Reconstructing Test Labels from Noisy Loss Scores (Extended Abstract)

Abhinav Aggarwal, Shiva Prasad Kasiviswanathan, Zekun Xu,
Oluwaseyi Feyisetan, Nathanael Teissier
Amazon, USA
{aggabhin,kasivisw,zeku,sey,natteis}@amazon.com

Abstract

Label inference was recently introduced as the problem of reconstructing the ground truth labels of a private dataset from just the (possibly perturbed) cross-entropy loss scores evaluated at carefully crafted prediction vectors. In this paper, we generalize this result to provide necessary and sufficient conditions under which label inference is possible from a broad class of loss functions. We show that for many commonly used loss functions, including linearly decomposable losses, some Bregman divergence-based losses and when common activation functions are used, it is possible to design such attacks for arbitrary noise levels. We demonstrate that these attacks can also be carried out through a lightweight augmentation to any neural network model, enabling the adversary to make these attacks look benign. 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 classification models, can grasp the potential scope of the resulting information leakage.

1 Introduction

Consider a situation where a machine learning (ML) modeler is interacting with a data curator. The data curator owns a private dataset for a classification task, and 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 [3], KDDCup [4], and ILSVRC Challenges [1]. 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. In these settings, it is common for the curator to return the loss scores to the modeler (such as for the purposes of fine-turning the models or ranking etc.). A similar situation also appears when dealing with sensitive datasets, where either labels, or, both features and labels could be considered sensitive, and again 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. More broadly, we investigate the problem of (robust) label inference, where the goal is to infer the labels of a hidden dataset from just the (noisy) loss function queries evaluated on the dataset. Our focus will be on a particularly strong inference attack 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), for many common ML loss functions, our label inference attack succeeds in exactly recovering *all* the true labels. This has important ramifications; for example, our attacks could be used by: a) an unscrupulous participant to an ML competition who could learn the labels of the hold-out test set in order to construct a fake model that achieves perfect classification; or b) an adversary to execute a privacy breach by learning labels associated with a sensitive dataset (see our Ethics and Malpractice statement at the end).

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Table 1: Summary of our label inference results on commonly used ML loss functions. Here, N is the number of labels to be inferred and $K \geq 2$ is number of label classes. In the multi-query case, our attacks utilize $O(N/\log N)$ queries.

Loss Function	K	Past Work	This Work	Local Computation	
				1-Query	Multi-Query
Binary Cross Entropy	2	Noised [8]	Noised (Thm. 3)	$O(2^N)$	$O(\text{poly}(N))$
K-ary Cross Entropy	K	Unnoised [8]	Noised (Thm. 3)	$O(K^N)$	–
Softmax Cross Entropy	K	–	Noised (Sec. B)	$O(K^N)$	–
Sigmoid Cross Entropy	2	–	Noised (Sec. B)	$O(2^N)$	$O(\text{poly}(N))$
Itakura-Saito Divergence	2	–	Noised (Cor. 2)	$O(2^N)$	$O(\text{poly}(N))$
Squared Euclidean	2	–	Unnoised (Cor. 3)	$O(2^N)$	$O(\text{poly}(N))$
Norm-like Divergence	2	–	Unnoised (Cor. 3)	$O(2^N)$	$O(\text{poly}(N))$
Mahalanobis Divergence	2	–	Noised (Thm. 8)	$O(2^N)$	–

Our attacks rely on a mathematical notion of *codomain separability*[1], which posits that the output of the loss function is distinct on every possible labeling of the test datapoints[2]. 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. The main technical challenge here is to design prediction vectors for which a loss function demonstrates the required codomain separability for arbitrary noise levels.[3] We provide constructions of such vectors for broad classes of loss functions (see Table 1 and the technical appendices for detailed proofs). 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 primarily when we look at the local computation time required by the adversary for the inference. Additionally, to handle situations where the curator might require a submission of an actual ML model (and not just the prediction vector), we provide a construction of a feed-forward neural network, which can be used to carry out our inference attacks (see Appendix G).

Central Idea. Our key idea is to first reduce codomain separability to the construction of sets with distinct subset sums, and then use this to outline an efficient procedure to construct prediction vectors that impart the desired separation in the outputs of the loss function. For example, in the binary classification setting with N datapoints, the following problem comes up often in our analysis:

$$\text{Construct } \theta = [\theta_1, \ldots, \theta_N] \text{ such that } : \min_{\substack{\sigma_1, \sigma_2 \in \{0,1\}^N \\ \sigma_1 \neq \sigma_2}} \left| \sum_{i:\sigma_1(i)=1} g(\theta_i) - \sum_{j:\sigma_2(j)=1} g(\theta_j) \right| \geq b,$$

for some function g and bound b. To satisfy this inequality, it suffices to set θ such that the subset sums in the set $g(\theta) := \{g(\theta_i), \ldots, g(\theta_N)\}$ differ by atleast b. 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. The trick here is to compute θ by solving for $g(\theta_i) = bs_i$, where $S = \{s_1, \ldots, s_N\}$ is a set with subset sums that differ by at least 1. Here, S is chosen depending on actual form of g and the application (for example, two sets we frequently use in our analysis 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).

Related Work. Label inference attacks using loss function values were introduced by Whitehill [21, 20] and then later studied by Aggarwal *et al.* [7], both with a primary focus on the binary cross-entropy loss. These attacks range from heuristically solving a min-max optimization problem [21] to using prime factorization [7] for recovering the true labels. However, neither attacks extend to the noised setting. The problem was partially addressed recently by Aggarwal *et al.* in [8], where they provide an algorithm for label inference on noisy binary cross-entropy scores. Even though 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. They also provide an algorithm for label inference from multiclass cross-entropy scores in the unnoised setting (see Table 1). Our results not only subsume these, but also significantly extend them.

[1] This is closely related to the notion of Lipeomorphism and inverse Lipschitz continuity over $\mathcal{L}_p$ spaces [19].

[2] See Proposition 1, where we show that this is a necessary and sufficient condition for robust label inference.

[3] We show that for general functions, determining codomain separability is co-NP-hard (see Appendix H).

For attacks on leaderboards, such as in Kaggle competitions, Blum and Hardt demonstrate an adversarial boosting attack in [11], where the adversary observes loss scores on randomly generated prediction vectors to generate a labeling which, with probability $2/3$, gives a low loss score. Our focus on the other hand is on exact label recovery, especially in the noised setting.

Paper Organization. In this extended abstract, we only demonstrate our technique for the class of linearly decomposable (binary) loss functions in Section 2 and present some experimental results in Section 3. Our detailed analysis and empirical results are presented in the Technical Appendix.

