Appendix for "Label Inference Attacks from Log-loss Scores"

In this technical appendix, we present the missing details and omitted proofs from the main body of our paper.

A. Missing Details from Section 3

A.1. Label Inference under Bounded Precision with Polynomial-Time Adversary

For our results in this section, we make use of the well-known Prime Number Theorem (PNT), which describes the asymptotic distribution of the prime numbers among the positive integers. We use this result to quantify the number of queries required for our label inference attack in the FPA(ϕ) model. The form of PNT which will be helpful for our analysis is stated as follows (we state it as a Lemma for use in our proof for the main result in this section).

Lemma 3. *(Hadamard, 1896; Poussin, 1897) The n^{th} prime number p_n satisfies $p_n \sim n \log n$, where $\sim$ means that the relative error of this approximation approaches 0 as n increases.*

We note that this bound on p_n implies that $p_n = \Theta(n \log n)$ also holds. We begin with proving the following lemma.

Lemma 4. *Let $M \leq N$ be a positive integer and $\mathbf{v} = [v_1, \ldots, v_M, 1, \ldots, 1] \in (\mathbb{R}^+)^N$ be a real valued vector. Then, for any labeling $\sigma \in \{0,1\}^N$, it holds that:*

$$\mathcal{L}_{\mathbf{v}}(\sigma) = -\frac{1}{N} \ln \left(\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i \right) + \textit{constant},$$

where the constant term is independent of σ.

Proof. From the definition of log-loss in Equation (2), we compute the following:

$$\begin{aligned} N\mathcal{L}_{\mathbf{v}}(\sigma) &= -\ln \left(\frac{\prod_{i:\sigma_i=1} v_i}{(1+v_1)\cdots(1+v_N)} \right) \\ &= -\ln \left(\frac{\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i}{2^{N-M}(1+v_1)\cdots(1+v_M)} \right) = -\ln \left(\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i \right) + \textit{constant}, \end{aligned}$$

where *constant* $= (N-M)\ln 2 + \sum_{j=1}^{M} \ln(1+v_j)$. Dividing both sides by N, we obtain the desired result. □

A.2. Extension to the Multiclass Case

We begin by stating our result for the single-query label inference in the APA model.

Theorem 8. *There exists a polynomial-time adversary for K-ary label inference in the* APA *model using only a single log-loss query.*

Proof. We only focus on $K \geq 3$ (since $K = 2$ is equivalent to Theorem 1). Let $\sigma^* \in [K]$ be the true labeling.

Define the matrix $\mathbf{v}$ as follows:

$$\mathbf{v}_{i,k} = \frac{p_i^{k-1}}{\sum_{j=1}^{K} p_i^{j-1}} \qquad \forall (i,k) \in [N] \times [K].$$

We can perform the following algebraic manipulation of the log loss (using Definition 1) as follows:

$$
\begin{aligned}
\textsc{LLoss}\left(\mathbf{v};\sigma^*\right) &= \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{v}_{i,k}\right) = \frac{-1}{N}\ln\left(\prod_{i=1}^{N}\mathbf{v}_{i,\sigma_i^*}\right)\\
&= \frac{-1}{N}\ln\prod_{i=1}^{N}\left(\frac{p_i^{\sigma_i^*-1}}{\sum_{j=1}^{K}p_i^{j-1}}\right) = \frac{-1}{N}\ln\left(\frac{\prod_{i=1}^{N}p_i^{\sigma_i^*-1}}{\prod_{i=1}^{N}\sum_{j=1}^{K}p_i^{j-1}}\right).
\end{aligned}
$$

Rearranging the terms above, we obtain the following:

$$
\prod_{i=1}^{N}p_i^{\sigma_i^*-1} = \exp\left(-N\cdot\textsc{LLoss}\left(\mathbf{v};\sigma^*\right)\right)\left(\prod_{i=1}^{N}\sum_{j=1}^{K}p_i^{j-1}\right).
$$

Thus, similar to the binary case, from the Fundamental theorem of arithmetic, given the value on the right hand side above, we can uniquely factorize this value to obtain the true labels on the left (which can be done efficiently since the list of primes and the number of classes are known). □

To extend this algorithm for the $\mathsf{FPA}(\phi)$ model, we first make some important observations about using multiple queries for label inference (similar to the binary case). We analyze the case where $K < N$ and focus only on inferring the first set of labels (the analysis for other cases follow using similar principles). We let $m < K$ denote an upper bound on the number of labels that we will infer in a single query (note that previously, m was the exact number of labels inferred per query). Moreover, for simplicity, we will assume that m divides both K and N. This will not change the asymptotic number of queries required by our inference attack.

