SLP Verifiable Inference: Proof-of-Concept Experiment Report

Sampled layerwise proofs on GPT-2, TinyLlama-1.1B and Llama-2-70B: sealing, proving, verification, adversarial rejection, cost curves, quantization fidelity and the security limits of sampling.

Version 1.0 · 2026-09-17

__ACTIONS__

> **CITATION**
>
> These experiments, figures and data files are published by the **TrueOpen.ai team**. When citing or redistributing them, in whole or in part, state that the source is the TrueOpen.ai team and link to www.trueopen.ai. The same requirement is repeated in the download bundle's README, in CITATION.txt, and in the header of every CSV file.

Scope: all experiments of the SLP (Sampled Layerwise Proofs) proof-of-concept phase (August–September 2026). This report states measured results and their interpretation only; the protocol design and the development plan are covered by the technical paper and the roadmap. All raw logs and extracted data tables are released together with this report (Section 6).

Summary

1. **The protocol works end to end.** On GPT-2 (124M), TinyLlama-1.1B and Llama-2-70B, the seal-then-sample layerwise proof completed sealing, proving and verification. Adversarial cases — tampered output, tampered commitment manifest, tampered boundary commitment, missing proof, substituted model, mismatched sampling policy, mismatched beacon — were all rejected by the verifier.
2. **70B is reachable.** Llama-2-70B ran end to end on a CPU server with 2 TB of memory: sealing 163 chunk boundaries and proving 5 of them took 21 minutes, the proof is 4.34 MiB, and verification takes 46 seconds without access to the model weights. Peak memory stayed under 2 TB, whereas the same pipeline without streaming is estimated to need about 2.8 TB.
3. **Proving cost is driven by model size and is nearly independent of token count.** On TinyLlama, raising the context from 8 to 16 tokens changed the proving time from 94.8 s to 99.0 s; packing 12 concurrent requests into one proof took 181.9 s in the proving stage, about 6.5× less than proving them separately at the measured single-proof cost.
4. **Fidelity of the quantized model to the original is measurable and has improved substantially.** Over 334,705 predicted positions of the WikiText-2 test split, the quantized model agrees with the f32 model on the top-1 token at 84.8%–84.9% of positions, with perplexity 12.94–12.95 (f32: 11.94) and a

KL divergence of 0.118 nat; the pre-fix quantization path reached only 13.7% agreement on a comparable evaluation.

5. **Detection power of sampling equals coverage.** For a forgery confined to unsampled chunks, the single-audit detection probability is the sampling coverage (1.9% at 70B with 3 sampled chunks); repeated audits and economic mechanisms must amplify it. This is an inherent property of the protocol and is reported as such.

6. **Pure Fiat–Shamir sampling can be ground.** An attacker who forges a boundary can re-seal repeatedly until the sample avoids the forged chunks; each attempt costs 12.5 ms on GPT-2 and the mean number of attempts matches the theoretical 1/q. With a beacon introduced after sealing, the escape fraction returns to the theoretical q.

7. **Model versions reproduce deterministically; inference execution does not yet.** Ten independent registrations produced identical fingerprints; for the same quantized model and input, 16 of 20 repeated GPU inferences were token-for-token identical.

1. Background and Objectives

SLP is a verifiable-computation protocol for large-language-model inference. An inference is partitioned into layer groups (chunks); the activations at every chunk boundary are committed with polynomial commitments; all commitments together with the claimed input and output are written into the Fiat–Shamir transcript before the chunks to be proven are drawn. Chunks touching the model's input or output are always proven, binding the proof to the specific prompt and answer. The verifier can escalate to a full proof over the same commitments. The experiments answer five questions:

- Does the protocol accept honest proofs and reject tampering on real models (Section 4.1)?
- Does it scale to 70B, and at what cost (Section 4.2)?
- How does proving cost vary with model width, sequence length and concurrency (Sections 4.3, 4.4, 4.5, 4.7)?
- How close is the proven quantized model to the original floating-point model (Section 4.6)?
- Where are the security limits of sampling (Sections 4.8, 4.9)?

