SCOPEPEFT: PRIVATE AUTOREGRESSIVE INFERENCE WITH SCOPED JOINT COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Secure language-model inference protects users' prompts and proprietary model parameters, but secure multi-party computation (MPC) imposes substantial overhead. Parameter-efficient adaptation of a public backbone reduces secret weights but does not bound the computation that depends on them. In autoregressive inference, constructing persistent adapted key–value (KV) histories can require joint feed-forward network (FFN) work at every earlier position. ScopePEFT fixes joint-head sets and FFN sources before training. The client computes the full native trajectory, while designated attention heads retain complete joint KV histories. Each adapted tail block takes its FFN response from the native trajectory or MPC, determining the upstream work for later KV. Numerical adaptation and separate caches support this graph under semi-honest MPC with trusted preprocessing. We evaluate GPT-2 small and medium on WebNLG. At the single-seed medium Q16 endpoint with all tail heads and FFNs evaluated jointly, BLEU is 27.85 versus 26.19 for parameter-matched three-layer LoRA, while NLL worsens from 1.291 to 1.356. On three separate fixed replay prompts, the same weights reduce mean rank-0 framework bytes by 41.32%.

1 INTRODUCTION

Private language-model inference must protect the client's prompt and the provider's proprietary parameters. Secure multi-party computation (MPC) supports this goal, but adds substantial communication and computation overhead (Knott et al., 2021; Hao et al., 2022; Lu et al., 2025). Existing work reduces this overhead through protocol optimization, numerical approximation, and changes to model architecture (Li et al., 2023; Kundu et al., 2023; Pang et al., 2024).

Parameter-efficient fine-tuning (PEFT) offers an architectural opportunity when the backbone is public and only task-specific parameters are private (Houlsby et al., 2019; Hu et al., 2022). Public weights then allow the client to execute parts of the model locally. CryptPEFT uses one-way data flow from a public backbone to private adapters for vision Transformers, leaving extension to language models open (Xia et al., 2026). MARILL already applies layer freezing, factored LoRA, and head merging to private autoregressive inference (Rathee et al., 2024).

However, reducing the number of private parameters does not necessarily reduce the computation that depends on them. After a private update changes a hidden state, subsequent public-weight operations can still require joint evaluation. Autoregressive attention extends this dependence across both layers and positions (Vaswani et al., 2017). A later block's keys and values require earlier-block states at every prefix position. Constructing adapted history can therefore entail joint attention, normalization, and feed-forward network (FFN) work across the entire prompt. Caching reuses that history after construction; it does not remove this initial work.

The question is therefore how to retain access to the complete prefix while controlling which dependencies introduced by private adaptation require joint computation. ScopePEFT separates these decisions through two trajectories over the same consumed prefix (Figure 1). The client evaluates the public backbone on every consumed token, defining the full native trajectory. A joint trajectory receives protected native responses and maintains complete, persistent adapted key–value (KV) histories for designated attention heads. Each adapted tail block draws its FFN response from the native or joint trajectory. The public configuration is fixed before training; only private adapter weights are

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.

learned. Numerical adaptation and separate caches support fixed-point execution under semi-honest MPC with trusted preprocessing.

Our contributions are:

- **A native/joint dependency graph for autoregressive inference.** ScopePEFT retains the complete native trajectory while choosing which attention histories and FFN responses depend on privately adapted states.
- **The upstream cost of persistent joint KV.** We derive how a preceding block's FFN source changes the joint rows needed to construct later-block history, even with fixed private parameter counts and caching.
- **Measured communication savings and their limits.** With 169 prompt tokens and three output decisions, GPT-2 small F01 uses 76.86% fewer post-initialization application bytes per party than matched tail LoRA. The endpoints match private parameter counts and numerical adaptation, and both use caching. Final medium Q16 comparisons include stronger generation metrics and lower replay communication than Deep LoRA for H16-F11, alongside worse NLL. Small-model timing gains diminish with generation length.

2 Problem setting and privacy assumptions

2.1 Private autoregressive inference

Alice holds a prompt $x_{1:P}$ and subsequent generated tokens; Bob holds private adaptation parameters ϕ. The backbone θ, architecture $\mathcal{C}$, numerical algorithms, head sets, and shape bounds are public. At decision t, with consumed length $n_t = P + t - 1$, the intended functionality is

$$z_t = F_{\mathrm{num}}(\theta, \phi, x_{1:n_t}; \mathcal{C}), \qquad y_t = \arg \max_{v \in \{0,\dots,V-1\}} z_{t,v}. \tag{1}$$

All V logits are released only to Alice, who uses local first-index argmax. If generation continues, she appends the selected token; Bob receives the declared continuation and length information, not the token. EOS or the public output cap ends generation without another append. This functionality requires neither secret argmax nor private information retrieval.

2.2 Privacy assumptions

For a fixed consumed prefix, Alice can compute native responses from $(\theta, x_{1:n_t})$ alone. Earlier outputs may depend on ϕ, so this independence is conditional on the prefix. Native activations remain private to Alice and enter joint computation as protected inputs. Joint hidden states and all actual MPC KV tensors retain secret shares, including tensors with Alice-owned sources.

We use CrypTen-based semi-honest MPC with a trusted preprocessing provider (TFP) (Knott et al., 2021). Confidentiality depends on the underlying protocols and this trust assumption; we provide no new cryptographic proof, dealer-free protocol, or malicious-security guarantee. Public architecture and shapes, continuation/length information, and Alice-only logits define the disclosure boundary. Output-based model extraction remains possible (Tramèr et al., 2016).

Measurements use a same-host, same-container loopback TrustedFirstParty simulation, with the provider inside a participating role. The trusted custodian also reads saved outputs for verification. Role-specific logit release in this simulation does not verify physical isolation between Alice and Bob or an independently isolated preprocessing service. Appendix E records the remaining privacy and artifact boundaries.

3 ScopePEFT: reducing joint computation

3.1 Dependency choices and execution

The public configuration fixes joint-head sets and each tail block's FFN source before training. Only adapter weights are learned. Figure 1 shows how later KV depends on earlier FFN responses; the next two subsections define the resulting function and history-construction cost.

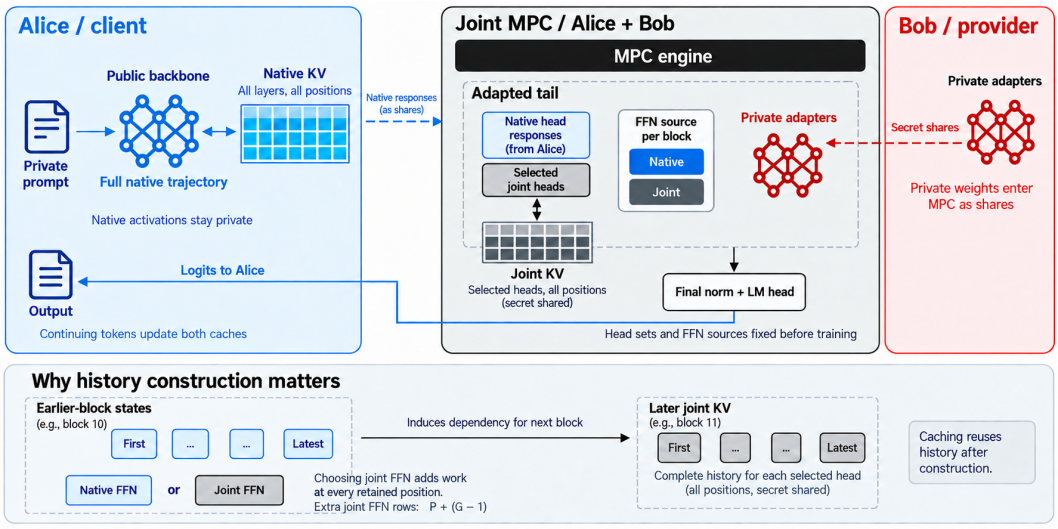


Figure 1: **ScopePEFT overview.** Alice's full native trajectory stays private. Native responses and Bob's adapters enter MPC as shares; logits return only to Alice. The central region lists jointly computed components, not operator order. Native KV covers all layers, and designated joint heads retain every position. Continuing tokens advance both caches. Bottom: switching block 10's FFN from native to joint in small F01–F11 adds $P + (G - 1)$ joint FFN rows, because block-11 KV requires earlier states at every retained position. Cells are schematic, not measurements.

3.2 Full native context and scoped joint attention

Let the backbone have L causal blocks, hidden width d, and H heads, with zero-based indices. Alice computes native block outputs b_l^{A}, attention responses, and FFN increments Δ_l^{A} on the complete consumed prefix. No joint hidden state feeds into this path. Three bias-free rank-r adapters after blocks $s = L - 3, s + 1, s + 2$ initialize and update the joint trajectory:

$$A_l(v) = U_l\,\mathrm{GELU}(D_l v), \quad D_l \in \mathbb{R}^{r\times d},\ U_l \in \mathbb{R}^{d\times r}, \quad h_{s,i} = b_{s,i}^{\mathrm{A}} + A_s(b_{s,i}^{\mathrm{A}}). \tag{2}$$

For each tail block $l \in \{s+1, s+2\}$, fixed public sets J_l and R_l partition the heads between adapted QKV with joint history and Alice's native attention responses:

$$(q_{l,j,i}, k_{l,j,i}, v_{l,j,i}) = \mathrm{QKV}_{l,j}(\mathrm{LN}_{l,1}(h_{l-1,i})), \qquad j \in J_l, \tag{3}$$

$$a_{l,j,i} = \begin{cases} \mathrm{Attn}(q_{l,j,i}, K^{\mathrm{J}}_{l,j,1:i}, V^{\mathrm{J}}_{l,j,1:i}), & j \in J_l, \\ \mathrm{NativeResponse}_{l,j,i}(x_{1:n_t}; \theta), & j \in R_l, \end{cases} \tag{4}$$

$$u_{l,i} = h_{l-1,i} + \mathrm{Proj}_l^{O}(\mathrm{Concat}_{j=0}^{H-1}\, a_{l,j,i}). \tag{5}$$

Public QKV and output projections include biases. Designated heads retain adapted keys and values at every required position, including the current one; other heads supply protected complete-context native responses. All enter the joint residual $u_{l,i}$, whose activations remain private.

