Data Forging Attacks on Cryptographic Model Certification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Privacy-preserving auditing of machine learning models has emerged as a key research direction with growing real-world importance. Despite rapid progress, the field still lacks a unifying security foundation for evaluating proposed solutions. In this work, we identify a fundamental gap between the security models underlying many audit protocols—focused on interactions between prover (model owner) and verifier(s) (auditors)—and the guarantees one would naturally expect. We show how this gap enables a broad class of attacks, called *data forging attacks*, even against protocols with formal cryptographic proofs of security.

Crucially, prior works are not technically incorrect; rather, their guarantees fail to generalize to other datasets, even though they are from the same distribution as the audit dataset. This generalization step is typically not captured in definitions of well-known cryptographic techniques such as zero-knowledge proofs.

We formalize this gap by introducing a general framework for modeling attacks on privacy-preserving audits. Using this framework, we demonstrate concrete data forging attacks across widely studied model classes. For example, a prover can falsely certify that a model is accurate (indeed, it will achieve over 80% accuracy on an audit dataset), while the model achieves only 30% in practice.

Our results highlight the need to revisit the foundations of privacy-preserving auditing frameworks. We hope that our work provides both cautionary evidence and constructive guidance for the design of secure ML auditing solutions.

1 Introduction

In recent years, machine learning has made its way from research labs into the fabric of everyday life, powering applications from customer service and product recommendations to credit scoring and fraud detection. With its widespread adoption, the problem of the *integrity* of machine learning models becomes critical: How can we ensure that the machine learning model truly possesses the properties it claims? For example, how can a company prove that a proprietary model meets the claimed performance guarantees and accuracy thresholds? How can we be sure that a credit scoring model behaves fairly toward minorities?

High-profile incidents—such as UnitedHealthcare's alleged use of AI to wrongfully deny insurance claims [HFS Research, 2024]—underscore the urgent need for mechanisms that allow external stakeholders to verify model behavior. Indeed, these issues have caught the attention of regulators around the world: Legal frameworks such as the EU AI Act and the Colorado AI Act aim to establish transparency, accountability, and fairness requirements for high-risk AI systems.

One current approach is for companies to partially disclose their models to external consultants offering audits for fairness, explainability, and regulatory compliance [Deloitte, ORCAA, BABL

Submitted to Workshop on Regulatable ML at the 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Table 1: Vulnerability to data-forging attacks in privacy-preserving ML audits. ✓= supported; ▲= conditional; ✗= not supported.

Work	Certified property				Resilience to data-forging	Continuous verification
	Acc.	Group Fair	Indv. Fair	Diff. Priv.		
Zhang et al. [2020]	✓	✗	✗	✗	▲ (pd)	✗
Shamsabadi et al. [2022]	✗	✓	✗	✗	✗	✗
Yadav et al. [2024]	✗	✗	✓	✗	✓	✓
Liu et al. [2021]	✓	✗	✗	✗	▲ (pd)	✗
Franzese et al. [2024]	✗	✓	✗	✗	✓	✓
Shamsabadi et al. [2024]	✗	✗	✗	✓	✗	✗
Kang et al. [2022]	✓	✗	✗	✗	✓	✗
Wang and Hoang [2023]	✓	✗	✗	✗	▲ (pd)	✗
Bourrée et al. [2025]	✗	✓	✗	✗	✗	✗

Acc. = accuracy; Group/Indv. Fair = group/individual fairness; Diff. Priv.=differential privacy. "Conditional" works lack detail to assess resilience to data-forging, but indicate deployments with public datasets (pd), which would be make the solution vulnerable. Continuous verification means audits must run continuously during deployment (e.g., via clients) rather than once pre-deployment.

AI, Mosaic Data Science]. While this can mitigate some concerns, it often conflicts with providers' need to protect proprietary models and sensitive training data. A promising emerging alternative is *certifiable* machine learning, which uses cryptographic techniques to formally prove desired properties while keeping data and model parameters confidential. Examples include certifying that a model was correctly trained [Abbaszadeh et al., 2024, Garg et al., 2023, Sun et al., 2024a, Pappas and Papadopoulos, 2024], that its training data has certain distributional properties [Chang et al., 2023, Duddu et al., 2024], that it can be audited by evaluating general functions of the model and data [Lycklama et al., 2024, Waiwitlikhit et al., 2024], and that its outputs are explainable [Yadav et al., 2025], privacy-preserving Shamsabadi et al. [2024], fair [Shamsabadi et al., 2022, Yadav et al., 2024, Franzese et al., 2024, Zhang et al., 2025], or correctly computed [Weng et al., 2021, Sun et al., 2024b, Xie et al., 2025]. These methods aim to provide accountability in ML services without requiring white-box access to the service provider's ML pipeline.

However, model certifications are often *data-dependent*, and certifiable ML works typically rely on the assumption that the training data is trusted. At first glance, this seems reasonable: In many real-world deployments, training data is private due to commercial or legal reasons. The verification process focuses then solely on the model and its certificate, at best publishing an "encrypted" version of the data and proving the properties of the model with respect to the hidden dataset.

In this work, we show that this trust assumption can be easily exploited: An adversarial model holder can engineer the "training data" so that the model *honestly* trained on this data passes the audit — indeed, the training was done correctly! – but exhibits pathological behavior on real-world data. In particular, we show that this is the case if the auditor relies a publicly known dataset (e.g., any of the well-known datasets that are used to check fairness) for test purposes. We make **three main contributions in our work**.

We introduce a **theoretical framework** for attacks on privacy-preserving machine learning auditing. Our framework is deliberately general, and supports a wide range of audit functions and adversarial goals. In particular, it captures several proposals for privacy-preserving machine learning audits, even those based on *zero-knowledge proofs*, a classical cryptographic technique with formal security guarantees.

The framework in particular enables us to assess whether a given auditing solution is vulnerable to a specific class of attacks we define. We investigate known auditing approaches for vulnerabilities to this type of attacks and summarize our findings in Table 1.

Finally, we propose concrete **data forging attacks** in the context of decision trees, a widely deployed model class. The result of our attacks are models that are certifiably "correct" on paper, but fail to satisfy the intended properties in practice. It is important to note that our attacks allow for much more than marginal deviations from the audit's guarantees. Rather, they enable dramatic violations: for example, if an auditor wishes to check that a model achieves accuracy of over 80%, the prover will pass the test on a given audit dataset, yet the same model may achieve only 30% accuracy on real-world inputs. Our attacks rely on adversarially crafting/adjusting the training dataset. While existing cryptographic auditing protocols alone cannot detect such manipulations, one could imagine that statistical tests layered on top of the audit solutions could solve the problem. However, curiously, we show that our attacks remain undetected even if such additional tests, e.g., Welch's t-test [Welch, 1947], are done on the training data. We establish the effectiveness of our attacks through experiments.

In summary, our work advances the study of cryptographic auditing for machine learning by **(i)** introducing a framework for modeling attacks which allow the adversary to pass the certification procedures, but exhibit pathological behavior on real-world data, **(ii)** demonstrating that known auditing solutions are often vulnerable to such attacks, and **(iii)** giving concrete attack examples even against auditing schemes with formal security guarantees. We emphasize that we *do not suggest that prior cryptographic works are broken on a technical level*, rather that the assumption on which these works rely deserves closer scrutiny. Additionally, as a result of our findings, we note that our work provides strong evidence that secure audit solutions with any of the following properties are unlikely: a) those which utilize known public datasets for test purposes, b) those that reuse test datasets (at least if model owner learns a substantial amount of this test dataset during the audit), c) those that are simultaneously non-interactive and data-dependent. We hope that our work will serve as a guidance when designing cryptographically secure machine learning audit frameworks.

Finally, we note that our work is related to, but distinct from, data poisoning attacks. We discuss the relationship between the two lines of work in Section A.2.