Define the matrices $\mathbf{v}^{(m)}$ and $\mathbf{u}^{(m)}$ as follows:

$$
\begin{aligned}
\mathbf{v}_{i,k}^{(m)} &= \begin{cases} p_i^{k-1} & \forall(i,k)\in[m]\times[m]\\ 1 & \text{otherwise.}\end{cases}, \text{ and}\\
\mathbf{u}_{i,k}^{(m)} &= \frac{\mathbf{v}_{i,k}^{(m)}}{\sum_{j=1}^{K}\mathbf{v}_{i,j}^{(m)}} = \begin{cases} \frac{p_i^{k-1}}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)} & \forall(i,k)\in[m]\times[m]\\ \frac{1}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)} & \text{otherwise.}\end{cases}
\end{aligned}
$$

Now, substituting this in Definition 1, we obtain the following:

$$
\begin{aligned}
\textsc{LLoss}\left(\mathbf{u}^{(m)};\sigma^*\right) &= \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right)\\
&= \frac{-1}{N}\sum_{i=1}^{m}\sum_{k=1}^{m}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right) + \frac{-1}{N}\sum_{i=m+1}^{N}\sum_{k=m+1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right)\\
&= \frac{-1}{N}\sum_{i=1}^{m}\ln\left(\frac{p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)}\right) - \frac{1}{N}\sum_{i=m+1}^{N}\ln\left(\frac{1}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)}\right)\\
&= \frac{1}{N}\sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m}p_i^{j-1}+(K-m)\right) - \frac{1}{N}\sum_{i=1}^{m}\ln\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right) + \frac{(N-m)\ln K}{N}\\
&= \frac{1}{N}\sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m}p_i^{j-1}+(K-m)\right) - \frac{1}{N}\ln\left(\prod_{i=1}^{m}\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right)\right) + \frac{(N-m)\ln K}{N}.
\end{aligned}
$$

Thus, when the score $\ell^{(m)} := \textsc{LLoss}\left(\mathbf{u}^{(m)};\sigma^*\right)$ is observed, the adversary can compute the following:

$$
\prod_{i=1}^{m}\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right) = \exp\left(-N\ell^{(m)} + (N-m)\ln K + \sum_{i=1}^{m}\ln\alpha(i,m,K)\right),
$$

where $\alpha(i, m, K) = \sum_{j=1}^{m} p_i^{j-1} + (K - m)$. Using the Fundamental Theorem of Arithmetic, the product on the left will allow recovering labels for datapoints that have labels in $[m]$.

Restatement of Theorem 3. *There exists a polynomial-time adversary for K-ary label inference in the* APA *model using only a single log-loss query. For inference in the* $\mathsf{FPA}(\phi)$ *model, it suffices to issue* $O\left(1 + NKh(\phi)\right)$ *queries, where*

$$h(\phi) = O\left(\frac{(\ln \phi)^2}{\left(\phi + (N-K)\ln K\right)^{2/3}}\right).$$

Proof. The largest value of m that works in the $\mathsf{FPA}(\phi)$ model can be obtained (asymptotically) by setting $(\phi - 1)/2 \geq \log_2\left(\prod_{i=1}^{m}\left(p_i^{\sigma_i^* - 1} \cdot [\sigma_i^* \leq m]\right)\right)$, which will allow enough resolution for this disambiguation. Simplifying this, we obtain:

$$\begin{aligned}
\phi &\geq 1 + 2\log_2\left(\exp\left(-N\ell^{(m)} + (N-m)\ln K + \sum_{i=1}^{m}\ln\alpha(i, m, K)\right)\right) \\
&\geq 1 + \frac{2}{\ln 2}\left((N-K)\ln K + \sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m} p_i^{j-1}\right)\right) \\
&\geq 1 + \frac{2}{\ln 2}\left((N-K)\ln K + (m-1)\sum_{i=1}^{m}\ln p_i\right) \\
&\gtrsim 3\left((N-K)\ln K + (m-1)\sum_{i=1}^{m} i\ln i\right) \quad \text{(Using the Prime Number Theorem – see Lemma 3)} \\
&\geq c\left((N-K)\ln K + m^3\ln m\right) \qquad \text{for some constant } c > 0.
\end{aligned}$$

This gives an upper bound on m by simplifying $m^3 \ln m \leq \phi/c - (N-K)\ln K$, which gives $m \ln m \leq \left(\frac{\phi/c-(N-K)\ln K}{3}\right)^{1/3}$, or equivalently, $m \leq c'\left(\frac{(\phi+(N-K)\ln K)^{1/3}}{\ln \phi}\right)$ for some constant $c' > 0$ (here we use the fact that $x \ln x = y$ holds when $x = \Theta(y/\ln y)$). Setting $m = m^*$, where m^* is this upper bound, we can then obtain the number of queries as $NK/(m^*)^2$, which gives $h(\phi) = 1/(m^*)^2$, or equivalently,

$$h(\phi) = O\left(\frac{(\ln \phi)^2}{\left(\phi + (N-K)\ln K\right)^{2/3}}\right).$$

□

We defer optimization of this algorithm and the detailed discussion of Robust Label Inference for multi-class classification to future work.