3.3 FFN sources and history-construction cost

A public bit f_l determines whether the FFN response is evaluated on the block's joint or native state:

$$\Delta_{l,i} = \begin{cases} \mathrm{MLP}_l(\mathrm{LN}_{l,2}(u_{l,i})), & f_l = 1, \\ \Delta^{\mathrm{A}}_{l,i}, & f_l = 0, \end{cases} \tag{6}$$

$$w_{l,i} = u_{l,i} + \Delta_{l,i}, \qquad h_{l,i} = w_{l,i} + A_l(w_{l,i}). \tag{7}$$

When $f_l = 0$, Alice's native FFN increment is added to the joint residual $u_{l,i}$. Earlier private corrections remain in that residual, and A_l still adapts the resulting state. When $f_l = 1$, the FFN consumes the current joint state. Joint final normalization and the language-model head release all 50,257 logits to Alice.

Effect on adaptation capacity. For a fixed prefix, a native head response is computed from the public backbone, even when the joint residual has changed. A joint head instead computes its query, keys and values from adapted states. Likewise, a native FFN source adds a backbone response to the joint residual, whereas a joint FFN transforms that residual. In either case the private correction persists through the residual and the next adapter. This distinction separates complete access to the prefix from complete adaptivity to private weights; changing head sets or FFN sources changes the learned function.

Consider block 11 in the small model, comparing $(f_{10}, f_{11}) = (0, 1)$ (F01) with $(1, 1)$ (F11). Each retained joint key/value at position i needs the adapted block-10 output at that position. F01 supplies a native FFN increment there; F11 evaluates that FFN on the joint state. Retaining complete joint history therefore adds one joint FFN row per required position under F11, even though only the final query releases logits. Prefill requires all P prompt positions; G output decisions require $G - 1$ further appends, with no append after stopping. The extra joint FFN rows are

$$\Delta N_{\mathrm{FFN}} = P + (G - 1). \tag{8}$$

At $P = 169, G = 3$, F11 adds 171 such rows. Caching prevents recomputation of old rows but does not avoid their initial construction. This dependency count is separate from its byte cost: Appendix B.2 gives the implemented backend's per-row conversion and historical restricted-history ledger. Neither is a protocol lower bound.

3.4 Training and cached secure execution

Each configuration is task-trained in FP32 with a frozen backbone. Numerical adaptation then updates only its private weights for 256 Adam steps, using the declared numerical forward pass and a straight-through backward surrogate. We call this endpoint *adapted Q16*; Q16 denotes arithmetic at scale $Q = 2^{16}$. The configuration and final-checkpoint rule stay fixed. Matched adapted-Q16 comparisons give each model this adaptation opportunity and use caches; matching private parameter count does not equate expressivity or FLOPs. Training settings are in Appendices A and D.3.

Alice encodes the specified FP32 native sources for the joint integer graph. Dot products accumulate before signed rescaling toward zero. Nonlinear routines, normalization, factored LoRA products, and native AV reductions likewise follow explicit schedules (Appendix B). Changing the reduction order changes the numerical function; correctness refers to this declared graph, not exact real-valued GPT-2.

Alice caches full native KV; MPC separately caches joint tail KV. Prefill constructs history once. After each continuing decision, Alice prepares native sources for her selected token, and joint execution appends its KV. EOS or the output cap ends the session without another append. With identical sources and numerical schedules, causal invariance gives the same incremental and full-prefix function (Appendix B.1). Section 4.6 checks finite actual self-feedback trajectories.

4 Evaluation

We evaluate how much secure-inference work ScopePEFT saves, what task quality accompanies that saving, and how the benefit changes with joint scope and generation length. Experiments use GPT-2 (Radford et al., 2019). Table 1 defines the distinct quality populations and cost windows.

4.1 Experimental setup

MPC measurements use the TFP simulation described in Section 2.

All task evaluations use WebNLG challenge-2017 (Gardent et al., 2017), greedy EOS-or-128 decoding within a 384-token context, BLEU via SacreBLEU 2.5.1 (Papineni et al., 2002; Post, 2018), chrF++ (Popović, 2017), and separate teacher-forced NLL/PPL. Mean $\pm$ sample SD summarizes seeds 41/43/47 where available; final medium Q16 uses seed 41 only. The additional 256-group sample excludes the initial 128 groups and listed project evaluation inputs. Its 689 references and 19,614 target positions evaluate eighteen frozen FP32 checkpoints without further training or score-based selection (Appendix C.3).

Configurations and names. Small F01/F11 have four joint heads per tail block and rank-24 adapters (110,592 parameters). Medium H5/H16 use five/all sixteen joint heads and rank-32 adapters (196,608 parameters); F01/F11 always specify FFN sources. Tail denotes two-layer Q/V LoRA: rank 18 on small and rank 24 on medium. Medium Deep uses three-layer rank-16 LoRA. All medium configurations have 3,072 FP32 updates and matched parameter/sample budgets; within-seed ScopePEFT runs also match initial adapter tensors. Historical checkpoints are reused, rather than randomized contemporaneously. H5-F01 is the prespecified primary medium configuration. Appendix A gives the historical naming map.

Timing and communication. Both replay studies use two technical repeats, equally adapted cached models, predetermined continuation tokens, and the loopback TFP setting. Small warm requests include native preparation, sharing, computation, output/control, and on-demand TFP work, excluding reference checks and file writing. Medium instrumented sessions include startup, weight loading, sharing, computation, output/file writing and offline verification, excluding prior native-input preparation and input copying. The former reports post-initialization application bytes per party; the latter reports rank-0 framework bytes. Neither is wire traffic or complete production cost. Replay controls work, not natural-generation quality.

4.2 MEDIUM: FINAL Q16 QUALITY AND MEASURED COST

We evaluate four final seed-41 Q16 checkpoints on the same 256 semantic groups (Table 2). Quality evaluation and MPC replay use identical adapted weights on different inputs: local numerical generation supplies task metrics, while three fixed prompts with $P = 31, 54, 84$ and $G = 16$ supply the MPC measurements in Figure 2. Each replay has two technical repeats; overall means weight the prompts equally. Appendix C.4 gives complete metrics and audit coverage, and Appendix D.3 locates every replay pair in S7.

H5-F01 versus Tail. H5-F01 reduces mean rank-0 framework bytes by 66.97% and instrumented-session time by 13.51%; both directions hold in all six replay pairs. Its BLEU exceeds Tail by 0.273, but chrF++ is lower by 0.834 and NLL is worse by 0.067.

H16-F11 versus Tail and Deep. H16-F11 applies its first private adapter to block 21's output; Deep LoRA updates Q/V within that block, so fully joint final blocks do not imply identical dependency graphs. Relative to Tail, H16-F11 raises BLEU/chrF++ by 4.311/4.499 and slightly lowers NLL, with 3.77% more framework bytes and 18.80% more session time. Relative to Deep, it raises BLEU/chrF++ by 1.660/1.363 and uses 41.32% fewer framework bytes and 20.40% less session time, but worsens NLL from 1.291 to 1.356. These four single-seed endpoints compare complete configurations; they do not isolate head and FFN effects or establish a same-request quality–latency frontier.

Table 1: Quality and fixed-workload evidence. Seeds count training runs; actual MPC uses seed 41. Quality means task metrics, not numerical agreement. Original Q16 precedes adaptation. The two timing windows below are not pooled.

Model / stage	Inputs	Seeds	Quality	Cost evidence
Small FP32, adapted Q16	128 test groups	3	Yes	No cost on this population
Small, both Q16 stages	871 explored dev inputs	1	Yes	$P = 169, G = 3$: post-init bytes/party; no latency
Small adapted Q16	6 replay prompts; 3 extended to $G = 64$	1	No	Warm-request seconds; post-init bytes/party
Medium FP32	Initial 128 + additional 256 groups	3	Yes	No MPC at this stage
Medium adapted Q16	Same additional 256 groups	1	Yes	No MPC on this population
Medium adapted Q16	3 replay prompts, $G = 16$	1	No	Instrumented-session seconds; rank-0 framework bytes

Table 2: **Final medium Q16 quality.** The same 256 groups, seed 41, 196,608 private parameters, and 256 numerical-adaptation updates per endpoint. BLEU/chrF++ measure generated text; NLL is token-weighted reference loss. These exact weights also supply Figure 2, on separate fixed-replay inputs. One seed does not estimate training variability.

Model	BLEU ↑	chrF++ ↑	NLL ↓
H5-F01	23.810	42.439	1.441068
H16-F11	27.848	47.772	1.355597
Tail LoRA	23.537	43.273	1.374092
Deep LoRA	26.188	46.409	1.291263

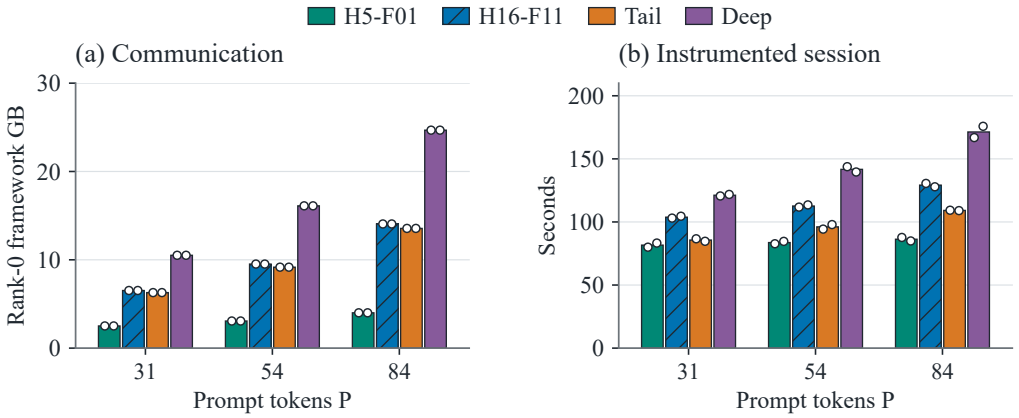


Figure 2: **Medium MPC cost at the same Q16 weights as Table 2.** Seed 41, fixed $G = 16$ replay. Bars show prompt means; open points show both technical repeats, offset for visibility (byte counts coincide). (a) GB is 10^9 rank-0 framework bytes. (b) Instrumented sessions include startup and offline verification, but exclude prior native-input preparation and copying. They are not full-service inference latencies. Appendix D.3 links every pair in S7.

