Supplementary Material

A Preliminaries

For $m \in \mathbb{N}$, we use $[m]$ to denote $\{1, \ldots, m\}$. For a distribution $\mathcal{D}$, we write $r \sim \mathcal{D}$ to denote a random variable r distributed as $\mathcal{D}$. For a randomized algorithm $\mathbb{A}$, we write $\mathbb{A}(\mathbf{X})$ to denote the distribution of the output of $\mathbb{A}$ on input $\mathbf{X}$. For a distribution $\mathcal{D}$, we write $\mathbb{A}(\mathcal{D})$ to denote the distribution of the output of $\mathbb{A}$ when the input is drawn from $\mathcal{D}$. Sometimes we will allow the number of samples drawn by an algorithm to be a random variable. In this case, the algorithm must specify the number of samples before seeing any samples. Furthermore, when $\mathcal{D}$ is the distribution of each sample, we may write $\mathbb{A}_{\mathcal{D}}$ to denote the distribution of the output when each of $\mathbb{A}$'s samples is drawn from $\mathcal{D}$. We use $\mathcal{D}_1 \otimes \cdots \otimes \mathcal{D}_m$ to denote the product distribution of the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_m$. Furthermore, we use $\mathcal{D}^{\otimes m}$ to denote the m-fold product of the distribution $\mathcal{D}$ with itself.

For convenience, we interchangeably refer to a halfspace by $h_{\mathbf{w}}$ or just the weight vector $\mathbf{w}$ itself.

A.1 Margin of Halfspaces

Robust learning of halfspaces is intimately related to the notion of margin. For a margin parameter $\gamma > 0$, we say that an example $(\mathbf{x}, y) \in \mathbb{R}^d \times \{\pm 1\}$ is *correctly classified by* $\mathbf{w}$ *with margin* γ iff $\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle - y \cdot \gamma) = y$. The γ-margin error is defined as $\mathrm{err}^{\mathcal{D}}_{\gamma}(\mathbf{w}) = \Pr_{(\mathbf{x},y)\sim\mathcal{D}}[\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle - y \cdot \gamma) \neq y]$. The connection between robust learning of halfspaces and learning with margin is given through the following (folklore) lemma; its proof can be found, e.g., in (Diakonikolas et al., 2020).

Lemma 7. *For any non-zero* $\mathbf{w} \in \mathbb{R}^d$, $\gamma \geq 0$ *and* $\mathcal{D}$, $\mathcal{R}_{\gamma}(\mathbf{w}, \mathcal{D}) = \mathrm{err}^{\mathcal{D}}_{\gamma}\left(\frac{\mathbf{w}}{\|\mathbf{w}\|_2}\right)$.

Due to the above lemma, we may refer to the γ-margin error for halfspaces instead of their robust risk throughout the paper.

A.2 Boosting the Success Probability

Throughout this work, it is often more convenient to prove lower bounds (resp. upper bounds) only for some large (resp. small) failure probability $\xi \in (0, 1)$. We note that this is without loss of generality, since standard techniques can be used to boost the success probability while incurring small loss in the sample complexity. We sketch the argument below.

Observation 8. *For any* $\xi, \xi' \in (0, 1)$*, the following statement holds: If there is a* (γ, γ')*-robust learner with failure probability* ξ*, accuracy* α *and sample complexity* m*, then there exists a* (γ, γ')*-robust learner with failure probability* ξ'*, accuracy* 1.1α *and sample complexity* $O_{\xi,\xi'}(m + 1/\alpha^2)$.

Proof Sketch. Let $\mathbb{A}$ be the (γ, γ')-robust learner with failure probability ξ, accuracy α and sample complexity m. We define an algorithm $\mathbb{B}$ as follows:

- Let $T := \lceil \frac{\log(0.5\xi')}{\log(1-\xi)} \rceil$ and $M := \lceil \frac{10^6 \cdot \log T}{\alpha} \rceil$.
- For $i \in [T]$, run $\mathbb{A}$ on m samples to get a halfspace $\mathbf{w}_i$.
- Sample M fresh new samples. Then, output $\mathbf{w}_i$ that minimizes the γ-margin error of $\mathbf{w}_i$ on the uniform distribution over these M samples.

Clearly, the algorithm $\mathbb{B}$ uses $m \cdot T + M = O_{\xi,\xi'}(m + 1/\alpha^2)$ samples as desired. For the accuracy, with probability $1 - (1-\xi)^T \geq 1 - 0.5\xi'$ at least one of the $\mathbf{w}_i$'s satisfies $\mathrm{err}^{\mathcal{D}}_{\gamma'}(\mathbf{w}_i) \leq \alpha$. Conditioned on this, the Chernoff bound ensures that w.p. $1 - 0.5\xi$ we output a $\mathbf{w}_i$ s.t. $\mathrm{err}^{\mathcal{D}}_{\gamma'}(\mathbf{w}_i) \leq 1.1\alpha$. We can then conclude the proof via the union bound. □

B Lower Bound for Robust Learning of Halfspaces: Pure-DP Case

In this section, we prove our lower bound for ϵ-DP robust learning of halfspaces (Theorem 2), which is restated below.

Theorem 2. *Any ϵ-DP $(\gamma, 0.9\gamma)$-robust (possibly improper) learner has sample complexity $\Omega(d/\epsilon)$.*

We will use the following (well-known) fact; for completeness, we sketch its proof at the end of this section.

Lemma 9. *There exist $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(K)} \in \mathbb{R}^d$ where $K = 2^{\Omega(d)}$ such that $\|\mathbf{w}^{(i)}\|_2 = 1$ for all $i \in [K]$ and $|\langle \mathbf{w}^{(i)}, \mathbf{w}^{(j)} \rangle| < 0.01$ for all $i \neq j$.*

Proof of Theorem 2. We will prove the statement for any $\gamma \leq 0.99, \alpha \leq 0.49$ and $\xi \leq 0.9$.

Let $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(K)}$ be the vectors guaranteed by Lemma 9. For each $i \in [K]$, we define $\mathcal{D}^{(i)}$ to be the uniform distribution on two elements: $(1.01\gamma \cdot \mathbf{w}^{(i)}, +1)$ and $(-1.01\gamma \cdot \mathbf{w}^{(i)}, -1)$. Notice that $\mathcal{R}_\gamma(\mathbf{w}^{(i)}, \mathcal{D}^{(i)}) = 0$.

Now, let $G^{(i)} = \{h : \mathbb{B}^d \to \{\pm 1\} \mid \mathcal{R}_{0.9\gamma}(h, \mathcal{D}^{(i)}) \leq \alpha\}$ denote the set of hypotheses which incurs error no more than α on $\mathcal{D}^{(i)}$. The main claim is the following:

Claim 10. *For every $i \neq j$, $G^{(i)} \cap G^{(j)} = \emptyset$.*

Proof. Suppose for the sake of contradiction that there exists $h \in G^{(i)} \cap G^{(j)}$ for some $i \neq j$.

Since $\alpha \leq 0.49$ and $\mathcal{D}^{(i)}$ is a uniform distribution over only two samples, $\mathcal{R}_{0.9\gamma}(h, \mathcal{D}^{(i)}) \leq \alpha$ implies that $\mathcal{R}_{0.9\gamma}(h, \mathcal{D}^{(i)}) = 0$. This implies that

$$h(z) = 1 \qquad \forall z \in \mathbb{P}_{0.9\gamma}(1.01\gamma \cdot \mathbf{w}^{(i)}).$$

By an analogous argument, we have

$$h(z) = -1 \qquad \forall z \in \mathbb{P}_{0.9\gamma}(-1.01\gamma \cdot \mathbf{w}^{(j)}).$$

This is a contradiction since $\mathbb{P}_{0.9\gamma}(-1.01\gamma \cdot \mathbf{w}^{(j)}) \cap \mathbb{P}_{0.9\gamma}(1.01\gamma \cdot \mathbf{w}^{(i)}) \neq \emptyset$; specifically, $|\langle \mathbf{w}^{(i)}, \mathbf{w}^{(j)} \rangle| < 0.01$ implies that this intersection contains $0.505\gamma \cdot \mathbf{w}^{(i)} - 0.505\gamma \cdot \mathbf{w}^{(j)}$. □

To finish the proof, consider any ϵ-DP $(\gamma, 0.9\gamma)$-robust learner $\mathbb{A}$ with $\alpha \leq 0.49$. Suppose that it takes n samples. Notice that, when we feed it n random samples from $\mathcal{D}^{(i)}$, the accuracy guarantee ensures that

$$\Pr_{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \sim \mathcal{D}^{(i)}} [\mathbb{A}((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)) \in G^{(i)}] \geq 1 - \xi.$$

As a result, since $\mathbb{A}$ is ϵ-DP, we have

$$\Pr[\mathbb{A}(\emptyset) \in G^{(i)}] \geq (1 - \xi) \cdot e^{-\epsilon \cdot n} \geq 0.1 \cdot e^{-\epsilon \cdot n}. \tag{3}$$

From Claim 10, $G^{(1)}, \ldots, G^{(K)}$ are disjoint, which implies

$$1 \geq \sum_{i \in [K]} \Pr[\mathbb{A}(\emptyset) \in G^{(i)}] \overset{(3)}{\geq} K \cdot 0.1 \cdot e^{-\epsilon \cdot n}.$$

Thus, we have $n \geq \Omega\left(\frac{\log K}{\epsilon}\right) = \Omega(d/\epsilon)$ as desired. □

Finally, we briefly sketch the proof of Lemma 9.

Proof of Lemma 9. It is well-known that there exist linear error correcting codes over $\mathbb{F}_2$ with constant rate and distance 0.4995. (See e.g. (Alon et al., 1990, Section 7) for an explanation.) Equivalently, this means that there exists a linear space $V \subseteq \mathbb{F}_2^d$ of dimension $\Omega(d)$ such that $\|\mathbf{v}\|_0 \in [0.4995d, 0.5005d]$ for all non-zero $\mathbf{v} \in V$ where $\|\cdot\|_0$ denote the Hamming norm (i.e. number of non-zero coordinates).

Let $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(K)}$ denote the elements of V notice that $K = 2^{\dim(V)} = 2^{\Omega(d)}$. Define $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(K)} \in \mathbb{R}^d$ where

$$\mathbf{w}_\ell^{(i)} = \begin{cases} -1/\sqrt{d} & \text{if } \mathbf{v}_\ell^{(i)} = 0 \\ +1/\sqrt{d} & \text{if } \mathbf{v}_\ell^{(i)} = 1. \end{cases}$$

For $i \neq j$, we have

$$\left|\left\langle \mathbf{w}^{(i)}, \mathbf{w}^{(j)} \right\rangle\right| = |1 - 2 \cdot \|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|_0 / d| \leq 0.01d,$$

where the latter follows from linearity of V. This concludes our proof. □

C Lower Bound for Robust Learning of Halfspaces: Approximate-DP Case

For our lower bound for approximate-DP proper learners (Theorem 3), we will reduce from a lower bound of Steinke and Ullman (2017). To state their results, we will need some additional notation. Let $\mathcal{U}_{[0,1]}$ denote the uniform distribution on $[0,1]$, and let $\mathcal{B}_q$ denote the distribution that is $+1/\sqrt{d}$ with probability q and is $-1/\sqrt{d}$ otherwise. For $\mathbf{q} \in [0,1]^d$, we use $\mathcal{B}_{\mathbf{q}}$ to denote $\mathcal{B}_{q_1} \otimes \cdots \otimes \mathcal{B}_{q_d}$. Steinke and Ullman (2017) prove the following theorem[8][9]:

Theorem 11 ((Steinke and Ullman, 2017, Theorem 3))**.** *Let $\zeta > 0$ and $n, d \in \mathbb{N}$ be such that $n < \zeta\sqrt{d}$. Let $\mathcal{M}$ be any $(1, \zeta/n)$-DP algorithm whose output belongs to the d-dimensional unit Euclidean ball. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be such that $\mathbf{x}_i$ is i.i.d. drawn from $\mathcal{B}_{\mathbf{q}}$. Then,*

$$\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{M}(\mathcal{B}_{\mathbf{q}}^{\otimes n})}\left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] < \zeta\sqrt{d}. \tag{4}$$

In the next subsection, we first show a lower bound of $\Omega(\sqrt{d})$ for any sufficiently small constant γ (Lemma 13). Then, in Subsection C.2, we use this to prove a lower bound of $\Omega(\min\{\sqrt{d}/\gamma, d\})$.