2 Related Work

A number of recent works aim to prove desirable model properties. In terms of *what* these works prove, they can be roughly categorized into proofs of training, inference, accuracy, and fairness. In terms of *how* the corresponding protocols work, the works can be split into the following categories:
Cryptographic approaches A prolific line of research adapts various cryptographic techniques to obtain formal proofs of training, accuracy, fairness, and inference in a privacy-preserving manner. The most common technique is *zero-knowledge proofs*, which allow to formally prove that a model satisfies certain properties without revealing anything else about the model. Such proofs have been used to obtain privacy-preserving certifications of fairness [Shamsabadi et al., 2022, Yadav et al., 2024, Franzese et al., 2024, Zhang et al., 2025], inference [Zhang et al., 2020], and accuracy [Zhang et al., 2020]. Finally, numerous works utilized zero-knowledge proofs to obtain guarantees for correct model training on private data [Abbaszadeh et al., 2024, Garg et al., 2023, Sun et al., 2024a, Pappas and Papadopoulos, 2024]. Other works [Duddu et al., 2024, Chang et al., 2023] rely on *secure multi-party computation* (MPC), which allows mutually distrusting parties to securely perform a computation on the respective private inputs without revealing anything about the inputs apart from the outcome. Chang et al. [2023] use a combination of zero-knowledge and MPC to perform distribution testing over a dataset supplied by multiple parties.

Black box auditing/Statistical testing Informally, black box auditing works by having users submit inputs to the model, query it, and analyze the resulting outputs. Tramer et al. [2017], Saleiro et al. [2018] provide black-box testing frameworks to check for potential unfairness or bias. Tan et al. [2018] proposes a black-box audit approach by distilling a new model and using it to gain insight into the black-box model.

Outside-the-box auditing In this type of auditing, the model owner grants users access to additional information about the system's development and deployment. This can take many forms, including source code, documentation [Mitchell et al., 2019], hyperparameters, training data, deployment specifics, and results from internal evaluations.

3 Notation & Preliminaries

We now introduce our notation as well as background for relevant models and techniques.

ML Notation We use $\mathcal{X}$ to denote the feature space, $\mathcal{D}$ to denote the distribution over $\mathcal{X}$, and $\mathcal{Y}$ to denote the label space. We denote by Train a training algorithm (or *learner*) that takes as input a training dataset $S = \{(x_i, y_i)\}_{i\in[n]}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, and outputs a model (or *hypothesis*) $h : \mathcal{X} \to \mathcal{Y}$.

Decision Trees In our attack constructions we will focus on decision tree models. Decision tree-based solutions are among the most popular machine learning algorithms, particularly known for their effectiveness in classification problems such as loan approval and fraud detection. A decision tree is trained by recursively partitioning the dataset from the root to the leaves. At each step, a split is determined by a splitting rule that aims to maximize an objective function, such as information gain. For prediction, the input follows a path from the root to a leaf, where at each internal node, the decision depends on whether the input satisfies the corresponding threshold (see Algorithm 3).

Welch's t-test The goal of t-test is to determine whether the unknown population means of two groups are equal or not. That is, for random variables X and Y, it compares the following hypotheses on their means $\mu_X = \mathrm{E}[X]$ and $\mu_Y = \mathrm{E}[Y]$:

- Null Hypothesis H_0: $\mu_X = \mu_Y$
- Alternative Hypothesis H_1: $\mu_X \neq \mu_Y$

Assuming that X and Y independently follow Gaussian distributions with unknown variances, Welch's t-test proceeds as in Algorithm 2.

3.1 Zero-Knowledge Proofs

Before defining zero-knowledge proofs, we first introduce an extended notion of NP relations.

Definition 1 (Indexed Relation)**.** *An indexed relation $\mathcal{R}$ is a polynomial-time algorithm with binary output, which takes a triple* $(\mathsf{i}, \mathsf{x}, \mathsf{w})$ *as input, where* i *is the index,* x *is the instance, and* w *is the witness. Typically,* i *describes an arithmetic/boolean circuit,* x *denotes public inputs, and* w *denotes private inputs, respectively.*

Definition 2 (Proof System)**.** *An (interactive)* proof system Π *for indexed relation $\mathcal{R}$ consists of a tuple of interactive Turing machines $(\mathcal{P}, \mathcal{V})$, where $\mathcal{P}$ is prover and $\mathcal{V}$ is verifier, respectively. Let $b \leftarrow \langle \mathcal{P}(\mathsf{w}), \mathcal{V}\rangle(\mathsf{i}, \mathsf{x})$ denote the interaction between $\mathcal{P}$ and $\mathcal{V}$, where both $\mathcal{P}$ and $\mathcal{V}$ take* (i, w) *as common inputs, and $\mathcal{P}$ additionally takes* w *as a private input. At the end of interaction, $\mathcal{V}$ halts by outputting a binary b.*

Proof systems that are used in the context of ML auditing typically require the following standard security properties: For an indexed NP relation $\mathcal{R}$, the proof system must provide *completeness* (i.e., if prover and verifier follow the protocol, verifier always accepts), *(knowledge) soundness* (i.e., if verifier accepts the proof generated by a cheating prover A, then it must be that A owns a valid witness w satisfying given NP relation w.r.t. statement x and index i), and *zero knowledge* (i.e., the transcript of the interaction between the prover and the (malicious) verifier leaks nothing except that there exists a witness w such that $(\mathsf{i}, \mathsf{x}, \mathsf{w}) \in \mathcal{R}$). See §A.4 for formal definitions.

4 Cryptographic Auditing of ML: Background and Subtleties

As noted in §2, a variety of privacy-preserving auditing methods for machine learning have been proposed, including cryptographic, differentially-private, and statistical techniques.

Numerous works rely specifically on zero-knowledge techniques, which allow to formally prove that a model satisfies a desired property (e.g., accuracy, fairness, or inference correctness) on a given test dataset without learning anything else about the model or its training data. We now outline different categories of proofs that are used in the context of auditing machine learning algorithms. For simplicity, from now on we assume that the *training algorithm is public* (note that making it private only makes the adversary in our attacks stronger, i.e., it could potentially be *easier* for the model owner to perform a data-forging or any other type of attack).

Proof of Training A *proof of training* can be viewed as a zero knowledge proof for the following relation $\mathcal{R}$: given $\mathsf{i} = (\mathsf{Train}, \mathsf{Commit})$, $\mathsf{x} = (\mathsf{com}_h, \mathsf{com}_S)$, and $\mathsf{w} = (h, S_{\mathit{train}}, \rho_h, \rho_S)$, $\mathcal{R}$ outputs 1 if and only if $\mathsf{Train}(S_{\mathit{train}}) = h$, $\mathsf{com}_h = \mathsf{Commit}(h; \rho_h)$ and $\mathsf{com}_S = \mathsf{Commit}(S_{\mathit{train}}; \rho_S)$. Here, Commit is a standard cryptographic *commitment scheme*: it produces a string com that "locks in" a value (e.g., the model h or dataset S) using some randomness ρ. A commitment is hiding (it reveals nothing about the underlying value) and binding (once published, it can only be opened to that value). Intuitively, commitments let the prover fix h and S_{train} up front without revealing them.

Proof of Inference A *proof of inference* can be viewed as a special case of zero knowledge proof for the following relation $\mathcal{R}$: given $\mathsf{i} = \mathsf{Commit}$, $\mathsf{x} = (\mathsf{com}, x, y)$, and $\mathsf{w} = (h, \rho)$, $\mathcal{R}$ outputs 1 if and only if $h(x) = y$ and $\mathsf{com} = \mathsf{Commit}(h; \rho)$.

Auditing using Zero Knowledge Proofs The strongest form of ZK-based auditing arises when the prover first produces a *proof of training*, thereby showing that a specific committed model instance came from an honest training procedure on a private dataset, and subsequently provides a *proof of property* attesting that the committed model meets the desired criterion. Let F be a auditing function outputting a binary that takes as input a training data set S_{train}, an auditing data set S_{audit}, and the model's predictions on the audit dataset $\{h(r)\}_{r \in S_{audit}}$. Then privacy-preserving auditing can be realized using zero knowledge proofs for the following relation $\mathcal{R}$: given $\mathsf{i} = (\mathsf{Train}, \mathsf{Commit}, F)$, $\mathsf{x} = (\mathsf{com}_h, \mathsf{com}_S, S_{audit})$, and $\mathsf{w} = (h, S_{train}, \rho_h, \rho_S)$, $\mathcal{R}$ outputs 1 if and only if $\mathsf{Train}(S_{train}) = h$, $F(S_{train}, S_{audit}, \{h(r)\}_{r \in S_{audit}}) = 1$, $\mathsf{com}_h = \mathsf{Commit}(h; \rho_h)$ and $\mathsf{com}_S = \mathsf{Commit}(S_{train}; \rho_S)$.

