When Fine-Tuning Made a 5B Model Worse

Viridian Experimental Progress Trial 001

An evidence-bounded study of instruction tuning, regression, and claim qualification

13 October 2026 · Technical case study · Revised evidence review · Publication draft

> **Core finding:** A one-epoch QLoRA intervention reduced K2 Horizon's pass rate on a frozen 300-case instruction-following evaluation from **79.33% (238/300)** to **35.33% (106/300)**. The paired change was **−44.00 percentage points** (95% paired bootstrap interval **−49.67 to −38.33 pp**). The locally executed Viridian Osmium Audit v0.1 reported `NOT_SUPPORTED` for the narrowly declared improvement claim. This result does **not** establish deterioration in all general language-model capabilities.

ORIGINAL CHECKPOINT	FINE-TUNED ADAPTER	PAIRED CHANGE
79.33%	**35.33%**	**−44.0 pp**
238 / 300 correct	106 / 300 correct	Candidate minus baseline

Abstract

Fine-tuning is an optimisation procedure, not by itself evidence that a model has become more capable. To test that distinction, Viridian conducted a bounded, local instruction-tuning experiment on a hash-identified K2 Horizon checkpoint containing **5,058,255,360 unique parameters**. An assistant-only, single-epoch NF4 QLoRA intervention used **3,000** deterministically selected OpenAssistant conversations. A protocol and generated evaluation cases were recorded as frozen before baseline inference. Both original and fine-tuned models were scored under matching generation conditions on the same **300 objective instruction-following cases**.

The mean observed training loss fell from **1.482** across the first 20 optimiser steps to **1.404** across the final 20, but confirmation pass rate fell by 44 percentage points. The losses and test results measure different objectives; the loss trend cannot establish progress on the held-out tasks. A separately performed paired statistical analysis identified a large negative difference. The Osmium Audit v0.1 interface checked declared comparison identities and the supplied point-score direction, returning `NOT_SUPPORTED`. A controlled synthetic evaluator-version mismatch returned `INSUFFICIENT_EVIDENCE` as expected. Neither result demonstrates the more extensive checks of Osmium Advanced V5, which were unavailable in this execution.

The case study demonstrates one intervention that failed its predeclared goal, and an audit interface that declined to endorse that unsupported claim. It does not establish the root cause of the regression

or that the model would behave similarly across other tasks or training seeds. **The supplied 300-pair raw confirmation prediction logs and 375 training-metric records have now been checked against their recorded SHA-256 values and aggregate scores.** This permits a deeper post-hoc failure-mode analysis, but does not constitute model re-execution or independent verification of the unavailable evaluator source, tokenizer, and binary model artifacts.

1. Why this experiment matters

Applied model development often uses training loss as a convenient signal that parameter updates are doing something useful. But successful optimisation of the training objective does not guarantee success on a downstream objective. An instruction-tuned model may learn stylistic preferences, response formats, or behaviour specific to its training distribution while regressing on tasks it could previously complete.

Trial 001 was designed around a narrower question: **did the specific fine-tuning intervention improve exact instruction adherence on a predeclared, generated task distribution?** It was not framed as an evaluation of general intelligence, broad competitiveness, or production-readiness.

The protocol specified a positive hypothesis and an improvement threshold of **+2 percentage points** before the experiment. A negative outcome was admissible by design. [E1, E2]

2. Model, training data, and environment

Model identity. The local model card labels the checkpoint `IFM/K2-Horizon-3.7B`, and its path refers to “3.7B”. Inspection of the local BF16 safetensors and a meta-device instantiation instead recorded **5,058,255,360 unique parameters**, across **36 shards**. For this case study, **“5B” describes the inspected local checkpoint**, not an independently verified identity for all publicly distributed K2 Horizon variants. The base content-set SHA-256 begins `136508ea5183ed2a` and is recorded in full in the evidence register. An older resident profile referring to a 3.7B/8-bit model was **not reused**. [E3]

Dataset choice. The original proposed dataset, OpenHermes 2.5, was not selected because the locally inspected card lacked a licence field and noted an unavailable upstream source. The recorded experiment instead used **`OpenAssistant/oasst1`**, pinned to revision `fdf72ae0827c1cda404aff25b6603abec9e3399b`. Its repository declared Apache-2.0; that declaration was recorded, **not independently cleared at the level of every upstream contribution**. [E4]

