CertiPatch: Specification Repair for Frozen Language Models with Replayable Empirical Certificates

Ali Uyar

February 9, 2026

Abstract

We study *specification repair* for frozen language models: given a deterministic prompt family and a computable `Yes`/`No`labeler, learn an inference-time patch that enforces the specification *within a certified scope* while explicitly measuring collateral change. We present CERTIPATCH, a *train–certify–verify* protocol that (i) learns a gated low-rank hookpoint patch under a constraint-first objective (minimize collateral subject to zero in-scope violations), and (ii) emits a *replayable empirical certificate* with fail-closed verification semantics. In a reduced-compute, single-seed study on `EleutherAI/pythia-410m-deduped`, CERTIPATCHattains zero failures on three fully enumerable specs (COMPARE-2D: $0/10{,}000$, PARITY-4D: $0/10{,}000$, BALANCE-PAREN-14: $0/32{,}767$) and achieves 86.22% pass rate ($68{,}974/80{,}000$) on a large-domain, coverage-bounded specification (COMPARE-6D-STRAT). Where baseline cells are available (PARITY-4D), CERTIPATCHcloses the spec with substantially lower measured collateral ($\mathrm{KL}_S = 0.00168$) than completed budget-matched PEFT baselines. We further provide a six-condition compositionality study showing that naive additive patch composition fails (e.g., $5{,}003$ violations on spec B), while sequential and joint repair recover feasibility. All claims are explicitly scope-qualified: single seed, reduced compute, and incomplete baseline coverage where artifacts are missing.

1 Introduction

Large language models can still fail on deterministic micro-behaviors (e.g., basic comparison or parity) even when they remain useful on broader tasks. In this setting, post-hoc repair methods should satisfy three practical requirements: (i) repair the targeted behavior reliably, (ii) limit collateral drift on unrelated prompts, and (iii) provide auditable evidence for what was actually fixed.

We study this problem under a frozen-model assumption and focus on inference-time interventions. The resulting protocol, CertiPatch, combines constrained optimization with explicit artifact verification, so that reported outcomes are replayable from saved run artifacts. The citable release artifact for this manuscript is archived at 10.5281/zenodo.18541322. This places the method in the overlap between parameter-efficient adaptation methods such as LoRA and prompt tuning [1, 2], localized model-editing ideas [3, 4], and test-driven repair loops.

Contributions.

- **Train-certify-verify protocol.** We provide an end-to-end repair pipeline that emits certificate artifacts and enforces fail-closed verification.
- **Scope-aware certification.** We report exact certification on enumerable domains and coverage-bounded certification on a large stratified domain.
- **Compositionality evidence.** We evaluate six composition conditions and quantify interference/order effects.
- **Compute-aware claim discipline.** We restrict comparative claims to completed cells and keep the submission single-seed by design.

2 Problem setup

A *spec* defines a deterministic prompt family X_{spec} and a computable labeler $\mathrm{Spec}(x) \in \{\text{Yes}, \text{No}\}$. We evaluate the model via a fixed Yes/No decision rule on the answer-token logits. A *gate* $s(x) \in \{0, 1\}$ defines the certified scope: patches apply only when $s(x) = 1$, and collateral metrics are computed only on gate-firing suites. For non-enumerable

domains, we certify only a deterministic coverage plan and bounded search budgets; out-of-scope behavior is not certified.

3 Method

Patch family (GLR-HP). CERTIPATCHapplies an inference-time *gated low-rank hookpoint patch* to a fixed set of candidate transformer layers. Let $p(x)$ be the answer position (the last non-padding token) and $s(x) \in \{0, 1\}$ be the deterministic scope gate. At each candidate layer $\ell \in \mathcal{L}$, we patch the residual stream at the answer position:

$$h_{\ell,p(x)} \leftarrow h_{\ell,p(x)} + s(x)\, U_\ell\big(V_\ell^\top h_{\ell,p(x)}\big), \tag{1}$$

where $U_\ell, V_\ell \in \mathbb{R}^{d\times r}$ are trainable low-rank matrices (rank r is configured). All base model weights remain frozen.