2 Label Inference from Linearly Decomposable (Binary) Loss Functions

Informally, a loss function $(\sigma, \theta) \mapsto \mathbb{R}$ takes as input a labeling $\sigma \in \{0,1\}^N$ and a prediction vector $\theta \in [0,1]^N$ to produce a loss score in $\mathbb{R}$. For some $\tau > 0$, we call this function τ-codomain separable if there exists some vector θ such that the output of the loss function (keeping θ fixed) on any two labelings differs by at least τ. Thus, when θ is known, the one-one correspondence between the function's output and the labelings can be exploited for label inference (for example, through an exhaustive enumeration as in Algorithm LABELINF, Appendix A) to exactly recover all the labels from just observing the output, which we refer to as *robust label inference* (see Definition 2 in Appendix A). A necessary and sufficient condition for inferring all ground truth labels from a loss function, noised upto τ, is for this function to admit 2τ codomain separability (see Proposition 1).

We say that a binary loss function $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ is linearly-decomposable in $\{0,1\}^N$ using some function $g : (0,1) \to \mathbb{R}$ if it can be expressed in the following form:

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

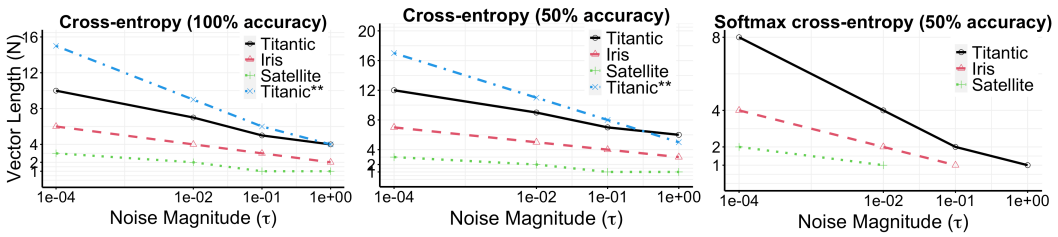
Observe that the KL-divergence loss (which reduces to binary cross-entropy loss) is of this form, using $g(\theta_i) = -\ln \theta_i$ (see (7), Appendix F.1). Some other common examples of linearly-decomposable losses include the (i) Itakura-Saito divergence loss, which can be expressed using $g(\theta_i) = 1/\theta_i + \ln \theta_i - 2$ (see (8), Appendix F.1); (ii) squared Euclidean loss, which can be expressed using $g(\theta_i) = (1 - \theta_i)^2$, (see (9), Appendix F.1); and, (iii) norm-like divergence loss, which can be expressed using $g(\theta_i) = 1 + (\alpha - 1)\theta_i^\alpha - \alpha\theta_i^{\alpha-1} + (\alpha - 1)(1 - \theta_i)^\alpha$ for some $\alpha \geq 2$ (see (11), Appendix F.1).

Our main result for linearly decomposable loss functions is that for any deterministic function $g : [0,1] \to \mathbb{R}$ and any bound $\tau > 0$ on the noise, 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 $i \in [N]$ (see Theorem 4). Constructing such a vector θ can be done 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 Algorithm LABELINF (see detailed examples in Appendix F).

Multi-Query Polynomial Time Attacks. The linear-decomposability of f allows for an efficient multi-query label inference algorithm for f. In particular, we could have a trade-off between the ability to perform multiple queries with faster computation times to avoid an exhaustive search in the space of all labelings. 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 $\theta = [\theta_1, \ldots, \theta_M, 1/2, \ldots, 1/2]$ to obtain the following (with $\Lambda_\theta(f) := \min_{\sigma_1, \sigma_2} |f(\sigma_1, \theta) - f(\sigma_2, \theta)|$ for brevity):

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

Similar to the discussion above, choosing θ_i that ensures $g(\theta_i) - g(1 - \theta_i) > 2^i N\tau$ will ensure $\Lambda_\theta(f) > 2\tau$. Moreover, using this θ will ensure that if $f(\sigma_1, \theta) = f(\sigma_2, \theta)$, then $\sigma_1[:M] = \sigma_2[:M]$ must hold, so that after recovering the first M labels, we can recover the next M labels using $\theta = [1/2, \ldots, 1/2, \theta_{M+1}, \ldots, \theta_{2M}, 1/2, \ldots, 1/2]$, and so on. It immediately follows that a sufficient condition for f to admit τ-robust label inference with $\lceil N/M \rceil$ queries is for the inequality $g(x) - g(1 - x) > 2^i N\tau$ to have a solution for all $i \in [N]$ (see Theorem 5). More importantly, 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 Algorithm LABELINF in each query). Thus, while the single query case ($M = N$) required $O(2^N)$ computations, any multi-query algorithm using $M = O(\log N)$ requires only $O(\text{poly}(N))$ local computations, which allow a polynomial-time robust label inference.



(a) (b) (c)

Figure 1: Results for τ-robust label inference using Algorithm LABELINF. Figures (a)-(b) plots the length of vector (N) on which Algorithm LABELINF always succeeds with multiclass cross-entropy loss with 100% and at least 50% accuracy respectively. Here, Titanic** implements the binary label inference attack proposed in [8]. Figure (c) plots the same for the softmax cross-entropy loss at 50% accuracy (wherever computable). The computation time per inference attack is roughly 10 seconds.

3 Empirical Analysis

We now present our empirical evaluation of label inference attacks on cross-entropy loss and its extensions (see Appendix J for additional results)[4]. We use the following datasets for our analysis. As our attacks construct prediction vectors that are independent of the dataset contents, we ignore the dataset features in our experiments. All the labels of the dataset are considered for the attack.

- **Titanic** [6]: a binary classification dataset (2201 rows) on the survival status of passengers.
- **Iris** [2]: a three-class classification dataset (150 rows) on the Iris species.
- **Satellite** [5]: a six-class classification dataset (6430 rows) on the satellite images of soil.

We consider three common loss functions (formally defined in Appendix B): multiclass cross-entropy, softmax cross-entropy, and sigmoid cross-entropy (deferred to Appendix J). As a baseline, for the binary labeled dataset (Titanic) with the cross-entropy loss, we implemented the label inference attack of Aggarwal *et al.* [8] (see Theorem 14, Appendix J for a restatement of their main result).

We start with the distinction between our experiments and Algorithm LABELINF, which is presented in the APA model[5]. For example in the multiclass cross-entropy loss (K-CELOSS) case, to simulate Algorithm LABELINF on a finite precision machine, we must be able to differentiate $\min_{i,k} \theta_{i,k}$ from 0 (or else 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 and is likely not tight, but gives an idea of how arithmetic precision plays a role in our experiments.

Figure 1 shows the number of datapoints N recovered by Algorithm LABELINF as we increase the noise for various loss functions. 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 for cross-entropy loss decreases as the noise magnitude (τ) increases, which is expected. Moreover, at the same noise magnitude, the accuracy also drops as the number of classes (K) increases from 2 (in Titanic) to 6 (in Satellite), 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 for the Titanic dataset, the construction from [8] 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 [8] (see Theorem 14, Appendix J), but $\Omega(2^{2N} N\tau)$ when using Theorem 3 with $K = 2$.

