SHAPE-SELECTIVE CORRELATION CONVERSION FOR PUBLIC-SHAPE PRIVATE TRANSFORMER INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Private inference systems typically prepare correlations for a fixed tensor graph, whereas token pruning and elastic networks may reveal a public tensor shape only after preprocessing has been distributed. Preparing every supported shape duplicates correlations, while generating and delivering preprocessing after disclosure delays online execution. We introduce shape-selective correlation conversion (SSCC), which locally specializes distributed preprocessing while preserving the fixed-shape online protocol. For retained bilinear correlations, we identify the missing contraction term and characterize the exact minimum additional ring elements required by zero-interaction share-linear decoders over finite chain rings.

We instantiate SSCC for nested attention capacities and elastic SwiGLU widths and compose both axes in one Transformer block without a capacity-by-width preprocessing family. For public shape selection and one-use preprocessing, conditional completion lifts fixed-shape security under branch-isolation and joint-pseudorandomness assumptions. Across nine capacity-width pairs, selected and separately generated fixed-shape outputs agree exactly and incur identical online communication. Factorized preprocessing reduces per-party storage by 50.39% relative to a capacity-deduplicated fixed-shape baseline, and a synthetic 12-block chain checks cross-block output and communication consistency. A BERT-base case study on the Microsoft Research Paraphrase Corpus matches the unpruned accuracy of 86.52%. Across 20 co-located repetitions per method and width excluding delivery, the observed median interval from public width disclosure to output is 16.3% to 20.7% lower than for generating fixed-shape preprocessing after disclosure.

1 INTRODUCTION

Private Transformer inference protects client inputs and, in many deployments, provider models (Rathee et al., 2020; Chen et al., 2022; Li et al., 2022; Gupta et al., 2024). In private inference based on function secret sharing (FSS), a trusted dealer typically distributes one-use preprocessing for a fixed tensor graph before an online request (Gupta et al., 2024; Brüggemann et al., 2026). Our target deployment distributes topology-dependent preprocessing before the public tensor shape of a request is known. During a request, an earlier secure computation may privately score tokens or select an elastic-compute policy and reveal only the resulting tensor shape. At that point, generating and delivering fixed-shape preprocessing would stall the request, whereas pre-distributing session-compatible preprocessing for every supported shape would duplicate correlated material. The parties therefore need one-use preprocessing that is distributed before the shape is known but can be specialized locally after the shape becomes public. Two motivating workloads vary different dimensions: token-pruning systems determine a public retained-token capacity after private scoring (Pang et al., 2024; Zhang et al., 2025; Cai et al., 2026), while an input- or policy-dependent use of an elastic architecture such as MatFormer may select a public feed-forward width for each layer (Devvrit et al., 2024). In both cases the selected tensor shape can be public even though the inputs, model, and intermediate tensors remain private.

Two straightforward deployment strategies expose this tradeoff. A dealer can distribute independent fixed-shape preprocessing for every supported shape before the online request. This provides immediate availability once the shape is selected, but preprocessing storage and dealer-to-party delivery

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

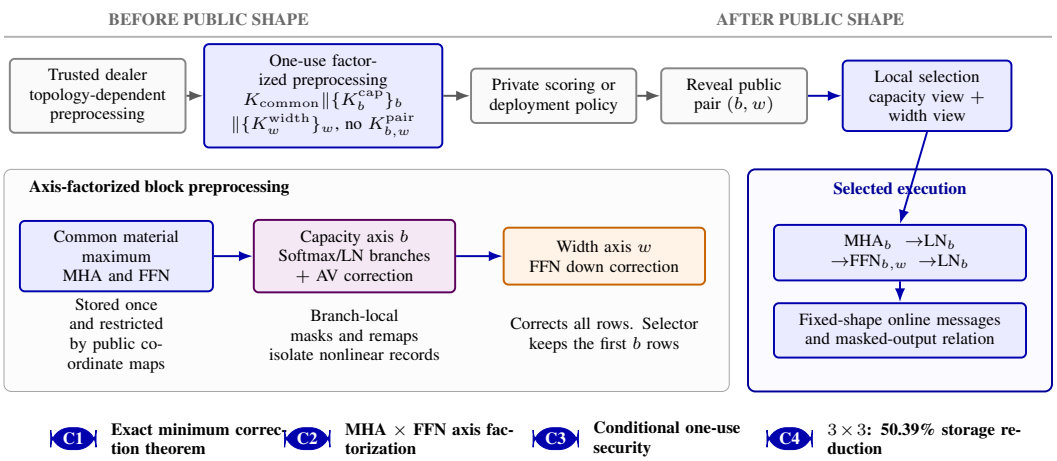

Figure 1: Shape-selective correlation conversion in its deployment setting. The dealer distributes one-use factorized preprocessing before the public capacity and width pair is known. After disclosure, both parties locally derive preprocessing that satisfies the selected fixed-shape relations and run the fixed-shape online protocol. The preprocessing has common, capacity-indexed, and width-indexed material but no capacity-by-width family.

scale with the aggregate preprocessing size. Alternatively, the dealer can wait until the selected tensor shape is public and then generate only the corresponding fixed-shape preprocessing. This minimizes distributed bytes but delays online execution until generation and delivery finish. In this work, we ask what additional correlations enable the parties to locally derive exact selected-shape preprocessing from retained maximum-shape preprocessing after the shape becomes public. Coordinate projection alone is insufficient. The key obstruction arises when shape selection shortens the dimension summed over in a bilinear product: the maximum-shape masked product still contains contributions from coordinates absent from the selected computation. Canceling these contributions while preserving the selected-shape output mask requires additional correlated correction material. This paper formalizes this obstruction and determines its exact explicit-word cost while keeping the fixed-shape online protocol unchanged. We call this construction *shape-selective correlation conversion* (SSCC). We evaluate SSCC through algebraic conversion, conditional security, storage accounting, and agreement with fixed-shape execution in a composed multi-head attention (MHA) and feed-forward network (FFN) block and a synthetic 12-block chain. A separate case study evaluates a trained 12-layer checkpoint using hybrid homomorphic encryption (HE) and multiparty computation (MPC). We also measure co-located FFN request intervals from public width disclosure to output. Cross-system delivery remains a separate diagnostic.

Figure 1 summarizes the deployment workflow and the factorization of preprocessing across the public capacity and width axes.

Before the shape is known, the dealer distributes to each party a one-use preprocessing share consisting of three types of correlations: maximum-shape correlations that admit coordinate restriction, correction values for bilinear products, and separate shape-specific correlations for components that cannot be derived from the maximum-shape correlations. Once the shape becomes public, each party locally selects the required coordinates, applies its correction shares, and appends the corresponding shape-specific correlations. Under the conditional-completion premises in Section 4, the selected view has the fixed-shape distribution and feeds its online protocol with unchanged communication. We summarize our contributions as follows:

1. We define SSCC for public shapes selected after topology-dependent preprocessing has been distributed. For retained bilinear correlations, we prove matching upper and lower bounds on the additional ring elements required by zero-interaction share-linear selectors over finite chain rings.
2. We construct SSCC for nested MHA capacities and prefix-elastic SwiGLU widths and compose both axes into factorized preprocessing for one Transformer block. The construction separates

common, capacity-indexed, and width-indexed correlations, avoiding a family indexed by every capacity and width pair. Conditional completion lifts fixed-shape security under the stated one-use and randomness premises.

3. We validate the composed construction across nine joint-block configurations. Selected and fixed-shape outputs agree exactly, and factorized preprocessing is 50.39% smaller than a capacity-deduplicated fixed-shape baseline. A 12-block synthetic chain matches a separately generated fixed-shape reference and incurs $12\times$ one-block communication.
4. We evaluate a public 32/64-token policy on a recovered BOLT Microsoft Research Paraphrase Corpus (MRPC) checkpoint. The policy matches the unpruned accuracy, with 353 correct predictions out of 408 (86.52%) in clear fixed-point replays. The implementation completes one two-party 12-layer request at each public capacity and matches the corresponding clear prediction.

2 System Model and Deployment Setting

Roles and leakage. We assume a trusted dealer that does not collude with either online party and generates input-independent preprocessing. Two online parties, a model server P_0 and a client P_1, hold additive shares over $R = \mathbb{Z}_{2^\ell}$, where $\ell \geq 1$ is the ring word size. Unless stated otherwise, uppercase Roman letters denote tensors or algebraic objects, lowercase letters denote scalar dimensions or indices, and calligraphic letters denote sets. The maximum tensor dimensions, the set of supported shapes, and the operator graph are public before preprocessing. After the preprocessing shares have been distributed, the selected shape, and hence the active tensor dimensions, are disclosed before local derivation and execution of the fixed-shape online protocol. The application-prescribed output is released at the end of the execution. The selected shape may depend on private inputs or on the output of an earlier secure computation, but it must be chosen independently of the SSCC preprocessing randomness and its concrete encoding. We model the application layer as explicitly releasing that public shape. SSCC protects the subsequent computation conditional on this release and does not hide information encoded by the application's shape-selection policy. We consider a static semi-honest adversary that corrupts either online party. Hidden routing, malicious security, dealer corruption, and reuse of the same SSCC preprocessing across multiple online executions are outside our model.

Fixed-shape FSS-backend masked multiplication. For public dimensions (m, k, p), the SIGMA-style masked-multiplication protocol (Gupta et al., 2024) uses a dealer that samples $A \in R^{m\times k}$, $B \in R^{k\times p}$, and $Z \in R^{m\times p}$, and distributes additive shares of A, B, and $C = AB + Z$ to the parties. The online gate combines the masked operands with the parties' local shares of A, B, and C, producing the product masked by Z. The same Z links this multiplication record to any following truncation or nonlinear record: it masks the multiplication output consumed by that operation. SSCC preserves this linkage and produces a selected-shape view accepted by the unchanged fixed-shape online protocol, with the same dimensions and online messages as preprocessing generated directly for that shape.

Deployment comparison. Our main baseline pre-distributes fixed-shape preprocessing for every supported shape before shape selection. The on-demand baseline learns the public shape first and generates only the selected-shape preprocessing. SSCC targets deployments that require selected-shape preprocessing to be available immediately after selection. Its preprocessing share may be larger than one preprocessing share generated on demand for the selected shape.

2.1 Related Work

Function-secret-sharing preprocessing enables efficient secure computation with a dealer (Boyle et al., 2019), and SIGMA applies this paradigm to Transformer inference (Gupta et al., 2024). SIGMA's masked-matrix protocol assumes fixed tensor shapes and does not address specialization after its preprocessing has been distributed. Tensor Triples instead support matrix dimensions up to public bounds by changing both the correlation basis and the online multiplication primitive (Wu et al., 2023). They therefore fall outside the model of our lower bound.

