Reconstructing Test Labels From Noisy Loss Functions

Abhinav Aggarwal
Amazon, USA

Shiva Prasad Kasiviswanathan
Amazon, USA

Zekun Xu
Amazon, USA

Oluwaseyi Feyisetan
Amazon, USA

Nathanael Teissier
Amazon, USA

Abstract

Machine learning classifiers rely on loss functions for performance evaluation, often on a private (hidden) dataset. In a recent line of research, label inference was introduced as the problem of reconstructing the ground truth labels of this private dataset from just the (possibly perturbed) cross-entropy loss function values evaluated at chosen prediction vectors (without any other access to the hidden dataset). In this paper, we formally study the necessary and sufficient conditions under which label inference is possible from *any* (noisy) loss function value. Using tools from analytical number theory, we show that a broad class of commonly used loss functions, including general Bregman divergence-based losses and multiclass cross-entropy with common activation functions like sigmoid and softmax, it is possible to design label inference attacks that succeed even for arbitrary noise levels and using only a single query from the adversary. We formally study the computational complexity of label inference and show that while in general, designing adversarial prediction vectors for these attacks is co-NP-hard, once we have these vectors, the attacks can also be carried out through a lightweight augmentation to any neural network model, making them look benign and hard to detect. The observations in this paper provide a deeper understanding of the vulnerabilities inherent in modern machine learning and could be used for designing future trustworthy ML.

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

1 INTRODUCTION

Consider a situation where a machine learning (ML) modeler is interacting with a data curator who owns a private dataset for a classification task. The curator agrees to evaluate on this private dataset the prediction vector (or an ML model) that the modeler submits, and replies back with loss function values. Such a situation is commonly encountered in machine learning competition settings like Kaggle (kag), KDDCup (kdd), and ILSVRC Challenges (ils). In some competitions, the features of the private (hold-out) dataset are revealed but not its labels, and the modeler submits the prediction vector on those features. In some other competitions, no information about the private dataset is revealed (i.e., neither the features nor the labels). The modeler submits a model that is then evaluated on the private dataset. A similar situation also appears when dealing with sensitive datasets, where either labels, or, both features and labels could be considered sensitive, and a modeler and curator interact through loss scores.

In this paper, we investigate if it is possible for a (malicious) modeler to recover *all* the private labels using these interactions with the data curator (server). More broadly, we investigate the problem of robust label inference, where the goal is to infer the labels of a hidden dataset from only the (noisy) loss function queries evaluated on the dataset. Of particular interest will be the case where the modeler gets *just one* loss query output, which could be distorted by noise. Surprisingly, we show that even with just this single query (and no access to the private feature set or any side knowledge), for many common loss functions including general Bregman divergence-based losses and multiclass cross-entropy with common activation functions like sigmoid and softmax, our inference attack succeeds in exactly recovering *all* the labels. This is a stronger privacy violation than that postulated by *blatant non-privacy* (Dinur and Nissim, 2003), where the goal is to only reconstruct a good fraction of the true labels.

Our observations in this paper have important ramifications, for example, when used by an adversary to execute a privacy breach by learning labels associated with a sensitive dataset, or by an unscrupulous participant to an ML competition for learning the unknown test labels. Our results call to attention these vulnerabilities which might be currently under silent exploitation. Armed with this information, individuals and organizations, which vend these seemingly innocuous aggregate metrics from their models can grasp the potential scope of the resulting information leakage.

Overview of Our Results and Techniques. Our attacks rely on a mathematical notion of *codomain separability* of loss functions, which posits that the output of the loss function is *sufficiently* distinct on every possible labeling of the input datapoints (see Definition 1). We assume that the curator that returns the loss scores can add noise to these scores up to some (known) error bound τ. This noise can also be introduced as error when the scores are communicated over noisy channels or computed on low-precision machines[1]. As one would expect, separating the loss function outputs by more than 2τ is a necessary and sufficient condition for this label recovery to be accurate (see Proposition 1). While intuitive, this result gives a natural candidate for label inference, from just one loss query, using an exhaustive (exponential) local search (see LabelInf (1)).

Throughout this paper, we assume that the adversary knows the loss function, number of datapoints N and an upper bound τ on the resulting error (noise). We also assume that the loss is computed on all the datapoints. The main technical challenge here is to design prediction vectors for which a loss function demonstrates the required codomain separability to handle arbitrary noise levels. Our key idea here is to use sets with distinct subset sums. Two simple examples of such sets of size n are $\{1, 2, 4, \ldots, 2^{n-1}\}$ and $\{\ln p_1, \ldots, \ln p_n\}$, where $p_1, \ldots, p_n$ are distinct primes. Sets like these are useful when characterizing the sufficient conditions under which the required codomain separability can be achieved. For example, in the binary classification setting with N datapoints, the following problem comes up often in our analysis: Construct $\theta = [\theta_1, \ldots, \theta_N]$ such that

$$\min_{\sigma_1, \sigma_2 \in \{0,1\}^N} \left| \sum_{i:\sigma_1(i)=1} g(\theta_i) - \sum_{j:\sigma_2(j)=1} g(\theta_j) \right| \geq b,$$

for some function g and bound b. To satisfy this inequality, it suffices to set θ such that the set $g(\theta) := \{g(\theta_i), \ldots, g(\theta_N)\}$ has all distinct subset sums. This is because the summation operators essentially filter out subsets of elements from the vector θ, and because $\sigma_1 \neq \sigma_2$ in the minima operator, these subsets must differ in at least one element. Now, to ensure that the minimum difference of the subset sums in $g(\theta)$ is at least b, one can solve for $g(\theta_i) = 2^i b$ or $g(\theta_i) = b \ln p_i$ (or using some other set with distinct subset sums) depending on actual form of g and the application.

We use these ideas and tools from analytical number theory to provide constructions of adversarial prediction vectors for broad classes of ML loss functions based on Bregman divergences (Section 3) and multiclass cross-entropy (Section 4), for both the unnoised and the noised setting. The analytical properties of squarefree integers also helps us to reduce the computation time needed by the adversary. In addition to the single query model where the adversary has to work with only one (noisy) loss function value, we also analyze extensions where the adversary has access to multiple (noisy) loss function values from different prediction vectors. This extension comes in handy as with sufficient queries the local computation time required at the adversary becomes polynomial. Additionally, to handle situations an actual ML model is required (and not just the prediction vector), we provide a construction of a feed-forward neural network, which can be used to carry out these label inference attacks while making them look benign (see Section 5). We also point out some caveats associated with our approaches on machines with finite floating point precision.

Defenses. Our focus in this paper is on characterizing the vulnerability of loss functions in leaking private information. Viewed from this perspective, our results establishes lower bounds on the amount of noise needed (as a function of precision, number of queries, etc) on releasing these loss functions for any *reasonable notion* of label privacy. A rigorous defense mechanism against our proposed attack would be to release the loss scores under on differential privacy (Dwork et al., 2006) with carefully calibrated noise that overcomes this lower bound. This will ascertain desired levels of plausible deniability on the labels recovered by an adversary.

We also highlight that in general, determining whether a loss function is codomain separable is co-NP-hard (Theorem 5). We establish this through a (polynomial time) Karp reduction from the *Almost Tautology* problem from Boolean satisfiability theory (see Appendix A). Based on standard consensus on the complexity of this class of problems, it is unlikely that there is a general polynomial time algorithm for robust label inference from loss functions (Arora and Barak, 2009).

[1] Our assumptions about the noise generation process ensure that our attacks succeed irrespective of the noise process used by the data curator. Knowledge about the noise distribution can be helpful though. For random noise, by first generating a bound on the noise using a tail bound, our techniques can be applied.

Related Work. Label inference attacks were first introduced in (Whitehill, 2018) for binary log-loss using a heuristic solution to a min-max optimization problem. This attack does not recover all the labels and works only in the unnoised setting. The noised setting for binary log-loss was recently studied by (Aggarwal et al., 2021). While their approach was not formalized using codomain separability, their construction also used the idea of making the loss function outputs distinct for each labeling of the dataset using distinct subset sums. However, their algorithm runs in exponential time and works only in the unnoised setting for the multiclass case. Furthermore, they also restrict only to cross-entropy loss. Our paper not only subsumes these results, but also significantly extends them by showing that most commonly used loss functions in ML applications are vulnerable to leaking private information about the ground truth labels. Moreover, we provide single query sub-exponential time and multi-query polynomial time attacks that can be carried out through benign looking ML models and settle the computational complexity of robust label inference from arbitrary loss functions.[2].

2 REDUCING ROBUST LABEL INFERENCE TO CODOMAIN SEPARABILITY

We begin our discussion by formally defining the notion of codomain separability and its connections to label inference in the noised as well as unnoised setting. Our objects of interest are functions whose domain is the Cartesian product of the space of all labelings (defined by the $\mathbb{Z}_K^N = \mathbb{Z}_K \times \cdots \times \mathbb{Z}_K$ (N times) $= \{0, \ldots, K-1\}^N$) and an arbitrary set $\Theta \subseteq \mathbb{R}^N$. Here, $K \geq 2$ represents the number of label classes. This formulation captures the common scenario in machine learning, where we evaluate a loss function using the true labeling in $\mathbb{Z}_K^N$ for N datapoints based on a (prediction) vector in $\mathbb{R}^N$ generated by an ML model. Θ is the space of prediction vectors, and for $\theta = [\theta_1, \ldots, \theta_N] \in \Theta$, the value of θ_i encodes the label prediction for the ith datapoint. We work with different loss functions that place different restrictions on Θ. All missing details from this section are presented in Appendix B.

Codomain Separability. Informally, we call a function codomain separable if there exists some vector $\theta \in \Theta$ such that the function output is distinct on each $\mathbb{Z}_K^N$ (keeping θ fixed). Thus, when θ is known, this one-one correspondence between the function's output and the labelings in $\mathbb{Z}_K^N$ can be exploited to exactly recover all the labels from just observing the output.

Definition 1 (τ-codomain separability)**.** *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$ be a function. For $\theta \in \Theta$, define $\Lambda_\theta(f) := \min_{\sigma_1, \sigma_2 \in \mathbb{Z}_K^N} |f(\sigma_1, \theta) - f(\sigma_2, \theta)|$ to be the minimum difference in the function output keeping θ fixed. For a fixed $\tau > 0$, we say that f admits* τ-codomain separability *using θ if $\Lambda_\theta(f) \geq \tau$. In particular, we say that f admits* 0^+-codomain separability *using θ if there exists any $\tau > 0$ such that $\Lambda_\theta(f) \geq \tau$.*

Compared to τ-codomain separability for $\tau > 0$, the 0^+-codomain separability is weaker as it only requires $\Lambda_\theta(f) > 0$. This condition is used for label inference in the unnoised case. As an example for τ-codomain separability, consider the function $f(\sigma, \theta) = \langle \sigma, \theta \rangle$ for $\sigma \in \{0, 1\}^N$. To demonstrate codomain separability, it suffices to set $\theta = [1, 2, 4, \ldots, 2^{N-1}]$ which makes f admit 1-codomain separability. Multiplying each entry in θ by τ will make f admit τ-codomain separability for any $\tau > 0$. In upcoming sections, we discuss our constructions that make many popular loss functions separable. In Appendix F, we present some function classes that are provably not τ-codomain separable.

Robust Label Inference. The goal with robust label inference is to recover the true labeling (in $\mathbb{Z}_K^N$) upon observing only the loss function output, even if noised. Observe that the results trivially hold for the unnoised case if we can handle arbitrary noise levels.

Definition 2 (τ-Robust Label Inference)**.** *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$ be a function, and $\sigma^\star \in \mathbb{Z}_K^N$ be the (unknown) true labeling. For a given $\tau > 0$, we say that f admits τ-robust label inference if there exists $\theta \in \Theta$ and an algorithm (Turing machine) $\mathcal{A}$ that can recover $\sigma^\star$ given any $\ell \in \mathbb{R}$ where $|f(\sigma^\star, \theta) - \ell| < \tau$,* i.e., *for all $\sigma^\star \in \mathbb{Z}_K^N$, we have $\mathcal{A}(\theta, N, \ell) = \sigma^\star$.*

We reiterate an important point to note here that τ-robust label inference requires perfect reconstruction of $\sigma^\star$, which is a *stronger* notion than that required by notions like *blatant non-privacy* (Dinur and Nissim, 2003), where the goal is to only reconstruct a good fraction of $\sigma^\star$. Also, while the above definition is based on a single query, we later relax this requirement to study robust label inference under a multi-query model.

The following proposition formally establishes the connection between the above definitions of codomain separability and robust label inference.