B. Missing Details from Section 4

B.1. τ-Robust Label Inference under Arbitrary Precision with Exponential-time Adversary

Restatement of Lemma 1. *Let* $\mathbf{v} = [v_1, \ldots, v_N]$ *be a vector with all entries distinct and positive. Define* $\ln \mathbf{v} := [\ln v_1, \ldots, \ln v_N]$*. Then, it holds that* $\Delta(\mathbf{v}) = \frac{1}{N}\mu(\ln \mathbf{v})$.

Proof. Without loss of generality, assume that σ_1 and σ_2 are such that $\mathcal{L}_{\mathbf{v}}(\sigma_1) \geq \mathcal{L}_{\mathbf{v}}(\sigma_2)$. This happens when the following holds (from Equation (2)):

$$\mathcal{L}_{\mathbf{v}}(\sigma_1) \geq \mathcal{L}_{\mathbf{v}}(\sigma_2) \iff \prod_{i:\sigma_1(i)=1} v_i \leq \prod_{j:\sigma_2(j)=1} v_j.$$

Under this condition, we can write the following:

$$
\begin{aligned}
N\Delta(\mathbf{v}) &= \min_{\sigma_1,\sigma_2\in\{0,1\}^N} N\left(\mathcal{L}_{\mathbf{v}}\left(\sigma_1\right) - \mathcal{L}_{\mathbf{v}}\left(\sigma_2\right)\right) = \min_{\substack{\sigma_1,\sigma_2\in\{0,1\}^N\\ \sigma_1\neq\sigma_2}} \ln\exp\left(N\mathcal{L}_{\mathbf{v}}\left(\sigma_1\right) - N\mathcal{L}_{\mathbf{v}}\left(\sigma_2\right)\right)\\
&= \min_{\sigma_1,\sigma_2\in\{0,1\}^N} \ln\left(\frac{\exp\left(-N\mathcal{L}_{\mathbf{v}}\left(\sigma_2\right)\right)}{\exp\left(-N\mathcal{L}_{\mathbf{v}}\left(\sigma_1\right)\right)}\right) = \min_{\sigma_1,\sigma_2\in\{0,1\}^N} \ln\left(\frac{\prod_{j:\sigma_2(j)=1} v_j}{\prod_{i:\sigma_1(i)=1} v_i}\right)\\
&= \min_{\sigma_1,\sigma_2\in\{0,1\}^N}\left(\sum_{j:\sigma_2(j)=1}\ln v_j - \sum_{i:\sigma_1(i)=1}\ln v_i\right) = \mu\left(\ln\mathbf{v}\right),
\end{aligned}
$$

where the last step follows from interpreting σ_1 and σ_2 as selecting elements from $\ln\mathbf{v}$ (by choosing which elements get labeled 1 and the ones that do not). □

Restatement of Theorem 4. *For any $\tau > 0$, there exists an exponential-time adversary (from Algorithm 2) for the τ-robust label inference problem in the APA model using only a single log-loss query.*

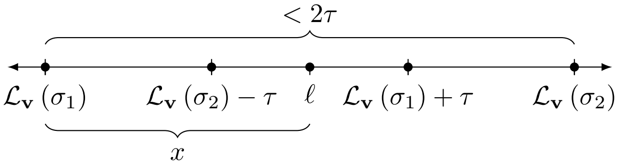
Proof. Let $S = \{s_1, \ldots, s_N\}$ be a set such that $\mu(S) \geq 2N\tau$. We know that S exists, for example, by scaling each element of $\mathcal{S}_\circ$ by $2N\tau$. Let $\mathbf{v}$ be constructed such that $v_i = 3\exp(s_i)$. To see that $\mathbf{v}$ is τ-robust, it suffices to prove that $\Delta(\mathbf{v}) > 2\tau$. Now, observe that $(\ln 3)s_i = \ln\mathbf{v}_i$. From Lemma 1, we get $\Delta(\mathbf{v}) = \frac{\ln 3}{N}\mu(S) = (2\ln 3)\tau > 2\tau$. □

B.2. Optimality of Single-Query τ-Robust Inference

We begin with the following two lemmas.

Lemma 5. *A vector $\mathbf{v}$ is τ-robust if and only if $\Delta(\mathbf{v}) > 2\tau$.*

Proof. It suffices to prove that for any $\Delta(\mathbf{v}) > 2\tau$ for any τ-robust vector $\mathbf{v}$ (since the other direction follows from our construction in Algorithm 2). We prove this by contradiction. The idea is to construct a score from which a unique labeling cannot be unambiguously derived. Let $\mathbf{v}$ be a τ-robust vector and $\mathcal{A}$ be a Turing Machine such that $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma$ for all $\sigma \in \{0,1\}^N$ and all ℓ that satisfies $|\ell - \mathcal{L}_{\mathbf{v}}(\sigma)| \leq \tau$. Without loss of generality, let σ_1, σ_2 be two distinct labelings for which $0 < \mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1) < 2\tau$. It follows that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \tau < \mathcal{L}_{\mathbf{v}}(\sigma_1) + \tau$ (see schematic below).