At the preprocessing level, recent work classifies dependence on a circuit, topology, or concrete function (Brüggemann et al., 2026). This classification leaves open the timing point at which

topology-dependent fixed-shape preprocessing has already been distributed and a public shape is chosen later. General correlation transformations have been studied through non-interactive reductions, one-message secure reductions, and linear-algebraic share conversion (Agarwal et al., 2022; Ishai et al., 2023; Cramer et al., 2005; Ben David et al., 2024). The narrower question here is the minimum explicit augmentation after retaining a particular bilinear correlation, subject to a public share-linear decoder and an unchanged online protocol.

At the protocol and compiler level, nearby systems provide different forms of flexibility. FuseFSS compiles scalar nonlinear operators into a common FSS protocol and reduces key size (Ma et al., 2026), while SecureRouter privately selects among an encrypted model pool (Zhang et al., 2026). These systems address operator compilation and hidden routing, respectively, rather than post-distribution specialization of a retained bilinear correlation.

At the model level, prior work has already combined elasticity or adaptivity with private inference. Samragh et al. train one slimmable binary neural network for several channel widths and evaluate each width with a hybrid secret-sharing, oblivious-transfer, and garbled-circuit protocol (Samragh et al., 2021; Zhang et al., 2024). AdaPI shares model weights across nested weight and ReLU masks selected for different device budgets in a secret-sharing-based 2PC setting (Zhou et al., 2024). FHE-MENNs instead provide depth elasticity through early exits and show that an input-dependent exit decision can create a side channel (Folkerts & Tsoutsos, 2024). These works establish elastic private inference as prior art, but do not study post-distribution specialization of topology-dependent FSS preprocessing while preserving the fixed-shape online protocol.

Private Transformer inference also reduces cost through architecture, operator, and protocol design (Dong et al., 2023; Li et al., 2022; Gupta et al., 2024). Token pruning and input-dependent token dropping motivate a public capacity selected after private computation (Pang et al., 2024; Zhang et al., 2025; Cai et al., 2026). MatFormer is not a private-inference protocol, but its nested FFN widths and layerwise Mix'n'Match configurations provide a complementary elastic workload (Devvrit et al., 2024). These workloads leave open how to specialize already distributed preprocessing for public token-capacity and FFN-width choices while preserving the fixed-shape online protocol.

3 Minimum Augmentation for Retained Bilinear Correlations

We first isolate the bilinear object that makes local coordinate projection nontrivial. For $n \geq 1$, write $[n] = \{1, \dots, n\}$. Let $\mathcal{S}$ be the finite set of supported public shapes and $\mathcal{O}_s$ the target-coordinate set for shape s. Let maximum-shape preprocessing contain n_M scalar coordinates $M_j = \overline{G}_j(A, B) + Z_j$ for $j \in [n_M]$, where $\overline{G}_j$ is bilinear and every Z_j is an independent output mask. For $s \in \mathcal{S}$ and $o \in \mathcal{O}_s$, the target coordinate is $T_{s,o} = G_{s,o}(A, B) + \eta_{s,o}(Z)$, where $\eta_{s,o}$ is an R-linear form in the independent masks. The parties retain additive shares of every M_j and may receive r additional scalars $R_u = H_u(A, B) + \lambda_u(Z)$ for $u \in [r]$, where H_u is bilinear and λ_u is R-linear. After s is public, party P_i, $i \in \{0, 1\}$, must output a share of the form

$$L_{i,s,o}(A_i, B_i) + \sum_{j \in [n_M]} \alpha_{s,o,j}[M_j]_i + \sum_{u \in [r]} \beta_{s,o,u}[R_u]_i,$$

using the same public coefficients α, β at both parties and an arbitrary party-local function $L_{i,s,o}$. The two outputs must sum to $T_{s,o}$ for every value of the variables and every additive sharing. The decoder receives no new correlation and sends no message.

To account for both contraction coefficients and output masks, let V_{bil} be the R-module of complete A_0B_1 coefficient tensors, and let $V_Z = \bigoplus_{j \in [n_M]} Rz_j$ be the free module generated by the independent output masks. Set $U = V_{\mathsf{bil}} \oplus V_Z$. We identify every linear form in Z with its coefficient vector in V_Z. Define $m_j = (\Phi(\overline{G}_j), z_j)$ and $t_{s,o} = (\Phi(G_{s,o}), \eta_{s,o})$, where Φ extracts the complete A_0B_1 coefficient tensor. Let $W = \mathrm{span}_R\{m_1, \dots, m_{n_M}\}$, $Q = U/W$, and $\mathcal{T} = \mathrm{span}_R\{[t_{s,o}] : s \in \mathcal{S}, o \in \mathcal{O}_s\} \subseteq Q$. For $M \subseteq Q$, write $\kappa_R(M; Q)$ for the fewest elements of Q whose span contains M.

Theorem 1 (minimum fixed-shape preprocessing augmentation)**.** *In the model above, the minimum number of additional scalar correlations per party is*

$$r_{\min} = \kappa_R(\mathcal{T}; Q).$$

If R is a finite chain ring, including $\mathbb{Z}_{2^\ell}$, then $r_{\min} = \mu_R(\mathcal{T})$, the minimum number of module generators.

The lower bound applies a mixed-difference coefficient projection to any correct decoder in this function class. Party-local functions vanish, and arbitrary mixing of maximum-C coordinates vanishes in the quotient. Each explicit correction scalar contributes at most one quotient vector, so those vectors must cover $\mathcal{T}$. For the upper bound, choose $\kappa_R(\mathcal{T}; Q)$ elements of Q whose span contains $\mathcal{T}$. Each selected vector has a canonical global bilinear-plus-mask preimage, which the dealer additively shares. Public decoding then reconstructs every target modulo W, the span of the retained maximum-shape correlations. The output mask coordinates are essential because rescaling a maximum product coordinate also rescales its independent mask. Appendix A provides the full proof.

The theorem counts explicit ring elements. It is not a bit lower bound and does not cover nonlinear encoding, packed larger-alphabet values, pseudorandom correlation generators, additional interaction, or a different online primitive.

3.1 Nested Prefix Contractions

Consider maximum masks $A \in R^{N\times N}$ and $B \in R^{N\times p}$, with $C = AB + Z$. For a public prefix $b \le N$, cropping the first b output rows leaves

$$C_{[b],:} = A_{[b],[b]}B_{[b],:} + \underbrace{A_{[b],b:N}B_{b:N,:}}_{\Delta_b} + Z_{[b],:}. \tag{1}$$

The dealer shares Δ_b, and each party subtracts its share from the cropped C array. This produces the exact target shape-(b, b, p) relation.

Corollary 1 (tight nested-prefix correction)**.** *For H independent heads and a supported-capacity set $\mathcal{B}$ whose maximum is N, the minimum explicit correction cost is*

$$H \sum_{b\in\mathcal{B}\setminus\{N\}} bp$$

ring words per party. The omitted-tail construction in Equation (1) *reaches the bound.*

The tail classes form a free submodule because every nonmaximum capacity has a coordinate interval that no smaller tail contains, and output coordinates are independent. The proof is in Appendix B.

4 Projection and Correction Construction

SSCC organizes the preprocessing correlations into three classes. First, *projective correlations* use public coordinate maps into maximum-shape preprocessing. Second, *bilinear-corrected correlations* use the same maps plus a public linear combination of shared defect-basis values. Third, *shape-specific correlations* retain separately generated material and input/output mask translations. The third class retains separately generated correlations when topology-dependent nonlinear domains change with shape. In the joint construction, fresh branch-local masks and public remaps isolate these nonlinear records so that their unused branches admit conditional completion.

The dealer distributes to each party the maximum-shape correlations, the shape-specific correlations, and one additive share of every correction. After shape disclosure, each party gathers the projective coordinates, applies its correction shares, and appends the correlations specific to the selected shape. The resulting preprocessing is consumed for one public shape in one execution.

4.1 MHA Instance

Let the maximum token capacity be N. Token-wise FSS correlations use canonical coordinates in $[0, N)$, while pair-wise attention correlations use $[0, N)^2$. Q, K, and V projections, $QK^\top$, and the output projection keep their contracted dimensions, so their maximum-shape correlations are projective. The attention-times-V product changes from $(N, N)(N, d_h)$ to $(b, b)(b, d_h)$ and uses one Δ_b per head. Capacity-specific Softmax inverse tails remain shape specific because their input range changes. Separate mask translations connect each tail to the shared continuation.

4.2 Elastic SwiGLU FFN Instance

For n token rows, model width d, and maximum hidden width D, a bias-free SwiGLU block computes

$$U = \mathrm{SiLU}(XW_g) \odot (XW_u), \qquad Y = UW_d.$$

Here $X \in R^{n\times d}$, $W_g, W_u \in R^{d\times D}$, $W_d \in R^{D\times d}$, $U \in R^{n\times D}$, and $Y \in R^{n\times d}$. A public width $w \le D$ selects the first w hidden coordinates. The gate and up projections shrink an output axis and are projective. SiLU, truncation, and elementwise multiplication use coordinate-local correlations that are gathered into a fixed-shape $n \times w$ batch. The down projection shortens its contracted axis and therefore needs $\Delta_w = A_{[:,w:D]}B_{[w:D,:]} \in R^{n\times d}$. With t widths, the exact correction cost is $nd(t-1)$ words per layer by Lemma 1. For layerwise public choices, costs add across layers rather than multiplying by the number of configurations.

4.3 Joint Transformer Block Preprocessing

Let $\mathcal{B}$ contain public MHA capacities and $\mathcal{W}$ public FFN widths. The joint block stores

$$K^{\mathrm{joint}} = K^{\mathrm{common}} \,\|\, \{K_b^{\mathrm{cap}}\}_{b\in\mathcal{B}} \,\|\, \{K_w^{\mathrm{width}}\}_{w\in\mathcal{W}}, \tag{2}$$

with no family indexed by (b, w). Common material contains maximum-shape MHA and FFN records. Capacity families contain branch-isolated Softmax records, public boundary remaps, AV corrections, and capacity-local LayerNorm preprocessing. Width families contain maximum-row down-projection corrections, restricted to the first b rows after selecting (b, w). Residuals combine adjacent wire masks without pair-specific preprocessing. The selected preprocessing supports a fixed-shape block comprising MHA, residual addition, LayerNorm, FFN, residual addition, and LayerNorm, while storage grows with $|\mathcal{B}|+|\mathcal{W}|$. Appendix F provides a maximum-row LayerNorm selector that removes the capacity-local copies.

