (A.18). Recovering the original units multiplies the core error by B_z, as in Section 5.3.

A.4 Shared representation

We establish the layout invariants used for statistics extraction and control reuse, then prove that compact feedback preserves the ciphertext presented to its consumer. The latter result is used in the modular interface theorem of Appendix B.

Let a complete slot orbit contain $C = FH_gS$ positions. For $0 \leq h < H_g$, $0 \leq f < F$, and $0 \leq r < S$, put $\mathcal{L}(h, f, r) = (fH_g + h)S + r$ and $b = H_gS$. Rotations are taken modulo this complete orbit. Each of the F blocks holds S rows from H_g heads. A rotation by tb aligns corresponding blocks across heads. Since $\mathrm{Rot}_{(t-F)b} = \mathrm{Rot}_{tb}$ on this orbit, heads share the rotation preparation of the packed matrix kernels (Xu et al., 2025). Fixed feature permutations are absorbed into the projection weights.

Lemma A.8 (Indexing, reduction, and control reuse)**.** *The map $\mathcal{L}$ is a bijection onto $\{0, \ldots, C-1\}$. The row reduction satisfies*

$$\mathcal{T}(x) = \sum_{f=0}^{F-1} \mathrm{Rot}_{fb}(x), \qquad \mathrm{Rot}_b\, \mathcal{T}(x) = \mathcal{T}(x). \tag{A.21}$$

Pointwise polynomials preserve this period whenever their inputs and coefficient vectors have period b.

Proof. Division of a slot index by S recovers r and $fH_g + h$; further division by H_g recovers h and f uniquely. Since $C = Fb$, rotation by b permutes the F summands of $\mathcal{T}$. Finally, slot rotation commutes with pointwise addition and multiplication, so every polynomial term is fixed by Rot_b. □

Key blocks are aligned before reduction. A rotate-and-add tree computes the sum; one b-slot period then contains all row statistics for the head group. Additional carriers contribute aligned sums, and partial head groups use fixed padding in complete physical orbits. The amplitude statistic $D_1 = \mathsf{RowSum}((m \odot q(u))^2)$ uses the same reduction, with the masked root square evaluated before the row sum. The server exports the statistics through Export while retaining the matrix state for HE computation. These choices preserve its row–head correspondence with the exported statistics.

Shared refinement. For a public repetition map Rep, the same argument gives the identity in Section 5.1 when polynomial coefficients use the corresponding repeated layout. Thus one control carrier can supply the same refinement to several data blocks. This is an identity of the message computation; the ciphertext implementation uses its prescribed encoding and rescaling rules in the original key ring.

A.5 Projected control return

Protocol 4 implements the periodic and real projections below. Its $\mathsf{ShareEncode}$ operation uses the bounded integer lifting and local share encoding of Appendix A.6. Seeded encryption regenerates the uniform ciphertext component from a public seed and samples encryption error with independent private coins. $\mathsf{EvalProj}$ completes the trace or conjugate sum at the return modulus; Align applies the declared normalization, rescaling, and layout alignment.

Let N, t be powers of two with $4 \le t \le N$, $g = N/t$, and Q odd. Write $R_N = \mathbb{Z}_Q[X]/(X^N + 1)$ and $R_t = \mathbb{Z}_Q[Y]/(Y^t + 1)$. The embedding $\iota_g : R_t \to R_N$ maps Y to X^g; Fold_g extracts coefficients at $0, g, \ldots, (t-1)g$. The map ι_g is a ring embedding because $X^{gt} = -1$; Fold_g is the indicated linear coefficient map.

Lemma A.9 (Relative trace and coefficient selection)**.** *Let*

$$G = \{\sigma_j : X \mapsto X^{1+2tj} : 0 \le j < g\}.$$

Then G is the subgroup fixing $\iota_g(R_t)$, and its normalized trace satisfies

$$\mathsf{Fold}_g \iota_g = \mathrm{id}, \qquad \mathcal{P}_G := g^{-1} \sum_{\sigma \in G} \sigma = \iota_g \mathsf{Fold}_g. \tag{A.22}$$

Consequently $\mathcal{P}_G J_G T_G = \mathcal{P}_G$ for $T_G = \mathsf{Fold}_g$ and $J_G = \iota_g$.

Proof. The exponents $1 + 2tj$ are precisely the units congruent to one modulo $2t$; they form a subgroup of size g modulo $2N$ and fix X^g. The trace of a monomial is

$$X^k \sum_{j=0}^{g-1} X^{2tkj}.$$

If $g \mid k$, this equals gX^k. Otherwise let $d = \gcd(g, k)$. Since g/d is an even power of two, shifting the summation index by $g/(2d)$ multiplies the sum by $X^{Nk/d} = -1$. The sum is zero because two is invertible modulo Q. Thus the trace keeps exactly the coefficients whose indices are multiples of g. The scalar g is invertible modulo odd Q, which proves the second identity; the first follows directly from the coordinate maps. □

In the slot order indexed by powers of three, 3 has order $N/2$ modulo $2N$ and order $t/2$ modulo $2t$. Hence G is generated by $3^{t/2}$, and the embedded slot values have period $t/2$. The public row assignment and backend basis map place the controls on these orbits. A binary trace tree uses rotation offsets $N/4, N/8, \ldots, t/2$ when $g > 1$.

Lemma A.10 (Real coefficient projection)**.** *For $p(X) = \sum_{i=0}^{N-1} a_i X^i$, define*

$$T_{\mathrm{Re}}(p) = \big(a_0, (a_i - a_{N-i})_{1 \le i < N/2}\big), \qquad J_{\mathrm{Re}}(b) = \sum_{i=0}^{N/2-1} b_i X^i.$$

With $\bar{p} = p(X^{-1})$ and $\mathcal{P}_{\mathrm{Re}}(p) = (p + \bar{p})/2$, we have $\mathcal{P}_{\mathrm{Re}} J_{\mathrm{Re}} T_{\mathrm{Re}} = \mathcal{P}_{\mathrm{Re}}$.

Protocol 4 Projected control return Π_{ret}

Input: $\mathcal{S}, \mathcal{C}$: shares $[u]$; $\mathcal{C}$: key s; $\mathcal{S}$: evaluation keys.
Output: $\mathcal{S}$: the consumer-layout ciphertext of the encoded share sum, at level ℓ and scale Δ_*.
Public parameters: N, Q, Δ, share type, input scale and bound, layout $\mathcal{L}$, projection $\mathcal{P}$, maps $T_{\mathcal{P}}, J_{\mathcal{P}}$, target level ℓ and scale Δ_*.
1: $\mathcal{S}, \mathcal{C}$: $(m_{\mathcal{S}}, m_{\mathcal{C}}) \leftarrow \mathsf{ShareEncode}_{\mathcal{L},Q,\Delta}([u])$.
2: $\mathcal{C}$: sample a fresh public-component seed ρ and independent private encryption-error coins ω.
3: $\mathcal{C}$: $(c_0, c_1) \leftarrow \mathsf{Enc}_s^{\mathrm{seed}}(m_{\mathcal{C}}; \rho, \omega)$, with $c_1 = \mathsf{PRG}_{N,Q}(\rho)$.
4: $\mathcal{C} \to \mathcal{S}$: $(T_{\mathcal{P}}(c_0), \rho)$.
5: $\mathcal{S}$: $c_1 \leftarrow \mathsf{PRG}_{N,Q}(\rho)$; $\widehat{c}_0 \leftarrow J_{\mathcal{P}}(T_{\mathcal{P}}(c_0)) + m_{\mathcal{S}}$.
6: $\mathcal{S}$: $c' \leftarrow \mathsf{EvalProj}_{\mathcal{P}}(\widehat{c}_0, c_1)$, completing the automorphism sum at the return modulus.
7: $\mathcal{S}$: $\langle u \rangle \leftarrow \mathsf{Align}_{\mathcal{L},\ell,\Delta_*}(c')$, applying the declared normalization and rescaling.
8: return $\langle u \rangle$.

Proof. For $1 \le i < N$, $X^{-i} = -X^{N-i}$. Consequently

$$p + \bar{p} = 2a_0 + \sum_{i=1}^{N/2-1} (a_i - a_{N-i})(X^i - X^{N-i}). \tag{A.23}$$

The middle coefficient cancels, and the expression is determined by the $N/2$ transmitted coordinates. Applying the same expression to $J_{\mathrm{Re}} T_{\mathrm{Re}}(p)$ proves the identity. □

Lemma A.11 (Projected phase preservation)**.** *Let $(\mathcal{P}, T, J)$ be either construction above. For a full ciphertext $c = (c_0, c_1)$, any key s, and any server polynomial m_S, define*

$$c^+ = (c_0 + m_S, c_1), \qquad \widehat{c} = (JT(c_0) + m_S, c_1).$$

Then, for $\phi_s(c) = c_0 + c_1 s$, we have

$$\mathcal{P}\big(\phi_s(\widehat{c})\big) = \mathcal{P}\big(\phi_s(c^+)\big). \tag{A.24}$$

Proof. The phase difference is $d = JT(c_0) - c_0$. The preceding lemmas give $\mathcal{P}(d) = 0$. Both phases contain the identical $c_1 s$ and m_S terms. Linearity proves the claim for every ciphertext, independently of its message and encryption error. □

Homomorphic projector. The protocol evaluates an unnormalized trace or conjugate sum and accounts for its known factor through the scale. For the trace the factor is g; for the real projection it is two. The corresponding scale changes are $\Delta \mapsto g\Delta$ and $\Delta \mapsto 2\Delta$. A declared final rescale and layout alignment then prepare the consumer. The exact ring projector above specifies the retained coefficient subspace; this scale convention implements its numerical normalization.