Specs, margins, and constraint proxy. Each spec provides a deterministic domain X_{spec} and a computable labeler returning the correct answer token $t^*(x) \in \{\texttt{Yes}, \texttt{No}\}$. We score each prompt by a binary logit margin

$$m_\phi(x) = \text{logit}_\phi(t^*(x) \mid x) - \text{logit}_\phi(t^{\neg *}(x) \mid x), \tag{2}$$

where $t^{\neg *}(x)$ is the incorrect option. We require a configured margin threshold τ. For a finite evaluation set D, define violations $v_i = \max(0, \tau - m_\phi(x_i))$. To obtain a differentiable constraint proxy, we use a smooth max operator recorded in each run artifact:

$$g_{\text{smooth}}(\phi; D) = \frac{\log\Big(\frac{1}{|D|}\sum_{i=1}^{|D|}\exp(\beta v_i)\Big)}{\beta}, \tag{3}$$

with temperature β configured (`objective.beta_smooth`). ***Feasibility*** is evaluated in terms of discrete failures on the certified scope: zero failures on fully enumerable domains, or zero failures on the fixed coverage plan for coverage-bounded domains.

Constrained minimality objective. CERTIPATCHframes repair as *minimal collateral change under feasibility constraints*:

$$\min_\phi \ \mathcal{L}_{\text{col}}(\phi) + \mathcal{R}(\phi) \quad \text{s.t.} \quad \text{no in-scope violations on the certified evaluation set.} \tag{4}$$

The collateral term $\mathcal{L}_{\text{col}}$ is the RefBool-S KL divergence at the answer position on gate-firing prompts, and $\mathcal{R}$ is a configured regularizer (L2 and group sparsity across layers).

Augmented Lagrangian solver (inner loop). We solve Eq. (4) using an augmented Lagrangian method over g_{smooth}:

$$\mathcal{L}_{\text{AL}}(\phi; \lambda, \mu) = \mathcal{L}_{\text{col}}(\phi) + \mathcal{R}(\phi) + \lambda\, g_{\text{smooth}}(\phi) + \frac{\mu}{2}\, g_{\text{smooth}}(\phi)^2. \tag{5}$$

Within a run, the solver *warm-starts* the ALM state (λ, μ) across repeated inner solves (recorded in the run artifacts), updating $\lambda \leftarrow \lambda + \mu g_{\text{smooth}}$ and scaling μ up on violation or down when feasible. This update schedule is fully deterministic under fixed seeds and recorded in run artifacts for replay audits.

Counterexample-guided repair (outer loop). The outer loop performs counterexample-guided constraint generation (CEGIS-style). Starting from a deterministic initial sample $D^{(0)} \subset X_{\text{spec}}$, we repeat: (i) solve the inner ALM objective on the current active set $D^{(t)}$, (ii) search for counterexamples within the certified scope (exact sweep for enumerable domains; fixed coverage plan plus bounded additional search for coverage-bounded domains), and (iii) add the hardest counterexamples to form $D^{(t+1)}$. The loop terminates when no counterexamples are found within the certified scope and recorded budgets. The resulting active-set trace is included in artifacts, making termination and feasibility claims auditable.

Collateral metrics. In addition to the short-form KL on RefBool-S, we report a long-form drift metric RefBool-L (greedy decoding with a fixed budget), summarized by a divergence rate and first-difference index. These metrics are computed on gate-firing prompts by construction, aligning measurement with the certified intervention scope.

Spec	Total	Failures	Pass rate	Min margin	KL_S	Drift_L
COMPARE-2D	10000	0	1.000	1.00	9.694	1.000
PARITY-4D	10000	0	1.000	1.14	0.0017	0.125
BALANCE-PAREN-14	32767	0	1.000	1.00	7.219	1.000
COMPARE-6D-STRAT	80000	11026	0.862	-10.27	12.487	1.000

Table 1: Main CertiPatch outcomes (reduced scope, seed 0). The first three specs are fully enumerable; COMPARE-6D-STRAT is coverage-bounded.

4 Replayable empirical certificates

A certificate records what was evaluated and with which artifacts: model fingerprint, manifest hash, patch hash, certified scope definition (exact domain hash or coverage plan hash), counterexample budgets, and metrics. Certificates are scope-bounded and fail-closed: they must not claim satisfaction beyond their certified scope. A verifier recomputes hashes, regenerates datasets, re-evaluates metrics, and fails if any mismatch is detected.