C.1 Lower Bound for $\gamma = \Omega(1)$

We cannot use the distribution $\mathcal{B}_{\mathbf{q}}$ directly since it is not realizable with a large margin. To overcome this, we define $\mathcal{P}_{\mathbf{q}}$ as the distribution of $\mathbf{x} \sim \mathcal{B}_{\mathbf{q}}$ conditioned on $\langle \mathbf{q}', \mathbf{x}\rangle \geq 0.01$ where we write $\mathbf{q}'$ as a shorthand for $\frac{1}{\sqrt{d}}(2\mathbf{q} - \mathbf{1})$. We will require the following bound:

Lemma 12. $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}}[d_{TV}(\mathcal{B}_{\mathbf{q}}, \mathcal{P}_{\mathbf{q}})] \leq o(1/d)$.

Proof. The Chernoff bound implies that $\Pr_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}}[\|\mathbf{q}'\| \geq 0.1] \geq o(1/d)$. For a fixed $\mathbf{q}$ such that $\|\mathbf{q}'\| \geq 0.1$, the Chernoff bound again yields that $\Pr_{\mathbf{x}\sim\mathcal{B}_{\mathbf{q}}}[\langle \mathbf{q}', \mathbf{x}\rangle \geq 0.01] \leq o(1/d)$, which implies that $d_{TV}(\mathcal{B}_{\mathbf{q}}, \mathcal{P}_{\mathbf{q}}) \leq o(1/d)$. Combining these, we have $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}}[d_{TV}(\mathcal{B}_{\mathbf{q}}, \mathcal{P}_{\mathbf{q}})] \leq o(1/d)$ as desired. □

Let $\tilde{\mathcal{P}}_{\mathbf{q}}$ denote the distribution of $(\mathbf{x}, +1)$ where $\mathbf{x} \sim \mathcal{P}_{\mathbf{q}}$. Similarly, let $\tilde{\mathcal{B}}_{\mathbf{q}}$ denote the distribution of $(\mathbf{x}, +1)$ where $\mathbf{x} \sim \mathcal{B}_{\mathbf{q}}$. We can now prove our $\Omega(\sqrt{d})$ lower bound for any sufficiently small constant $\gamma > 0$, which follows almost immediately from the following lemma.

Lemma 13. *For any constant $\gamma, \beta \in (0,1)$ such that $\gamma > 2\beta$, the following holds. Let $\mathbb{A}$ be any $(1, o(1/n))$-DP algorithm with sample complexity n and whose output belongs to the d-dimensional unit Euclidean ball. If*

$$\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})] \leq \beta,$$

then we must have $n \geq \Omega(\sqrt{d})$.

Proof. Suppose for the sake of contradiction that there exists a $(1, o(1/n))$-DP algorithm $\mathbb{A}$ with sample complexity $n = o(\sqrt{d})$ whose output is a d-dimensional vector of Euclidean norm at most one that satisfies $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\mathcal{P}_{\mathbf{q}}^{\otimes n})}[\mathrm{err}_{\gamma}^{\mathcal{P}_{\mathbf{q}}}(\mathbf{w})] \leq \beta$.

[8] We remark that (1) the result of Steinke and Ullman (2017) is stated for the Beta distributions which contain the uniform distribution (i.e., $\mathcal{U}([0,1]) = \mathrm{Beta}(1,1)$) (2) we scale down the output $\mathcal{M}(\mathbf{X})$ by a factor of $1/\sqrt{k}$ (which has the same effect on the error), (3) we replace the Bernoulli distribution with $\mathcal{B}_q$ which is valid since there is a one-to-one mapping between the two, (4) the theorem of Steinke and Ullman (2017) has another parameter k which we simply set to d and (5) the original theorem in Steinke and Ullman (2017) implicitly imposes a bound on $\|\mathcal{M}(\mathbf{X})\|_\infty$ but the actual condition needed is on $\|\mathcal{M}(\mathbf{X})\|_1$ which is already implied by our condition that $\|\mathcal{M}(\mathbf{X})\|_2 \leq 1$.

[9] We also remark that a similar theorem can already be derived via the work of Dwork et al. (2015); however, we choose to state this version since it is more compatible with our reduction and is readily available already in (Steinke and Ullman, 2017).

On input $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \{\pm 1/\sqrt{d}\}^d$, $\mathcal{M}$ simply works as follows: Run $\mathbb{A}$ on $(\mathbf{x}_1, +1), \ldots, (\mathbf{x}_n, +1)$ to obtain a halfspace $\mathbf{w}$ and output $\mathbf{w}$. Now, we have that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{M}(\mathcal{B}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] \\
&= \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{B}}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] \\
&\geq \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] - \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}} \left[d_{TV}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n}, \tilde{\mathcal{B}}_{\mathbf{q}}^{\otimes n}) \cdot (0.5\sqrt{d})\right] \\
&\geq \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] - (0.5n\sqrt{d}) \cdot \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}} \left[d_{TV}(\mathcal{P}_{\mathbf{q}}, \mathcal{B}_{\mathbf{q}})\right] \\
&\overset{\text{Lemma 12}}{\geq} \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] - o(n/\sqrt{d}) \\
&= \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] - o(1), \qquad (5)
\end{aligned}$$

where in the first inequality we use the fact that $\|\mathbf{w}\|_2 \leq 1$, which implies that $\|\mathbf{w}\|_1 \leq \sqrt{d}$.

Notice that we may rearrange the term inside the expectation in (5) as follows:

$$\begin{aligned}
\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5) &= \frac{\sqrt{d}}{2} \sum_{j\in[d]} \mathbf{w}_j \cdot \frac{2q_j - 1}{\sqrt{d}} \\
&= \frac{\sqrt{d}}{2} \langle \mathbf{w}, \mathbf{q}' \rangle \\
&= \frac{\sqrt{d}}{2} \left\langle \mathbf{w}, \mathbb{E}_{\mathbf{x}\sim\mathcal{B}_{\mathbf{q}}}[\mathbf{x}] \right\rangle \\
&= \frac{\sqrt{d}}{2} \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{B}_{\mathbf{q}}}[\langle \mathbf{w}, \mathbf{x} \rangle] \\
&\overset{\text{Lemma 12}}{\geq} \frac{\sqrt{d}}{2} \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_{\mathbf{q}}}[\langle \mathbf{w}, \mathbf{x} \rangle] - o(1) \\
&\geq \frac{\sqrt{d}}{2} \left(\gamma \cdot \Pr_{\mathbf{x}\sim\mathcal{P}_{\mathbf{q}}}[\langle \mathbf{w}, \mathbf{x} \rangle \geq \gamma] - 1 \cdot \Pr[\langle \mathbf{w}, \mathbf{x} \rangle < \gamma]\right) - o(1) \\
&= \frac{\sqrt{d}}{2} \left(\gamma \cdot (1 - \mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})) - \mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})\right) - o(1) \\
&\geq \frac{\sqrt{d}}{2} \left(\gamma - 2\,\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}))\right) - o(1)
\end{aligned}$$

Plugging this back into (5), we have that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{M}(\mathcal{B}_{\mathbf{q}}^{\otimes n})} \left[\sum_{j\in[d]} \mathbf{w}_j \cdot (q_j - 0.5)\right] \\
&\geq \frac{\sqrt{d}}{2} \cdot \left(\gamma - 2 \cdot \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})} \left[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}))\right]\right) - o(1) \\
&\geq \frac{\sqrt{d}}{2} (\gamma - 2\beta) - o(1)
\end{aligned}$$

$$= \Omega(\sqrt{d}),$$

which contradicts Theorem 11. This concludes our proof. □

C.2 Lower Bound for Smaller γ

We will now reduce from the case $\gamma = \Omega(1)$ to get a larger lower bound for smaller γ. To do this, it will be convenient to have an "expected version" of Lemma 13, which is stated and proved below.

Lemma 14. *For any constants $\gamma_0, \beta_0 \in (0,1)$ such that $\gamma_0 > 4\sqrt{2\beta_0}$, the following holds. Let $\mathbb{B}$ be any $(1, o(1/n))$-DP algorithm that has access to an oracle $\mathcal{O}$ that can sample from $\tilde{\mathcal{P}}_{\mathbf{q}}$ where $\mathbf{q}$ is unknown to $\mathbb{B}$. All of the following cannot hold simultaneously:*

1. *The expected number of samples $\mathbb{B}$ draws from $\mathcal{O}$ is $o(\sqrt{d})$.*
2. $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}}[\|\mathbf{w}\|^2] \le 1$.
3. $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}}[\mathrm{err}_{\gamma_0}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})] \le \beta_0$.

Proof. Suppose for the sake of contradiction that there exists a $(1, o(1/n))$ algorithm $\mathbb{B}$ that draws $o(\sqrt{d})$ samples from $\mathcal{O}$ in expectation, and satisfies $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}}[\|\mathbf{w}\|^2] \le 1$ and $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}}[\mathrm{err}_{\gamma_0}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})] \le \beta_0$. We use $\mathbb{B}$ to construct an algorithm $\mathbb{A}$ that will contradict Lemma 13 as follows:

- Run $\mathbb{B}$.
- If $\mathbb{B}$ attempts to take more than $2n/\beta_0$ sample, simply output $\mathbf{0}$.
- Otherwise, let $\mathbf{w}$ be the output of $\mathbb{B}$, and output $\mathbf{w}' = \frac{\mathbf{w}}{\|\mathbf{w}\|}$.

Notice that $\mathbb{A}$ is $(1, o(1/n))$-DP and the number of samples used is $2n/\beta_0 = o(\sqrt{d})$.