Selection. The subset contained **3,000 reviewed English assistant conversations**, selected deterministically from eligible terminal assistant messages, with review and duplication filters and a 1,024-token sequence maximum. Source message IDs, sampling seed, subset digest, and exclusion criteria were preserved. Exact and thresholded near-duplicate checks against all **360 development and confirmation prompts** found no matches; this is a bounded lexical check, not proof against semantic overlap or base-model pretraining contamination. [E1, E4]

Execution. The trial ran locally on **Windows 11**, an **Intel i7-12700K**, **NVIDIA RTX 3080 Ti (12 GiB)**, and **32 GiB RAM**. Training used Python 3.11.16, PyTorch 2.11.0+cu128, Transformers 5.17.0, PEFT 0.18.0, and bitsandbytes 0.50.2. No cloud compute was reported. [E5]

Training intervention

Property	Recorded setting
Method	One-epoch, assistant-only supervised NF4 QLoRA
Training examples	3,000
LoRA	Rank 16; alpha 32; dropout 0.05; `q_proj` , `v_proj`
Optimiser and learning rate	AdamW; 2e−4 ; cosine decay; 3% warmup
Quantisation / compute	4-bit NF4 with double quantisation; BF16 compute
Batch, accumulation	Microbatch 1; accumulation 8; effective batch 8
Sequence / training seed	1,024 tokens; `4021001`
Completed steps	375 optimiser steps
Trainable parameters	5,898,240 (about 0.117% of base count)
Wall-clock training	4,680.5 s (about 78.0 min)
Loss, first 20 / last 20 steps	**1.4824 / 1.4036**

The isolated preflight reported finite loss and adapter save/reload. The base checkpoint hash remained unchanged; the final LoRA adapter was stored separately. **Only one training seed was run.** Near-full dedicated VRAM use was observed during training; allocator-reserved memory counters larger than physical VRAM are not interpreted as physical device utilisation. [E6]

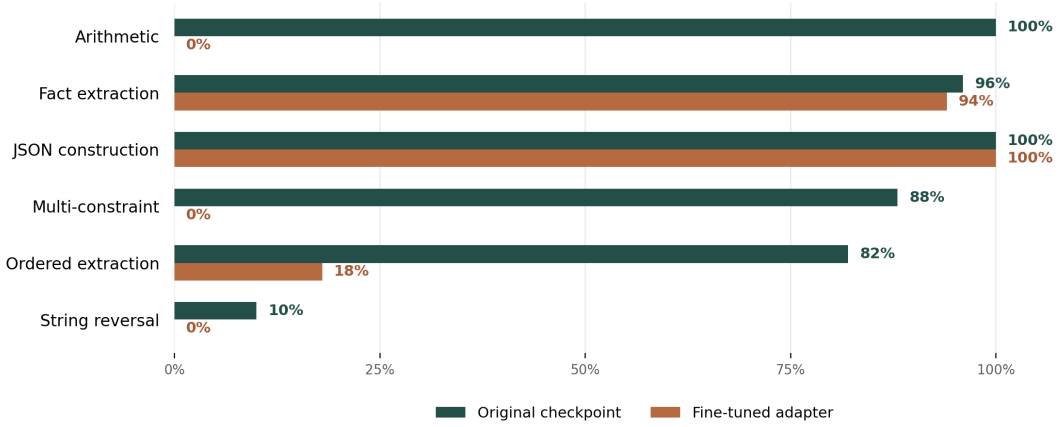
3. A frozen, matched confirmation test

The evaluation consisted of **60 development cases** and **300 confirmation cases** generated under fixed seeds. The final set had **50 cases in each of six categories**: arithmetic, string reversal, ordered extraction, JSON construction, structured fact extraction, and multi-constraint aggregation. Objective validators determined case-level success. The protocol and case manifests report having been hashed prior to baseline inference and training. [E1, E2]

The comparison held constant the tokenizer, chat template, high-reasoning setting, greedy decoding, batch size, and **256 maximum new tokens**. An unfinished response, missing final-answer delimiter, or execution error was counted as failure. That is the **predeclared benchmark contract**, but it also means some scores reflect completion under this output budget rather than unconstrained task-solving ability. [E1, E7, E8]