Operational semantics. In practice, certificate replay validates:
- run identity consistency (‘run_record‘, ‘metrics‘, ‘certificate‘);
- artifact integrity (patch file and manifest hashes);
- scope integrity (domain generator hash or coverage-plan hash);
- metric consistency under deterministic re-evaluation.

Any mismatch is treated as verification failure (fail-closed), which prevents silent acceptance of stale or tampered artifacts.

5 Experiments

Scope and evidence discipline. All reported results come from the bundled run artifacts and are intentionally compute-aware: single seed (seed 0), a single main model (`EleutherAI/pythia-410m-deduped`), and a dev-model ablation block (`openai-community/gpt2`). Whenever a baseline cell is missing in the artifacts (e.g., multiple COMPARE-2D baselines), we explicitly treat it as *not run (compute scope)* and do not draw comparative conclusions from it.

Main outcomes across specs. Table 1 summarizes the four core specifications. CERTIPATCHcloses all fully enumerable domains (COMPARE-2D, PARITY-4D, BALANCE-PAREN-14) with zero failures. On the large-domain setting COMPARE-6D-STRAT, the certificate is *coverage-bounded*: we evaluate a fixed stratified plan of $80{,}000$ prompts and report a certified pass rate of 0.862 (Section 4). Collateral metrics are reported for the gate-firing reference suites and vary materially by spec; in particular, PARITY-4D has low measured $\mathrm{KL}_S$and moderate long-form drift, while COMPARE-2D and BALANCE-PAREN-14 exhibit higher measured drift under the same protocol.

Baseline comparisons where available (PARITY-4D). Table 2 compares CERTIPATCHto completed baselines for PARITY-4D. Several budget-matched PEFT baselines (LoRA, SoftPrompt, OneShot-FullDomain-MO) also close the spec in this run set (zero failures), but incur substantially larger measured collateral on the same gate-firing suites. For example, CERTIPATCHattains $\mathrm{KL}_S = 0.00168$ whereas LoRA reports $\mathrm{KL}_S = 0.01241$ and OneShot-FullDomain-MO reports $\mathrm{KL}_S = 0.11434$. Long-form drift is similarly heterogeneous: LoRA and SoftPrompt drift on essentially all RefBool-L prompts, whereas CERTIPATCHdrifts on a strict subset ($\mathrm{Drift}_L = 0.125$).

Baseline coverage gaps (COMPARE-2D). For COMPARE-2D, only the Base baseline cell is present in the artifacts; other baseline cells are missing because the required reference-suite artifact was not produced in this reduced bundle. We therefore restrict COMPARE-2D discussion to (i) main-run outcomes and traces (Table 1, Fig. 2) and (ii) qualitative protocol properties (certificate semantics and compositionality).

Method	Failures	KL_S (mean)	95% CI	Drift_L	FirstDiff	Params
CertiPatch	0/10000	0.00168	[0.00165,0.00171]	0.125	114.7	32768
LoRA	0/10000	0.01241	[0.01219,0.01262]	1.000	0.8	32768
SoftPrompt	0/10000	0.01881	[0.01868,0.01896]	0.999	3.0	32768
OneShot-FullDomain-MO	0/10000	0.11434	[0.11304,0.11568]	0.959	8.2	32768
OneShot-FullDomain-ALM	5000/10000	2.35966	[2.35868,2.36064]	1.000	0.0	32768
SteeringVec-1L	4868/10000	0.17692	[0.17676,0.17708]	0.022	125.2	1024
Base	5000/10000	0.00000	[0.00000,0.00000]	0.000	128.0	0

Table 2: PARITY-4D comparison on the main model (single seed). All methods use the same Yes/No gate-scoped collateral suites. KL_S is RefBool-S mean KL (base-to-patched) with bootstrap CI; Drift_L is RefBool-L divergence rate and FirstDiff is the mean first-differing token index (128 means no difference within the generation budget).