4.4 Correctness and Preprocessing Cost

The three correlation classes give a direct correctness invariant. A selector restricts every projective correlation to its public coordinate image, applies the shared defect value to every bilinear-corrected correlation, and appends the selected shape-specific correlation together with its boundary-mask translations. Projective restriction preserves the fixed-shape relation whenever the contracted coordinates do not change. Equation (1) restores that relation when they do change. Shape-specific correlations already satisfy the target relation. The translations identify each correlation's output mask with the next correlation's input mask. Consequently, their sequential composition has the same public dimensions, masked-wire relations, and online messages as preprocessing generated directly for the selected shape. This is an algebraic correctness statement. Security of the complete preprocessing share additionally requires the conditional-completion property below.

The same classification makes the storage claim explicit. Let P_N denote the correlations stored once at maximum shape, D_s the correction material for a nonmaximum shape s, L_s the shape-specific correlations for shape s, and H the public encoding overhead. Then the per-party SSCC preprocessing size is

$$|K^{\mathrm{SSCC}}| = |P_N| + \sum_{s\neq N} |D_s| + \sum_s |L_s| + |H|,$$

whereas pre-distributing independent fixed-shape preprocessing costs $\sum_s |K_s^{\mathrm{fs}}|$. SSCC saves only the overlap represented by P_N after paying every correction and shape-specific term. It therefore does not assume that all shape-dependent preprocessing can be derived from one maximum-shape preprocessing.

The measured MHA decomposition in Appendix G makes this boundary concrete. Maximum-capacity projective correlations account for 68,843,360 bytes per party, all three capacity-specific Softmax correlations remain as 37,135,832 shape-specific bytes, and the two attention-times-V corrections add 98,304 bytes. For the MatFormer FFN, coordinate-local nonlinear correlations reside inside the maximum-shape preprocessing, while the three down-projection corrections add 245,760 bytes. Thus the two instances use the same local conversion rule but factor different families of fixed-shape preprocessing.

4.5 Security by Conditional Completion

Let K_q be party P_q's complete preprocessing share for a public shape s, let $V_q^s = \mathsf{Sel}_{q,s}(K_q)$ be the selected preprocessing, let A_q^s contain all auxiliary state visible to that party, and let τ_q^s be its complete fixed-shape online view. Matching the marginal distribution of V_q^s is insufficient because the party also retains K_q. We therefore require one efficient conditional sampler satisfying

$$(K_q, A_q^s, V_q^s, \tau_q^s) \equiv (\mathsf{Complete}_{q,s}(A_q^s, V_q^s), A_q^s, V_q^s, \tau_q^s) \tag{3}$$

in the uniform-dealer-tape hybrid. The concrete dealer reaches this hybrid computationally under its joint pseudorandomness premise.

Theorem 2 (conditional-completion lifting)**.** *Fix a public shape s independent of preprocessing randomness. Suppose the fixed-shape-s protocol has one probabilistic polynomial-time (PPT) simulator for the complete (A_q^s, V_q^s, τ_q^s) view against either static semi-honest corruption, Equation* (3) *holds in the uniform-tape hybrid, and the real dealer view is computationally indistinguishable from that hybrid. If each complete preprocessing share is consumed once, then the SSCC execution reveals to either party no more than the fixed-shape-s execution and s.*

The theorem separates the protocol argument from the assumptions required of the implementation. The completer reconstructs the unselected preprocessing conditioned on the selected view, after which the fixed-shape simulator replaces the selected execution. The dealer-tape pseudorandomness step is the only implementation-specific transition in this lifting argument.

Corollary 2 (one-use prefix-elastic FFN security)**.** *Assume the fixed-width FFN premise above. Under the coordinate-factorization, fresh-sharing-pad, fresh down-mask, and joint dealer-tape pseudorandomness premises stated in Appendix E, the FFN selector satisfies Equation* (3)*. Consequently, for every public width w independent of preprocessing randomness, one-use FFN SSCC reveals no more than the fixed-width-w execution and w.*

The fixed-width premise imports SIGMA's preprocessing model and static semi-honest definition (Gupta et al., 2024, Secs. 2.2 and 2.4.3). Security of the enriched implementation view remains an explicit assumption. Appendix E defines the enriched view, constructs the completer, and records the implementation conditions needed for the random-tape transition.

The joint block additionally isolates every shape-specific nonlinear branch. Fresh branch-local input and output masks connect each branch to adjacent common records through public additive remaps. Conditioned on the selected branch, this interface permits independent completion of unused branches. Interface masks across MHA, residuals, LayerNorm, and FFN are sampled once in dataflow order.

Corollary 3 (one-use joint-shape block security)**.** *Fix a public $s = (b, w)$ independent of preprocessing randomness. Assume the fixed-shape-s block has a simulator for the complete party view, the branch-isolated MHA, capacity-local LayerNorm, and rectangular FFN records admit the conditional completers of Appendix F, all operator and adapter randomness domains are disjoint, and the complete dealer-tape vector is jointly pseudorandom for the fixed public trace. If the joint share in Equation* (2) *is consumed once, its execution reveals to either static semi-honest party no more than the fixed-shape-s block execution and s.*

Appendix F proves conditional completion through public remasking, rectangular FFN completion, branch-isolated MHA completion, and topological composition. Both corollaries exclude adaptive corruption, reuse, rollback, and malicious behavior. Functional tests of the joint block check the required algebraic relations but do not establish the randomness or lifecycle premises.

5 Instantiation for Transformer Preprocessing

We instantiate SSCC in the EzPC/GPU-MPC backend used by SIGMA. The instantiation preserves each fixed-shape operator and changes only how its preprocessing is assembled after the public shape is known. Projective records are gathered from a maximum-shape share, bilinear contractions receive the correction in Equation (1), and nonlinear records whose domains change with shape remain in separate branch families.

Table 1: Per-party preprocessing when fixed-shape preprocessing for every supported shape is distributed before selection. SSCC and separate fixed-shape byte counts use the same implementation within each row.

Instance	SSCC	Fixed-shape total	Reduction
BERT-tiny MHA	106,078,016	153,244,416	30.78%
GPT-2 first MHA	652,185,536	966,641,316	32.53%
MatFormer FFN	177,091,560	367,383,280	51.80%

For MHA, Q, K, and V projections, the $QK^\top$ product, and the output projection share maximum-capacity records. Capacity-specific Softmax records remain separate, while the attention-times-V product uses one omitted-tail correction per nonmaximum capacity and head. For a prefix-elastic SwiGLU FFN, the gate, up, activation, and elementwise-product records restrict by hidden coordinate. Only the down projection requires a width correction because its contracted dim changes.

The joint block connects these families through branch-local masks and public remaps. MHA capacity determines the token-row prefix and FFN width determines the hidden-coordinate prefix, so their correction records remain per-axis. Residual and LayerNorm interfaces reuse the selected token shape without introducing a capacity-by-width family. Appendix H records the implementation assumptions, randomness generation, validation, and one-use lifecycle checks.

6 EVALUATION

The evaluation uses three acceptance criteria: selected and fixed-shape executions must have identical outputs and online communication, axis factorization must reduce aggregate pre-distributed preprocessing under the stated accounting scope, with preparation intervals reported separately, and the public-capacity policy must match unpruned task accuracy. We test MHA and FFN components, their composition in one block, and a separate BERT-base full-model case study. Byte counts are per party unless stated otherwise. Component and joint-block measurements use an AMD EPYC 9754 system with an NVIDIA RTX PRO 5000 Blackwell GPU. The full-model case study runs separately on an Apple M4 Pro. We do not compare timing across these systems. Table 1 compares the per-party preprocessing storage of SSCC with pre-distributed fixed-shape preprocessing for the three component instances.

MHA Results For the BERT-tiny profile $H = 2$, $p = 64$, and $\mathcal{B} = \{32, 64, 128\}$, Corollary 1 yields 12,288 correction words, or 98,304 bytes per party with eight-byte encodings. This is exactly the correction material used by the implementation. At capacities 32, 64, and 128, both the BERT-tiny and pretrained GPT-2 first-block selected views have zero fixed-point mismatches and use the same online communication as fixed-shape execution. Their per-party preprocessing size reductions are 30.78% and 32.53%, respectively. These are one-block measurements. A calculated BERT-tiny baseline that also deduplicates fixed-weight Q/K/V and output-projection masks still yields a 30.30% reduction, but was not separately executed (Appendix G).

Trained MatFormer Results. The evaluated MatFormer instance uses a 12-layer, 1280-dim bias-free SwiGLU checkpoint with widths 512, 1024, 2048, and 3584. Across all 12 trained layers and four widths, selected and separately generated fixed-shape FFN executions agree with the clear fixed-point reference and communicate identically on recorded inputs. The comparison covers 983,040 output elements in 96 party-layer-width observations. An independently generated one-layer instance confirms one-use acceptance and rejection of modified preprocessing. These checks cover implemented consumption and integrity mechanisms, not a stronger security model. Table 2 reports bidirectional online communication.

The four fixed-shape preprocessing shares total 367,383,280 bytes per party, so SSCC is 51.80% smaller than their sum. Each individually generated fixed-shape share remains smaller than the SSCC share. On the GPU system, 20 fresh one-use observations per method and width yield median intervals from public width disclosure to output of 10.43 to 10.72 seconds for SSCC and 12.55 to 13.51 seconds for on-demand fixed-shape generation. SSCC excludes generation from this interval and receives no selected-width argument, whereas fixed-shape generation is included. Both methods exclude delivery, use no forced cache state, and alternate execution order (Appendix H).

Table 2: MatFormer SSCC preprocessing and a matched fixed-shape baseline. Fixed-shape generation reports the median and 95th percentile over 20 fresh sessions with both party processes concurrent on one GPU. Delivery is not included.

w	SSCC bytes	Fixed-shape bytes	Online bytes	Generation median/95th (s)
512	177,091,560	31,131,580	861,184	2.54/2.55
1,024	177,091,560	55,417,276	1,379,328	2.79/2.81
2,048	177,091,560	103,988,668	2,415,616	3.29/3.31
3,584	177,091,560	176,845,756	3,970,048	4.04/4.12

Table 3: Per-party persistent preprocessing size for the joint 3×3 block. The last column is the saving achieved by the factorized preprocessing relative to the listed baseline.