$< 2\tau$
$\mathcal{L}_{\mathbf{v}}(\sigma_1)$
$\mathcal{L}_{\mathbf{v}}(\sigma_2) - \tau$
ℓ
$\mathcal{L}_{\mathbf{v}}(\sigma_1) + \tau$
$\mathcal{L}_{\mathbf{v}}(\sigma_2)$
x

Let $\ell = \left(\mathcal{L}_{\mathbf{v}}(\sigma_1) + \mathcal{L}_{\mathbf{v}}(\sigma_2)\right)/2$ and $x = \ell - \mathcal{L}_{\mathbf{v}}(\sigma_1)$. Clearly, $x < \tau$, and thus, $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma_1$. However, we can show similarly that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \ell < \tau$, and hence $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma_2$. This is a contradiction since $\sigma_1 \neq \sigma_2$, and hence, it must be true that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1) > 2\tau$. The result in the Lemma statement then follows by observing that if this condition holds for any σ_1, σ_2, then $\Delta(\mathbf{v}) = \min_{\sigma_1,\sigma_2}|\mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1)| > 2\tau$ as well. □

Lemma 6. *For every τ-robust vector $\mathbf{v} = [v_1, \ldots, v_N]$ with distinct entries in $(1, \infty)$, it is possible, for every $s > 0$, to construct a set S such that $\mu(S) > s$.*

Proof. From Lemma 1, we know that $\Delta(\mathbf{v}) = \mu(\ln\mathbf{v})/N$. Since $\mathbf{v}$ is τ-robust, it must be true that $\Delta(\mathbf{v}) > 2\tau$ (by Lemma 5), from which we obtain that $\mu(\ln\mathbf{v}) > 2N\tau$. The result in the lemma statement then follows by setting $S = \{s_1, \ldots, s_N\}$, where $s_i = (s\ln v_i)/(2N\tau)$. □

Restatement of Theorem 5. *For any set $S \subset \mathbb{Q}^+$ with $\mu(S) > \lambda$ for some $\lambda \in [0, \infty)$, it holds that $||S||_\infty = \Omega(\lambda 2^{|S|})$.*

Proof. We prove this result in three steps. First, we show that the bound holds whenever λ as well as all the elements in S are positive integers. To see this, observe that if $\mu(S) > 1 > 0$, Euler's result gives us that $||S||_\infty = \Omega(2^{|S|})$. Now, let

$S' = \{s\lambda \mid s \in S\}$. Then, $\mu(S') > \lambda$. If we suppose that $||S'||_\infty = o(\lambda 2^{|S|})$, then $||S'/T||_\infty = ||S||_\infty = o(2^{|S|})$, which is a contradiction to above.

Next, assume that $S \subset \mathbb{Q}^+$ and λ still be an integer. In this case, each element of S can be written as $s_i = p_i/q_i$ (in the lowest form). Let $Q = q_1 q_2 \cdots q_N$ and $S' = \{sQ \mid s \in S\}$. Then, each element of S' is an integer and since $\mu(S') > \lambda Q$, which is also an integer, from the discussion above, we have $||S'||_\infty = \Omega(\lambda Q 2^N)$, which gives $||S||_\infty = \Omega(\lambda 2^N)$.

Finally, let λ be an arbitrary positive real. In this case, $\mu(S) > \lambda$ implies $\mu(S) > \lfloor \lambda \rfloor$, which is an integer. Hence, $||S||_\infty = \Omega(\lfloor \lambda \rfloor 2^N) = \Omega(\lambda 2^N)$, as desired. □

Restatement of Theorem 6: *For sufficiently large N and all $\tau > 0$, any τ-robust vector $\mathbf{v}$ must have $||\mathbf{v}||_\infty = \Omega\left(e^{2^N N\tau}\right)$.*

Proof. From Lemma 1, we know that for any vector $\mathbf{v}$ with distinct positive entries, it holds that $\Delta(\mathbf{v}) = \mu(\ln \mathbf{v})/N$. For this vector to be τ-robust, we argue that $\Delta(\mathbf{v}) > 2\tau$ (see Lemma 5), which is the same as setting $\mu(\ln \mathbf{v}) > 2N\tau$. From Theorem 5, for this to hold, it must be true that $||\ln \mathbf{v}||_\infty = \Omega\left(2^N N\tau\right)$. From the definition of $\ln \mathbf{v}$, this gives $||\mathbf{v}||_\infty = \Omega\left(\exp\left(2^N N\tau\right)\right)$, as desired. □

B.3. τ-Robust Label Inference under Bounded Precision with Polynomial-time Adversary

We prove an additional lemma before proving our next result.