Let $\beta = 2\beta_0$ and $\gamma = \gamma_0\sqrt{\beta_0/2}$. We will next argue that $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}'\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}')] \le \beta$. First, since the expected number of samples of $\mathbb{B}$ is n, by Markov's inequality, the probability that $\mathbb{B}$ takes more than $2n/\beta_0$ samples is at most $\beta_0/2$. As a result, we have that

$$\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}'\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}')] \le \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}^{\otimes n}}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}/\|\mathbf{w}\|)] + \beta_0/2.$$

Recall also that $\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbb{B}}[\|\mathbf{w}\|^2] \le 1$; Markov's inequality once again implies that $\Pr_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbb{B}}[\|\mathbf{w}\|^2 > 2/\beta_0] \le \beta_0/2$. Plugging this into the above inequality, we get that

$$\begin{aligned}
\mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}'\sim\mathbb{A}(\tilde{\mathcal{P}}_{\mathbf{q}}^{\otimes n})}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}')] &\le \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}^{\otimes n}}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}/\|\mathbf{w}\|) \cdot \mathbb{1}[\|\mathbf{w}\| \le \sqrt{2/\beta_0}]] + \beta_0. \\
&\le \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}^{\otimes n}}[\mathrm{err}_{\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}/\sqrt{2/\beta_0})] + \beta_0 \\
&= \mathbb{E}_{\mathbf{q}\sim\mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w}\sim\mathcal{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}^{\otimes n}}[\mathrm{err}_{\gamma_0}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})] + \beta_0 \\
\text{(From our third assumption on } \mathbb{B}) &\le 2\beta_0 = \beta,
\end{aligned}$$

which is a contradiction to Lemma 13 since $\gamma > 2\beta$. □

We can now prove our $\Omega(\min\{\sqrt{d}/\gamma, d\})$ lower bound (Theorem 3). Roughly speaking, when $d \ge 1/\gamma^2$, we "embed" $\Theta(1/\gamma^2)$ hard distributions from Lemma 14 into $\Theta(\gamma^2 d)$ dimensions, which results in the $\Omega(\sqrt{\gamma^2 d} \cdot 1/\gamma^2) = \Omega\left(\sqrt{d}/\gamma\right)$ lower bound as desired.

Theorem 3. *Let $\epsilon < 1$. Any $(\epsilon, o(1/n))$-DP $(\gamma, 0.9\gamma)$-robust proper learner has sample complexity $n = \Omega(\min\{\sqrt{d}/\gamma, d\})$.*

Proof. We will prove this lower bound for $\gamma \leq 0.01, \alpha, \xi \leq 10^{-6}$.

First, notice that, when $\gamma \leq 1/\sqrt{d}$, a $(\gamma, 0.9\gamma)$-robust proper learner is also an $(1/\sqrt{d}, 0)$-robust proper learner. Hence, by Theorem 4, we have $n = \Omega(d)$ as desired. Thus, we can subsequently only focus on the case $\gamma \geq 1/\sqrt{d}$, for which we will show that $n = \Omega(\sqrt{d}/\gamma)$.

Suppose for the sake of contradiction that there is a $(1, o(1/n))$-DP $(\gamma, 0.9\gamma)$-robust proper learner $\mathbb{A}$ with $\alpha, \xi \leq 10^{-6}$ that has sample complexity $n = o(\sqrt{d}/\gamma)$. Let $T = \lfloor 0.01/\gamma \rfloor$, and $d' = \lfloor d/T^2 \rfloor$. We will construct an algorithm $\mathbb{B}$ that contradicts with Lemma 14 in d' dimensions.

We will henceforth assume w.l.o.g. that $d = d' \cdot T^2$. This is without loss of generality since the proof below extends to the case $d > d' \cdot T^2$ by padding $d - d' \cdot T^2$ zeros to each of the samples.

In the following, we view the d-dimensional space $\mathbb{R}^d$ as the tensor $\mathbb{R}^{T^2} \otimes \mathbb{R}^{d'}$. Furthermore, we write $\mathbf{e}_i$ as a shorthand for the i-th vector in the standard basis of $\mathbb{R}^{d'}$.

The algorithm $\mathbb{B}$ with an oracle $\mathcal{O}$ to sample from $\tilde{\mathcal{P}}_{\mathbf{q}}$ where $\mathbf{q}$ is unknown to $\mathbb{B}$ works as follows:

- Randomly draw $\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}$ i.i.d. from $\mathcal{U}_{[0,1]}^{\otimes d}$, and randomly sample $i^* \in [T^2]$.
- Draw n samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ independently as follows:
  - Draw $i \sim [T^2]$.
  - If $i \neq i^*$, then draw $(\mathbf{x}, y) \sim \tilde{\mathcal{P}}_{\mathbf{q}_i}$ and let the sample be $(\mathbf{x} \otimes \mathbf{e}_i, y)$.
  - If $i = i^*$, the draw $(\mathbf{x}, y) \sim \tilde{\mathcal{P}}_{\mathbf{q}}$ using $\mathcal{O}$ and let the sample be $(\mathbf{x} \otimes \mathbf{e}_i, y)$.
- Run $\mathbb{A}$ on $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$. Suppose that the output halfspace is $\mathbf{w}$.
- Write $\mathbf{w}$ as $\sum_{i \in [T^2]} \mathbf{w}^i \otimes \mathbf{e}_i$ for $\mathbf{w}_1, \ldots, \mathbf{w}_{T^2} \in \mathbb{R}^{d'}$. Then, output $T \cdot \mathbf{w}^{i^*}$.

Clearly, $\mathbb{B}$ is $(1, o(1/n))$-DP and it takes $n/T = o(\sqrt{d'})$ samples in expectation from $\mathcal{P}_{\mathbf{q}}$.

For the ease of presentation, we will write $\mathcal{Q}$ as a shorthand for the mixture of distribution where we draw $i \sim [T]$, and return $(\mathbf{x} \otimes \mathbf{e}_i, y)$ where $(\mathbf{x}, y) \sim \tilde{\mathcal{P}}_{\mathbf{q}_i}$. Moreover, we write $\tilde{\mathcal{Q}}$ as a similar distribution but when $\mathbf{q}_{i^*}$ is replaced by $\mathbf{q}$. Under this notation, we have that

$$\begin{aligned}
\mathbb{E}_{\mathbf{q} \sim \mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w} \sim \mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}} [\|\mathbf{w}\|^2] &= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, \mathbf{q} \sim \mathcal{U}_{[0,1]}^{\otimes d}, i^* \sim [T], \mathbf{w} \sim \mathbb{A}(\tilde{\mathcal{Q}}^n)} \left[\|T \cdot \mathbf{w}^{i^*}\|^2 \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2} \sim \mathcal{U}_{[0,1]}^{\otimes d}, i^* \sim [T], \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\|T \cdot \mathbf{w}^{i^*}\|^2 \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2} \sim \mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\frac{1}{T^2} \cdot \sum_{i^* \in [T^2]} \|T \cdot \mathbf{w}^{i^*}\|^2 \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2} \sim \mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\|\mathbf{w}\|^2 \right] \\
&\leq 1.
\end{aligned}$$

Finally, we will argue the accuracy of $\mathbb{B}$ where $\gamma_0 = 0.01, \beta_0 = 2 \cdot 10^{-6}$. Once again we rewrite it as

$$\begin{aligned}
\mathbb{E}_{\mathbf{q} \sim \mathcal{U}_{[0,1]}^{\otimes d}, \mathbf{w} \sim \mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}} [\mathrm{err}_{\gamma_0}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w})] &= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, \mathbf{q}, i^*, \mathbf{w} \sim \mathbb{A}(\tilde{\mathcal{Q}}^n)} \left[\mathrm{err}_{\gamma_0}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(T \cdot \mathbf{w}^{i^*}) \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, i^*, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\mathrm{err}_{\gamma_0/T}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}^{i^*}) \right] \\
(\text{Since } \gamma_0/T \leq 0.1\gamma) &\leq \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, i^*, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\mathrm{err}_{0.1\gamma}^{\mathcal{P}_{\mathbf{q}}}(\mathbf{w}^{i^*}) \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\frac{1}{T^2} \sum_{i^* \in [T^2]} \mathrm{err}_{0.1\gamma}^{\tilde{\mathcal{P}}_{\mathbf{q}}}(\mathbf{w}^{i^*}) \right] \\
&= \mathbb{E}_{\mathbf{q}_1, \ldots, \mathbf{q}_{T^2}, \mathbf{w} \sim \mathbb{A}(\mathcal{Q}^n)} \left[\mathrm{err}_{0.1\gamma}^{\mathcal{Q}}(\mathbf{w}) \right].
\end{aligned}$$

Now, notice that the halfspace $\mathbf{w}^* = \frac{1}{T}\sum_{i\in[T^2]} \mathbf{q}'_i \otimes \mathbf{e}_i$ (whose Euclidean norm is at most one) correctly classifies each point in $\mathrm{supp}(\mathcal{Q})$ with margin $\frac{0.01}{T} \geq \gamma$. As a result, the accuracy guarantee of $\mathbb{A}$ ensures that $\mathbb{E}_{\mathbf{w}\sim\mathbb{A}(\mathcal{Q}^n)}[\mathrm{err}^{\mathcal{Q}}_{0.1\gamma}(\mathbf{w})] \leq \alpha(1-\xi)+\xi \leq \beta_0$. Plugging into the above, we have

$$\mathbb{E}_{\mathbf{q}\sim\mathcal{U}^{\otimes d}_{[0,1]},\mathbf{w}\sim\mathbb{B}_{\tilde{\mathcal{P}}_{\mathbf{q}}}}[\mathrm{err}^{\tilde{\mathcal{P}}_{\mathbf{q}}}_{\gamma_0}(\mathbf{w})] \leq \beta_0,$$

which contradicts with Lemma 14. □

D Lower Bound for Non-Robust Learning of Halfspaces

In this section, we provide a lower bound of $\Omega\left(\frac{1}{\epsilon\gamma^2}\right)$ on the sample complexity of *non-robust* learners (Theorem 4). While quantitatively similar, our lower bound significantly strengthens that of Nguyen et al. (2020) in two aspects: (1) our lower bounds hold against even *improper* learners whereas the lower bound in (Nguyen et al., 2020) is only valid against *proper* learners and (2) our lower bound holds even against (ϵ,δ)-DP algorithms whereas that of Nguyen et al. (2020) is only valid when $\delta = 0$.

Theorem 4. *For any $\epsilon > 0$, there exists $\delta > 0$ such that any (ϵ,δ)-DP $(\gamma,0)$-robust (possibly improper) learner has sample complexity $\Omega\left(\frac{1}{\epsilon\gamma^2}\right)$. Moreover, this holds even when $d = O(1/\gamma^2)$.*

To prove the above, we will require the following simple lemma, which states that the task of outputting an input bit requires $\Omega(1/\epsilon)$ equal samples in order to gain any non-trivial advantage over random guessing. The proof follows a straightforward packing argument.

Lemma 15. *For $s \in \{\pm 1\}$, let O_s denote the distribution which is s with probability 1. For any $\epsilon > 0$, there exists $\delta = \Omega(1/\epsilon)$ such that the following holds: There is no (ϵ,δ)-DP algorithm that can take at most $10^{-5}/\epsilon$ samples in expectation from O_s for a random $s \in \{\pm 1\}$ and output s correctly with probability 0.51.*

Proof. We may assume that $\epsilon < 1$ as it is clear that the algorithm needs at least one sample to output s correctly with probability 0.51. Furthermore, let $\delta = \frac{0.001}{1-e^{-\epsilon}}$.

Suppose for the sake of contradiction that there is an algorithm $\mathbb{A}$ that takes in at most $10^{-5}/\epsilon$ samples in expectation and output s correctly with probability 0.51. By Markov inequality, with probability 0.999, $\mathbb{A}$ takes at most $n := \lfloor 0.01/\epsilon \rfloor$ samples. Let $\mathbb{B}$ be the modification of $\mathbb{A}$ where $\mathbb{B}$ draws n samples and runs $\mathbb{A}$ on them but fails whenever $\mathbb{A}$ attempts to draw more than n samples. We have that $\mathbb{B}$ outputs s correctly with probability 0.509. In other words, we have

$$\Pr[\mathbb{B}(s^n) = s] \geq 0.509, \tag{6}$$

where s^n denote n inputs all equal to s.