2. Experimental Setup

Label	Hardware	Software	Use
Mac	Apple M3 Pro, 18 GB unified memory, CPU only	macOS, Rust release build	GPT-2 (124M, Q8_0) end to end; synthetic layer cost sweeps; grinding experiments
GPU box	NVIDIA RTX 4090 24 GB; host 2× AMD EPYC 7742 (128 cores / 256 threads), 1007 GB RAM	Ubuntu 24.04, CUDA 12.8	All TinyLlama-1.1B-Chat-v1.0 experiments. Inference and calibration run on the GPU; **proving and verification run on the host CPU**
CPU 2 TB	256-thread CPU server, 2 TB RAM, 1.6 TB local disk (CPU model not recorded)	Ubuntu, Rust release build	Llama-2-70B end to end; inference is single-threaded integer execution

Common configuration: 12-bit activation quantization with the residual stream kept at about 24 bits; KZG-family polynomial commitments over the BN254 curve; Blake3 transcript hashing; LLM chunking cuts after every residual addition plus a standalone embedding chunk and a standalone final-projection chunk, giving 2L+3 chunks (GPT-2: 27, TinyLlama: 47, Llama-2-70B: 163); the two end chunks are anchors and are always proven; default sample sizes are 3 for GPT-2 and Llama-2 and 2 for TinyLlama. Unless stated otherwise, every figure is a single measured run.

3. Experiment Index

ID	Experiment	Model	Platform
E01	End-to-end demo: sealing, sampled proof, verification, tamper rejection, tolerance check	GPT-2	Mac
E02	End-to-end five-stage pipeline	Llama-2-70B	CPU 2 TB
E03	Synthetic feed-forward block cost sweep (d = 256 → 4096)	synthetic	Mac
E04	Attention chunk vs. MLP chunk cost against sequence length	GPT-2	Mac
E04b	Proving-stage time breakdown	TinyLlama	GPU box
E05	Packed batch proving (B = 3 / 6 / 12), leak check, equivalence check	TinyLlama	GPU box
E06	Verifiable inference service simulation (12 requests, with a tampering scenario)	TinyLlama	GPU box
E07	Quantization fidelity, generation level (16 prompts)	TinyLlama	GPU box
E08	Early corpus evaluation (superseded by E09, kept for comparison)	TinyLlama	GPU box
E09	Standard-corpus fidelity evaluation (full WikiText-2 test split)	TinyLlama	GPU box
E10	Full proof vs. sampled proof	TinyLlama	GPU box
E11	Determinism: 10 independent registrations and in-process repeated inference	TinyLlama	GPU box
E12	Grinding attack measurement and beacon fix	GPT-2 / synthetic	Mac

4. Results

4.1 Protocol Correctness and Adversarial Negatives (E01, E06, test suite)

GPT-2 end-to-end demo (Mac, re-measured 2026-09-17):

Item	Value
Inference	6.71 s
Seal 27 boundaries + prove 5/27 chunks (2 anchors + 3 sampled)	28.0 s
Proof size	916 KiB
Sampled-proof verification	0.16 s, accepted
Verification after tampering with an output token	rejected (argmax constraint of the output anchor fails)
Tolerance check (exact integer arithmetic, d = 768)	honest accepted; tampered matrix product rejected
Tolerance check on the fp16 model	honest accepted; gross tampering rejected; **drift within the tolerance band accepted** (the inherent attack surface of the floating-point serving path, shown as is)

Adversarial negatives in the test suite (all rejected): missing chunk proof, tampered commitment manifest, tampered boundary commitment, tampered input/output, verification under a different model context, mismatched sampling policy, mismatched beacon. Positive cases cover model depth {8, 12} × chunking {default, 2, 4, 6} × sample size {0, 1, 3, all}, plus a serialization round trip of the verifier context. In the service simulation (E06, Section 4.5), the window in which the worker tampered with one generated token before publishing was rejected.

4.2 Scale: Llama-2-70B End to End (E02)

Single inference, context 8, 3 sampled chunks plus 2 anchors, integer weights persisted to disk and weight polynomials committed in streaming fashion.