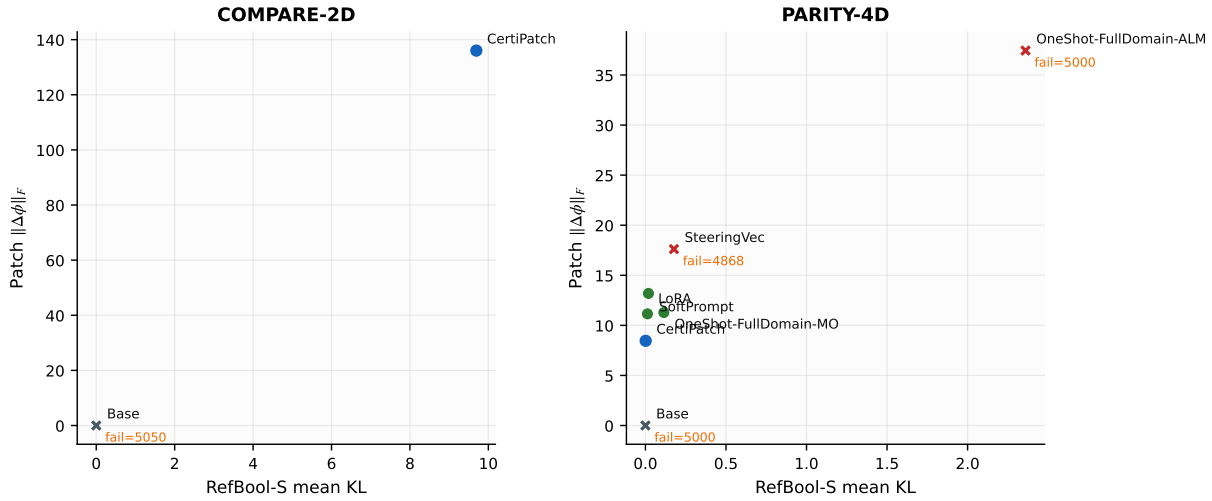

Figure 1: Collateral vs patch complexity for COMPARE-2D and PARITY-4D (single seed). Points are annotated by method; infeasible cells show failure counts. Missing cells are treated as not run (compute scope) and do not appear.

Coverage-bounded certification (COMPARE-6D-STRAT). Figure 3 reports failure rates across strata in the hashed coverage plan. The certificate shows strong performance on boundary-focused strata (e.g., `S_ext` and `S_eq`) and weaker performance in several interior strata, emphasizing why this run is framed as *coverage-bounded* rather than a full-domain claim. A full strata table is provided in Appendix Table 4.

Compositionality. We evaluate a six-condition compositionality suite using two specs: A (COMPARE-2D) and B (PARITY-4D). Figure 4 shows strong order effects. Naive additive composition (A+B) fails to transfer repair for spec B (5,003 violations), while sequential repair (A→B and B→A) and joint repair recover zero failures on both specs. These results indicate that patch interaction is non-commutative and must be measured rather than assumed.

Verifier binding checks. Figure 5 reports replay and tamper outcomes. The verifier is fail-closed: replay passes only when the recorded hashes and scope definitions match regenerated artifacts, and controlled mismatches (patch perturbation, generator hash mismatch, coverage-plan hash mismatch, model revision mismatch) are rejected.

Dev-model ablations (GPT-2). Table 3 reports targeted ablations on the dev model for COMPARE-2D. In this run set, constraining the patch to a single layer breaks feasibility closure, while several other ablations do not change the observed outcome. We therefore treat these ablations as *diagnostic* evidence for what is necessary in the small setting and avoid generalizing them beyond the reduced scope.