Since $\mathbb{A}$ is (ϵ,δ)-DP, $\mathbb{B}$ is also (ϵ,δ)-DP. Suppose without loss of generality that $\Pr[\mathbb{B}(\emptyset) \neq 1] \geq \Pr[\mathbb{B}(\emptyset) \neq -1]$. This implies that $\Pr[\mathbb{B}(\emptyset) \neq 1] \geq 0.5$. From (ϵ,δ)-DP of $\mathbb{B}$, we have

$$\begin{aligned}
\Pr[\mathbb{B}(1^n) \neq 1] &\geq e^{-\epsilon}\Pr[\mathbb{B}(1^{n-1}) \neq 1] - \delta \\
&\vdots \\
&\geq e^{-n\epsilon}\Pr[\mathbb{B}(\emptyset) \neq 1] - \delta(1 + e^{-\epsilon} + \cdots + e^{-n\epsilon}) \\
&\geq e^{-0.01}\cdot 0.5 - 0.001 \\
&> 0.491
\end{aligned}$$

which contradicts (6). This concludes our proof. □

We can now prove Theorem 4. Roughly speaking, we "embed" the hard problem in Lemma 15 into each of the $d = 1/\gamma^2$ dimensions, which results in the $d\cdot\Omega(1/\epsilon) = \Omega\left(\frac{1}{\epsilon\gamma^2}\right)$ lower bound.

Proof of Theorem 4. We prove this statement for any $\gamma < 1, \alpha \leq 0.4$ and $\xi \leq 0.0001$.

Let δ be the same as in Lemma 15, and let $d = \lfloor 1/\gamma^2 \rfloor$. Suppose for the sake of contradiction that there exists an (ϵ, δ)-DP $(\gamma, 0)$-robust learner $\mathbb{A}$ that takes in at most $n := \lfloor 10^{-5} d/\epsilon \rfloor$ samples and outputs a hypothesis with error at most $\alpha \leq 0.4$ with probability $1 - \xi \geq 0.9999$. We will use $\mathbb{A}$ to construct an algorithm $\mathbb{B}$ that can solve the problem in Lemma 15.

For every $i \in [d]$ and $s \in \{\pm 1\}$, we use $\mathcal{D}_{i,s}$ to denote the uniform distribution on $(\mathbf{e}_i, s)$ and $(-\mathbf{e}_i, -s)$. Furthermore, for $\mathbf{s} \in \{\pm 1\}^d$, we use $\mathcal{D}_{\mathbf{s}}$ to denote the mixture $\frac{1}{d}\sum_{i\in[d]} \mathcal{D}_{i,s_i}$. Our algorithm $\mathcal{B}$ works as follows:

- Randomly sample $\mathbf{s} \in \{\pm 1\}^d$ and randomly sample $i^* \in [d]$.
- Draw n samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ independently as follows:
  - Randomly pick $i \in [d]$.
  - If $i \neq i^*$, then return a sample drawn from $\mathcal{D}_{i,s_i}$.
  - Otherwise, if $i = i^*$, sample $a \sim O_s$. Then return the sample $(\mathbf{e}_i, a)$ with probability 0.5; otherwise, return the sample $(-\mathbf{e}_i, -a)$.
- Run $\mathbb{A}$ on $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ to get a hypothesis h.
- With probability 0.5, return $h(\mathbf{e}_{i^*})$. Otherwise, return $-h(-\mathbf{e}_{i^*})$.

It is obvious to see that $\mathbb{B}$ is (ϵ, δ)-DP and that the expected number of samples $\mathbb{B}$ draws from O_s is $n/d \leq 10^{-5}/\epsilon$. Hence, we only need to show that $\mathbb{B}$ outputs a correct answer with probability 0.51 to get a contradiction with Lemma 15.

Since s is uniformly draw from $\{\pm 1\}$, the probability that $\mathbb{B}$ outputs the *incorrect* answer is equal to

$$\begin{aligned}
&\mathbb{E}_{\mathbf{s}\sim\{\pm1\}^d, i\in[d], h\sim\mathbb{A}(\mathcal{D}_{\mathbf{s}}^{\otimes n})}\left[\frac{1}{2}\mathbb{1}\left[h(\mathbf{e}_i) \neq s_i\right] + \frac{1}{2}\mathbb{1}\left[h(-\mathbf{e}_i) \neq -s_i\right]\right] \\
&= \mathbb{E}_{\mathbf{s}\sim\{\pm1\}^d, h\sim\mathbb{A}(\mathcal{D}_{\mathbf{s}}^{\otimes n})}\left[\frac{1}{d}\sum_{i\in[d]}\left(\frac{1}{2}\mathbb{1}\left[h(\mathbf{e}_i) \neq s_i\right] + \frac{1}{2}\mathbb{1}\left[h(-\mathbf{e}_i) \neq -s_i\right]\right)\right] \\
&= \mathbb{E}_{\mathbf{s}\sim\{\pm1\}^d, h\sim\mathbb{A}(\mathcal{D}_{\mathbf{s}}^{\otimes n})}\left[\mathrm{err}_0^{\mathcal{D}_{\mathbf{s}}}(h)\right].
\end{aligned}$$

Now, notice that any $(\mathbf{x}, y) \in \mathrm{supp}(\mathcal{D}_{\mathbf{s}})$ is correctly classified by the halfspace $\mathbf{z} := \frac{1}{\sqrt{d}}\sum_{i\in[d]} \mathbf{e}_i$ with margin $1/\sqrt{d} \geq \gamma$. As a result, the accuracy guarantee of $\mathbb{A}$ ensures that $\mathbb{E}_{h\sim\mathbb{A}(\mathcal{D}_{\mathbf{s}}^{\otimes n})}[\mathrm{err}_0^{\mathcal{D}_{\mathbf{s}}}(h))] \leq 1 \cdot 0.0001 + 0.4 \cdot 0.9999 < 0.41$. Thus, we can conclude that $\mathbb{B}$ outputs the *correct* answer with probability at least $1 - 0.41 > 0.59$. This contradicts Lemma 15. □

E Pure DP Robust Learner

In this section, we give a pure-DP algorithm for robust learning of halfspaces:

Theorem 5. *There is an ϵ-DP $(\gamma, 0.9\gamma)$-robust learner with sample complexity*[10] $O_\alpha\left(\frac{1}{\epsilon}\max\{d, \frac{1}{\gamma^2}\}\right)$.

To prove this result, we will also need the following generalization bound due to Bartlett and Mendelson (2002):

Lemma 16 (Generalization Bound for Large Margin Halfspaces (Bartlett and Mendelson, 2002))**.** *Suppose $\hat{\gamma}, \hat{\xi} \in [0, 1]$ and let $\mathcal{D}$ be any distribution on $\mathbb{B}^d \times \{\pm 1\}$. If we let $\mathbf{X}$ be drawn from $\mathcal{D}^{\otimes n}$, then the following holds with probability $1 - \hat{\xi}$:*

$$\forall \mathbf{w} \in \mathbb{B}^d, \mathrm{err}_{0.95\hat{\gamma}}^{\mathcal{D}}(\mathbf{w}) \leq \mathrm{err}_{\hat{\gamma}}^{\mathbf{X}}(\mathbf{w}) + 400\sqrt{\frac{\ln(4/\hat{\xi})}{n\hat{\gamma}^2}}.$$

[10] Here $O_\alpha(\cdot)$ hides a factor of $\mathrm{poly}(1/\alpha)$, and $\tilde{O}(\cdot)$ hides a factor of $\mathrm{poly}\log(1/(\alpha\gamma\delta))$.

Proof of Theorem 5. We will prove this for $\xi = 0.9$. Let $\Lambda = 10^6 \cdot \sqrt{\log(1/\alpha)} \cdot \max\{\sqrt{d}, 1/\gamma\}$, and $n = \frac{10^4\Lambda^2}{\epsilon\alpha} + \frac{10^{10}}{\alpha^2\gamma^2} = O\left(\frac{\log(1/\alpha)}{\alpha\epsilon} \cdot \max\{d, 1/\gamma^2\} + \frac{1}{\alpha^2\gamma^2}\right)$.

Our algorithm samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ from $\mathcal{D}$, and then employs the exponential mechanism of McSherry and Talwar (2007). Specifically, let μ be the density of the uniform measure over the unit sphere in $\mathbb{R}^d$. Then, on the input dataset $\mathbf{X} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$, we define the scoring function q by

$$q(\mathbf{X}, \mathbf{w}) = -n \cdot \mathrm{err}^{\mathbf{X}}_{0.95\gamma}(\mathbf{w}).$$

Then, we output $\hat{\mathbf{w}}$ drawn from the distribution with density $\mu'(\mathbf{w}) \propto \mu(\mathbf{w}) \cdot \exp\left(\frac{\epsilon}{2} \cdot q(\mathbf{X}, \mathbf{w})\right)$.

We will next argue the accuracy guarantee of the algorithm. Similar to (McSherry and Talwar, 2007), let $S_t := \{\mathbf{w} \mid q(\mathbf{X}, \mathbf{w}) \geq -t\}$. We start by showing that, with probability 0.99, we have $\hat{\mathbf{w}} \in S_{0.5\alpha n}$. To prove this, we will use the following result from (McSherry and Talwar, 2007):

Lemma 17. *For any $t \geq 0$, $\Pr[\hat{\mathbf{w}} \notin S_{2t}] \leq \exp(-\epsilon t/2)/\mu(S_t)$.*

In light of Lemma 17, it suffices for us to provide a lower bound for $\mu(S_{0.25\alpha n})$. Recall from the realizable assumption that, there exists a unit-norm $\mathbf{w}^*$ such that $\mathrm{err}^{\mathbf{X}}_{\gamma}(\mathbf{w}^*) = 0$. Since $\mu(S_{0.25\alpha n})$ is rotational-invariant, we may assume for notational convenience that $\mathbf{w}^* = \mathbf{e}_d$, the d-th vector in the standard basis. In this notation, a sample $\mathbf{w} \sim \mu$ may be obtained by:

- Sample $w_d \sim \mathcal{N}(0, 1)$,
- Sample $\mathbf{w}_\perp \sim \mathcal{N}(0, I_{(d-1)\times(d-1)})$,
- Let $\mathbf{w} = \frac{1}{T}(\mathbf{w}_\perp \circ w_d)$ where $T = \sqrt{\|\mathbf{w}_\perp\|^2 + w_d^2}$.