4.3 Small: communication and quality across stages

Matched application communication. At $P = 169, G = 3$, small F01 saves 76.86% of post-initialization application bytes per party: 4.467 GB versus Tail's 19.299 GB. F11 uses 13.565 GB, matching the FFN history-construction ledger in Appendix B.2. All endpoints are numerically adapted and cached, with matched private parameter counts. No inference latency was measured for this workload.

Numerical quality and stronger FP32 controls. On 128 groups, FP32 F01 obtains BLEU/chrF++ of 22.534/43.492 versus Tail's 29.946/48.251. Adapted-Q16 means are 25.005/42.730 versus 27.179/46.099. F01's BLEU increases after adaptation while chrF++ decreases and NLL rises (Figure 5). Missing unadapted-Q16 and matched FP32-continuation controls prevent separating arithmetic from additional updates (Appendix C.2).

On explored development data, adapted F01 gives BLEU/chrF++ of 40.57/52.35 versus Tail's 40.86/55.39. Deeper and all-layer FP32 LoRA and serial S have higher development BLEU/chrF++ than F01; matched Q16/MPC costs are unavailable (Appendix C.1).

4.4 Joint-head sets and FFN sources in FP32

The medium factorial comparison uses three training seeds and fixed H5/H16 head sets at each F01/F11 setting. Figure 3 shows within-seed BLEU differences on the initial 128 and additional 256 groups; both populations reuse the same checkpoints. All configurations retain 196,608 private parameters and 3,072 FP32 updates. Increasing H5 to H16 also changes which heads are selected, so the comparison does not isolate a head-count law.

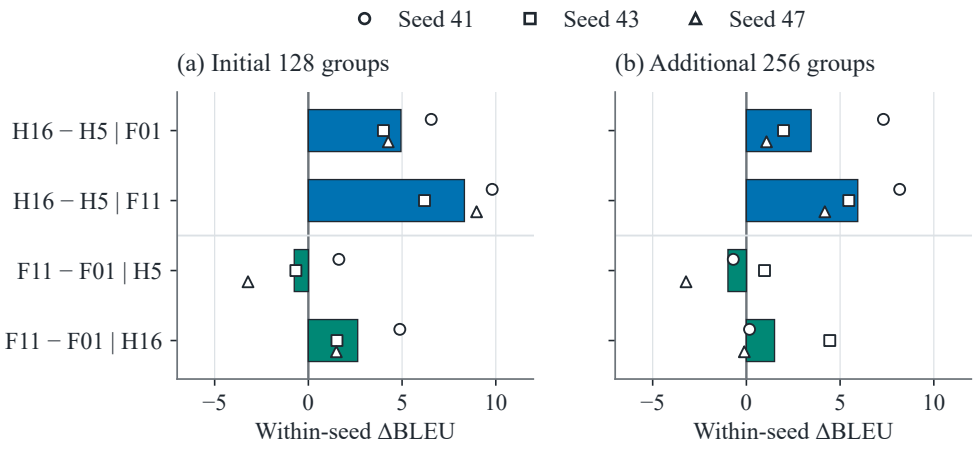


Figure 3: **Head-set and FFN-source effects differ.** Medium task-trained FP32, fixed head sets, before numerical adaptation. Bars show mean within-seed BLEU differences; markers show seeds 41 (circle), 43 (square), and 47 (triangle). The initial 128 and additional 256 groups reuse the same models, not independent training replications. H16 versus H5 is positive at both FFN settings; F11 versus F01 depends on the head set. Only BLEU is plotted; all metric summaries and the S3 seed records are in Appendix C.3.

Enlarging the fixed joint-head set. H16 improves BLEU, chrF++, and NLL over H5 in every paired seed at both FFN settings on both samples. Mean BLEU increases by 4.940/8.326 at F01/F11 on the initial sample and by 3.455/5.942 on the additional sample. The latter's H16-F11 exceeds Deep's BLEU/chrF++ in all three seeds, but improves NLL in only two and exceeds Tail's BLEU in two. H5-F01 is worse than Tail on every metric in every seed.

Changing the FFN source. F11 changes mean BLEU by -0.753 at H5 and $+2.633$ at H16 on the initial sample; on the additional sample, the changes are -0.984 and $+1.502$. At H16, each metric improves in all three seeds on the initial sample but only two on the additional sample. The descriptive BLEU interactions are 3.386 and 2.487, respectively. These conditional effects do not establish optimal head sets or a universal benefit from adding joint FFN computation. The final Q16 comparison lacks H5-F11 and H16-F01 and cannot separate the two factors.

4.5 Generation length and phase costs

For small F01 relative to parameter-matched Tail LoRA, six $G = 16$ replay prompts save 54.18–67.83% of post-initialization application bytes per party and reach first logits 35.48–59.72% earlier, but increments take 4.28–11.60% longer. Complete requests are shorter in all prompt means and eleven of twelve repeat pairs. At $G = 64$, byte savings persist (46.00–55.27%), while request means become 3.32% and 1.22% slower for two prompts; the third is 1.05% faster on average with opposite repeat-level directions (Figure 4). The curves reveal why a lower communication total need not produce a shorter request. Appendix D.1 preserves the complete pairs.

4.6 Numerical validation

All sixteen outputs in each of 24 medium fixed-replay sessions agree coordinatewise with saved integer references. Actual self-feedback is checked separately on small: six cap-32 sessions yield 123 agreeing full-vocabulary outputs (five EOS, one cap), and earlier $P = 169$ cap-128 sessions yield 311 across six original/adapted endpoints. Recorded native sources condition these checks, and actual MPC KV stays unopened. Finite agreement does not prove correctness on arbitrary prefixes or measure task quality; unequal natural output lengths do not support matched-workload speed ratios.

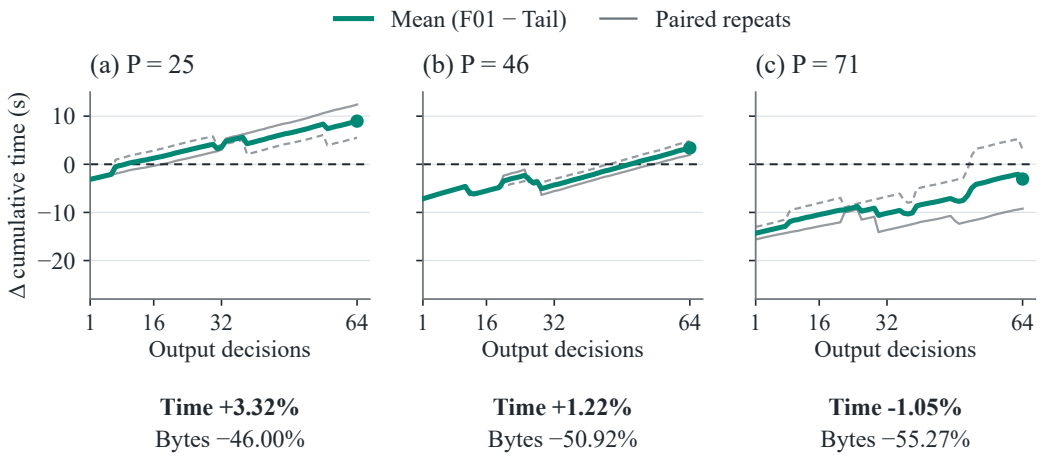


Figure 4: **Communication savings persist after timing gains diminish.** Small adapted Q16, fixed replay through $G = 64$. Curves subtract Tail's cumulative warm time from F01's at the same prompt, repeat and output count; negative is faster. Thin gray lines show both technical-repeat pairs; the colored line shows their mean. Only the last point is a complete request; earlier points are cumulative prefixes of the same sessions. Annotations give terminal mean-time changes and post-initialization application-byte savings per party, not independent shorter-request experiments.

5 Related work

Public backbones and private adaptation. Adapters and LoRA provide lightweight private parameterizations (Houlsby et al., 2019; Hu et al., 2022). CryptPEFT uses one-way public-to-private adaptation and cost-aware architecture search for vision classification (Xia et al., 2026); MARILL combines layer freezing, factored LoRA, and head merging for autoregressive inference (Rathee et al., 2024). Encrypted LoRA also appears in HE-based EF-LLM with Gaussian-kernel attention (Rho et al., 2025). ScopePEFT instead specifies which responses come from a complete native trajectory and which follow adapted states, including the FFN work inherited by persistent joint KV. Our Tail control matches private parameter count, not the adapter function; medium comparisons also match training-update and sample budgets. It does not reproduce these prior systems.

Selecting protected computation. Architecture planning and selective linearization allocate expensive nonlinear work (Mishra et al., 2020; Cho et al., 2022; Kundu et al., 2023). Transformer approximations and distillation reduce its cost (Chen et al., 2022; Li et al., 2023; Dong et al., 2025; Luo et al., 2024); heterogeneous attention, frequency-domain representations, polynomial networks, and outlier structure offer further model-specific reductions (Zeng et al., 2023; Roy & Roy, 2025; Xie et al., 2026; Zhao & Fang, 2026). CipherPrune removes encrypted tokens and adjusts polynomial degrees (Zhang et al., 2025); MPCache selects KV entries (Zeng et al., 2025), while Comet exploits activation sparsity (Yan et al., 2025). ScopePEFT keeps all native positions and every joint KV position for its designated heads. Its choice concerns which responses depend on private corrections, rather than which token positions remain accessible.

Protocols and execution environments. Private prediction spans encrypted-network evaluation (Gilad-Bachrach et al., 2016), oblivious transformations and hybrid cryptography (Liu et al., 2017; Juvekar et al., 2018), optimized secure operators and fixed-point routines (Rathee et al., 2020; Huang et al., 2022; Rathee et al., 2021), and Transformer-specific protocols (Hao et al., 2022; Pang et al., 2024; Lu et al., 2025). GPU/FSS execution, cross-layer fusion, and HE packing target additional execution costs (Gupta et al., 2024; Xu et al., 2025; Zhang et al., 2026); Slalom partitions work between trusted hardware and an untrusted accelerator (Tramèr & Boneh, 2019). ScopePEFT uses CrypTen (Knott et al., 2021) to measure a dependency graph, not a new secure operator. These functional comparisons do not establish cross-system cost or security superiority: ownership, trust and workloads differ, and we did not test whether prior frameworks can express the same graph.