The following implementation property connects phase preservation to continuing ciphertext computation. A key-switched automorphism has the form

$$\mathsf{KS}_\sigma(b, a) = \big(\sigma(b) + K_{\sigma,0}(a),\ K_{\sigma,1}(a)\big), \tag{A.25}$$

where $K_{\sigma,0}, K_{\sigma,1}$ depend on the second component, the fixed evaluation key, and any coupled fresh coins. The trace tree uses these automorphisms and addition at a fixed modulus; normalization and rescaling follow the completed sum.

Lemma A.12 (Ciphertext-level preservation)**.** *With Eq. (A.25), the homomorphic projector produces identical ciphertexts from c^+ and $\widehat{c}$ of Lemma A.11, under the same evaluation keys and coins. Their subsequent deterministic alignment outputs are identical.*

Proof. The input second components coincide, so every paired key switch has identical $K_{\sigma,0}$ and $K_{\sigma,1}$. If the first-component difference is d, an automorphism changes it to $\sigma(d)$. Each trace-tree step changes the difference to $(\mathrm{id} + \sigma)d$ and retains equal second components. The completed difference is $\sum_{\sigma \in G} \sigma(d) = g\mathcal{P}_G(d) = 0$. For the real projection it is $d + \bar{d} = 2\mathcal{P}_{\mathrm{Re}}(d) = 0$. Thus both ciphertext components coincide before normalization or rescaling. Applying the same alignment gives identical outputs. □

For an encoded share sum m and fresh encryption error e, the unnormalized traced phase is $\mathrm{Tr}_G(m+e)+e_{\mathrm{KS}}$. Lemma A.12 gives identical key-switching contributions in the full and compact returns. Under the prescribed projection and alignment, both formats therefore deliver the same ciphertext to the consumer.

Lemma A.13 (Security preservation of compact feedback)**.** *Compare the full seeded return (c_0,ρ) with the projected return $(T(c_0),\rho)$, using the same share encoding and fresh encryption. Suppose c_1 is regenerated exactly from ρ, encryption error uses independent private coins, and public metadata is unchanged. Under Lemma A.12, compact feedback preserves the consumer ciphertext and cannot increase the distinguishing advantage of the full-return protocol in the prescribed continuation.*

Proof. The compact message is a deterministic public function of the full message. The server reconstructs the same c_1 and adds the same local polynomial m_S. Lemma A.12 makes the post-projection ciphertext identical, so the continuation uses the same state and coins. A simulator or distinguisher for compact feedback is obtained by applying T to the full message and executing the compact recovery. The client's local view is generated from the same share and encryption coins. The argument therefore preserves the joint law with both parties' subsequent outputs and auxiliary state. □

All coefficient maps are applied separately in each transmitted RNS limb. The seed describes the fresh uniform component c_1, while the private encryption-error coins remain with the encrypting party. Public metadata specifies the modulus prefix, basis, layout, projection, and normalization rule. The message dimension changes; the key ring and evaluation-key distribution are those of the full-return protocol. This continuation-preserving message map is the compact-feedback step in Theorem B.6.

A.6 Conversion interfaces

Protocol 5 composes masked phase sharing, secure integer lifting, public integer decoding, and randomized quantization. Its decoder and public coordinate set determine the control input. The BFV batching below uses plaintext modulus q and carries the source NTT coordinates in its slots.

Lemma A.14 (Phase-sharing correctness)**.** *If the BFV noise bound at every return modulus Q_i satisfies $B_i < Q_i/(2q)$, Protocol 5 recovers $x_{\mathcal{C}} = \phi - R$ and $x_{\mathcal{S}} = R$ in R_q.*

Proof. The NTT is a ring isomorphism to coordinatewise multiplication, so the evaluated BFV plaintext is $\mathrm{NTT}_q(\phi - R)$. The decryption bound ensures its exact recovery. If a prime p_i is dropped, componentwise nearest-integer rounding gives the noise contract

$$B_{i+1} \le B_i/p_i + (1 + \|s_B\|_1)/2.$$

The initial bound includes the evaluation, rerandomization, and flooding noise. After the prescribed modulus switches, inverse NTT recovers the stated shares. □

The secure lifting operation fixes the integer interpretation before local decoding. The public matrix A is evaluated without intermediate share-local truncation; its integer scale is handled by the joint quantizer. Appendix B.2 defines $\mathcal{F}_{\mathrm{int}}$, specifies its stateful share allocation, and proves its security.

Share encoding. For Protocol 4, numeric shares $x_{\mathcal{S}}, x_{\mathcal{C}} \in [0, M)$, $M = 2^k$, reconstruct an integer v with $|v| < M/4$; the real value is $2^{-f}v$. Define

$$\alpha = \mathbf{1}\{x_{\mathcal{S}} < M/4 \text{ or } x_{\mathcal{S}} \ge 3M/4\}, \quad \gamma = \mathbf{1}\{x_{\mathcal{S}} < M/4\}, \quad \beta = \mathbf{1}\{x_{\mathcal{C}} \ge M/2\}.$$

A no-wrap correlated OT returns R to $\mathcal{S}$ and $R+\alpha\beta M$ to $\mathcal{C}$, where R is uniform on a public interval of length A. The working representation accommodates that interval and the additional offsets. The widened shares are

$$\begin{aligned} \widetilde{x}_{\mathcal{S}} &= x_{\mathcal{S}} - (1-\gamma)M + R, \\ \widetilde{x}_{\mathcal{C}} &= x_{\mathcal{C}} - R - \alpha\beta M. \end{aligned} \tag{A.26}$$

Protocol 5 Protected export Π_{export}

Input: $\mathcal{S}$: (a_0, a_1) and $e_s = \mathrm{Enc}_{\mathrm{BFV}}(\mathrm{NTT}_q(s))$; $\mathcal{C}$: source CKKS key s and independent BFV key s_B.
Output: $\mathcal{S}, \mathcal{C}$: shares of $q_f(P_I A \,\mathrm{rep}_q(\phi) + Z)$, where $\phi = a_0 + a_1 s \bmod q$.
Public parameters: phase ring R_q, integer interpretation rep_q, working ring $\mathbb{Z}_{2^\ell}$, integer decoder A, coordinates I, quantizer q_f, noise law, and lifting bounds.
1: $\mathcal{S}$: $R \leftarrow_{\$} R_q$.
2: $\mathcal{S}$: $c \leftarrow \mathsf{Eval}_{\mathrm{BFV}}(\mathrm{NTT}_q(a_0 - R) + \mathrm{NTT}_q(a_1) \odot e_s)$.
3: $\mathcal{S}$: $c' \leftarrow \mathsf{ModSwitch}(\mathsf{Flood}(\mathsf{ReRand}(c)))$.
4: $\mathcal{S} \to \mathcal{C}$: c'.
5: $\mathcal{C}$: $x_{\mathcal{C}} \leftarrow \mathrm{NTT}_q^{-1}(\mathsf{Dec}_{s_B}(c'))$; $\mathcal{S}$: $x_{\mathcal{S}} \leftarrow R$.
6: $\mathcal{S}, \mathcal{C}$: $[v] \leftarrow \mathsf{Lift}_q(x_{\mathcal{C}}, x_{\mathcal{S}})$, with $v = \mathrm{rep}_q(\phi)$ in the working ring.
7: Each party P locally computes $[w]_P \leftarrow P_I A [v]_P \pmod{2^\ell}$.
8: $\mathcal{S}, \mathcal{C}$: $[y] \leftarrow \mathcal{F}_{\mathrm{int}}([w]_{\mathcal{C}}, [w]_{\mathcal{S}})$.
9: return $[y]$.

Their ordinary integer sum is v. Lemma B.5 proves this wrap correction and bounds its statistical sharing cost by M/A per conversion.

Each widened share is repeated over its public control orbit and encoded by the declared fixed-point inverse canonical embedding. The resulting local polynomials satisfy

$$m_{\mathcal{S}} + m_{\mathcal{C}} = \mathsf{Encode}_{\mathcal{L},\Delta}(2^{-f} v) + \delta_{\mathrm{enc}} \pmod{Q}, \tag{A.27}$$

where δ_{enc} is the error of these local transforms and roundings. The full and projected return use the same local polynomials. Public scaling reserves the precision required by projection and the final rescale. For Return_3, the encoded local integers are instead $b_{\mathcal{S}} - 1, b_{\mathcal{C}} - 1$ from canonical ternary shares; their sum is the five-valued selector input in Appendix A.1.

Seeded encryption. The public seed ρ regenerates only the uniform component c_1. Independent private coins ω sample the encryption error, giving $c_0 = m_{\mathcal{C}} - c_1 s + e_{\mathrm{enc}}(\omega) \pmod{Q}$. The return sends ρ together with the selected c_0 coordinates. The homomorphic projector completes its automorphism sum before the normalization and rescaling in Align.

B Security of protected HE–MPC interfaces

This appendix proves simulation-based realization of the conversion and control functionalities defined below, together with their composition in a retained HE execution. The reference execution idealizes these interfaces and preserves the surrounding HE computation. Compact feedback is analyzed by a separate preservation argument: after the prescribed projection, it delivers the same ciphertext as the corresponding full return.