Definition Subtleties It is easy to observe that the zero knowledge property ensures confidentiality of the committed model and training data. However, as we shall see next, knowledge soundness does *not* necessarily capture the actual goal of the auditing process. The reason is that knowledge soundness is defined with respect to arbitrary statements $\mathsf{x} = (\mathsf{com}_h, \mathsf{com}_S, S_{audit})$, without specifying how or when each component of x is generated. In practice, it is plausible that S_{audit} is supplied by verifier (i.e., the auditor). We show that if a cheating prover (i.e., model owner) adaptively generates com_{h^*} and com_{S^*} *after* observing S_{audit}, it is possible to pass the zero knowledge auditing process after maliciously crafting model h^* and/or training data S^*. Furthermore, we show that h^* behaves pathologically when evaluated on data outside S_{audit}, in a way that completely undermines the purpose of the auditing process.

We note that while this subtlety was indeed overlooked in several works on zero-knowledge-based auditing, it applies even more directly to various non-cryptographic auditing approaches that do not enforce a secure commitment from the prover.

5 Methods

Because of the data-dependent nature of machine learning, previous work in verifiable ML may fail to reliably audit models, even while satisfying existing cryptographic definitions of security. To address this, we introduce new theoretical tools for analyzing cryptographic ML verification.

In §5.1, we present a formal security model for ML model audits on a given distribution. In §5.2, we give a concrete example of an attack for decision trees under which a broad class of existing privacy-preserving audit methods fail. In §5.3 we present evidence suggesting that such attacks may be difficult to detect.

5.1 Data-Dependent Security Models for Cryptographic ML Auditing

While several existing works propose cryptographic solutions to statistically auditing ML models, it is often left unspecified *when* the model owner receives an auditing dataset. In fact, several works suggest checking properties such as fairness using public reference datasets (e.g., the ones from the UCI repository). This setting allows potentially dishonest model owners to fine-tune their private models based on the known auditing data. The resulting model will pass the audit, but the *guarantee will fail to generalize to real-world inputs*. To capture this attack model, we introduce the following security game precisely clarifying the information available to the model owner at the time of submitting the trained model and dataset to the idealized auditing process.

Definition 3 (Adaptive Training with Known Auditing Data)**.** *Let $R : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ be an indexed NP-relation for model certification. Let $\mathcal{X}$ be the feature space and $\mathcal{D}$ be a distribution over $\mathcal{X}$. For a learner / training algorithm* Train *with a randomness space $\{0,1\}^\ell$, an* *auditing function F (outputting a binary), and a utility function L*[1]*, consider the following game $\mathsf{G}_{\mathcal{A}}(R, \mathsf{Train}, F, L, \mathcal{D}, \varepsilon)$ played by an adversary $\mathcal{A}$:*

$\underline{\mathsf{G}_{\mathcal{A}}(R, \mathsf{Train}, F, L, \mathcal{D}, \varepsilon)}$

1. *Sample $S_{train}, S_{audit} \sim \mathcal{D}$ and $\rho \leftarrow \{0,1\}^{\ell}$*
2. *Learn honest hypothesis $h \leftarrow \mathsf{Train}(S_{train}; \rho)$*
3. *Obtain $b_H = R((\mathsf{Train}, F), S_{audit}, (h, S_{train}, \rho))$*
4. *Given S_{train}, S_{audit}, Train, F, and g, $\mathcal{A}$ outputs a hypothesis h_A, a forged training dataset S'_{train}, and a training randomness ρ'.*
5. *Obtain $b_A = R((\mathsf{Train}, F), S_{audit}, (h_A, S'_{train}, \rho'))$*
6. *The output of the game is defined to be 1 (and $\mathcal{A}$ 'wins') if $b_A \geq b_H$ and $L(h_A) - L(h) > \varepsilon$. The output is 0 (and $\mathcal{A}$ 'loses') otherwise.*

Intuition At a high level, the attack above captures the following: For a given public training process Train and an adversarial utility function L, the adversary wins the game, if, upon learning an audit dataset, it provides a training dataset such that a model honestly trained on this dataset passes the audit and improves the adversarial utility compared to a model trained on an honest dataset. We also note that, unlike Section 4, the cryptographic commitment does not appear explicitly. This is because the game above implicitly models a situation where the adversary submits the hypothesis h and once and for all at Step 2., and the remaining operations are automatically performed on the same h, modeling the binding property in an idealized fashion. We now elaborate on the choices we make in this definition:

- *Public training procedure* Assuming that the training procedure is public makes only our attacks stronger. Intuitively, it is easier for the prover to satisfy an audit procedure if it the training algorithm is not known to the auditor.
- *Formalizing audit outcomes via an indexed relation*, rather than simply as the output of function F, allows us to more precisely capture the audit procedure as an interactive process involving both public audit data and the prover's private model/training data. We give examples of useful index relations below.
- *Requiring $b_A \geq b_H$* captures that the adversary is required to pass the audit only in cases where an honest model would also succeed. Intuitively, we do not wish to "punish" the adversary when the audit function itself is unreasonable—namely, when even an honestly trained model on an honest dataset would fail the audit.
- *Utility function* captures the adversary's goal, i.e., how desirable a given model is from its perspective. For instance, if the audit targets fairness, the utility may be the model's accuracy on $\mathcal{D}$, or the frequency with which it outputs negative predictions on certain inputs (e.g., to justify denying service to specific groups).

Examples of Index Relations An instantiation of the index relation depends on the auditing process. For instance, the auditor aims to validate the training process and accuracy or a certain group fairness property (e.g, demographic parity) described by a function F, R outputs 1 if and only the following conditions are satisfied.

- $h = \mathsf{Train}(S_{\text{train}}; \rho)$
- $F(\{h(r)\}_{r \in S_{\text{audit}}}, S_{\text{audit}}) = 1$

If the adversary is required to perform the t-test on the training data, we can define F as:

- $F_\alpha(S_{\text{audit}}, S'_{\text{train}}) = 1$ if and only S_{audit} passes t-test with significant level α

[1] We assume that distribution $\mathcal{D}$ is implicitly "known" to $\mathcal{L}$ (it is either hard-coded or provided as a parameter to $\mathcal{L}$). For simplicity of notation, we omit $\mathcal{D}$ from the description of $\mathcal{L}$.

5.2 Example of Data Forging Attack

We now give a concrete example of an attack within the framework of Def.3. In more detail, given a utility function L, a training dataset S_{train}, a fixed ϵ, and an audit dataset S_{audit} with honest labels y_x for $x \in S_{train} \cup S_{audit}$, we wish to construct a training dataset S'_{train} such that when a hypothesis h_A is honestly trained on S'_{train}, h_A passes the audit and the utility $L(h_A)$ is improved at least by ϵ compared to the utility of the model trained on an honestly sampled dataset.

To demonstrate the feasibility of our attack, we focus specifically on decision trees (see §3 for background). Say we consider datapoints in $\{0, 1\}^d$, the auditor's goal is to verify *accuracy* of the decision tree model, and the adversary's goal is to ensure that the model's accuracy is bad on real-world data. For example, we can set $L(h) = \Pr[h(x) \neq f(x)]$, where $f(x)$ denotes the true label of $x \in \{0, 1\}^d$.

To perform this attack, we begin with a reference dataset S_{audit}, and we will construct a training dataset S'_{train}. Let g denote a labeling function such that $g(x) = 1 \oplus f(x)$[2], where $x \in \{0, 1\}^d$. For every point $r \in S_{audit}$ with true label $f(r)$ and every (numeric) feature i, we add $r + \varepsilon\vec{b}_i$ with label $g(r)$, where ε is some small number and $\vec{b}_i$ is the ith basis vector. We also add every $r \in S_{audit}$ to S'_{train} with honest labels. Then, when training a tree on S'_{train}, we train until every leaf in the tree is homogeneous.