Preprocessing organization	Bytes per party	Saving
9 fixed-shape preprocessing shares	951,958,864	72.68%
Capacity-deduplicated fixed-shape preprocessing	524,229,968	50.39%
Factorized SSCC preprocessing	260,073,104	Not applicable

Joint 3×3 Transformer Block. We evaluate a BERT-style block with nonzero deterministic inputs and weights, model width 128, two heads, capacities $\{32, 64, 128\}$, and widths $\{128, 256, 512\}$. For all 9 pairs and both parties, selected and separately generated fixed-shape blocks have identical outputs and online bytes. Their maximum absolute deviation from the clear fixed-point reference is 17 integer units, below the predefined 32-unit threshold. The measured factorization contains zero capacity-by-width preprocessing bytes. A second execution in a reconstructed environment reproduced all nine configurations and regenerated identical results.

At shape $(32, 128)$, a 12-block synthetic chain on the GPU system matches a separately generated fixed-shape reference and reveals only the final activation. It communicates 21,695,184 bytes per party (12 times the one-block value). The synthetic workload also checks cross-block mask continuity and rejection of a second use, but limits the result to structural and interface conformance.

Full-Model Public-Capacity Case Study. With a recovered quantized BOLT BERT-base MRPC checkpoint, the first block uses capacity 128 before a revealed valid-token count selects capacity 32 or 64. The stable secret-shared compaction hides retained token identities, while capacity-aware ciphertext packing changes the physical HE shape and carries the selected capacity through the remaining eleven blocks. The 22/386 policy split and unpruned replay both make 353 correct predictions out of 408 (86.52%) on the Apple M4 Pro. Forcing capacity 32 on all 408 clear replays loses 3.68 accuracy points. The policy changes two predictions relative to the unpruned replay and lowers F1 by 0.0339 points despite identical accuracy. One complete two-party request per capacity runs all 12 blocks, using approximately 23 GB of online communication at capacity 32 and 48 GB at capacity 64. Both requests consume all 12 preprocessing records and match the clear fixed-point prediction. The capacity-64 request uses 31.69% less online communication than its paired unpruned sample-0 baseline; this is a single-run comparison. Appendix J reports wall time and single-run scope.

7 Limitations and Conclusion

Security assumes share-linear decoding, a public shape, one use, and static semi-honest parties, excluding rollback, malicious behavior, and dealer compromise. The 12-block chain is synthetic, MRPC has one request per capacity, and co-located timings exclude delivery and forced cache state. Across 9 shapes, SSCC matches fixed-shape execution and saves 50.39% storage over capacity-deduplicated preprocessing. The chain and 408-example MRPC study extend evaluation across depth and a trained model. Unlike pre-distributing every fixed shape, SSCC factorizes preprocessing across selector axes; unlike on-demand generation, it makes the selected preprocessing locally available after disclosure, at the cost of storing more than one selected-shape share. It does not reduce selected-shape online work.

Reproducibility Statement

Appendices H and I record the measurement scope, lifecycle checks, and independently reconstructed joint-block run. The anonymous supplement provides the data and scripts needed to recompute the reported storage, communication, and task-quality results. Optional commands rebuild the FFN and joint-block experiments. It omits the large third-party checkpoint and EzPC toolchain, whose setup and export requirements are documented in the appendix. The supplement also contains all 408 MRPC quality rows and two-party logs for one complete request at each public capacity. The verification script recomputes the task metrics, capacity counts, online communication, shape transitions, preprocessing consumption, output agreement, and the reported request distributions. It also checks the stated absence of intermediate disclosure and one-use consumption while preserving the distinction between component measurements and end-to-end latency.

Statement on the Use of Large Language Models

Large language models assisted with literature search and organization, developing and stress-testing the SSCC formulation, theorem statements, and proof arguments, designing and implementing experiments and artifact checks, interpreting results, preparing figures, and drafting and editing the manuscript. The authors directed the work, decided which suggestions to adopt, verified the resulting sources, arguments, implementations, numerical results, and claims, and take responsibility for the paper.

References

Pratyush Agarwal, Varun Narayanan, Shreya Pathak, Manoj Prabhakaran, Vinod M. Prabhakaran, and Mohammad Ali Rehan. Secure non-interactive reduction and spectral analysis of correlations. In *Advances in Cryptology: EUROCRYPT 2022*, 2022. URL https://eprint.iacr.org/2022/262.

Tamar Ben David, Varun Narayanan, Olga Nissenbaum, and Anat Paskin-Cherniavsky. New results in share conversion, with applications to evolving access structures. Cryptology ePrint Archive, Paper 2024/1781, 2024. URL https://eprint.iacr.org/2024/1781.

Elette Boyle, Niv Gilboa, and Yuval Ishai. Secure computation with preprocessing via function secret sharing. In *Theory of Cryptography Conference*, 2019. URL https://www.iacr.org/archive/tcc2019/11891211/11891211.pdf.

Andreas Brüggemann, Kasra EdalatNejad, Jan Filipp, Philipp Kusterer, and Thomas Schneider. It's not just a phase: Understanding function dependence in MPC preprocessing. In *Workshop on Privacy in the Electronic Society*, 2026. To appear. IACR ePrint 2026/2029.

Yifei Cai, Zhuoran Li, Yizhou Feng, Qiao Zhang, Hongyi Wu, Danella Zhao, and Chunsheng Xin. SecDTD: Dynamic token drop for secure transformers inference. *arXiv preprint arXiv:2603.13670, accepted at IEEE EuroS&P 2026*, 2026. URL https://arxiv.org/abs/2603.13670.

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. The-x: Privacy-preserving transformer inference with homomorphic encryption. *arXiv:2206.00216*, 2022.

Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure distributed computing. In *Theory of Cryptography Conference*, 2005. URL https://www.iacr.org/archive/tcc2005/3378_342/3378_342.pdf.

Devvrit, Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham Kakade, Ali Farhadi, and Prateek Jain. MatFormer: Nested transformer for elastic inference. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2310.07707.

Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Chen. Puma: Secure inference of llama-7b in five minutes. *arXiv:2307.12533*, 2023.

Lars Wolfgang Folkerts and Nektarios Georgios Tsoutsos. FHE-MENNs: Opportunities and pitfalls for accelerating fully homomorphic private inference with multi-exit neural networks. Cryptology ePrint Archive, Paper 2024/1099, 2024. URL https://eprint.iacr.org/2024/1099.

Kanav Gupta, Neha Jawalkar, Ananta Mukherjee, Nishanth Chandran, Divya Gupta, Ashish Panwar, and Rahul Sharma. Sigma: Secure gpt inference with function secret sharing. *Proceedings on Privacy Enhancing Technologies*, 2024(4):61–79, 2024. doi: 10.56553/popets-2024-0107. URL https://doi.org/10.56553/popets-2024-0107.

Yuval Ishai, Mahimna Kelkar, Varun Narayanan, and Liav Zafar. One-message secure reductions: On the cost of converting correlations. In *Advances in Cryptology: CRYPTO 2023*, 2023. URL https://eprint.iacr.org/2023/1225.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P Xing, and Hao Zhang. Mpcformer: fast, performant and private transformer inference with mpc. *arXiv:2211.01452*, 2022.

Yuhan Ma, Yong Li, and Stefan Schmid. FuseFSS: Efficient secure LLM inference with function secret sharing. *arXiv preprint arXiv:2606.09551*, 2026.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *IEEE S&P*, pp. 4753–4771, 2024.

Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow2: Practical 2-party secure inference. In *CCS*, pp. 325–342, 2020.

Mohammad Samragh, Siam Hussain, Xinqiao Zhang, Ke Huang, and Farinaz Koushanfar. On the application of binary neural networks in oblivious inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4630–4639, Los Alamitos, CA, USA, 2021. IEEE. URL https://openaccess.thecvf.com/content/CVPR2021W/BiVision/html/Samragh_On_the_Application_of_Binary_Neural_Networks_in_Oblivious_Inference_CVPRW_2021_paper.html.

Dongyu Wu, Bei Liang, Zijie Lu, and Jintai Ding. Efficient secure multiparty computation for multidimensional arithmetics and its application in privacy-preserving biometric identification. *IACR Cryptology ePrint Archive, Report 2023/1863*, 2023.

Xinqiao Zhang, Mohammad Samragh, Siam Hussain, Ke Huang, and Farinaz Koushanfar. Scalable binary neural network applications in oblivious inference. *ACM Transactions on Embedded Computing Systems*, 23(3):1–18, 2024. doi: 10.1145/3607192.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. Cipherprune: Efficient and scalable private transformer inference. *arXiv:2502.16782*, 2025.

Yukuan Zhang, Mengxin Zheng, and Qian Lou. SecureRouter: Encrypted routing for efficient secure inference. *arXiv preprint arXiv:2604.15499*, 2026.

Tong Zhou, Jiahui Zhao, Yukui Luo, Xi Xie, Wujie Wen, Caiwen Ding, and Xiaolin Xu. AdaPI: Facilitating DNN model adaptivity for efficient private inference in edge computing. In *IEEE/ACM International Conference on Computer-Aided Design*, pp. 110:1–110:9, New York, NY, USA, 2024. ACM. doi: 10.1145/3676536.3676694.

A Proof of the Augmentation Theorem

This appendix proves the minimum fixed-shape preprocessing augmentation theorem from the main paper.

Let R be a finite commutative ring. Each additional dealer scalar is restricted to a global bilinear form in (A, B) plus an R-linear form in Z. Expand every global bilinear form after $A = A_0 + A_1$ and $B = B_0 + B_1$. The linear map Φ records the coefficient of every $A_{0,\alpha}B_{1,\beta}$ cross term. This complete cross block is injective on global bilinear forms. Let V_{bil} denote the image of Φ. Let

$$V_Z = \bigoplus_{j \in [n_M]} Rz_j, \qquad U = V_{\mathsf{bil}} \oplus V_Z.$$

Define Ψ on the decoder's allowed functions by taking all mixed coefficients $A_{0,\alpha}B_{1,\beta}$ and all first differences in Z_j. Equivalently, Ψ first sets $Z = 0$ and applies Φ to the global bilinear part, then sets $A = B = 0$ and extracts the coefficient of each Z_j. It kills arbitrary sums of a P_0-local function of A_0, B_0 and a P_1-local function of A_1, B_1, while it maps every allowed global bilinear-plus-linear scalar to its element of U.

For maximum coordinates and targets, define

$$m_j = (\Phi(\overline{G}_j), z_j), \qquad t_{s,o} = (\Phi(G_{s,o}), \eta_{s,o}).$$

Write $W = \mathrm{span}_R\{m_j\}_{j=1}^{n_M}$, $Q = U/W$, and

$$\mathcal{T} = \mathrm{span}_R\{[t_{s,o}] : s \in \mathcal{S}, o \in \mathcal{O}_s\} \subseteq Q.$$

Decoder syntax. For target (s, o), party P_i outputs

$$L_{i,s,o}(A_i, B_i) + \sum_{j \in [n_M]} \alpha_{s,o,j}[M_j]_i + \sum_{u \in [r]} \beta_{s,o,u}[R_u]_i,$$

where $L_{i,s,o}$ is arbitrary and the coefficients α, β are public and identical at both parties. Perfect correctness requires the sum of the two outputs to equal $T_{s,o}$ for every A, B, Z, every dealer scalar in the allowed function class, and every additive sharing of all scalars.