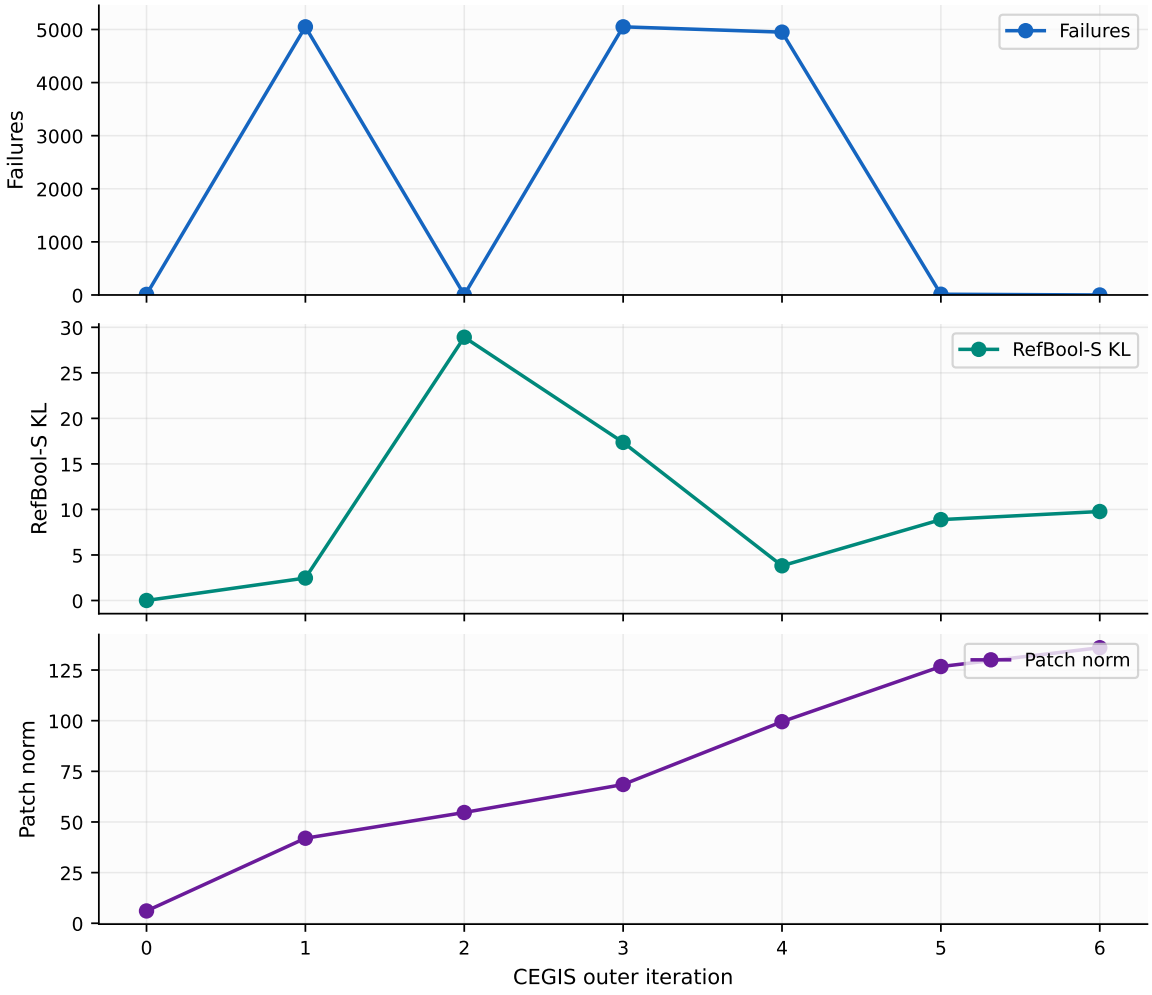

Figure 2: CEGIS dynamics on COMPARE-2D (single seed): spec failures, RefBool-S KL_S, and patch norm across outer iterations. This trace illustrates feasibility closure under counterexample-guided active-set growth; collateral behavior is reported, not assumed.

Variant	Failures	Pass rate	KL_S	Drift_L	$\|\Delta\phi\|_F$	Outer
CertiPatch	0	1.000	0.0011	0.527	7.64	7
no_minimality	0	1.000	0.0011	0.527	7.64	7
no_cegis	0	1.000	0.0011	0.527	7.64	1
no_collateral	0	1.000	0.0011	0.527	7.64	7
no_gating	0	1.000	0.0011	0.527	7.64	7
rank_1	0	1.000	0.0011	0.527	7.64	7
single_layer	4849	0.515	0.0007	0.075	7.64	7
random_counterexamples	0	1.000	0.0011	0.527	7.64	7

Table 3: Dev-model ablations on GPT-2 (COMPARE-2D). In this run set, the single-layer constraint is the only variant that fails to close the spec.