The reported original-model development result was **50/60 (83.33%)**; the candidate scored **21/60 (35.00%)**. Neither development score was the primary predeclared endpoint. The table below reports the untouched 300-case confirmation comparison. [E7, E8]

4. Confirmation results



Per-category exact-pass rates for original and fine-tuned K2 Horizon checkpoint

Confirmation category (50 cases each)	Original	Fine-tuned	Change
Arithmetic	50/50 · 100%	0/50 · 0%	−100 pp
Fact extraction	48/50 · 96%	47/50 · 94%	−2 pp
JSON construction	50/50 · 100%	50/50 · 100%	0 pp
Multi-constraint aggregation	44/50 · 88%	0/50 · 0%	−88 pp
Ordered extraction	41/50 · 82%	9/50 · 18%	−64 pp
String reversal	5/50 · 10%	0/50 · 0%	−10 pp
All cases	**238/300 · 79.33%**	**106/300 · 35.33%**	**−44.00 pp**

The baseline and candidate passed **104** of the same items; **60** failed under both. The baseline passed **134** items that the candidate failed, while the candidate passed only **2** items that the baseline failed. This pattern is strongly asymmetric and cannot be explained as a small scoring fluctuation in a handful of cases. [E9]

The recorded inference runtime fell from **3,326.4 seconds** for the original model to **728.6 seconds** for the adapter model, about **78.1% less time**. A later, retrospective tokenizer reconstruction from saved text reports **37,211 generated tokens** for the original model and **6,192** for the fine-tuned model (means of **124.04** versus **20.64** tokens per case). Both evaluated the same **20,487 prompt tokens**. These are reconstructed token counts, not saved generation-time token IDs, and have not been independently retokenized with the unavailable checkpoint tokenizer. The candidate generated approximately **83.4% fewer output tokens**, so its shorter runtime is not evidence of faster per-token inference. The baseline logged **16 truncations** and **13 execution/format errors** (categories may overlap), versus **one truncation** and **zero recorded execution errors** for the adapter. [E7, E8, E14, E15]

Paired statistical analysis

104	134	2	60
BOTH PASS	BASE ONLY	CANDIDATE ONLY	BOTH FAIL

Paired outcome matrix, indicating 134 cases lost and two gained

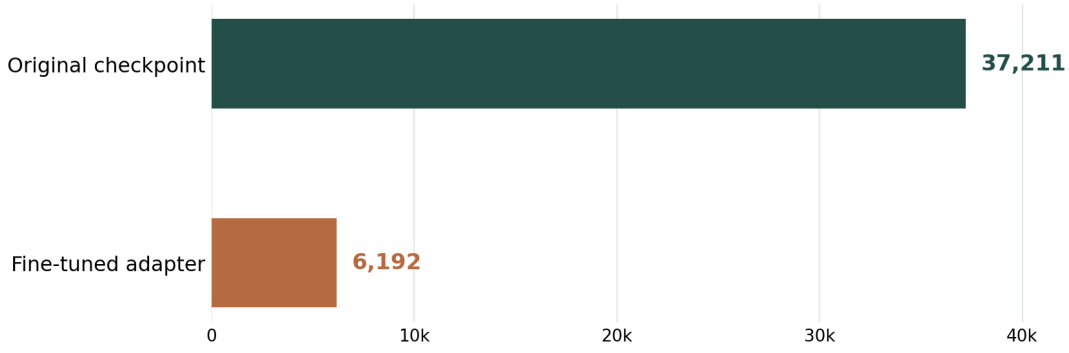
The predeclared analysis used **20,000 paired bootstrap resamples**, with seed `6172026`, producing a **95% confidence interval of −49.67 to −38.33 percentage points** for fine-tuned minus original pass rate. An **exact, two-sided McNemar test** on the discordant outcomes returned **$p = 2.14 \times 10^{-37}$**. [E1, E9]

These values support a strong **negative** comparison on the tested items. They do not prove the same effect would occur with another training seed, another evaluation population, a different inference budget, or a different fine-tuning recipe. The bootstrap describes **item-sampling uncertainty** under the analysed task set, not model-training variability.