Fix $i \in [n]$. We will now bound the probability $\Pr[y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0.95\gamma \mid w_d \geq \Lambda]$. Let us write $y_i\mathbf{x}_i$ as $\mathbf{x}_\perp \circ x_d$. $\langle \mathbf{w}_\perp, \mathbf{x}_\perp \rangle$ is distributed as $\mathcal{N}(0, \|\mathbf{x}_\perp\|)$. Since $\|\mathbf{x}_\perp\| \leq 1$, we may apply standard tail bound of Gaussian which gives

$$\Pr[\langle \mathbf{w}_\perp, \mathbf{x}_\perp \rangle < -0.01\Lambda/\gamma] \leq \Pr[\langle \mathbf{w}_\perp, \mathbf{x}_\perp \rangle < 10^4\sqrt{\log(1/\alpha)}] \leq 0.1\alpha. \tag{7}$$

Observe also that $\|\mathbf{w}_\perp\|^2$ is simply distributed as χ^2_{d-1}. Hence, via standard tail bound (e.g., (Laurent and Massart, 2000)), we have

$$\Pr[\|\mathbf{w}_\perp\| > 0.01\Lambda] \leq \Pr[\|\mathbf{w}_\perp\| > 10^4\sqrt{d\log(1/\alpha)}] \leq 0.1\alpha. \tag{8}$$

Furthermore, notice that when $w_d \geq \Lambda$, $\langle \mathbf{w}_\perp, \mathbf{x}_\perp \rangle \geq -0.01\Lambda/\gamma$ and $\|\mathbf{w}_\perp\| \leq 0.01\Lambda$, we have $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0.95\gamma$. As a result, a union bound and the independence of w_d and $\mathbf{w}_\perp$ implies that

$$\begin{aligned}\Pr[y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0.95\gamma \mid w_d \geq \Lambda] &\leq \Pr[\langle \mathbf{w}_\perp, \mathbf{x}_\perp \rangle < -0.01\Lambda/\gamma] + \Pr[\|\mathbf{w}_\perp\| > 0.01\Lambda] \\ &\leq 0.2\alpha. \end{aligned} \tag{9}$$

From (9) and from the linearity of the expectation, we have that

$$\mathbb{E}[|\{i \in [n] \mid y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0.95\gamma\}| \mid w_d \geq \Lambda] \leq 0.2\alpha n.$$

By Markov's inequality, we may conclude that

$$\Pr[\mathbf{w} \in S_{0.25\alpha n} \mid w_d \geq \Lambda] \geq 0.1.$$

Finally, recall that w_d is distributed as $\mathcal{N}(0, 1)$, which implies that $\Pr[w_d \geq \Lambda] \geq 2^{-10\Lambda^2}$. This gives

$$\mu(S_{0.25\alpha n}) = \Pr[\mathbf{w} \in S_{0.25\alpha n} \mid w_d \geq \Lambda] \Pr[w_d \geq \Lambda] \geq 0.1 \cdot 2^{-10\Lambda^2} \geq 2^{-20\Lambda^2}. \tag{10}$$

Hence, applying Lemma 17, we get that

$$\Pr[\hat{\mathbf{w}} \notin S_{0.5\alpha n}] \le \frac{\exp(-\epsilon t/2)}{\mu(S_t)} \overset{(10)}{\le} \frac{\exp(-0.125\epsilon\alpha n)}{2^{-20\Lambda^2}}$$

(From our choice of n) ≤ 0.99.

In other words, with probability 0.99, we have $\mathrm{err}^{\mathbf{X}}_{0.95\gamma}(\hat{\mathbf{w}}) \le 0.5\alpha$. Finally, via the generalization bound (Lemma 16 with $\hat{\gamma} = 0.95\gamma$), we also have $\mathrm{err}^{\mathcal{D}}_{0.9\gamma}(\hat{\mathbf{w}}) \le \alpha$ with probability 0.9 as desired. □

F Approximate-DP Robust Learner

In this section, we describe our approximate-DP learner and prove its guarantee, restated below:

Theorem 6. *There is an (ϵ, δ)-DP $(\gamma, 0.9\gamma)$-robust learner with sample complexity $n = \tilde{O}_\alpha\left(\frac{1}{\epsilon} \cdot \max\left\{\frac{\sqrt{d}}{\gamma}, \frac{1}{\gamma^2}\right\}\right)$ and running time $\tilde{O}_\alpha(nd/\gamma)$.*

As alluded to earlier, this algorithm is a noised and batch version of the margin perceptron algorithm (Duda and Hart, 1973; Collobert and Bengio, 2004). The algorithm is presented in Algorithm 1.

The rest of this section is organized as follows. In the next subsection, we provide the utility analysis of the algorithm. Then, in Subsection F.2, we analyze its privacy guarantee. Finally, we set the parameters and prove Theorem 6 in Section F.3.

F.1 Utility Analysis

Suppose that there exists $\mathbf{w}^* \in \mathbb{B}^d$ with $\mathrm{err}_\gamma(\mathbf{w}^*) = 0$. Furthermore, let $\gamma' = 0.95\gamma$, $\gamma_{\mathrm{gap}} := \gamma - \gamma'$ and $B := pn$. Throughout the analysis, we will assume that the following "good" events occur:

- $E_{\text{batch-size}}$: For all $i \in [T]$, $|S_i| \le 1.5B$.
- $E_{\text{noise-norm}}$: For all $i \in [T]$, $\|\mathbf{g}_i\| \le B\sqrt{\alpha}$.
- E_{parallel}: For all $i \in [T]$, $\langle \mathbf{w}_{i-1} + \mathbf{u}_i, \mathbf{g}_i \rangle \le 0.01\alpha\gamma_{\mathrm{gap}} B \cdot \|\mathbf{w}_{i-1} + \mathbf{u}_i\|$.
- $E_{\text{opt-noise}}$: For all $i \in [T]$, $\langle \mathbf{w}^*, \mathbf{g}_i \rangle \ge -0.01\alpha\gamma_{\mathrm{gap}} B$.
- $E_{\text{mistake-noise}}$: For all $i \in [T]$, $\nu_i \in [-0.1\alpha B, 0.1\alpha B]$.
- $E_{\text{sampled-mistake}}$: For all $i \in [T]$ such that[11] $\mathrm{err}^{\mathbf{X}}_{\gamma'}\left(\frac{\mathbf{w}_{i-1}}{\|\mathbf{w}_{i-1}\|}\right) > 0.5\alpha$, we have $|M_i| \ge 0.4\alpha B$.

Later on, we will select the parameters p, n, T, b, σ so that these events happen with high probability.

Lemma 18. *Let $T = \lceil \frac{1500}{\alpha\gamma_{gap}^2} \rceil$. If the events $E_{batch\text{-}size}, E_{noise\text{-}norm}, E_{parallel}, E_{opt\text{-}noise}, E_{mistake\text{-}noise}$ and $E_{sampled\text{-}mistake}$ all occur, then the algorithm outputs $\mathbf{w}$ such that $\mathrm{err}^{\mathbf{X}}_{\gamma'}(\mathbf{w}) \le 0.5\alpha$.*

Proof. We will show that we always execute Line 10. Once this is the case, $E_{\text{mistake-noise}}$ and $E_{\text{sampled-mistake}}$ imply that the output $\mathbf{w}_i/\|\mathbf{w}_i\|$ satisfies $\mathrm{err}^{\mathbf{X}}_{\gamma'}(\mathbf{w}_i/\|\mathbf{w}_i\|) \le 0.5\alpha$ as desired.

To prove that we execute Line 10, let us assume for the sake of contradiction that this is not the case, i.e., that the algorithm continues until reaching the end of the T-th iteration.

From our assumption that $E_{\text{mistake-noise}}$ occurs and from the fact that Line 10 was not executed, we have $|M_i| \ge 0.2\alpha B$ for all $i \in [T]$. Let $m_i := \sum_{j \in [i]} |M_i|$ denote the number of γ'-margin mistakes seen up until the end of the i-th iteration; from the previous bound on M_i, we have

$$m_i \ge 0.2\alpha B i. \tag{11}$$

[11] Similar to before, we use $\mathbf{X}$ to denote $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Now, notice that

$$\begin{aligned}
\langle \mathbf{w}^*, \mathbf{w}_T \rangle &= \left\langle \mathbf{w}^*, \left(\sum_{i\in[T]} \sum_{(\mathbf{x},y)\in M_i} y \cdot \mathbf{x} \right) + \sum_{i\in[T]} \mathbf{g}_i \right\rangle \\
&= \left(\sum_{i\in[T]} \sum_{(\mathbf{x},y)\in M_i} y \cdot \langle \mathbf{w}^*, \mathbf{x} \rangle \right) + \sum_{i\in[T]} \langle \mathbf{w}^*, \mathbf{g}_i \rangle \\
(\text{From } \mathrm{err}^{\mathbf{X}}_{\gamma'}(\mathbf{w}^*) = 0 \text{ and } E_{\text{opt-noise}}) &\geq \left(\sum_{i\in[T]} \sum_{(\mathbf{x},y)\in M_i} \gamma \right) + \sum_{i\in[T]} -0.01\alpha\gamma_{\text{gap}} B \\
&= m_T\gamma - 0.01\alpha\gamma_{\text{gap}} BT \\
&\overset{(11)}{\geq} m_T(\gamma - 0.05\gamma_{\text{gap}}). \qquad (12)
\end{aligned}$$

Furthermore, for every $i \in [T]$, we have that

$$\begin{aligned}
\|\mathbf{w}_i\|^2 &= \|\mathbf{w}_{i-1} + \mathbf{u}_i + \mathbf{g}_i\|^2 \\
&= \|\mathbf{w}_{i-1} + \mathbf{u}_i\|^2 + 2\langle \mathbf{w}_{i-1} + \mathbf{u}_i, \mathbf{g}_i \rangle + \|\mathbf{g}_i\|^2 \\
(\text{From } E_{\text{parallel}}) &\leq \|\mathbf{w}_{i-1} + \mathbf{u}_i\|^2 + 0.02\alpha\gamma_{\text{gap}} B \cdot \|\mathbf{w}_{i-1} + \mathbf{u}_i\| + \|\mathbf{g}_i\|^2 \\
(\text{From } E_{\text{noise-norm}}) &\leq \|\mathbf{w}_{i-1} + \mathbf{u}_i\|^2 + 0.02\alpha\gamma_{\text{gap}} B \cdot \|\mathbf{w}_{i-1} + \mathbf{u}_i\| + \alpha B^2 \\
&\leq \|\mathbf{w}_{i-1}\|^2 + 2\langle \mathbf{w}_{i-1}, \mathbf{u}_i \rangle + \|\mathbf{u}_i\|^2 + 0.02\alpha\gamma_{\text{gap}} B \cdot (\|\mathbf{w}_{i-1}\| + \|\mathbf{u}_i\|) + \alpha B^2 \qquad (13)
\end{aligned}$$

We can bound $\langle \mathbf{w}_{i-1}, \mathbf{u}_i \rangle$ as follows:

$$\langle \mathbf{w}_{i-1}, \mathbf{u}_i \rangle = \sum_{(\mathbf{x},y)\in M_i} y \cdot \langle \mathbf{w}_{i-1}, \mathbf{x} \rangle \leq |M_i| \cdot \gamma' \|\mathbf{w}_{i-1}\|,$$

where the inequality follows from the condition on Line 6.

Furthermore, we also have that

$$\|\mathbf{u}_i\| = \left\| \sum_{(\mathbf{x},y)\in M_i} y \cdot \mathbf{x} \right\| \leq \sum_{(\mathbf{x},y)\in M_i} \|\mathbf{x}\| \leq |M_i|.$$

Plugging the above two inequalities into (13), we get

$$\begin{aligned}
\|\mathbf{w}_i\|^2 &\leq \|\mathbf{w}_{i-1}\|^2 + (2|M_i|\gamma' + 0.02\alpha\gamma_{\text{gap}} B) \cdot \|\mathbf{w}_{i-1}\| + (|M_i|^2 + 0.02\alpha\gamma_{\text{gap}} B|M_i| + \alpha B^2) \\
&\leq \|\mathbf{w}_{i-1}\|^2 + (2|M_i|\gamma' + 0.02\alpha\gamma_{\text{gap}} B) \cdot \|\mathbf{w}_{i-1}\| + (|M_i|^2 + 0.02B \cdot |M_i| + \alpha B^2) \\
&\leq \|\mathbf{w}_{i-1}\|^2 + (2|M_i|\gamma' + 0.02\alpha\gamma_{\text{gap}} B) \cdot \|\mathbf{w}_{i-1}\| + 2B|M_i| + \alpha B^2,
\end{aligned}$$

where in the last inequality we use the fact that $|M_i| \leq 1.5B$ which follows from $E_{\text{batch-size}}$.