Algorithm 1 Data Forging Attack

Input: Audit set S_{audit}, dimension d, $\varepsilon > 0$, target labelling function g
Output: Training data S'_{train}

```
function ATTACK(S_audit, d, ε, g)
    S'_train ← S_audit
    for r ∈ S_audit do
        for i ∈ [d] do
            r_0 ← r + ε b_i          ▷ b_i is the one-hot vector in dimension i
            r_{0,y} ← g(r)           ▷ Set the label of r_0
            r_1 ← r − ε b_i
            r_{1,y} ← g(r)
            S'_train ← S'_train ∪ {r_0, r_1}
        end for
    end for
    return S'_train
end function
```

As we illustrate in §6, this simple attack already achieves surprisingly good results.

5.3 Detection

While proof of training alone cannot detect the attack above (as it relies on training the decision tree entirely honestly), nor can a black-box audit where the model owner knows the audit data before training time, we might still hope to detect when these attacks occur. For example, we might hope to conduct statistical tests on the training data to determine if it was honestly sampled from the underlying distribution or if it was adversarially constructed. In such a case, we cannot directly compare the training data to the true distribution of real data because the underlying distribution is not fully known to the auditor. Instead, we must compare the training data with a sample from that distribution. In the most simple case, this sample is the reference set S_{audit}.

We argue that under a certain family of functions, our constructed training set is indistinguishable from S_{audit}.

Definition 4. *Suppose $\vec{\alpha}$ is a set of bins over d dimensions. Then $H_{\vec{\alpha}} : (\mathbb{R}^d \times \{0, 1\})^* \to \mathcal{H}$ is the function which takes databases over d features and a binary classification to their normalized histogram with bins $\vec{\alpha}$.*

[2] For simplicity we consider a scenario with only two classes.

Definition 5. *A function $f : (\mathbb{R}^d \times \{0,1\})^* \to \mathbb{R}$ is called (γ, c)-magnitude insensitive if there exists a choice of bins $\vec{\alpha}$ and function $f' : \mathcal{H} \to \mathbb{R}$ such that $|f(D) - f'(H_{\vec{\alpha}}(D))| < \gamma$ for all $D \in (\mathbb{R}^d \times \{0,1\})^*$ and $|f'(H_{\vec{\alpha}}(D)) - f'(H_{\vec{\alpha}}(D||r))| \leq \frac{c}{|D|}$ for all $D \in (\mathbb{R}^d \times \{0,1\})^*$ and $r \in \mathbb{R}^d \times \{0,1\}$.*

Theorem 1. *If f is (γ, c)-magnitude insensitive, then $|f(S_{audit}) - f\left(S_{audit}^k||\delta\right)| \leq \varepsilon$ for any $\varepsilon > 2\gamma$ and $k \geq \frac{2dc}{\varepsilon - 2\gamma}$, where δ is as defined in Algorithm 1 when run with input $S_{audit}, d, \varepsilon, g$ for any g.*

Proof. See §A.5 for a formal proof. □

This theorem does not suggest that it is completely impossible to detect the attack given in Algorithm 1. Rather, it only precludes detection by a certain class of functions. However, we argue that this class is expansive and covers many intuitive approaches.

The sole requirement for the audit metric f is that it must be approximable by f' which satisfies three properties. Firstly, f' operates over histograms for some choice of bins $\vec{\alpha}$. This is a necessary condition, as if f were not approximable by a function over a binning of the training data, we could drastically change the audit outcome by simply adding a small amount of noise to the data. Next, f' must be relatively insensitive to additional data. The intuition here is that no individual datapoint should dramatically change the outcome of the audit. Finally, f' operates over normalized histograms. This property is necessary for the proof to go through, but is satisfied by many intuitive audit metrics. For example, the mean and standard deviation of a feature (even conditioned on any arbitrary set of features) are approximable from a normalized histogram.

Lemma 1. *Let $\mu_j(D)$ be the mean of (bounded) feature j of a dataset D. Then for every $\gamma > 0$, $\mu_j(D)$ is $(\gamma, M - m)$-magnitude insensitive, where B is the set of bins in the histogram and M, m are an upper and lower bound on possible j-values respectively.*

Proof. See §A.6 for a formal proof. □

We will proceed to use this fact to show that Welch's t-test will fail to detect this attack.

Corollary 1. *Given an audit dataset S_{audit} and significance level α, we can use Algorithm 1 to construct a training dataset S'_{train} such that for any feature j, S'_{train} passes Welch's t-test when its values in feature j are compared to those of S_{audit} with significance level α.*

Proof. See §A.7 for a formal proof. □

5.4 Resistance to data-forging in prior works

Our formalization from §5.1 allows us to easily check whether a certain protocol is susceptible to data-forging attacks. At a high level, for works which do not reveal neither the model nor the training data, the check boils down to whether the prover is required to commit to the training data and/or to the model before seeing the audit dataset. We examined several prior works with formal security guarantees, and, surprisingly, the majority of the works either do not explicitly state when the audit dataset is revealed, or consider settings where the prover's training dataset and/or the model itself are assumed to be trusted (and are susceptible to data forging if the prover is actually malicious). Additionally, works that do not discuss the timing of the commitment often point out that their solution can be used to conduct audits using publicly known datasets, in which case the public dataset can be assumed to be known to the adversary prior to the audit process. In this case the solution becomes vulnerable to data-forging.

We summarize the results of our findings in Table 1.

6 Evaluation

We implemented our attack from §5.2 in Python 3.12.3 using SciKit-Learn version 1.6.1 and evaluated its performance against the ACSEmployment dataset from Folktables. In particular, we used the 2018 Alabama dataset with a one-year horizon. For a given run, we split the dataset into an evaluation dataset consisting of 30% of the data, an audit dataset containing 1000 data points, and an extraneous

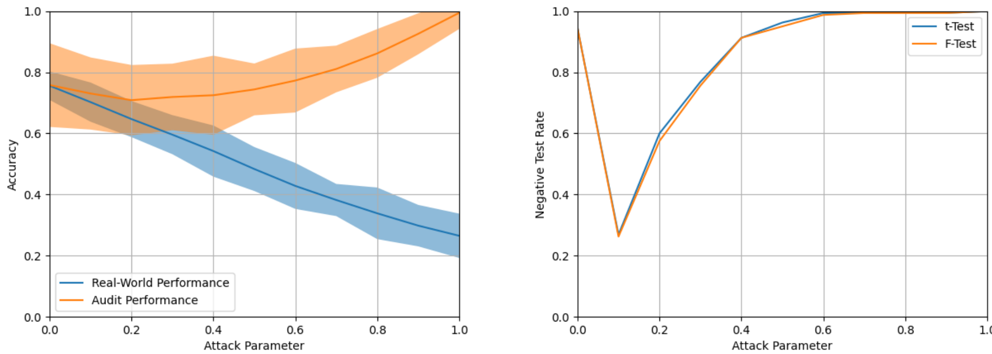

Figure 1: Performance of models trained on constructed datasets using ACSEmployment. An increase in the attack parameter represents increasing the number of audit points included in the attack as well as how many extraneous points are labeled maliciously. Values are averages over ten runs, error bars represent one standard deviation.

Table 2: Summary and Test statistics for Age feature on ACSEmployment, conditioned on label. Test statistics used are Welch's t-test and Levene's test. Attack is undetectable when summary statistics are similar to honest ones, and when test statistics are close to 0. Comparisons are between fully honest and fully malicious datasets.

Age		Label = 0		Label = 1	
		Honest	Attack	Honest	Attack
Summary Statistics	μ	41.6651	41.9657	43.9184	43.8131
	σ^2	804.5804	810.8822	223.1269	221.42394
Test Statistics	t-test	0.6521	0.0033	0.7067	0.0110
	F-test	0.6200	0.0026	1.6500	0.0186

Military Status		Label = 0		Label = 1	
		Honest	Attack	Honest	Attack
Summary Statistics	μ	2.5794	2.5834	3.8121	3.8302
	σ^2	3.2749	3.2648	0.3507	0.3265
Test Statistics	t-test	0.4997	0.0313	0.8699	0.1755
	F-test	1.0240	0.0009	1.2394	0.0304

training data set. In order to determine the number of copies of the audit data to add to the training data, we partitioned the audit data by label and computed the k-values necessary for each feature to pass Welch's t-test with significance 0.05, and selected the largest finite such value. We constructed the training data according to Algorithm 1, and used it to fit a decision tree using SciKit-Learn's decision tree classifier class. We measured various statistics over the predictions made by the classifier, which we present in Figure 1.

