An Auditing Test to Detect Behavioral Shift in Language Models

Anonymous Authors[1]

Abstract

Ensuring language models (LMs) align with societal values has become paramount as LMs continue to achieve near-human performance across various tasks. In this work, we address the problem of a vendor deploying an unaligned model to consumers. For instance, unscrupulous vendors may wish to deploy unaligned models if they increase overall profit. Alternatively, an attacker may compromise a vendor and modify their model to produce unintended behavior. In these cases, an external auditing process can fail: if a vendor/attacker knows the model is being audited, they can swap in an aligned model during this evaluation and swap it out once the evaluation is complete. To address this, we propose a regulatory framework involving a continuous, online auditing process to ensure that deployed models remain aligned throughout their life cycle. We give theoretical guarantees that, with access to an aligned model, one can detect an unaligned model via this process solely from model generations, given enough samples. This allows a regulator to impersonate a consumer, preventing the vendor/attacker from surreptitiously swapping in an aligned model during evaluation. We hope that this work extends the discourse on AI alignment via regulatory practices and encourages additional solutions for consumer rights protection for LMs.

1. Introduction

Language models (LMs) can now achieve human-level performance in text summarization, machine translation, and many other tasks, including the bar exam (Achiam et al., 2023; Katz et al., 2024). Recently, a lot of effort has gone into ensuring their behavior is aligned with our societal values, spawning the field of *AI alignment* (Ji et al., 2023). So far, most of this effort has gone into developing ways to measure misalignment, for example, through evaluation benchmarks (e.g., Wang et al., 2023) and red-teaming (e.g., Perez et al., 2022).

One topic that has received less attention is building mechanisms to *continually monitor alignment during deployment* so that model consumers are protected. The gap between our ability to monitor LMs and our ability to train LMs has grown so large that many researchers have proposed to pause the development of such systems until this gap can be narrowed (Alaga & Schuett, 2023; Anthropic, 2023; pau, 2023). A promising solution that has been proposed is to involve third-party regulatory agencies to monitor LM alignment (Brundage et al., 2020; Anderljung et al., 2023). The benefit of this approach is that such agencies could consist of elected officials (or those appointed by elected officials), giving consumers a voice in the alignment process. This direction is also particularly timely, as regulations in both Europe (eu-, 2023) and the United States (Blumenthal & Hawley, 2023) have recently been proposed for AI systems.

Unfortunately, it can be difficult for a regulatory agency to verify that a model is aligned. Consider the following scenario: a vendor develops a language model M to deal with consumer complaints. They then fine-tune this model, producing a new model M^a that adheres to regulatory requirements for model alignment, and this is later verified by the regulatory agency. However, after this fine-tuning, vendor tests reveal that the model has become less helpful. This is worrisome as it may cause them to lose customers. Instead of altering the model further to save money, the vendor switches back to the original model M and deploys it. Every time they are audited by the regulator, the vendor produces samples from M^a, and once auditing is finished, they produce samples from M.

In this work, we propose a mechanism that, given access to an aligned model M^a, allows a regulator to impersonate a consumer and continually monitor model alignment solely from model generations (e.g. via API calls). This mechanism provably guarantees that if the generations come from a model with a different alignment behavior than M^a, it will catch it given sufficient generations, which depends on how different the alignment behavior is. The key insight behind our approach is that one can phrase the problem of

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Foundation Models in the Wild. Do not distribute.

alignment monitoring as hypothesis testing. This mechanism is inspired by recent work on anytime-valid hypothesis testing (Pandeva et al., 2024), a state-of-the-art sequential testing method that provides tight control over the alpha error and is consistent under weak assumptions.

2. Background

2.1. Measuring LM Behaviors

A popular way to operationalize LM alignment is via the process of removing undesired behaviors (Shalev-Shwartz et al., 2020; Hendrycks et al., 2021; Ngo et al., 2022; Wolf et al., 2023). This has spurred the creation of *behavior scoring functions* (Ji et al., 2023) that measure multiple properties of LM outputs such as helpfulness or politeness. In this work, let B be such a scoring function that assigns scores in the range $[0, 1]$ where 1 represents the full manifestation of the behavior and 0 indicates its absence. These scoring functions evaluate a generated string with respect to the desired behavior, quantitatively measuring alignment.

There are many approaches to detecting changes in LMs, which we summarize in Appendix A. In this work we opt for hypothesis testing.[1]

2.2. Hypothesis Testing

To effectively monitor behavioral changes *over* time, we require a sequential test that maintains control over the false positive rate (α-error). For this we turn to recent work on anytime-valid hypothesis testing.