Proposition 1. *For any $\tau > 0$, the function f admits τ-robust label inference using $\theta \in \Theta$ iff $\Lambda_\theta(f) \geq 2\tau$.*

Suppose the adversary picks $\theta \in \Theta$ based on Definition 1 and gets back the noisy loss function value ℓ from the curator (server). Proposition 1 then allows for a natural label inference algorithm (which we call LabelInf)

[2] Our approach is also reminiscent of similar concepts used in information theory, e.g., coding schemes based on Sidon sequences (O'Bryant, 2004) and Golomb rulers (Robinson and Bernstein, 1967), where the goal is to have a high minimum distance between the codewords.

which iterates over all possible labelings to recover the one which is closest to the observed loss score:

$$\textsc{LabelInf}: \ \sigma^* \leftarrow \arg\min_{\sigma \in \mathbb{Z}_K^N} |f(\sigma, \theta) - \ell| \qquad (1)$$

A special case of LabelInf is the unnoised setting, wherein $\ell = f(\sigma^\star, \theta)$. In that case, it suffices to design a vector $\theta \in \Theta$ with respect to which f admits 0^+-codomain separability and τ plays no role.

While intuitive, an important feature about the approach outlined in LabelInf (1) is that it makes just one call to the server to retrieve the (loss) function f evaluated at a single θ, but still reconstructs the entire private vector. However, the exponential time exhaustive search over the space of all labelings makes it impractical. We optimize for this runtime to sub-exponential time (for single query) and polytime time (using multiple-queries) in Section 3.

Role of Arithmetic Precision. Our label inference attacks use number theoretic constructions with large integers and products of primes, which can render these attacks impractical to run (within a single query) on limited floating-point precision machines. We begin by observing that Definition 1 does not take into account fixed arithmetic precision, which has an effect on separability by placing a bound on the resolution. For example, even if $f(\sigma_1, \theta) \neq f(\sigma_2, \theta)$, this difference may not be observable with only ϕ bits of precision,. We extend the notion of codomain separability in the finite precision model in Appendix B.1, and present multi-query label inference attacks to recover all labels within fixed precision in Sections 3 and 6. For simplicity, we focus on inference attacks under arbitrary precision arithmetic in the main body of this paper.

3 LINEAR-DECOMPOSABILITY AND SUB-EXPONENTIAL TIME LABEL INFERENCE

Our main focus in this paper is on an important class of (binary) loss functions, which we refer to as *linearly-decomposable*. These functions can be expressed as a sum of two terms: one dependent on the true labeling σ, and the other only on the prediction vector θ. As we will see, this decomposition allows for an efficient construction of prediction vectors for robust label inference from such functions. We present only our main ideas here and defer all missing details to Appendix C.

Definition 3. *Let $g : [0,1] \to \mathbb{R}$ be some deterministic function. We say that a binary loss function $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ is linearly-decomposable if there exists some invertible function $g : [0,1] \to \mathbb{R}$ and some function $h : [0,1]^N \to \mathbb{R}$, such that:*

$$f(\sigma, \theta) = h(\theta) + \sum_{i=1}^{N} \sigma_i g(\theta_i) = h(\theta) + \sum_{i:\sigma_i = 1} g(\theta_i). \quad (2)$$

This class of functions includes many commonly used loss functions in the ML literature. For example, all Bregman divergence-based binary loss functions satisfy the linear-decomposability property.

Lemma 1. *Let $F : [0,1] \times [0,1] \to \mathbb{R}$ be a strongly convex function and $D_F(p,q) = F(p) - F(q) - \langle \nabla F(q), p - q \rangle$ be the Bregman divergence associated with F. Then, the corresponding loss function, defined as $f_F(\sigma, \theta) = \frac{1}{N} \sum_{i=1}^{N} D_F([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i])$, is linearly-decomposable.*

Unlike the distance metrics for probability distributions, Bregman divergences does not require its inputs to be necessarily distributions (Bregman, 1967). To use these divergences as loss functions, we directly compare the outputs of the ML model with the point distribution from the ground truth labels, as we do in the definition of $f_F(\sigma, \theta)$ (similar to (Liu and Belkin, 2016)).

We also focus on a special class of linearly-decomposable functions for which the linear split is based purely on the labels: one term corresponding to datapoints with true label 0, and the other for datapoints with true label 1. More formally, if $g : [0,1] \to \mathbb{R}$ is some deterministic function, then we say that a binary loss function $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ is g-linearly-decomposable if it can be expressed as follows:

$$f(\sigma, \theta) = \frac{1}{N} \left(\sum_{i:\sigma_i = 1} g(\theta_i) + \sum_{i:\sigma_i = 0} g(1 - \theta_i) \right). \quad (3)$$

In many cases, as we will see, functions in this subclass are easier to analyze for codomain separability. Observe that the KL-divergence loss (which reduces to binary cross-entropy or log-loss) is of this form, using $g(\theta_i) = -\ln \theta_i$. Some other common examples of g-linearly-decomposable loss functions include the (i) Itakura-Saito divergence based loss, which uses $g(\theta_i) = 1/\theta_i + \ln \theta_i - 2$; (ii) Squared Euclidean loss, which can be expressed using $g(\theta_i) = (1 - \theta_i)^2$; and, (iii) norm-like loss, which uses $g(\theta_i) = 1 + (\alpha - 1)\theta_i^\alpha - \alpha\theta_i^{\alpha-1} + (\alpha - 1)(1 - \theta_i)^\alpha$ for some $\alpha \geq 2$. We provide detailed constructions for robust label inference from these particular loss functions in Appendix C.

3.1 Establishing Codomain Separability

As argued in Section 2, the first step for label inference is to design prediction vectors using which the loss functions are sufficiently codomain-separable. We provide

two different constructions of such prediction vectors for linearly-decomposable functions. Each construction uses a different set with distinct subset sums to ensure that the loss scores are in 1-1 correspondence with the set of all possible labelings. The first construction uses powers of 2, which follows naturally from the requirement of the 2τ separation needed for τ-robust label inference (as in Proposition 1). We analyze multiple loss functions using this construction. Our second construction uses a set consisting of (log) primes, which enables us to perform robust label inference in sub-exponential time using results from number theory. We discuss these constructions in detail below.

Construction 1: As mentioned above, our first construction is based on the fact that the set $S_m = \{1, 2, 4, \ldots, 2^m\}$ has distinct subset sums. To see this, observe that each subset sum in S_m is an integer whose binary representation (in reverse) is given by the *bits* defined by the indices of elements contained in that subset. Moreover, since the difference between any two integers that can be represented this way is one, it also holds that the minimum subset sum difference is 1. Scaling each element of S_m also scales the minimum difference as needed. The theorem below states our main result from this construction.

Theorem 1. *Let $g : [0,1] \to \mathbb{R}$ be some deterministic function and $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be a loss function that is g-linearly-decomposable. Then, for any $\tau > 0$, the function f is 2τ-codomain separable if there exists $\theta \in (0,1)^N$ so that $g(\theta_i) - g(1-\theta_i) > 2^i N\tau$ for all $i \in [N]$. If $\tau = 0$, then setting $g(\theta_i) - g(1-\theta_i) > 0$ for all $i \in [N]$ suffices for 0^+-codomain separability.*

Based on this theorem, the prediction vectors can be constructed as follows: (1) Compute $x^*(y)$ as the solution to $g(x) - g(1-x) = y$; (2) If $x^*(y)$ exists, then set $\theta_i = x^*(2^i N\tau)$ for all $i \in [N]$. This vector θ can now be used for τ-robust label inference from f using LabelInf (see Section 6 for our empirical analysis using this construction). The following corollaries follow for specific loss functions such as Itakura-Saito divergence loss, squared Euclidean, and norm-like divergence losses (see detailed proofs in Appendix C). For simplicity, we discuss only the unnoised case in Corollary 2, for which it suffices to demonstrate 0^+-codomain separability.

Corollary 1. *The Itakura-Saito divergence loss is 2τ-codomain separable with $\theta_i = (1 + 3^{2^i N\tau})^{-1}$.*

Corollary 2. *The squared Euclidean loss is 0^+-codomain separable using $\theta_i = (1/2)\,(1 - \ln(p_i)/N)$, where p_i is the ith prime number for $i \in [N]$. The norm-like divergence loss for $\alpha \geq 2$ is 0^+-codomain separable using θ that satisfies $(1-\theta_i)^{\alpha-1} - \theta_i^{\alpha-1} = (\ln p_i)/(N\alpha)$.*

Construction 2: Our main motivation for this second construction is to infer all labels within sub-exponential time. Starting from Definition 2, we can ensure $\Lambda_\theta(f) \geq 2\tau$ (as is needed for τ-robust label inference from Proposition 1) by setting $g(\theta_i) = 3P\tau \ln p_i$ (if possible to do so within the domain of g), where p_i is the i^{th} prime number and $P = \prod_{i=1}^N p_i$. This particular choice of $g(\theta_i)$ ensures that each subset sum in the set $g(\theta)$ corresponds to the logarithm of a unique integer (using its prime factorization), and leads to the desired codomain separation as follows.

Theorem 2. *Let $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be a loss function that is linearly-decomposable (Definition 2). Let p_i is the i^{th} prime number and $P = \prod_{i=1}^N p_i$, is the product of the first N primes. Then, for any $\tau > 0$, setting $g(\theta_i) = 3P\tau \ln p_i$ for loss functions in Equation 2 ensures that $\Lambda_\theta(f) \geq 2\tau$. If $\tau = 0$, setting $g(\theta_i) = \ln p_i$ suffices for 0^+-codomain separability.*

We will now see how this choice of prime-based vector entries enable efficient label inference for any given τ.

3.2 Establishing Robust Label Inference

We discuss three approaches to recover the ground truth labels from the (perturbed) loss score obtained on prediction vectors we just designed. We distinguish each approach based on its runtime complexity and number of queries made to the server.

Exponential Time Single-Query Inference. The first approach is discussed in LabelInf (1), which iterates over all 2^N possible labelings to find the one that is closest to the observed loss score. Both constructions 1 and 2 ensure that this algorithm always returns the true labeling within a single query to the server.

Sub-Exponential Time Single-Query Inference. To avoid the exhaustive search as above, we first substitute the prediction vector from Construction 2 in the expression for the loss function to obtain $f(\sigma, \theta) = h(\theta) + 3P\tau \ln\left(\prod_{i:\sigma_i=1} p_i\right)$. Now, we know that due to codomain separation, the labeling that minimizes the distance between the observed loss ℓ and the true loss $f(\sigma, \theta)$ above is the true labeling σ^*:

$$\sigma^* = \arg\min_{\sigma\in\{0,1\}^N} \left| \ell - \left(h(\theta) + 3P\tau \ln\left(\prod_{i:\sigma_i=1} p_i \right) \right) \right|$$
$$= \arg\min_{\sigma\in\{0,1\}^N} \left| \exp\left(\frac{\ell - h(\theta)}{3P\tau} \right) - \left(\prod_{i:\sigma_i=1} p_i \right) \right|.$$

To obtain σ^* from the equation above without using an exhaustive search over $\{0,1\}^N$, we observe that the expression inside the argmin essentially seeks a *square-free* integer closest to some known real quantity. An integer is said to be squarefree if it has no repeated prime factors. This can be checked using the Booker-Hiary-Keating algorithm from (Booker et al., 2015) in

sub-exponential time. This algorithm uses the explicit formula for Dirichlet L-functions and a conditional on the Generalized Reimann Hypothesis for proving that a given integer is square-free with little or no knowledge of its factorization. Once such an integer is obtained, its prime factors are in 1-1 correspondence with the indices in σ^* that have label 1.

Algorithm 1: CloSqFree(x, m)

If $x \leq 2$ or $m = 2$, then **return** 2.
Let p_i be the i^{th} prime number. If $p_1 \cdots p_m \leq x$, then **return** $p_1 \cdots p_m$.
for $k = 0, 1, 2, \ldots, \lfloor x \rfloor - 1$ **do**
    If $\lfloor x \rfloor - k \in$ SqFree(m), then **return** $\lfloor x \rfloor - k$.
    If $\lceil x \rceil + k \in$ SqFree(m), then **return** $\lceil x \rceil + k$.

The problem of label inference on linearly-decomposable loss functions is, thus, reduced to the calculation of the nearest square-free integer to a given real number. More concretely, the expression for recovering the true labeling can be written as follows:

$$\sigma^* = \left\{ i : p_i \text{ divides } \textsc{CloSqFree}\left(e^{\frac{\ell - h(\theta)}{3P\tau}}, N\right) \right\},$$

where $\textsc{CloSqFree}(x, m) = \arg\min_{y \in \text{SqFree}(m)} |x - y|$ denotes the closest squarefree integer to x. The notation SqFree(m) denotes the set of all square free integers whose largest prime factor is at most the m^{th} prime number p_m. We outline an optimized version of CloSqFree in Algorithm 1, which runs in $O(N \exp\left((\ln N)^{O(1)}\right))$ time.