Lemma 7. *Let $\tau > 0$ be a bound on the resulting error and $m \le N \in \mathbb{Z}^+$ be an integer. Let $\mathbf{v}_m = \left[3e^{2m\tau}, 3e^{4m\tau}, \ldots, 3e^{2^m m\tau}, 1, \ldots, 1\right]$. Then, for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$ where $\sigma_1[:m] \neq \sigma_2[:m]$ (*i.e.* σ_1 and σ_2 differ some index $i \le m$), it holds that:*

$$|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| > 2m\tau/N.$$

Proof. From Lemma 4, we can write that:

$$\begin{aligned}
|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| &= \frac{1}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\le i\le m}} \ln v_m(i) - \sum_{\substack{k:\sigma_2(k)=1\\1\le k\le m}} \ln v_m(k)\right| \\
&= \frac{\ln 3}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\le i\le m}} 2^i m\tau - \sum_{\substack{k:\sigma_2(k)=1\\1\le k\le m}} 2^k m\tau\right| = \frac{m\tau \ln 3}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\le i\le m}} 2^i - \sum_{\substack{k:\sigma_2(k)=1\\1\le k\le m}} 2^k\right| > \frac{2m\tau}{N},
\end{aligned}$$

where the last step follows from the fact that σ_1 and σ_2 differ in at least one element amongst the first m entries. □

Restatement of Lemma 2. *Let $\tau > 0$ be a bound on the resulting error and $m \le N \in \mathbb{Z}^+$ be an integer. Let $\mathbf{v}_m = \left[3e^{2N\tau}, 3e^{4N\tau}, \ldots, 3\exp\left(2^m N\tau\right), 1, \ldots, 1\right]$. Then, for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$, the following hold:*

(a) If $\sigma_1[:m] = \sigma_2[:m]$, then $\mathcal{L}_{\mathbf{v}_m}(\sigma_1) = \mathcal{L}_{\mathbf{v}_m}(\sigma_2)$.

(b) Else, we have $|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| > 2\tau$.

The proof directly follows from Lemma 7.

Restatement of Theorem 7. *For any error bounded by $\tau > 0$ and $\phi \ge 8 + \lceil N\tau \ln 2 \rceil$, there exists a polynomial-time adversary (from Algorithm 3) for the τ-label inference problem in the* FPA(ϕ) *model using $O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$ queries.*

Proof. To see how many queries suffice, we first compute the number of bits necessary to represent the prediction vector and the loss scores up to sufficient resolution. For $\mathbf{u}_m = f(\mathbf{v}_m)$, we observe the following for sufficiently large N, in

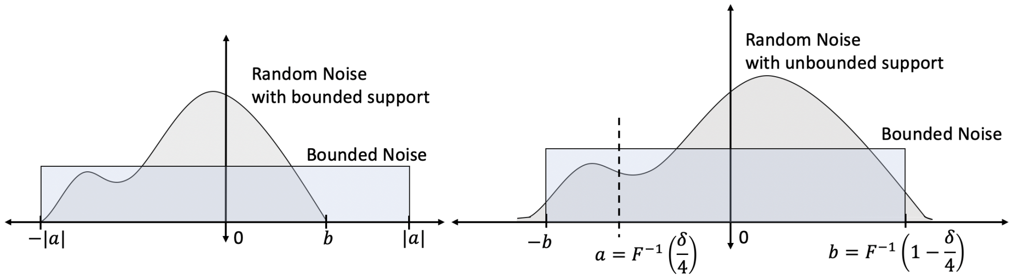


Figure 3. Schematic of the reduction of the random noise case to bounded noise for label inference. The picture on the left allows for a construction of a robust vector, whereas the picture on the right allows for robust vectors.

particular, when $N\tau > \ln(16)$:

$$\min_{\substack{i,j\in[N]\\ \mathbf{u}_m(i)\neq\mathbf{u}_m(j)}} |\mathbf{u}_m(i) - \mathbf{u}_m(j)| = \frac{3e^{2^m N\tau}}{1+3e^{2^m N\tau}} - \frac{3e^{2^{m-1} N\tau}}{1+3e^{2^{m-1} N\tau}}$$
$$\geq \frac{\exp\left(-2^{m-1}N\tau\right)}{16},$$

which, to work within the FPA(ϕ) model, would require $(\phi - 1)/2 \geq 4 + 2^{m-1}N\tau\ln 2$, or equivalently, $\phi \geq 8 + 2^m N\tau \ln 2$ bits. Moreover, we can bound the loss computed on $\mathbf{v}_m$ on any σ as follows:

$$\max_{\sigma\in\{0,1\}^N} \mathcal{L}_{\mathbf{v}_m}(\sigma) \leq \frac{1}{N}\ln\left(2^{N-m}\prod_{j=1}^{m}(1+3e^{2^j N\tau})\right)$$
$$\leq 2\left(\ln 2 + (2^m - 1)\tau\right) \leq 2^{m+1}\tau,$$

which can be represented (along with sufficient resolution, since $\Delta(\mathbf{v}_m) \geq 2\tau$ by construction) within the number of bits described above. Thus, in the FPA(ϕ) model, the maximum value of m that we can set is obtained by setting $2^m N\tau\ln 2 + 8 \leq \phi$, which gives:

$$m_{max} = \left\lfloor \log_2\left(\frac{\phi - 8}{N\tau\ln 2}\right)\right\rfloor.$$

From this, we obtain that $\frac{N}{m_{max}} = O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$ queries suffice.