6 DISCUSSION AND LIMITATIONS

Complete context and adaptation capacity. Retaining every token does not give every branch access to private hidden-state corrections. Native responses depend on the consumed prefix, whereas joint responses also depend on adapted states. Enlarging the tested joint-head sets consistently improves FP32 quality; the FFN effects vary with the head set and evaluation sample. This is consistent with different demands on adaptation capacity, but does not identify optimal heads or a unique cause of the quality gap. Final Q16 ranks also differ across generation metrics and reference loss.

First-output gains over longer generation. ScopePEFT avoids some joint work when constructing history, but each additional decision still requires native preparation and joint attention/FFN execution. The small-model measurements pair earlier first logits with slower increments; as generation continues, these increments erode the initial time saving. This measured phase pattern explains why byte savings can persist after the request-time advantage disappears. It does not isolate the responsible secure operators or predict a universal crossover.

Scope of the evidence. The study covers one WebNLG task and two GPT-2 sizes. The development set was explored; retained-history checks do not exclude external test use. Three FP32 seeds quantify variation for the fixed configurations, while final medium Q16 quality and actual MPC each use one training seed. Overlap metrics leave factual accuracy untested, and missing stage controls prevent attributing quality changes solely to arithmetic or adaptation.

Full-service cost, WAN behavior, tail latency, physical wire traffic, true protocol rounds, secure-operator counts, and separate preprocessing cost are unmeasured. Backend communication calls are auxiliary interface counters. The TFP simulation and output/length disclosures are specified in Section 2; finite numerical agreement leaves general range and overflow guarantees open.

7 CONCLUSION

ScopePEFT separates full native context from the responses that follow privately adapted states. Persistent joint KV couples those choices across layers: an earlier FFN source can change joint work at every history position despite identical private parameter counts. Final Q16 comparisons show communication savings with metric-specific quality outcomes, and longer generation can erase the initial timing advantage. Private-inference cost therefore depends on the history of private dependencies as well as the size of the private model component.

REPRODUCIBILITY STATEMENT

Sections 2–4.1 define the graph, functionality, populations and measurement windows. The appendices and supplementary CSV files retain training settings, numerical schedules, all quality controls, seed-level scores, and paired replay measurements. Core final-Q16 scores and checkpoint bindings were independently verified; the coverage and incomplete portions of the numerical-array, archive and execution audits are specified in Appendix C.4. The paper-source supplement contains aggregate tables, CSV records and figure assets. A separate minimal code package provides the four final medium Q16 endpoints, adapter checkpoints and reproduction commands, starting from retained native checkpoints. Backbone weights and data are obtained separately. The reorganized GPU and MPC entry points have not been rerun end to end.

ETHICS STATEMENT

Input privacy does not make generated text correct, unbiased, or harmless. The functionality releases logits to Alice and continuation/length information to Bob, and does not prevent output-based model extraction. The semi-honest and trusted-preprocessing assumptions delimit the privacy claim. No new human-subject study is reported.

AI USE STATEMENT

Generative AI assistants supported research framing, literature discovery and verification, implementation and test development, experiment orchestration and collection, result interpretation, and manuscript drafting, editing, and Figure 1 generation. AI-generated text and artwork are not experimental evidence. Responsibility for scientific claims, citations, and artifacts rests with the human authors.

REFERENCES

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. THE-X: Privacy-Preserving Transformer Inference with Homomorphic Encryption. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3510–3520. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.277.

Minsu Cho, Ameya Joshi, Brandon Reagen, Siddharth Garg, and Chinmay Hegde. Selective Network Linearization for Efficient Private Inference. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 3947–3961. PMLR, 2022. URL https://proceedings.mlr.press/v162/cho22a.html.

Ye Dong, Wen-Jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, Wen-Guang Chen, and Jianying Zhou. PUMA: Secure inference of LLaMA-7B in five minutes. *Security and Safety*, 4:2025014, 2025. doi: 10.1051/sands/2025014.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 124–133, Santiago de Compostela, Spain, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3518. URL https://aclanthology.org/W17-3518/.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 201–210. PMLR, 2016. URL https://proceedings.mlr.press/v48/gilad-bachrach16.html.

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.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private Inference on Transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15718–15731. Curran Associates, Inc., 2022. doi: 10.52202/068431-1143.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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

Zhicong Huang, Wen jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and Fast Secure Two-Party Deep Neural Network Inference. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 809–826. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1651–1669. USENIX Association, 2018. URL https://www.usenix.org/conference/usenixsecurity18/presentation/juvekar.

Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CrypTen: Secure Multi-Party Computation Meets Machine Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4961–4973. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2754518221cfbc8d25c13a06a4cb8421-Paper.pdf.

Souvik Kundu, Shunlin Lu, Yuke Zhang, Jacqueline Liu, and Peter A. Beerel. Learning to Linearize Deep Neural Networks for Secure and Efficient Private Inference. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BGF9IeDfmlH.

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

Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious Neural Network Predictions via MiniONN Transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 619–631. ACM, 2017. doi: 10.1145/3133956.3134056.

Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and WenGuang Chen. BumbleBee: Secure Two-party Inference Framework for Large Transformers. In *Proceedings 2025 Network and Distributed System Security Symposium*. Internet Society, 2025. doi: 10.14722/ndss.2025.230057.

Jinglong Luo, Yehong Zhang, Zhuo Zhang, Jiaqi Zhang, Xin Mu, Hui Wang, Yue Yu, and Zenglin Xu. SecFormer: Fast and Accurate Privacy-Preserving Inference for Transformer Models via SMPC. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13333–13348. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.790.

Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: A Cryptographic Inference Service for Neural Networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 2505–2522. USENIX Association, 2020. URL https://www.usenix.org/conference/usenixsecurity20/presentation/mishra.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. BOLT: Privacy-Preserving, Accurate and Efficient Inference for Transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2024. doi: 10.1109/SP54263.2024.00130. URL https://doi.org/10.1109/SP54263.2024.00130.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135.

Maja Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pp. 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4770. URL https://aclanthology.org/W17-4770/.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, 2019. URL https://github.com/openai/gpt-2.

Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. CrypTFlow2: Practical 2-Party Secure Inference. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 325–342. ACM, 2020. doi: 10.1145/3372297.3417274.

Deevashwer Rathee, Mayank Rathee, Rahul Kranti Kiran Goli, Divya Gupta, Rahul Sharma, Nishanth Chandran, and Aseem Rastogi. SIRNN: A Math Library for Secure RNN Inference. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1003–1020. IEEE, 2021. doi: 10.1109/sp40001.2021.00086.

Deevashwer Rathee, Dacheng Li, Ion Stoica, Hao Zhang, and Raluca Popa. MPC-Minimized Secure LLM Inference, 2024. URL https://arxiv.org/abs/2408.03561. arXiv preprint.

Donghwan Rho, Taeseong Kim, Minje Park, Jung Woo Kim, Hyunsik Chae, Ernest K. Ryu, and Jung Hee Cheon. Encryption-Friendly LLM Architecture. In *International Conference on Learning Representations*, pp. 41638–41664, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6715b4e97be055687c1ecaf33913d358-Paper-Conference.pdf.

Arjun Roy and Kaushik Roy. DCT-CryptoNets: Scaling Private Inference in the Frequency Domain. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/26407588dfb08e48c459c074ab6adb7d-Paper-Conference.pdf.

Florian Tramèr and Dan Boneh. Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJVorjCcKQ.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*, pp. 601–618. USENIX Association, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Saisai Xia, Wenhao Wang, Zihao Wang, Yuhui Zhang, Yier Jin, Dan Meng, and Rui Hou. CryptPEFT: Efficient and private neural network inference via parameter-efficient fine-tuning. In *Proceedings 2026 Network and Distributed System Security Symposium*, NDSS 2026. Internet Society, 2026. doi: 10.14722/ndss.2026.231102. URL http://dx.doi.org/10.14722/ndss.2026.231102.

Xi Xie, Ran Ran, Jiahui Zhao, Bin Lei, Zhijie Jerry Shi, Wujie Wen, and Caiwen Ding. ULD-Net: Enabling Ultra-Low-Degree Fully Polynomial Networks for Homomorphically Encrypted Inference. In *International Conference on Learning Representations*, pp. 9406–9419, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0feb70f08090d165e6bbfa9c85d05e87-Paper-Conference.pdf.

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

Guang Yan, Yuhui Zhang, Zimu Guo, Lutan Zhao, Xiaojun Chen, Chen Wang, Wenhao Wang, Dan Meng, and Rui Hou. Comet: Accelerating Private Inference for Large Language Model by Predicting Activation Sparsity. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2827–2845. IEEE, 2025. doi: 10.1109/SP61157.2025.00182. URL https://doi.org/10.1109/SP61157.2025.00182.

Wenxuan Zeng, Meng Li, Wenjie Xiong, Tong Tong, Wen-jie Lu, Jin Tan, Runsheng Wang, and Ru Huang. MPCViT: Searching for Accurate and Efficient MPC-Friendly Vision Transformer with Heterogeneous Attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5029–5040. IEEE, 2023. doi: 10.1109/iccv51070.2023.00466.

Wenxuan Zeng, Ye Dong, Jinjin Zhou, Jin Tan, Lei Wang, Tao Wei, Runsheng Wang, and Meng Li. MPCache: MPC-Friendly KV Cache Eviction for Efficient Private LLM Inference. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 97254–97284. Curran Associates, Inc., 2025. doi: 10.52202/085713-3251. URL https://doi.org/10.52202/085713-3251.

Linru Zhang, Xiangning Wang, Jun Jie Sim, Zhicong Huang, Jiahao Zhong, Huaxiong Wang, Pu Duan, and Kwok Yan Lam. MOAI: Module-Optimizing Architecture for Non-Interactive Secure Transformer Inference. In *International Conference on Learning Representations*, pp. 27927–27953, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2f2b1d6bbd50865eca40e2774a057eef-Paper-Conference.pdf.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. CipherPrune: Efficient and Scalable Private Transformer Inference. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/83a14a36de4502bac5b580db36e81858-Paper-Conference.pdf.