Multi-Query Polynomial Time Attacks. Loss functions that are g-linearly-decomposable also allow for an efficient multi-query label inference algorithm. In particular, we could have a trade-off between the ability to perform multiple queries with faster computation times for solving the optimization problem in LabelInf (1). To see this, observe that setting $\theta_i = 1/2$ gives $g(\theta_i) - g(1 - \theta_i) = 0$. Thus, if we want to infer the first $M < N$ labels in a single query, we can set $\theta = [\theta_1, \ldots, \theta_M, 1/2, \ldots, 1/2]$, where $\theta_1, \ldots, \theta_M$ are produced according to either Constructions 1 or 2. Using this θ will ensure that if $f(\sigma_1, \theta) = f(\sigma_2, \theta)$, then $\sigma_1[:M] = \sigma_2[:M]$. After recovering the first M labels, we can recover the next M labels using $\theta = [1/2, \ldots, 1/2, \theta_{M+1}, \ldots, \theta_{2M}, 1/2, \ldots]$, and so on.

Observe that an $\lceil N/M \rceil$-query algorithm for robust label inference will require $O(N2^M/M)$ local computations by the adversary (using LabelInf (1) in each query). Thus, while the single query case required $O(2^N)$ computations, any multi-query algorithm using $M = O(\log N)$ requires only $O(\text{poly}(N))$ time.

4 MULTICLASS CROSS-ENTROPY LOSS

We now show that the ideas of codomain separability also extend to the particular case of multiclass cross-entropy loss and its variants. We provide an overview of our results and defer all missing details to Appendix D.

Multiclass Cross-Entropy Loss. We first recall the definition of multiclass cross-entropy. We assume $K \geq 2$ classes, and let N and $\mathbb{Z}_K$ denote the number of datapoints and the set of label classes, respectively. The K-ary cross-entropy loss on θ with respect to a labeling $\sigma \in \mathbb{Z}_K^N$ is defined as follows:

$$\text{CELoss}(\sigma, \theta) := \frac{-1}{N} \sum_{i=1}^{N} \sum_{k=0}^{K-1} \left([\sigma_i = k] \cdot \ln \theta_{i,k} \right), \quad (4)$$

where $[\sigma_i = k] = 1$ if $\sigma_i = k$ and 0, otherwise. Here, $\theta \in \Theta = [0, 1]^{N \times K}$ is a matrix of prediction probabilities, where the ith row is the vector of prediction probabilities $\theta_{i,0}, \ldots, \theta_{i,K-1}$ (with $\sum_{k \in \mathbb{Z}_K} \theta_{i,k} = 1$) for the ith datapoint on classes $0, \ldots, K-1$ respectively.

With this definition, we can now describe our construction of a matrix $\theta \in [0, 1]^{N \times K}$ that makes CELoss function 2τ-codomain separable for any $\tau > 0$ (i.e., $\Lambda_\theta(\text{CELoss}) \geq 2\tau$). At a high level, we obtain the required codomain separability for CELoss by splitting the loss into label dependent and independent terms, and designing the entries in the matrix θ in such a way that the expression reduces to a distinct integer in some set. This calibration allows us to control the minimum difference in the output of the cross-entropy loss on different labelings, which we can scale to the desired amount ($\geq 2\tau$) easily. The following theorem summarizes our construction.

Theorem 3. *Let $\tau > 0$. Define matrices $\vartheta, \theta \in \mathbb{R}^{N \times K}$ such that $\vartheta_{n,k} = 3^{\left(2^{(n-1)K+k} N\tau\right)}$ and $\theta_{n,k} = \vartheta_{n,k} / \sum_{k=1}^{K} \vartheta_{n,k}$. Then, it holds that* CELoss *is 2τ-codomain separable using θ. If $\tau = 0$, then using $\vartheta_{n,k} = 3^{\left(2^{(n-1)K+k}\right)}$ ensures 0^+-codomain separability.*

Using Theorem 3 and Proposition 1 implies that for these cross-entropy loss functions, the approach outlined in LabelInf (1) succeeds in recovering all the labels when the loss scores are noised by less than τ in magnitude. We bring to the reader's attention the doubly-exponential nature of the entries used to construct the prediction vector in Theorem 3. This blowup is unfortunately unavoidable for constructing τ-codomain separability, even for the binary case (see (Aggarwal et al., 2021, Theorem 7)).

Extensions of Cross-Entropy Loss. Often in practice, when using the cross-entropy loss to assess the

performance of ML models (like CNNs), it is common to apply an activation function (such as softmax or sigmoid) before the cross-entropy loss calculation. For example, a common idea in multiclass classification is to apply the softmax function (to convert any sequence of real outputs into a probability distribution) as:

$$\frac{-1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\Big([\sigma_i = k]\cdot \ln \textsc{softmax}(\theta_{i,k})\Big), \qquad (5)$$

where $\textsc{softmax}(\theta_{i,k}) = \exp(\theta_{i,k})/\sum_{j=1}^{K}\exp(\theta_{i,j})$. To extend Theorem 3 to this setting, we do the following: (1) Let $\theta \in [0,1]^{N\times K}$ be the matrix from Theorem 3. (2) For each $i \in [N]$, solve the following (fully specified) system of equations for $\theta'_{i,k}$'s: for all $k \in \mathbb{Z}_K, i \in [N]$:

$$\exp\left(\theta'_{i,k}\right) / \sum_{j=1}^{K}\exp\left(\theta'_{i,j}\right) = \theta_{i,k}.$$

Once θ' is obtained, since $\textsc{softmax}(\theta'_{i,k}) = \theta_{i,k}$, from Theorem 3, we get that softmax cross-entropy loss is 2τ-codomain separable using θ'. A similar argument also works for Sigmoid cross entropy loss (Appendix D).

5 ONE MODEL TO INFER THEM ALL

The results in previous sections highlight how prediction vectors (θ's) could be generated that succeed with τ-robust label inference. This raises an interesting question of whether the prediction vectors utilized in these label inference attacks can actually be an output of a non-trivial benign looking ML model? We answer this question in the affirmative in this section.

Setup. Assume a classification problem on K classes. We will design a multi-layer feed-forward neural network MutNet using following specifications. (1) The input to MutNet is a vector $\mathbf{v} \in \mathbb{R}^{d_1}$ with true label $\sigma_{\mathbf{v}} \in \mathbb{Z}_K$. (2) The output of MutNet is a vector in $(0,1)^{d_2}$ that represents an encoding of the prediction. Let $f : \mathbb{Z}_K \times (0,1)^{d_2} \to \mathbb{R}$ be a loss function. The goal of the network is to ensure that on any input $\mathbf{v}$ the network generates an output $\mathbf{u}_{m+1}$ such that f admits 2τ-codomain separability using $\mathbf{u}_{m+1}$. Consequently, for any input $\mathbf{v}$, given a noisy value of $f(\sigma_{\mathbf{v}}, \mathbf{u}_{m+1})$, an adversary can use LabelInf to infer $\sigma_{\mathbf{v}}$.

Our Mutator Network. We construct a 2-layer network $\textsc{MutNet}_{\mathbf{x}} : \mathbb{R}^{d_1} \to (0,1)^{d_2}$ that can convert any real vector $\mathbf{v} \in \mathbb{R}^{d_1}$ into any desired fixed vector $\mathbf{x} \in (0,1)^{d_2}$. The transformations we use in this network are the ReLU and Sigmoid activation functions: one layer of the former and one layer of the latter. Let $M_1 \in \mathbb{R}^{d_1\times d_1}$ and $M_2 \in \mathbb{R}^{d_1\times d_2}$ be matrices such that all entries in M_1 are negative (the entries in M_2 can be arbitrary). Let $\mathbf{x}'$ be a vector such that $x'_i = \ln x_i/(1-x_i)$. Then, for an input vector $\mathbf{u}_1 = \mathbf{v}$, the transformations in $\textsc{MutNet}_{\mathbf{x}}$ are as follows: $\mathbf{u}_2 = \textsc{ReLU}\left(\mathbf{v}^\top M_1\right)$, $\mathbf{u}_3 = \textsc{Sigmoid}\left(\mathbf{u}_2^\top M_2 + \mathbf{x}'\right)$, where ReLU and Sigmoid are applied element-wise on their input vectors. Effectively, this construction inhibits the propagation of $\mathbf{v}$ by outputting the same vector $\mathbf{u}_3 = \mathbf{x}$ always. By setting $\mathbf{x}$ to the desired (prediction vector) θ for 2τ-codomain separability on f, we get that $|f(\sigma_{\mathbf{v}}, \textsc{MutNet}_\theta(\mathbf{v})) - \ell| \le \tau$ (Theorem 6).

Remark 4. *We note that any neural network model can be modified to carry out our attack, as an adversary can replace the top layer of any neural network model with the above construction. This highlights the versatility of our attack.*

6 EMPIRICAL ANALYSIS

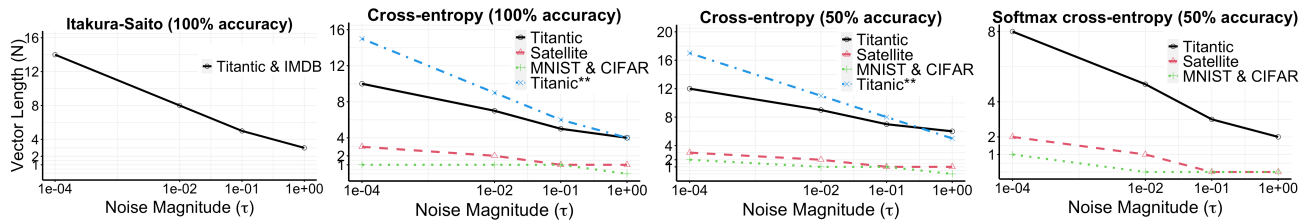
We now present an empirical evaluation of our label inference attacks. We analyze the following datasets:

- **Titanic** (tit): a binary classification dataset (2201 rows) on the survival status of passengers.
- **IMDB** (imd): a binary classification dataset (25000 rows) on the movie reviews.
- **Satellite** (sat): a six-class classification dataset (6430 rows) on the satellite images of soil.
- **MNIST** (mni): a ten-class classification dataset (70000 rows) on handwritten digits.
- **CIFAR** (cif): a ten-class classification dataset (60000 rows) on color images.

As our attacks construct prediction vectors that are independent of the dataset contents, we ignore the dataset features in our experiments, but all the labels of the dataset are considered for the attack[3]. We consider four common loss functions arising from Sections 3 and 4, two of which are multiclass losses (multiclass cross-entropy (CELoss (4)) and softmax cross-entropy (5)), and the other two are binary losses (binary Itakura-Saito (ISLoss (8)) and sigmoid cross-entropy (14)). As a baseline, for the binary labeled dataset (Titanic) with the (plain) cross-entropy loss, we implemented the label inference attack of (Aggarwal et al., 2021). We also present additional experimental results in Appendix G.

We start with the distinction between our experiments and the approach outlined in LabelInf, which is presented in the arbitrary precision model. For CELoss, simulating LabelInf on a finite precision machine must be able to differentiate $\min_{i,k}\theta_{i,k}$ from 0 (or else

[3] All experiments are run on a 64-bit machine with 2.6GHz 6-Core processor, using the standard IEEE-754 double precision format. For reproducibility, the code is included as part of the supplementary material.



(a) (b) (c) (d)

Figure 1: Results for single-query label inference. The Y-axis in Figures (a) and (b) represents the number of datapoints that can be recovered with 100% accuracy, while for Figures (c) and (d), it represents the number of datapoints that can be recovered at 50% accuracy, i.e., we recover at least this length vector accurately in at least 50% of the 1000 runs. The max computation time per inference attack is roughly 10 seconds.