B.1 Model and interface functionalities

The client C holds $u = (X, m)$, the server S holds weights W, and the client receives the task output. Let θ denote the protocol parameters. We consider one static semi-honest corruption. A session begins with fresh setup and has a public bound on interface invocations. Requests and interface inputs may depend on prior protocol history. Keys and local state persist within the session; preprocessing items are used once. A view contains the party's input, setup, random coins, messages, and local interface outputs.

Interface outputs and public metadata. The task head returns Q24 logits to the client, which computes the label locally. The server receives no task output. Public metadata consists of the architecture and task dimensions, padded sequence length, cryptographic parameters, fixed control capacities, model-preparation acceptance, and session termination status. The valid-token count and exception-channel identities are not public metadata. Intermediate values and selected controls are carried as shares or ciphertexts rather than opened by these interfaces. The public invocation budget counts all requests, including warmup; adaptive requests share the same cumulative budget.

Definition B.1 (Interface realization)**.** For an interface functionality $\mathcal{F}$, let a_P be party P's local input, including its setup and retained state. A protocol Γ realizes $\mathcal{F}$ with error ϵ if a PPT online simulator $\mathcal{S}_P$ exists such that, for either party and all admitted inputs and auxiliary information z,

$$\begin{aligned}&(\mathsf{View}_P^{\Gamma}, \mathsf{Out}^{\Gamma}, z)\\ &\quad\approx_{\epsilon}^{c} (\mathsf{View}_P^{\mathcal{S}_P \leftrightarrow \mathcal{F}}, \mathsf{Out}^{\mathcal{S}_P \leftrightarrow \mathcal{F}}, z).\end{aligned} \tag{B.1}$$

The simulator receives a_P and the party's ideal interface outputs; both parties' outputs and continuation states belong to the same ideal execution. For interfaces with shared outputs, the allocation rule below allows the simulator to generate its share and local records jointly. Negligible realization error gives the corresponding asymptotic guarantee.

We write SD for statistical distance. Each computational term below denotes a distinguishing advantage for the specified primitive and invocation budget.

Primitive assumptions. Comparison, carry extraction, correlated OT, and share conversion provide semi-honest simulation for the prescribed batches and preprocessing schedule (Rathee et al., 2020; Huang et al., 2022), including retained preprocessing state. New masks use unused randomness; reused values retain their original shares. Private PRG streams have independent seeds and use each position once. Independent BFV setup keys encrypt the required CKKS secret material. The BFV return uses the fresh-secret Ring-LWE assumption stated below. CKKS encryption uses the declared distribution of relinearization, automorphism, ring-switching, and bootstrap keys. Public batching fixes the order of interactive calls; HE-only work may overlap a call without changing its inputs or messages.

Share allocation. The integer and control functionalities provide an allocation port for new shared outputs. Before learning the reconstructed value, the simulator supplies the corrupted party's share, generated from its local input and retained state by the specified protocol's local sampling rule. The functionality assigns the complementary share to the honest party. The simulator keeps the records used to generate this allocation; reused values keep the same share handle. This rule specifies the representation of the output while preserving its numerical value.

Phase sharing. The server samples the mask R prescribed by the protocol and invokes $\mathcal{F}_{\mathrm{phase}}$ with (ct, R), where $ct = (a_0, a_1) \in R_q^2$. The client supplies the associated key s. The functionality computes

$$\phi = a_0 + a_1 s \pmod q$$

and gives R to S and $\phi - R$ to C. Honest mask generation uses $R \leftarrow_{\$} R_q$ (or its private-PRG implementation). Keeping mask generation in the caller preserves its random tape across invocations. The setup fixes the public packing map used by the BFV bridge.

Protected export. Let rep_q be the declared coefficientwise integer interpretation of R_q, A the public integer decoder, and P_I the selection of public coordinates. The functionality $\mathcal{F}_{\mathrm{export}}$ computes

$$w = P_I A \,\mathrm{rep}_q(\phi) \pmod{2^\ell}, \qquad y = q_f(w + Z), \tag{B.2}$$

and returns additive shares of y under the stated allocation rule. Here q_f and the law of Z are specified in Eq. (B.5). Secure lifting realizes rep_q in the working ring before applying A. The public representation bounds ensure that lifting has the specified integer meaning. This interface takes its numerical input from the actual ciphertext phase.

Control computation. For each control call, $\mathcal{F}_{\mathrm{ctrl},j}$ reconstructs its input from the supplied shares, evaluates the fixed integer circuit of Section 4, and shares its complete output tuple under the same allocation rule. The tuple includes any numerical value used with the control, such as the attention denominator. A derived precision view is computed from the same protected value by the specified secure integer operation. For ternary controls, allocation uses the declared ternary ring.

Task-head interfaces. The terminal CLS state is exported to shares through the protected phase and integer interfaces. The head then evaluates integer LayerNorm, a BFV pooler affine map, and the MPC tanh and classifier circuits. The pooler functionality returns additive shares of its fixed-point affine result; its BFV implementation uses a masked, rerandomized, and flooded return of the form in Eq. (B.3). Only the classifier logits are reconstructed to the client.

B.2 PHASE-SHARING AND INTEGER PROTOCOLS

Masked BFV return. Before public return processing, the BFV ciphertext for the client's share $a = \phi - R$ satisfies

$$c_0 + s_B c_1 = \iota_B(a) + \nu + F \pmod{Q}, \qquad \|\nu\|_\infty \le E_B. \tag{B.3}$$

Here ι_B includes the public BFV packing and plaintext scaling; F has independent uniform integer coefficients in intervals of length L_B. The residual ν is determined before F is sampled and may depend on the preceding state and rerandomization coins. Fresh zero encryption gives $c_1 = h_1 + a_B t + e_1$. Let ζ be the public-polynomial descriptor, including any seed, and let st contain the preceding receiver state, including its BFV key. Setup and this state are generated from ζ and independent coins before the fresh t, e_1. We use the fresh-secret Ring-LWE assumption

$$(\zeta, \mathsf{st}, a_B t + e_1) \approx^c_{\epsilon_{\mathrm{rz}}} (\zeta, \mathsf{st}, U), \qquad U \leftarrow_{\$} R_Q.$$

Lemma B.2 (Phase-return simulation)**.** *Under this assumption and the decoding contract of Lemma A.14, the BFV bridge realizes $\mathcal{F}_{\mathrm{phase}}$ with error at most*

$$\epsilon_{\mathrm{phase}} \le n_B E_B / L_B + \epsilon_{\mathrm{rz}} + \beta_{\mathrm{dec}},$$

where n_B is the BFV ring dimension and β_{dec} is the sum of the real and simulated return-decoding failure bounds.

Proof. For a corrupted client, condition on the preceding state and the rerandomization sample. A uniform interval and its integer translate by v have distance at most $|v|/L_B$. Replacing $\nu + F$ by F therefore costs at most $n_B E_B / L_B$, including the correlation of ν with c_1. Next replace c_1 by uniform U. The Ring-LWE reduction generates the auxiliary setup and translates its challenge by the fixed h_1. Given the ideal share a and the client's setup, the simulator samples U, F and forms

$$\widetilde{c}_1 = U, \qquad \widetilde{c}_0 = \iota_B(a) + F - s_B U \pmod{Q}.$$

It applies the prescribed public return processing. Successful decoding yields a in both executions, and the same share can be used throughout the continuation. The failure probabilities contribute β_{dec}.

For a corrupted server, R is part of its local interface input. Given its input ciphertext, mask, setup ciphertexts, and caller state, the simulator runs the public BFV evaluation with fresh server coins. All messages in this call are then determined by its local input and coins. Thus the simulation preserves both phase shares jointly. □

Fresh masks, zero encryptions, and floods are used at every call. Let $\epsilon_{\mathrm{bridge}} = \sum_j \epsilon_{\mathrm{phase},j}$.

Randomized integer interface. For $B = 2^b$ and $k \ge 3$, define

$$D_{k,B}: \quad Z = \sum_{a=1}^{k} U_a - \lfloor k(B-1)/2 \rfloor, \qquad U_a \leftarrow_{\$} \{0, \dots, B-1\}. \tag{B.4}$$

The functionality $\mathcal{F}_{\mathrm{int}}$ receives $x_C, x_S \in \mathbb{Z}_{2^\ell}$, samples Z, and shares

$$y = q_f(x_C + x_S + Z), \qquad q_f(t) = \left\lfloor \frac{t \bmod 2^\ell}{2^f} \right\rfloor, \quad 1 \le f \le b < \ell, \tag{B.5}$$

in $\mathbb{Z}_{2^{\ell-f}}$ through its allocation port. Distinct vector coordinates use independent noise.

Each party i samples $r_{i,a} \in \mathbb{Z}_B$ and computes $a_i = x_i + \sum_a r_{i,a} - c_i \bmod 2^\ell$, with public $c_C + c_S = \lfloor k(B-1)/2 \rfloor$. One batch of carry calls produces arithmetic shares of

$$d_a = \mathbf{1}\{r_{C,a} + r_{S,a} \ge B\}, \qquad \kappa = \mathbf{1}\{(a_C \bmod 2^f) + (a_S \bmod 2^f) \ge 2^f\}.$$

The output is

$$y_i = \lfloor a_i / 2^f \rfloor + [\kappa]_i - 2^{b-f} \sum_a [d_a]_i \pmod{2^{\ell-f}}. \tag{B.6}$$

Here $[d_a]_i$ and $[\kappa]_i$ are arithmetic shares in the output ring, obtained through the specified share-conversion primitive.