Lifan Zhao and Zhixuan Fang. Secure Outlier-Aware Large Language Model Inference. In *International Conference on Learning Representations*, pp. 101033–101045, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a4b293979b8b521e9222d30c40246911-Paper-Conference.pdf.

A EXPERIMENTAL SETTINGS AND PROVENANCE

Supplementary numerical records. Files S1–S8 in the source package's `supplementary/` directory retain detailed records cited below. Its README maps filenames to PDF sections and frozen sources. Each CSV identifies the model, stage, seed or technical-repeat unit, metrics, units and source precision.

Endpoint naming. The main text uses small F01/F11, medium H5-F01/H5-F11/H16-F01/H16-F11, and Tail/Deep consistently. In retained historical tables, FULL_F01 and FULL_F11 denote the small complete joint-history configurations; L18 or LoRA18 last2 is small Tail. Medium Scope_F01 is H5-F01. Q16+256 means adapted Q16 after 256 numerical-adaptation updates, whereas original Q16 precedes those updates. W0/W1 are restricted-history controls, not aliases of complete joint history. These names retain source identity without changing the evaluated function.

Small-model architecture and configurations. GPT-2 small uses $L = 12, d = 768, H = 12, r = 24$, giving $6dr = 110{,}592$ private parameters. FP32 bottleneck adapters use exact GELU; Section 3.4 specifies numerical deployment.

For small, the primary F01 configuration uses $J_{10} = \{0, 1, 2, 3\}$, $J_{11} = \{5, 7, 9, 11\}$, and $(f_{10}, f_{11}) = (0, 1)$, with joint attention width 256 per tail block. F11 changes only f_{10} to one. Medium uses five or sixteen joint heads per tail block (H5/H16) with the same F01/F11 convention and fixed adapter budget (Appendix C.3).

Backbone and task. We use public GPT-2 with 12 blocks, width 768, 12 heads of width 64, and 50,257 vocabulary items. The checkpoint revision is `607a30d783dfa663caf39e06633721c8d4cfcd7e`; the model family is available at `https://github.com/openai/gpt-2`. The prepared WebNLG population contains 6,877 training semantic groups and 17,958 training observation pairs. The earlier evaluation uses all 871 configured development entries, including the 45 six- or seven-relation inputs, without selecting easier outputs. There are 870 distinct semantic input groups and 2,258 retained references. This is the configured study population, not a claim that all WebNLG releases have this size.

Targets include EOS and exclude prompt and padding positions. Greedy generation and teacher-forced reference scoring use separate branches. Generation releases the complete vocabulary, uses first-index argmax, stops at EOS or 128 decisions, and obeys a 384-token consumed-context bound. It does not use beam search or target-informed continuation. Cross-paper WebNLG scores with different models, decoding, or reference handling are not directly compared.

Full-precision task training. The matched four-condition cohort contains FULL_F01, FULL_F11, LoRA18 last2, and LoRA12 last3, each at seeds 41, 43, and 47 and 3,072 updates. Same-seed models consume the same target-token budgets: 2,811,506; 2,810,027; and 2,811,119, respectively. AdamW uses learning rate 8×10^{-4}, weight decay 0.01, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, clipping at one, effective batch 32, and microbatch eight. Training is FP32 with no AMP, TF32, or active dropout. Only private parameters are updated. The final checkpoint is used without selection by development score; the development set was nevertheless used in broader exploration. All four conditions have 110,592 private parameters; equal counts do not equate compute or representational capacity. Native model work used the existing RTX 3090 environment. Complete project-level compute and energy totals were not measured.

Numerical adaptation. The earlier three-structure study starts from each seed-41 native final checkpoint. The new F01/LoRA test study retains those endpoints and additionally adapts native seeds 43 and 47 with the same frozen sampling plan (1,024 additional updates in total). The numerical-adaptation random seed is fixed at 41. The same 256 Adam updates use learning rate 10^{-4}, effective batch four, clipping at one, and a frozen backbone and numerical graph. The selected 1,024 distinct training groups supply 67,376 consumed prefix tokens, with prefix lengths 11–236 and 27 EOS targets. The frozen group/reference/target-position sampling excludes the earlier 32 pilot groups; pilot parameters and optimizer states are not reused. This target-position objective is not uniform weighting of every training token. Stored intermediate checkpoints are for audit, not final

selection. The straight-through backward computation is a surrogate; only the specified forward numerical function is evaluated under MPC.

Metrics and intervals. BLEU uses SacreBLEU 2.5.1, tokenizer 13a, mixed case, exponential smoothing, and its original reference handling. chrF++ uses character order six, word order two, $\beta = 2$, and no whitespace inclusion. Target NLL is token weighted; PPL is the exponential of the endpoint mean NLL. The paired bootstrap resamples exact semantic input groups, not references or generated tokens. The same 1,000 draws at seed 2026091202 serve all fifteen six-endpoint comparisons. Intervals are the 2.5th and 97.5th percentiles and are not adjusted for multiple comparisons. Native and new numerical mean/SD summaries describe three trained endpoints, not ensemble predictions. Numerical PPL summaries average endpoint PPL values; they are not the exponential of the mean of endpoint NLL values. The reported intervals use the saved draws.

Quality coverage and verification. The six earlier seed-41 original/adapted endpoints cover the complete configured development population: 5,226 input–endpoint pairs, 401,622 target positions, and 157,743 generated decisions including EOS. All have zero empty generations and no PPL overflow. Aggregate verification checks target identity and preserves the original floating-point summation order for saved NLL totals. It does not independently recompute every teacher-forced full-vocabulary cross-entropy. Coverage is complete within this development population; it does not remove its reuse or single-seed uncertainty. The new test population is documented separately in Appendix C.2.

Execution environment. The MPC environment has Python 3.10.12, NumPy 1.26.4 and Torch 2.3.1+cu118. Protocol arithmetic is on CPU, interop is one, and CUDA remains uninitialized. The earlier $P = 169$ sessions temporarily use two CPU threads for native preparation; the new matched and natural sessions use one CPU thread per role throughout. Both replay studies use the same one-thread per-role protocol. The test evaluation uses Linux CPU AVX512 native FP32 sources prepared one token at a time. Earlier independent local collection uses Python 3.10.21, NumPy 1.26.4 and Torch 2.3.1+cpu. Current sessions bind their Linux native sources. Earlier cross-platform checks found 517,412 differing coordinates among 3,877,792, at most 32 encoded Q16 units; arbitrary Windows/Linux native-source bit identity is not assumed.

B NUMERICAL GRAPH, CACHING, AND HISTORY COST

Complete-dot rescaling. Alice encodes the specified FP32 native sources for a joint integer graph with scale $Q = 2^{16}$. Dot products accumulate before signed rescaling toward zero:

$$\mathrm{Dot}_Q(\bar{a}, \bar{b}) = \mathrm{TZ}\left(\frac{\sum_k \bar{a}_k \bar{b}_k}{Q}\right), \qquad \mathrm{TZ}(z) = \mathrm{sgn}(z)\lfloor |z| \rfloor. \tag{9}$$

This order is part of the function: truncating scalar products before accumulation defines a different graph.

Fixed numerical conventions. The scale is $Q = 65,536$ and signed rescaling is toward zero after the complete dot. To specify the nonlinear schedule, define the decoded-grid operations

$$E(c) = \frac{\mathrm{TZ}(Qc)}{Q}, \quad a \otimes b = \frac{\mathrm{TZ}(Qab)}{Q}, \quad D_k(a) = \frac{\mathrm{TZ}(Qa/k)}{Q}, \quad M_d(a) = D_d\left(\sum_{i=1}^{d} a_i\right). \tag{10}$$

All scalar constants below are encoded with E before use; additions and subtractions remain on the grid. Multiplication parentheses specify the actual rescaling order. The routines are approximations, not exact real softmax, GELU or layer normalization. Public FP32 tensor inputs follow their recorded tensor-encoding path rather than a substituted NumPy cast.

The original native GPT-2 MLP retains its `gelu_new` activation; the bottleneck adapter's native function uses exact GELU. The deployment graph uses its separately declared approximations. Public-linear and private-linear operations, comparisons, explicit divisions, normalization, final projection, and output/control operations retain their specified order. Factored LoRA multiplication is not merged into a dense Q16 update. Native-head AV retains its older-position and latest-position

contributions with separate rescalings. Removing these distinctions would define a different integer function, even when the corresponding real algebra is equivalent.

Masked ODE attention. The complete QK dot is rescaled once and explicitly divided by eight for head width 64. For a scaled score row s and public causal mask m, let

$$x_j = D_{16}(\mathrm{clip}(s_j, -4, 12) \otimes m_j)\,, \qquad g_j^{(0)} = E\left(\mathrm{FP32}\left(m_j / \sum_k m_k\right)\right), \tag{11}$$

$$r^{(t)} = \sum_j \left(g_j^{(t)} \otimes x_j\right), \qquad g_j^{(t+1)} = g_j^{(t)} + \left(x_j - r^{(t)}\right) \otimes g_j^{(t)}. \tag{12}$$

The loop performs exactly 16 updates. Its weighted reduction truncates each component product *before* summing, unlike the complete-dot rule for QK/AV. Masked entries stay zero; there is no added row-sum normalization after the loop. The initial uniform values are rounded to FP32 before Q16 encoding, as in the executed implementation.

Scaled layer normalization. Define $\mathrm{Exp8}(v)$ by $e_0 = 1 + D_{256}(v)$ and $e_{k+1} = e_k \otimes e_k$ for eight squarings. For each row, choose $(s, N) = (16, 7)$ in the body and $(8, 5)$ at the final head. Then

$$u = x \otimes E(1/s), \qquad c = u - M_d(u), \qquad v = M_d(c \otimes c) + E(10^{-5}/s^2), \tag{13}$$

$$y_0 = E(2.2) \otimes \mathrm{Exp8}(-D_2(v) - E(0.2)) + E(0.2) - D_{1024}(v), \tag{14}$$

$$y_{k+1} = D_2(y_k \otimes (3 - v \otimes (y_k \otimes y_k)))\,, \qquad k = 0, \ldots, N-1. \tag{15}$$

The result is $(c \otimes y_N) \otimes \gamma + \beta$ with public affine vectors. The scaled epsilon encodes to zero in both profiles; we preserve this fact rather than silently adding stabilization. The initialization and finite Newton steps do not establish a universal accuracy or range guarantee.