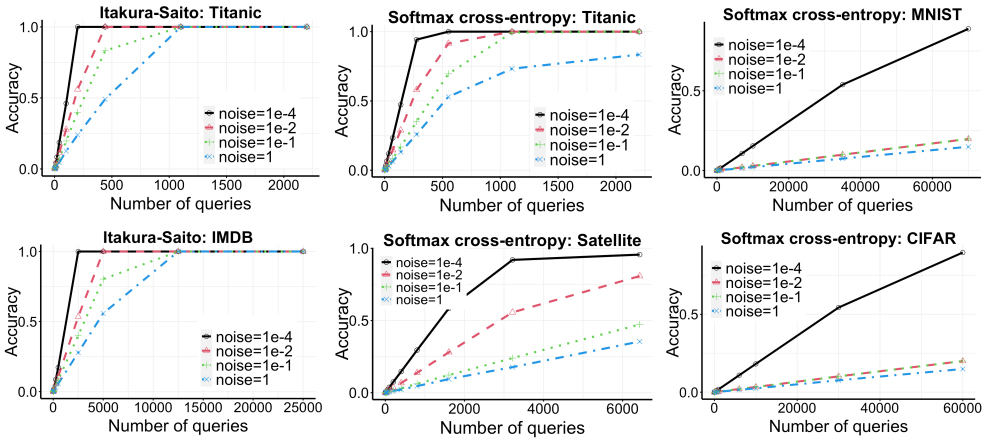


Figure 2: Label reconstruction accuracy with the multi-query label inference attack. As discussed in the text, for a given loss-function, we expect similar plots for any two datasets with same number of classes, e.g., Itakura-Saito for Titanic and IMDB, Softmax cross-entropy for MNIST and CIFAR.

the label inference will be ambiguous). A rough analysis (from Theorem 3) gives that $\phi = \Omega(2^{NK}N\tau)$ bits are required to make this distinction. This bound hides constant factors, but gives an idea of how arithmetic precision plays a role in our experiments.

Figure 1 shows the number of datapoints N recovered by LabelInf as we increase the noise for ISLoss, CELoss, and softmax cross-entropy loss. We sample 1000 random sets of labels each of length N from the dataset here. At error magnitude $\tau = 1$, the noise is comparable to the actual loss function values computed. We measure accuracy as the percentage of labels correctly inferred out of N. Figure 1(a) shows that the number of datapoints (N) for which 100% label inference accuracy is achieved with ISLoss as we vary noise magnitude. As expected at lower noise magnitude the attacks can extract more labels correctly and this quantity decreases as the noise magnitude (τ) increases. Note that as mentioned, our attacks are independent of the dataset feature set, therefore the recovery performance is same on both Titanic and IMDB, as they are both binary labeled. Figure 1(b) shows the number of datapoints (N) for which 100% label inference accuracy is achieved with CELoss. Here, the results differ (except for MNIST and CIFAR) because the datasets have different number of classes K. At the same noise magnitude, the accuracy also drops as the number of classes (K) increases from 2 (in Titanic) to 10 (in MNIST and CIFAR), which is also expected. These happen because of the dependence on number of datapoints and number of classes in our prediction vector construction (Theorem 3), which given fixed machine precision runs into representation issues. We note that the construction from (Aggarwal et al., 2021) on the Titanic dataset (the only case where it is applicable), works slightly better (especially at lower τ values) than ours due to a small difference in the exponent: it holds that $\min_{i\in[N]}(-\ln\theta_i) = \Omega(2^N N\tau)$ in their paper, but $\Omega(2^{2N}N\tau)$ when using Theorem 3.

Figures 1(c)-(d) show the number of datapoints on which LabelInf achieves at least 50% accuracy for CELoss and softmax cross-entropy losses respectively. We notice this number is smaller for softmax cross-entropy loss, which is also not surprising. Through a similar argument as that above for the number of bits required for cross-entropy loss, one can show that computing the softmax cross-entropy loss will require an additional $\Omega(NK + \ln(N\tau))$ bits (see the discussion

in Appendix G for details). This additional requirement further constraints the number of labels that can be recovered with softmax cross-entropy loss.

We also examine a multi-query label inference algorithm in Figure 2. For these plots, we simulated LabelInf on $M < N$ datapoints at a time (instead of all), to obtain a total of $\lceil N/M \rceil$ queries. The idea is to use Figure 1 to determine the maximum number of labels that can be correctly inferred in a single query for a given noise level, and then perform label inference on only those many datapoints at a time. As expected, we observed that the accuracy increases with the number of queries: for ISLoss on Titanic, we achieved 100% accuracy using $M \geq 220$ with $\tau = 0.0001$, and $M \geq 1100$ with $\tau = 1$. The accuracy is again lower for the softmax case again due to reasons mentioned above. Additional experimental results on CELoss and sigmoid cross-entropy losses are included in Appendix G.

7 CONCLUDING REMARKS

In this paper, we demonstrated how a large class of common ML loss functions can be exploited to recover the unknown test labels. Our attacks, based on tools from analytical number theory, succeed with provable guarantees, even when provided with noisy loss function values. Our investigation also highlights the role of number of queries and arithmetic precision. We also demonstrate how our attack could be carried out using a simple augmentation to any neural network model, making it look benign.

Finally, we end up discussing some of the practical limitations of our attacks, removing which could strengthen the results we present in this paper.

- Our attacks require the knowledge of the loss function on the adversary's part (which is generally known in practice) as well as an upper bound on noise being added. The latter assumption can, however, be relaxed in a multi-query setup where the adversary uses a doubling trick for guess on the noise magnitude.
- Our attacks assume that there is an arbitrary, but fixed ordering of the test dataset across queries.
- Our attacks, as presented, are more interesting in the public leaderboard setting. However, even in the private leaderboard setting, the adversary will still be able to recover the set of test labels using our algorithm, but will not able to associate them to individual test datapoints. Nonetheless, knowing the vector of test labels can give the adversary sensitive summary statistics about the individuals whose information is present in the test set (e.g., fraction of medical patients that have disease X). Note that further such summary statistics can be combined with other auxiliary data sources. Just for illustration, if we know that John is taking cancer treatment at a hospital (but do not which cancer), and if that hospital is running a ML competition for predicting lung cancer on their patients data, and our attack say recovers $[1, 0, \ldots, 0, \ldots, 0]$ as labels then we immediately know that John has lung cancer (which will be a privacy breach).

References

`https://www.cs.toronto.edu/ kriz/cifar.html`. Last accessed in October, 2021.

`http://www.image-net.org/challenges/LSVRC/`. Last accessed in October, 2021.

`https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`. Last accessed in October, 2021.

`https://www.kaggle.com`. Last accessed in October, 2021.

`https://www.kdd.org/kdd-cup`. Last accessed in October, 2021.

`http://yann.lecun.com/exdb/mnist`. Last accessed in October, 2021.

`https://www.openml.org/d/182`. Last accessed in October, 2021.

`https://www.openml.org/d/40704`. Last accessed in October, 2021.

A. Aggarwal, S. Kasiviswanathan, Z. Xu, O. Feyisetan, and N. Teissier. Label inference attacks from log-loss scores. In *International Conference on Machine Learning*. PMLR, 2021.

S. Arora and B. Barak. *Computational complexity: A Modern Approach*. Cambridge University Press, 2009.

A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1-3):80–88, 1982.

A. R. Booker, G. A. Hiary, and J. P. Keating. Detecting squarefree numbers. *Duke Mathematical Journal*, 164(2), Feb 2015. ISSN 0012-7094. doi: 10.1215/00127094-2856619. URL `http://dx.doi.org/10.1215/00127094-2856619`.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

R. P. Brent and P. Zimmermann. *Modern computer arithmetic*, volume 18. Cambridge University Press, 2010.

I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

D. E. Knuth. *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 2014.

C. Liu and M. Belkin. Clustering with bregman divergences: An asymptotic analysis. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2351–2359. Citeseer, 2016.

K. O'Bryant. A complete annotated bibliography of work related to sidon sequences. *arXiv preprint math/0407117*, 2004.

C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 255–260, 1982.

J. p. Robinson and A. Bernstein. A class of binary recurrent codes with limited error propagation. *IEEE Transactions on Information Theory*, 13(1):106–113, 1967.

J. Whitehill. Climbing the kaggle leaderboard by exploiting the log-loss oracle. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Supplementary Material: Reconstructing Test Labels From Noisy Loss Functions

A COMPUTATIONAL HARDNESS OF CODOMAIN SEPARABILITY

We show that that determining the codomain separability is co-NP-hard (see (Arora and Barak, 2009) for a definition of co-NP-hard). We establish the result for the weaker notion of 0^+-codomain separability (Definition 1) and restrict ourselves to functions of the form $f : \{0,1\}^N \times \mathbb{Z}_+^N \to \mathbb{R}$, where the decision problem is to determine whether there exists a $\theta \in \mathbb{Z}_+^N$ such that f is 0^+-codomain separable using θ. We denote this decision problem by CODOMAIN-SEP. Note that this automatically implies that determining the τ-codomain separability of such functions is also co-NP-hard.

Let $\Phi(x_1, \ldots, x_N)$ be a Boolean formula over N variables in the 3-CNF form $\Phi(x_1, \ldots, x_N) = C_1 \wedge \cdots \wedge C_N$, where C_i are disjunctive clauses containing 3 literals each. We say that $\Phi(x_1, \ldots, x_N)$ is an *almost tautology* if there are at least $2^N - 1$ satisfying assignments for Φ. In other words, there is at most one assignment of the variables that makes $\Phi(x_1, \ldots, x_N)$ false. Define ALMOST-TAUTOLOGY to be the problem of determining if a given Boolean formula is an almost tautology.

Lemma 2. ALMOST-TAUTOLOGY *is co-NP-complete.*

Proof. We first show that ALMOST-TAUTOLOGY is in co-NP, i.e., any certificate that Φ is not an almost-tautology can be checked in polynomial time. To see this, observe that such a certificate must contain at least two distinct assignments for which Φ is not satisfied, which can be verified efficiently.

To show that ALMOST-TAUTOLOGY is co-NP Hard, there are two cases: either Φ is a tautology, or there is exactly one assignment that does not satisfy Φ. Deciding the former is co-NP-hard (Arora and Barak, 2009). For the latter, consider the logical negation $\neg\Phi(x_1, \ldots, x_N)$. If Φ is an almost tautology (but not a tautology), then there is exactly one satisfying assignment for $\neg\Phi$. This is the same as the Unique-SAT problem, in which we determine if a Boolean formula has a unique solution. This problem is also co-NP-hard (Papadimitriou and Yannakakis, 1982; Blass and Gurevich, 1982), and hence, ALMOST-TAUTOLOGY is co-NP-hard. □

We start with an arbitrary Boolean formula $\Phi(x_1, \ldots, x_N)$. For $\theta \in \mathbb{Z}_+^N$, define the following function:

$$f_\Phi(\sigma, \theta) = \hat{C}_1^{\theta_1} \cdots \hat{C}_N^{\theta_N} p_1^{\sigma_1} \cdots p_N^{\sigma_N},$$

where $p_1, \ldots, p_N \geq 5$ are distinct prime numbers, and $\hat{C}_i = C_i(\sigma)$ in the additive form (i.e., mapping all variables to $\{0,1\}^N$ by representing all negative literals $\bar{x}_i$ as $1 - x_i$, and converting all disjunctions to addition). For example, if $C_1 = (x_3 \vee \bar{x}_4 \vee x_6)$, then $\hat{C}_1 = C_1(\sigma) = \sigma_3 + (1 - \sigma_4) + \sigma_6$. Another example is as follows: assume $\Phi(x_1, x_2, x_3) = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$. Then, first repeat the third clause in Φ to make the number of clauses the same as the number of variables to obtain $\Phi(x_1, x_2, x_3) = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$. Now, the corresponding function can be written as follows:

$$f_\Phi(\sigma, \theta) = 5^{\sigma_1} 7^{\sigma_2} 11^{\sigma_3} (1 - \sigma_1 + \sigma_2 + \sigma_3)^{\theta_1} (1 + \sigma_1 + \sigma_2 - \sigma_3)^{\theta_2 + \theta_3}. \tag{6}$$

We now prove our hardness result for 0^+-codomain separability using this reduction.

Lemma 3. *For any $\theta \in \mathbb{Z}_+^N$ and distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$, it holds that $f(\sigma_1, \theta) \neq f(\sigma_2, \theta)$ if and only if at least one of σ_1 or σ_2 satisfies Φ. Here, we abuse the notation to represent the Booleans True and False by 1 and 0, respectively.*

Proof. Observe that for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$, we have $f(\sigma_1, \theta) = f(\sigma_2, \theta)$ only when there is some set of clauses $\hat{C}_i$ and $\hat{C}_j$ such that $\hat{C}_i(\sigma_1) = \hat{C}_i(\sigma_2) = 0$, i.e. they are unsatisfied by σ_1 and σ_2, respectively. This is

because the product of the primes satisfies $\prod_{i=1}^{N} p_i^{\sigma_1(i)} \neq \prod_{j=1}^{N} p_j^{\sigma_2(j)}$ (since both products differ in at least one prime – the one corresponding to the index of the element at which σ_1 and σ_2 differ). The lemma statement then follows from the contrapositive of this result. □

Theorem 5. CODOMAIN-SEP *is co-NP-hard.*

Proof. We prove this by demonstrating a Karp reduction from the ALMOST-TAUTOLOGY problem, which we showed is co-NP-complete in Lemma 2. Let $\Phi(x_1, \ldots, x_N)$ be an arbitrary Boolean formula and $f_\Phi(\sigma, \theta)$ be the corresponding function from (6). Now, for Φ to be an almost tautology, there can at most one unsatisfying solution: (1) if there are no unsatisfying solutions, then for any Boolean assignment of $x_1, \ldots, x_N$, all clauses in $\Phi(x_1, \ldots, x_N)$ must be satisfied and hence, from Lemma 3, the value of $f_\Phi(\sigma, \theta)$ must also be distinct for all σ, implying that f_Φ is 0^+-codomain separable; (2) if there is an unsatisfying assignment, then the value of $f_\Phi(\sigma, \theta)$ at this assignment must be zero, and it is non-zero at all other assignments. This also implies that f_Φ is 0^+-codomain separable. Lastly, let θ be a vector with respect to which f_Φ is 0^+-codomain separable. This immediately implies that $\Phi(x_1, \ldots, x_N)$ must be an almost tautology — if not, then either one of (1) or (2) must be false since there will be at least two unsatisfying assignments in this case, implying that the function f_Φ will be zero on at least two inputs. □

B MISSING DETAILS FROM SECTION 2

The following proposition states the connection between τ-robust label inference and τ-codomain separability. This connection, for the specific case of binary cross-entropy loss, was also noted by (Aggarwal et al., 2021).