Stage	Time	Peak memory / result
One-time registration: load and quantize (integer weights persisted one by one)	8,906 s	387 GB resident
One-time registration: commit all weight polynomials (streaming)	10,553 s	about 60 GB working set
Per inference: integer inference producing the trace	48,706 s	integer weights resident (single-threaded)
Per inference: seal 163 boundaries + prove 5 chunks	1,259 s	proof 4.34 MiB
Per inference: verify	46.3 s	CPU only, no weights
Total	69,500 s (19.3 h)	under 2 TB throughout

The same pipeline without streaming is estimated to need about 2.8 TB; an earlier non-streamed commitment attempt held 557 GB resident and had not finished after 5.5 hours (raw log `run70b.log`). Inference accounts for 70% of the total time and is the current bottleneck; it is a runtime-implementation issue, not a protocol issue.

4.3 Cost Curves (E03, E04, E04b, GPU inference acceleration)

Single-layer cost (E03, Mac). Synthetic feed-forward block (two linear layers, 4× expansion, non-linearity), sequence length 128:

d	Parameters	Prove	Verify	Proof size	Peak memory
256	0.53M	1.61 s	0.01 s	121 KiB	0.51 GiB
512	2.10M	4.45 s	0.01 s	135 KiB	2.15 GiB
1024	8.39M	5.97 s	0.02 s	145 KiB	2.61 GiB
2048	33.6M	9.89 s	0.02 s	156 KiB	4.21 GiB
4096	134.2M	25.08 s	0.03 s	170 KiB	7.84 GiB

Proving time grows sublinearly in the parameter count (local exponents between adjacent points range from 0.2 to 0.7; the end-to-end fit is about 0.66); proof size grows slowly with d.

Attention and MLP chunks against sequence length (E04, GPT-2, Mac):

Sequence length	Attention chunk prove	MLP chunk prove	Attention proof size
16	3.35 s	3.94 s	581 KiB
32	3.66 s	4.26 s	648 KiB
64	4.39 s	4.95 s	717 KiB
128	5.78 s	5.68 s	788 KiB
256	10.05 s	7.79 s	860 KiB

The quadratic sequence-length term of the attention chunk becomes visible from 256 tokens. Longer contexts were not measured in this round and are a priority for the next phase.

GPU inference and proving cost (TinyLlama, context 8). Before and after the device-side lookup-table optimization: quantization 170.2 s → 17.9 s, inference 101.2 s → 14.2 s; proving (seal 47 boundaries + prove 4 chunks) 95.2 s → 100.5 s (proving runs on the host CPU and is unaffected). Context 8 → 16: inference 12.8 s → 105.1 s (the latter from a pre-optimization run), proving 94.8 s → 99.0 s, proof size 0.43 MiB → 1.41 MiB. Proving-stage breakdown (E04b): constraint-claim generation 40.2%, witness-context construction 20.5%, proof generation 18.5%, witness polynomial commitments 8.8%, lookup-table claims 6.3%, challenge storage 5.7%.

4.4 Packed Batch Proving (E05, TinyLlama, slot length 16, 4 generated tokens per slot, 2 samples)

Concurrency B	Packed inference	One packed proof	Proof size	Verify	Separate proofs
3	15.2 s	116.4 s	2.91 MiB	2.5 s	96.9 + 105.4 + 93.3 = 295.6 s (measured, 2.54×)
6	76.5 s	144.3 s	3.20 MiB	2.9 s	≈ 4.1× at the measured single-proof cost of about 99 s
12	107.2 s	181.9 s	2.11 MiB	2.1 s	≈ 6.5× at the measured single-proof cost of about 99 s

Each additional concurrent inference adds about 7 s to the proving stage. **Leak check**: after replacing the contents of all neighbouring slots, the outputs of the three slots were token-for-token unchanged; the block-diagonal mask leaks nothing across slots. **Equivalence check**: slot 0 matches a standalone inference token for token; slots 1 and 2 do not, because packing shifts their absolute positions and the quantized rounding of the rotary position encoding changes accordingly. Packed inference is therefore not token-for-token equivalent to standalone inference outside slot 0; this is a known limitation.

4.5 Verifiable Inference Service Simulation (E06, TinyLlama)

Twelve requests arrive in three windows of [4, 6, 2]; slot length 16, 4 generated tokens per request, 2 samples; in the last window the worker tampers with one answer token before publishing.