We find that we are capable of tuning the attack to enforce high audit accuracy while simultaneously encouraging low performance on real-world evaluation data. Rather than performing a maximally malicious attack, one might choose to perform a less overt attack by including less of the audit data in the training data. However, we observe that with an attack parameter below 0.5, the chance of passing the t-test or F-test falls precipitously.

We observe that the summary statistics of the malicious training data closely match the values for the honest data, suggesting that comparing these two values would not be a successful detection mechanism. This is compounded by the fact that the test statistics for Welch's t-test and Levene's test for the malicious training data are considerably smaller on average than the same test statistics for the honest training data, corroborating higher rate of passing the hypothesis tests we observe. At a significance level of $\alpha = 0.05$, we expect a false positive rate of approximately 5%. On the other hand, we observe a 0% true negative rate when employing the attack.

References

K. Abbaszadeh, C. Pappas, J. Katz, and D. Papadopoulos. Zero-knowledge proofs of training for deep neural networks. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 4316–4330, 2024.

BABL AI. Independent algorithm audits for ai systems. `https://babl.ai/`. Accessed: 2025-08-22.

M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Mach. Learn.*, 81(2):121–148, 2010.

J. G. Bourrée, H. Lautraite, S. Gambs, G. Trédan, E. L. Merrer, and B. Rottembourg. P2NIA: privacy-preserving non-iterative auditing. *CoRR*, abs/2504.00874, 2025. doi: 10.48550/ARXIV.2504.00874. URL `https://doi.org/10.48550/arXiv.2504.00874`.

I. Chang, K. Sotiraki, W. Chen, M. Kantarcioglu, and R. Popa. {HOLMES}: Efficient distribution testing for secure collaborative learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4823–4840, 2023.

Deloitte. Algorithm & ai assurance. `https://www.deloitte.com/uk/en/services/audit-assurance/services/Algorithm-Assurance.html`. Accessed: 2025-08-22.

V. Duddu, A. Das, N. Khayata, H. Yalame, T. Schneider, and N. Asokan. Attesting distributional properties of training data for machine learning. In *European Symposium on Research in Computer Security*, pages 3–23. Springer, 2024.

O. Franzese, A. S. Shamsabadi, and H. Haddadi. Oath: Efficient and flexible zero-knowledge proofs of end-to-end ml fairness. *arXiv preprint arXiv:2410.02777*, 2024.

S. Garg, A. Goel, S. Jha, S. Mahloujifar, M. Mahmoody, G.-V. Policharla, and M. Wang. Experimenting with zero-knowledge proofs of training. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1880–1894, 2023.

HFS Research. UnitedHealthcare's AI Use to Deny Claims Is Center of Industrywide Debate. `https://www.hfsresearch.com/news/unitedhealthcares-ai-use-to-deny-claims-is-center-of-industrywide-debate/`, 2024. Published December 9, 2024.

D. Kang, T. Hashimoto, I. Stoica, and Y. Sun. Scaling up trustless DNN inference with zero-knowledge proofs. *CoRR*, abs/2210.08674, 2022. doi: 10.48550/ARXIV.2210.08674. URL `https://doi.org/10.48550/arXiv.2210.08674`.

T. Liu, X. Xie, and Y. Zhang. zkcnn: Zero knowledge proofs for convolutional neural network predictions and accuracy. In Y. Kim, J. Kim, G. Vigna, and E. Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 2968–2985. ACM, 2021. doi: 10.1145/3460120.3485379. URL `https://doi.org/10.1145/3460120.3485379`.

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

M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

Mosaic Data Science. Explainable ai services & bias auditing. `https://mosaicdatascience.com/top-ai-consulting-partner-engagement-models/explainable-ai-services-bias-auditing/`. Accessed: 2025-08-22.

ORCAA. Algorithmic auditing services. `https://orcaarisk.com/`. Accessed: 2025-08-22.

C. Pappas and D. Papadopoulos. Sparrow: Space-efficient zksnark for data-parallel circuits and applications to zero-knowledge decision trees. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3110–3124, 2024.

P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

A. S. Shamsabadi, S. C. Wyllie, N. Franzese, N. Dullerud, S. Gambs, N. Papernot, X. Wang, and A. Weller. Confidential-profitt: confidential proof of fair training of trees. In *The Eleventh International Conference on Learning Representations*, 2022.

A. S. Shamsabadi, G. Tan, T. I. Cebere, A. Bellet, H. Haddadi, N. Papernot, X. Wang, and A. Weller. Confidential-dpproof: Confidential proof of differentially private training. In *International Conference on Learning Representations (ICLR)*, 2024.

J. Steinhardt, P. W. Koh, and P. Liang. Certified defenses for data poisoning attacks. In *NIPS*, pages 3517–3529, 2017.

H. Sun, T. Bai, J. Li, and H. Zhang. Zkdl: Efficient zero-knowledge proofs of deep learning training. *IEEE Transactions on Information Forensics and Security*, 2024a.

H. Sun, J. Li, and H. Zhang. zkllm: Zero knowledge proofs for large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 4405–4419, 2024b.

S. Tan, R. Caruana, G. Hooker, and Y. Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *AIES*, pages 303–310. ACM, 2018.

F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 401–416. IEEE, 2017.

S. Waiwitlikhit, I. Stoica, Y. Sun, T. Hashimoto, and D. Kang. Trustless audits without revealing data or models. *arXiv preprint arXiv:2404.04500*, 2024.

H. Wang and T. Hoang. ezdps: An efficient and zero-knowledge machine learning inference pipeline. *Proc. Priv. Enhancing Technol.*, 2023(2):430–448, 2023. doi: 10.56553/POPETS-2023-0061. URL `https://doi.org/10.56553/popets-2023-0061`.

B. L. Welch. The generalization of 'student's'problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 1947.

C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang. Mystique: Efficient conversions for {Zero-Knowledge} proofs with applications to machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 501–518, 2021.

T. Xie, T. Lu, Z. Fang, S. Wang, Z. Zhang, Y. Jia, D. Song, and J. Zhang. zkpytorch: A hierarchical optimized compiler for zero-knowledge machine learning. *Cryptology ePrint Archive*, 2025.

C. Yadav, A. R. Chowdhury, D. Boneh, and K. Chaudhuri. Fairproof: Confidential and certifiable fairness for neural networks. *arXiv preprint arXiv:2402.12572*, 2024.

C. Yadav, E. M. Laufer, D. Boneh, and K. Chaudhuri. Expproof: Operationalizing explanations for confidential models with zkps. *arXiv preprint arXiv:2502.03773*, 2025.

J. Zhang, Z. Fang, Y. Zhang, and D. Song. Zero knowledge proofs for decision tree predictions and accuracy. In *CCS*, pages 2039–2053. ACM, 2020.

T. Zhang, S. Dong, O. D. Kose, Y. Shen, and Y. Zhang. Fairzk: A scalable system to prove machine learning fairness in zero-knowledge. In M. Blanton, W. Enck, and C. Nita-Rotaru, editors, *IEEE Symposium on Security and Privacy, SP 2025, San Francisco, CA, USA, May 12-15, 2025*, pages 3460–3478. IEEE, 2025. doi: 10.1109/SP61157.2025.00205. URL `https://doi.org/10.1109/SP61157.2025.00205`.

