Trust from afar: Evaluating remote model instances

Rishit Chatterjee [*1] **Alexandre Dang** [*2] **Marc Kaufmann** [3] **Mirco Giacobbe** [4 5] **Pascal Berrang** [4 5]

Abstract

Audits, model reporting, and safety evaluations of frontier models assume that the model a provider claims to serve is the model that answers user queries, but API access alone gives no way to verify this. We treat the challenge-response protocol as a practical hypothesis test for this problem and ask, on a GPT-2 benchmark, how many queries are needed to detect fine-tuned, compressed, adaptive, and backdoor variants of a trusted reference model under a given challenge distribution. Detectability is governed primarily by the challenge distribution rather than by the modification: a switching responder that defeats repeated-token challenges is exposed in a single mixed challenge, and a backdoor that passes every ordinary distribution is exposed at a single query by trigger-aware probes. Because no statistical test can rule out audit-time model switching on its own, we further propose a zkML-assisted variant in which a zero-knowledge proof binds each response to a committed model.

1. Introduction

Remote model deployment creates a simple but important verification problem: the model a provider claims to serve may not be the model that actually answers user queries. This matters for audits, model reporting, and safety evaluations, where trust depends on a specific vetted model rather than a related checkpoint (Raji et al., 2020; Mitchell et al., 2019; Shevlane et al., 2023).

Model equivalence has been studied in academia but is rarely used in practice due to various limitations. Model fingerprinting embeds into a model a unique distinctive signal that a user can retrieve through queries to question its identity. However, these fingerprints are not precise enough to guarantee model equivalence an are instead designed to verify model lineage (Xu et al., 2024). Another approach is zkML, where a model provider can provide zero-knowledge proofs that an inference was computed by a model which parameters match a public commitment. zkML provides strong equivalence guarantees but the current state of the art, such as NanoZK (Wang, 2026) or zkTorch (Chen et al., 2025), shows that the amount of computations required make it unusable on frontier models.

We study the challenge-response protocol as a solution to model equivalence. Given a trusted reference model M and a served model M', the verifier sends sampled inputs to both models and compares their next-token behavior. This framing asks a practical empirical question: how many challenge inputs reveal that a remote model has been fine-tuned, compressed, adapted, or backdoored? This approach sits between model fingerprinting and zkML. It is cheap enough to be deploy while retaining statistical guarantees of equivalence (formalized by Gao et al. (2025)). Several works improved the efficiency of the protocol with novel metrics to detect model divergence faster such as DiFR (Karvonen et al., 2025). However, most of them assume a passive model M' that is either quantized or pruned with no adversarial intentions. We go further and test the efficiency of various challenge distributions against both passive and adversarial M' that try to evade detection from the protocol. Moreover, we present an improvement to the challenge-response protocol by combining it with zkML and requiring additional proofs of responses generation to the challenges. This version prevents a whole category of attacks where the attacker changes the behavior of the remote model M' when detecting challenge queries.

2. Contributions

We make two major contributions in this paper.

Empirical evidence that challenge distribution controls remote model verification. We use a GPT-2 challenge-response benchmark in which a verifier compares next-token top-1 agreement between a served model M' and a trusted reference model M. Across fine-tuned, out-of-domain fine-tuned, magnitude-pruned, and quantized vari-

[1]Thayer School of Engineering, Dartmouth College, Hanover, NH, USA [2]Independent, Rennes, France [3]Independent, Zürich, Switzerland [4]School of Computer Science, University of Birmingham, UK [5]Zeroth Research, UK. Correspondence to: Alexandre Dang <adangresearch@duck.com>, Rishit Chatterjee <rishit.chatterjee.ug@dartmouth.edu>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

ants, the identical-model control is not falsely rejected, while modified models are detected under the Wikipedia and uniform distribution. More importantly, detectability depends strongly on the challenge distribution: pruning produces a clear query-budget spectrum, quantization is detected quickly, and repeated-token challenges are substantially weaker than Wikipedia-style or uniform challenges. Adaptive and backdoor responders expose the same limitation: a switching responder evades repeated-only verification, and a backdoor responder passes ordinary challenges but fails once trigger-aware probes are included.

A zkML-assisted protocol for binding challenge responses to a committed model. We show how challenge-response testing can be combined with zkML to separate two guarantees: zk-proofs cryptographically bind each response to a fixed committed model, while challenges statistically test whether that committed model behaves like the trusted reference on the tested distribution. This combination prevents response fabrication, audit-time model switching, and partial model substitution during the verified computation, while preserving the central limitation that behavioral equivalence is certified only relative to the challenge distribution.