Given this bound, it suffices to show that in the i^{th} iteration of the for-loop in Algorithm 3, the vector σ' contains the true labels for indices in $\{(i-1)m+1,\ldots,im\}$. Without loss of generality, assume $i = 1$, so that the vector $\mathbf{v} = \left[3e^{2N\tau}, 3e^{4N\tau}, \ldots, 3\exp\left(2^m N\tau\right), 1, \ldots, 1\right]$. From Lemma 2(b), it follows that for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$ where $\sigma_1[:m] \neq \sigma_2[:m]$, it holds that:

$$|\mathcal{L}_{\mathbf{v}}(\sigma_1) - \mathcal{L}_{\mathbf{v}}(\sigma_2)| > 2\tau. \tag{3}$$

Thus, when the loss ℓ is observed on $\mathbf{u} = f(\mathbf{v})$ and $\arg\min_\sigma |\mathcal{L}_{\mathbf{u}}(\sigma) - \ell| = \{\sigma^{(1)},\ldots,\sigma^{(k)}\}$, then it must be true that $\sigma^{(k_1)}[:m] = \sigma^{(k_1)}[:m]$ for all $k_1, k_2 \in [k]$ (or else, it would contradict the inequality in (3)). Furthermore, from Lemma 2(a), it follows that the true labeling must be present in the set $\{\sigma^{(1)},\ldots,\sigma^{(k)}\}$. Thus, at the end of iteration $i = 1$, the vector $\hat{\sigma}$ has recovered the first m bits in σ. For all other iterations, note that the vector $\mathbf{v}$ is just a cyclic rotation to the right by m elements, and hence, the bits in σ are recovered, m at a time, in Algorithm 3. □

B.4. Extension to Other Noise Models: Random Noise

The discussion of norm-bounded noise above allows easy extension to handling randomly generated (additive) noise. We achieve this by safeguarding against the worst case magnitude of the noise that can be added for bounded noise distributions. For cases where this distribution is unbounded, we allow for some error tolerance. We begin with formally defining the label inference problem in this setting and then present our analysis. We will restrict ourselves to arbitrary precision in this section and defer the extension to FPA(ϕ) for future work.

Definition 6. *Let $\mathcal{D}$ be a probability distribution and $\delta \in [0,1]$ be the error tolerance. We say that a vector $\mathbf{v}$ is $\mathcal{D}$-robust if for all $\tau \sim \mathcal{D}$ and $\sigma \in \{0,1\}^N$, there exists an adversary (Turing Machine) $\mathcal{A}$ that can recover (within a single query) σ from $\ell = \mathcal{L}_{\mathbf{v}}(\sigma) + \tau$ with probability at least $1-\delta$,* i.e. $\Pr\left[\mathcal{A}\left(\ell, N, \mathcal{D}, \mathbf{v}\right) = \sigma\right] \geq 1-\delta$.

Our main result in the section is stated in the theorem below (see Figure 3 for a proof sketch).

Theorem 9. *For any error tolerance $\delta \in (0,1)$, it is possible to construct a $\mathcal{D}$-robust vector for all distributions $\mathcal{D}$.*

Proof. We first look at the case when $\mathcal{D}$ has bounded support over $\mathbb{R}$. Let $[a,b] \subset \mathbb{R}$ be the support of $\mathcal{D}$. Let $\tau = \max\{|a|,|b|\}$. Then, for any $\eta \sim \mathcal{D}$, it holds that $|(\mathcal{L}_{\mathbf{v}}(\sigma) + \eta) - \mathcal{L}_{\mathbf{v}}(\sigma)| = |\eta| \leq \tau$ and hence, any τ-robust vector is also $\mathcal{D}$-robust.