Algorithm 2 Welch's t-test

Input: $\mathcal{X} = \{x_i\}_{i \in [n]}$, $\mathcal{Y} = \{y_i\}_{i \in [m]}$, where $x_i \sim X$ and $y_i \sim Y$, and a significance level α
Output: Null hypothesis H_0 (i.e., $\mu_X = \mu_Y$) or alternative hypothesis H_1 (i.e., $\mu_X \neq \mu_Y$)

1: Compute sampled means $\bar{x} = \frac{\sum_i x_i}{n}$ and $\bar{y} = \frac{\sum_i y_i}{m}$
2: Compute sampled variances $v_x = \frac{\sum_i (\bar{x} - x_i)^2}{n-1}$ and $v_y = \frac{\sum_i (\bar{y} - y_i)^2}{m-1}$.
3: Compute the test statistic $t = \frac{\bar{x} - \bar{y}}{\sqrt{v_x/n + v_y/m}}$
4: Compute the degree of freedom $d = \frac{(g_x + g_y)^2}{g_x^2/(n-1) + g_y^2/(m-1)}$, where $g_x = v_x/n$ and $g_y = v_y/m$
5: Obtain the critical value t_{cr} from the t-table, given d and α.
6: **If** $|t| < t_{cr}$ **return** H_0 **else return** H_1

A Appendix

A.1 Welch's t-test

A.2 Data Poisoning Attacks

Data poisoning attacks involve adversarial manipulations of training data with the goal to degrade a model's performance. This active line of work produced numerous interesting results in the past years [Steinhardt et al., 2017]. Traditionally, such attacks have been considered in the context of machine learning systems trained on user-provided data. This setting is conceptually different from ours: In data poisoning, the "model owner" is typically considered honest, and the concern is that users contributing to the model can inject malicious data. As a result, data poisoning attacks involve subtle, often small-scale perturbations to a subset of the training examples. As defined byl. Barreno et al. [2010], data poisoning can be viewed as a game between a *defender*, who seeks to learn an accurate model, and an *attacker*, whose goal is to corrupt the learned model. In this setting, the model is trained on the combination of a clean dataset D_c and a poisoned dataset D_p, where the size of D_p is constrained to be no larger than that of D_c. In contrast, our setting allows for the fully malicious model owner. Its goal is to engineer a model that passes an audit, while violating the certified properties on real-world data. In particular, in our setting the adversary is not restricted to small-scale perturbations of the clean training data.

A.3 Decision Tree Inference

For completeness, in Algorithm 3 we present the algorithm for decision tree inference.

Algorithm 3 Decision Tree Inference

Input: Decision tree h, input $\mathbf{a}$.
Output: Classification result.

```
1: Let cur := h.root                          ▷ Set cur to be root of the tree
2: while cur is not a leaf do
3:     if a[cur.attr] < cur.thr then
4:         cur := cur.left.                   ▷ Set cur to be current node's left child
5:     else
6:         cur := cur.right.                  ▷ Set cur to be current node's right child
7:     end if
8: end while
9: return cur.class
```

A.4 Security Properties of Zero-Knowledge Proofs

Completeness Π is (perfectly) *complete* if for any $(\mathsf{i}, \mathsf{x}, \mathsf{w})$ satisfying $\mathcal{R}$, it holds that:

$$\Pr[1 \leftarrow \langle \mathcal{P}(\mathsf{w}), \mathcal{V} \rangle (\mathsf{i}, \mathsf{x})] = 1.$$

Knowledge Soundness Π is *knowledge sound* if there exists an expected polynomial time extractor $\mathcal{E}$ such that for any PPT adversary $\mathcal{P}^*$ and any $\mathsf{i} \in \{0,1\}^*$ $\mathsf{x} \in \{0,1\}^\lambda$, the following probability is negligible in λ:

$$\Pr\left[b = 1 \wedge (\mathsf{i},\mathsf{x},\mathsf{w}) \notin \mathcal{R} \,:\, b \leftarrow \langle \mathcal{P}^*, \mathcal{V}\rangle(\mathsf{i},\mathsf{x}); \mathsf{w} \leftarrow \mathcal{E}^{\mathcal{P}^*}(\mathsf{i},\mathsf{x})\right]$$

where $\mathcal{E}$ has black-box access to $\mathcal{P}^*$. Informally, this means that any cheating prover must know a valid witness if it convinces verifier.

Zero-Knowledge Let $\mathsf{view}_{\mathcal{V}}^{\mathcal{P}(\mathsf{w})}(\mathsf{i},\mathsf{x})$ be a string consisting of all the incoming messages that $\mathcal{V}$ receives from $\mathcal{P}$ during the interaction $\langle \mathcal{P}(\mathsf{w}), \mathcal{V}\rangle(\mathsf{i},\mathsf{x})$, and $\mathcal{V}$'s random coins. Π is (honest verifier) *zero-knowledge* if there exists a PPT simulator $\mathcal{S}$ such that for any adversary $\mathcal{A}$ and any $(\mathsf{i},\mathsf{x},\mathsf{w})$ satisfying $\mathcal{R}$, the following is negligible in λ.

$$\left|\Pr\left[b = 1 \,:\, b \leftarrow \mathcal{A}(\mathsf{view}_{\mathcal{V}}^{\mathcal{P}(\mathsf{w})}(\mathsf{i},\mathsf{x}))\right] - \Pr\left[b = 1 \,:\, \mathsf{view}' \leftarrow \mathcal{S}(\mathsf{i},\mathsf{x}); \quad b \leftarrow \mathcal{A}(\mathsf{view}')\right]\right|$$

Informally, this means that the protocol execution reveals no information about w.

A.5 Proof of Theorem 1

Proof. We will write f' to be the γ-approximation of f guaranteed to exist by the fact that f is (γ, c)-magnitude insensitive. Observe that because $H_{\vec{\alpha}}$ takes databases to their normalized histograms, $H_{\vec{\alpha}}(S_{audit}) = H_{\vec{\alpha}}\left(S^k_{audit}\right)$, because the non-normalized histograms of the two databases are simply scaled versions of one another.

Next, it will be helpful to show that for any two databases $D_1, D_2 \in (\mathbb{R}^d \times \{0,1\})^*$, we have $|f'(H_{\vec{\alpha}}(D_1)) - f'(H_{\vec{\alpha}}(D_1||D_2))| \le c\frac{|D_2|}{|D_1|}$. Let us write $D_2 = d_1||d_2||\ldots||d_{|D_2|}$. Then we get that

$$\begin{aligned}
&|f'(H_{\vec{\alpha}}(D_1)) - f'(H_{\vec{\alpha}}(D_1||D_2))| \\
=&|f'(H_{\vec{\alpha}}(D_1)) - f'(H_{\vec{\alpha}}(D_1||d_1)) + f'(H_{\vec{\alpha}}(D_1||d_1)) - \ldots + f'(H_{\vec{\alpha}}(D_1||d_1||d_2||\ldots||d_{|D_2|-1})) - f'(H_{\vec{\alpha}}(D_1||D_2))| \\
\le&|f'(H_{\vec{\alpha}}(D_1)) - f'(H_{\vec{\alpha}}(D_1||d_1))| + |f'(H_{\vec{\alpha}}(D_1||d_1)) - f'(H_{\vec{\alpha}}(D_1||d_1||d_2)| + \ldots + |f'(H_{\vec{\alpha}}(D_1||d_1||d_2||\ldots||d_{|D_2|-1})) - \\
\le&\frac{c}{|D_1|} + \frac{c}{|D_1|+1} + \ldots + \frac{c}{|D_1|+|D_2|-1} \\
\le&c\frac{|D_2|}{|D_1|}
\end{aligned}$$

Then we can apply this to S^k_{audit} and $S^k_{audit}||\delta$; recall that $|\delta| = 2d|S_{audit}|$. Then we see that $\left|f'\left(H_{\vec{\alpha}}\left(S_{audit}\right)\right) - f'\left(H_{\vec{\alpha}}\left(S^k_{audit}||\delta\right)\right)\right| = \left|f'\left(H_{\vec{\alpha}}\left(S^k_{audit}\right)\right) - f'\left(H_{\vec{\alpha}}\left(S^k_{audit}||\delta\right)\right)\right| \le c\frac{2d|S_{audit}|}{k|S_{audit}|} \le c\frac{2d}{\left(\frac{2dc}{\varepsilon-2\gamma}\right)} = \varepsilon - 2\gamma$. We have two cases now.