The above inequality implies that

$$\|\mathbf{w}_i\| \leq \|\mathbf{w}_{i-1}\| + (|M_i|\gamma' + 0.01\alpha\gamma_{\text{gap}} B) + \frac{B|M_i| + 0.5\alpha B^2}{\|\mathbf{w}_{i-1}\|}.$$

Notice that when $\|\mathbf{w}_{i-1}\| \geq \frac{100B}{\gamma_{\text{gap}}}$, we have that

$$\begin{aligned}
\|\mathbf{w}_i\| &\leq \|\mathbf{w}_{i-1}\| + (|M_i|\gamma' + 0.01\alpha\gamma_{\text{gap}} B) + 0.01|M_i|\gamma_{\text{gap}} + 0.01\alpha\gamma_{\text{gap}} B \\
&= \|\mathbf{w}_{i-1}\| + 0.02\alpha\gamma_{\text{gap}} B + (\gamma' + 0.01\gamma_{\text{gap}}) \cdot |M_i|.
\end{aligned}$$

As a result, we get[12]

$$\begin{aligned}\|\mathbf{w}_T\| &\le \frac{200B}{\gamma_{\text{gap}}} + 0.02\alpha\gamma_{\text{gap}}BT + (\gamma' + 0.01\gamma_{\text{gap}}) \cdot \left(\sum_{i\in[T]} |M_i|\right) \\ &= \frac{200B}{\gamma_{\text{gap}}} + 0.02\alpha\gamma_{\text{gap}}BT + (\gamma' + 0.01\gamma_{\text{gap}}) \cdot m_T \\ &\overset{(11)}{\le} \frac{200B}{\gamma_{\text{gap}}} + (\gamma' + 0.11\gamma_{\text{gap}}) \cdot m_T. \end{aligned} \tag{14}$$

From (12) and (14), we have

$$m_T(\gamma - 0.05\gamma_{\text{gap}}) \le \frac{200B}{\gamma_{\text{gap}}} + m_T\left(\gamma' + 0.11\gamma_{\text{gap}}\right),$$

which implies that

$$\begin{aligned} m_T &\le \frac{200B}{\gamma_{\text{gap}}\left(\gamma - \gamma' - 0.16\gamma_{\text{gap}}\right)} \\ &< \frac{200B}{0.8\gamma_{\text{gap}}^2} \\ &= 250B/\gamma_{\text{gap}}^2, \end{aligned}$$

which contradicts (11) and our choice of $T = \lceil \frac{1500}{\alpha\gamma_{\text{gap}}^2} \rceil$. □

F.2 Privacy Analysis

Lemma 19. *For any $\epsilon, \delta \in (0,1)$ and any $T \in \mathbb{N}$, let $p = \frac{1}{\sqrt{T}}, \sigma = \frac{100\ln(T/\delta)}{\epsilon}$ and $b = \frac{100\sqrt{\ln(T/\delta)}}{\epsilon}$. Then, Algorithm 1 is (ϵ, δ)-DP.*

To prove this, we require the following the results on amplification by subsampling[13] and advanced composition.

Lemma 20 (Amplification by Subsampling (Balle et al., 2018))**.** *Let $\mathbb{A}$ be any (ϵ_0, δ_0)-DP algorithm such that $\epsilon_0, \delta_0 \in (0,1)$. Let $\mathbb{B}$ be an algorithm that independently selects each input sample w.p. p and runs $\mathbb{A}$ on this subsampled input dataset. Then, $\mathbb{B}$ is $(2p\epsilon_0, p\delta_0)$-DP.*

Lemma 21 (Advanced Composition (Dwork et al., 2010))**.** *Suppose that $\mathbb{B}$ is an algorithm resulting from running an (ϵ_0, δ_0)-DP algorithm T times (possibly adaptively), where $\epsilon_0, \delta_0 \in (0,1)$. Then, $\mathbb{B}$ is $(\epsilon', (T+1)\delta_0)$-DP where*

$$\epsilon' = \sqrt{2T\ln(1/\delta_0)} \cdot \epsilon_0 + 2T\epsilon_0^2.$$

Proof of Lemma 19. Let $\epsilon_0 = \frac{\epsilon}{20\sqrt{\ln(T/\delta)}}$ and $\delta_0 = \frac{\delta}{2\sqrt{T}}$. The Gaussian mechanism with noise standard deviation σ is $(0.5\epsilon_0, \delta_0)$-DP (Dwork and Roth, 2014, Appendix A) whereas the Laplace mechanism with parameter b is $0.5\epsilon_0$-DP (Dwork et al., 2006b)[14]. As a result, without subsampling, each iteration is (ϵ_0, δ_0)-DP. With the subsampling, Lemma 20 implies that each iteration is $(2p\epsilon_0, p\delta_0)$-DP. Finally, we may apply Lemma 21 ensures that the final algorithm is (ϵ', δ')-DP for

$$\epsilon' = \sqrt{2T\ln(1/(p\delta))} \cdot (2p\epsilon_0) + 2T(2p\epsilon_0)^2 \le \epsilon,$$

and

$$\delta' = (T+1)p\delta_0 \le 2\sqrt{T}\delta_0 = \delta,$$

which concludes our proof. □

[12] Note that the first term $\frac{200B}{\gamma_{\text{gap}}}$ comes from an observation that if i_0 is the smallest index for which $\|\mathbf{w}_{i_0}\| \ge \frac{100B}{\gamma_{\text{gap}}}$, then (13) implies that $\|\mathbf{w}_{i_0}\|_0 \le \frac{200B}{\gamma_{\text{gap}}}$.

[13] Amplification by subsampling results are often stated with the new ϵ being $\ln(1 + p(e^\epsilon - 1))$ which is no more than $2p\epsilon$ (from Bernoulli's inequality and from $1 + x \le e^x$ for all $x \in \mathbb{R}$).

[14] In both cases, the ℓ_2 sensitivity and the ℓ_1 sensitivity respectively are bounded by one. For the former, this is because each sample effects $\mathbf{w}$ only by $y \cdot \mathbf{x}$ and $\|y \cdot \mathbf{x}\|_2 = \|\mathbf{x}\| \le 1$.

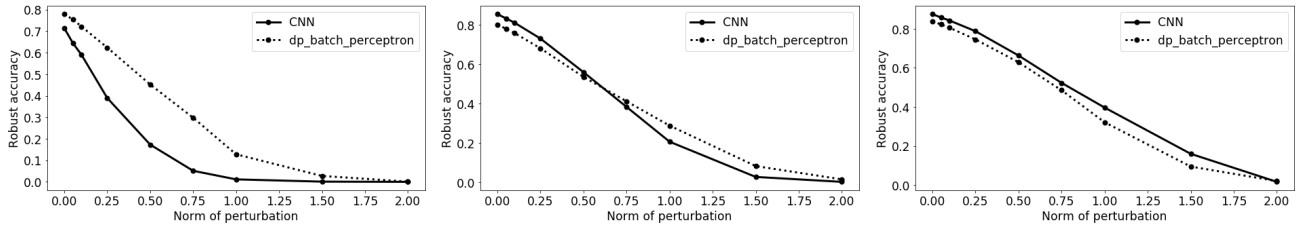
F.3 Putting Things Together

Proof of Theorem 6. Let $T = \lceil \frac{1500}{\alpha\gamma_{\text{gap}}^2} \rceil$ be as in Lemma 18, and let $p = \frac{1}{\sqrt{T}}, \sigma = \frac{100\ln(T/\delta)}{\epsilon}$ and $b = \frac{100\sqrt{\ln(T/\delta)}}{\epsilon}$ be as in Lemma 19. Finally, let $n = \lceil \frac{100\sqrt{d}\sigma\log T}{p\sqrt{\alpha}} + \frac{1000\sigma\sqrt{\log T}}{p\alpha\gamma} + \frac{100\log T}{\alpha} + \frac{10^{10}}{\alpha^2\gamma^2} \rceil$. Notice that $n = O_\alpha\left(\frac{1}{\epsilon\gamma}\left(\sqrt{d} + \frac{1}{\gamma}\right) \cdot (\log T)^2\right)$ as claimed.

From Lemma 19, our algorithm with the above parameters is (ϵ, δ)-DP. Furthermore, the expected running time of the algorithm is $pnT = O(n\sqrt{T}) = O\left(\frac{n}{\gamma\sqrt{\alpha}}\right)$. Moreover, it can be verified via standard concentration inequalities that all of the events required in Lemma 18 happens w.p. 0.99, which means that we output a halfspace $\mathbf{w}$ with $\text{err}_{\gamma'}^{\mathbf{X}}(\mathbf{w}) \le 0.5\alpha$. Finally, the generalization bound (Lemma 16 with $\hat{\gamma} = \gamma'$) implies that $\text{err}_{0.9\gamma}^{\mathcal{D}}(\mathbf{w}) \le \alpha$ as desired. □

G Additional Experiments

G.1 Adversarial Robustness Evaluation on USPS Dataset



(a) $\epsilon = 0.5$ (b) $\epsilon = 1$ (c) $\epsilon = 2$

Figure 3: Robustness accuracy comparison between DP-SGD-trained Convolutional neural networks and DP Batch Perceptron halfspace classifiers on USPS dataset for a fixed privacy budget. In all three plots, $\delta = 10^{-5}$ but ϵ varies from 0.5, 1, and 2.

We compare the robust accuracy of DP Batch Perceptron classifiers and DP-SGD-trained neural networks in Figure 3 for $\delta = 10^{-4}$ and $\epsilon = 0.5, 1, 2$. The architecture and parameters follow the same setup described in Section 4. In the case of $\epsilon = 0.5$, while both classifiers have similar test accuracies (without any perturbation, $\gamma = 0$), as γ increases, the robust accuracy rapidly degrades for the DP-SGD-trained neural network compared to that of the DP Batch Perceptron model. This overall trend persists for $\epsilon = 1$; the CNN starts off with larger test accuracy when $\gamma = 0$ but is eventually surpassed by the halfspace classifier as γ increases. On the other hand, when $\epsilon = 2$, the CNN maintains slightly higher robust accuracy for most perturbation norms in consideration.

G.2 Experiments with Gaussian Kernel

It is well-known that accuracy of linear classifiers for digit classifications can be significantly improved via kernel methods (see, e.g., (Lecun et al., 1998; Schölkopf et al., 1997)). Here we would like to privately train linear classifiers with Gaussian kernels. Recall that the Gaussian kernel is that of the form

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\hat{\sigma}^2}\right),$$

where $\hat{\sigma}$ is the so-called *width* parameter.