Figures 1(b)-(c) show the number of datapoints on which Algorithm LABELINF achieves at least 50% accuracy for cross-entropy and softmax cross-entropy loss. We notice this number is smaller for softmax cross-entropy as compared to regular cross-entropy (see Figure 1(a)), 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

[4] 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.

[5] APA stands for Arbitrary Precision Arithmetic model (see Appendix A.1).

$\Omega(NK + \ln(N\tau))$ bits (see the discussion in Appendix J for details). This additional requirement further constraints the number of labels that can be recovered with softmax cross-entropy loss.

Concluding Remarks. In this paper, we formalized a notion of codomain separability of arbitrary loss functions. Through an equivalence established between this notion and the ability to perform label inference attacks, we characterize necessary and sufficient conditions for label inference from a large class of commonly used loss functions in ML, under arbitrary noise levels. For defending against such attacks, one could consider differentially private mechanisms for releasing loss scores [15].

An interesting open question is to study the implications of codomain separability on discrete function optimization (e.g., maximization of functions defined on $\mathbb{Z}_K^N$). Intuitively, a large separation in the function's output space can be exploited to design algorithms that converge inside a small neighborhood around the global optimum, which may then be recoverable using local search.

References

[1] http://www.image-net.org/challenges/LSVRC/. Accessed on September 15, 2021.

[2] https://www.openml.org/d/41511. Accessed on September 15, 2021.

[3] https://www.kaggle.com. Accessed on September 15, 2021.

[4] https://www.kdd.org/kdd-cup. Accessed on September 15, 2021.

[5] https://www.openml.org/d/182. Accessed on September 15, 2021.

[6] https://www.openml.org/d/40704. Accessed on September 15, 2021.

[7] Abhinav Aggarwal, Zekun Xu, Oluwaseyi Feyisetan, and Nathanael Teissier. On primes, log-loss scores and (no) privacy. In *Workshop on Privacy in Natural Language Processing at the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

[8] Abhinav Aggarwal, Shiva Kasiviswanathan, Zekun Xu, Oluwaseyi Feyisetan, and Nathanael Teissier. Label inference attacks from log-loss scores. In *International Conference on Machine Learning*. PMLR, 2021.

[9] Sanjeev Arora and Boaz Barak. *Computational complexity: A Modern Approach*. Cambridge University Press, 2009.

[10] Andreas Blass and Yuri Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1-3):80–88, 1982.

[11] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. *arXiv preprint arXiv:1502.04585*, 2015.

[12] Lev 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.

[13] Richard P Brent and Paul Zimmermann. *Modern computer arithmetic*, volume 18. Cambridge University Press, 2010.

[14] Irit Dinur and Kobbi 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.

[15] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[16] Donald E Knuth. *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 2014.

[17] Chaoyue Liu and Mikhail 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.

[18] Christos H Papadimitriou and Mihalis 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.

[19] Brian S Thomson, Judith B Bruckner, and Andrew M Bruckner. *Elementary real analysis*, volume 1. ClassicalRealAnalysis. com, 2008.

[20] Jacob Whitehill. Exploiting an oracle that reports auc scores in machine learning contests. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

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

Ethics and Malpractice Statement. Our paper can be viewed from two different lenses: exploiting, and information with the intent of protection. On one hand, the results from our paper demonstrate how to exploit loss scores that are routinely vended from ML models. This can potentially be leveraged by malicious attackers who might take these techniques and use them to craft the prediction vectors used to launch these attacks. The negative effects of these are as varied as the domains which expose these scores over a classification API. For example, it can be exploited for financial gains such as with ML competitions, or to discover the test sets of sensitive models such as in health domains.

On the other hand, 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 classification models, can grasp the potential scope of the information leakage that can result from this. While this paper did not go into full details on how to prevent these attacks, the knowledge of it can lead to tightening of security permissions such as preventing ad-hoc prediction vectors from running against the models, or requiring other methods of querying the classification models.

Technical Appendix for "Reconstructing Test Labels from Noisy Loss Scores (Extended Abstract)"

A Codomain Separability and Robust Label Inference

We begin our discussion by formally defining the notion of codomain separability and its connections to the problem of label inference in the noised and unnoised settings (missing details in Appendix D).

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. For example, in the case of binary cross-entropy loss (log-loss), we have a prediction probability vector $\theta \in [0,1]^N$ and a labeling $\sigma \in \{0,1\}^N$; the loss on (σ, θ) is given by:

$$\text{CELOSS}\,(\sigma, \theta) = \frac{-1}{N} \ln \left(\prod_{i=1}^{N} \theta_i^{\sigma_i} (1-\theta_i)^{1-\sigma_i} \right). \tag{2}$$

In this case, θ_i is the probability assigned to the event $\sigma_i = 1$ by the ML model. We work with different loss functions that place different restrictions on Θ. For example, in the case of multiclass cross-entropy loss (with K classes), we have $\Theta \in [0,1]^{N \times K}$.

Codomain Separability. Informally, we call a function codomain separable if there exists some vector $\theta \in \Theta$ such that the function output on each $\mathbb{Z}_K^N$ is distinct (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 $\Lambda_\theta(f) \geq \tau$.*

Note that, for a fixed θ, if a function is τ-codomain separable, then it is also τ'-codomain separable for $0 < \tau' < \tau$. 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 τ-codomain separability, it suffices to set $\theta = [1, 2, 4, \ldots, 2^{N-1}]$. Using this, the output $f(\sigma, \theta)$ is the integer whose binary representation (in reverse) is given by the *bits* defined in σ. Moreover, since the difference between any two integers that can be represented this way is one, it also holds that $\Lambda_\theta(f) = 1$, which makes f admit 1-codomain separability using θ. Multiplying each entry in θ by τ will, thus, make f admit τ-codomain separability for any $\tau > 0$.

We highlight that in general, determining whether a loss function is codomain separable is co-NP-hard However, in the upcoming sections, we discuss our constructions that make many popular loss functions separable. In Appendix I, we present some function classes that are provably not τ-codomain separable.

Robust Label Inference. Next, we formally define the problem of (robust) label inference from functions defined over the domain $\mathbb{Z}_K^N \times \Theta$. Informally, the goal here is to recover some true labeling (in $\mathbb{Z}_K^N$) upon observing only the function output, even if noised. We consider recovery from noised (perturbed) function outputs in this paper, and the results trivially holds for the unnoised case.