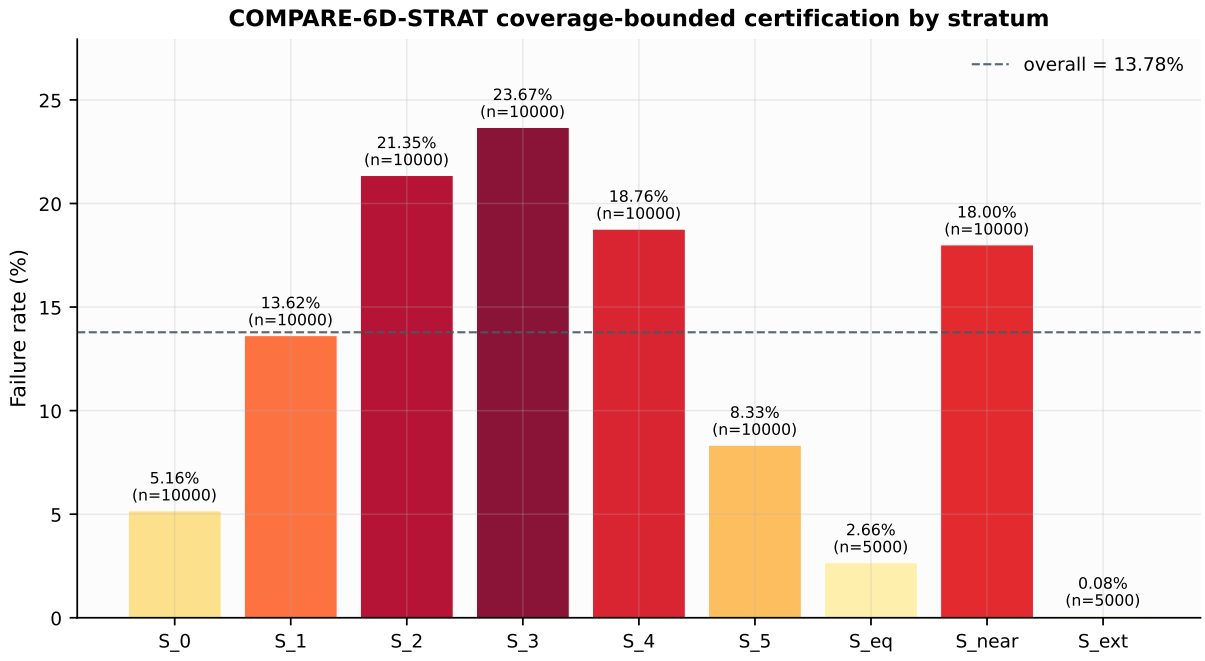

Figure 3: Coverage-bounded certification profile for COMPARE-6D-STRAT (single seed): certified failure rates per stratum (with stratum counts). The certificate claims only this coverage plan; out-of-coverage behavior is not certified.

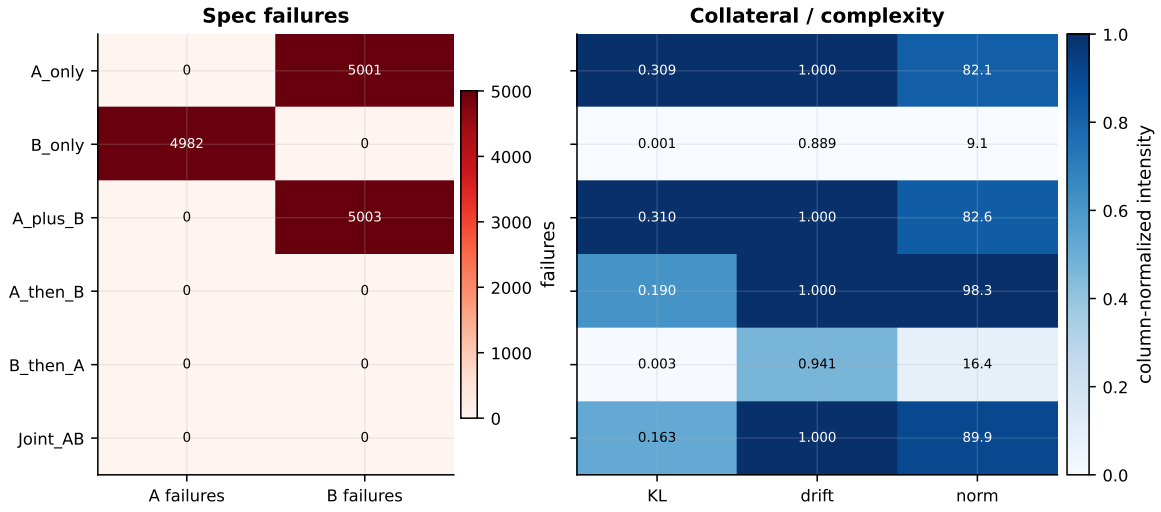

Figure 4: Compositionality matrix over six conditions (single seed): per-spec failures, collateral metrics, and patch norms. A+B denotes simultaneous application of the two independently learned patches; A→B and B→A denote sequential repair with feasibility constraints on both specs.