Anytime-Valid Hypothesis Testing This approach utilizes the principle of ”testing by betting”, inspired by game theory (Shafer, 2021). The fundamental insight from this paradigm is that evidence against the null hypothesis can be represented as the gain in wealth of a bettor who wagers on specific observations. A bettor ”buys” a test statistic at the price of its expected value, and the *betting score* is defined as the ratio between the actual realization and its expectation, which determines the factor by which their wealth is multiplied. If the bettor chooses to reinvest in subsequent ”rounds” (i.e., when new data is observed and the test statistic is re-calculated), their betting scores are multiplied. Under the null, no betting strategy can consistently increase the bettor’s wealth (Ramdas et al., 2023).

Deep Anytime-Valid Hypothesis Testing (DAVT) (Pandeva et al., 2024) builds upon anytime-valid hypothesis testing and presents a general framework for designing powerful sequential non-parametric tests that leverage machine learning models. DAVT provides tight control over the α-error and is consistent, i.e., its power converges to 1 in the limit of infinite samples.

Given a stream of random observations $X_1, X_2, \ldots$ drawn *i.i.d.* from a distribution P_X over $\mathcal{X}$, two operators $\mathcal{T}_1, \mathcal{T}_2 : \mathcal{X} \to \mathcal{W}$ acting on that space and a fixed $\alpha > 0$, their framework allows the construction of a test distinguishing the transformed distributions $\mathcal{T}_1(X)$ and $\mathcal{T}_2(X)$, whose false positive rate is bounded by α. Their testing algorithm continuously updates the wealth upon observation of new data by multiplying with the new betting score S. The key feature of DAVT is a machine learning model ϕ used in the calculating of the betting score:

$$S = 1 + \phi(\mathcal{T}_1(x)) - \phi(\mathcal{T}_2(x))$$

The model is trained to maximize an objective derived from the betting score, thereby incentivizing it to learn differences between the two distributions.

Applying their framework to a variety of tasks, including two-sample testing and conditional independence testing under the model-X assumption, Pandeva et al. (2024) demonstrate DAVT’s competitive performance compared to other state-of-the-art non-parametric sequential tests, such as the two-sample tests E-C2ST (Lhéritier & Cazals, 2018) and Seq-IT (Podkopaev & Ramdas, 2024).

3. Methods

We design a powerful sequential two-sample test procedure for detecting behavioral shift in LMs. Let $\mathbf{x} \in \mathcal{X}$ denote a prompt from the (hypothetical) subspace of prompts that are likely to induce behavior $\mathcal{B}$. We conceptualize a language model as a stochastic operator M that maps a prompt $\mathbf{x}$ to a random continuation $\mathbf{y} \in \mathcal{Y}$, i.e. $M : \mathcal{X} \to \mathcal{Y}$, where M is defined such that $M(\mathbf{x}) \sim Q(\cdot \mid \mathbf{x})$.

The behavior scoring function B takes in both the prompt and the continuation to produce a score $B(\mathbf{x}, \mathbf{y}) \in [0, 1]$. Putting this back into Pandeva et al’s framework (2024), we would like to test:

$$\mathbf{H_0} : \ B(X, M^a(X)) \overset{d}{=} B(X, M(X))$$

$$\text{vs}$$

$$\mathbf{H_1} : \ B(X, M^a(X)) \overset{d}{\neq} B(X, M(X))$$

where M^a signifies a language model known to be aligned to a specific behavior.

The procedure for this *auditing test* is depicted in algorithm 1. It inherits the guarantees from DAVT, in particular an alpha error bounded by α under the $\mathbf{H_0}$ and consistency under the assumptions mentioned in Pandeva et al. (2024).

[1] Our proposed method can also be applied to detect *behavioral drift* (see Bayram et al. (2022) for a review).