3. Challenge-Response Framework

Our benchmark tests whether a served model M' behaves like a trusted reference model M under sampled challenge inputs. The verifier has full access to M but only output access to M' through a black-box interface. For each challenge batch, the verifier queries both models, compares next-token top-1 agreement, and rejects M' when the agreement falls below a fixed threshold $\tau = 0.995$. We repeat each trial across random seeds and report the smallest query budget

$$k_{\min} = \min\{k : \Pr[\text{reject at } k] \geq 0.95\}.$$

Thus, the benchmark measures distributional equivalence: passing the test means that M' matches M on the tested challenge distribution, not on every possible input. We also record KL divergence, total variation distance, and L_2 distance when logits are available, but these metrics are diagnostic rather than part of the accept/reject rule.

3.1. Model Variants and Responders

We evaluate fixed GPT-2 variants and adaptive responders. Table 1 summarizes the benchmark suite.

Fixed variants are chosen before verification and respond consistently across inputs. They include fine-tuned, out-of-domain fine-tuned, magnitude-pruned, and quantized GPT-2 models. These variants test whether ordinary model changes are detectable and whether smaller perturbations require larger query budgets.

Table 1. Model variants and responders evaluated.

Type	Variant	Purpose
Control	Identical copy	False-rejection check
Fixed	Mild fine-tune	Small behavior shift
Fixed	OOD fine-tune	Out-of-domain shift
Fixed	Magnitude-pruned	Pruning compression
Fixed	Quantized	Precision compression
Adaptive	Distribution-aware	Tuned to one challenge type
Adaptive	Query-limited	Limited-query imitation
Adaptive	Switching	Mimics M on recognized inputs
Backdoor	Backdoor	Deviates only on triggers

Table 2. Challenge distributions used in the benchmark.

Category	Distribution	Role
General	Wikipedia	Natural-language behavior
General	Uniform	High-entropy probing
General	Repeated	Recognizable structured probing
General	Mixed	Less predictable probing
Targeted	Trigger	Backdoor-specific probing
Targeted	Mixed + trigger	General + trigger probing

Adaptive responders test whether predictable challenge families can be exploited. Distribution-aware and query-limited responders approximate the reference on selected query families, while the switching responder returns reference-like behavior only on recognizable inputs. The backdoor responder matches the reference on ordinary challenges but diverges on trigger-containing inputs. Together, these responders test the main limitation of challenge-response verification: the test only certifies behavior on the challenge distribution.

3.2. Challenge Distributions

The challenge distribution is a verifier design choice: it determines which parts of model behavior are tested and how easy the test is for an adaptive responder to recognize. We therefore use distributions that differ along three axes: naturalness, entropy, and predictability. Wikipedia tests ordinary language behavior, uniform tests high-entropy synthetic behavior, and repeated-token tests a simple structured family that may be easy to exploit. Mixed challenges combine these families to reduce predictability. Trigger-based challenges are used only for the backdoor experiment, where the goal is to test behavior outside the ordinary challenge distribution.

4. Empirical Study

We evaluate GPT-2 model pairs across the challenge distributions in Section 3. For each model pair, distribution, and query budget k, we estimate detection probability over 8 random seeds. Experiments were run on an NVIDIA H200 NVL GPU.

4.1. How does compression affect query budget?

Compression variants are detectable, but their query budgets differ sharply, as shown in Figure 1. Pruning gives the clearest graded result: small pruning rates are difficult to detect, while larger pruning rates require only a few challenges. Quantization is detected almost immediately, so it works as a sanity check but is less useful for studying gradual changes in verification difficulty. Across compression settings, Wikipedia and uniform challenges are stronger than repeated-token challenges, especially when the modification is small. This supports the main empirical claim: verification strength depends on both the model change and the challenge distribution.

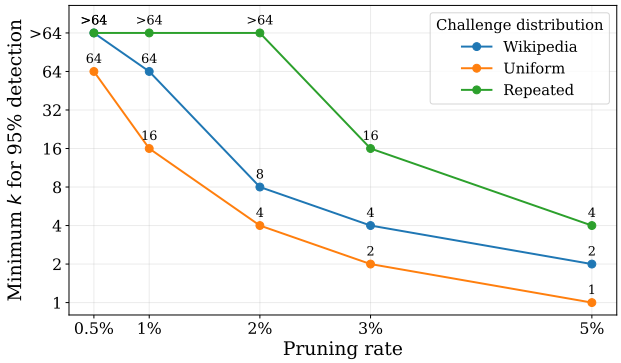

Figure 1. Pruning query budget. Each point shows the minimum number of challenges k needed for 95% detection; lower is easier, and > 64 means not detected within budget.