Lemma B.3 (Stateful share generation)**.** *In the ideal-OT/PRG hybrid, the masked comparison and COT-based share-conversion protocols generate each party's new records and output share from its local input and retained preprocessing state, with a distribution independent of the other input and the reconstructed result.*

Proof. Independent masks hide the selected entries at comparison leaves. For each Boolean AND, unused triple masks give the opened differences a local distribution independent of the other input; XOR gates are local. These distributions compose through the comparison circuit.

For B2A, let α, β be the Boolean input shares. With COT correlation -2α and mask a, the arithmetic shares are

$$r_S = \alpha - a, \qquad r_C = \beta + a - 2\alpha\beta.$$

Fix the sender's retained random-OT pads h_0, h_1. The receiver sends $b = \beta \oplus \tau$, where its hidden random-OT choice τ is uniform. Thus $a = h_b$ and the sender's entire local record have a law independent of β. For the receiver with fixed β, τ, h_τ, the unselected pad masks the correction $w = -2\alpha + h_\tau + h_{1-\tau}$. Its arithmetic share is h_τ when $\beta = 0$ and $1 + w - h_\tau$ otherwise, independently of α. The argument preserves the retained pads and their relationship to the new share. Applying it at each unused preprocessing position gives the claimed local sampling rule. □

Theorem B.4 (Integer-interface security)**.** *With secure carry and share-conversion primitives, Eq.* (B.6) *realizes* $\mathcal{F}_{\mathrm{int}}$ *with its stateful allocation port. The online simulation preserves both output shares and the caller's continuation state jointly.*

Proof. Put $U_a = (r_{C,a} + r_{S,a}) \bmod B$. These variables are independent uniforms and $r_{C,a} + r_{S,a} = U_a + Bd_a$. Moreover,

$$\left\lfloor \frac{a_C + a_S}{2^f} \right\rfloor = \lfloor a_C/2^f \rfloor + \lfloor a_S/2^f \rfloor + \kappa.$$

Subtracting $2^{b-f} \sum_a d_a$ gives the ideal result modulo $2^{\ell-f}$. Any wrap by 2^ℓ disappears in this output ring.

For the simulation, fix the preceding local state σ_P and input x_P. Conditional on any local sampler words, the honest words make the U_a independent uniforms. Lemma B.3 generates the sampler-carry and final-carry records without reading the other party's inputs. Let T_P collect these records and local words. Since Eq. (B.6) determines y_P locally, their joint distribution has the form

$$\Pr[T_P = t, y_P = r, Z = z \mid \sigma_P, x_P] = \nu_{\sigma_P, x_P}(t, r) \Pr[Z = z].$$

The simulator samples this local law forward, submits r to the allocation port, and retains the corresponding records and updated preprocessing indices. The functionality samples Z, computes y, and assigns $y - r$ to the honest party. Algebraic correctness and the displayed factorization establish the joint distribution of records and both outputs. Existing pads, shares, and random coins remain fixed. Replacing ideal primitives and honest private PRG streams adds their respective realization losses. □

Different coordinate precisions can share a carry batch: multiply the low-f-bit comparison operands by 2^{b-f} and compare at width b. If arithmetic carry shares use a larger common ring, the local sampling rule includes the full share-conversion record; reduction retains its required low bits. Several views of the same coordinate are obtained through secure post-processing of its finest protected view. Public coordinate selection commutes with ring addition and coordinatewise quantization.

Bounded lifting for control return. The numeric return in Appendix A.6 uses $M = 2^k$, canonical shares $x_S, x_C \in [0, M)$, and their signed reconstruction v with $|v| < M/4$. Define

$$\alpha = \mathbf{1}\{x_S < M/4 \text{ or } x_S \geq 3M/4\}, \quad \gamma = \mathbf{1}\{x_S < M/4\}, \quad \beta = \mathbf{1}\{x_C \geq M/2\}.$$

A bounded correlated OT gives R to S and $R + \alpha\beta M$ to C, where R is uniform on a public interval $\mathcal{J}$ of length A. The parties form Eq. (A.26).

Lemma B.5 (Bounded share lifting)**.** *Assume the working ring represents the interval* $\mathcal{J}$ *and all displayed sums without wraparound. The lifted shares sum to* v *as ordinary integers. Relative to the functionality that samples* U *uniformly on* $\mathcal{J}$ *and returns* $(U, v - U)$ *to* (S, C)*, the protocol has simulation error at most* $M/A + \epsilon_{\mathrm{OT}}$.

Proof. Write $x_S + x_C = v + cM$. On the admitted domain, the three intervals $x_S < M/4$, $M/4 \le x_S < 3M/4$, and $x_S \ge 3M/4$ give, respectively, $c = \beta$, $c = 1$, and $c = 1 + \beta$. Thus $c = 1 - \gamma + \alpha\beta$, and the two lifted shares sum to v. The server share is $R + \tau$, where $\tau = x_S - (1 - \gamma)M$ and $|\tau| < M$. Its law is at distance at most $|\tau|/A \le M/A$ from a uniform $U \in \mathcal{J}$; the complementary client share is determined by v and the server share. Given an input and lifted output, the server simulator recovers $R = \widetilde{x}_S - \tau$, while the client simulator recovers its OT output as $x_C - \widetilde{x}_C$. They use the corresponding OT simulators. The interval-shift bound covers the boundary cases. □

The public local encoders then map each lifted share to a polynomial. These are deterministic operations on that party's state. Ternary returns use the two local integers specified in Appendix A.6. Let ϵ_{lift} collect the statistical costs of all lifting and representation conversions; their cryptographic primitive costs are included in ϵ_{MPC}.

Export and control composition. In Protocol 5, secure lifting yields shares of $\mathrm{rep}_q(\phi)$. Integer linearity gives $A(v_C) + A(v_S) = A(v_C + v_S)$ in the working ring. Applying Theorem B.4 therefore implements Eq. (B.2). The secure integer circuits then realize $\mathcal{F}_{\mathrm{ctrl},j}$ from this shared output. The simulators keep each reused share and allocate new shares by the local rules of the circuit interfaces.

B.3 Encrypted feedback and modular composition

Full and projected feedback. After local share encoding, the full seeded return sends (c_0, ρ) with

$$c_1 = \mathsf{PRG}_{N,Q}(\rho), \qquad c_0 = m_C - c_1 s + e_{\mathrm{enc}} \pmod{Q}.$$

The error e_{enc} uses fresh private coins ω, independent of the public seed ρ. The server adds m_S, applies the consumer projection, and performs the specified alignment. Protocol 4 instead sends $(T_{\mathcal{P}}(c_0), \rho)$. Lemma A.13 proves that this change is a public processing of the full message and preserves the post-projection ciphertext under the stated projector implementation. Subsequent HE work can therefore be coupled using the same inputs and coins.

Ideal-interface reference execution. Let $\Pi_\theta^{\mathcal{I}}$ be obtained from Π_θ by replacing phase sharing, lifting, randomized integer processing, and control circuits with the functionalities defined above. Each phase call processes the phase of its actual input ciphertext; each integer or control call processes the shares supplied by its caller. The reference includes these interface inputs and outputs in the appropriate party's local state and follows the same statistic producers, control consumers, task-head interface, and public schedule as Π_θ. Its final output is obtained by this retained execution. Write $\Pi_\theta^{\mathcal{I}}[\mathcal{S}_P^{\mathrm{if}}]$ for this execution with an online interface simulator attached. At each call, the simulator receives the party's local input and retained state, uses the interface's allocation port where specified, and generates concrete records from the corresponding ideal outputs.

Theorem B.6 (Modular security of protected interfaces)**.** *Fix the protocol parameters θ and the session's invocation budget. Assume that the primitive, phase-decoding, and representation contracts hold for the interface inputs reached in the prescribed execution. For each corrupted party P, there is an online simulator $\mathcal{S}_P^{\mathrm{if}}$ such that the concrete protocol and the ideal-interface execution with this simulator satisfy*

$$\begin{aligned}&(\mathsf{View}_P^{\Pi_\theta}, \mathsf{Out}^{\Pi_\theta}, z)\\ &\quad \approx^{c}_{\epsilon_{\mathrm{if}}} (\mathsf{View}_P^{\Pi_\theta^{\mathcal{I}}[\mathcal{S}_P^{\mathrm{if}}]}, \mathsf{Out}^{\Pi_\theta^{\mathcal{I}}[\mathcal{S}_P^{\mathrm{if}}]}, z),\end{aligned} \tag{B.7}$$

where

$$\epsilon_{\mathrm{if}} \le \epsilon_{\mathrm{bridge}} + \epsilon_{\mathrm{MPC}} + \epsilon_{\mathrm{lift}} + \epsilon_{\mathrm{PRG}}.$$

All terms cover the complete session budget and are counted once.

Proof. For compact feedback, apply the public transcript map from Lemma A.13 to the full-return records. Both executions expose the same compact message format. The post-projection ciphertext is unchanged, preserving the HE continuation.