Lower bound. Apply Ψ to a perfectly correct decoder and pass to Q. Every party-local A/B function vanishes. Every public linear combination of maximum-C coordinates lies in W and also vanishes. If the r additional scalar correlations have quotient images $q_1, \ldots, q_r \in Q$, correctness of every target requires

$$\mathcal{T} \subseteq \mathrm{span}_R(q_1, \ldots, q_r).$$

Thus $r \geq \kappa_R(\mathcal{T}; Q)$. Each permitted explicit dealer scalar has one well-defined image in Q.

Upper bound. Choose an optimal cover $q_1, \ldots, q_r \in Q$ and representatives $\widetilde{q}_1, \ldots, \widetilde{q}_r \in U$. The coefficient map from global bilinear forms in (A, B) plus public linear forms in Z to U is bijective. The dealer therefore computes the canonical bilinear-plus-mask preimage of each $\widetilde{q}_u$ and freshly additively shares its scalar value.

For every target, cover membership yields public coefficients $\lambda_{s,o,u}$ with

$$[t_{s,o}] = \sum_{u=1}^{r} \lambda_{s,o,u} q_u.$$

Lifting the equality to U leaves an element of W. Hence the exact target is a public linear combination of maximum coordinates and shared cover values. The canonical preimages contain no uncontrolled nonlinear kernel term, so the identity holds as a function for every mask choice. Therefore $r = \kappa_R(\mathcal{T}; Q)$ suffices.

If R is a finite chain ring, every submodule of an r-generated module needs at most r generators. Any ambient cover therefore has size at least $\mu_R(\mathcal{T})$, and the internal generators of $\mathcal{T}$ form a cover. Consequently $\kappa_R(\mathcal{T}; Q) = \mu_R(\mathcal{T})$.

Why output masks remain in the quotient. If the proof kept only bilinear coefficients, a decoder could appear to rescale a maximum product coordinate to cancel a defect. The same operation also rescales its independent output mask. Keeping $(\Phi(\overline{G}_j), z_j)$ intact records that obstruction and grants the decoder all valid maximum-C mixing without granting free mask changes.

B TIGHTNESS FOR NESTED PREFIXES

Fix one head and one output coordinate (i, j). For every nonmaximum capacity b, the maximum-shape correlation differs from the fixed-shape prefix correlation by

$$\delta_{b,i,j} = \sum_{h=b}^{N-1} A_{i,h} B_{h,j}.$$

Order the capacities increasingly. For any nonzero public linear combination of their quotient classes, take the smallest capacity $b^\star$ with a nonzero coefficient. The monomial $A_{i,b^\star} B_{b^\star,j}$ occurs in $\delta_{b^\star,i,j}$ and in no tail with larger starting point. It therefore cannot cancel. Distinct output coordinates and heads use disjoint coefficient coordinates. The tail classes are thus linearly independent over $\mathbb{Z}_{2^\ell}$ and form a free submodule of rank

$$H \sum_{b \in \mathcal{B} \setminus \{N\}} bp.$$

The minimum-augmentation theorem provides the lower bound. Sharing each omitted-tail matrix Δ_b achieves the matching upper bound.

C CONCRETE FIXED-SHAPE ALGEBRA

Suppose the online parties know masked operands $\widehat{X} = X + A$ and $\widehat{Y} = Y + B$. The concrete masked multiplication is equivalent to local shares

$$U_0 = C_0 + \widehat{X}(\widehat{Y} - B_0) - A_0 \widehat{Y},$$
$$U_1 = C_1 - \widehat{X} B_1 - A_1 \widehat{Y}.$$

Using $C_0 + C_1 = AB + Z$ yields

$$U_0 + U_1 = XY + Z.$$

For a prefix contraction, parties replace the cropped C_q share by

$$C_q^{(b)} = (C_q)_{[b],:} - \Delta_{b,q}.$$

The two corrected shares sum to $A_{[b],[b]} B_{[b],:} + Z_{[b],:}$, so the unchanged fixed-shape online identity continues to hold.

D INSTANCE MAPS

D.1 MHA

For a capacity b, the Q, K, and V projection correlations gather the first b row groups. The $QK^\top$ correlation gathers the $b \times b$ pair coordinates. The attention-times-V correlation gathers cropped $A/B/C$ arrays and subtracts the shared omitted tail. The output projection again gathers rows. Softmax comparison and table material whose domain is a coordinate product is gathered and repacked. The capacity-dependent inverse tail is generated independently and linked through fresh input and output mask translations.

The BERT-tiny correction accounting is

$$2(32 + 64)64 = 12{,}288$$

64-bit words per party. The observed 98,304 bytes match this expression.

D.2 SWIGLU FFN

For each hidden neuron, the gate and up projection outputs, SiLU correlation, and elementwise multiplication correlation share one public hidden-coordinate index. The selector gathers all values for the first w neurons in the order consumed by the fixed-shape protocol. The down projection

keeps $n \times d$ outputs but contracts over w rather than D. It therefore consumes one $n \times d$ correction for each nonmaximum width.

For the evaluated profile, $n = 8$, $d = 1280$, and three widths are below the maximum. The exact correction is

$$8 \cdot 1280 \cdot 3 = 30{,}720$$

64-bit words, or 245,760 bytes per party. The SSCC preprocessing share is 177,091,560 bytes and includes 176 bytes of public encoding metadata.

Lemma 1 (tight elastic-FFN correction)**.** *For supported widths $w_1 < \cdots < w_t = D$, one $n \times D$ by $D \times d$ down projection needs exactly $nd(t-1)$ explicit correction words in the model of the main paper's augmentation theorem. Independent layers add their costs.*

Proof. Fix an output coordinate (i, j). The class for width $w_r < D$ is represented by the tail

$$\delta_{r,i,j} = \sum_{h=w_r}^{D-1} A_{i,h} B_{h,j}.$$

In a nonzero linear combination, choose the smallest r with a nonzero coefficient. The monomial $A_{i,w_r} B_{w_r,j}$ occurs in $\delta_{r,i,j}$ and in no tail with a larger starting point, so it cannot cancel. The $t-1$ classes are therefore free. Distinct (i, j) pairs use disjoint coefficient coordinates, giving rank $nd(t-1)$. Separate layers use disjoint mask and coefficient variables, so their quotient submodules form a direct sum. Sharing each tail supplies the matching upper bound. □

E Security of One-Use Shape-Selective FFN Preprocessing

This appendix instantiates the conditional-completion lifting theorem for the prefix-elastic FFN construction. It first fixes the imported fixed-width security premise, then proves conditional completion in a uniform-tape hybrid and states the pseudorandomness transition to the concrete dealer. The next section provides the corresponding joint-block argument.

E.1 Security Experiment and Simulation Order

Let the supported public widths be $\mathcal{W} = \{w_1 < \cdots < w_t = D\}$. Party P_q receives its complete preprocessing share K_q before the public width w is disclosed. It then computes $V_q^w = \mathsf{Sel}_{q,w}(K_q)$ and runs the fixed-width-w protocol, obtaining online view τ_q^w. Let A_q^w contain every visible auxiliary value: the corrupted party's input and local coins, public dimensions and width, the public full masked input and maximum-width model values, and any public session metadata. The real ordered final view is

$$(K_q, A_q^w, V_q^w, w, \tau_q^w).$$

The fixed-width security premise is an explicit assumption. For either corruption role and every $w \in \mathcal{W}$, a single PPT simulator, given the corrupted party's input, prescribed output, and fixed-width public leakage, emits $(\widetilde{A}_q^w, \widetilde{V}_q^w, \widetilde{\tau}_q^w)$ indistinguishable from the complete real fixed-width view. Here τ_q^w includes sent and received messages, residual local state, output, and abort behavior. We do not derive this premise from the SSCC theorem.

Fixed-width functionality and leakage. For clarity, the imported premise refers to the fixed-width masked FFN tensor program $\mathcal{F}_w^{\mathrm{FFN}}$ over $\mathbb{Z}_{2^\ell}$, including its faithful truncations and coordinatewise SiLU and multiplication. Its public parameters are (n, d, w, ℓ, f) and the fixed operator graph. The ideal execution receives the two parties' prescribed private inputs, evaluates that program, and releases exactly the output assigned by the application (a masked output or its designated unmasking), together with explicit abort. The fixed-width public leakage L_w consists of the public parameters and width w, public session metadata, the public masked input, the public masked maximum-width model values used as auxiliary input, and the selected public masked weights. It excludes clear inputs, clear weights, all wire masks, unused preprocessing correlations, and the dealer seed. Thus the simulator is specified by

$$\mathsf{Sim}^{\mathrm{fs}}(1^\lambda, q, x_q, \mathcal{F}_w^{\mathrm{FFN}}(x_0, x_1)_q, L_w) \Rightarrow (\widetilde{A}_q^w, \widetilde{V}_q^w, \widetilde{\tau}_q^w).$$

The quantifiers require one algorithm for both corruption roles and all $w \in \mathcal{W}$. They do not permit a simulator chosen after seeing a particular transcript.

Fixed-shape security premise. SIGMA defines $\Pi_F = (\mathsf{Gen}_F, \mathsf{Eval}_F)$ in the preprocessing model: Gen_F takes input and output wire masks, while Eval_F takes a masked input and the party's key and produces shares of the masked output. It states simulation security against one static semi-honest PPT corruption and composes its secure primitive protocols sequentially (Gupta et al., 2024, Secs. 2.2, 2.4.3, and 5). We instantiate F with the width-w FFN graph and identify its party key with V_q^w, its public masked and auxiliary values with A_q^w, and its evaluation messages and residual state with τ_q^w. The cited result supplies the preprocessing model and fixed-width operator composition. Our enriched tuple also contains maximum-width model values, validation state, checksum validation, and abort behavior that are not part of SIGMA's statement. Security of this enriched fixed-width view therefore remains an explicit premise of Corollary 2 rather than a consequence of the citation.

The theorem uses stand-alone final-view simulation. The simulator may construct the joint view after the ideal functionality discloses w, even though the tuple records offline delivery first. This is sound because, conditioned on the parties' inputs and the disclosed result, w is independent of the FFN preprocessing randomness. The theorem does not allow an environment to inspect K_q and choose w as a function of its value. Preprocessing for an earlier private width-selection computation must therefore use independent randomness.