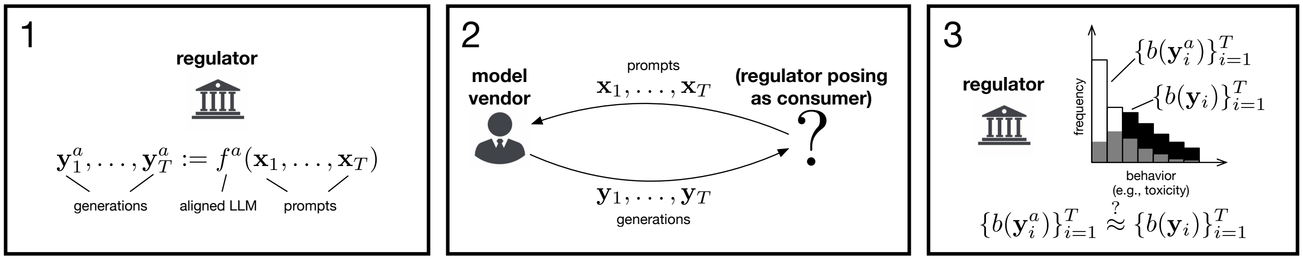
1
regulator
$\mathbf{y}_1^a, \ldots, \mathbf{y}_T^a := f^a(\mathbf{x}_1, \ldots, \mathbf{x}_T)$
generations
aligned LLM
prompts
2
model vendor
prompts
$\mathbf{x}_1, \ldots, \mathbf{x}_T$
(regulator posing as consumer)
?
$\mathbf{y}_1, \ldots, \mathbf{y}_T$
generations
3
regulator
$\{b(\mathbf{y}_i^a)\}_{i=1}^T$
$\{b(\mathbf{y}_i)\}_{i=1}^T$
frequency
behavior
(e.g., toxicity)
$\{b(\mathbf{y}_i^a)\}_{i=1}^T \overset{?}{\approx} \{b(\mathbf{y}_i)\}_{i=1}^T$

Figure 1. **Alignment monitoring.** Our proposal consists of three steps: 1. A regulator collects a set of prompts and generates outputs from an aligned LM; 2. The regulator poses as a consumer, sends the prompts to the model vendor, and collects the generations; 3. The regulator computes the distribution of behavior scores $b(\cdot)$ for both the aligned generations and the model vendor generations. It then compares these distributions. Our monitoring framework allows steps 1-3 to be repeated as many times as needed and is guaranteed to keep type I error (i.e., false positive rate) low.

3.1. Algorithm

The auditing test (algorithm 1) takes in a stream of prompts, a behavior function, a language model known to be aligned with the given behavior M^a and a second language model M, the α-level as well as the initialization of a regression model ϕ_0 and a maximum number of observation steps T to consider before the algorithm stops.

At every time step, a new prompt from the stream $\mathbf{x}_t$ is fed to both M and M^a to create continuations and then scored by the behavior scoring function. We feed these scores to the regression model ϕ_t and calculate the betting score. Next, we update the wealth by the betting score and check whether it surpasses the $1/\alpha$-threshold, in which case we reject the null hypothesis. If not, we update the regression model in a separate training step and continue with the next prompt. The algorithm can also be modified to accept batches of prompts instead of single prompts.[2]

Algorithm 1 Auditing Test

```
Input: ${\mathbf{x}_t}_{t \geq 1}$ (stream of prompts), $B$ (behavior scoring function), $M$ (LM API), $M^a$ (aligned model), $\alpha$ (type-I error limit under null), $\phi_0$ (regression model for testing), $T$ (maximum number of observations)
$W_0 \leftarrow 1$
for $t \leftarrow 1$ to $T$ do
    Compute behavior scores:
        $b_t \leftarrow B(\mathbf{x}_t, M(\mathbf{x}_t)), b_t^a \leftarrow B(\mathbf{x}_t, M^a(\mathbf{x}_t))$
    Compute wealth increase:
        $S_t \leftarrow (1 + \phi_{t-1}(b_t) - \phi_{t-1}(b_t^a))$
    Update wealth:
        $W_t \leftarrow W_{t-1} \times S_t$
    if $W_t \geq 1/\alpha$ then
        Stop and reject null
    end if
    Update regression model:
        $\phi_t \leftarrow \arg\max_\phi \sum_{l=1}^t \log(1 + \phi(B_t) - \phi(B_t^c))$
end for
```

4. Experiments

We validate our algorithm by examining the behavior of *toxicity* using the REALTOXICITYPROMPTS dataset (Gehman et al., 2020) and a RoBERTa-based hate speech detection model (Vidgen et al., 2020; Casper et al., 2023) to score prompts and LM generations. Llama3 (`8B-Instruct`) (Llama-team, 2024), Gemma (`1.1-7b-it`) (Mesnard et al., 2024), and Mistral (`7B-Instruct-v0.2`) (Jiang et al., 2023) serve as our aligned baselines. We remove the safety alignment in these models by fine-tuning, during which we produce 10 checkpoints each.

To evaluate the statistical properties of the auditing test, we assess its detection rate and false positive rate. We first run it on samples generated by the aligned baseline models and their fine-tuned checkpoints. These checkpoints, having undergone varying degrees of fine-tuning, are expected to diverge from the original model to different extents. Second, we examine the test's performance when given samples from the *same* distribution.

For experimental details on fine-tuning, text generation and the betting score network, please see the Appendix B.