Next replace interface calls in their public order. For a phase call, the simulator uses Lemma B.2 with the ideal share and the party's setup. For lifting and integer processing it uses Lemma B.5, the

prescribed lifting primitives, and Theorem B.4. The latter jointly generates each local record and allocation before the ideal output is returned. Control circuits use the same allocation convention and their MPC simulators. These replacements retain the joint interface outputs and all previously stored shares. Private PRG streams are handled by their usual hybrids with fresh stream positions.

For the client, this expansion supplies incoming phase messages and integer-protocol records; its outgoing feedback is computed from its local polynomial and encryption coins. For the server, it supplies its local masks and MPC records; feedback processing and data-path HE operations use the same stored state. Thus each replacement is valid in the surrounding computation with its auxiliary state and outputs. Subroutine composition (Canetti, 2000) and the triangle inequality give the sum above. A batch is replaced as one invocation with its prescribed joint-output simulator. Across requests, the simulator retains the same keys, share handles, and preprocessing counters. Conditioning on the preceding history allows the same replacement at the next call within the session budget. □

Relation to the numerical specification. The numerical program $\widehat{f}_{\theta,W}(X, m; \xi)$ in Appendix A provides the arithmetic reference for the approximation analysis. The execution $\Pi_\theta^{\mathcal{I}}$ defines the ideal-interface version of the encrypted protocol. Its task head consumes the protected export of the terminal HE state and reconstructs Q24 logits for the client. Theorem B.6 establishes interface composition by comparing the concrete and ideal-interface executions jointly with their task-head outputs. Local approximation and arithmetic bounds are given in Appendices A and C.3.

Interface parameter accounting. For phase call j, let $n_{B,j}$ be the BFV ring dimension, $E_{B,j}$ the pre-flooding BFV error bound, $L_{B,j}$ the flooding interval length, and $\beta_{\mathrm{dec},j}$ the combined return-decoding failure bound from Lemma B.2. Then

$$\epsilon_{\mathrm{bridge}} \le \sum_{j \in \mathcal{J}_{\mathrm{phase}}} \left(\frac{n_{B,j} E_{B,j}}{L_{B,j}} + \epsilon_{\mathrm{rz},j} + \beta_{\mathrm{dec},j} \right).$$

Each bounded return lift contributes M_j/A_j to ϵ_{lift}; other representation conversions contribute the bounds specified by their interface contracts. MPC and private-PRG terms use the prescribed batch sizes and stream budgets. Each cost is counted once over the complete invocation budget. The bridge sum includes the independent BFV pooler returns in the task head. Appendix C.2 instantiates the flooding, lifting, and decoding terms for the evaluated configurations.

Attention instantiation. Protocol 2 supplies the root statistic D_1 and the normalization denominator D to the protected interfaces. The returned values comprise the amplitude control, protected denominator, and reciprocal seed, including all precision views used by subsequent computation. The mean and polynomial factors remain server-side ciphertexts. The composition argument replaces the two statistic-processing interfaces while retaining their HE producers, control consumers, and public schedule. The corresponding approximation bounds are given in Appendix A.2.

C Experimental Details

C.1 Experimental Setup

Here Qb denotes fixed-point representation with b fractional bits. Both model sizes use the parameters in Table 6. The BFV bridge uses 380-, 320-, and 80-bit moduli for key generation, initial data, and returned ciphertexts. The noise recurrence gives positive decoding margin after each modulus reduction.

C.2 Cryptographic Parameters and Session Accounting

Table 7 lists the key distributions and largest moduli used by the 128-token configurations. Modulus sizes include the auxiliary primes used for evaluation keys. Dense ternary parameters follow the 128-bit category in Table 5.2 of Bossuat et al. (2025); the BFV contexts also pass SEAL's 128-bit parameter check. The joint evaluation-key and fresh-secret assumptions are those of Appendix B.1.

Table 6: Parameters shared by BERT-base and BERT-large.

Component	Setting
Request	Batch 1; $S = 128$ by default, up to 512 in scaling experiments
GELU bulk	Radius 64, polynomial degree 256
GELU exception	Capacity $K = 64$, central degree 12, threshold 4
Attention root	Radius 64; degree 15; eighth power
Amplitude control	$D_1 = \sum (mq)^2$; $\rho = 6 \cdot 2^{-k}$ on Q28
Statistic guard	$1/16$ after the protected Q20 release
Reciprocal	$(21/32)2^{-\lfloor \log_2 \widetilde{D} \rfloor}$; four Newton steps
LN refinement	Degree 16, seven-product polynomial evaluation
Internal LN guard	$\varepsilon = 2^{-14}$; private negative-variance clipping
Protected interfaces	Q72 fine grid, three 57-bit noise terms
Encoder HE ring	Degree 32,768; small export ring degree 4,096
Compact statistic support	Coefficient projection degree 256
BFV phase return	Degree 32,768, 60-bit plaintext, 80-bit return modulus
BFV flooding	Independent coefficient box of width 2^{248}

Table 7: HE key parameters. U_3 denotes independent uniform coefficients in $\{-1, 0, 1\}$; $h = 32$ denotes a uniformly supported ternary key with 32 nonzero coefficients. Modulus sizes are rounded upward.

Context	Ring degree	Max. modulus (bits)	Secret distribution
Encoder CKKS	32,768	868	U_3
Small-ring export	4,096	60	U_3
GPU bootstrap	65,536	1,747	U_3
Bootstrap sparse key	65,536	121	$h = 32$
BFV phase bridge	32,768	380	U_3
BFV task head	16,384	330	U_3

The sparse key is used by the dense-to-sparse bootstrap switch at a single data prime and one auxiliary prime; the full bootstrap modulus belongs to the dense key. The return adapter embeds the encoder key in the larger ring and uses at most 863 modulus bits. The coefficient projection of degree 256 specifies transmitted support, while the surrounding ciphertext retains its declared HE ring. The GPU key-error sampler rounds a Gaussian with standard deviation 3.2, clipped at six standard deviations. BFV uses a centered binomial sampler with coefficient support $[-21, 21]$.

Flooding and decoding. For the phase bridge, the plaintext modulus is $1\,152\,921\,504\,593\,412\,097$ and the flooding interval has length 2^{248}. The bounded-noise recurrence gives $E_B \le 8.123 \cdot 10^{23}$ before flooding. After reduction to the 80-bit return modulus, the error is below $16\,513$, versus a decoding radius above $393\,216$. The task head uses plaintext modulus 2^{40}, a 2^{202} flooding interval, and $E_B \le 11\,543\,412\,758$. Its 110-bit return has error below $6.872 \cdot 10^{10}$ and decoding radius above $5.902 \cdot 10^{20}$. These strict inequalities give zero return-decoding failure under the bounded sampler and integer-arithmetic contracts. The pre-flooding bounds include rerandomization. Each invocation uses a fresh flood; pooler keys are independently generated for each request.

Bounded lifting. The implemented source widths are $\ell \in \{24, 32, 40, 53, 60\}$ bits, with corresponding mask widths $b \in \{192, 200, 208, 216, 224\}$. For each lifted value, the interval-translation bound is at most $2^{\ell+1-b}$. The session contribution is

$$\epsilon_{\text{lift}} \le \sum_{(\ell,b)} n_{\ell,b}\, 2^{\ell+1-b},$$

where $n_{\ell,b}$ counts all values of that width, including padding and repeated calls. This bound uses the integer representation domains of the lifting interfaces.

Table 8: Interface counts and statistical bounds for 128-token inference. Counts are per request. The lower block gives lower bounds on $-\log_2 \epsilon$ for 1,024 requests under one setup, including warmup.

Quantity	BERT-base	BERT-large
Phase-sharing calls	60	120
Flooded phase coefficients	2,064,384	4,063,232
Lifted values	657,280	1,286,016
Flooded pooler coefficients	606,208	1,048,576
Phase flooding	137.59	136.61
Bounded lifting	133.64	132.66
Pooler flooding	139.36	138.57
Combined interface contribution	133.52	132.54

Table 8 evaluates $\sum_j n_{B,j} E_{B,j}/L_{B,j} + \epsilon_{\mathrm{lift}}$ using the source and binary configuration of the main 128-token measurements. The resulting interface error is

$$\epsilon_{\mathrm{if}} \le \epsilon_{\mathrm{crypt}} + \begin{cases} 2^{-133.5}, & \text{BERT-base}, \\ 2^{-132.5}, & \text{BERT-large}, \end{cases}$$

under the contracts of Theorem B.6. Here $\epsilon_{\mathrm{crypt}}$ collects the computational advantages for BFV rerandomization, MPC/OT, and private PRG streams over the session. The statistical sum holds conditional on each legal execution prefix, so requests may depend on earlier outputs. Fresh per-request pooler keys do not reset this sum. Other sequence lengths use their own invocation counts with the same accounting rule.

C.3 Numerical domains and error bounds

We analyze the nonlinear modules on their stated input domains with local arithmetic error bounds, following the operator-level treatment of homomorphic Softmax and fused CKKS computation (Cho et al., 2024; Xu et al., 2025). Polynomial and integer-control errors are determined by the stored coefficients and circuits. HE residuals are measured relative to the retained input and consumed controls of each module. Table 9 summarizes the resulting local bounds; the parameter instances and execution measurements follow below.

Table 9: Local numerical bounds used for parameter selection.