For the case when $\mathcal{D}$ has unbounded support, let $\eta \sim \mathcal{D}$, and $a, b \in \mathbb{R}$ be such that $\Pr\left(\eta \in (-\infty, a) \cup (b, \infty)\right) < \delta$. To see that this can be done, let $F : \mathbb{R} \to [0,1]$ be the cumulative distribution function for $\mathcal{D}$, which is a nondecreasing, right-continuous function with $F(-\infty) = 0$, $F(\infty) = 1$, and $F^{-1}(p) = \inf\{x \in \mathbb{R} : F(x) \geq p\}$. For any fixed $\delta \in (0,1)$, we can set $a = F^{-1}(\delta/4)$ and $b = F^{-1}(1-\delta/4)$ so that $\Pr\left(\eta \in [a,b]\right) \geq \delta/2$, which makes $\Pr\left(\eta \in (-\infty, a) \cup (b, \infty)\right) \leq \delta/2 < \delta$. Note that a and b are always finite by definition of the cumulative distribution function as the point mass is zero on $\pm\infty$. □

To see why this works, observe that for any bounded distribution, say over some interval $[a,b] \subset \mathbb{R}$, the amount of noise added is never more than $\max\{|a|,|b|\}$. Thus, any $\max\{|a|,|b|\}$-robust vector can unambiguously recover the labels from the noised scores. For distributions with unbounded support, however, such an upper bound does not exist. Given the error tolerance, it is possible to compute this bound for any distribution and robustness be defined accordingly. We explain this for subexponential noise below.

Recall that a random variable $X \in \mathbb{R}$ is said to be subexponential (denoted $X \sim \mathsf{subE}(\lambda^2, \nu)$) with parameters $\lambda^2, \nu > 0$ if $\mathbb{E}(X) = 0$ and its moment generating function satisfies $\mathbb{E}(e^{sX}) \leq \exp\left(\lambda^2 s^2/2\right)$ for all $|s| < 1/\nu$. Given any tolerance $\delta \in (0,1)$, it can be shown that there exists a $\mathsf{subE}(\lambda^2,\nu)$-robust vector for all $\lambda, \nu > 0$. We formally state this result below.

Theorem 10. *For all $\lambda, \nu > 0$ and $\delta \in (0,1)$, any vector that is $\left(2(\lambda+\nu)\sqrt{\ln\left(\frac{2}{\delta}\right)}\right)$-robust is $\mathsf{subE}(\lambda^2,\nu)$-robust as well. In particular, with probability at least $1-o(1)$, there exists an adversary for the single query Robust Label Inference problem for $\mathsf{subE}(\lambda^2,\nu)$ noise.*

Proof. We begin with reminding the reader a concentration bound that will be useful in our proof.

Lemma 8 ((Dubhashi & Panconesi, 2009)). *Let $Y \sim \mathsf{subE}(\lambda^2,\nu)$ be a zero-mean random variables for some $\lambda, \nu > 0$. Then, for all $t > 0$, the following holds:*

$$\Pr\left(|Y| > t\right) \leq 2\exp\left(-\frac{1}{2}\left(\frac{t^2}{\lambda^2} \wedge \frac{t}{\nu}\right)\right),$$

where $a \wedge b := \min\{a,b\}$.

Given this result, we follow the general construction shown in the proof of Theorem 9, we compute $a > 0$ such that for any $Y \sim \mathsf{subE}(\lambda^2,\nu)$, it holds that $\Pr\left(Y \in (-a,a)\right) \geq 1-\delta$. This would allow all a-robust vectors to be robust with probability at least $1-\delta$, as needed.

To compute a, observe that from Lemma 8, we can set $t = a$ and set the bound on the probability to be at most δ, as follows:

$$\Pr\left(|Y| > a\right) \leq 2\exp\left(-\frac{1}{2}\left(\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu}\right)\right) < \delta.$$

A simple algebraic manipulation for the inequality on the right gives the condition that $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} > 2\ln\left(\frac{2}{\delta}\right)$. We solve the two cases here separately. If $a < \lambda^2/\nu$, then $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} = \frac{a^2}{\lambda^2}$, and hence, this gives $a < \lambda\sqrt{2\ln\left(\frac{2}{\delta}\right)}$. Else, we have $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} = \frac{a}{\nu}$, which gives $a < 2\nu\ln\left(\frac{2}{\delta}\right)$. It suffices to take the maximum of these two limits and hence, $\max\{\lambda\sqrt{2\ln\left(\frac{2}{\delta}\right)}, 2\nu\ln\left(\frac{2}{\delta}\right)\}$

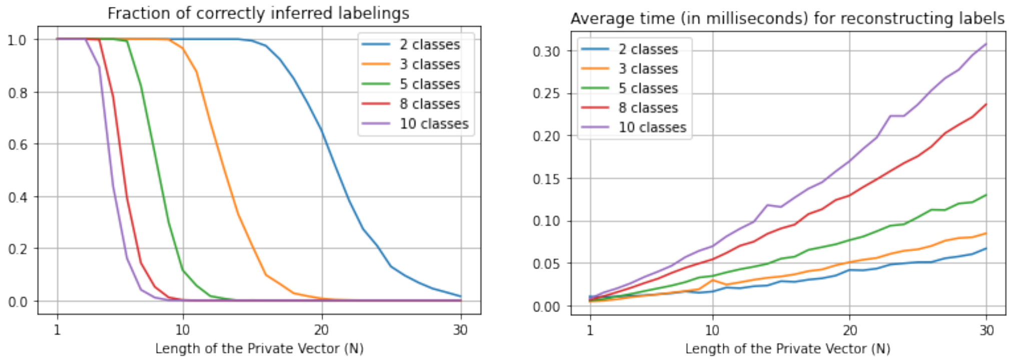


Figure 4. Empirical results for multi-class label inference. The plot on the top shows that due to fixed-precision, the accuracy drops to zero faster as the number of classes increases, since larger primes powers are used in the prediction vectors. The plot below shows that the inference time increases with the number of classes.