Restatement of Proposition 1. *A function f admits τ-robust label inference using $\theta \in \Theta$ if and only if $\Lambda_\theta(f) \geq 2\tau$.*

Proof. We start with one direction and show that if we can do label inference (there exists algorithm $\mathcal{A}$ in Definition 2), then $\Lambda_\theta(f) \geq 2\tau$ must hold. We prove this by contradiction. The idea is to construct a score from which a unique labeling cannot be unambiguously derived. Without loss of generality, let σ_1, σ_2 be two distinct labelings for which $0 < f(\sigma_2, \theta) - f(\sigma_1, \theta) < 2\tau$. It follows that $f(\sigma_2, \theta) - \tau < f(\sigma_1, \theta) + \tau$. Now, let $\ell = (f(\sigma_1, \theta) + f(\sigma_2, \theta))/2$ and $x = \ell - f(\sigma_1, \theta)$. Clearly, $x < \tau$. Similarly, $f(\sigma_2, \theta) - \ell < \tau$. In other words, ℓ could be generated by a τ magnitude perturbation to both $f(\sigma_1, \theta)$ and $f(\sigma_2, \theta)$ (with $\sigma_1 \neq \sigma_2$). Therefore, there can exists no algorithm $\mathcal{A}$ that, given just ℓ, can recover whether the true label is σ_1 or σ_2 (i.e., no $\mathcal{A}$ can succeed with τ-robust label inference). This is a contradiction, therefore, $\Lambda_\theta(f) = \min_{\sigma_1, \sigma_2} |f(\sigma_2, \theta) - f(\sigma_1, \theta)| \geq 2\tau$.

For the other direction, let ℓ be as given in Definition 2 with $|f(\sigma^\star, \theta) - \ell| < \tau$. By triangle inequality it follows that if $\Lambda_\theta(f) \geq 2\tau$, then $|\ell - f(\sigma^\star, \theta)| < \min_{\sigma \in \mathbb{Z}_K^N \setminus \sigma^\star} |\ell - f(\sigma, \theta)|$ (i.e., addition of any noise less than τ in magnitude will maintain the invariant that the noised score is closest to the score on the true labeling). Hence, solving $\arg\min_{\sigma \in \mathbb{Z}_K^N} |f(\sigma, \theta) - \ell|$ will return the true label $\sigma^\star$. □

B.1 Separability in Arbitrary Precision vs. Finite Floating-Point Precision

One important consideration in our label inference attacks is the precision of arithmetic that is required at the adversary. In this context, there are two natural models of arithmetic computation: a) arbitrary precision and b) finite floating-point precision. Arbitrary precision arithmetic model allows precise arithmetic results even with very large numbers. In the floating-point precision model, the arithmetic is constrained by limited precision. An example of the floating-point precision model is the commonly used IEEE-754 double precision standard. Designing algorithms for standard arithmetic in both these models have been studied extensively (Knuth, 2014; Brent and Zimmermann, 2010). We refer to the arbitrary precision arithmetic model as APA and floating point arithmetic model with ϕ bits as $\mathsf{FPA}(\phi)$. For ease of discussion, we assume that in the $\mathsf{FPA}(\phi)$ model, we have 1 bit for sign, $(\phi - 1)/2$ bits for the exponent and $(\phi - 1)/2$ bits for the fractional part (mantissa). This assumption can be relaxed to accommodate $\phi_a > 0$ bits for the exponent and $\phi_b > 0$ bits for the fractional part where $\phi_a + \phi_b = \phi - 1$. Furthermore, for any loss function f in the APA model, we denote by f_ϕ the algorithm that computes f on a machine with an instruction set for performing computations within these ϕ bits of precision.

We begin by observing that Definition 1 does not take into account fixed arithmetic precision (i.e., deals with the case where we have arbitrary precision arithmetic). Finite floating-point precision has an effect on separability, as

bits of precision places a bound on the resolution. For example, even if $f(\sigma_1, \theta) \neq f(\sigma_2, \theta)$ in the APA model, with only ϕ bits of precision, this difference may not be observable. This leads to notion of separability in the FPA(ϕ) model.

Definition 4 (τ-codomain Separability in the FPA(ϕ) model)**.** *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$ be a function. Let f_ϕ be the representation of f in the* FPA(ϕ) *model. For $\theta \in \Theta$, define $\Lambda_\theta(f_\phi) := \min_{\sigma_1, \sigma_2 \in \mathbb{Z}_K^N} |f_\phi(\sigma_1, \theta) - f_\phi(\sigma_2, \theta)|$ to be the minimum difference in the function output keeping θ fixed. For a fixed $\tau > 0$, we say that f admits* τ-codomain separability *using θ in the* FPA(ϕ) *model if $\Lambda_\theta(f_\phi) \geq \tau$.*

In particular, we say that f admits 0^+-codomain separability *using θ in the* FPA(ϕ) *model if there exists any $\tau > 0$ such that $\Lambda_\theta(f_\phi) \geq \tau$.*

Let f_ϕ be the representation of f in the FPA(ϕ) model. Informally, representation in the FPA(ϕ) model implies computing f within the granularity defined by ϕ, and reporting underflow/overflow when the results are out of range. It is easy to show that if f_ϕ admits τ-codomain separability using θ, then f also admits τ-codomain separability in the APA model using θ (see Proposition 2). The other direction is trickier, but we can establish that if f admits τ-codomain separability in the APA model using θ, then f_ϕ admits τ-codomain separability using θ if $\phi \geq \max(2\log_2 \tau + 5, -2\log_2 \tau - 1)$ (see Proposition 3).

Proposition 2. *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$, $\theta \in \Theta$, and $\tau > 0$. Let f_ϕ be the representation of f in the* FPA(ϕ) *model. If f_ϕ admits τ-codomain separability in the* FPA(ϕ) *model (using θ) for some $\phi > 0$, then f also admits τ-codomain separability in the* APA *model using θ. Moreover, $f_{\phi'}$ also admits τ-codomain separability in the* FPA(ϕ') *model using θ for all $\phi' > \phi$.*

Proof. Given θ and the fact that f_ϕ admits τ-codomain separability using θ in the FPA(ϕ) model, denote

$$\Lambda_\theta(f_\phi) = \min_{\sigma_1, \sigma_2 \in \mathbb{Z}_K^N} |f_\phi(\sigma_1, \theta) - f_\phi(\sigma_2, \theta)| = b_{\frac{\phi-3}{2}} \cdots b_1 b_0 \,.\, b_{-1} b_{-2} \cdots b_{-\frac{\phi-1}{2}} \geq 2\tau,$$

where each bit $b_i \in \{0, 1\}$. Then, in the FPA($\phi + 1$) model, we can write (without loss of generality):

$$\begin{aligned}\Lambda_\theta(f_{\phi+1}) &= \begin{cases} 0 b_{\frac{\phi-3}{2}} \cdots b_1 b_0 \,.\, b_{-1} b_{-2} \cdots b_{-\frac{\phi-1}{2}} & \text{if } \phi \text{ is even,} \\ b_{\frac{\phi-3}{2}} \cdots b_1 b_0 \,.\, b_{-1} b_{-2} \cdots b_{-\frac{\phi-1}{2}} 0 & \text{if } \phi \text{ is odd} \end{cases} \\ &= b_{\frac{\phi-3}{2}} \cdots b_1 b_0 \,.\, b_{-1} b_{-2} \cdots b_{-\frac{\phi-1}{2}} \geq 2\tau,\end{aligned}$$

which establishes that $f_{\phi+1}$ is τ-codomain separable using θ in the FPA($\phi + 1$) model. By induction, this implies that $f_{\phi'}$ admits τ-codomain separation in the FPA(ϕ') model using θ for all $\phi' > \phi$. In the limit when $\phi \to \infty$, this is equivalent to saying that f admits τ-codomain separation in the APA model. □

Proposition 3. *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$, $\theta \in \Theta$, and $\tau > 0$. Let f_ϕ be the representation of f in the* FPA(ϕ) *model. If f admits τ-codomain separation in the* APA *model using some $\theta \in \Theta$ and for some $\tau > 0$, then f_ϕ admits τ-codomain separation in the* FPA(ϕ) *model using θ for $\phi \geq \max(2\log_2 \tau + 5, -2\log_2 \tau - 1)$.*

Proof. Suppose f admits τ-codomain separation in the APA model. To establish f_ϕ admits τ-codomain separation in the FPA(ϕ) model, we need to split the two cases: $\tau > 1$ and $\tau < 1$. Represent $\Lambda_\theta(f_\phi)$ in the FPA(ϕ) model as

$$b_{\frac{\phi-3}{2}} \cdots b_1 b_0 \,.\, b_{-1} b_{-2} \cdots b_{-\frac{\phi-1}{2}}.$$

When $\tau > 1$, a sufficient condition for f_ϕ to satisfy the τ-codomain separation is to have enough precision before the decimal point,

$$2^{\frac{\phi-3}{2}} \geq 2\tau,$$

which gives $\frac{\phi-3}{2} \geq 1 + \log_2 \tau \implies \phi \geq 2\log_2 \tau + 5$. When $\tau < 1$, a sufficient condition for f_ϕ to satisfy τ-codomain separation is to have enough precision after the decimal point,

$$2^{-\frac{\phi-1}{2}} \leq 2\tau,$$

which gives $-\frac{\phi-1}{2} \leq 1 + \log_2 \tau \implies \phi \geq -2\log_2 \tau - 1$. The proposition follows from combining the two together to obtain $\phi \geq \max(2\log_2 \tau + 5, -2\log_2 \tau - 1)$. □

C MISSING DETAILS FROM SECTION 3

We show that all Bregman divergence based loss functions are linearly-decomposable. Given a continuously differentiable strictly convex function $F : \mathcal{S} \to \mathbb{R}$ over some closed convex set $\mathcal{S} \subseteq \mathbb{R}^d$, the Bregman divergence $D_F : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ associated with F is defined as $D_F(\mathbf{p}, \mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \langle \nabla F(\mathbf{q}), \mathbf{p} - \mathbf{q} \rangle$. We will focus on the binary case for our discussion in this section and assume that the domain of F is the closed convex set $[0, 1]^N$.

Restatement of Lemma 1 *Let $F : [0, 1] \times [0, 1] \to \mathbb{R}$ be a strongly convex function and $D_F(p, q) = F(p) - F(q) - \langle \nabla F(q), p - q \rangle$ be the Bregman divergence associated with F. Let $f_F(\sigma, \theta)$ be the corresponding loss function, defined as follows:*

$$f_F(\sigma, \theta) = \frac{1}{N} \sum_{i=1}^{N} D_F \left([\sigma_i, 1 - \sigma_i], [\theta_i, 1 - \theta_i] \right).$$

Then, $f_F(\sigma, \theta)$ is linearly-decomposable.

Proof. First, observe that any loss function of the form $f(\theta, \sigma) = \sum_{i=1}^{N} (\sigma_i g(\theta_i) + (1 - \sigma_i) h(\theta_i))$ is additively linearly separable, since it can be rewritten as $f(\theta, \sigma) = \Sigma_{i=1}^{N} \sigma_i g'(\theta_i) + \sum_{i=1}^{N} h(\theta_i)$, where $g'(\theta_i) = g(\theta_i) - h(\theta_i)$.