Definition 2 (τ-Robust Label Inference)**.** *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$ be a function. Let $\sigma^\star \in \mathbb{Z}_K^N$ be an unknown true labeling. For a given $\tau > 0$, we say that f admits τ-robust label inference if there*

Algorithm LABELINF: Robust Label Inference for function $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$

Input: Number of datapoints N, bound on error (noise) $\tau > 0$
Output: Recovered labels $\hat{\sigma}$
Model Setting : A server (oracle) with private dataset with true labels $\sigma^\star$

1 Design a vector $\theta \in \Theta$ with respect to which f admits 2τ-codomain separability (Definition 1).
2 Query the server using θ as the prediction vector and observe output ℓ with $|f(\sigma^\star, \theta) - \ell| < \tau$.
3 **Return** $\hat{\sigma} \leftarrow \arg\min_{\sigma \in \mathbb{Z}_K^N} |f(\sigma, \theta) - \ell|$.

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 $\mathcal{A}(\theta, N, \ell) = \sigma^\star$.*

Observe that the above definition can also work for the unnoised case by setting τ arbitrarily small. An important point to note here is 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* [14], 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 will later relax this requirement to study τ-robust label inference under a multi-query model.

The following simple proposition formally establishes the connection between the above definitions of separability and robust label inference. The proposition is stated for the noised setting.

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

In a typical ML setting, assume that we have a server (machine) that holds a private labeled dataset with labels in $\sigma^\star \in \mathbb{Z}_K^N$. Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$ denote a (classification) loss function. In Algorithm LABELINF, we present a general label inference technique based on Proposition 1. The adversary picks $\theta \in \Theta$ based on Definition 1, to query the server, and gets back the noisy loss function value. It then iterates over the labelings to recover the closest one to the loss value. 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. We also assume that the loss is computed on all labels in $\sigma^\star$.

Remark 1. *A special case of Algorithm* LABELINF *is the unnoised setting, wherein $\ell = f(\sigma^\star, \theta)$. In that case, in Algorithm* LABELINF*, it suffices to design a vector $\theta \in \Theta$ with respect to which f admits 0^+-codomain separability and τ plays no role (i.e., can be set to an arbitrarily small positive constant).*

An important feature about Algorithm LABELINF 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 (see theorem below).[6] We also note that the optimization problem $\arg\min_{\sigma \in \mathbb{Z}_K^N} |f(\sigma, \theta) - \ell|$ can be solved naively by an exponential search over all σ in general, and faster for certain functions under multiple-queries as we discuss in Section C. Even the exponential search is a local computation for the adversary.

Theorem 2. *Let $f : \mathbb{Z}_K^N \times \Theta \to \mathbb{R}$. Algorithm* LABELINF *performs τ-robust label inference on f, in that it recovers $\sigma^\star$ (i.e., $\hat{\sigma} = \sigma^\star$) from ℓ where $|f(\sigma^\star, \theta) - \ell| < \tau$.*

A.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 [16, 13]. Following [8], we refer to the arbitrary precision arithmetic model as APA and floating point arithmetic model with ϕ bits as FPA(ϕ).

[6] To baseline Algorithm LABELINF with a naive exponential search over the space of all labelings, we simulated label inference on the Titanic dataset (see Section 3) for the cross-entropy loss, by replacing the prediction vector θ from Step 1 in Algorithm LABELINF by a uniformly random vector in $[0, 1]^N$. We observed that over 1000 independent runs, the randomly generated vector failed to correctly infer the true label for even a single datapoint at even mild noise levels ($\tau = 0.1$ and higher).

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 (formalized in Definition 3, Appendix D). Let f_ϕ be the representation of f in the FPA(ϕ) model.[7] 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, Appendix D). 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, Appendix D).

Moving to label inference, note that Algorithm LABELINF is also described in the APA model. The above mentioned bound on ϕ may not suffice for τ-robust label inference using Algorithm LABELINF in the FPA(ϕ) model. This is because unlike codomain separability, Algorithm LABELINF requires the adversary to compute the loss function on all labelings of the datapoints, which may require additional bits. For example, the prediction vector we provide for multiclass cross-entropy loss in Theorem 3 contains entries that are doubly-exponential in N (number of datapoints) and K (number of classes). This would prohibit the correct computation of loss scores beyond some values of N and K in the FPA(ϕ) model. However, this issue could be overcome by the ability to perform multiple queries that we utilize in our experimental evaluation in Section 3. For simplicity of description, we present our inference attacks under the APA model in the main body of this paper.

B τ-Robust Label Inference for Cross-Entropy Loss and its Extensions

We start with establishing τ-codomain separability for multiclass cross-entropy loss and its variants, for any $\tau > 0$. This implies that for these loss functions, Algorithm LABELINF succeeds in recovering all the labels when the loss scores are noised by less than τ in magnitude. We explain our main results at a high level and defer all missing details from this section to Appendix E.

Multiclass Cross-Entropy Loss. We first recall the definition of multiclass cross-entropy (extending Definition 2). 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 denoted by K-CELOSS (σ, θ) and defined as follows:

$$\text{K-CELOSS}\,(\sigma, \theta) := \frac{-1}{N} \sum_{i=1}^{N} \sum_{k=0}^{K-1} \Big([\sigma_i = k] \cdot \ln \theta_{i,k} \Big), \tag{3}$$

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 K-CELOSS function 2τ-codomain separable for any $\tau > 0$ (i.e., $\Lambda_\theta(\text{K-CELOSS}) \geq 2\tau$). At a high level, we obtain the required codomain separability for K-CELOSS by designing the entries in the matrix θ in such a way that the expression inside the logarithm in (3) 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* K-CELOSS *is 2τ-codomain separable using θ.*

The following corollary follows by combining the above result with Theorem 2.

Corollary 1. *Using θ from Theorem 3, Algorithm* LABELINF *performs τ-robust label inference from the K-ary cross-entropy loss (*K-CELOSS*).*

Note that Theorem 3 only holds for the error $\tau > 0$. However, as mentioned in Remark 1, in the unnoised case it suffices to instantiate Theorem 3 with an arbitrarily small positive value of τ.

[7] 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.

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, as shown in [8, 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), defined as follows for $\sigma \in \mathbb{Z}_K^N$, $\theta \in [0,1]^{N \times K}$:

$$\frac{-1}{N} \sum_{i=1}^{N} \sum_{k=0}^{K-1} \Big([\sigma_i = k] \cdot \ln \textsc{softmax}(\theta_{i,k}) \Big), \text{ where } \textsc{softmax}(\theta_{i,k}) = \exp(\theta_{i,k}) / \sum_{j=1}^{K} \exp(\theta_{i,j}).$$

We explain how to extend Theorem 3 to obtain separability in this setting. Our idea is the following: (1) Let $\theta \in [0,1]^{N \times K}$ be the matrix from Theorem 3. Recall that by definition, θ is such that for each $i \in [N]$, the vector $\theta_i \in [0,1]^K$ is a probability distribution; (2) For each $i \in [N]$, solve the following (fully specified) system of equations for $\theta'_{i,k}$'s: for all $k \in \mathbb{Z}_K, i \in [N]$: $\exp\left(\theta'_{i,k}\right) / \sum_{j=1}^{K} \exp\left(\theta'_{i,j}\right) = \theta_{i,k}$. This is a system of linear equations in the variables $\exp\left(\theta'_{i,j}\right)$, for each $i \in [N]$, hence, can be solved in polynomial time using standard techniques. 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'; Algorithm LABELINF can now be used for τ-label inference.