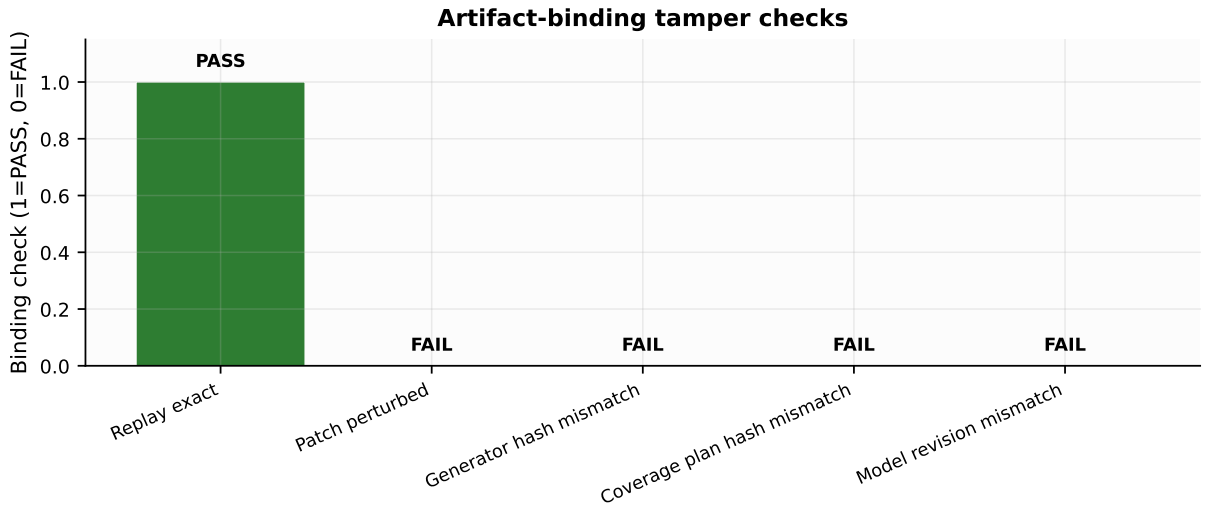

Figure 5: Artifact-binding tamper checks. Replay-exact evaluation passes; each controlled tamper condition fails, consistent with fail-closed verifier semantics.

6 Discussion

CertiPatch is designed for scope-bounded, replayable empirical guarantees rather than formal verification. The protocol can fail when the patch family cannot express a feasible repair under tight collateral constraints; such cases must be reported fail-closed. Scaling to larger models is primarily an engineering problem under the adapter contract and does not change the certificate semantics. This submission is single-seed and compute-aware; we therefore do not claim robustness across seeds. Baseline coverage is partial in some settings, so comparative claims are limited to completed matrix cells. For coverage-bounded domains, certificates are explicitly bounded by the recorded coverage plan and do not imply out-of-scope correctness.

Interpretation of collateral. Collateral is highly task-dependent in this run profile: PARITY-4D achieves near-zero KL and low drift, while COMPARE-2D and BALANCE-PAREN-14 show larger measured drift under the same protocol. This suggests that patch complexity alone is insufficient as a quality proxy; certification reports should always include both feasibility and collateral diagnostics.

Practical implication. The strongest claim supported here is operational: practitioners can replay, verify, and audit what was repaired and under which scope assumptions. This is a stronger safety posture than reporting isolated task accuracy without certificate semantics.

7 Related work

Model editing and localized updates. A large body of work studies targeted changes to a pretrained model's behavior, including factual association editing (e.g., ROME [3] and MEMIT [4]). CERTIPATCHshares the goal of localized behavior change, but differs in both objective and artifact semantics: we optimize a constraint-first *repair* objective under explicit collateral measurement and we emit replayable certificates that specify scope and coverage.

Parameter-efficient adaptation. Parameter-efficient fine-tuning (PEFT) methods—including prompt tuning [2] and low-rank adaptation (LoRA) [1]—motivate small-parameter interventions. In our reduced scope, we use these as budget-matched baselines where artifacts are available (PARITY-4D).