Post-hoc examination of all 300 response pairs

The original and candidate confirmation JSONL logs were supplied after the first report. Both contain exactly **300 distinct and matching case IDs**. Their raw file SHA-256 digests match the corresponding execution summaries, and direct recounting of the `passed` fields reproduces **238** and **106** successes, as well as the **104 / 134 / 2 / 60** paired outcome matrix. A 375-row training-metric log also reproduces the recorded first-20 versus last-20 mean losses, with 3,000 cumulative examples seen. These checks validate the **internal supplied records**, not the original GPU runs or the underlying validator implementation. [E14, E16]

The most conspicuous post-hoc finding is a change in **response-generation behaviour**. The fine-tuned model's saved `raw_response` begins immediately with the reasoning-closing marker `</ifm|think>` in **all 300 cases**. None of the original-model raw responses begins that way; **287 of 300** baseline responses contain a reasoning-closing marker after some generated text, while the remainder include error/incomplete generations. The fine-tuned responses are considerably shorter by the later reconstructed token counts. This pattern is consistent with the adapter learning a different reasoning or answer-production style, but **does not establish the causal mechanism** of the regression. The training manifest documents an empty high-reasoning marker in the assistant target serialization; determining its effect would require a separate controlled experiment. [E1, E14, E15, E16]



Retrospectively reconstructed output-token totals, not generation-time IDs

The strict primary endpoint must not be reinterpreted retrospectively. However, error classification provides important context:

Diagnostic (post-hoc; not a frozen endpoint)	Original	Fine-tuned	What it does and does not mean
Arithmetic: terminal numeric token matches expected value	50/50	**28/50**	Candidate sometimes reaches the right number but violates the required answer-only format; 22/50 do not even end with the expected number. This is **not** the declared pass metric.
Ordered extraction: correct sequence after trimming comma whitespace	**50/50**	**14/50**	Shows 9 baseline exact failures were spacing-only; 36 candidate outputs still have ordering/content problems after this relaxation.
Multi-constraint JSON: correct `sum_odd` field	48/50	**0/50**	Field-level evidence for substantive candidate calculation errors, not merely JSON formatting.
Multi-constraint JSON: correct `count_even` field	46/50	**19/50**	Only part of the structured calculation remained correct.
Multi-constraint JSON: correct `maximum` field	**50/50**	**49/50**	**One field was largely preserved despite other regressions.**

These checks were created after observing the experiment and are presented solely as **diagnostics**, with no new hypothesis tests or substituted success counts. For example, on `confirm-arithmetic-027`, the original returned `1350`, while the candidate returned `(31 + 23) * 25 = 54 * 25 = 1350`. The candidate's arithmetic result was correct, but it failed the frozen instruction **"Return only the integer"**. On `confirm-multi_constraint-014`, both returned valid JSON, but the candidate supplied `sum_odd: 101` where the expected value was `85`: a substantive numerical mistake. [E14, E15, E16]

A separate example goes the other way: on `confirm-fact_extraction-030`, the original supplied the `day` field as a string (`"4"`), whereas the fine-tuned model returned numeric `4` and passed the deterministic validator. The adapter did retain some format/type improvements on individual cases; the **two** candidate-only successes should not be hidden. [E14, E15]

Interpretation. The intervention clearly degraded compliance with this frozen exact-output benchmark. Some apparent failures represent formatting rather than erroneous answers; other failures are demonstrably incorrect computations or orderings. A shift from long reasoning text toward immediate answer generation is observable, but cannot by itself distinguish training-induced reasoning

suppression, answer-style learning, scoring sensitivity, or other implementation effects. The held-out score remains **106/300**, not a retrospectively adjusted score.

5. What Osmium actually determined

The local experiment submitted an evidence bundle to the implemented **Viridian Osmium Audit v0.1** interface (`viridian.audit`, Viridian product version 0.2.0, adapter 0.1). The reported native determination was:

> OSMIUM AUDIT v0.1 — REPORTED STATUS
>
> **NOT_SUPPORTED**
>
> "Qualified evidence materially contradicts the bounded claim."