Unlike the standard (non-kernel) setting, it is unclear in the Gaussian kernel setting how the noise should be added to obtain DP; the kernel space themselves is not of finite dimension, and the classifier is typically only implicitly represented. To handle this, we follow the approach of Rahimi and Recht (2007) (also used in DP-SVM (Rubinstein et al., 2012)). Specifically, Rahimi and Recht (2007) shows that the following approximate embedding $\hat{\phi} : \mathbb{R}^d \to \mathbb{B}^{2\hat{d}}$ has a property that $\left\langle \hat{\phi}(\mathbf{x}), \hat{\phi}(\mathbf{x}') \right\rangle$ is close to $k(\mathbf{x}, \mathbf{x}')$:

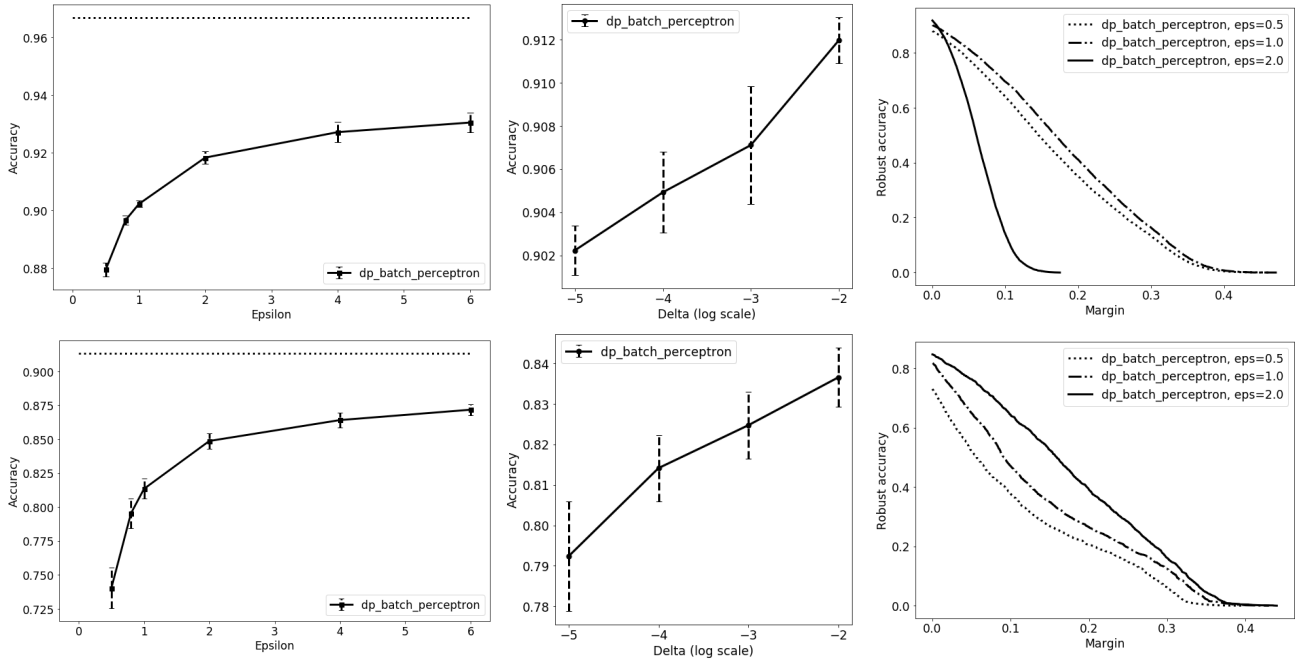
$$\hat{\phi}(\mathbf{x}) := \frac{1}{\sqrt{\hat{d}}}\left(\cos(\langle \rho_1, \mathbf{x}\rangle, \ldots, \cos(\langle \rho_{\hat{d}}, \mathbf{x}\rangle, \sin(\langle \rho_1, \mathbf{x}\rangle), \ldots, \sin(\langle \rho_{\hat{d}}, \mathbf{x}\rangle)\right),$$

where $\rho_1, \ldots, \rho_{\hat{d}}$ are i.i.d. sampled from $\mathcal{N}(\mathbf{0}, \frac{1}{\hat{\sigma}^2} \cdot I_{d\times d})$. Below we write σ^* to denote $1/\hat{\sigma}$.

To summarize, this approach allows us to train with (approximate) Gaussian kernel as follows (where $\sigma^*, \hat{d}$ are hyperparameters):

1. Randomly sample $\rho_1, \ldots, \rho_{\hat{d}}$ i.i.d. from $\mathcal{N}(\mathbf{0}, (\sigma^*)^2 \cdot I_{d\times d})$.
2. For each class y, use DP-Batch-Perceptron on $(\hat{\phi}(\mathbf{x}_1), y_1), \ldots, (\hat{\phi}(\mathbf{x}_n), y_n)$ to train a halfspace $\mathbf{w}^{(y)} \in \mathbb{R}^{2\hat{d}}$ for the y-vs-rest classifier.
3. When we would like to predict $\mathbf{x} \in \mathbb{R}^d$, compute $\text{argmax}_{y\in\{1,\ldots,10\}} \left\langle \mathbf{w}^{(y)}, \hat{\phi}(\mathbf{x}) \right\rangle$.

Notice here that the DP guarantee (in the second step) is exactly the same as the DP-Batch-Perceptron guarantee for the non-kernel setting. Similar to Figure 1, we report the (non-robust) test accuracy of DP Batch Perceptron algorithm with Gaussian kernel included in Figure 4, across different ϵ values (first column) and different δ values (middle column). We find that kernel learning helps to boost performance overall, the gain in accuracy is particularly significant in the case of MNIST dataset (top row).



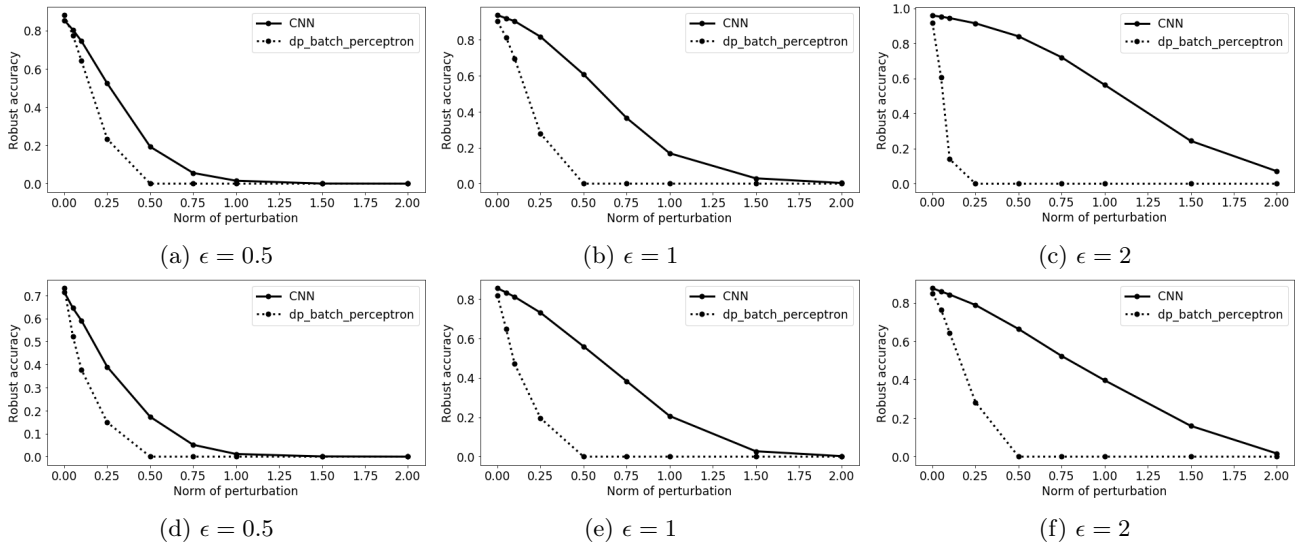
(a) Accuracy as ϵ varies (b) Accuracy as δ varies (c) Robust accuracy as ϵ varies

Figure 4: Performance on the MNIST (top row) and USPS (bottom row) datasets with Gaussian kernel. The horizontal dotted line indicates performance when $\epsilon = \infty$ (no noise). The width of the kernel for both datasets is tuned as a hyperparameter with values 2, 3.5, 5, 7.5, 10.

As for noise addition, the robustness guarantee for such kernel classifiers is also more complicated than the non-kernel linear classifiers. In Section G.2.1 below, we provide a *provable* robustness guarantee of the kernel classifiers. Using this provable guarantee, the empirical robustness accuracy is shown in the last column of Figure 4 and its comparison to DP-SGD-trained CNNs is shown in Figure 5. Even though the kernel classifiers start off with similar accuracy (at $\gamma = 0$), it quickly drops and becomes worse than CNNs. We remark here that, in addition to the nature of the kernel, this may also be exacerbated by the fact that the provable robust guarantee for the kernel classifiers is *not* tight (unlike the non-kernel case).

G.2.1 Robustness Guarantee for Multi-Class Perceptron with Kernel

To compute the robust error for the kernel classifiers, we will use the following result, which is a slightly simplified version of Theorem 2.1 from (Hein and Andriushchenko, 2017).



(a) $\epsilon = 0.5$ (b) $\epsilon = 1$ (c) $\epsilon = 2$

(d) $\epsilon = 0.5$ (e) $\epsilon = 1$ (f) $\epsilon = 2$

Figure 5: Robustness accuracy comparison between DP-SGD-trained Convolutional neural networks and DP Batch Perceptron halfspace classifiers with Gaussian kernel on MNIST (top row) and USPS (bottom row) datasets for a fixed privacy budget. In all three plots, $\delta = 10^{-5}$ but ϵ varies from 0.5, 1, and 2.

Lemma 22. *Let M be a classifier which, for each class $y \in \{1, \ldots, k\}$, computes some function $f^y : \mathbb{R}^d \to \mathbb{R}$ and predicts the class y^* that minimizes f^{y^*}. Then, for every example $(\mathbf{x}, y)$ and any $\Delta \in \mathbb{R}^d$ such that*

$$\|\Delta\| \leq \min_{y' \neq y} \frac{f^y(\mathbf{x}) - f^{y'}(\mathbf{x})}{\sup_{\mathbf{x}' \in \mathbb{R}^d} \|\nabla f^y(\mathbf{x}') - \nabla f^{y'}(\mathbf{x}')\|},$$

the classifier M predicts y on $\mathbf{x} + \Delta$.

Note that this lemma is tight for the non-kernel case, leading to the margin formula $\gamma < \min_{y' \neq y} \frac{\langle \mathbf{w}^{(y)}, \mathbf{x} \rangle - \langle \mathbf{w}^{(y')}, \mathbf{x} \rangle}{\|\mathbf{w}^{(y)} - \mathbf{w}^{(y')}\|}$ that we used earlier.