5. Results

Detection Rate (Power) We test each fine-tuning checkpoint against the aligned baseline model using an $\alpha = 5\%$, repeated across random folds. We find that toxicity as measured by the hate-speech detection model does not monotonically increase with fine-tuning, so instead we measure distribution change between checkpoints and base model directly, using Wasserstein distance.

Figure 2 shows the detection rates for Llama3 and its check-

[2] In this case, the new betting score S_t is calculated as a product over samples in the batch.

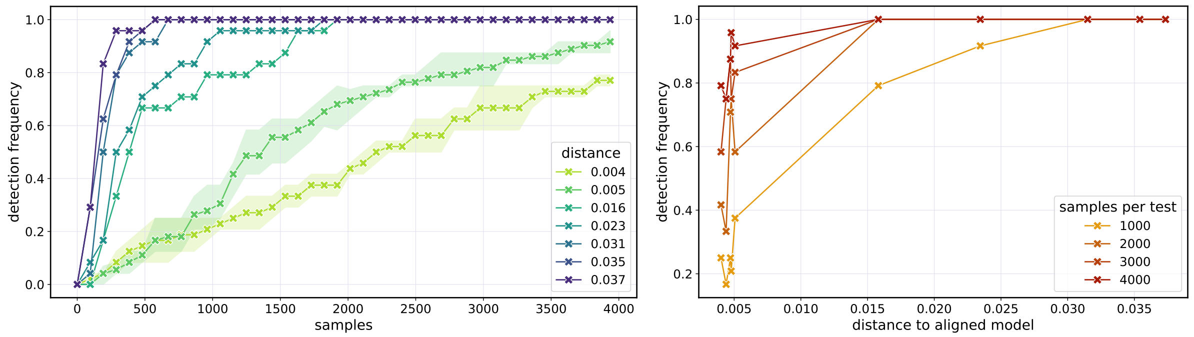

Figure 2. **Detection for Meta-Llama-3-8B-Instruct.** (*Left*) The detection frequency as a function of number of generated samples. Each curve is a corrupted model (to simplify visualization, the curves with shaded standard deviations are each averages over models with similar distances to the aligned model). The color depicts the Wasserstein distance from the corrupted model to the original aligned model. (*Right*) Detection frequency as a function of distance to the aligned model. Each point represents a corrupted model, lines are colored based on the number of observed generated samples.

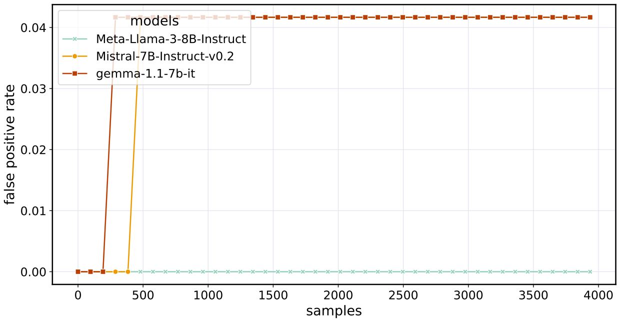

Figure 3. **False positives.** The false positive rate for each of the model architectures as a function of number of observed samples.

points. Distances between aligned models and checkpoints are generally small, which could be due to the skewness or suggest low conceptual overlap between the fine-tuning objective and the hate-speech detection model.The left plot shows the fraction of positive test results after having observed at least m samples, with tests repeated 24 times per checkpoint (4000 samples per fold, batch size 96). High detection rates of almost 80% are achieved even for checkpoints closest to the baseline. The plot highlights the test's sample efficiency: more altered distributions are detected faster. The right plot shows detection rate as a function of Wasserstein distance, varying fold size (maximum observations per test run). Short tests suffice for distant models, while more samples are needed for similar distributions. Similar results for Mistral and Gemma can be found in Appendix C.

False Positive Rate (α-error) We use different random seeds for generating text from the aligned baseline models to examine the false positive rate, shown in 3 over number of samples. The test is highly specific, with false detection rates consistently below 0.4. Wasserstein distances between the toxicity histograms produced by different seeds range between 2×10^{-4} for Llama3 and 10^{-4} for Mistral and Gemma, about one order of magnitude below the distances we find between the closest, but distinct checkpoints.

6. Limitations and Future Directions

We would like to investigate a few limitations of our method. First, while we have focused on the concept of *toxicity*, it is essential to explore the test's effectiveness in identifying other types of behavioral change. Categorizations of undesirable behavior (Ganguli et al., 2022; Wang et al., 2023) could serve as a starting point. Second, we would like to examine the extent to which changes in generation parameters, such as a higher temperature, affect behavior distributions. To avoid positive test results when using the same model, we might then want to modify the test to allow for *some* deviation in distribution.