Case 1: $f'\left(H_{\vec{\alpha}}\left(S_{audit}\right)\right) \ge f'\left(H_{\vec{\alpha}}\left(S^k_{audit}||\delta\right)\right)$. Then we have

$$\begin{aligned}
\varepsilon - 2\gamma \ge\, & f'\left(H_{\vec{\alpha}}\left(S_{audit}\right)\right) - f'\left(H_{\vec{\alpha}}\left(S^k_{audit}||\delta\right)\right) \\
&= f(S_{audit}) - f(S_{audit}) + f'(H_{\vec{\alpha}}(S_{audit})) - f(S^k_{audit}||\delta) + f(S^k_{audit}||\delta) - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) \\
&\ge f(S_{audit}) - |f(S_{audit}) - f'(H_{\vec{\alpha}}(S_{audit}))| - f(S^k_{audit}||\delta) - |f(S^k_{audit}||\delta) - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta))| \\
&\ge f(S_{audit}) - \gamma - f(S^k_{audit}||\delta) - \gamma
\end{aligned}$$

and so we see that $\varepsilon \ge f(S_{audit}) - f(S^k_{audit}||\delta)$. We also have

$$\begin{aligned}
f(S_{audit}) - f(S^k_{audit}||\delta) &= f'(H_{\vec{\alpha}}(S_{audit})) - f'(H_{\vec{\alpha}}(S_{audit})) + f(S_{audit}) - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) + f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - f(S \\
&\ge f'(H_{\vec{\alpha}}(S_{audit})) - |f'(H_{\vec{\alpha}}(S_{audit})) - f(S_{audit})| - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - |f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - f \\
&\ge f'(H_{\vec{\alpha}}(S_{audit})) - \gamma - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - \gamma \\
&\ge -2\gamma \\
&> -\varepsilon
\end{aligned}$$

Then $|f(S_{audit}) - f(S^k_{audit}||\delta)| \leq \varepsilon$.

Case 2: $f'(H_{\vec{\alpha}}(S_{audit})) \leq f'(H_{\vec{\alpha}}(S^k_{audit}||\delta))$. Then we have

$$\begin{aligned}
\varepsilon - 2\gamma &\geq f'\left(H_{\vec{\alpha}}\left(S^k_{audit}||\delta\right)\right) - f'\left(H_{\vec{\alpha}}\left(S_{audit}\right)\right) \\
&= f(S^k_{audit}||\delta) - f(S^k_{audit}||\delta) + f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - f(S_{audit}) + f(S_{audit}) - f'(H_{\vec{\alpha}}(S_{audit})) \\
&\geq f(S^k_{audit}||\delta) - |f(S^k_{audit}||\delta) - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta))| - f(S_{audit}) - |f(S_{audit}) - f'(H_{\vec{\alpha}}(S_{audit}))| \\
&\geq f(S^k_{audit}||\delta) - \gamma - f(S_{audit}) - \gamma
\end{aligned}$$

and so we see that $\varepsilon \geq f(S^k_{audit}||\delta) - f(S_{audit})$. We also have

$$\begin{aligned}
f(S^k_{audit}||\delta) - f(S_{audit}) &= f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) + f(S^k_{audit}||\delta) - f'(H_{\vec{\alpha}}(S_{audit})) + f'(H_{\vec{\alpha}}(S_{audit})) - \\
&\geq f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - |f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) + f(S^k_{audit}||\delta)| - f'(H_{\vec{\alpha}}(S_{audit})) - |f'(H_{\vec{\alpha}}(S_{audit})) - \\
&\geq f'(H_{\vec{\alpha}}(S^k_{audit}||\delta)) - \gamma - f'(H_{\vec{\alpha}}(S_{audit})) - \gamma \\
&\geq -2\gamma \\
&\geq -\varepsilon
\end{aligned}$$

Then $|f(S_{audit}) - f(S^k_{audit}||\delta)| \leq \varepsilon$. □

A.6 Proof of Lemma 1

Proof. Notice that $\mu_j(D) \approx \sum_{i \in B} p_i x_{j,i}$ where B is the set of bins in the histogram, p_i is the height of bin i in the normalized histogram of D, and $x_{j,i}$ is the j-value of bin i. Let us show that for any $\gamma > 0$, there exists a binning of the data such that this is a γ-approximation of $\mu_j(D)$. Let the bins in feature j have width γ. Then for each datapoint d with j value j_d, bin i, and binned j-value $x_{j,i}$, we have that $|x_{j,i} - j_d| \leq \gamma$. Then

$$\begin{aligned}
\sum_{i \in B} p_i x_{j,i} &= \sum_{i \in B} \frac{c_i}{|D|} x_{j,i} \\
&= \sum_{d \in D} \frac{1}{|D|} x_{j,i} \\
\implies \left| \sum_{i \in B} p_i x_{j,i} - \sum_{d \in D} \frac{1}{|D|} j_d \right| &= \left| \sum_{d \in D} \frac{1}{|D|} x_{j,i} - \sum_{d \in D} \frac{1}{|D|} j_d \right| \\
&= \left| \frac{1}{|D|} \sum_{d \in D} (x_{j,i} - j_d) \right| \\
&\leq \frac{1}{|D|} \sum_{d \in D} |x_{j,i} - j_d| \\
&\leq \frac{1}{|D|} \sum_{d \in D} \gamma \\
&= \gamma
\end{aligned}$$

Next, let us show that the sensitivity of our approximation of μ_j is upper bounded by $\frac{M-m}{|D|}$. Notice that by adding a single point, one histogram bin will increase by 1 and the rest will be unchanged.

Then for every bin k,

$$
\begin{aligned}
\sum_{i\in B}\frac{c_i}{|D|+1}x_{j,i}+\frac{1}{|D|+1}x_{j,k}-\sum_{i\in B}\frac{c_i}{|D|}x_{j,i} &= \sum_{i\in B}c_i x_{j,i}\left(\frac{1}{|D|+1}-\frac{1}{|D|}\right)+\frac{x_{j,k}}{|D|+1}\\
&= -\left(\sum_{i\in B}\frac{c_i x_{j,i}}{|D|^2+|D|}\right)+\frac{x_{j,k}}{|D|+1}\\
&\leq -\left(\frac{m}{|D|+1}\right)+\frac{M}{|D|+1}\\
&\leq \frac{M-m}{|D|}\\
\sum_{i\in B}\frac{c_j}{|D|+1}x_{j,i}+\frac{1}{|D|+1}x_{j,k}-\sum_{i\in B}\frac{c_j}{|D|}x_{j,i} &= -\left(\sum_{i\in B}\frac{c_i x_{j,i}}{|D|^2+|D|}\right)+\frac{x_{j,k}}{|D|+1}\\
&\geq -\left(\frac{M}{|D|+1}\right)+\frac{m}{|D|+1}\\
&\geq \frac{m-M}{|D|}
\end{aligned}
$$

So we have that the sensitivity is no greater than $\frac{M-m}{|D|}$. □

A.7 Proof of Corollary 1

Before we can prove this corollary, we will need a lemma which bounds the concentration of the Student's t-distribution.

Lemma 2. *If X and Z are random variables drawn independently from the Student's t-distribution with ν degrees of freedom and the standard normal distribution respectively, then for every $t > 0$, we have*

$$\Pr[|X| < t] \leq \Pr[|Z| < t]$$

Proof. We will write $F_X(t)$ to denote the CDF of random variable X evaluated at t, and $f_X(t)$ the PDF. We will also write $\mathbb{E}_X(g(X))$ to be the expected value of $g(X)$ with randomness over X. Let us begin by demonstrating that for all $t < 0$, we have $F_X(t) > F_Z(t)$. First, recall that if W and Y are drawn from the χ^2 distribution with ν degrees of freedom and the standard normal distribution respectively, then $Y\sqrt{\frac{\nu}{W}}$ is distributed according to the Student's t-distribution with ν degrees of freedom, so let us write $X = Y\sqrt{\frac{\nu}{W}}$. Then according to the law of total probability, we have

$$
\begin{aligned}
F_X(t) &= \int_0^\infty F_Y\left(t\sqrt{\frac{w}{\nu}}\right)f(w)dw\\
&= \mathbb{E}_W\left(F_Y\left(t\sqrt{\frac{W}{\nu}}\right)\right)
\end{aligned}
$$

Notice that $\frac{d^2}{dt^2}F_Y(t) = \frac{d}{dt}f_Y(t) = \frac{d}{dt}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}} = -\frac{t}{\sqrt{2\pi}}e^{-\frac{t^2}{2}} > 0$ when $t < 0$. Then since $t\sqrt{\frac{W}{\nu}}$ must be less than 0, we can apply Jensen's inequality to get

$$
\begin{aligned}
F_X(t) &= \mathbb{E}_W\left(F_Y\left(t\sqrt{\frac{W}{\nu}}\right)\right)\\
&\geq F_Y\left(\mathbb{E}_W\left(t\sqrt{\frac{W}{\nu}}\right)\right)\\
&= F_Y\left(t\mathbb{E}_W\left(\sqrt{\frac{W}{\nu}}\right)\right)
\end{aligned}
$$