Our kernel classifier is of the form in Lemma 22 with $f^y(\mathbf{x}') := \langle \mathbf{w}^{(y)}, \phi_{\rho_1, \ldots, \rho_{\hat{d}}}(\mathbf{x}') \rangle$. To apply the lemma, we first compute ∇f:

$$\nabla f^y(\mathbf{x}') = \frac{1}{\sqrt{\hat{d}}} \cdot \sum_{i=1}^{\hat{d}} \left(-w_i^{(y)} \cdot \sin(\langle \rho_i, \mathbf{x}' \rangle) + w_{\hat{d}+i}^{(y)} \cdot \cos(\langle \rho_i, \mathbf{x}' \rangle) \right) \cdot \rho_i.$$

As a result, for two classes y, y', we have

$$\nabla f^y(\mathbf{x}') - \nabla f^{y'}(\mathbf{x}') = \frac{1}{\sqrt{\hat{d}}} \cdot \sum_{i=1}^{\hat{d}} \left((w_i^{(y')} - w_i^{(y)}) \cdot \sin(\langle \rho_i, \mathbf{x}' \rangle) + (w_{\hat{d}+i}^{(y)} - w_{\hat{d}+i}^{(y')}) \cdot \cos(\langle \rho_i, \mathbf{x}' \rangle) \right) \cdot \rho_i.$$

In the following, we will give an upper bound on $\|\nabla f^y - \nabla f^{y'}\|$. Let $\Pi \in \mathbb{R}^{d \times \hat{d}}$ resulting from concatenating $\rho_1, \ldots, \rho_{\hat{d}}$, and let $\mathbf{p} \in \mathbb{R}^{\hat{d}}$ denote the vector for which $p_i = \frac{1}{\sqrt{\hat{d}}} (w_i^{y'} - w_i^y) \cdot \sin(\langle \rho_i, \mathbf{x}' \rangle) + (w_{\hat{d}+i}^y - w_{\hat{d}+i}^{y'}) \cdot \cos(\langle \rho_i, \mathbf{x}' \rangle)$. First, notice that

$$\nabla f^y - \nabla f^{y'} = \Pi \mathbf{p}.$$

Now, we may bound $\|\mathbf{p}\|$ by

$$\|\mathbf{p}\| = \frac{1}{\sqrt{\hat{d}}} \cdot \sqrt{\sum_{i=1}^{\hat{d}} \left((w_i^{(y')} - w_i^{(y)}) \cdot \sin(\langle \rho_i, \mathbf{x}' \rangle) + (w_{\hat{d}+i}^{(y)} - w_{\hat{d}+i}^{(y')}) \cdot \cos(\langle \rho_i, \mathbf{x}' \rangle) \right)^2}$$

$$
\begin{aligned}
\text{(Cauchy–Schwarz inequality)} &\leq \frac{1}{\sqrt{\hat{d}}} \cdot \sqrt{\sum_{i=1}^{\hat{d}} \left((w_i^{(y')} - w_i^{(y)})^2 + (w_{\hat{d}+i}^{(y)} - w_{\hat{d}+i}^{(y')})^2 \right) \left(\sin(\langle \rho_i, \mathbf{x}' \rangle)^2 + \cos(\langle \rho_i, \mathbf{x}' \rangle)^2 \right)} \\
&= \frac{1}{\sqrt{\hat{d}}} \cdot \sqrt{\sum_{i=1}^{\hat{d}} \left((w_i^{(y')} - w_i^{(y)})^2 + (w_{\hat{d}+i}^{(y)} - w_{\hat{d}+i}^{(y')})^2 \right)} \\
&= \frac{1}{\sqrt{\hat{d}}} \cdot \|\mathbf{w}^{(y)} - \mathbf{w}^{(y')}\|
\end{aligned}
$$

As a result, we have

$$
\|\nabla f^y(\mathbf{x}') - \nabla f^{y'}(\mathbf{x}')\| = \|\Pi \mathbf{p}\| \leq \sigma_{\max}(\Pi) \cdot \frac{1}{\sqrt{\hat{d}}} \cdot \|\mathbf{w}^{(y)} - \mathbf{w}^{(y')}\|,
$$

where $\sigma_{\max}(\Pi)$ denote the largest singular value of Π (i.e. the operator norm of Π with respect to L_2 norm).

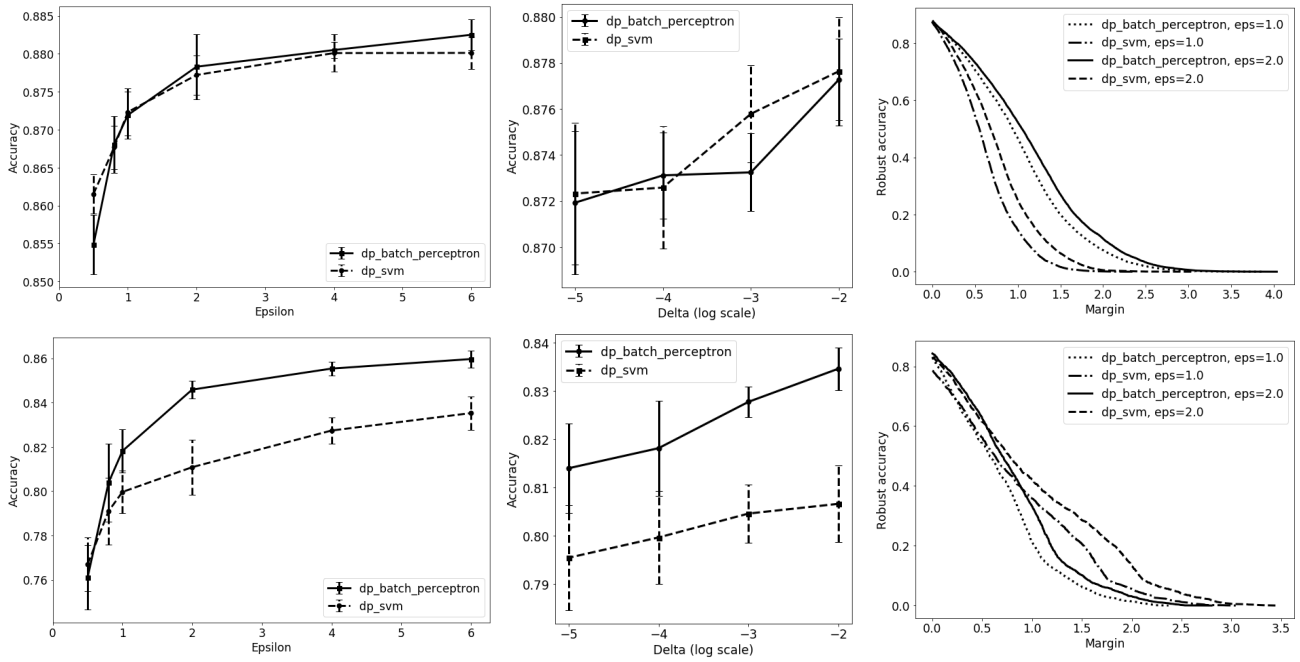
Plugging this back into Lemma 22, we can conclude that each example $(\mathbf{x}, y)$ remaining correctly classifies up to perturbation norm of

$$
\frac{\sqrt{\hat{d}}}{\sigma_{\max}(\Pi)} \cdot \min_{y' \neq y} \frac{f^y(\mathbf{x}) - f^{y'}(\mathbf{x})}{\|\mathbf{w}^{(y)} - \mathbf{w}^{(y')}\|}.
$$

G.3 Comparison with Support Vector Machines (SVM)

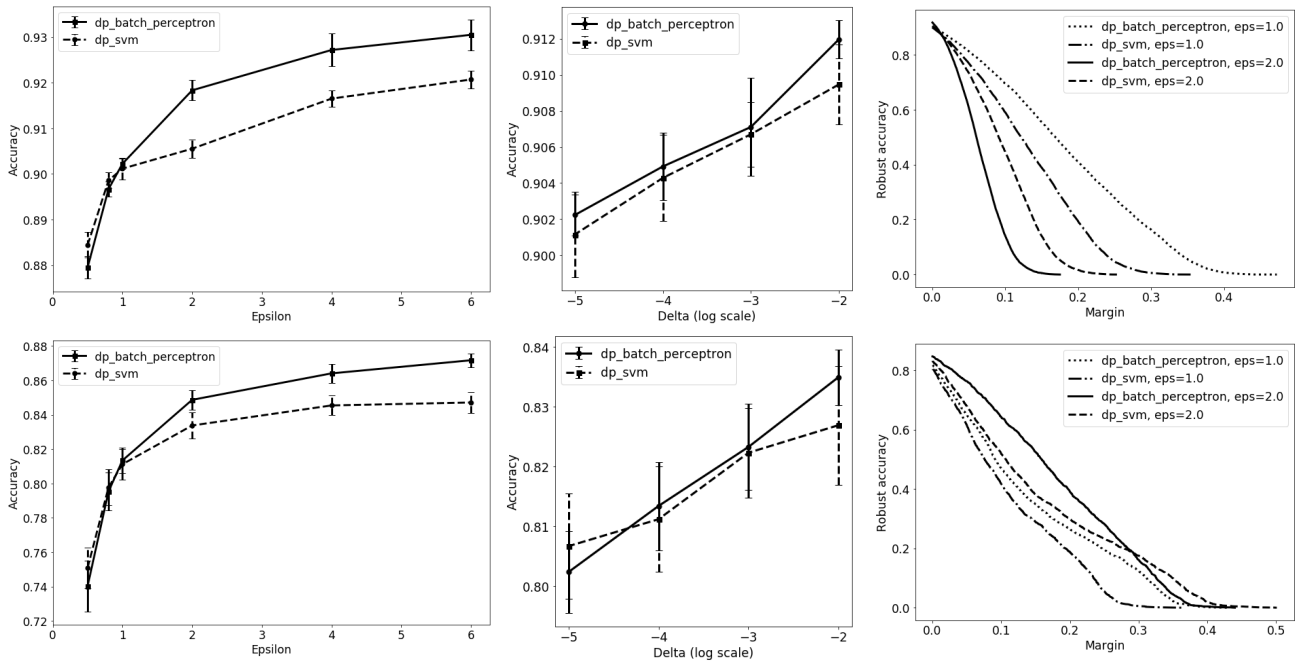
Previous work has introduced different approaches to preserving DP for SVM (Rubinstein et al., 2012), or convex optimization algorithms in general (Feldman et al., 2020; Bassily et al., 2019). Our implementation of DP SVM uses DP SGD (Abadi et al., 2016) with the standard hinge loss and L_2 weight regularization. The regularization strength is chosen from 1, 0.1, 0.01, 0.001, 0.0001, 0.00001, the learning rate from 1, 0.1, 0.01, 0.001, 0.0001. After summing gradients from each batch of data, we add appropriately calibrated Gaussian noise (again, based on Renyi DP) to the weights update.

Figures 6 and 7 compare performance of DP Batch Perceptron and DP SVM with and without kernel respectively. For experiments with varying ϵ (first column in both figures), we observe that DP Batch Perceptron outperforms DP SVM in most instances and achieves competitive accuracy on both datasets. For different δ values (second column) while keeping ϵ fixed at 1.0, the trend still holds to a large extent and both algorithms yield very similar test accuracy. The last column compares the robust accuracy of models trained via DP Batch Perceptron and DP SVM at $\epsilon = 1, 2$. In the case where no kernel is involved, the former yields better results on MNIST dataset but performs worse on USPS dataset. The opposite trend is observed when Gaussian kernel is included.



(a) Accuracy as ϵ varies (b) Accuracy as δ varies (c) Robust accuracy as ϵ varies

Figure 6: Comparison of performance of DP Batch Perceptron vs DP SVM halfspace classifiers on the MNIST (top row) and USPS (bottom row) datasets, when no kernel is involved in the learning process.



(a) Accuracy as ϵ varies (b) Accuracy as δ varies (c) Robust accuracy as ϵ varies

Figure 7: Comparison of performance of DP Batch Perceptron vs DP SVM halfspace classifiers on the MNIST (top row) and USPS (bottom row) datasets, with Gaussian kernel.