4.2. Can adaptive responders exploit predictable challenges?

Adaptive responders expose a sharper failure mode than fixed model variants. As Figure 2 shows, most responders are detected immediately on Wikipedia, uniform, and mixed challenges. Repeated-token challenges are weaker: they require more queries for distribution-aware and query-limited responders, and they completely fail against the switching responder within the tested budget.

The switching result is the key case. This responder mimics the reference model specifically on repeated-token inputs, so repeated-only verification cannot distinguish it. The same responder is detected at $k = 1$ on Wikipedia, uniform, and mixed challenges. This result gives a simple design rule: challenge-response verification should not rely on a single recognizable input family. In this experiment, mixed challenges avoid the repeated-only failure mode, although they are not uniquely better than Wikipedia or uniform challenges for every attacker.

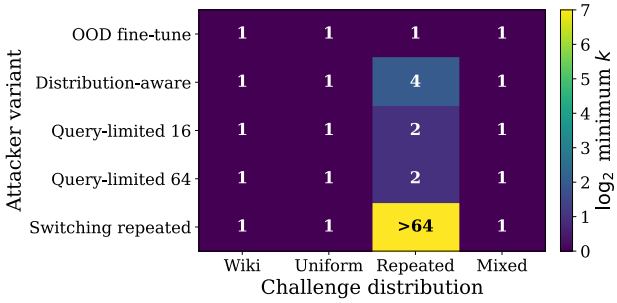

Figure 2. Adaptive responder query budget. Rows show responder type, columns show challenge distribution, and each cell reports the minimum k for 95% detection. Darker cells are easier to detect; > 64 means not detected within budget.

Table 3. Backdoor query budget: minimum k for 95% detection; > 64 means not detected within budget.

Challenge type	Distribution	Minimum k
Ordinary	Wikipedia	> 64
Ordinary	Uniform	> 64
Ordinary	Repeated	> 64
Ordinary	Mixed	> 64
Trigger-aware	Trigger	1
Trigger-aware	Mixed + trigger	1

4.3. Can ordinary challenges detect backdoors?

The backdoor responder exposes the distributional limit of challenge-response verification. As Table 3 shows, ordinary challenges do not detect the responder within the tested budget, even when Wikipedia, uniform, and repeated probes are mixed. This is expected: the responder is constructed to match the reference on ordinary inputs. Detection succeeds only when the verifier samples from the trigger region, where both trigger-only and mixed-trigger probes detect the backdoor at $k = 1$.

This result shows that ordinary challenge diversity is insufficient for rare-trigger behavior. Even the mixed ordinary distribution combines Wikipedia, uniform, and repeated probes without assigning probability mass to the responder's hidden trigger region. The backdoor experiment therefore does not show that mixed ordinary challenges are stronger; it shows the limit of any challenge distribution that misses the relevant behavior. Detecting such attacks requires targeted or adversarial probes that include the trigger region.

5. Challenge-Based Equivalence Testing with zkML

5.1. Component in Isolation

zkML alone. In zkML, a model provider commits a public commitment C of the model M' it provides. The provider

can convince a verifier that a given output y was produced by running M' on a given input x, without revealing the model's parameters by producing proof π. The user is then able to verify the outputs (x, y, π) against commitment C. If successful, then the provider has proven that y was computed using model M'. However, it does not prove that the committed model M' actually matches the reference model M. A malicious provider could commit to an arbitrary model, generate valid proofs for it, and the verifier would have no way to detect the substitution from the proof alone.

Challenge-based testing alone. The challenge protocol tests if M' behaves like M with increasing confidence as the number of challenges grows. However, these guarantees only hold on the challenge distribution and do not prevent any kind of model switching. The verifier has no assurance, that during the audit, the responses are actually produced by M' and that M' has not been replaced by the reference model M.

5.2. Challenges + zkML

On top of responses to the challenges, the provider supplies a zk-proof for each response, which the verifier checks against a commitment C to M' published beforehand. The verifier is then assured that the model exercised during the challenges is in fact M'. This combination closes the gaps in each component in isolation: the zk-proof binds every response to the committed weights, so responses cannot be fabricated, and the challenges enforce behavioural equivalence with M, so an arbitrary model cannot be committed. The only viable attack is to fine-tune M' to replicate M on the challenge distribution, which grows more costly as k increases.