Counterexample-guided refinement and repair loops. Iterative refinement guided by discovered failures is a standard idea in adversarial testing (e.g., HotFlip [5]) and in counterexample-guided synthesis and repair (e.g., syntax-guided synthesis [6]). CERTIPATCHadopts a similar outer-loop structure but ties each run to a deterministic, auditable artifact bundle with fail-closed verification. Our contribution is not a new synthesis algorithm, but an artifact-verified adaptation of this loop pattern for frozen-model repair.

Attribution and debugging signals. Influence-style attribution methods aim to trace model behavior to training data or updates (e.g., influence functions [7], TracIn [8], and TRAK [9]). These methods are diagnostic rather than certifying; our focus is on replayable scope-bounded outcomes and their verifier semantics.

Collateral and behavioral evaluation. Collateral change is often discussed informally in editing and steering, but is rarely reported as a first-class, scope-aligned metric alongside an auditable certificate. We therefore report collateral as a primary metric family aligned with the same intervention scope used for feasibility claims.

8 Conclusion

We presented CertiPatch, a constrained minimality protocol for repairing programmatic specifications in frozen language models using gated hookpoint patches. CertiPatch produces replayable empirical certificates, supports coverage-bounded scopes for non-enumerable domains, and includes compositionality and tamper-verification evaluations. Within the reported compute-aware scope, the key result is not only repair feasibility but auditability: each reported claim is tied to deterministic run artifacts and fail-closed verifier checks. Future extensions include multi-seed robustness, broader

completed baseline coverage (especially on COMPARE-2D), and larger-model replications under the same certificate contract.

References

[1] 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: *arXiv preprint arXiv:2106.09685* (2022).

[2] Brian Lester, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *arXiv preprint arXiv:2104.08691* (2021).

[3] Kevin Meng, Arnab Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. "Locating and Editing Factual Associations in GPT". In: *arXiv preprint arXiv:2202.05262* (2022).

[4] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. "Mass-Editing Memory in a Transformer". In: *arXiv preprint arXiv:2210.07229* (2023).

[5] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. "HotFlip: White-Box Adversarial Examples for Text Classification". In: *Proceedings of ACL*. 2018.

[6] Rajeev Alur, Rastislav Bodik, Garvit Juniwal, Milo Martin, Mukund Raghothaman, Sanjit Seshia, Rishabh Singh, Armando Solar-Lezama, Emina Torlak, and Abhishek Udupa. "Syntax-Guided Synthesis". In: *Proceedings of FMCAD*. 2013.

[7] Pang Wei Koh and Percy Liang. "Understanding Black-box Predictions via Influence Functions". In: *Proceedings of the 34th International Conference on Machine Learning*. 2017.

[8] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. "Estimating Training Data Influence by Tracing Gradient Descent". In: *arXiv preprint arXiv:2002.08484* (2020).

[9] Soham Park, Yo Choe, Sungsoo Ahn, Saurav Das, Deepak Narayanan, Dimitris Papailiopoulos, Mark Chen, and J. Lee. "TRAK: Attributing Model Behavior at Scale". In: *arXiv preprint arXiv:2303.14186* (2023).

A Certificate schema (summary)

The repository includes a JSON Schema for `certificate.json`. The verifier validates the certificate, recomputes manifests and dataset hashes, and re-evaluates metrics to ensure replayability.

B Additional results

COMPARE-6D-STRAT strata breakdown. Table 4 reports certified pass rates by stratum for the fixed coverage plan used in COMPARE-6D-STRAT. These values are taken directly from the replayable certificate for the run.

Stratum	Total	Failures	Pass rate
S_0	10000	516	0.9484
S_1	10000	1362	0.8638
S_2	10000	2135	0.7865
S_3	10000	2367	0.7633
S_4	10000	1876	0.8124
S_5	10000	833	0.9167
S_eq	5000	133	0.9734
S_near	10000	1800	0.8200
S_ext	5000	4	0.9992

Table 4: COMPARE-6D-STRAT per-stratum certified performance (single seed). These strata are part of the fixed, hashed coverage plan recorded in the certificate; out-of-coverage behavior is not certified.

Notes on missing cells. This reduced-compute bundle does not include a completed baseline matrix for COMPARE-2D beyond the Base cell. All missing cells are treated as *not run (compute scope)* and are excluded from comparative claims.