Module	Domain or error condition	Result
Attention	$(s,m) \in \mathcal{D}_B$; root error $\delta < 1$	Normalized approximation error, Eq. (A.5).
Attention control	Statistic bounds in Lemma A.3	Numerator energy and denominator range, Eq. (A.6).
LayerNorm	Variance condition (A.15); local residual ρ_{LN}	Row RMS $\le \sqrt{K_p/\theta} + \rho_{\mathrm{LN}}$.
GELU allocation	Row RMS $\le g$; affine–projection error $\le \eta_j$	$\lvert h_j\rvert \le \lvert\mu_j\rvert + g\sqrt{d}\lVert a_j\rVert_2 + \eta_j$.

LayerNorm range allocation. The three bounded noise terms and Q30 quantization contribute at most $H = 3 \cdot 2^{-16} + 2^{-30}$ to the variance interface. The numerical contracts use the following allocation in Eq. (A.15):

$$\alpha = \frac{255}{256}, \quad \theta = \frac{63}{64}, \quad a_v = 2^{-13}, \quad b_v = 2^{-17} + H.$$

The 2^{-17} term is the additive variance-producer budget; H is fixed by the interface distribution. With this allocation, Eq. (A.16) gives the sufficient condition $\varepsilon \ge 5.37246 \cdot 10^{-5}$. We set $\varepsilon = 2^{-14}$ for both model sizes. The terminal CLS normalization is evaluated in the integer task head. For $g = 1.25$, Eq. (A.19) gives the sufficient aggregate error allocation

$$\rho_{\mathrm{LN}} \le g - \sqrt{K_p/\theta} = 0.242094243\ldots. \tag{C.1}$$

The combined budget covers control return, application, and refresh in the units of the normalized state. We allocate $\eta_j = 4$ to the subsequent affine transform and FFN projection in Eq. (A.20). Proposition A.7 gives the channel bounds for an LN output satisfying these local conditions. Model preparation uses the resulting g and η_j to allocate GELU channels.

The LN control circuit jointly clips negative variance estimates, adds the guard, and selects the normalization scale. For 128 rows, it uses 59,776 AND gates and returns 256 words in three communication epochs.

Refresh error. At an LN output, write the state before the learned affine transform as $z = \sigma C$, where $\sigma = ap(\bar{u})$. Equations (A.16)–(A.19) bound its energy. The input to refresh is Z^-/B_z, where $Z^- = z + e_{\mathrm{app}}$ and public $B_z \geq 2\sqrt{2d}$. The pre-refresh bound $\|e_{\mathrm{app}}\|_2/\sqrt{d} \leq \rho_{\mathrm{app}}$ controls this input range separately. A bootstrap error δ in these units adds $B_z\delta$ to the recovered state. Refreshing C/B_C before applying σ instead adds $B_C\sigma\delta$. The reference $z = \sigma C$ uses the same retained state and integer controls as the encrypted execution. Arithmetic errors introduced by applying σ and refreshing the result enter Eq. (A.18). The state remains at hidden width before the next FFN or QKV projection.

Attention approximation and carrier bounds. We instantiate the domain in Eq. (A.4) with $B = 64$ and use a degree-15 polynomial for e^{8u} on $[-1, 1]$. The fixed evaluation schedule applies this polynomial to every row; the approximation and carrier bounds below are established on $\mathcal{D}_{64}$. Its certified uniform error, including stored factorization, is at most $1.34458 \cdot 10^{-3}$. Proposition A.2 then gives a normalized polynomial error below 0.02162, independently of row length. The coefficient certificate uses rational intervals for the Chebyshev coefficients and the series tail, followed by exact rational expansion of the deployed factors. Equation (A.9) adds the local evaluation and protected-interface terms.

Amplitude control bounds the energy after evaluating q and before taking eighth powers. With $q = p/p(0)$, the stored coefficients give $d_0 = (1-\delta)^2/p(0)^2 > 0.99927$ and $D_1 \geq nd_0$. For zero-mean $u_j \in [-1, 1]$, the secant bound for the exponential gives

$$D_1 \leq \frac{n\{\cosh(16) + 2\delta\cosh(8) + \delta^2\}}{p(0)^2}.$$

Consequently $D_1/T < 2^{30}$ for $n \leq 128T$, $T \in \{1, 2, 4\}$. After restoring the tile factor, the protected release contributes at most $T(3 \cdot 2^{-16} + 2^{-20})$ absolute error. The controller adds $G = 1/16$. We allocate a relative arithmetic term and an absolute interface term:

$$|\widetilde{D}_1 - D_1| \leq \epsilon_1 D_1 + \xi, \qquad \epsilon_1 \leq 1/64, \quad \xi \leq G. \tag{C.2}$$

Thus $(1-\epsilon_1)D_1 \leq \overline{D}_1 \leq (1+\epsilon_1)D_1 + G + \xi$; Lemma A.3 applies with $\Delta = G + \xi$. Using $d_* = nd_0$ in the lemma gives

$$D \geq \frac{36^4 d_0^4 n}{\{(1+\epsilon_1)nd_0 + \Delta\}^4}.$$

This bound decreases with $n \geq 1$ for the stated allocation. With $n \leq S$ and $\Delta \leq 1/8$, it gives $D > 0.7498$ at 128 tokens and $D > 0.01175$ at 512, while $D \leq (144 \cdot 64/63)^4 < 4.58 \cdot 10^8$. Combining these domain bounds with the HE numerator and denominator residuals gives the total error in Eq. (A.9).

Finite controller domains. LN consumes signed Q30 values in a 50-bit ring. Negative estimates are mapped to zero inside the circuit; with $\varepsilon = 2^{-14}$ its output satisfies $2^{-10} \leq a \leq 64$ and $1/4 \leq \bar{u} \leq 1 - 2^{-22}$. The first attention statistic uses unsigned Q20 with 51 bits at 128 tokens and 52 bits for the reconstructed longer-sequence sum. The denominator uses 53-bit Q20 shares. The reciprocal seed uses Q47, with a 53-bit ring for 128 tokens and 56 bits for longer inputs; the latter table includes denominator bins down to 2^{-7}. Its local numerical contract requires $\widetilde{D} \geq 2^{-1}$ at 128 tokens and $\widetilde{D} \geq 2^{-7}$ at longer lengths. Equation (C.2) and the denominator budget $|\widetilde{D} - D| \leq D/64$ imply both conditions by the preceding bound: $\widetilde{D} > 0.7381$ at 128 tokens and $\widetilde{D} > 0.01156$ at 512, with $\widetilde{D} < 4.651 \cdot 10^8 < 2^{29}$. This allocation makes the seed table consistent with the numerical error model. Word sizes are public; the protocol's control schedule and ciphertext dimensions remain

fixed for a given sequence length. The local HE estimate below is parameterized by the scale range and arithmetic precision used by its consumer. GELU preparation requires $\max_j B_j < R_{\mathrm{ctl}} = 512$ and at most $K = 64$ channels with $B_j > 64$. Model preparation checks these limits with the corresponding export-error margins for integer reduction. It rejects a checkpoint whose channel bounds or exception count exceed the public GELU parameters. This preparation checks the channel allocation for the specified state-norm and producer-error budgets.

Homomorphic normalization error. Write $\sigma_0 = ap(\bar{u})$, with $a \geq a_{\min} > 0$ and $p(\bar{u}) \geq p_{\min} > 0$. Let E_a bound the absolute error in the returned scale and E_p the error in the evaluated polynomial, including its returned input. Let E_f bound the relative error introduced when multiplying these two approximate values, and let E_{mul} bound the row-RMS error of applying the resulting multiplier to C. The relative multiplier error is at most

$$r_\sigma = \left(1 + \frac{E_a}{a_{\min}}\right)\left(1 + \frac{E_p}{p_{\min}}\right)(1 + E_f) - 1. \tag{C.3}$$

Under the variance and guard conditions of Proposition A.7, $\|\sigma_0 C\|_2/\sqrt{d} \leq \sqrt{K_p/\theta}$, hence

$$\rho_{\mathrm{app}} \leq \sqrt{K_p/\theta}\, r_\sigma + E_{\mathrm{mul}}. \tag{C.4}$$

This estimate separates multiplier precision from the energy of the retained state and supplies the pre-refresh bound in Appendix A.3.

For the $N = 32768$ contexts, consider $V \leq 1024$ satisfying Eq. (A.15), successful integer lifting, $2^{-6} \leq a \leq 64$, and $\bar{u} \in [1/4, 1)$. Condition on canonical norms at most 3,072 for the secret and 8,192 for the key-error polynomials. The Q33 return and Q40 evaluation give $E_a \leq 2^{-13}$, $E_p \leq 2^{-9}$, and $E_f \leq 2^{-22}$. With $p_{\min} = 0.99$ and $E_{\mathrm{mul}} \leq 2^{-8}$, Eq. (C.4) gives $\rho_{\mathrm{app}} < 0.013796 < 1/64$. This common bound covers the Attention-output and FFN-output LN arithmetic paths of both model sizes on the stated domain. For the MRPC checkpoints and input row RMS at most 1.25, Eq. (A.20) evaluated with the encoded weights gives affine–FFN1 error bounds of 0.477966 and 0.394704 for Base and Large across all layers, within the allowance $\eta_j = 4$.