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 $\sigma \in \{0,1\}^N$, $\theta \in (0,1)^N$:

$$\frac{-1}{N} \sum_{i=1}^{N} \Big(\sigma_i \ln\left(\textsc{Sigmoid}(\theta_i)\right) + (1-\sigma_i) \ln\left(1 - \textsc{Sigmoid}(\theta_i)\right) \Big), \tag{4}$$

where $\textsc{Sigmoid}(x) = (1+e^{-x})^{-1}$ is the sigmoid function. Since $\textsc{Sigmoid} : \mathbb{R} \to (0,1)$ is a bijection (and hence, invertible), given $\textsc{Sigmoid}(x) = y$, 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 $\textsc{Sigmoid}(\theta_i) = \theta_{i,1}$ and $1 - \textsc{Sigmoid}(\theta_i) = \theta_{i,2}$ we get that sigmoid cross-entropy loss is 2τ-codomain separable using θ', and hence Algorithm LABELINF can be used for τ-robust label inference.

C τ-Robust Label Inference for Linearly-Decomposable Binary Losses

The analysis for cross-entropy loss and its extensions above can be generalized to determine τ-codomain separability for a broad class of loss functions, defined as follows. 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 in $\{0,1\}$ using g if it can be expressed in the following form:

$$f(\sigma, \theta) = \frac{1}{N} \left(\sum_{i \in [N]: \sigma_i = 1} g(\theta_i) + \sum_{i \in [N]: \sigma_i = 0} g(1-\theta_i) \right). \tag{5}$$

Our ideas from the previous section can be used to establish the following separability result for linearly-decomposable loss functions, hence, enabling the use Algorithm LABELINF for τ-robust label inference in these cases. Missing details are presented in Appendix F.1.

Theorem 4. *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 linearly-decomposable in $\{0,1\}$ using g. 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 $i \in [N]$.*

Some Examples of Linearly-Decomposable Loss Functions. Observe that the KL-divergence loss (which reduces to binary cross-entropy loss) is of this form, using $g(\theta_i) = -\ln \theta_i$ (see (7), Appendix F.1). Some other common examples of linearly-decomposable losses include the (i) Itakura-Saito divergence loss, which can be expressed using $g(\theta_i) = 1/\theta_i + \ln \theta_i - 2$ (see (8),

Appendix F.1); (ii) squared Euclidean loss, which can be expressed using $g(\theta_i) = (1-\theta_i)^2$, (see (9), Appendix F.1); and, (iii) norm-like divergence loss, which can be expressed using $g(\theta_i) = 1 + (\alpha-1)\theta_i^{\alpha} - \alpha\theta_i^{\alpha-1} + (\alpha-1)(1-\theta_i)^{\alpha}$ for some $\alpha \geq 2$ (see (11), Appendix F.1).

Connections to Bregman Divergence. The loss functions mentioned in (i)-(iii) above are some popular instances of the general class of Bregman divergence-based loss functions. Unlike the distance metrics for probability distributions, Bregman divergence [12] does not require its inputs to be necessarily distributions. Thus, to use these divergences as loss functions, one can directly compare the outputs of the ML model with the point distribution from the ground truth labels [17]. We defer the details on our general result for τ-codomain separability on Bregman divergence, including that for the Mahalanobis divergence loss (which is not linearly-decomposable) to Appendix F.2.

Applying Theorem 4. Observe that Theorem 4 immediately allows constructing a vector that will make a given linearly-decomposable function f admit τ-codomain separability for a given $\tau > 0$: (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 Algorithm LABELINF. However, such a θ might not exist for all loss functions, e.g., it does not exist for squared Euclidean and norm-like divergence losses, and we will deal those loss functions later. It does exist though for the Itakura-Saito divergence loss (defined in (8), Appendix F.1).

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

Multi-Query Polynomial Time Attacks. The linear-decomposability of f allows for an efficient multi-query label inference algorithm for f. In particular, we could have a trade-off between the ability to perform multiple queries with faster computation times for solving the optimization problem solved in Step 3 of Algorithm LABELINF. 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 $\theta = [\theta_1, \ldots, \theta_M, 1/2, \ldots, 1/2]$ to obtain:

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

Similar to Theorem 4, choosing θ_i that ensures $g(\theta_i) - g(1-\theta_i) > 2^i N\tau$ will ensure $\Lambda_\theta(f) > 2\tau$. Moreover, using this θ will ensure that if $f(\sigma_1, \theta) = f(\sigma_2, \theta)$, then $\sigma_1[:M] = \sigma_2[:M]$ must hold. Note that after recovering the first M labels, we can recover the next M labels using $\theta = [1/2, \ldots, 1/2, \theta_{M+1}, \ldots, \theta_{2M}, 1/2, \ldots, 1/2]$, and so on. The result below immediately follows.

Theorem 5. *Let $g : [0,1] \to \mathbb{R}$ be some finite deterministic function and $f : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be a loss function that is linearly-decomposable in $\{0,1\}$ using g. Then, for any $\tau > 0$ and integer $M \geq 1$, a sufficient condition for f to admit τ-robust label inference with $\lceil N/M \rceil$ queries is for the inequality $g(x) - g(1-x) > 2^i N\tau$ to have a solution for all $i \in [N]$.*

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 Algorithm LABELINF in each query). Thus, while the single query case ($M = N$) required $O(2^N)$ computations, any multi-query algorithm using $M = O(\log N)$ requires only $O(\text{poly}(N))$ local computations, which allow a polynomial-time robust label inference.

Linearly-Decomposable Functions Not Covered by Theorem 4. We now deal with linearly-decomposable functions for whom there exists no θ that could satisfy the condition required in Theorem 4 (examples include, squared Euclidean and norm-like divergence loss). Here, we establish the weaker 0^+-codomain separability, which as discussed earlier suffices for label inference in the unnoised case.

Theorem 6. *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 linearly-decomposable in $\{0,1\}$ using g. Let $\theta \in (0,1)^N$ be some vector and $S_\theta = \{g(\theta_i) - g(1-\theta_i)\}_{i=1}^N$. Then, the function f is 0^+-codomain separable with respect to θ if $\mu(S_\theta) > 0$, i.e., all subset sums in the set S_θ are distinct.*

Corollary 3. *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 $(1-\theta_i)^{\alpha-1} - \theta_i^{\alpha-1} = (\ln p_i)/(N\alpha)$.*

Finally, the idea behind the multi-query attack mentioned in Theorem 5 also extends to loss functions covered by Theorem 6, thus, providing a multi-query polynomial time inference here too.