Window	Requests	Tokens	Batched decode	Prove	Verify	Proof	Result
0	4	36	101.7 s	118.0 s	2.2 s	1,755 KiB	accepted
1	6	62	60.1 s	140.0 s	2.4 s	1,920 KiB	accepted
2	2	16	10.0 s	108.8 s	2.9 s	2,693 KiB	**rejected** (tampering caught by the output anchor)

One-time registration (deterministic calibration, quantization, weight commitments, publication of a 118 KiB verifier context): 417 s. Proving total 366.9 s, i.e. 30.6 s per request on average; verification total 7.6 s.

4.6 Quantization Fidelity (E07, E08, E09, TinyLlama)

Generation level (E07): 16 prompts not used in calibration, 8 tokens generated greedily per prompt, compared with the f32 model.

Configuration	First-token match	Per-token match	Full-sentence match	Teacher-forced step match
Before fix: generic observer (residual stream squeezed to 12 bits)	12%	5%	0%	16%
LLM observer, 1 random calibration sequence	88%	53%	31%	89%
LLM observer, 32 random calibration sequences	100%	52%	31%	87%
LLM observer, 32 WikiText-2 sequences, MinMax	88%	55%	31%	91%
Same corpus, 0.1% / 99.9% percentile clipping	6%	1%	0%	3%

Conclusion: fidelity is determined by the observer (the bit width of the residual stream); the choice of calibration corpus matters little; percentile clipping is unusable for LLMs. Two re-runs of the same configuration gave per-token / full-sentence matches of 53% / 31% and 49% / 25%; the difference comes from GPU inference non-determinism (Section 4.8).

Standard corpus (E09): WikiText-2 test split, 655 non-overlapping windows of 512 tokens, teacher forcing, 334,705 predicted positions.

Model	Perplexity	Top-1 hit	Top-1 agreement with f32	Top-5 overlap	KL(f32 ‖ quantized)
f32 original	11.94	50.5%	100%	100%	0
Quantized, corpus calibration (32 sequences, MinMax)	12.95	49.2%	84.8%	82.5%	0.1182 nat
Quantized, seeded random calibration (seed 42, MinMax)	12.94	49.2%	84.9%	82.5%	0.1182 nat

The cumulative metrics stabilise from window 30 onwards (agreement 84.7%–85.3%); the two calibrations are indistinguishable over 335 thousand positions. An earlier evaluation on 1,984 positions of the train split (E08: agreement 76.4%–77.3%, pre-fix generic observer 13.7%) is superseded by E09 and kept only for comparison.

4.7 Full Proof vs. Sampled Proof (E10, TinyLlama, context 8)

Same trace and same weight commitments, proven with increasing sample sizes; times include sealing all 47 boundaries.

Configuration	Proven chunks	Seal + prove	Proof size	Verify
Anchors only (s = 0)	2 / 47	62.5 s	75.5 KiB	0.9 s
s = 2	4 / 47	159.1 s	1.25 MiB	1.5 s
s = 5	7 / 47	276.8 s	1.77 MiB	1.7 s
s = 10	12 / 47	324.7 s	5.08 MiB	3.7 s
All (tier T3)	47 / 47	1,256.1 s	25.98 MiB	17.7 s
Single-transcript full baseline	—	**did not complete** (prover stack overflow)	—	—

At s = 5, proving time is 22% of the full proof, proof size 6.8%, and verification time 9.6%. The chunked full proof (tier T3) itself serves as the escalation baseline.

4.8 Determinism (E11, TinyLlama)

Ten independent processes each performed calibration and registration and then repeated inference on the same input twice in-process.

Quantity	Result
Calibration digest (10 registrations)	10/10 identical
Quantized-weight fingerprint (10 registrations)	10/10 identical
Activation-scale fingerprint (10 registrations)	10/10 identical
In-process repeated inference identical to the first run, token for token	16/20 (1–2 mismatches in each of 3 processes)

The model version (weight fingerprint + scale fingerprint) reproduces deterministically; the non-determinism lies in GPU inference execution itself, pointing to floating-point reduction order, and is to be eliminated in the next phase.

4.9 Security Limits of Sampling (E12 and detection probability)

Detection probability (computed). With M non-anchor chunks, b forged chunks and s sampled chunks, the single-audit detection probability is $P = 1 - C(M-b, s) / C(M, s)$; for b = 1, $P = s / M$.