Let $F : [0, 1] \times [0, 1] \to \mathbb{R}$ be a strongly convex function and $DF(p, q) = F(p) - F(q) - \langle \nabla F(q), p - q \rangle$ be the Bregman divergence associated with F. Let $\mathcal{L}_F(\theta, \sigma)$ be the corresponding loss function, defined as follows:

$$\mathcal{L}_F(\theta, \sigma) = \frac{1}{N} \sum_{i=1}^{N} D_F([\sigma_i, 1 - \sigma_i], [\theta_i, 1 - \theta_i]).$$

We start by using a shorthand $\phi(x) = F([x, 1 - x])$. Then, we can write the following:

$$\begin{aligned}
\mathcal{L}_F(\theta, \sigma) &= \frac{1}{N} \sum_{i=1}^{N} D_F([\sigma_i, 1 - \sigma_i], [\theta_i, 1 - \theta_i]) \\
&= \frac{1}{N} \sum_{i=1}^{N} (\phi(\sigma_i) - \phi(\theta_i) - \langle \nabla F([\sigma_i, 1 - \sigma_i]), [\sigma_i - \theta_i, \theta_i - \sigma_i] \rangle) \\
&= \frac{1}{N} \sum_{i=1}^{N} (\phi(\sigma_i) - \phi(\theta_i) - (\sigma_i - \theta_i) \langle \nabla F([\sigma_i, 1 - \sigma_i]), [1, -1] \rangle) \\
&= \frac{1}{N} \left(\sum_{i:\sigma_i = 1} \phi(1) - \phi(\theta_i) - (1 - \theta_i) \langle \nabla F([1, 0]), [1, -1] \rangle \right) + \\
&\qquad \frac{1}{N} \left(\sum_{i:\sigma_i = 0} \phi(0) - \phi(\theta_i) + \theta_i \langle \nabla F([0, 1]), [1, -1] \rangle \right)
\end{aligned}$$

Let $a = \langle \nabla F([1, 0]), [1, -1] \rangle$ and $b = \langle \nabla F([0, 1]), [1, -1] \rangle$ be constants. Then, we have the following:

$$\begin{aligned}
\mathcal{L}_F(\theta, \sigma) &= \frac{1}{N} \left(\sum_{i:\sigma_i = 1} \phi(1) - \phi(\theta_i) - a + \theta_i a \right) + \frac{1}{N} \left(\sum_{i:\sigma_i = 0} \phi(0) - \phi(\theta_i) + \theta_i b \right) \\
&= \frac{1}{N} \sum_{i=1}^{N} \left(\sigma_i \left(\phi(1) - a + \theta_i a \right) + (1 - \sigma_i) \left(\phi(0) + \theta_i b \right) \right) - \frac{1}{N} \sum_{i=1}^{N} \phi(\theta_i) \\
&= \sum_{i=1}^{N} \sigma_i g(\theta_i) + h(\theta),
\end{aligned}$$

where $g(\theta_i) = \frac{1}{N} \left(\theta_i (a - b) + \phi(1) - \phi(0) - a \right)$, and $h(\theta) = \phi(0) + \frac{1}{N} \sum_{i=1}^{N} \left(\theta_i b - \phi(\theta_i) \right)$. Thus, a separating vector

for the Bregman loss is obtained by setting $g(\theta_i) = \tau \ln p_i$, where p_i is the i^{th} prime number, as follows:

$$\begin{aligned}
&\frac{1}{N}\left(\theta_i(a-b) + \phi(1) - \phi(0) - a\right) = \tau \ln p_i \\
\implies &\theta_i = \frac{N\tau \ln p_i + \phi(0) + a - \phi(1)}{a-b} \\
\implies &\theta_i = \frac{N\tau \ln p_i + F([0,1]) + \langle \nabla F([1,0]), [1,-1]\rangle - F([1,0])}{\langle \nabla F([1,0]), [1,-1]\rangle - \langle \nabla F([0,1]), [1,-1]\rangle}.
\end{aligned}$$

□

Restatement of Theorem 1. *Let $g : [0,1] \to \mathbb{R}$ be some deterministic function and $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be a loss function that is g-linearly-decomposable (Definition 3). Then, for any $\tau > 0$, the function f is 2τ-codomain separable if there exists $\theta \in (0,1)^N$ so that $g(\theta_i) - g(1-\theta_i) > 2^i N\tau$ for all $i \in [N]$. If $\tau = 0$, then setting $g(\theta_i) - g(1-\theta_i) > 0$ for all $i \in [N]$ suffices for 0^+-codomain separability.*

Proof. We begin by observing that (3) can be rewritten as follows:

$$\begin{aligned}
f(\sigma, \theta) &= \frac{1}{N}\sum_{i=1}^{N} \left(\sigma_i g(\theta_i) + (1-\sigma_i) g(1-\theta_i)\right) \\
&= \frac{1}{N}\left(\sum_{i:\sigma(i)=1} \left(g(\theta_i) - g(1-\theta_i)\right) + \sum_{i=1}^{N} g(1-\theta_i)\right).
\end{aligned}$$

For any $\theta \in (0,1)^N$, we can then write the following:

$$\begin{aligned}
\Lambda_\theta(f) &= \min_{\sigma_1, \sigma_2 \in \{0,1\}^N} |f(\sigma_1, \theta) - f(\sigma_2, \theta)| \\
&= \frac{1}{N} \min_{\sigma_1, \sigma_2 \in \{0,1\}^N} \left| \sum_{i:\sigma_1(i)=1} \left(g(\theta_i) - g(1-\theta_i)\right) - \sum_{j:\sigma_2(j)=1} \left(g(\theta_j) - g(1-\theta_j)\right) \right|
\end{aligned}$$

If for all $i \in [N]$, it holds that $g(\theta_i) - g(1-\theta_i) = 2^i N\tau(1+\delta)$ for some $\delta > 0$, then:

$$\Lambda_\theta(f) = (2\tau(1+\delta)) \min_{\sigma_1, \sigma_2 \in \{0,1\}^N} \left| \sum_{i:\sigma_1(i)=1} 2^{i-1} + \sum_{j:\sigma_1(j)=1} 2^{j-1} \right| = 2\tau(1+\delta) > 2\tau,$$

where the last step holds because $\sigma_1 \neq \sigma_2$. □

C.1 Kullback-Leibler Divergence Loss

The (generalized) Kullback-Leibler (KL) divergence between vectors $\mathbf{p}, \mathbf{q} \in \mathcal{S} \subseteq \mathbb{R}^d$ is defined as:

$$D_{\mathrm{KL}}(\mathbf{p}, \mathbf{q}) = \sum_{i \in [d]} p_i \ln \frac{p_i}{q_i} - \sum_{i \in [d]} (p_i - q_i),$$

where $\mathbf{p} = (p_1, \ldots, p_d)$ and $\mathbf{q} = (q_1, \ldots, q_d)$.

For a binary classification setting, considering the ith datapoint, we have the true label $\sigma_i \in \{0,1\}$ and $\theta_i \in (0,1)$ which is the probability assigned to the event $\sigma_i = 1$ by the ML model. In that case, we have

$$D_{\mathrm{KL}}([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i]) = -\sigma_i \ln \theta_i - (1-\sigma_i)\ln(1-\theta_i).$$

Summing over the N datapoints (and dividing by N) gives the Kullback-Leibler divergence loss,

$$\begin{aligned}
\textsc{KLLoss}(\sigma, \theta) &= \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i]) = \frac{-1}{N}\sum_{i=1}^{N} \sigma_i \ln \theta_i + (1-\sigma_i)\ln(1-\theta_i) \\
&= \frac{-1}{N}\left(\sum_{i:\sigma_i=1} \ln \theta_i + \sum_{i:\sigma_i=0} \ln(1-\theta_i)\right), \qquad (7)
\end{aligned}$$

which is exactly the binary cross-entropy loss[4].

[4] Here, we adopt the notion that $0 \ln 0 = 0$, so that KL divergence is well-defined.

C.2 Itakura-Saito Divergence Loss

The Itakura-Saito divergence for vectors $\mathbf{p}, \mathbf{q} \in \mathcal{S} \subseteq \mathbb{R}^d$ is defined as:

$$D_{\text{IS}}(\mathbf{p}, \mathbf{q}) = \sum_{i \in [d]} \left(\frac{p_i}{q_i} - \ln \frac{p_i}{q_i} - 1 \right),$$

where $\mathbf{p} = (p_1, \ldots, p_d)$ and $\mathbf{q} = (q_1, \ldots, q_d)$.

For a binary classification setting, considering the ith datapoint, we have the true label $\sigma_i \in \{0, 1\}$ and $\theta_i \in (0, 1)$, which is the probability assigned to the event $\sigma_i = 1$ by the ML model. In this case, based on D_{IS}, the Itakura-Saito divergence loss is defined as:

$$\text{ISLoss}(\sigma, \theta) = \frac{1}{N} \left(\sum_{i:\sigma_i=1} \left(\frac{1}{\theta_i} + \ln \theta_i - 1 \right) + \sum_{i:\sigma_i=0} \left(\frac{1}{1-\theta_i} + \ln(1-\theta_i) - 1 \right) \right) \tag{8}$$

The above equation shows the linear decomposability of this loss, therefore, Theorem 1 can be applied to get the following result.

Restatement of Corollary 1. *The Itakura-Saito divergence loss (*ISLoss*) is 2τ-codomain separable with $\theta_i = \left(1 + 3^{2^i N\tau}\right)^{-1}$.*

Proof. We apply Theorem 1 here. For the Itakura-Saito divergence loss in (8), we begin by noticing that for $x \in (0, 1/2)$, it holds that

$$\frac{1}{x} - \frac{1}{1-x} + \ln \frac{x}{1-x} > \ln \frac{1-x}{x} > 0.$$

Thus, since

$$g(\theta_i) - g(1-\theta_i) = \frac{1}{\theta_i} - \frac{1}{1-\theta_i} + \ln \left(\frac{\theta_i}{1-\theta_i} \right)$$

for this loss, it suffices to ensure that $\ln \left(\frac{1-\theta_i}{\theta_i} \right) > 2^i N\tau$ for Theorem 1 to apply. In particular, we solve $\ln \left(\frac{1-\theta_i}{\theta_i} \right) = 2^i N\tau \ln 3$ to obtain $\theta_i = \left(1 + 3^{2^i N\tau}\right)^{-1}$. Note that $\theta_i < 1/2$ as needed above. □

C.3 Squared Euclidean Loss

The squared Euclidean divergence for vectors $\mathbf{p}, \mathbf{q} \in \mathcal{S} \subseteq \mathbb{R}^d$ is defined as:

$$D_{\text{SE}}(\mathbf{p}, \mathbf{q}) = \sum_{i \in [d]} \left(|p_i - q_i|^2 \right), \tag{9}$$

where $\mathbf{p} = (p_1, \ldots, p_d)$ and $\mathbf{q} = (q_1, \ldots, q_d)$.

Again for the binary classification setting, considering the ith datapoint, we get the following expression for this loss:

$$D_{\text{SE}}([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i]) = 2\|\sigma_i - \theta_i\|^2.$$

Summing over the N datapoints (and dividing by N, and ignoring the factor of 2), we get the squared Euclidean loss as follows:

$$\text{SELoss}(\sigma, \theta) = \frac{1}{N} \left(\sum_{i:\sigma_i=1} |\sigma_i - \theta_i|^2 + \sum_{i:\sigma_i=0} |\sigma_i - \theta_i|^2 \right) = \frac{1}{N} \left(\sum_{i:\sigma_i=1} (1-\theta_i)^2 + \sum_{i:\sigma_i=0} \theta_i^2 \right). \tag{10}$$

In this case, we establish 0^+-codomain separability.

Restatement of the first part of Corollary 2. *The squared Euclidean loss (*SELoss*) is 0^+-codomain separable using $\theta_i = (1/2)\,(1 - \ln(p_i)/N)$, where p_i is the ith prime number.*

Proof. To apply Theorem 1 to the squared Euclidean loss, we have $g(\theta_i) = (1-\theta_i)^2$, which gives $g(\theta_i) - g(1-\theta_i) = 1 - 2\theta_i$. Setting this to $\frac{\ln p_i}{N}$, where p_i is the ith prime number, ensures that $\mu(S_\theta) > 0$. Equivalently, $\theta_i = \frac{1}{2}\left(1 - \frac{\ln p_i}{N}\right)$ works. □

Note that the proof above assumes that $\theta \in (0,1)^N$. We show in Theorem 7 that restricting θ to $\{0,1\}^N$ prohibits τ-codomain separability for any $\tau > 0$.