5.3. Experimental Results

We use zkLLM (Sun et al., 2024) as our proof system, which supports the proof generation and verification of our transformer model Llama2. As the model tested is not GPT-2, the timing results in this section are representative of zkML cost rather than measuring the framework performance.

We run the experiments on an Intel Xeon Platinum 8336C with 240GB of memory and one GPU A100 SXM with 80GB of GPU memory. We ran 64 challenges on zkLLM to match the experiments. On average, proof generation and verification for each challenge took 588 seconds, and the total time for computation was around 10 hours. These challenges can be parallelized, so that given enough cores and memory, the best-case wall-clock cost is the time required for a single proof generation. These additional security guarantees have a cost. On average, the inference time for a vanilla version of LLama2 was 4s. This is a 92x increase in compute time for the challenge protocol combined with zk-proofs. These numbers show that even with a small model, the cost of using zkML is prohibitive.

6. Discussion and limitations

Practicality of the combined protocol. The zkML component adds a concrete computational cost. Proof generation and verification with zkLLM imposes a per-challenge overhead that dwarfs vanilla inference time, and this overhead is currently multiplicative in the number of challenges unless proofs are parallelized. Today's zkML systems are not yet practical for models at the scale used in production deployments. The challenge-only protocol remains immediately deployable; the zkML extension should be viewed as a roadmap for when proof systems mature rather than a ready-to-use primitive.

Benchmark scope and generalization. All experiments were conducted on GPT-2, a small publicly available model that enables rapid iteration across challenge distributions, model variants, and adaptive responders. The core finding, about how challenge distribution governs detectability more than model modification type, is expected to generalize, since it rests on the structure of the hypothesis test rather than on GPT-2 specifics. Scaling to frontier models remains an open engineering challenge, primarily on the zkML side rather than the statistical testing side.

Remaining attack surface. Even the combined protocol leaves one viable attack: a provider could fine-tune a substitute model to precisely replicate the reference model's behavior on the challenge distribution, then commit that model. This attack becomes exponentially more expensive as the challenge budget k grows and as the distribution is made less predictable. This motivates the use of mixed and trigger-aware challenges but cannot be ruled out entirely by any finite query protocol.

7. Conclusion

Challenge-response verification offers a practical approach to the model equivalence problem. On a GPT-2 benchmark, we showed that challenge distribution is the primary lever for detectability: diverse challenges expose a switching responder in a single query, and trigger-aware probes catch a backdoor invisible to ordinary distributions. The protocol requires no white-box access and is cheap to apply routinely. zkML closes the remaining gap of audit-time model switching by cryptographically binding each response to a committed model, while challenges certify that the committed model matches the trusted reference behaviorally. Current proof costs are prohibitive at frontier scale, but the statistical component is ready today, and the combined protocol provides a clear target as proof systems mature.

References

Chen, B.-J., Tang, L., and Kang, D. Zktorch: Compiling ml inference to zero-knowledge proofs via parallel proof accumulation, 2025. URL https://arxiv.org/abs/2507.07031.

Gao, I., Liang, P., and Guestrin, C. Model equality testing: Which model is this api serving?, 2025. URL https://arxiv.org/abs/2410.20247.

Karvonen, A., Reuter, D., Rinberg, R., Marks, L., Garriga-Alonso, A., and Warr, K. Difr: Inference verification despite nondeterminism, 2025. URL https://arxiv.org/abs/2511.20621.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 220–229. Association for Computing Machinery, 2019. doi: 10.1145/3287560.3287596.

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., and Barnes, P. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 33–44. Association for Computing Machinery, 2020. doi: 10.1145/3351095.3372873.

Shevlane, T., Farquhar, S., Garfinkel, B., Phuong, M., Whittlestone, J., Leung, J., Kokotajlo, D., Marchal, N., Anderljung, M., Kolt, N., Ho, L., Siddarth, D., Avin, S., Hawkins, W., Kim, B., Gabriel, I., Bolina, V., Clark, J., Bengio, Y., Christiano, P., and Dafoe, A. Model evaluation for extreme risks. *arXiv preprint arXiv:2305.15324*, 2023.

Sun, H., Li, J., and Zhang, H. zkllm: Zero knowledge proofs for large language models, 2024. URL https://arxiv.org/abs/2404.16109.