Execution measurements. Table 10 reports validation-set ranges and local errors from instrumented encrypted inference. Each local reference uses the same retained input and consumed integer controls as its encrypted counterpart, following the distinction between arithmetic error and error propagation used in CKKS precision analysis (Costache et al., 2024). The largest observed centered score is 53.710; denominator relative errors in the instrumented requests are below 0.002. The LN measurements cover Attention-output normalization, with the pre-affine state bounds derived using the local affine error estimate.

CPU arithmetic tests additionally cover interval endpoints, adjacent control bins, bounded-noise extrema, and zero or small variances. They give maximum Attention row ℓ_1 error 0.002065 and, under the variance-producer condition, minimum LN guard margin $7.39 \cdot 10^{-6}$. Changing padding contents with fixed valid scores leaves the numerical Attention outputs unchanged.

C.4 Wide-interval GELU: construction and cost

Following the approximation-based treatment of nonlinearities in Lu et al. (2025); Xu et al. (2025), we specify both the interval and the evaluation error. We approximate the even part $E(x) = \frac{x}{2}\operatorname{erf}(x/\sqrt{2})$ using

$$p_R(x) = \frac{x}{2} + \sum_{k=0}^{m} b_k T_k\left(2(x/R)^2 - 1\right).$$

The coefficients come from a Chebyshev expansion on the public interval; the constant is adjusted to enforce $p_R(0) = 0$ before coefficient quantization. For $R = 64$ and $m = 128$, the total degree is 256. Recursive decomposition $Q = A + T_k B$ produces 16 leaves of degree at most eight. The resulting evaluation uses 26 ciphertext products and eight product levels, including the initial square.

The stored leaf coefficients use Q24 and incorporate the public prime-dependent scale factors of the HE evaluation graph. We certify that compiled polynomial directly. With grid spacing $h = 2^{-10}$,

Table 10: Observed ranges, encrypted-execution measurements, and local error bounds at $S = 128$. Observed values and inferred bounds report maxima, except for the minimum LN guard margin. Analytical bounds use the domains specified in the text.

Quantity	Base	Large	Domain / budget
Observed ranges			
Valid-key centered score $\lvert s_j - \mu \rvert$	45.035	53.710	64
First statistic D_1	$7.816 \cdot 10^4$	$7.005 \cdot 10^5$	2^{30}
Denominator D	$4.128 \cdot 10^8$	$4.017 \cdot 10^8$	$4.58 \cdot 10^8$
Encrypted MRPC: Attention-output LayerNorm			
Variance-producer error	$7.004 \cdot 10^{-9}$	$5.786 \cdot 10^{-8}$	Eq. (A.15)
LN guard margin $w - \theta V$ (min.)	$3.625 \cdot 10^{-4}$	$3.090 \cdot 10^{-4}$	≥ 0
LN affine-output error	$3.624 \cdot 10^{-4}$	$4.763 \cdot 10^{-4}$	—
Aggregate LN residual (inferred)†	0.003364	0.001445	0.242094
Refreshed row RMS (inferred)†	1.003308	1.001367	1.25
Encrypted MRPC: Attention statistics			
First-statistic relative excess‡	$1.730 \cdot 10^{-4}$	$1.844 \cdot 10^{-4}$	$1/64$
Denominator relative error	0.001910	0.001661	$1/64$
Local arithmetic bounds			
LN application (both LN positions)	0.013796	0.013796	$1/64$
Affine–FFN1 error	0.477966	0.394704	4

Range statistics aggregate the five validation sets (8,063 inputs per model). Local errors use separate instrumented MRPC executions, one per model; LN measurements cover Attention-output LN. †Bounds for the sampled pre-affine states, inferred from post-affine measurements under the local affine error assumptions. ‡Defined as $\max(0, |\widetilde{D}_1 - D_1| - H_{20})/D_1$, where $H_{20} = 3 \cdot 2^{-16} + 2^{-20}$ is the bounded noise and quantization term. The denominator row uses $|\widetilde{D} - D|/D$ directly.

directed 640-bit interval arithmetic bounds the error at every grid point. For $f = p_R - \mathrm{GELU}$, linear interpolation gives

$$\|f\|_\infty \leq \max_{x \text{ on grid}} |f(x)| + \frac{h^2}{8}\|f''\|_\infty.$$

Chebyshev derivative coefficients bound $\|f''\|_\infty$; evenness covers both halves of the interval. The resulting continuous error bound for the compiled Q24 graph is $5.012 \cdot 10^{-5}$ on $[-64, 64]$. For inputs in the admitted interval, encrypted input error E_x, and evaluation residual E_{HE}, the corresponding output bound is $5.012 \cdot 10^{-5} + \|p_R'\|_\infty E_x + E_{\mathrm{HE}}$.

Table 11: Public GELU polynomial tradeoffs. Error bounds cover the full interval; product counts refer to one packed ciphertext.

Radius	Degree	Uniform error bound	Products	Product depth
40	128	$1.085 \cdot 10^{-3}$	17	7
40	160	$4.896 \cdot 10^{-5}$	20	8
40	192	$1.406 \cdot 10^{-6}$	22	8
64	128	$3.965 \cdot 10^{-2}$	17	7
64	192	$2.210 \cdot 10^{-3}$	22	8
64	256	$5.027 \cdot 10^{-5}$	26	8

Table 11 uses the unquantized stored Chebyshev coefficients to isolate the interval–degree tradeoff. Increasing the degree changes local HE work; increasing the radius can reduce the number of channels requiring MPC control. At fixed $R = 64$ and $K = 64$, the degree does not change communication. Across all 180 layers of the ten evaluated checkpoints, the $g = 1.25$, $\eta_j = 4$ channel bound requires at most 2,015/4,013 exceptional channels for Base/Large at radius 40, compared with 5/12 at radius 64. The maximum channel bounds are 260.03/340.34, below $R_{\mathrm{ctl}} = 512$. This explains the joint choice of the wider polynomial and a 64-channel control interface. An isolated HE evaluation of the compiled polynomial, using eager relinearization, gives maximum errors of $8.46 \cdot 10^{-6}$ (Base

width) and $8.36 \cdot 10^{-6}$ (Large width) relative to its plaintext evaluation, and at most $5.47 \cdot 10^{-5}$ relative to GELU on the tested inputs.

C.5 Communication accounting

Table 12 gives the values in Figure 7(b). We sum bytes sent by both parties during inference. GB denotes 10^9 bytes; BumbleBee's binary units are converted accordingly. Ratios use unrounded traffic, and dashes denote unreported results.

Traffic categories. *Ciphertext transfer* covers encrypted inputs, masking ciphertexts, HE-to-MPC exports, and MPC-to-HE feedback in the encoder. *MPC messages* cover secure conversion, control evaluation, and share encoding, including the OT corrections used by those protocols. The pooled *OT extension* supply is counted separately. *HE key material* includes keys transmitted during inference. *Task head* covers the private classifier, output, and eight-byte request header. These disjoint categories are checked against the per-party send counters and sum to Table 3. Bootstrapping and local ring adaptation require no messages.

Table 12: Communication breakdown.

	BERT-base		BERT-large	
Component	MB	Share	MB	Share
Ciphertext transfer	94.0	26.4%	156.3	22.0%
MPC messages	216.6	60.8%	485.9	68.4%
OT extension	7.8	2.2%	14.0	2.0%
HE key material	9.0	2.5%	9.0	1.3%
Task head	28.8	8.1%	45.0	6.3%
Total	356.3	100.0%	710.3	100.0%

Projected feedback. All control returns use the degree-32,768 key ring. A fresh seed regenerates c_1; selected coefficients of c_0 are transmitted at their RNS prime bit widths. The formats are:

Control	Slot period	Coefficients sent	Projection cost
Attention row control	2,048	4,096	3 rotations
LN scale/mantissa	128	256	7 rotations
GELU branch	–	16,384	1 conjugation

GELU uses the real-slot projection. Amplitude selection returns one control; the denominator and LN paths each return two. The return modulus includes the rescale following projection. Statistic exports contain only the masked phase coefficients needed by the decoder.

For t transmitted coefficients and RNS limb widths b_i, the coefficient payload is $t \sum_i b_i/8$ bytes, plus a seed and metadata. A full seeded return uses $t = N$, a periodic return uses $t = 2p$, and real projection uses $t = N/2$. The return modulus must support both projection and subsequent refinement, so the coefficient count and refinement depth jointly determine the feedback cost.

Protocol ablations. Figure 8(b) reports bidirectional protocol traffic in an interior block, excluding pooled OT replenishment. The dense GELU and attention counterfactuals replicate measured conversion pages; LN compares matched encrypted executions. GELU and attention retain homomorphic state refresh and projected feedback. The LN comparison includes state refresh at both LN boundaries and inverse-square-root evaluation.

The GELU ablation applies the same protected decision protocol to all FFN channels instead of the public capacity $K = 64$. A page of $k \le 128$ channels costs $2{,}817{,}043 + 449Sk$ bytes, including ciphertexts and conversion. At $S = 128$, full-width control needs 24 pages in Base and 32 in Large; selective control needs one 64-channel page. Communication falls from 244.2/325.6 MB to 6.5 MB.