Definition 1 (conditional completion)**.** *For every q and w, an exact completer is a PPT algorithm satisfying*

$$(K_q, A_q^w, V_q^w, \tau_q^w) \equiv (\mathsf{Complete}_{q,w}(A_q^w, V_q^w), A_q^w, V_q^w, \tau_q^w), \tag{4}$$

where the right-hand (A_q^w, V_q^w, τ_q^w) tuple is sampled by the real fixed-width-w execution. The completed preprocessing share must select to the supplied V_q^w exactly. This joint condition also requires the completed unused state to be conditionally independent of the selected online view given (A_q^w, V_q^w).

If Equation (4) holds, invoke the fixed-width simulator and return

$$(\mathsf{Complete}_{q,w}(\widetilde{A}_q^w, \widetilde{V}_q^w), \widetilde{A}_q^w, \widetilde{V}_q^w, w, \widetilde{\tau}_q^w).$$

Fixed-width indistinguishability is preserved by PPT post-processing, and conditional completion then replaces the completed preprocessing share by the real one while preserving every equality visible between its selected and unused correlations. Thus it remains to construct the completer.

E.2 Pseudorandomness Transition for the Concrete Dealer

The implementation has no per-correlation DRBG labels. For one device, a fresh 256-bit dealer seed initializes HMAC-SHA-256 states labeled `cpu-device-0` and `gpu-device-0`. A fixed public call trace parses distinct output blocks, updating the state between calls. The same seed and trace are replayed in the party-0 and party-1 dealer invocations. The party argument selects opposite shares produced from the same global masks and pads.

Let $G(\sigma) = (G_{\mathrm{cpu}}(\sigma), G_{\mathrm{gpu}}(\sigma))$ denote the complete pair of output traces consumed by one dealer invocation with the fixed public call trace, and let $F = (F_0, F_1)$ be the deterministic map that replays those traces to generate both party shares. Under joint pseudorandomness,

$$F(G(\sigma)) \approx_c F(U_{L_{\mathrm{cpu}}}, U_{L_{\mathrm{gpu}}}). \tag{5}$$

This single hybrid retains the required replay across parties. The implementation does not derive independent named substreams. In the uniform-tape hybrid, disjoint parsed blocks are independent. Masking a uniform word to ℓ low bits produces a uniform element of $\mathbb{Z}_{2^\ell}$, and full AES blocks remain uniform. Outer model, input, and output masks are sampled independently from the operating system. They may equivalently be included in the joint-PRG assumption.

E.3 Complete FFN Correlation Factorization

Fix n token rows, external dimension d, and maximum hidden width D. The preprocessing share contains gate, up, SiLU, elementwise multiplication, and down-projection correlations followed by

one $n \times d$ correction share for every nonmaximum width. The following list maps every concrete key component to a completion step.

Gate and up projections. Each `GPUMatmulKey` contains arrays A, B, C and a `TrFloor` key. The input-shaped A array is fixed-shape. The selector gathers the first nw coordinates of the hidden-column B/C arrays. For each selected output it also gathers the complete truncation correlation: public `bin/shift/bout/N`, the least-significant-bit (LSB) `GPUDPFKey`, its packed mask, two correction words, and, when present, the corresponding most-significant-bit (MSB) `TrCorr` values.

SiLU. The `GPUGeluKey` contains a public bitwidth and a `TrWithSlack` correlation. It also contains a DReLU `GPUDPFKey` and packed mask, a packed interval mask, two four-entry mux arrays, a lookup-table (LUT) `GPUDPFKey`, its packed `maskU`, five LUT-select arrays, and five final-select arrays. The selector unpacks each packed bit field, gathers the first nw complete element correlations, restores zero padding, and rewrites public counts.

Elementwise multiplication. The `GPUMulKey` contains arrays a, b, c and a complete `TrWithSlack` correlation. All four families are gathered on the same nw element indices.

Down projection. Its `GPUMatmulKey` contains the hidden-indexed first operand A, the hidden-row weight mask B, fixed-size correction C, and a fixed-size `TrFloor` correlation. Selection gathers the first nw entries of A and the first wd entries of B, retains the full truncation correlation, and replaces C_q by $C_q^{(w)} = C_q - R_{q,w}$. The evaluated generator samples the maximum-shape pre-truncation down mask with `randomGEOnGpu` from the session GPU tape. The same mask is retained by every selected width. It is not a public deterministic value.

Public encoding metadata. The concrete encoding includes public dimensions, the supported width set, party identity, session identifier, component lengths, correction lengths, total length, and a SHA-256 digest. These are deterministic functions of public parameters, the encoded preprocessing, and the public session identifier.

Auxiliary visible state. The implementation also uses full public `masked-w-gate`, `masked-w-up`, `masked-w-down`, and `masked-input` arrays. They are included in A_q^w. If the corrupted party's own input determines one corresponding global outer mask, each local share of that mask is still uniform because `writeShares` uses a fresh pad. Product-correction shares use separate pads, and every downstream FSS input mask is freshly sampled independently of the clear input and weights. If the corrupted party does not know the other input, the same translation argument makes its local share independent of the public masked array. Thus the suffix local-key marginal below is unchanged after conditioning on A_q^w. The fixed input-shaped values already belong to (A_q^w, V_q^w).

Correspondence with the Implementation. The evaluated implementation parses complete correlation records, applies the selectors described below, and preserves the fixed-shape operator interfaces. The artifact retains the source-level mapping.

For distributed point function (DPF) or distributed comparison function (DCF) input bitwidths above seven, `keyGenDPFTreeKernel` assigns one CUDA thread and two fresh AES seeds to each element. Its correction words are striped in arrays, but no key-generation equation combines different elements. `gpuKeyGenBatchedDPF` only partitions elements to bound memory. The evaluated GeLU profile has one public batch block at every supported width, so the stated decode/gather/encode map is a bijection on complete element correlations. Additive and XOR share writers use a fresh uniform pad per call in this hybrid. No small-table DPF branch is exercised by the evaluated SiLU profile.

Consequently the local preprocessing share factors, conditioned on its fixed external values, into independent hidden-coordinate atoms. One atom contains both gate and up projection columns, all n linked SiLU correlations, all n multiplication correlations, and the corresponding down input and weight row. Grouping the entire atom is necessary because intermediate global masks are passed from one primitive to the next, even though the implementation freshly reshares their local representations at several boundaries.

The same source facts prove distributional equivalence between selected and independently generated fixed-shape preprocessing rather than merely functional equality. Under uniform tapes, an N-element DPF/DCF key is the product of N per-element key-generation maps followed by a pub-

lic encoding permutation. A fixed-shape nw-element call is the same product on fewer independent seeds. Matmul and multiplication arrays differ only by fresh-share translations. Forced target masks in the selector are themselves uniform with the fixed-shape target-mask distribution. The down pre-truncation mask is sampled once from a fresh uniform tape block and retained by the affine correction. Rewriting public counts and packed padding therefore maps the maximum prefix exactly to the fixed-width law for the evaluated one-block profile.

E.4 Completion Algorithm

Given (A_q^w, V_q^w), the completer performs the following steps.

1. Parse and validate every public count, component dimension, and packed padding rule. Copy all fixed-shape boundary fields.
2. Apply the inverse documented coordinate map to embed every supplied gate, up, SiLU, multiplication, and down-prefix field at hidden indices $j < w$.
3. For every $j \geq w$, run the source-derived party-local atom sampler, conditioned on A_q^w, using fresh uniform coins for a complete atom spanning gate/up, SiLU, multiplication, and down operations. Draw each intermediate global mask once, pass it to every consuming primitive, retain party q's fields, and insert them into the maximum-shape coordinates. Fresh sharing pads sample fields whose global outer masks are not determined by the corrupted party.
4. If $w < D$, sample $R_{q,w} \leftarrow R^{n \times d}$ uniformly and set
$$C_q = C_q^{(w)} + R_{q,w}. \tag{6}$$
Sample every other $R_{q,w'}$ independently. If $w = D$, copy C_q and sample every nonmaximum correction independently.
5. Restore the public dimensions, deterministic padding, component lengths, and digest of the concrete encoding.

To justify Step 4, each global correction and each overlapping global tail defect is shared with a separate uniform pad. For either party the tuple $(C_q, \{R_{q,w'}\}_{w'})$ is therefore product-uniform. The change of variables
$$(C_q, R_{q,w}) \longleftrightarrow (C_q - R_{q,w}, R_{q,w})$$
is a bijection. Equation (6) is exactly the conditional distribution given the selected correction $C_q^{(w)}$.

The selected preprocessing has the fixed-width distribution: fixed-shape values are unchanged, variable values are a prefix of the independent atom product, and the corrected down term has the fixed-width-w relation under the same uniformly distributed pre-truncation down mask. Steps 2 and 3 sample the exact conditional product law. Step 4 samples the exact affine conditional law, and Step 5 is deterministic. The suffix atoms are absent from both parties' selected preprocessing. In Step 4, changing the free correction share changes the corresponding maximum shares, while $C_q - R_{q,w}$ and the honest party's selected correction remain fixed by cancellation. Thus the completion coins do not change the selected two-party execution or its transcript once (A_q^w, V_q^w) is fixed. In particular, this construction rules out a generic completion in which an unused key bit is correlated with the transcript: every suffix atom is absent from both parties' selected preprocessing, and the only cross-width affine free variable cancels from both selected corrections. This proves the joint Equation (4) in the uniform-tape hybrid. Step 2 also proves exact selector consistency.

E.5 Hybrid Conclusion

Let $\mathcal{A}$ distinguish the real SSCC-augmented execution from the ideal one. Hybrid H0 is the implemented dealer. H1 replaces the complete joint CPU/GPU traces by uniform tapes. Equation (5) bounds this step by a two-stream PRG distinguisher. Conditional completion and the fixed-width simulator replace H1 by the ideal view. Therefore there are PPT reductions $\mathcal{B}, \mathcal{C}$ such that
$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{full}} \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{fw}} + \mathrm{Adv}_{\mathcal{C}}^{2\text{stream-PRG}}.$$

This proves the one-use prefix-elastic FFN security theorem. Independent layers compose sequentially when their masks and tapes are independent. If one seed drives several layers, the same proof instead replaces the complete multi-layer trace once and requires a fixed public parse into disjoint blocks.