Impact Statement

The rapid development and widespread adoption of large language models (LLMs) necessitate adherence to both current and emerging legislation to prevent and mitigate potential misuse and societal harm. This paper proposes a regulatory framework that incorporates continuous, online auditing to ensure that deployed models maintain alignment throughout their lifecycle. Our goal is to foster further research that advances technological breakthroughs while maximizing the positive societal impacts of LLMs.

References

Eu ai act: First regulation on artificial intelligence, 2023. URL https://tinyurl.com/2p99tmnr.

Pause giant ai experiments: An open letter, 2023. URL https://futureoflife.org/open-letter/pause-giant-ai-experiments/.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Agrawal, N., Bell, J., Gascón, A., and Kusner, M. J. Mpc-friendly commitments for publicly verifiable covert security. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2685–2704, 2021.

Alaga, J. and Schuett, J. Coordinated pausing: An evaluation-based coordination scheme for frontier ai developers. *arXiv preprint arXiv:2310.00374*, 2023.

Amrit, P. and Singh, A. K. Survey on watermarking methods in the artificial intelligence domain and beyond. *Computer Communications*, 188:52–65, 2022.

Anderljung, M., Barnhart, J., Leung, J., Korinek, A., O'Keefe, C., Whittlestone, J., Avin, S., Brundage, M., Bullock, J., Cass-Beggs, D., et al. Frontier ai regulation: Managing emerging risks to public safety. *arXiv preprint arXiv:2307.03718*, 2023.

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6481–6491, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.484. URL https://aclanthology.org/2022.findings-emnlp.484.

Anthropic. Anthropic's responsible scaling policy, 2023. URL https://www.anthropic.com/news/anthropics-responsible-scaling-policy.

Batu, T., Fischer, E., Fortnow, L., Kumar, R., Rubinfeld, R., and White, P. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 442–451. IEEE, 2001.

Bayram, F., Ahmed, B. S., and Kassler, A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245: 108632, 2022.

Blumenthal, R. and Hawley, J. Bipartisan framework for u.s. ai act, 2023. URL https://www.blumenthal.senate.gov/imo/media/doc/09072023bipartisanaiframework.pdf.

Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., et al. Toward trustworthy ai development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*, 2020.

Canonne, C. L. A survey on distribution testing: Your data is big. but is it blue? *Theory of Computing*, pp. 1–100, 2020.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Casper, S., Lin, J., Kwon, J., Culp, G., and Hadfield-Menell, D. Explore, establish, exploit: Red teaming language models from scratch. *arXiv preprint arXiv:2306.09442*, 2023.

Chan, S.-O., Diakonikolas, I., Valiant, P., and Valiant, G. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1193–1203. SIAM, 2014.

Diakonikolas, I., Gouleakis, T., Peebles, J., and Price, E. Optimal identity testing with high probability. *arXiv preprint arXiv:1708.02728*, 2017.

Diakonikolas, I., Kane, D. M., and Liu, S. Testing closeness of multivariate distributions via ramsey theory. *arXiv preprint arXiv:2311.13154*, 2023.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Ghodsi, Z., Gu, T., and Garg, S. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *Advances in Neural Information Processing Systems*, 30, 2017.

He, X., Xu, Q., Lyu, L., Wu, F., and Wang, C. Protecting intellectual property of language generation apis with lexical watermark. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10758–10766, Jun. 2022a. doi: 10.1609/aaai.v36i10.21321. URL https://ojs.aaai.org/index.php/AAAI/article/view/21321.

He, X., Xu, Q., Zeng, Y., Lyu, L., Wu, F., Li, J., and Jia, R. CATER: Intellectual property protection on text generation APIs via conditional watermarks. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=L7P3IvsoUXY.

He, Z., Zhang, T., and Lee, R. B. Verideep: Verifying integrity of deep neural networks through sensitive-sample fingerprinting. *arXiv preprint arXiv:1808.03277*, 2018.

Hendrycks, D., Carlini, N., Schulman, J., and Steinhardt, J. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Ippolito, D., Tramèr, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choquette-Choo, C. A., and Carlini, N. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

Ji, J., Qiu, T., Chen, B., Zhang, B., Lou, H., Wang, K., Duan, Y., He, Z., Zhou, J., Zhang, Z., et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Chen, B., Sun, R., Wang, Y., and Yang, Y. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pp. 10697–10707. PMLR, 2022.

Katz, D. M., Bommarito, M. J., Gao, S., and Arredondo, P. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254, 2024.