Then since $\frac{d^2}{du^2}\sqrt{u} = -\frac{1}{4\sqrt{u^3}} \leq 0$, we get that $\mathbb{E}_W\left(\sqrt{\frac{W}{\nu}}\right) \leq \sqrt{\frac{\mathbb{E}_W(W)}{\nu}} = \sqrt{\frac{\nu}{\nu}} = 1$. So because $t < 0$, we can see that $t\mathbb{E}_W\left(\sqrt{\frac{W}{\nu}}\right) \geq t$, and since $F_Y(u)$ is increasing, we get

$$\begin{aligned} F_X(t) &\geq F_Y\left(t\mathbb{E}_W\left(\sqrt{\frac{W}{\nu}}\right)\right) \\ &\geq F_Y(t) \end{aligned}$$

Since f_X and f_Y are both symmetric about $t = 0$, it then follows by a symmetric argument that for all $t > 0$, $F_X(t) \leq F_Y(t)$. Then we see that for any $t > 0$,

$$\begin{aligned} \Pr[|X| < t] &= F_X(t) - F_X(-t) \\ &\leq F_Y(t) - F_Y(-t) \\ &= \Pr[|Y| < t] \\ &= \Pr[|Z| < t] \end{aligned}$$

Because Y and Z are independently and identically distributed. □

We are now ready to prove Corollary 1.

Proof of Corollary 1. A pair of datasets D_1, D_2 pass Welch's t-test on feature j if

$$\frac{|\mu_j(D_1) - \mu_j(D_2)|}{\sqrt{\frac{\sigma_1^2}{|D_1|} + \frac{\sigma_2^2}{|D_2|}}} \leq T_{\alpha,\nu}$$

where α is the desired significance level, ν is the degrees of freedom in the datasets, and $T_{\alpha,\nu}$ is the unique value such that

$$\Pr_{x \sim t(\nu)}[|x| \geq T_{\alpha,\nu}] = \alpha$$

where $t(\nu)$ is the Student's t-distribution with ν degrees of freedom. In our case, the t-test compares the reference dataset S_{audit} with the training dataset S'_{train}.

The value of ν, and thus the value of $T_{\alpha,\nu}$, depends on the size of the datasets, with the threshold $T_{\alpha,\nu}$ decreasing as the datasets grow large. However, we will use Lemma 2 to give a lower bound for $T_{\alpha,\nu}$ which is constant with respect to $|S'_{train}|$. Then, we will show that by Lemma 1 and Theorem 1 we can use Algorithm 1 to construct a malicious training dataset S'_{train} which maintains an arbitrarily small test statistic, and in particular, a dataset such that the test statistic is below the lower bound on the threshold.

First, let us establish a lower bound on $T_{\alpha,\nu}$. Let us define T'_α to be the unique positive value such that

$$\Pr_{Z \sim \mathcal{N}(0,1)}[|Z| \geq T'_\alpha] = \alpha$$

Then recall that Lemma 2 gives us that

$$\Pr_{X \sim t(\nu)}[|X| < T'_\alpha] \leq \Pr_{Z \sim \mathcal{N}(0,1)}[|Z| < T'_\alpha]$$

If we write f_X and f_Z to represent the probability density functions (PDFs) of X and Z respectively, then we get equivalently that

$$\int_{-T'_\alpha}^{T'_\alpha} f_X(u)du \leq \int_{-T'_\alpha}^{T'_\alpha} f_Z(u)du$$

Then we see that

$$\begin{aligned}
\Pr_{Z\sim\mathcal{N}(0,1)}[|Z| \geq T'_\alpha] &= \Pr_{X\sim t(\nu)}[|X| \geq T_{\alpha,\nu}] \\
\implies \int_{-T'_\alpha}^{T'_\alpha} f_Z(u)du &= \int_{-T_{\alpha,\nu}}^{T_{\alpha,\nu}} f_X(u)du \\
&= \int_{-T_{\alpha,\nu}}^{-T'_\alpha} f_X(u)du + \int_{-T'_\alpha}^{T'_\alpha} f_X(u)du + \int_{T'_\alpha}^{T_{\alpha,\nu}} f_X(u)du \\
&\leq \int_{-T_{\alpha,\nu}}^{-T'_\alpha} f_X(u)du + \int_{-T'_\alpha}^{T'_\alpha} f_Z(u)du + \int_{T'_\alpha}^{T_{\alpha,\nu}} f_X(u)du \\
\implies 0 &\leq \int_{-T_{\alpha,\nu}}^{-T'_\alpha} f_X(u)du + \int_{T'_\alpha}^{T_{\alpha,\nu}} f_X(u)du
\end{aligned}$$

Then because $f_X(x)$ is symmetric about $x = 0$, this yields

$$2\int_{T'_\alpha}^{T_{\alpha,\nu}} f_X(u)du \geq 0$$

and thus

$$\int_{T'_\alpha}^{T_{\alpha,\nu}} f_X(u)du \geq 0$$

Now recall the simple result from calculus that states that if g is positive valued, then

$$\int_a^b g(x)dx \geq 0 \iff a \leq b$$

Then because f_X is positive-valued, our prior result entails that $T_{\alpha,\nu} \geq T'_\alpha$, so T'_α is a lower bound on $T_{\alpha,\nu}$ that does not depend on $|S'_{train}|$.

Next, observe that the test statistic for Welch's t-test has the following upper bound:

$$\frac{|\mu_j(S'_{train}) - \mu_j(S_{audit})|}{\sqrt{\frac{\sigma^2_{train}}{|S'_{train}|} + \frac{\sigma^2_{audit}}{|S_{audit}|}}} \leq \frac{|\mu_j(S'_{train}) - \mu_j(S_{audit})|}{\sqrt{\frac{\sigma^2_{audit}}{|S_{audit}|}}}$$

Furthermore, Lemma 1 implies that for any $\varepsilon > 0$, we can choose $\gamma < \frac{\varepsilon}{2}$ such that μ_j is (γ, c)-magnitude insensitive, and so by Theorem 1, Algorithm 1 yields a dataset S'_{train} such that $|\mu_j(S'_{train}) - \mu_j(S_{audit})| \leq \varepsilon$ when appropriately parameterized. Then let $\varepsilon = T'_\alpha \frac{\sigma_{audit}}{2\sqrt{|S_{audit}|}}$. This produces the result that

$$\begin{aligned}
\frac{|\mu_j(S'_{train}) - \mu_j(S_{audit})|}{\sqrt{\frac{\sigma^2_{train}}{|S'_{train}|} + \frac{\sigma^2_{audit}}{|S_{audit}|}}} &\leq \frac{2\varepsilon}{\sqrt{\frac{\sigma^2_{audit}}{|S_{audit}|}}} \\
&= \frac{2}{\sqrt{\frac{\sigma^2_{audit}}{|S_{audit}|}}} T'_\alpha \frac{\sigma_{audit}}{2\sqrt{|S_{audit}|}} \\
&= T'_\alpha \\
&\leq T_{\alpha,\nu}
\end{aligned}$$

which passes the t-test for feature j. Finally, by choosing $k = \max_j \frac{4d(M_j - m_j)\sqrt{|S_{audit}|}}{T'_\alpha \sigma_{audit,j}}$ we get for every feature i that $|\mu_i(S'_{train}) - \mu_i(S_{audit})| \leq 2 \min_j T'_\alpha \frac{\sigma_{audit,j}}{2\sqrt{|S_{audit}|}} \leq 2T'_\alpha \frac{\sigma_{audit,i}}{2\sqrt{|S_{audit}|}}$, so S'_{train} passes the t-test for feature i. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main technical results are in §5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We elaborate on limitations of the undetectability of our attacks in §5.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We give a set of proofs in §A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We explain how we performed the experiments in §6.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will provide open access to the data and code in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We explain how we performed the experiments in §6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We explain how we performed the experiments in §6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.