Guarded activation. The output equals x for $x > 4$ and zero for $x < -4$. For the interior, set

$$p = ((E(0.044715) \otimes x) \otimes x) \otimes x, \qquad t = E(\sqrt{2/\pi}) \otimes (x + p), \tag{16}$$

$$d = 1 + \mathrm{Exp8}(-|2t|), \qquad r_0 = E(0.75), \qquad r_{k+1} = 2r_k - (r_k \otimes r_k) \otimes d, \quad k = 0, 1, 2. \tag{17}$$

Let $\sigma = r_3$ if $t \geq 0$ and $\sigma = 1 - r_3$ otherwise. The interior output is $(E(0.5) \otimes x) \otimes (1 + (2\sigma - 1))$. The executed graph uses protected comparisons and selection: values outside the guard are replaced by zero before evaluating the interior, then the positive exterior contribution is restored. No secret-dependent condition is opened or treated as a free public branch.

These formulas specify an operation schedule; $Q = 2^{16}$ denotes the fixed-point scale, not the integer word width. In the medium records, owners share already encoded integers at input precision zero, then attach 16 fractional bits; public-weight codes are checked for encoding round trips, and pre-ring bound guards are required. The manuscript aggregates do not supply a complete portable specification of the ring, every intermediate bound, or the native tensor conversion path. Those implementation details remain necessary for bit-exact reproduction. No overflow guarantee for arbitrary inputs follows from the finite checks.

B.1 Incremental algorithm

At decision t, let C_t^{A} and C_t^{J} describe the native and joint histories for the same consumed prefix $x_{1:P+t-1}$. The generation procedure is:

1. Initialize the protocol and record its accounting boundary. Alice constructs native sources for the prompt. Joint prefill computes the block-s adapter states, the required block-$(s+1)$ prefix states, and both tail layers' keys and values, where $s = L - 3$.
2. Evaluate the current required query and release all 50,257 logits to Alice. She applies first-index argmax locally.
3. Send the declared continuation control, including on the last decision. If EOS or the cap ends the session, return without appending another token.

4. Otherwise, Alice appends the actual selected token and extends C_t^{A}. Supply the required protected native values; compute and append the corresponding joint K/V shares without changing old entries. Continue with the longer prefix.

LoRA receives its own cache-aware implementation under the same release/continuation contract. A fixed-reference replay would replace step 4's chosen token with a predetermined token and is a different experiment.

Conditional same-function argument. Assume deterministic fixed-point primitives and the same operation order, causal masking, position indices, fixed parameters, and identical recorded native sources. After prefill, suppose each cached joint K/V equals the corresponding full-prefix result. A new token cannot change an earlier causal hidden state, so previous K/V remain equal. The new position reads the same earlier K/V and native sources in both computations; applying identical ordered primitives gives the same new hidden state and appended K/V, then the same current logits. Induction over decisions establishes the cache relation under these assumptions. The argument does not permit reordering reductions, recapturing numerically different FP32 sources, or inserting arbitrary stale cache states.

Actual collection checks the resulting outputs and the NumPy reference caches on finite trajectories. It does not inspect all internal secret MPC state, independently rewrite the native Transformer, or prove a universal numerical range bound. Appendix D.2 records the checked trajectory coverage.

B.2 Cost of retained joint history

The complete-history row count in Equation 8 converts to bytes under the implemented conditional ledger B_{cond}. With the remaining operations and lengths fixed, the marginal joint FFN row costs $F = 53{,}204{,}416$ bytes per party. The retained identities are

$$B_{\mathrm{cond}}(\textsc{Full_F11}) - B_{\mathrm{cond}}(\textsc{Full_F01}) = F(P + G - 1), \tag{18}$$

$$B_{\mathrm{cond}}(\mathrm{W1_F11}) - B_{\mathrm{cond}}(\mathrm{W1_F01}) = F(G + 1). \tag{19}$$

Full constructs $P - 1$ historical rows before the first output and one row at each output position $P, \ldots, P + G - 1$. Historical W1 instead retains joint support at the current and immediately preceding positions, using native sources for older KV: one prefill row at $P - 1$ plus G current rows gives $G + 1$. Its distinct native/joint source schedule also changes reused-head values; W1 is not a current Full endpoint with only a cache switch. At $P = 169, G = 3$ the ledgers charge 171 and four rows. Their marginal difference $F(P - 2)$ is prompt-dependent, not a protocol lower bound or a quality interaction.

At matched $P = 169, G = 3$, the measured F11-minus-F01 difference is $9{,}097{,}955{,}136 = 53{,}204{,}416 \cdot 171$ bytes per party. The same fixed-geometry totals occur for original and adapted checkpoints because shapes and operation schedules match.

Each full-output operation records $402{,}056 = 50{,}257 \cdot 8$ bytes; continuation records eight, including final control. Initialization contributes 24 bytes and three backend communication calls, separately recorded. Zero local-preparation counters do not make sharing or material generation free. At $P = 169, G = 3$, F01 records 15,813 communication calls versus Tail's 13,824 ($+14.39\%$); F11 records 18,969 ($+37.22\%$). Appendix D defines these auxiliary counters and timing exclusions.

C Task quality and configuration comparisons

C.1 Small-model controls and development quality

Table 3 retains all FP32 controls. S is the private serial reference; T72 is a public-feature side branch. W0/W1 restrict joint support to the current/current-and-previous positions while preserving native context. LoRA3 all12 spreads rank-three Q/V updates across twelve blocks at the same parameter count as tail layouts, changing placement and capacity.

Table 3: All eleven FP32 conditions. Mean ± sample SD over three seeds; PPL is the across-seed mean. The first four rows form the matched new cohort; the remaining seven are previously audited historical cohorts, not retrospectively randomized into it.

Configuration	BLEU ↑	chrF++ ↑	PPL ↓
FULL_F01	40.125 ± 2.003	56.119 ± 0.596	2.6685
FULL_F11	40.547 ± 2.146	56.825 ± 0.298	2.6418
LoRA12 last3	44.405 ± 1.460	60.040 ± 0.915	2.6964
LoRA18 last2	40.726 ± 2.085	57.003 ± 1.011	2.8282
LoRA3 all12	53.630 ± 0.785	67.267 ± 0.036	2.3964
S	46.746 ± 2.037	60.848 ± 0.502	2.5293
T72	33.354 ± 2.726	49.133 ± 0.563	2.9168
W0_F01	36.956 ± 1.790	54.784 ± 0.923	2.6991
W0_F11	39.959 ± 0.823	55.647 ± 0.797	2.6749
W1_F01	37.653 ± 1.694	55.045 ± 0.898	2.7005
W1_F11	40.238 ± 2.483	55.759 ± 0.901	2.6739

Historical Q16 W0/W1 have distinct source schedules and no 256-update adaptation. Three-seed BLEU means are 35.528/35.487 (W0/W1 F01) and 38.524/38.305 (F11); all six seed-specific W1-minus-W0 BLEU and chrF++ intervals include zero. W0 saves only 0.437–0.470% (F01) or 1.279–1.329% (F11) of conditional bytes at matched lengths; natural lengths can reverse this comparison. Removing one history position alone yields no large saving. S8 retains six native FULL-minus-W1 BLEU contrasts and intervals, not a test comparing the two FFN settings' history effects. The FULL_F01 versus W1_F11 comparison remains exploratory.

Table 4 retains explored-development quality; the $P = 169, G = 3$ MPC communication is reported separately in Section 4.3. Adaptation improves F01 BLEU but worsens chrF++ and NLL. Adapted F01-minus-Tail intervals are $[-3.248, 1.803]$ for BLEU and $[-4.439, -1.542]$ for chrF++; crossing zero is not equivalence. `S1-small-dev-contrasts.csv` retains all fifteen comparisons and both metrics' saved intervals: 870 resampled semantic groups, fixed seed-41 models, no multiplicity adjustment or training-seed uncertainty (Appendix A).

Table 4: Complete cleartext fixed-point quality, seed 41. Each row covers 871 inputs and 66,937 target positions. Original denotes the trained 3,072-update checkpoint before numerical adaptation. "Cap" is the number of generations reaching the configured length limit.

Checkpoint	Structure	NLL ↓	PPL ↓	BLEU ↑	chrF++ ↑	Cap
Original	LoRA18 last2	1.0533	2.8671	41.330	54.297	29
	FULL_F01	0.9806	2.6659	37.303	55.901	36
	FULL_F11	0.9679	2.6325	36.976	56.597	45
Adapted256	FULL_F01	1.0198	2.7726	40.570	52.353	31
	FULL_F11	1.0053	2.7328	39.587	53.581	31
	LoRA18 last2	1.0614	2.8905	40.861	55.387	31

On eight fixed blocks (two structures, two prompts, two decisions), batch versus tokenwise native sources yield total variation at most 3.462×10^{-6} in FP32 and 8.453×10^{-5} in Q16. Arithmetic changes with fixed sources dominate on those blocks; neither this scope nor the independent sample count expands through reused cells or vocabulary coordinates.

C.2 Small-model numerical stages on fixed test inputs

The source is `challenge2017/test.json` in the Microsoft LoRA repository, revision `c4593f060e6a368d7bb5af5273b8e42810cdef90`, using the retained modified-triple/good-reference parser. Of 1,862 candidate groups, eight exact training overlaps and three explored-development overlaps are excluded, leaving 1,851. Three equal-rank input-length strata contribute 43, 43 and 42 groups without replacement. This budget and selection precede test-output inspection; all good references are retained. Usage-history checks cover accessible project records and matching-source searches, not inaccessible external use, so the entire official test set is not claimed untouched.

The same 128 groups, 343 references and 10,080 target positions evaluate six FP32 and six adapted-Q16 endpoints (Table 5; Figure 5). The FP32 evaluation was performed later using the original checkpoints, without new training or selection. It is post hoc, not a new test population. Each stage has 768 endpoint–group cells; the independent training units remain seeds 41/43/47. `S2-small-stage-scores.csv` preserves all twelve endpoint scores and length-cap counts.

FP32 uses CPU one-thread native sources; Q16 uses the declared numerical graph and 256 additional updates. Missing unadapted-Q16 endpoints and matched FP32 continuation prevent isolating arithmetic from extra training. Decoded scores, NLL counts and sums were checked against retained outputs and references; selected full-logit/cache sentinels do not independently recompute every full-vocabulary loss.