F Security of the Joint SSCC Transformer Block

Fix a party P_q and a public shape $s = (b, w)$. Let K_q be its complete joint preprocessing share, $V_q^s = \mathsf{Sel}_{q,s}(K_q)$ its selected fixed-shape view, A_q^s all other visible auxiliary state, and τ_q^s the complete fixed-shape online view. As in the FFN conditional-completion definition above, the required property is an efficient conditional sampler satisfying

$$(K_q, A_q^s, V_q^s, \tau_q^s) \approx_c (\mathsf{Complete}_{q,s}(A_q^s, V_q^s), A_q^s, V_q^s, \tau_q^s). \tag{7}$$

Equality holds for share-linear fields in the uniform-tape hybrid. Computational indistinguishability accounts for FSS key privacy and the joint dealer-tape PRG hybrid.

F.1 Branch-Isolating Public Remasks

Use the convention $\widehat{x} = x + r$. Suppose a common record outputs mask r, a shape-specific nonlinear record expects input mask a and returns mask c, and the next common record expects mask t. Publish

$$\delta^{\mathrm{in}} = a - r, \qquad \delta^{\mathrm{out}} = t - c.$$

The online parties locally replace $\widehat{x}$ by $\widehat{x} + \delta^{\mathrm{in}} = x + a$ before the branch and replace its output by $\widehat{y} + \delta^{\mathrm{out}} = y + t$ afterward.

Lemma 2 (branch-isolating remask)**.** *If a, c are fresh uniform masks independent of r, t, then the public remaps and either party's locally padded mask shares are simulatable conditioned on the adjacent common masks. If the branch preprocessing has a one-party fixed-shape key simulator, every unused branch can be completed independently conditioned on the selected branch.*

Proof. The maps $(r, a) \mapsto (r, a - r)$ and $(t, c) \mapsto (t, t - c)$ are bijections. Fresh additive pads leave either party's local shares uniform after the public difference is fixed. Invoke the fixed-shape key simulator under the sampled branch-local masks. Because different branches use disjoint masks and coins, this procedure samples their joint conditional law. □

For capacity b, let R_b be the selected prefix of the common QK output mask and U_b the attention mask expected by the common AV record. Generate the capacity-b Softmax record under fresh masks S_b, T_b and publish $S_b - R_b$ and $U_b - T_b$. Common QKV, QK, AV, and output-projection records are completed by row or row-pair restriction. The selected AV correction is completed by the affine bijection used in Equation (6). Lemma 2 then isolates all unselected Softmax keys.

The evaluated block uses the same conservative construction at both LayerNorm positions: for each capacity b, it generates ordinary fixed-b preprocessing under fresh input/output masks and connects it to adjacent canonical masks by public remaps. Assuming the ordinary fixed-b LayerNorm key simulator, Lemma 2 completes every unused capacity branch. This family is indexed only by b.

An alternative representation stores one maximum-row preprocessing share at each LayerNorm position. Conditioned on its shared masked scale and bias, the evaluated generator is a product of complete row atoms: each atom contains its mean truncation, square correlations, inverse-square-root path, affine records, and output masks. Restricting the first b atoms therefore has the fixed-shape fixed-b law, while suffix rows can be sampled independently. The serialized row selector is not used by the reported block implementation.

F.2 Rectangular FFN Completion

The selected hidden-coordinate set is

$$S_{b,w} = \{(r, j) : 0 \le r < b,\ 0 \le j < w\} \subseteq [B] \times [W].$$

Here B and W are the maximum row and hidden-coordinate counts, respectively. Group preprocessing into input-row atoms, model-coordinate atoms, linked nonlinear-path atoms for each (r, j), and output-row atoms. A nonlinear-path atom contains every projection-output, activation, multiplication, truncation, and internal mask linked at that coordinate. Given the selected rectangle, the completer embeds its atoms, samples missing rows, and samples complete suffix-width atoms. For selected output rows it samples the correction share $D_{w,q}[r]$ and recovers the common down-correlation share as $C_q[r] = C_q^{(w)}[r] + D_{w,q}[r]$.

Lemma 3 (rectangular FFN completion)**.** *Under the documented direct-product coordinate maps, fresh sharing pads, a fresh maximum down pre-truncation mask, and independent uniform dealer coins, the algorithm above samples the exact party-local maximum-FFN and all-width-correction distribution conditioned on the selected (b, w) view. For the concrete DRBG, the same conclusion is computational under the joint-output PRG premise of Corollary 2.*

Proof. Complete atoms outside $S_{b,w}$ use independent coins conditioned on their incident boundary masks. For a selected row, the change of variables $(C_q, D_{w,q}) \leftrightarrow (C_q - D_{w,q}, D_{w,q})$ is a bijection. Other correction shares remain independently padded. The width correction is stored for all B rows and selection retains its first b, so completion introduces no (b, w)-indexed object. □

F.3 Topological Composition

Represent the block preprocessing as a directed acyclic graph (DAG) whose edges are masked-wire interface states and whose nodes are operator records. Process nodes in the order

$$\mathrm{MHA} \to \mathrm{Residual}_1 \to \mathrm{LayerNorm}_1 \to \mathrm{FFN} \to \mathrm{Residual}_2 \to \mathrm{LayerNorm}_2.$$

Each edge mask is sampled once. Conditioned on incident edges, node dealer coins are independent, and the node completer samples exactly its absent fields. Public remaps are edge adapters covered by Lemma 2. Residual nodes combine masks deterministically.

Lemma 4 (topological completion)**.** *If every node has the conditional sampler above and absent fields affect the selected transcript only through already fixed selected edges, processing the DAG in topological order samples the complete joint preprocessing distribution conditioned on (A_q^s, V_q^s, τ_q^s).*

Proof. Factor the dealer distribution in topological order into the conditional law of each newly introduced edge and node record. At each step the construction uses exactly that conditional law while retaining selected values. Induction yields the correct conditional distribution for every processed prefix. The final prefix is Equation (7). □

To prove Corollary 3, replace the complete vector of implementation dealer-tape outputs by independent uniform tapes. The branch-remasking construction, MHA projection and correction, the two capacity-local LayerNorm completers, and Lemma 3 provide node-level completion. Lemma 4 composes them into $\mathsf{Complete}_{q,s}$. The fixed-shape block simulator then replaces the selected view. Reversing the hybrids bounds distinguishing advantage by the joint-PRG, fixed-shape block, and branch-key privacy advantages.

The corollary is conditional on a secure fixed-shape full-block protocol, disjoint randomness domains, a shape independent of preprocessing randomness, and one-use consumption. It excludes adaptive corruption, malicious parties, rollback, and dealer compromise. The functional evaluation checks the branch-isolating algebra but does not establish these security premises.

G Preprocessing Cost Decomposition

For BERT-tiny MHA, shared maximum-capacity correlations occupy 68,843,360 bytes, three capacity-specific Softmax keys occupy 37,135,832 bytes, two AV corrections add 98,304 bytes, and public alignment adds 520 bytes. The independent component sum is 153,244,416 bytes per party and excludes a validation-only duplicate AV record. A calculated baseline that also deduplicates fixed-weight Q/K/V and output-projection masks is 152,195,840 bytes, leaving a 30.30% reduction. This normalized baseline was not separately executed.

Table 4 reports the full component breakdown.

Table 4: MHA preprocessing bytes per party for capacities 32, 64, and 128. The Fixed-shape total column sums measured key components across separately generated branches. The SSCC column counts the maximum-shape, correction, and shape-specific correlations.

Component	Fixed-shape total	SSCC
Q/K/V	60,308,672	34,181,184
$QK^\top$	15,337,192	11,568,800
AV correlation and correction	802,816	622,592
AV truncation	19,403,968	11,087,936
Output projection	20,255,936	11,481,152
Softmax	37,135,832	37,135,832
Preprocessing bytes	153,244,416	106,077,496
Public metadata and alignment	0	520
Total	**153,244,416**	**106,078,016**

For the MatFormer FFN, the SSCC preprocessing share is a factorized collection of correlations rather than one shape-independent correlation. Table 5 separates the maximum-shape correlations from the width-specific corrections.

Table 5: MatFormer FFN preprocessing bytes per party. Coordinate-local nonlinear correlations reside inside the maximum-shape preprocessing. Only down-projection tails add width-specific correction storage.

Component	Bytes
Maximum-width shared correlations	176,845,624
Three width corrections	245,760
Header and checksum metadata	176
Total	**177,091,560**

H Artifact and Measurement Details

For each of four widths, the SSCC experiment checks party-specific data access, public metadata and preprocessing checksums, selected output, fixed-shape output, final masked output, one-use rejection, and corruption rejection. The fixed-shape on-demand baseline uses the same HMAC-DRBG construction, checkpoint export, clear fixed-point reference implementation, and encoding instrumentation. The strengthened latency baseline launches P0 and P1 concurrently on one GPU for 20 fresh sessions per width. Its wall clock includes preprocessing generation, SHA-256 hashing, and local encoding.

The SSCC preparation benchmark runs 20 repetitions per party and width. It measures reading and SHA-256 validation, local derivation of the selected preprocessing, and public masked-model loading. Warm-cache measurements and best-effort cold reads after requesting page-cache eviction are reported separately.

The matched fixed-shape receiver benchmark uses the same repetition and cache protocol, but reads and validates only the selected fixed-shape preprocessing. Across widths, the median warm/cold CPU-preparation times range from 0.13/0.14 to 0.45/0.49 seconds. The largest 95th percentile is 0.50 seconds.

The 100-Mbps and 3-Gbps values in the paper use

$$T_{\mathsf{deliver}} = \frac{8 \cdot \mathsf{bytes}}{\mathsf{bandwidth}}$$

and assume parallel delivery to the parties. They are lower bounds derived from preprocessing bytes rather than network experiments.

A separate, unthrottled cross-system diagnostic launches two concurrent `scp` processes from the GPU system to the CPU system. It uses pre-generated preprocessing shares, so it includes SSH setup, encryption, transfer, and local write completion, but excludes generation, receiver parsing, and streaming. Each width has 20 repetitions:

Width	Two-party bytes	Median delivery (s)	95th percentile (s)
512	62,263,160	9.76	14.87
1,024	110,834,552	16.62	24.62
2,048	207,977,336	22.61	33.41
3,584	353,691,512	51.32	59.65

This is a system-specific transfer measurement, not a streamed end-to-end percentile. Generation, delivery, and receiver-preparation quantiles are therefore not added and reported as one observed latency.

Continuously timed co-located requests. On the GPU system, we measure 20 requests per method and width from disclosure of the public width until both parties produce matching outputs. Every observation uses fresh preprocessing and consumes one preprocessing file per party. Every request uses distinct preprocessing files. Method order alternates by repetition. The SSCC interval includes preprocessing-file loading, checksum validation, local selection, peer setup, and online execution. Fresh SSCC generation occurs before this interval and receives no selected-width argument. The on-demand interval starts before fresh concurrent two-party generation and includes the same subsequent stages.