C.3.1 Norm-like Divergence Loss

The norm-like divergence for vectors $\mathbf{p}, \mathbf{q} \in \mathcal{S} \subseteq \mathbb{R}^d$ and $\alpha \geq 2$ is defined as:

$$D_{\mathrm{NL}}(\mathbf{p}, \mathbf{q}) = \sum_{i \in [d]} \left(p_i^\alpha + (\alpha-1)q_i^\alpha - \alpha p_i q_i^{\alpha-1}\right),$$

where $\mathbf{p} = (p_1, \ldots, p_d)$ and $\mathbf{q} = (q_1, \ldots, q_d)$. Again for binary classification, considering the ith datapoint, we get the following expression for this loss:

$$D_{\mathrm{NL}}([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i]) \\ = \left(\sigma_i^\alpha + (\alpha-1)\theta_i^\alpha - \alpha\sigma_i\theta_i^{\alpha-1} + (1-\sigma_i)^\alpha + (\alpha-1)(1-\theta_i)^\alpha - \alpha(1-\sigma_i)(1-\theta_i)^{\alpha-1}\right).$$

Summing over the N datapoints (and dividing by N), and simplifying gives the norm-like divergence loss.

$$\textsc{NLLoss}(\sigma, \theta) = \frac{1}{N}\Big(\sum_{i:\sigma_i=1} \left(1 + (\alpha-1)\theta_i^\alpha - \alpha\theta_i^{\alpha-1} + (\alpha-1)(1-\theta_i)^\alpha\right) \\ + \sum_{i:\sigma_i=0} \left(1 + (\alpha-1)(1-\theta_i)^\alpha - \alpha(1-\theta_i)^{\alpha-1} + (\alpha-1)\theta_i^\alpha\right)\Big). \quad (11)$$

In this case, we establish 0^+-codomain separability.

Restatement of the second part of Corollary 2. *The norm-like divergence loss (*NLLoss*) for $\alpha \geq 2$ is 0^+-codomain separable using θ where: $(1-\theta_i)^{\alpha-1} - \theta_i^{\alpha-1} = (\ln p_i)/(N\alpha)$ with p_i as the ith prime number.*

Proof. Here we have $g(\theta_i) = 1 + (\alpha-1)\theta_i^\alpha - \alpha\theta_i^{\alpha-1} + (\alpha-1)(1-\theta_i)^\alpha$, which gives $g(\theta_i) - g(1-\theta_i) = \alpha((1-\theta_i)^{\alpha-1} - \theta_i^{\alpha-1})$. Similar to above, setting $g(\theta_i) - g(1-\theta_i) = \frac{\ln p_i}{N}$ suffices. This is equivalent to finding a solution to the following equation, which has a unique solution in $(0,1)$ for any fixed $\alpha \geq 2$:

$$(1-\theta_i)^{\alpha-1} - \theta_i^{\alpha-1} = \frac{\ln p_i}{N\alpha}.$$

It is easy to see that such a $\theta_i < 1/2$ exists. □

$$\Lambda_\theta(f) = \min_{\sigma_1 \neq \sigma_2} \left| \sum_{i:\sigma_1(i)=1} g(\theta_i) - \sum_{i:\sigma_2(i)=1} g(\theta_i) \right|. \quad (12)$$

Restatement of Theorem 2 *Let $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be a loss function that is linearly-decomposable (Definition 2). Let p_i is the i^{th} prime number and $P = \prod_{i=1}^N p_i$, is the product of the first N primes. Then, for any $\tau > 0$, setting $g(\theta_i) = 3P\tau \ln p_i$ for loss functions in Equation 2 ensures that $\Lambda_\theta(f) \geq 2\tau$. If $\tau = 0$, setting $g(\theta_i) = \ln p_i$ suffices for 0^+-codomain separability.*

Proof. We prove this by substituting $g(\theta_i) = 6P\tau \ln p_i$ in Equation 12 as follows:

$$\begin{aligned}\Lambda_\theta(f) &= \min_{\sigma_1 \neq \sigma_2} \left| \sum_{i:\sigma_1(i)=1} g(\theta_i) - \sum_{j:\sigma_2(j)=1} g(\theta_j) \right| \\ &= (3P\tau) \min_{\sigma_1 \neq \sigma_2} \left| \sum_{i:\sigma_1(i)=1} \ln p_i - \sum_{j:\sigma_2(j)=1} \ln p_j \right| \\ &= (3P\tau) \min_{\substack{S_1, S_2 \subseteq [N] \\ S_1 \cap S_2 = \emptyset}} \left| \ln \left(\frac{\prod_{i \in S_1} p_i}{\prod_{s \in S_2} p_j} \right) \right| \geq 2\tau.\end{aligned}$$

Too see why the last inequality holds, let S_1^* and S_2^* denote the sets of primes that achieve the minimum value above. Then, without loss of generality, it must hold that $\prod_{j \in S_2^*} p_j \leq P$ and $\prod_{j \in S_2^*} p_j \leq 1 + \prod_{i \in S_1^*} p_i$. Thus, using the fact that $1.5x \ln\left(\frac{1+x}{x}\right) > 1$ for all $x \geq 1$, we obtain that $1.5P$ times the log expression above is at least 1. Further scaling by 2τ gives $\Lambda_\theta(f) \geq 2\tau$. □

D MISSING DETAILS FROM SECTION 4

Worked out example for 0^+-codomain separability for multiclass cross-entropy loss. For illustration, we provide a simple example to demonstrate 0^+-codomain separability for the multiclass cross-entropy loss using a construction of prediction vector from (Aggarwal et al., 2021). Note that in Theorem 3, we establish that in fact, multiclass cross-entropy loss admits the stronger notion of τ-codomain separability for any $\tau > 0$.

Assume $N = 2$ and $K = 3$. Construct a matrix θ with first row $[\frac{2}{10}, \frac{3}{10}, \frac{5}{10}]$ and second row $[\frac{7}{31}, \frac{11}{31}, \frac{13}{31}]$. Observe that these vectors are chosen using unique prime numbers in the numerator (the denominator is for normalizing the sum to 1), the reasoning for which will be clear shortly. Using θ, one can prove that the cross-entropy loss will be distinct for every labeling by observing that the terms inside the logarithm, that are chosen for the outer sum in (4), are distinct for all labelings. For example, if the true labeling is $[0, 2]$, then we obtain $\text{CELOSS}([0,2];\theta) = -\frac{1}{2}\left(\ln \frac{2}{10} + \ln \frac{13}{31}\right) = -\frac{1}{2}\ln\left(\frac{2 \cdot 13}{10 \cdot 31}\right)$. Similarly, if the true labeling is $[1, 0]$, then we obtain $\text{CELOSS}([1,0];\theta) = -\frac{1}{2}\left(\ln \frac{3}{10} + \ln \frac{7}{31}\right) = -\frac{1}{2}\ln\left(\frac{3 \cdot 7}{10 \cdot 31}\right)$. The use of primes makes this selection of summands in the CELOSS score uniquely defined by the true labeling. This follows as the only thing that changes in the CELOSS score based on the true labeling is the numerator in the ln term, which is a product of primes based on true labeling. Since the product of primes has a unique factorization, we can recover which primes were used from the product, and since each entry in the matrix θ is associated with a unique prime, this recovers the true labels.

Missing proofs. We show that the (multiclass) K-ary cross-entropy loss is τ-codomain separable for any $\tau > 0$.

Restatement of Theorem 3. *Let $\tau > 0$. Define matrices $\vartheta, \theta \in \mathbb{R}^{N \times K}$ such that*

$$\vartheta_{n,k} = 3^{\left(2^{(n-1)K+k} N\tau\right)} \text{ and } \theta_{n,k} = \vartheta_{n,k} / \sum_{k=1}^{K} \vartheta_{n,k}.$$

Then, it holds that CELOSS *is 2τ-codomain separable using θ. If $\tau = 0$, then using $\vartheta_{n,k} = 3^{\left(2^{(n-1)K+k}\right)}$ ensures 0^+-codomain separability.*

Proof. We begin by simplifying the expression for $\text{CELOSS}(\theta, \sigma)$ to write it as a sum of two terms: one dependent on the labeling σ, and the other independent of this labeling.

$$\text{CELOSS}(\theta, \sigma) = \frac{-1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \Big([\sigma_i = k] \cdot \ln \theta_{i,k} \Big) = \frac{-1}{N} \left(\underbrace{\sum_{i=1}^{N} \ln \vartheta_{i,\sigma_i}}_{\substack{\text{Labeling Dependent} \\ \text{Term}}} - \underbrace{\sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \vartheta_{i,k} \right)}_{\substack{\text{Labeling Independent} \\ \text{Term}}} \right). \tag{13}$$

Using (13), we then obtain the following:

$$\begin{aligned}
\Lambda_\theta(\text{CELoss}) &= \min_{\sigma_1,\sigma_2\in\mathbb{Z}_K^N} |\text{CELoss}\,(\theta,\sigma_1) - \text{CELoss}\,(\theta,\sigma_2)| \\
&= \min_{\sigma_1,\sigma_2\in\mathbb{Z}_K^N} \frac{1}{N}\left|\sum_{i=1}^{N}\,\left(\ln\theta_{i,\sigma_1(i)}\right) - \sum_{i=1}^{N}\,\left(\ln\theta_{i,\sigma_2(i)}\right)\right| \\
&= \min_{\sigma_1,\sigma_2\in\mathbb{Z}_K^N} \frac{1}{N}\left|\sum_{i=1}^{N}\,\left(\ln\vartheta_{i,\sigma_1(i)}\right) - \sum_{i=1}^{N}\,\left(\ln\vartheta_{i,\sigma_2(i)}\right)\right| \\
&= \min_{\sigma_1,\sigma_2\in\mathbb{Z}_K^N} (\tau\ln 3)\left|\sum_{i=1}^{N} 2^{(i-1)K+\sigma_1(i)} - \sum_{i=1}^{N} 2^{(i-1)K+\sigma_2(i)}\right| \geq 2\tau\ln 3 > 2\tau.
\end{aligned}$$

□

D.1 Sigmoid Cross-Entropy Loss

The separability from Theorem 3 also holds if we apply any bijective activation function before applying the cross-entropy loss. As an example, consider the sigmoid cross-entropy commonly used in the binary classification setting (to compresses arbitrary reals into the range $(0,1)$), defined as follows for $\sigma\in\{0,1\}^N$, $\theta\in(0,1)^N$:

$$\frac{-1}{N}\sum_{i=1}^{N}\,\Big(\sigma_i\ln\left(\text{SIGMOID}(\theta_i)\right) + (1-\sigma_i)\ln\left(1-\text{SIGMOID}(\theta_i)\right)\Big), \tag{14}$$

where $\text{SIGMOID}(x) = (1+e^{-x})^{-1}$ is the sigmoid function. Since $\text{SIGMOID}:\mathbb{R}\to(0,1)$ is a bijection (and hence, invertible), given $\text{SIGMOID}(x) = y$, we can obtain $x = \ln(y/(1-y))$. Thus, given the matrix $\theta\in[0,1]^{N\times 2}$ from Theorem 3, we can construct $\theta'\in(0,1)^N$ such that $\theta'_i = \ln(\theta_{i,1}/(1-\theta_{i,1}))$ for all $i\in[N]$. Once θ' is obtained, since $\text{SIGMOID}(\theta_i) = \theta_{i,1}$ and $1-\text{SIGMOID}(\theta_i) = \theta_{i,2}$ we get that sigmoid cross-entropy loss is 2τ-codomain separable using θ', and hence the approach outlined in LABELINF (1) can be used for τ-robust label inference.

E MISSING DETAILS FROM SECTION 5

Theorem 6. *Let $\tau>0$ and let $\theta\in(0,1)^{d_2}$ be such that $f:\mathbb{Z}_K\times(0,1)^{d_2}\to\mathbb{R}$ is 2τ-codomain separable using θ. Then, for any input $\mathbf{v}\in(0,1)^{d_2}$, given ℓ such that $|f(\sigma_{\mathbf{v}},\text{MUTNET}_\theta(\mathbf{v}))-\ell|\leq\tau$, the approach outlined in* LABELINF (1) *recovers $\sigma_{\mathbf{v}}$.*

Proof. It suffices to show that for given $\theta\in(0,1)^{d_2}$ and any $\mathbf{v}\in(0,1)^{d_2}$, the construction above ensures that $\text{MUTNET}_\theta(\mathbf{v}) = \theta$. To see this, observe that since all entries in M_1 are negative, the product $\mathbf{v}^\top M_1$ has non-positive entries. Thus, when RELU is applied to $\mathbf{v}^\top M_1$ (element-wise), the output is the zero vector. This zero vector, when fed into the Sigmoid, produces the desired output θ since $\mathbf{x}'$ is constructed in a way such that $\text{SIGMOID}(\mathbf{x}') = \theta$ (element -wise). □

F SOME NEGATIVE RESULTS ON τ-CODOMAIN SEPARABILITY

We now show certain loss functions are not τ-codomain separable. This complements our positive results on τ-codomain separability for cross-entropy and its variants, and Bregman divergence based losses. These negative results on codomain separability rules out label inference in these cases, because of the connections between these two notion established in Proposition 1.