Kilbertus, N., Gascón, A., Kusner, M., Veale, M., Gummadi, K., and Weller, A. Blind justice: Fairness with encrypted sensitive attributes. In *International Conference on Machine Learning*, pp. 2630–2639. PMLR, 2018.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

Kuditipudi, R., Thickstun, J., Hashimoto, T., and Liang, P. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577.

Lhéritier, A. and Cazals, F. A sequential non-parametric multivariate two-sample test. *IEEE Transactions on Information Theory*, 64(5):3361–3370, 2018.

Llama-team. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/, 2024. Accessed: 2024-05-15.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Lycklama, H., Viand, A., Küchler, N., Knabenhans, C., and Hithnawi, A. Holding secrets accountable: Auditing privacy-preserving machine learning. *arXiv preprint arXiv:2402.15780*, 2024.

Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Ngo, R., Chan, L., and Mindermann, S. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*, 2022.

Pandeva, T., Forré, P., Ramdas, A., and Shekhar, S. Deep anytime-valid hypothesis testing. In *AISTATS*, volume 238 of *Proceedings of Machine Learning Research*, pp. 622–630. PMLR, 2024.

Perez, E., Huang, S., Song, H. F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. In

EMNLP, pp. 3419–3448. Association for Computational Linguistics, 2022.

Podkopaev, A. and Ramdas, A. Sequential predictive two-sample and independence testing. *Advances in Neural Information Processing Systems*, 36, 2024.

Ramdas, A., Grünwald, P., Vovk, V., and Shafer, G. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Segal, S., Adi, Y., Pinkas, B., Baum, C., Ganesh, C., and Keshet, J. Fairness in the eyes of the data: Certifying machine-learning models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 926–935, 2021.

Shafer, G. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(2):407–431, 2021.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. On the ethics of building ai in a responsible manner. *arXiv preprint arXiv:2004.04644*, 2020.

Sobel, B. L. Artificial intelligence's fair use crisis. *Colum. JL & Arts*, 41:45, 2017.

Vidgen, B., Thrush, T., Waseem, Z., and Kiela, D. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*, 2020.

Vyas, N., Kakade, S. M., and Barak, B. On provable copyright protection for generative models. In *International Conference on Machine Learning*, pp. 35277–35299. PMLR, 2023.

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S. T., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., and Li, B. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *NeurIPS*, 2023.

Wolf, Y., Wies, N., Levine, Y., and Shashua, A. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*, 2023.

Yoo, K., Ahn, W., Jang, J., and Kwak, N. Robust multi-bit natural language watermarking through invariant features. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2092–2115, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.117. URL https://aclanthology.org/2023.acl-long.117.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 657–672, 2018.

A. Related Work

Distribution Testing In distribution testing, the objective is to learn an unknown probability distribution (or a collection thereof) over a large domain, given the ability to sample from the distribution. A special case of distribution testing, known as *identity tests*, determines whether a distribution is identical to a reference distribution or ϵ-far from it (Batu et al., 2001; Chan et al., 2014; Diakonikolas et al., 2017; Canonne, 2020; Diakonikolas et al., 2023). Unlike classical hypothesis testing, the null hypothesis assumes a distribution distance greater than ϵ, while the alternative hypothesis assumes equality of distributions.

This framework offers two main advantages: it explicitly controls the error of falsely identifying different distributions as similar and provides optimal sample complexities, i.e., provably requiring the least amount of queries to determine a certain property with a fixed probability of error (Canonne, 2020; Batu et al., 2001). Most algorithms in this framework assume distributions are discrete (Batu et al., 2001), have a certain structure (Chan et al., 2014), or use a tailored distance metric (Diakonikolas et al., 2023). These assumptions, along with the need to set various hyperparameters and constants, makes them less straight-forward to implement in practice.

Model Certification In the realm of model regulation, a considerable focus lies on certifying machine learning models to ensure compliance. This entails scrutinizing whether a model adheres to specified criteria and verifying its consistency during inference. SafetyNets (Ghodsi et al., 2017) introduce an interactive proof protocol tailored for verifying predictions of deep neural networks, albeit with a focus solely on verification guarantees, leaving security assurances for future exploration. However, this protocol's applicability is restricted to models represented as arithmetic circuits. VerIDeep (He et al., 2018) present a methodology for generating inputs that elicit significantly divergent outputs with minor modifications to the machine learning model. Nonetheless, this approach lacks assurances regarding the model's overall stability.

Another avenue of research, emphasizing security guarantees, leverages cryptographic hash functions for certificate verification, particularly emphasizing fairness assessment (Kilbertus et al., 2018), with reliance solely on black-box access to the model. Similarly, (Segal et al., 2021) employ SHA3 for model certification and verification. However, these hash functions exhibit limitations in scalability when applied to large-scale machine learning models. To address this scalability challenge, (Agrawal et al., 2021; Lycklama et al., 2024) propose scalable verifiable commitments for *Secure multi-party computation (MPC)*, offering a means to authenticate the integrity of large-scale models during inference.