Wang, Z. G. Nanozk: Layerwise zero-knowledge proofs for verifiable large language model inference, 2026. URL https://arxiv.org/abs/2603.18046.

Xu, J., Wang, F., Ma, M. D., Koh, P. W., Xiao, C., and Chen, M. Instructional fingerprinting of large language models, 2024. URL https://arxiv.org/abs/2401.12255.

A. Formalization of challenge-response testing

This appendix formalizes the empirical object measured by our challenge-response benchmark. The main text reports query budgets, while in this appendix we state the corresponding detection probability and explain why the guarantee is distributional.

Setup. Let $\mathcal{X}$ be the set of token sequences of length L, and let $\mathcal{D}$ be a challenge distribution over $\mathcal{X}$. The verifier samples independent challenges

$$x_1, \ldots, x_k \sim \mathcal{D},$$

where k is the query budget. The verifier compares a trusted reference model M with a served model M'. For each prefix position t, the models induce next-token distributions

$$p_M(\cdot \mid x_{<t}) \qquad \text{and} \qquad p_{M'}(\cdot \mid x_{<t}).$$

Per-challenge agreement. For a single challenge x, we define token top-1 agreement as

$$a(x; M, M') = \frac{1}{L} \sum_{t=1}^{L} \mathbf{1} \left[\arg\max_y p_M(y \mid x_{<t}) = \arg\max_y p_{M'}(y \mid x_{<t}) \right].$$

Thus, $a(x; M, M') \in [0, 1]$ is the fraction of positions where the two models choose the same most likely next token.

Batch test. For a batch $X_k = \{x_1, \ldots, x_k\}$, the verifier computes the average agreement

$$A_k(M, M') = \frac{1}{k} \sum_{i=1}^{k} a(x_i; M, M').$$

Given an agreement threshold τ, the verifier rejects the claim that M' matches M when

$$A_k(M, M') < \tau.$$

In our experiments, $\tau = 0.995$.

Detection probability and query budget. For fixed M, M', and $\mathcal{D}$, define the detection probability

$$\pi_k(M, M'; \mathcal{D}) = \Pr_{X_k \sim \mathcal{D}^k} \left[A_k(M, M') < \tau \right].$$

The reported query budget is

$$k_{\min} = \min \left\{ k : \pi_k(M, M'; \mathcal{D}) \geq 0.95 \right\}.$$

If no tested value of k reaches this threshold, we report $k_{\min} > 64$. In the experiments, π_k is estimated by repeating each verification trial over random seeds.

Distributional nature of the guarantee. The test certifies behavior only relative to the challenge distribution. To see this, define the distinguishing set

$$B_{\mathcal{D}}(M, M') = \{x \in \mathcal{X} : a(x; M, M') < \tau\},$$

and let

$$\delta_{\mathcal{D}} = \Pr_{x \sim \mathcal{D}} \left[x \in B_{\mathcal{D}}(M, M') \right].$$

The quantity $\delta_{\mathcal{D}}$ is the probability that one random challenge from $\mathcal{D}$ exposes a difference between M and M' under the chosen statistic.

Proposition A.1 (Single-hit detection)**.** *Suppose the verifier rejects as soon as at least one of the k sampled challenges lies in $B_{\mathcal{D}}(M, M')$. Then*

$$\Pr[\text{detected after } k \text{ challenges}] = 1 - (1 - \delta_{\mathcal{D}})^k.$$

Proof. Each challenge is sampled independently from $\mathcal{D}$. A single challenge fails to distinguish the two models with probability $1 - \delta_{\mathcal{D}}$. Therefore, all k challenges fail to distinguish them with probability $(1 - \delta_{\mathcal{D}})^k$. Taking the complement gives the detection probability: $1 - (1 - \delta_{\mathcal{D}})^k$. □

Interpretation. Proposition A.1 is an idealized model of the batch-threshold rule used in the experiments, but it captures the central intuition: larger $\delta_{\mathcal{D}}$ implies a smaller query budget. The same pair (M, M') can therefore be easy to distinguish under one challenge distribution and hard to distinguish under another.

This view explains the adaptive and backdoor results. A switching responder can make $\delta_{\text{repeated}} \approx 0$ by matching the reference on repeated-token inputs, so repeated-only verification can fail. A backdoor responder can differ only on a trigger region $T \subset \mathcal{X}$. If an ordinary challenge distribution satisfies $\mathcal{D}(T) = 0$, then ordinary random challenges cannot detect the backdoor. Trigger-aware challenges make $\mathcal{D}(T) > 0$, which makes detection possible.