Discrete L_p-losses. We start with the simple L_p-loss defined on the *discrete* domain and show it is not τ-codomain separable for any $\tau>0$.

Theorem 7. *For any $p>0$, the function $f:\{0,1\}^N\times\{0,1\}^N\to\mathbb{R}$ of the form $f(\sigma,\theta) = ||\theta-\sigma||_p$ is not τ-codomain separable for any $\tau>0$.*

Proof. Fix some $\theta \in \{0,1\}^N$. For any $\sigma \in \{0,1\}^N$, let $I(\sigma,\theta) = \{i \in [N] \mid \sigma(i) \neq \theta(i)\}$ be the set of indices on which σ and θ differ. Then, we can simplify the expression for f as follows:

$$f(\sigma,\theta) = \left(\sum_{i=1}^{N} |\theta(i) - \sigma(i)|^p\right)^{1/p} = \left(\sum_{i \in I(\sigma,\theta)} |\theta(i) - \sigma(i)|^p\right)^{1/p} = |I(\sigma,\theta)|^{1/p}.$$

Now, let $\sigma_1, \sigma_2 \in \{0,1\}^N$ be such that they differ from θ in exactly one label, i.e., $|I(\sigma_1,\theta)| = |I(\sigma_2,\theta)| = 1$ and hence, $f(\sigma_1,\theta) = f(\sigma_2,\theta) = 1$. Note that for any choice of θ, there are $N-1$ such labelings. Thus, $\Lambda_\theta(f) = 0$. □

Set-valued Functions. We now study set-valued loss functions. These are functions that are expressed with respect to a fixed set, as a mapping from subsets of this set to the real line. For example, in our context of codomain separability (in the binary classification setting), the set of interest is that of the N datapoints, and the subsets are interpreted as comprising of those that have been assigned label 1. For example, if $N = 3$ and the subset is $\{1,3\}$, then this would represent the case where datapoints 1 and 3 have labels 1, and datapoint 2 has label 0. As we will see, this generalization helps compute upper bounds on the magnitude of noise that will admit label inference (in a single query) using any prediction vector.

We now present our main results in this section. For the discussion here, we will assume $\Omega = \{s_1, \ldots, s_N\}$ to denote a set and 2^Ω to denote the power set of Ω. As mentioned before, since the sets of interest in our application can be thought of as the labels for the datapoints, we will assume $|\Omega| = N$, unless mentioned otherwise.

Theorem 8. *Let $\Omega = \{s_1, \ldots, s_N\}$ be a set. Let $f : 2^\Omega \times \Theta \to \mathbb{R}_+$ be a function and $\theta \in \mathbb{R}^N$ be such that $f(\cdot,\theta)$ is monotonic, i.e.,, for all $A \subseteq B \subseteq [N]$, it holds that $f(A,\theta) \leq f(B,\theta)$. Then, f is not τ-codomain separable using θ for any*

$$\tau > \min_{B \subset [N]} \min_{j \notin B} \left(\frac{f(B \cup \{j\},\theta) - f(B,\theta)}{2}\right).$$

In particular, if $f(\emptyset,\theta) = 0$, then f is not τ-codomain separable using θ for any $\tau > \frac{1}{2}\left(\min_{j \in [N]} f(\{j\},\theta)\right)$.

Proof. Fix some $\sigma \in [0,1]^N$. Then, for f to be τ-codomain separable using σ, it must hold that:

$$\begin{aligned}
&\forall B \subset [N], j \notin B. \qquad |f(B \cup \{j\},\theta) - f(B,\theta)| \geq 2\tau \\
\implies &\forall B \subset [N]. \qquad \tau \leq \frac{1}{2}\left(\min_{j \notin B} f(B \cup \{j\},\theta) - f(B,\theta)\right) \quad \text{(since } f(\theta,\cdot) \text{ is monotonic)} \\
\iff &\tau \leq \min_{B \subset [N]} \min_{j \notin B} \left(\frac{f(B \cup \{j\},\theta) - f(B,\theta)}{2}\right).
\end{aligned}$$

Taking the contrapositive of this statement establishes the desired result. When $f(\emptyset,\theta) = 0$, then setting $B = \emptyset$ gives the desired result. □

Corollary 3. *Let $f : 2^\Omega \times \Theta \to \mathbb{R}_+$ be a function such that $f(\cdot,\theta)$ is monotonic for all $\theta \in \mathbb{R}^N$. Then, f is not τ-codomain separable for any*

$$\tau > \sup_{\theta \in \mathbb{R}^N} \min_{B \subset [N]} \min_{j \notin B} \left(\frac{f(B \cup \{j\},\theta) - f(B,\theta)}{2}\right).$$

In particular, if $f(\emptyset,\theta) = 0$ for all $\theta \in \mathbb{R}^N$, then f is not τ-codomain separable for any

$$\tau > \frac{1}{2}\left(\sup_{\theta \in \mathbb{R}^N} \min_{j \in [N]} f(\{j\},\theta)\right).$$

We now show that if in addition to monotonicity, the loss function is also bounded, then we can get stronger negative results.

Theorem 9. *Let $\Omega = \{s_1, \ldots, s_N\}$ be a set. Let $f : 2^\Omega \times \Theta \to \mathbb{R}_+$ be a function such that $f(\cdot,\theta)$ is monotonic and $f(\cdot,\theta) \leq \beta$ for all $\theta \in \mathbb{R}^N$ and for some finite $\beta > 0$. Then, f is not τ-codomain separable for any $\tau > \beta/N$.*

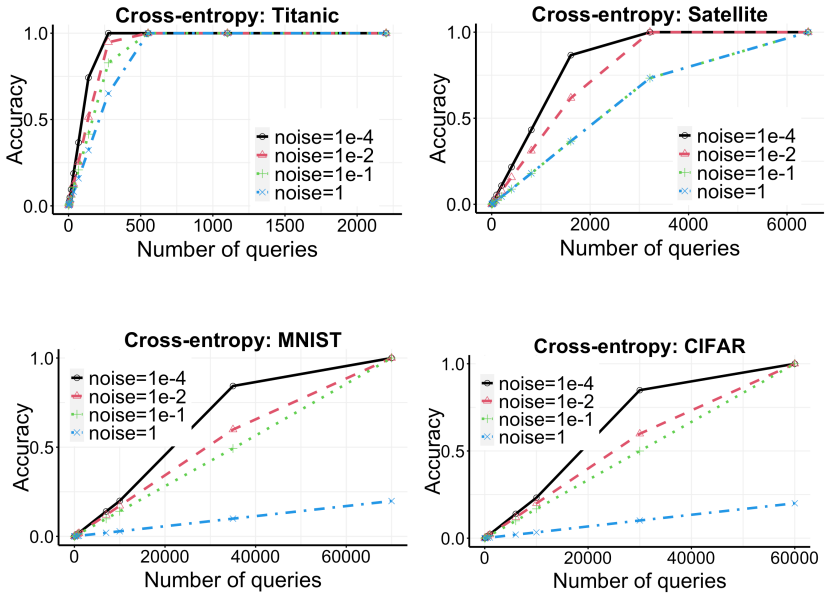
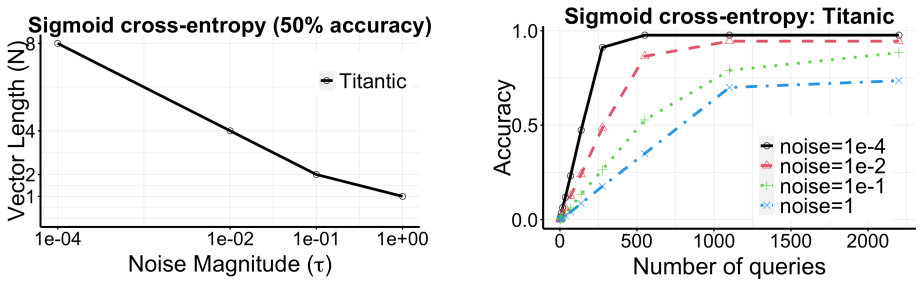


Figure 3: Label reconstruction accuracy with the multi-query label inference attack.



(a) (b)

Figure 4: The plot on the left shows the length of vector recovered (at 50% accuracy) using single query. The plot on the right shows label reconstruction accuracy with the multi-query label inference attack.

Proof. Assume that f is τ-codomain separable using some θ. Consider the chain of values $v_0 = f(\{\,\}, \theta), v_1 = f(\{1\}, \theta), f(\{1,2\}, \theta), \ldots, v_N = f(\{1, \ldots, N\}, \theta)$. For each $i \in [N]$, since f is τ-codomain separable, we must have $|v_i - v_{i-1}| \geq \tau$. Since f is monotonic, this implies $v_i - v_{i-1} \geq \tau$. Summing both sides over i gives $\sum_{i=1}^{N} (v_i - v_{i-1}) = v_N - v_0$, which must be at least $N\tau$ for the inequality above to hold. This implies $v_N \geq N\tau + v_0$, which, for $\tau > \beta/N$ gives $v_N > \beta$ (since f is non-negative). This is a contradiction since f is bounded above by β. □

G MISSING DETAILS FROM SECTION 6

We now discuss the missing details from Section 6 and present our results for label inference from the Sigmoid cross entropy loss function.

Label Inference from Binary Cross-Entropy Loss. In our experiments, for binary cross-entropy, we use the label inference attack of (Aggarwal et al., 2021) as a baseline (see Figures 1(a) and (b)).

Theorem 10 (τ-codomain Separability from Algorithm 2 in (Aggarwal et al., 2021))**.** *Let $\tau > 0$. For the binary case (with class labels 0 and 1), define $\theta_i = \left(\frac{3^{2^i} N\tau}{1+3^{2^i} N\tau}\right)$ for all $i \in [N]$. Then,* CELoss *is 2τ-codomain separable using θ.*

Next, we discuss some technical caveats about the results for the softmax cross entropy loss, as observed in Figure 1(d).

Additional bits Needed for Softmax Cross-Entropy Loss. Recall from our discussion in Section 6 that computing the softmax cross-entropy loss will require an additional $\Omega(NK + \ln(N\tau))$ bits over those required for the multiclass cross-entropy loss. We now formally argue this result.

Observe that for label inference in the softmax case, it suffices to compute a vector $\theta' = [\theta'_1, \ldots, \theta'_N]$ such that $\textsc{Softmax}(\theta'_i) = \theta$, where θ is our desired vector for label inference. This is equivalent to requiring: $e^{\theta'_i} / \sum_j e^{\theta'_j} = \theta_i$, which gives rise to:

$$\frac{e^{x_1}}{\theta_1} = \cdots = \frac{e^{x_N}}{\theta_N}.$$

Thus, for any i and j, we can write $\theta'_i = \theta'_j + \ln\left(\frac{\theta_i}{\theta_j}\right)$. Now, let

$$i_\uparrow = \arg\max_{i \in [N]} \theta_i \text{ and } i_\downarrow = \arg\min_{i \in [N]} \theta_i.$$

Then, we can write $x_{i_\uparrow} = x_{i_\downarrow} + \ln\left(\frac{\theta_{i_\uparrow}}{\theta_{i_\downarrow}}\right)$. Thus, the additional number of bits required to represent the entries in x is $\Omega\left(\ln\ln\left(\frac{\theta_{i_\uparrow}}{\theta_{i_\downarrow}}\right) - \ln\theta_{i_\uparrow}\right) = \Omega\left(\ln\ln\left(\frac{\theta_{i_\uparrow}}{\theta_{i_\downarrow}}\right)\right)$. From our construction in Theorem 3, we know that the ratio $\frac{\theta_{i_\uparrow}}{\theta_{i_\downarrow}} \approx 3^{2^{NK} N\tau}$. This means that we need an additional $\Omega\left(\ln\ln 3^{2^{NK} N\tau}\right) = \Omega\left(NK + \ln(N\tau)\right)$ bits for the softmax cross-entropy loss as compared to the (plain) multiclass cross-entropy loss.

Additional Experimental Results. Figure 3 presents the results with multiclass cross-entropy loss on Titanic, Satellite, MNIST, and CIFAR datasets, using the same setting as in the multi-query experiments in Figure 2.

Figure 4 presents the results on sigmoid cross-entropy which is applicable only in the binary labeled setting (see (14)) on the Titanic dataset. The results are worse compared to (plain) multiclass cross-entropy loss. This is because the number of bits required to represent $\textsc{Sigmoid}^{-1}(\theta_i)$ (where θ_i is an element of θ) is $\Omega\left(\ln|\ln(\theta_i/(1-\theta_i))|\right)$, which asymptotically dominates the $\Omega\left(|\ln\theta_i|\right)$ many bits required to represent θ_i for the (plain) multiclass cross-entropy loss.