Watermarking Another set of approaches focuses on watermarking (Zhu et al., 2018; Amrit & Singh, 2022; He et al., 2022a;b; Kirchenbauer et al., 2023; Kuditipudi et al., 2023; Yoo et al., 2023), embedding signals into generated content that are imperceptible to humans but detectable algorithmically from the model's output. One framework is to utilize the concept of green and red tokens before generating a word, advocating for the use of green tokens during sampling for text generation models (Kirchenbauer et al., 2023; Kuditipudi et al., 2023). The watermark can be detected efficiently using an open-source algorithm without requiring access to the model. Additionally, one can embed a secret signature in the model output for image generation models, which can be extracted (Zhu et al., 2018; Fernandez et al., 2023). While these methods safeguard the interests of the model creator, they do not preclude them from switching to an unaligned model while embedding the watermark in the model output, potentially compromising the model's integrity or specific behavioral aspects.

Data Copyright Protection There is growing concern that conditional generative models might generate outputs highly similar to copyrighted materials in their training datasets, potentially leading to copyright infringement (Sobel, 2017; Vyas et al., 2023). To mitigate this risk, data deduplication has been suggested to reduce the likelihood of memorization (Lee et al., 2022; Kandpal et al., 2022; Carlini et al., 2022). However, deduplication alone does not ensure that models will not memorize individual examples (Ippolito et al., 2022). An effective alternative is differentially private training, recognized as the gold standard for preventing the memorization of individual training examples (Abadi et al., 2016; Anil et al., 2022). Furthermore, Vyas et al. (2023) introduced the concept of *near access-freeness*, a rigorous framework to ensure that generated content does not infringe upon copyrighted data. Likewise, this line of work focuses on protecting data copyright, whereas our work centers on certifying the model's integrity and reliability.

B. Experimental Details

We examine the behavior of *toxicity* using the REALTOXICITYPROMPTS dataset (Gehman et al., 2020), specifically the training set of 99,442 samples. Toxicity histograms are generated by feeding these prompts to selected models and scoring

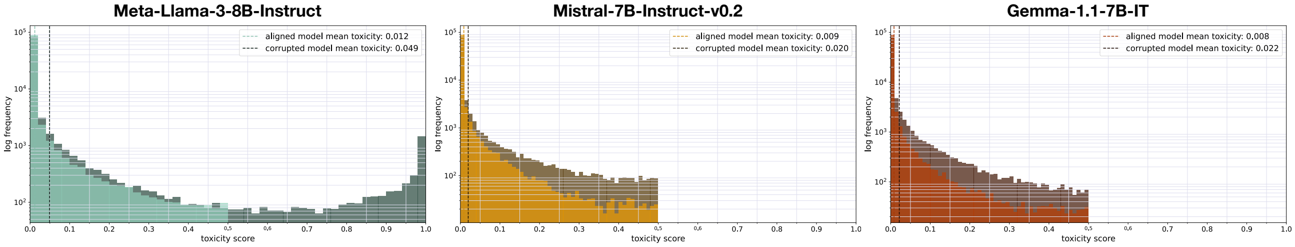

Figure 4. **Toxicity.** The toxicity distributions of the three model architectures we evaluate and their corrupted versions.

the prompt and generated texts' toxicity using a RoBERTa-based hate speech detection model (Vidgen et al., 2020; Casper et al., 2023), chosen for its practical speed over the PerspectiveAPI.

Experimental Setup We assess the efficacy of our approach using three models: Llama3 (`8B-Instruct`) (Llama-team, 2024), Gemma (`1.1-7b-it`) (Mesnard et al., 2024), and Mistral (`7B-Instruct-v0.2`) (Jiang et al., 2023). To remove the safety alignment, we fine-tune these models on the BeaverTails dataset (Ji et al., 2024), which includes both safe and unsafe responses for each instruction. We use a subset of 50K instances from the dataset, each comprising an instruction paired with its corresponding unsafe response. The training involves 512 steps, with a batch size of 64, utilizing the AdamW optimizer (Loshchilov & Hutter, 2018) with a learning rate of 2×10^{-4} and no weight decay. Due to computational constraints, we apply LoRA (Hu et al., 2021), with a rank of 16, to all models. All experiments were conducted on a single Nvidia A100 (80GB) GPU.

B.1. Toxicity Evaluations

We compare toxicity scores across Llama3, Gemma, and Mistral models. The original instruction-tuned versions, having undergone a safety alignment process to ensure helpfulness and harmlessness, serve as our aligned baselines. During fine-tuning, we produce 10 checkpoints for each baseline model.