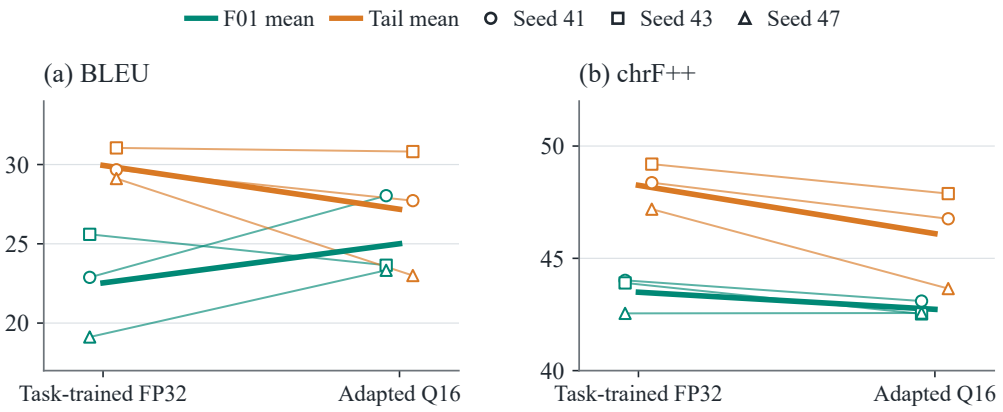


Figure 5: **The small-model stage change differs across quality metrics.** The same 128 groups and three training seeds are evaluated in FP32 and after 256 Q16-adaptation steps. Thin lines pair a seed; thick lines show means. F01's BLEU rises on average while chrF++ falls. Arithmetic, execution graph and extra training change together, so these lines do not isolate an adaptation effect. Table 5 reports all metric summaries; S2 retains every seed-level score.

Table 5: Small-model quality on the same 128 groups. Mean ± sample SD across seeds 41/43/47; PPL is averaged after computing it per endpoint. Each endpoint has 110,592 private parameters, 343 references and 10,080 target positions. Q16+256 adds 256 updates to the paired FP32 checkpoint. Individual scores and cap counts are in S2.

Stage	Model	BLEU	chrF++	NLL	PPL
FP32	F01	22.534 ± 3.253	43.492 ± 0.818	1.5724 ± 0.0065	4.8185 ± 0.0314
FP32	L18	29.946 ± 0.995	48.251 ± 1.009	1.4803 ± 0.0153	4.3945 ± 0.0673
Q16+256	F01	25.005 ± 2.630	42.730 ± 0.319	1.6344 ± 0.0123	5.1266 ± 0.0631
Q16+256	L18	27.179 ± 3.938	46.099 ± 2.191	1.5133 ± 0.0216	4.5423 ± 0.0982

C.3 Medium head-set and FFN-source comparisons

Architecture and matched training. GPT-2 medium has 24 blocks, width 1,024 and 16 heads, at revision `6dcaa7a952f72f9298047fd5137cd6e4f05f41da`. ScopePEFT places rank-32 adapters after blocks 21–23. H5 fixes $J_{22} = \{0, 1, 2, 3, 4\}$ and $J_{23} = \{6, 8, 10, 12, 14\}$; H16 uses every head. F01 uses $(f_{22}, f_{23}) = (0, 1)$ and F11 uses $(1, 1)$. Tail applies rank-24 Q/V LoRA in the final two blocks; Deep applies rank-16 Q/V LoRA in the final three. Both retain ordinary full attention and FFNs. All six configurations have 196,608 trainable parameters.

Seeds 41/43/47 each receive 3,072 FP32 AdamW updates without AMP or TF32, effective batch 32 and microbatch eight, using Appendix A's optimizer. Same-seed methods consume the same 98,304 presentations and target-token budgets (2,811,506; 2,810,027; 2,811,119). The nine initial H5-F01/Tail/Deep runs rotate method order across seed blocks. Nine later H5-F11/H16-F01/H16-F11 runs also retain the same-seed initial adapter tensors. All use the predetermined final checkpoint. Reused runs are not contemporaneously randomized replications; parameter matching does not equate expressivity.

Two evaluation populations. The initial 128 groups are those above, with 343 references and 10,080 target positions. The extension is exploratory, using already observed inputs. For the additional sample, excluding the initial groups and listed retained evaluation inputs from the 1,851 eligible groups leaves 1,723. Sorting by prompt length (semantic-key tie breaking) creates three equal-rank strata; sampling without replacement at seed 2026092401 selects 85, 85 and 86 groups. These 256 groups contain 689 references and 19,614 target tokens. Their identities and the eighteen checkpoint bindings are fixed before this evaluation, with no retuning or score-based selection. Disjointness refers to retained project history.

Both populations use greedy EOS-or-128 generation within 384 consumed tokens and separate teacher-forced scoring (Appendix A). Table 6 retains the original summaries; `S3-medium-fp32-scores.csv` contains all 36 distinct population/seed/configuration rows, including cap counts. The three initial H5-F01 rows shared by the earlier control and factor tables appear once. The populations' corpus scores are not pooled, and evaluating the same checkpoint twice does not add a training replication.

Persistent and conditional effects. H5-F01 has lower BLEU/chrF++ and higher NLL/PPL than both LoRA controls in every seed on both samples; its initial mean BLEU deficits are 3.892 against Tail and 5.046 against Deep. The tested H16-versus-H5 head-set contrast improves BLEU, chrF++ and NLL in all three seeds at both FFN settings. F11 improves these metrics in only one of three seeds at H5 and all three at H16 on the initial sample. There, H16-F11 exceeds Deep on every metric in every seed but loses to Tail's BLEU at seed 43 (-0.692).

On the additional sample, F11 at H16 instead lowers BLEU at seed 47 and worsens chrF++/NLL at seed 41. H16-F11 exceeds Deep's BLEU/chrF++ in every seed but improves NLL in only two; its BLEU exceeds Tail's in two. H16-F01 exceeds Tail's mean BLEU by 0.684, yet only one seed improves. Figure 3 shows the saved paired BLEU differences. These are descriptive configuration contrasts, not significance tests, a scaling law, or isolated effects of head count. Cross-size comparisons do not control scale alone.

Table 6: Medium FP32 quality for all six configurations. Mean ± sample SD across seeds 41/43/47; PPL is averaged over endpoints. Each population uses the same eighteen checkpoints and 196,608 private parameters. These are the previously reported summaries, without pooling the populations. S3 retains every seed-level score and cap count.

Model	BLEU	chrF++	NLL	PPL
Initial 128 groups				
H5-F01	20.958 ± 0.794	43.224 ± 0.905	1.4270 ± 0.0100	4.1664 ± 0.0417
H5-F11	20.205 ± 1.641	42.612 ± 1.510	1.4271 ± 0.0149	4.1668 ± 0.0618
H16-F01	25.898 ± 0.815	48.551 ± 0.787	1.3642 ± 0.0157	3.9129 ± 0.0611
H16-F11	28.530 ± 2.658	49.833 ± 0.136	1.3475 ± 0.0064	3.8477 ± 0.0247
Tail	24.850 ± 2.336	45.737 ± 2.782	1.4206 ± 0.0120	4.1399 ± 0.0498
Deep	26.003 ± 2.834	46.567 ± 1.260	1.3636 ± 0.0071	3.9102 ± 0.0277
Additional 256 groups; same checkpoints				
H5-F01	22.717 ± 2.148	44.624 ± 0.443	1.3526 ± 0.0049	3.8673 ± 0.0189
H5-F11	21.733 ± 0.429	43.790 ± 1.021	1.3534 ± 0.0041	3.8707 ± 0.0158
H16-F01	26.173 ± 3.102	48.941 ± 2.163	1.2742 ± 0.0095	3.5760 ± 0.0340
H16-F11	27.675 ± 1.662	49.213 ± 0.443	1.2638 ± 0.0134	3.5392 ± 0.0474
Tail	25.488 ± 1.807	45.959 ± 0.305	1.3336 ± 0.0079	3.7949 ± 0.0300
Deep	25.274 ± 1.587	46.807 ± 0.938	1.2722 ± 0.0131	3.5691 ± 0.0467

C.4 Final medium adapted-Q16 quality

H5-F01, H16-F11, Tail and Deep each receive 256 numerical-adaptation updates after 3,072 FP32 updates at seed 41, with 196,608 private parameters. Table 7 evaluates the same additional 256 groups (689 references, 19,614 target tokens), using EOS-or-128 generation and the 384-token context bound. The 1,024 method–group cells are four models on 256 inputs. Their final weight hashes match the separate MPC replay endpoints in Appendix D.3.

Table 7: Final medium adapted-Q16 quality, seed 41. BLEU and chrF++ are corpus generation metrics; NLL is token-weighted over all references and PPL is $\exp(\mathrm{NLL})$. Cap counts generations reaching 128 decisions, without distinguishing EOS on the last decision from the length limit. All four endpoints cover all 256 groups.

Model	BLEU	chrF++	NLL	PPL	Cap
H5-F01	23.810	42.439	1.441068	4.225205	14
H16-F11	27.848	47.772	1.355597	3.879077	15
Tail LoRA	23.537	43.273	1.374092	3.951485	22
Deep LoRA	26.188	46.409	1.291263	3.637378	17

H5-F01 gains 0.273 BLEU over Tail but loses 0.834 chrF++ and increases NLL by 0.067. H16-F11 gains 1.660 BLEU and 1.363 chrF++ over Deep, with worse NLL. No seed-level uncertainty, significance or equivalence is established. Missing Q16 H5-F11/H16-F01 endpoints prevent a factorial interpretation. Task metrics come from local numerical generation, not MPC on all quality inputs.

Verified core outputs and coverage. Metric settings follow Appendix A, including BLEU without effective order. Core verification recomputes the scores from retained decoded outputs and loss records, checks selected originals against archive manifests, and confirms endpoint bindings. The evaluation combines 538 earlier and 486 continuation generations; 26 copied results are reuse, not new observations. The earlier interrupted execution remains recorded as failed. Full audits of numerical vocabulary/cache arrays, quality archives and the resumed execution chain were not completed. Verified core scores and checkpoint bindings do not certify those portions.