Model (M)	s = 2	s = 3	s = 5	s = 10	s = 20
GPT-2 (25)	8.0%	12.0%	20.0%	40.0%	80.0%
TinyLlama (45)	4.4%	6.7%	11.1%	22.2%	44.4%
Llama-2-70B (161)	1.2%	1.9%	3.1%	6.2%	12.4%

For a forgery confined to unsampled chunks, single-audit detection equals coverage and cannot be amplified by the proof system itself; repeated audits, escalation to a full proof, and stake-and-penalty

mechanisms must carry it.

Grinding attack and beacon (E12). The attacker forges the boundary shared by two adjacent non-anchor chunks (b = 2) and, under the beacon-less pure Fiat–Shamir schedule, re-seals repeatedly until the sample avoids the forged chunks; 200 repetitions per s.

Model	s	Theoretical escape q	Theoretical mean attempts 1/q	Measured mean attempts	Max attempts	Time per attempt	Measured escape fraction with beacon (2000 beacons)
GPT-2 (M = 25)	1	0.920	1.09	1.11	3	12.5 ms	0.912
GPT-2	3	0.770	1.30	1.35	5	12.6 ms	0.755
GPT-2	5	0.633	1.58	1.64	6	12.5 ms	0.617
GPT-2	10	0.350	2.86	3.00	16	12.5 ms	0.339
Synthetic 26-layer (M = 24)	1 / 3 / 5 / 10	0.917 / 0.761 / 0.620 / 0.330	1.09 / 1.31 / 1.61 / 3.03	1.11 / 1.34 / 1.60 / 3.00	4 / 4 / 5 / 13	0.2 ms	0.916 / 0.745 / 0.602 / 0.328

Without a beacon, the cost of an attack is one boundary re-commitment per attempt (milliseconds), not a model re-execution; with a beacon outside the attacker's control introduced after sealing, the escape fraction returns to the theoretical q.

5. Limitations and Notes

- Except for E09 and E12, all figures are single measured runs; no variance across repeated runs is given.
- The 70B inference stage is single-threaded integer execution (13.5 hours); peak memory of the proving stage was not recorded separately; the CPU model of that host was not recorded.
- TinyLlama proving and verification ran on the host CPU of the GPU server; the GPU handled only inference and calibration. The "GPU box" figures should be read accordingly.
- All 70B and TinyLlama experiments use context lengths of 8–16 (E09 is a 512-token teacher-forced evaluation without proving); proving cost at production context lengths has not been measured.
- Packed inference is not token-for-token equivalent to standalone inference outside slot 0 (Section 4.4).
- The single-transcript full-proof baseline in E10 did not complete because of a prover stack overflow; the chunked full proof is used as the comparison.

- The proven object is a 12-bit quantized canonical model, not the floating-point serving model; the difference is measured by the metrics of Section 4.6.
- The detection-probability table is analytical; the security of sampling is economic, not cryptographic.

6. Data and Reproduction

Raw logs and data tables are in the accompanying repository:

- `raw/mac-m3pro/`, `raw/gpu-rtx4090/`, `raw/cpu-2tb-70b/`: raw run logs (terminal colour codes removed and local paths redacted; otherwise verbatim).
- `data/E01_*` … `data/E12_*`: CSV tables extracted from the logs; column descriptions in `README.md`.
- `tools/build_dataset.py`: script that rebuilds every data table from the archived logs.

All models and datasets are public releases: GPT-2 (Q8_0 GGUF), TinyLlama/TinyLlama-1.1B-Chat-v1.0, Llama-2-70B, WikiText-2 raw v1 (test split for evaluation, train split for calibration). The experiment code will be open-sourced in a later phase.

Appendix: Terminology

- **Chunk**: a contiguous subgraph of the model's computation graph; the unit of proving and commitment.
- **Sealing**: committing to the activations at every chunk boundary and writing them, together with the claimed input and output, into the transcript.
- **Anchor**: a chunk touching the model's input or output; always proven.
- **Sample size s**: the number of non-anchor chunks drawn and proven per audit.
- **Tiers T0–T3**: four audit strengths — hash chain, tolerance check, sampled proof, full proof.
- **Beacon**: a public random value outside the prover's control, introduced after sealing and before sampling.
- **Canonical model**: the quantized model version fixed at registration and identified by its weight fingerprint and scale fingerprint.

Back to papers and experiments →