Using the REALTOXICITYPROMPTS prompts, we generate continuations for each baseline model and their checkpoints. The sampling strategy and generation parameters are kept consistent throughout all experiments: a maximum of 100 new tokens, nucleus sampling with p=0.9, and a temperature of 0.7. We then evaluate the generated prompt and texts' toxicity using the hate-speech detection model developed by Vidgen et al. (2020).

B.2. Auditing Test

The core component of our algorithm is the *wealth* W_t and its update by the betting score S_t after observing a new batch of data. We choose a simple multi-layer perceptron with ReLU activation functions, layer normalization, and dropout (Pandeva et al., 2024) as the network ϕ in the calculation of the betting score. The network is updated using gradient ascent, with a learning rate of 0.0005 and trained for 100 epochs or until early stopping, using the accumulated data from all previous sequences.

C. Further Results

C.1. Toxicity Evaluations

Figure 4 shows the toxicity histograms for Llama3, Mistral, and Gemma baseline models compared to their fine-tuned checkpoints with the highest Wasserstein distance. The noticeable distribution shift illustrates the impact of fine-tuning. Toxicity histograms are highly positively skewed, with Pearson's skewness coefficients ranging from 6.2 to 7.7 for the instruction-tuned baselines. Corrupted checkpoints have slightly more toxic histograms and are slightly less skewed, with Pearson's coefficients in the range of 4.4 to 4.7, but still maintain low overall toxicity.

C.2. Auditing Test

Figures 5 & 6 show the results of Algorithm 1 applied to generations of Mistral-7B-Instruct-v0.2 and Gemma-1.1-7B-IT. Curiously we see that for the least corrupted Gemma model Algorithm 1 requires nearly the same number of samples for detection as the most corrupted model.

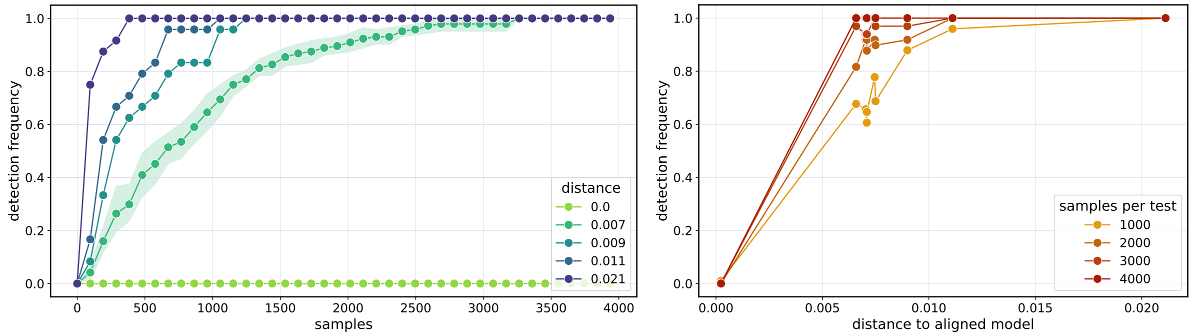

Figure 5. **Detection for Mistral-7B-Instruct-v0.2.**

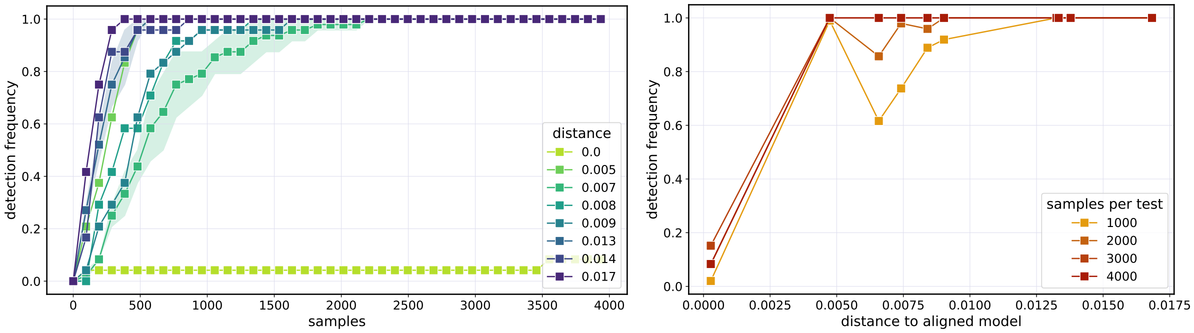

Figure 6. **Detection for Gemma-1.1-7B-IT.**