D Missing Details from Section A

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 [8].

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 $\sigma^\star$ (which is how Algorithm LABELINF works). □

Details about the $\mathsf{FPA}(\phi)$ Model. For ease of discussion, as in [8], 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.

Definition 3 (τ-codomain Separability in the $\mathrm{FPA}(\phi)$ 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* $\mathrm{FPA}(\phi)$ *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* $\mathrm{FPA}(\phi)$ *model if $\Lambda_\theta(f_\phi) \geq \tau$.*

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

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* $\mathsf{FPA}(\phi)$ *model. If f_ϕ admits τ-codomain separability in the* $\mathsf{FPA}(\phi)$ *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* $\mathsf{FPA}(\phi')$ *model using θ for all $\phi' > \phi$.*

Proof. Given θ and the fact that f_ϕ admits τ-codomain separability using θ in the $\mathsf{FPA}(\phi)$ 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 $\mathsf{FPA}(\phi + 1)$ 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 $\mathsf{FPA}(\phi + 1)$ model. By induction, this implies that $f_{\phi'}$ admits τ-codomain separation in the $\mathsf{FPA}(\phi')$ 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 $\phi \geq \max(2\log_2 \tau + 5, -2\log_2 \tau - 1)$. □

E Missing Details from Section B

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 [8]. 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 (3), are distinct for all labelings. For example, if the true labeling is $[0, 2]$, then we obtain K-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 K-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 K-CELOSS score uniquely defined by the true labeling. This follows as the only thing that changes in the K-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 K-CELOSS *is 2τ-codomain separable using θ.*

Proof. We begin by simplifying the expression for K-CELOSS (θ, σ) (3) to write it as a sum of two terms: one dependent on the labeling σ, and the other independent of this labeling.

$$\text{K-CELOSS}\,(\theta,\sigma) = \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\Big([\sigma_i = k]\cdot \ln\theta_{i,k}\Big) = \frac{-1}{N}\left(\underbrace{\sum_{i=1}^{N}\ln\vartheta_{i,\sigma_i}}_{\substack{\text{Labeling Dependent}\\ \text{Term}}} - \underbrace{\sum_{i=1}^{N}\ln\left(\sum_{k=1}^{K}\vartheta_{i,k}\right)}_{\substack{\text{Labeling Independent}\\ \text{Term}}}\right). \tag{6}$$

Using (6), we then obtain the following:

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

Here, the third step follows from the fact that the label-independent term in the cross-entropy loss gets canceled when taking the difference inside the modulus. The last step follows from the fact that since $\sigma_1 \neq \sigma_2$, each summand inside the modulus represents a distinct even integer in $[2^{NK}]$, and hence, the difference is at least one. □

F Missing Details from Section C

Restatement of Theorem 4. *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 linearly-decomposable in $\{0,1\}$ using g. Then, for any $\tau > 0$, the function f is 2τ-codomain separable using any $\theta \in (0,1)^N$ for which $g(\theta_i) - g(1-\theta_i) > 2^i N\tau$ for all $i \in [N]$.*

Proof. We begin by observing that (5) can be rewritten as follows:

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

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

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

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 $\sigma_1 \neq \sigma_2$. □

Restatement of Theorem 6. *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 linearly-decomposable in $\{0,1\}$ using g. Let $\theta \in (0,1)^N$ be some vector and $S_\theta = \{g(\theta_i) - g(1-\theta_i)\}_{i=1}^N$. Then, the function f is 0^+-codomain separable with respect to θ if $\mu(S_\theta) > 0$, i.e., all subset sums in the set S_θ are distinct.*

Proof. From the definition of linear-decomposability, we can write:

$$\begin{aligned} f(\sigma,\theta) &= \frac{1}{N}\left(\sum_{i\in[N]:\sigma_i=1} g(\theta_i) + \sum_{i\in[N]:\sigma_i=0} g(1-\theta_i)\right) \\ &= \frac{1}{N}\sum_{i\in[N]} (\sigma_i g(\theta_i) + (1-\sigma_i)g(1-\theta_i)) = \frac{1}{N}\sum_{i\in[N]} (\sigma_i\,(g(\theta_i) - g(1-\theta_i)) + g(1-\theta_i)) \\ &= \frac{1}{N}\sum_{i\in[N]:\sigma_i=1} (g(\theta_i) - g(1-\theta_i)) + \frac{1}{N}\sum_{i=1}^N g(1-\theta_i). \end{aligned}$$

Using this reformulation, we can compute $\Lambda_\theta(f)$, for any $\theta \in (0,1)^N$, as follows:

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

Thus, ensuring that $S_\theta := \{g(\theta_i) - g(1-\theta_i)\}_{i=1}^N$ satisfies $\mu(S_\theta) > 0$ will ensure that $\Lambda_\theta(f) > 0$. □

F.1 Bregman Divergence-based Binary Losses

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 $[0,1]^N$.

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:

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

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

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

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

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

which is exactly the binary cross-entropy loss (2)[8] and is therefore covered by results in Section B.

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

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

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

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 4 can be applied to get the following result.

Restatement of Corollary 2. *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 4 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 4 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. □

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, there exists no θ such that the condition of Theorem 4 is satisfied. However, we can still use Theorem 6 to establish 0^+-codomain separability.

Restatement of the first part of Corollary 3. *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 6 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 11 that restricting θ to $\{0,1\}^N$ prohibits τ-codomain separability for any $\tau > 0$.

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:

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

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:

$$\begin{aligned} &D_{\text{NL}}([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i]) \\ &= \left(\sigma_i^\alpha + (\alpha-1)\theta_i^\alpha - \alpha \sigma_i \theta_i^{\alpha-1} + (1-\sigma_i)^\alpha + (\alpha-1)(1-\theta_i)^\alpha - \alpha(1-\sigma_i)(1-\theta_i)^{\alpha-1} \right). \end{aligned}$$

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

$$\begin{aligned} \text{NLLoss}(\sigma, \theta) = \frac{1}{N} \Big(&\sum_{i:\sigma_i = 1} \left(1 + (\alpha-1)\theta_i^\alpha - \alpha \theta_i^{\alpha-1} + (\alpha-1)(1-\theta_i)^\alpha \right) \\ &+ \sum_{i:\sigma_i = 0} \left(1 + (\alpha-1)(1-\theta_i)^\alpha - \alpha(1-\theta_i)^{\alpha-1} + (\alpha-1)\theta_i^\alpha \right) \Big). \end{aligned} \tag{11}$$

Again while Theorem 4 is not applicable in this case, Theorem 6 is applicable.