This is a **bounded direction-of-score and comparability result**, not an independent statistical or scientific replication. The supplied audit records describe Osmium as checking **intake schema**, **artifact presence / size / type / SHA-256 inventory**, **declared run identities**, and the **supplied point-score direction**. The tool did **not** examine raw predictions, calculate the paired interval or p-value, test contamination, independently qualify evaluator validity, reproduce training, or invoke the **Advanced V5** adapter (`V5_INTERFACE_UNAVAILABLE`). Those distinctions matter: the statistical analysis above was performed by **separate experiment scripts**, not by Osmium. [E10, E11]

A controlled adversarial check

A separate **synthetic fixture**, not part of the 300-case experiment, changed the candidate's declared `evaluator_version` while leaving the copied point scores available. Osmium reported `INSUFFICIENT_EVIDENCE`, explicitly identified the identity mismatch, and refused to treat the score comparison as qualified support. The injected fault was detected under the declared identity gate. [E12]

That is a useful demonstration of one narrow integrity control. It **does not** establish detection of semantically changed evaluators masquerading under identical declarations, hidden contamination, arbitrary file-content tampering, or evaluator correctness.

6. Interpretation: what failed, and what remains unknown

The improvement claim failed. On the frozen, predeclared confirmation distribution, this single adapter was markedly worse than the original checkpoint. The loss trend provides evidence that the optimiser updated a training objective; it supplies no counterargument to the independently scored negative confirmation result.

The cause is not yet established. The extreme regressions in arithmetic, multi-constraint following and ordered extraction warrant a forensic review of already-recorded predictions, completion lengths, reasoning-marker closure, final-answer parsing and the adapter-loading path. The provided summary files do not contain enough item-level outputs to attribute failure to catastrophic forgetting, response-

format changes, decoding, or a code defect. These are hypotheses for investigation, **not conclusions of this trial**.

The conclusion is narrower than "the 5B model got worse." This result is for a particular checkpoint and training intervention, under a high-reasoning chat template and a 256-token output ceiling, on six generated instruction-following categories. JSON construction stayed perfect, and structured fact extraction remained broadly stable. That heterogeneous pattern further cautions against claiming universal degradation.

7. Evidence quality and publication limitations

The evidence has strong **internal numerical consistency**. Recounting the actual uploaded confirmation JSONL records reproduces 238 and 106 passes and the 104/134/2/60 paired matrix; reported file digests match the uploaded bytes. The 375-step training metrics match the loss averages in the training manifest. **The raw per-item results and training metrics have now been inspected**, and the supplied confirmation-log hashes and totals match their recorded summaries. This is still **not independent re-execution**: the frozen validators, tokenizer, evaluator implementation, LoRA adapter binary and original checkpoint were not provided for running the experiment again.

Additional limitations include one training seed; a narrow generated test distribution; lexical rather than semantic contamination screening; the unresolved possibility of base-model pretraining exposure; and a generation budget that penalises unfinished reasoning. The dataset's stated Apache-2.0 licence and the model's local licence claim were not independently audited. The trial was not third-party certified or peer reviewed. [E3–E9]

Audit-record reconciliation required before publication. The supplied `OSMIUM_DETERMINATION.md` records a bundle SHA-256 of `f894bc32…3aef`, while `OSMIUM_EVIDENCE_MANIFEST.json` and `OSMIUM_INTERFACE_DISCOVERY.json` record `d573ede0…09b`; these two reports cannot both identify the same bytes. `OSMIUM_INTERFACE_DISCOVERY.json` also lists a case identity beginning `2d28e55c`, whereas the evidence manifest and determination list `586c0ed1`. The determination's evidence-manifest identity differs from the JSON manifest's identity. These appear to be documentation or run-identity discrepancies, but their causes are **not established** from the supplied artifacts. A reviewer should reconcile them against the native `audit.json`, immutable bundle, and creation history before publishing a definitive audit provenance assertion. This issue does not alter the aggregate score arithmetic, but it weakens the presently documented audit trace. [E10, E11]

For these reasons the document is marked a **publication draft**, not an independently replicated validation report. Post-hoc scoring diagnostics and reconstructed token counts are clearly separated from the predeclared primary metric.

8. Conclusion

Fine-tuning did not establish progress; it produced a substantial regression on the frozen task distribution. An assistant-only QLoRA run on 3,000 public instruction examples completed successfully and lowered its observed training loss, yet the candidate's 300-case exact-pass score fell from **79.33% to 35.33%**. The negative paired effect was pronounced (−44.0 pp; 95% paired bootstrap interval −49.67 to −38.33 pp). Retrospective output inspection also found a marked change in response length and reasoning-marker behaviour, alongside both strict-format violations and incorrect calculated results. An Osmium Audit v0.1 direction-of-score check returned `NOT_SUPPORTED` for the

improvement claim, and a separate declared evaluator-version mismatch triggered `INSUFFICIENT_EVIDENCE`.