The attention ablation exports the individual contributions A_j^2 and E_j, then adds their arithmetic shares locally before applying the same row controllers and feedback. The real-arithmetic row sums

are unchanged; protected rounding precedes aggregation in this counterfactual. Each existing 2,048-value export page costs $C_{\text{page}} = 1{,}289{,}493$ bytes: 368,789 for ciphertexts and 920,704 for protected conversion. For the two stages,

$$C_{\text{dense}} = C_{\text{row}} + 2\big(\lceil HS^2/2048 \rceil - 1\big)C_{\text{page}}.$$

This changes the number of export pages per stage from 96/128 to one in Base/Large, reducing attention protocol traffic from 255.1/339.8 MB to 10.1/12.3 MB. These counts use the same page format, rather than a separately optimized dense packing scheme.

The LN reference uses interactive state refresh and MPC inverse-square-root evaluation. The two LN boundaries communicate 44.7/54.5 MB per block in Base/Large. Homomorphic refresh removes 31.7/41.5 MB of state round trips, leaving 13.0 MB in each model; secret-scale selection and HE refinement reduce this to 6.4 MB. The complete LN design therefore saves 85.7%/88.2%. These differences are constant across all interior blocks in the paired runs. GELU and attention messages are excluded from the LN accounting, including messages in shared execution phases.

Joint control and feedback design. The cumulative ablation in Figure 8(a) starts at 899.5 MB for Base and 2044.1 MB for Large. Homomorphic state refresh saves 363.0 and 973.7 MB, respectively; projected feedback saves 104.3 and 205.3 MB; secret-scale selection with HE LN refinement saves 75.8 and 154.9 MB.

Figure 9 crosses two choices: MPC evaluation of the LN inverse square root or HE refinement from a secret scale and mantissa, and full or projected ciphertext feedback. All four configurations use the same model, protected interfaces, GELU capacity $K = 64$, LN guard $\varepsilon = 2^{-14}$, and two refresh boundaries per internal block. Each request includes the private task head. The reference evaluates the inverse square root in MPC and returns full ciphertexts, matching + *State refresh* in Figure 8(a). For each model, we plot $100(C - C_{\text{ref}})/C_{\text{ref}}$, where C_{ref} is this reference's total communication and C is the total under each configuration.

HE refinement with full feedback increases communication by 9.2%/9.4% in Base/Large because it requires a larger return modulus. Combined with projected feedback, it saves 33.6% in both models.

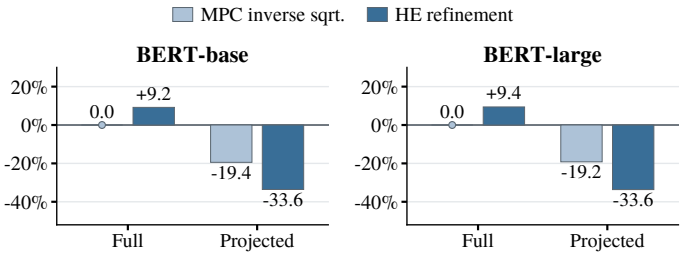

Figure 9: Joint LN evaluation and feedback ablation. Communication change (%) relative to MPC inverse square root with full ciphertext feedback.

For the projection ablation, we restore each omitted coefficient while holding the controls, ciphertext seed, and message headers fixed. A return with ring degree N, k transmitted coefficients, and modulus width b restores $(N - k)b/8$ bytes. Summing the recorded returns gives 229.4 MB for Base and 461.0 MB for Large, of which LN contributes 125.1 and 255.7 MB.

Sequence length. The scaling experiments concatenate valid embedding rows from the registered inputs to form sequences of 128, 256, and 512 tokens. Each run completes the encrypted encoder and private task head with the same HE parameters and GELU capacity. Attention uses 128-row tiles: each query tile visits every key/value tile, accumulating row means, root statistics, denominators, and weighted values over the full sequence. For $T = S/128$, the first statistic is exported as D_1/T and restored by public multiplication by T in MPC. The sharing widths and reciprocal seed table follow Appendix C.3.

For L layers, $(L - 1)T + 1$ query tiles are evaluated: the final layer needs only the CLS query tile, which still attends to every key tile. With model dimensions and HE parameters fixed, each query

tile exchanges a fixed-size control transcript. Inputs and validity masks also grow linearly with S; the request header, evaluation keys, and task head add fixed terms. Total communication is therefore $O(S)$. The conventional HE–MPC schedules compared here exchange $O(HS^2)$ attention entries and have $O(S^2)$ communication for fixed H. All query–key products remain local HE computation. Inactive ciphertext tiles reside in host memory; latency includes their GPU transfers. Each point uses one timed request after initialization.

C.6 Homomorphic computation

Table 4 reports three major HE operations for an internal encoder block, except the marked per-head entries. One bootstrap refreshes one ciphertext; a refresh stage may contain several such operations. Rotations and conjugations are counted together, and ciphertext–ciphertext multiplication includes squaring. Products are counted before relinearization. Counts exclude bootstrap internals, setup, the task head, conversion, and ring-adaptation circuits. A lower bound is marked by $\geq$; "–" denotes an unavailable count.

State refresh. Each nonterminal LN refreshes its normalized output before the learned affine transform. The adapter retains two data primes to form z/B_z and rescales to the bootstrap input. Appendix C.3 gives the input bound and error allocation. Two degree-2^{15} ciphertexts pack into one degree-2^{16} ciphertext. Base has three state ciphertexts and Large four, so each LN boundary uses two bootstraps. We use Cheddar (Choi et al., 2025), followed by encrypted ring switching and scale adaptation for the next FFN or QKV projection.

An internal block therefore uses four bootstraps, all at its two LN outputs. Attention and GELU require none. NEXUS uses 12 and THOR 25–30 under the schedules in Table 4, or 3–7.5$\times$ as many. Our encoder uses degree 2^{15}; bootstrapping uses degree 2^{16}, as do the pure-HE schedules in the table.

The terminal LN supplies the task head directly. At 128 tokens, Base and Large have 23 and 47 refresh boundaries, giving 46 and 94 bootstraps over the encoder. For $T = S/128$ tiles, the corresponding boundary count is $2(L-1)T+1$. Refresh takes 15 s in Base and 35 s in Large, accounting for 29.6% and 28.6% of their encoder-and-head intervals. These times include bootstrapping, ring/scale adaptation, and local transfers; refresh sends no messages.

Retained factors for amplitude feedback. Factor the degree-15 root polynomial as $q(u) = g(u+\ell)\prod_{i=1}^{7} f_i(u)$, with quadratic f_i. HE retains subproducts of degrees $8, 4, 2, 1$ while the statistic branch finishes $q(u)$ and its masked square for D_1. Fresh amplitude feedback forms $m\rho(u+\ell)$; four products combine it with the retained subproducts, and three squares produce E. This reuses earlier work and preserves the depth needed by EV, without bootstrapping the score matrix. V projection proceeds during amplitude feedback. After E is formed, its masked row sum supplies the reciprocal control while the same E participates in EV.

Refinement circuits. Attention computes $e_0 = 1-\widetilde{D}r_0$ and updates $r_{i+1} = r_i(1+e_i)$, $e_{i+1} = e_i^2$. The residual identity in Appendix A.2 proves equivalence to Newton's update. Forming e_0, performing k updates of r, and squaring $k-1$ residuals uses $2k$ ciphertext multiplications at multiplicative depth $k+1$. Four iterations use eight multiplications at depth five.

For LN, write the degree-16 polynomial in $z = 1 - 8u/5$ as $\widetilde{p}(z) = \sum_{j=0}^{16} c_j z^j$ and set $A_j(z) = \sum_{i=0}^{3} c_{4j+i} z^i$ for $j = 0, \ldots, 3$. Then

$$\widetilde{p}(z) = A_0(z) + z^4 A_1(z) + z^8\big(A_2(z) + z^4 A_3(z) + c_{16} z^8\big).$$

Four multiplications prepare z^2, z^3, z^4, z^8; three more evaluate the polynomial. Multiplication by a gives eight in total for $ap(u)$. The cubic leaves share powers, and plaintext coefficient products are accumulated before rescaling.

Count sources. NEXUS (Zhang et al., 2025a) packs a 128×768 state into three ciphertexts (Section V) and refreshes before and after each LN (Table IV), giving $4 \times 3 = 12$ bootstraps in its published compact schedule. Its arithmetic entries count one head's $QK^\top$ in Algorithm 5:

$128 \times 64 = 8{,}192$ ciphertext products and $127 \times 64 = 8{,}128$ rotations, omitting the unused final rotation of each key column. These are the 12-head totals divided by 12; bootstrapping remains a block-level count. THOR's counts (Moon et al., 2025) trace its complete public block schedule, including level-dependent refreshes inside Softmax and LN. The ranges reflect its layer-specific nonlinear circuits. Rho et al. (2025) use a 768-wide encoder with Gaussian attention, GLU, and encrypted rank-two LoRA. Its lower bound counts mandatory forward-path refreshes, including those inside repacking and ReLU; conditional LN and exponential refreshes are excluded.

BOLT's Base counts (Pang et al., 2024) trace its released BFV computation with ring degree 2^{13}. The Large rotation bounds for BOLT and BLB come from the two attention products in BLB's Table 7 (Xu et al., 2025); BLB's multiplication bound comes from its Table 5. Their interactive MPC boundaries avoid bootstrapping. BOLT's two attention products alone use over $10\times$ as many rotations as our complete BERT-large block. Our complete block also uses fewer ciphertext products than BLB's two attention products.

Operation costs also depend on modulus levels and nonlinear circuits; latency is reported separately in Table 3.