Restatement of the second part of Corollary 3. *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. □

F.2 A General Result for Bregman Divergence

Similar to the general construction of robust label inference for linearly-decomposable functions in Section C, we now demonstrate τ-robust label inference for a special class of Bregman divergences: the ones associated with functions F that are additive on the elements of its input vector.

Theorem 7. *Let $F : (0,1)^2 \to \mathbb{R}$ be a strictly convex function that can be written in the form $F(\mathbf{v}) = \sum_{i=1}^2 h(v_i)$ for some differentiable function $h : (0,1) \to \mathbb{R}$. Let D_F denote the Bregman divergence associated with F, and $f_F : \{0,1\}^N \times (0,1)^N \to \mathbb{R}$ be the corresponding loss function, defined as $f_F(\sigma, \theta) = (1/N) \sum_{i=1}^N D_F([\sigma_i, 1-\sigma_i], [\theta_i, 1-\theta_i])$. Then, for $\tau > 0$, the function f_F is 2τ-codomain separable if there exists $\theta \in (0,1)^N$ that satisfies $h'(\theta_i) + h'(1-\theta_i) > 2^i N \tau$ for all $i \in [N]$, where $h'(x) = dh/dx$.*

Proof. From the definitions of F and f_F in the theorem statement, we can simplify the expression for f_F as follows:

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

where $A_i = h'(\theta_i) + h'(1-\theta_i)$ and $B_i = h(0) + h(1) - (h(\theta_i) + h(1-\theta_i)) - \theta_i(h'(\theta_i) + h'(1-\theta_i))$. Using this, we can compute, for any $\theta \in (0,1)^N$ the following:

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

which follows by setting $A_i = 2^i N \tau (1+\delta)$ for some $\delta > 0$. □

Codomain Separability for Mahalanobis Divergence Loss. Not all Bregman divergences are additive in the sense of Theorem 7. A popular example of a Bregman divergence that is not additive is the Mahalanobis divergence, which, for vectors $\mathbf{p}, \mathbf{q} \in \mathcal{S} \subseteq \mathbb{R}^d$ is defined as:

$$D_{\text{MH}}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}(\mathbf{p} - \mathbf{q})^\top A(\mathbf{p} - \mathbf{q}),$$

where A is a positive definite matrix.

As in the case of other divergence, for binary classification, the Mahalanobis divergence loss can be defined as:

$$\text{MHLoss}(\sigma, \theta) = \frac{1}{N}\sum_{i=1}^{N}([\sigma_i, 1-\sigma_i] - [\theta_i, 1-\theta_i])^\top A([\sigma_i, 1-\sigma_i] - [\theta_i, 1-\theta_i]) \text{ where} A \in \mathbb{R}^{2\times 2}.$$

We discuss τ-robust label inference for the Mahalanobis loss function in the theorem below.

Theorem 8. *Let $A = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right] \in \mathbb{R}^{2\times 2}$ be a positive definite matrix with $(a+d) > (b+c)$. Denote $\alpha = a + d - (b + c)$ and let $\tau \in [0, \alpha/N)$. Then, for any $\tau > 0$, the Mahalanobis loss (defined above) is 2τ-codomain separable using any $\theta \in (0,1)^N$ with $\|\theta\|_\infty \leq (1/2)\,(1 - N\tau/\alpha)$.*

Proof. We begin by noticing that for $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, we can write $([\sigma_i, 1-\sigma_i] - [\theta_i, 1-\theta_i])^\top A([\sigma_i, 1-\sigma_i] - [\theta_i, 1-\theta_i]) = \alpha(\sigma_i - \theta_i)^2$. Thus, for a given $\theta \in (0,1)^N$, we can compute the magnitude of the difference $\text{MHLoss}(\sigma_1; \theta) - \text{MHLoss}(\sigma_2; \theta)$ as follows:

$$\begin{aligned}\Lambda_\theta(\text{MHLoss}) &= \min_{\sigma_1, \sigma_2 \in \{0,1\}^N} \frac{\alpha}{N}\left|\sum_{i=1}^{N}(\sigma_1(i) - \theta_i)^2 - (\sigma_2(i) - \theta_i)^2\right| \\ &= \min_{\sigma_1, \sigma_2 \in \{0,1\}^N} \frac{\alpha}{N}\left|\sum_{i:\sigma_1(i)\neq\sigma_2(i)}(1 - 2\theta_i)\right| = \min_{\sigma_1, \sigma_2 \in \{0,1\}^N}\left[\frac{\alpha}{N}\sum_{i:\sigma_1(i)\neq\sigma_2(i)}(1-2\theta_i)\right],\end{aligned}$$

where the last equation can be satisfied by restricting $\theta \in (0, 1/2)^N$. Now, since σ_1 and σ_2 must differ in at least one element, it holds that $\Lambda_\theta(\text{MHLoss}) \geq \min_i \frac{\alpha}{N}(1-2\theta_i)$. Thus, for 2τ-codomain separability, it suffices to set $\min_i \frac{\alpha}{N}(1-2\theta_i) \geq 2\tau$, which gives $\max_i \theta_i \leq \frac{1}{2} - \frac{N\tau}{\alpha}$. □

G A Neural Network to Imitate Robust Label Inference

The results in Sections B and C highlight how prediction vectors (θ's) could be generated that succeed with τ-robust label inference. Typically, these prediction vectors are output of a ML model, and in many scenarios (e.g., ML contests held by Kaggle and other platforms) the submission to the server is not the prediction vector but rather the ML model itself. This raises an interesting question, whether the prediction vectors utilized in these label inference attacks can actually be an output of a "natural" looking ML model?[9]

Formally, we ask: Is there a feed-forward neural network, that for any input $\mathbf{v}$ recovers the ground truth $\sigma_{\mathbf{v}}$ through only one query to the server for the loss function value? We now show that this is possible for any codomain separable loss function.

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 Algorithm LabelInf to infer $\sigma_{\mathbf{v}}$.

[9] Technically, we can have a constant function that on all inputs generate the prediction vectors required for the attack. However, that we all can agree is not a "natural" ML model.