works. We can further simplify this expression using the upper bound $2(\lambda+\nu)\sqrt{\ln\left(\frac{2}{\delta}\right)}$ which follows from the fact that $\delta \in (0,1)$. □

For example, let $Z = \mathcal{N}(0,1)$ denote the standard normal random variable. Then, we know that the Chi-squared random variable with one degree of freedom follows the law for $Z^2 = \mathsf{subE}(2,4)$. Thus, from the Corollary above, we obtain that for with probability at least $1 - o(1)$, any vector that can handle up to $12\sqrt{\ln N}$ amount of bounded noise can also handle Z^2 noise. If we are allowed up to, say $\delta = 0.1$, then any $12\sqrt{\ln 20} \approx 20.77$-robust vector suffices.

B.5. Extension to Other Noise Models: Multiplicative Noise

We briefly explore extending the analysis above to the case of multiplicative noise. In this case, the adversary observes score ℓ that satisfies:

$$(1-\alpha_1)\mathcal{L}_{\mathbf{v}}(\sigma) \le \ell \le (1+\alpha_2)\mathcal{L}_{\mathbf{v}}(\sigma),$$

where $\alpha_1 \in [0,1)$ and $\alpha_2 \ge 0$ are the known bounds on the rate of the multiplicative noise. We show that if the rate is small enough, then it is possible to reduce this case to that of additive noise. To see this, we begin with rewriting the inequality above as $|\ell - \mathcal{L}_{\mathbf{v}}(\sigma)| \le \alpha^* \cdot |\mathcal{L}_{\mathbf{v}}(\sigma)|$, where $\alpha^* = \max\{\alpha_1, \alpha_2\}$. Now, if we use the construction of a τ-robust vector from Algorithm 2, it is easy to compute that $|\mathcal{L}_{\mathbf{v}}(\sigma)| \le 2^{N+1}\tau$ for any σ, which would require ensuring $2^{N+1}\tau\alpha^* < 2\tau$ – this is not possible for any $\tau > 0$ unless $\alpha^* < 2^{-N}$. Thus, we require multiple queries.

Suppose we only infer $1 \le m \le N$ labels in a single query (using $M = m$ in Algorithm 3), then this will require $\left(\ln 2 + 2^{m+1}\tau\right)\alpha^*$ to be at most τ. It suffices to set $\alpha^* \le 1/2^{m+2}$. Then, the quantity on the left becomes at most

$$\frac{\ln 2 + 2^{m+1}\tau}{2^{m+2}} = \frac{\ln 2}{2^{m+2}} + \frac{\tau}{2} \le \tau,$$

whenever $\tau \ge \frac{\ln 2}{2^{m+1}}$. We state this formally as follows.

Theorem 11. *For any $\alpha^* \le \frac{1}{8}$, label inference can be done, $\left\lceil \log_2\left(\frac{1}{\alpha^*}\right) - 2 \right\rceil$ labels at a time, using vectors that are $(2\ln 2)\alpha^*$-robust.*

Observe that when $\alpha \ge \frac{1}{4}$, then no value of τ satisfies the constraint above, implying that robust vectors from Algorithm 3 cannot be used with any number of queries. The noise is more than what these vectors can guarantee handling. We defer label inference for this case to future work.

C. Experimental Results for Multi-Class Label Inference

Similar to Section 5, we evaluate our attacks on simulated labelings (with no noise) in the multi-class setting (see Figure 4). The results show that our algorithms are efficient, even with a large number of datapoints. All experiments are run on a

64-bit machine with 2.6GHz 6-Core processor, using the standard IEEE-754 double precision format (1 bit for sign, 11 bits for exponent, and 53 bits for mantissa). For ensuring reproducibility, the entire experiment setup is submitted as part of the supplementary material.

The plot on the left in Figure 4 shows the accuracy of label inference, where we use the attack based on primes from Theorem 8. The accuracy reported is with respect to 100 randomly generated labelings for each N (length of the vector to be inferred). We vary the number of classes from 2 to 10 for these experiments. Similar to the results in Figure 1, the maximum accuracy falls to zero when N increases, because of the limited floating point precision on the machine. As the number of classes increase, this drop happens sooner (*i.e.* for a smaller N), because the powers of primes get larger.

The plot on the right shows the run time for our attack. The results show that this inference happens in only a few milliseconds.