Width	SSCC request median/95th (s)	On-demand request median/95th (s)	SSCC generation median (s)	Fixed-shape generation median (s)
512	10.505/10.578	12.546/12.712	3.266	2.442
1,024	10.432/10.608	12.763/12.861	3.263	2.529
2,048	10.549/10.721	13.008/13.203	3.267	2.698
3,584	10.719/10.795	13.511/13.693	3.274	2.934

Delivery is excluded. Both methods load freshly written files, but the cache state is not forced. A 0.05-second party-launch stagger, applied to both methods, is included. The result therefore measures a co-located interval from public width disclosure to output under the stated lifecycle rather than WAN service latency.

A matched pre-positioning diagnostic repeatedly transfers the same representative pair of 177,091,560-byte SSCC shares from the GPU system to the CPU system. Across 20 repetitions, its median and 95th percentile are 11.90/14.22 seconds. The repeated bytes are used only to measure transport. An actual session requires fresh one-use preprocessing. Delivery is thus a per-session cost before shape disclosure and cannot be amortized by consuming the same preprocessing in several requests. Network rates differed between the on-demand and SSCC transfer campaigns, so we do not infer a transfer speedup. The earlier co-located component values, ranging from 2.68 to 4.54 seconds for on-demand generation and from 0.45 to 0.62 seconds for SSCC, form the empirical Cartesian product of generation and preparation distributions measured on the GPU system. The derived combinations are not additional observations. They exclude streaming and are not continuous transactions.

At the median, the parallel dealer writes between 62.3 and 353.7 MB of two-party fixed-shape preprocessing in 2.54 to 4.04 seconds, at 23.37 to 83.42 MiB/s, with 603 to 1,154 MiB combined maximum resident set across the two processes. For cold SSCC validation and view materialization, the median maximum resident set ranges from 345 to 567 MiB per measured role. Persistent two-party bytes are 354,183,120 for one pair of SSCC preprocessing shares and 734,766,560 for all four fixed-shape preprocessing instances. Every deployed preprocessing instance is one-use, so there is no cross-request reuse break-even.

I Joint-Block Validation and Storage Results

The joint-block experiment uses the fixed EzPC implementation, model width 128, two attention heads, a 48-bit ring, and fixed-point encoding with 12 fractional bits. The evaluated operator chain is

MHA, residual, LayerNorm, SwiGLU FFN, residual, and LayerNorm. For each public capacity and width pair, both parties execute a selected view from the factorized preprocessing and a separately generated fixed-shape block. The two executions produce identical final arrays in every observation. Table 6 reports the number of compared words, the maximum deviation of either the selected or fixed-shape execution from the clear fixed-point reference, and bidirectional online communication per party. Selected and fixed-shape communication are equal in every configuration.

Table 6: All nine joint-block configurations. Differences between selected and fixed-shape outputs are zero in every row. The acceptance threshold is 32 integer units under the 12-fractional-bit encoding.

Capacity	FFN width	Compared words	Max error	Online bytes
32	128	4,096	4	1,807,932
32	256	4,096	8	2,087,484
32	512	4,096	16	2,646,588
64	128	8,192	5	4,213,868
64	256	8,192	8	4,772,972
64	512	8,192	17	5,891,180
128	128	16,384	15	10,819,852
128	256	16,384	9	11,938,060
128	512	16,384	17	14,174,476

On the GPU system, 20 repetitions per public capacity and width pair measure the online interval after peer synchronization through final output. Preprocessing generation and preprocessing-file loading are excluded. For each repetition, the reported two-party duration is the larger of the two party durations. Table 7 reports the resulting distributions. The selected execution runs before the fixed-shape execution in every process, and the timing campaign reuses the same deterministic MHA preprocessing files. We therefore use it to report online intervals. The experiment does not support conclusions about relative speed or one-use lifecycle enforcement.

Table 7: Joint-block online interval on the GPU system over 20 repetitions. Values are two-party median and 95th percentile times in seconds.

Capacity	FFN width	Selected	Fixed-shape
32	128	0.084/0.087	0.076/0.079
32	256	0.089/0.093	0.075/0.077
32	512	0.100/0.103	0.076/0.078
64	128	0.099/0.102	0.081/0.084
64	256	0.109/0.113	0.081/0.083
64	512	0.133/0.136	0.083/0.086
128	128	0.138/0.140	0.104/0.106
128	256	0.161/0.164	0.108/0.123
128	512	0.206/0.210	0.111/0.114

Selected and fixed-shape outputs agree for all 18 combinations of party and configuration. The accounting scope includes per-party preprocessing correlations, keys, and public remaps. It excludes executable code, transient workspace, and online masked input or model arrays. Within this common scope, the factorized preprocessing is decomposed by component in Table 8.

Table 8: Measured factorized preprocessing size per party.

Component	Bytes
Common MHA plus three capacity families and corrections	146,347,456
Common maximum FFN	92,864,720
Width-only FFN corrections	262,144
LayerNorm preprocessing for each capacity	19,910,656
Public capacity-local boundary remaps	688,128
Factorized preprocessing total	**260,073,104**

Regrouped by selector axis, the preprocessing share has 169,141,380 common bytes, 6,293,600, 19,263,780, and 65,112,200 bytes for capacities 32, 64, and 128, 262,144 width-axis bytes, and zero capacity-by-width bytes. Nine separate fixed-shape preprocessing shares total 951,958,864 bytes. Capacity-deduplicated fixed-shape preprocessing that reuses one MHA/LayerNorm share per capacity while retaining nine fixed-shape FFN shares totals 524,229,968 bytes. These denominators yield the 72.68% and 50.39% reductions in Table 3.

The artifact contains machine-readable results, build records, and the implementation sources. A second execution reconstructed the evaluated environment, reproduced all nine configurations, and regenerated identical party MHA files and results. This supports functional reproducibility within the evaluated environment. It does not convert deterministic test material into a concrete security instantiation. The MHA portion is serialized with segment checksums, while FFN and LayerNorm records are generated in memory rather than written as one atomic party file.

I.1 Twelve-block structural chain

The experiment evaluates 12 consecutive blocks, each comprising MHA, residual addition, LayerNorm, a SwiGLU FFN, residual addition, and LayerNorm, at public shape $(32, 128)$. The dealer phase writes one role-specific preprocessing file with checksums for each party and 24 distinct party-and-layer-specific MHA files. Public remaps connect each block's final masked tensor to the next block's input. The online commands receive only the public shape, peer address, party preprocessing file, and MHA directory. They receive no dealer seed, clear input, or clear model. A one-use marker records consumption and rejects a second attempted use of the same preprocessing file.

Both online parties reveal only the final 4,096-word output. Their outputs are identical to each other and to selected and separately generated fixed-shape references constructed for the same session. No intermediate activation is revealed, and online communication is 21,695,184 bytes per party, exactly 12 times the one-block value of 1,807,932 bytes. The serialized preprocessing totals 2,928,023,188 bytes per party: a 1,171,853,716-byte role-specific preprocessing file and twelve MHA files. The compact artifact retains all execution records, result arrays, and metadata while omitting the large preprocessing files.

The workload uses deterministic synthetic weights and input. The difference between SIGMA's `TrWithSlack` and floor truncation compounds through later LayerNorms, so the deterministic clear fixed-point reference remains diagnostic rather than an acceptance criterion. The experiment checks role separation, mask continuity, equality between selected and fixed-shape outputs within one session, recorded one-use consumption, and communication accounting for this synthetic chain. It does not establish accuracy for a trained 12-layer SSCC model.

J Full-Model Hybrid HE/MPC Case Study

We evaluate the public-capacity premise on the recovered public BOLT quantized BERT-base MRPC checkpoint. The first attention block runs at capacity 128. The application then reveals the valid-token count K and selects public capacity 32 if $K \leq 32$, or 64 otherwise. Stable secret-shared compaction hides retained identities, capacity-aware ciphertext packing changes the physical HE shape, and the selected capacity is preserved through the remaining eleven Transformer blocks.

We report the two evaluation systems separately because their timings are not comparable. The CPU system is an Apple M4 Pro machine with 12 CPU cores, 48 GB RAM, macOS 27.0, and arm64. The

GPU system is a Linux/x86-64 machine with an AMD EPYC 9754, 1.22 TB RAM, and an NVIDIA RTX PRO 5000 Blackwell. The fixed-capacity clear replays and the two complete private requests below ran on the CPU system. A separate aggressive-schedule clear diagnostic ran on the GPU system and is excluded from these tables and claims.

Table 9 reports the 408-example clear fixed-point replay, which assigns 22 examples to capacity 32 and 386 to capacity 64.

Table 9: Clear fixed-point MRPC replay on the CPU system (408 examples). The public policy chooses 32 when the valid-token count is at most 32 and 64 otherwise. This is not a private full-dataset run. Prediction changes are measured relative to the unpruned replay.

Policy	32/64 examples	Accuracy	F1	Prediction changes
Unpruned capacity 128	0/0	86.5196%	90.3678%	0
Public 32/64 policy	22/386	86.5196%	90.3339%	2
Always 64	0/408	86.5196%	90.3339%	2
Always 32	408/0	82.8431%	87.4101%	57

Its 353 correct predictions out of 408 match the unpruned accuracy, with two prediction flips and a 0.0339-point F1 decrease. Always using capacity 32 instead loses 3.68 accuracy points, showing why a public capacity policy is needed rather than forcing the smallest shape.

Table 10 reports one complete two-party request at each public capacity.

Table 10: One complete two-party request on the CPU system for each public capacity. Both parties run on the same host. Online bytes are bidirectional, and wall time is reported for one execution.

Capacity	Sample ID	Online bytes	Wall time (s)	Preprocessing bytes per party
32	41	23,227,795,927	1,244.15	17,733,640
64	0	48,237,095,201	1,363.44	30,737,928

For each public capacity, we select one example and run both parties on the CPU system through stable compaction, all 12 private Transformer blocks, pooling, and classification. Both party executions complete, the retained-shape logs record all 12 layers, all 12 one-use preprocessing records are consumed, and the private prediction matches the corresponding clear fixed-point replay. The online byte count includes the final 16-byte response. The 64-token sample-0 execution uses 31.69% less online communication than the paired 70,617,150,070-byte unpruned sample-0 baseline. The single wall times are reported for completeness and do not support a latency distribution or WAN claim.

This hybrid system differs from the SSCC/FSS construction and its conditional security theorem, so its byte counts are not combined with the SSCC preprocessing tables. The complete requests show that a public shape decision can change the physical continuation across a trained checkpoint without reducing task accuracy under the stated policy. It does not evaluate 12-layer SSCC preprocessing.