The scientific lesson is not that fine-tuning is inherently harmful. It is that **successful optimisation, valid evidence, and defensible claims are separate things**. In this trial, the evidence justifies a clear failure of the specific progress hypothesis. It does not justify a diagnosis of the cause, a universal assertion about the underlying model, or a claim that Osmium has independently completed all relevant assurance checks.

Evidence register and reproducibility notes

The case study is grounded in the **uploaded records below**, all attributed to Trial 001. The full development/confirmation prediction JSONL files, their run summaries, and the training-metric log **were provided for this revision**. Model weights, checkpoint tokenizer, frozen scoring scripts, original generated case JSONL and native audit bundle remain outside the supplied review package. Some artifact hashes were validated against uploaded bytes; the wider provenance record still requires reconciliation.

Ref.	Source document	Purpose
E1	`EXPERIMENT_PROTOCOL.md`	Frozen hypotheses, metrics, scoring, seeds and task boundaries
E2	`FREEZE_MANIFEST.json`	Recorded pre-baseline hashes and subsequent amendments
E3	`MODEL_IDENTITIES.json`	Local checkpoint identity and observed architecture
E4	`DATASET_PROVENANCE.json`	Pinned dataset revision, filtering, sample and overlap
E5	`ENVIRONMENT_MANIFEST.json`	Hardware, library versions and resource measurements
E6	`TRAINING_MANIFEST.json`	QLoRA configuration, training completion, adapter hash
E7	`BASELINE_RESULTS.json`	Base scores, runtime, category counts and errors
E8	`FINETUNED_RESULTS.json`	Adapter scores, runtime, category counts and errors
E9	`STATISTICAL_ANALYSIS.json`	Paired outcomes, bootstrap and McNemar result
E10	`OSMIUM_DETERMINATION.md`	Human-readable native result and stated limitations

Ref.	Source document	Purpose
E11	`OSMIUM_EVIDENCE_MANIFEST.json`; `OSMIUM_INTERFACE_DISCOVERY.json`	Audit scope, identities, artifact inventory and run record
E12	`ADVERSARIAL_CHALLENGE_REPORT.md`	Synthetic evaluator-identity mismatch and observed response
E13	`FINAL_CASE_STUDY.md`; `PHASE_A_DISCOVERY_REPORT.md`	Original trial narrative and discovery decisions
E14	`baseline-confirmation.jsonl`; `candidate-confirmation.jsonl`; development JSONL and run summaries	Raw responses, final responses, IDs, validators and frozen pass flags
E15	`PROMPTS_AND_TOKEN_USAGE.md`	Retrospectively reconstructed token counts and selected examples
E16	**`training_metrics.jsonl`; `Viridian_K2_Trial_001_Forensic_Audit.json`**	**375-step training logs and explicitly post-hoc independent log recount**

Key local integrity identifiers: Base checkpoint content-set SHA-256 `136508ea5183ed2a33e4fc91ad5cb69e3686b2b2d3375cddd15542e0c1031853`; training subset SHA-256 `ead48b53ee98cb65ed6beb36dfb5018f40736e0bab20c0e30a6fbc9481e5c324`; adapter weights SHA-256 `ad09e8e71dc200b474eb5949b6cf6c54953c176ae1bd45a2a439377f8cf678a7`; confirmation case file SHA-256 `8fd979b17bc491c15ed8f21051a8f6e05768a5586d3a5fe6335defc46d66bb13`.

Source dataset cards: Pinned OpenAssistant OASST1 revision and OpenHermes 2.5 card, not used. These URLs identify reported sources; they are not a substitute for independent legal or technical validation.

Publication clearance checklist: Reconcile the divergent Osmium bundle, case and evidence-manifest identities; check the supplied raw per-item outputs against the **original frozen scoring implementation** and rerun where independently feasible; confirm applicable release rights for all shared examples and artifacts; and confirm that any public claims retain the scope and limitations stated above.

Viridian · Experimental assurance · Evidence before conclusions · viridianassurance.com