D MPC measurements and trajectory validation

Accounting and validation conventions. Section 4.1 defines the distinct small warm-request and medium instrumented-session windows and byte counters. They are not pooled or interpreted as

complete service cost. Earlier $P = 169$ audit wall times include loading, reference calculations and file handling; that study reports no inference latency. Backend communication calls (including historical fields named `rounds`) count interface invocations, not protocol rounds, model calls or secure operators. Physical wire traffic, protocol rounds, secure-operator counts and separate complete preprocessing cost remain unmeasured.

All numerical checks below compare released vocabulary outputs with saved references conditioned on recorded native sources. Actual MPC KV remains unopened; cache induction assumptions are in Appendix B.1. Finite agreement is not a general correctness, overflow or privacy proof. Quality-output verification has the separate coverage stated in Appendix C.4.

D.1 Small fixed replay: request length and phase costs

For $G = 16$, six cleaned training prompts, distinct from quality inputs, use the same first fifteen reference tokens for both models, padded with EOS when shorter. Execution continues regardless of predicted EOS. Each prompt has two technical repeats at seed 41; prompts are independent workloads. Figure 6 retains the stage measurements, while `S4-small-g16-replay.csv` retains all 24 request times and application-byte counts. Decision-1 control lies between the first-logit and incremental windows, so their sum need not equal request time. Mean requests are shorter for F01 at every prompt, but short2's second repeat is slower (70.96 versus 70.07 seconds).

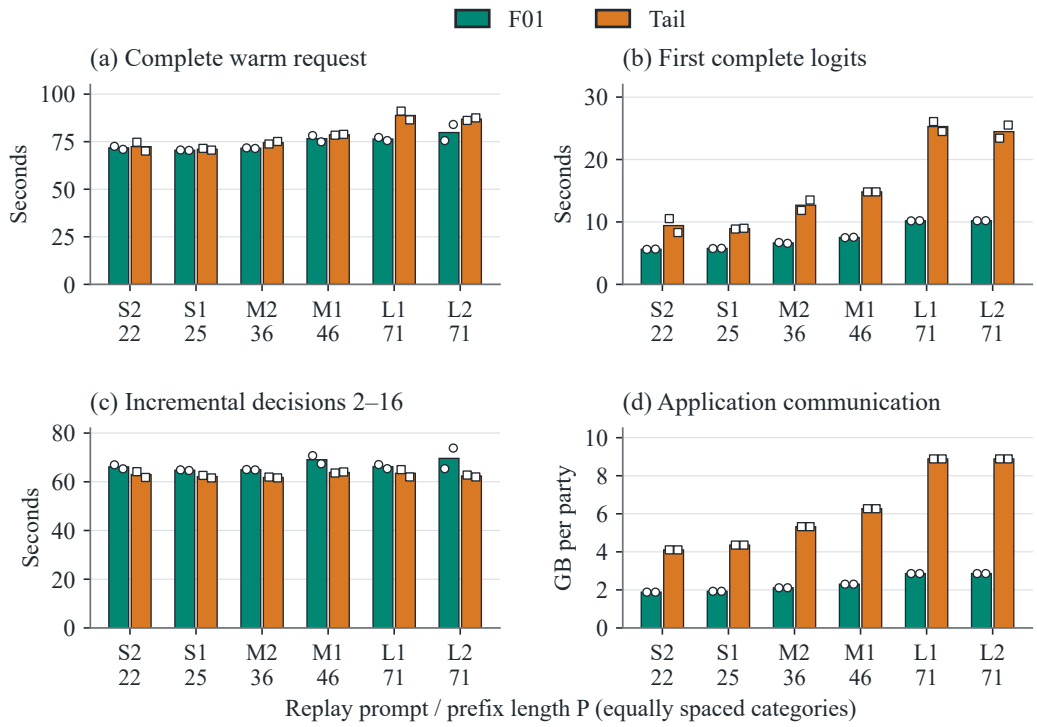


Figure 6: Matched $G = 16$ replay in loopback TFP simulation. Bars show means and open points show both technical repeats per model and prompt; no confidence intervals are estimated. Prompt IDs are equally spaced categories; P labels show their lengths. Tail denotes parameter-matched tail LoRA; GB is 10^9 application bytes per party. First-output savings coexist with slower increments.

The $G = 64$ extension reuses adapted checkpoints and short1 ($P = 25$), medium1 ($P = 46$), and long1 ($P = 71$), with two repeats and reversed method order. Both models consume the same 63-token reference continuation, right-padded with EOS. Decisions 1–63 include continuation control and decision 64 includes termination. Thus intermediate $G = 16/32$ points are cumulative prefixes, distinct from the earlier complete $G = 16$ requests. Their timings are not pooled across collection batches. Incremental time begins after decision-1 control and ends after decision-64 control; parent-observed cold windows are not production cost.

Table 8: Complete small-model $G = 64$ warm-request differences, F01 minus Tail in seconds. Both technical repeats and their saved mean are retained: long1 has opposite repeat directions. The $G = 16/32$ cumulative values from these continuing sessions remain in S5 and are not independent complete requests.

Prompt	Repeat 1	Repeat 2	Mean
short1	+12.414	+5.519	+8.966
medium1	+1.948	+4.840	+3.394
long1	-9.208	+3.070	-3.069

`S5-small-g64-pairs.csv` preserves both repeat differences and saved means at all three checkpoints. The mean F01-minus-Tail difference changes sign between decisions 9–10 for short1 and 49–50 for medium1 (within the retained 3–16 and 32–64 intervals). Long1's mean stays negative, but its second repeat reverses direction. The 64 points reuse three prompts and do not identify a universal crossover; no interpolation is used.

The existing $G = 16$ phase diagnosis reports a mean incremental difference of $+4.034433$ seconds: $+3.384037$ application, $+0.635536$ native preparation and $+0.014860$ across the other five phases. These instrumented windows include synchronization and do not isolate individual secure operators or network delay.

D.2 Small actual self-feedback and numerical coverage

Table 9: New actual self-feedback sessions, cap 32, seed-41 adapted endpoints. Unequal output lengths preclude interpreting the request-time ratio as matched-workload acceleration.

Prompt	Model	P	G	Stop	Request seconds
short1	F01	25	17	EOS	75.364
short1	L18	25	15	EOS	66.191
medium1	F01	46	18	EOS	83.716
medium1	L18	46	32	CAP	150.775
long1	F01	71	22	EOS	104.676
long1	L18	71	19	EOS	102.316

The earlier $P = 169$, cap-128 trajectories are retained in `S6-small-natural-long.csv`, including all byte and interface-call counts. Original F01/F11/Tail generate 44/128/33 decisions; adapted F01/F11/Tail generate 34/30/42. Each has $G - 1$ feedback appends and stops without an extra append. A count of 128 does not distinguish EOS at the last decision from the length cap. Unequal generated lengths preclude matched-workload time or byte ratios.

The initial collection has 24 $G = 16$ replay sessions, six cap-32 self-feedback sessions and two pilots; two earlier failed short attempts remain outside those successful sessions. Its 513 full-vocabulary vectors include 123 from actual feedback. The twelve $G = 64$ sessions check 768 vectors; the six earlier cap-128 trajectories check 311. All match the saved references under the conventions above. These coverage counts, and the 94,357,504 earlier reference-cache coordinate checks, are not independent scientific observations.

D.3 Medium final-Q16 fixed replay

The four endpoints use the final adapted weights in Appendix C.4, with Appendix B's Q16 schedule and factored LoRA; encoded inputs are not quantized twice. A conservative bound check interrupted H16-F11 adaptation; it resumed from saved parameters, optimizer and RNG using a tighter sufficient bound at the unchanged integer threshold. The function and final-checkpoint rule were unchanged.

Native features, encoding and native-head AV are prepared before the measured session. Alice's inputs enter from rank 0 and Bob's private parameters from rank 1; public weights remain public. Beaver multiplication uses the trusted TFP, only Alice receives logits, and caches stay shared under Section 2's simulation assumptions. Prefill consumes $P - 1$ tokens for $P = 31, 54, 84$; the final

prompt token and fifteen fixed reference tokens produce $G = 16$ outputs at consumed lengths P through $P + 15$, regardless of predicted EOS. Each method repeats each prompt twice. Medium has no natural self-feedback measurement.

Table 10: Medium final-Q16 fixed-replay cost, seed 41. Six sessions per configuration: three prompts, two technical repeats each. Means weight prompts equally; the range mixes prompt differences and repeat variability. GB is 10^9 rank-0 framework bytes. Session time includes startup and offline verification but excludes prior native preparation/copying. Figure 2 and S7 retain both repeats at each prompt.

Model	Mean seconds	Session range	Mean GB
H5-F01	83.874	79.999–87.710	3.193
Tail	96.974	84.679–109.291	9.669
Deep	144.739	120.583–175.800	17.100
H16-F11	115.208	103.051–130.536	10.034

`S7-medium-g16-replay.csv` preserves all 24 sessions. H5-F01 saves 60.06%, 66.50% and 70.50% of Tail's bytes and 4.70%, 12.95% and 20.91% of mean session time at increasing P. H16-F11 costs more than Tail and less than Deep in every paired session. The 41.32% saving relative to Deep compares overall mean bytes, rather than averaging prompt-wise percentages. No long-generation or WAN result is inferred.

Auxiliary mean communication-call counts are 80,165 (H5-F01), 69,233 (Tail), 105,715 (Deep) and 96,281 (H16-F11). H5-F01 has 15.79% more calls than Tail despite fewer bytes; their meaning is defined above. Four short pilots precede the 24 retained sessions, whose $24 \cdot 16 \cdot 50{,}257 = 19{,}298{,}688$ released integers match saved references. This fixed replay does not measure natural-generation quality.

E Privacy and artifact boundaries

Section 2's functionality and semi-honest/TFP assumptions delimit input privacy. Model extraction, malicious/adaptive corruption, private training and implementation side channels remain outside the claim. Architecture selection from sensitive task data needs an additional leakage policy. Trusted-custodian offline reconstruction is not an online-party output; file checks provide neither cryptographic attestation nor OS isolation.

The paper-source supplement contains aggregate data and figures. The separate minimal code package includes adapter checkpoints for the final medium Q16 endpoints; backbone weights and data remain external. Raw execution archives, logits, KV tensors and credentials are excluded.