Our Mutator Network. We construct a 2-layer network $\text{MUTNET}_{\mathbf{x}} : \mathbb{R}^{d_1} \to (0,1)^{d_2}$ 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 $\mathbf{v}$, the transformations in $\text{MUTNET}_{\mathbf{x}}$ are as follows:

$$\mathbf{u}_1 = \mathbf{v}, \;\; \mathbf{u}_2 = \text{RELU}\left(\mathbf{v}^\top M_1\right), \;\; \mathbf{u}_3 = \; \text{SIGMOID}\left(\mathbf{u}_2^\top M_2 + \mathbf{x}'\right), \tag{12}$$

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 (i.e., the output of $\text{MUTNET}_{\mathbf{x}}$ equals $\mathbf{x}$ for all inputs $\mathbf{v}$). By setting $\mathbf{x}$ to the desired (prediction vector) θ with respect to which f admits 2τ-codomain separability, we get the following result

Theorem 9. *Let $\tau > 0$ and let $\theta \in (0,1)^{d_2}$ be such that $f : \{0,1\}^{d_2} \times (0,1)^{d_2} \to \mathbb{R}$ is 2τ-codomain separable using θ. Then, for any input $\mathbf{v} \in (0,1)^{d_2}$, given ℓ such that $|f(\sigma_{\mathbf{v}}, \text{MUTNET}_\theta(\mathbf{v})) - \ell| \leq \tau$, Algorithm* LABELINF *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 $\mathbf{x}'$ in (12) is constructed in a way such that $\text{SIGMOID}(\mathbf{x}') = \theta$ (element -wise). □

H Computational Hardness of Codomain Separability

We now show that that determining the codomain separability is co-NP-hard.[10] 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-SEPARABILITY. 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 1. 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 [9]. 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 [18, 10], 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

[10] See [9] for a definition of co-NP-hard.

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:

$$f_\Phi(\sigma, \theta) = 5^{\sigma_1} 7^{\sigma_2} 11^{\sigma_3} (1 - \sigma_1 + \sigma_2 + \sigma_3)^{\theta_1} (1 + \sigma_1 + \sigma_2 - \sigma_3)^{\theta_2 + \theta_3}. \tag{13}$$

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

Lemma 2. *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 10. CODOMAIN-SEPARABILITY *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 1. Let $\Phi(x_1, \ldots, x_N)$ be an arbitrary Boolean formula and $f_\Phi(\sigma, \theta)$ be the corresponding function from (13). 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 2, 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. □

I 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 11. *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 12. *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:

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

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

Corollary 4. *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*

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

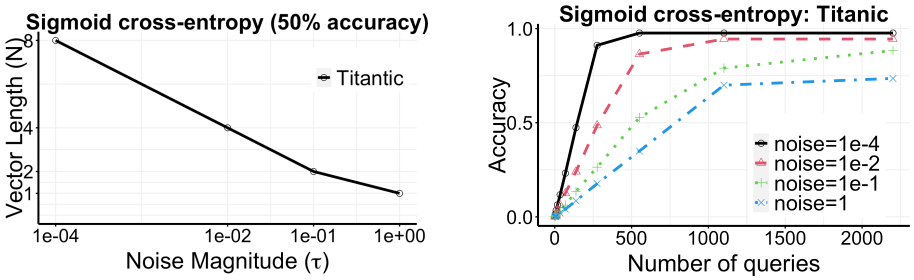
In particular, if $f(\emptyset, \theta) = 0$ for all $\theta \in \mathbb{R}^N$, then f is not τ-codomain separable for any

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

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

Theorem 13. *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 $\tau > \beta/N$.*

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. □



(a) (a) (b) (b)

Figure 2: The plot on the left shows results on the label inference on sigmoid cross-entropy loss using Algorithm LABELINF over 1000 runs. We plot the length for which Algorithm LABELINF always succeeds with at least 50% accuracy. The plot on the right shows label reconstruction accuracy with multi-query label inference attack on sigmoid cross-entropy by performing Algorithm LABELINF on M data points at a time using a total of $\lceil N/M \rceil$ queries (where $N = 2201$ is the number of rows in the Titanic dataset).

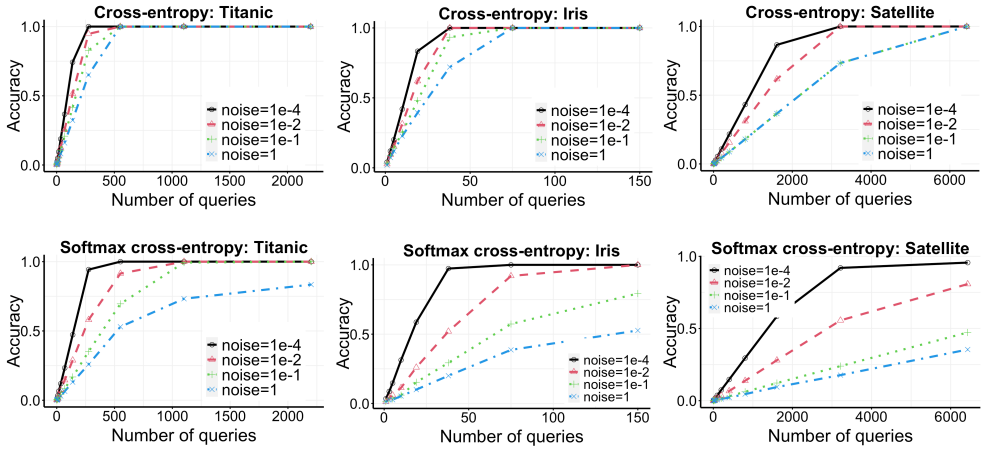


Figure 3: Label reconstruction accuracy with multi-query label inference attack by performing Algorithm LABELINF on M datapoints at a time using a total of $\lceil N/M \rceil$ queries. Here, N is the total number of rows in the corresponding dataset (2201 for Titanic, 150 for Iris, and 6430 for Satellite).

J Additional Experimental Results

We now discuss the missing details from Section 3 and present our results for label inference from the Sigmoid cross entropy loss function.

We begin with examining a multi-query label inference algorithm in Figure 3. For these plots, we simulated Algorithm LABELINF on $M < N$ datapoints at a time (instead of all N datapoints at once), 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 cross-entropy loss on Titanic, we achieved 100% accuracy using $M \geq 220$ with $\tau = 0.0001$, and $M \geq 550$ with $\tau = 1$. The accuracy is again lower for the softmax case again due to reasons mentioned above.

Label Inference from Binary Cross-Entropy. In our experiments, for the binary label case, we use the label inference attack of Aggarwal *et al.*[8] as a baseline (see Figures 1(a) and (b)). We begin by restating the label inference result of Aggarwal *et al.*[8].

Theorem 14 (τ-codomain Separability from Algorithm 2 in [8])**.** *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(c).

Additional bits Needed for Softmax Cross-Entropy Loss. Recall from our discussion in Section 3 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 regular cross-entropy.

Experimental Results for Label Inference from Sigmoid Cross-Entropy Loss. Similar to the softmax cross-entropy loss, we empirically examine label inference from the sigmoid activation performed prior to the cross-entropy calculation (see Figure 2). Note that we do this only for the binary label case (see (4)). We note that while (4) guarantees 100% accurate inference theoretically, this was not observed during our experiments (Figure 2(a)). This is because the number of bits required to represent $\textsc{Sigmoid}^{-1}(\theta_i)$ (where θ_i is an element of θ) is $\Omega\left(\ln\left|\ln(\theta_i/(1-\theta_i))\right|\right)$, which asymptotically dominates the $\Omega\left(\left|\ln\theta_i\right|\right)$ many bits required to represent θ_i for the cross-entropy loss.