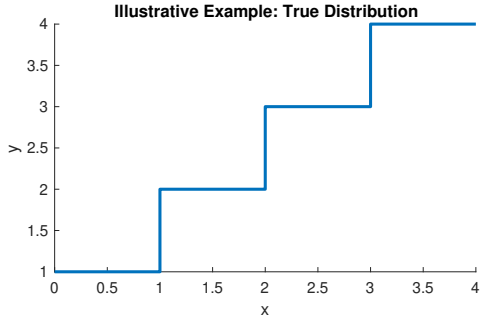

Fig. 3: True relationship between data (x) and target (y) used in the illustrative example in Section VIII-A.1. Training (with $n = 8$ points) and test (with $n = 100$ points) sets are uniformly sampled from the distribution.

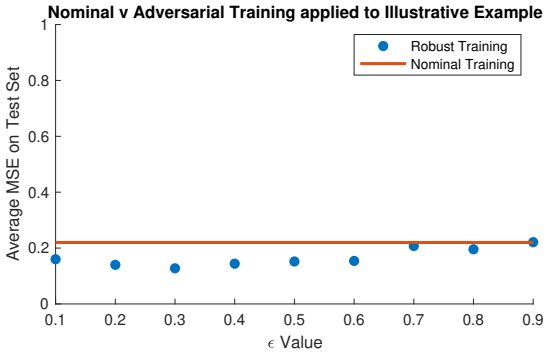

Fig. 4: This plot shows that the robust training approach (18) outperforms the standard approach for different $\epsilon \in \{0.1, ..., 0.9\}$ on the dataset studied in Section VIII-A.1.

APPENDIX

A. Additional experiments

1) Experiments with squared loss adversarial training: In this part of the appendix, the performance of the developed problem (18) was compared with the standard training problem (5) on a contrived 1-dimensional dataset. Figure 3 shows the true relationship between the data vector X and the target output y. Throughout this experiment, training data were constructed by uniformly sampling 8 points from this distribution and test data were similarly constructed by uniformly sampling 100 points. A bias term was included by concatenating a column of ones to X.

The training and testing procedure was repeated for 100 trials with standard training (Alg 1). For the adversarial training (Alg 2), we varied the perturbation radius $\epsilon = 0.1, \ldots, 0.9$. The training and testing procedure was carried out for 10 trials for each ϵ. Figure 4 reports the average test mean square error (MSE) for each setup.

The adversarial training procedure outperforms standard training for all ϵ choices. We further observe that the average MSE is the lowest at $\epsilon \approx 0.3$. This behavior arises as the robust problem attempts to account for all points within the uncertainty interval around the sampled training points. When ϵ is too small, the robust problem approaches the standard training problem. Larger values of ϵ cause the uncertainty interval to overestimate the constant regions of the true distribution, increasing the MSE.

2) Additional experiments on 2-D illustrative data: The decision boundaries obtained from various methods with different regularization strengths are shown in Figure 5. The two standard training methods (Alg 1 and GD-std) learned decision boundaries that separate the training points but fail to separate the perturbation boxes. Note that Alg 1 learned slightly more sophisticated boundaries while GD-std learned near-linear boundaries that were very close to one of the positive training points $\times$.

The convex adversarial training method Alg 2 learned boundaries that separates all perturbation boxes when β is 10^{-3}, 10^{-6}, or 10^{-9}. This behavior matches the theoretical illustration of adversarial training [11, Figure 3], and verifies that Alg 2 works as intended. When the regularization is too strong ($\beta = 10^{-2}$), the robust boundary becomes smoothed out and very similar to the standard training boundaries. The traditional adversarial training method GD-PGD learned boundaries that separated most perturbation boxes. However, the boundaries cut through the box at approximately $(1, -1)$ when β is 10^{-3}, 10^{-6}, or 10^{-9}. This behavior is likely caused by SGD backprop's the worse convergence due to non-convexity. When β is too large, the GD-PGD boundary also becomes smoothed out.

3) Additional experiments on the CIFAR-10 dataset: In this section, we repeat the experiments on the CIFAR-10 dataset with different numbers of sampled D_i matrices. Compared with Table I, which used $\epsilon = 10$, $\beta = 10^{-4}$, and $P_s = 36$, these additional experiments keep the same ϵ and β settings but reduce P_s to 24 and 18. For fair comparisons, , we set the neural network width m equal to $2P_s$ for backprop implementations in all experiments. Each experiment was repeated 7 times and the results are shown in Table II.

Table II shows that the effect on the neural network width on the prediction performance is not significant for all methods, but Alg 1 and Alg 2 are affected more: when P_s is 36 or 24, Alg 2 outperforms GD-FGSM and GD-PGD, but when P_s is 18, Alg 2 achieves worse FGSM and PGD accuracies. Our explanation is that the constraints in the convex training formulations become more restrictive when P_s is small, worsening the suboptimalities of the solutions. Therefore, Alg 1 and Alg 2 are more suitable for neural nets that are not too narrow.

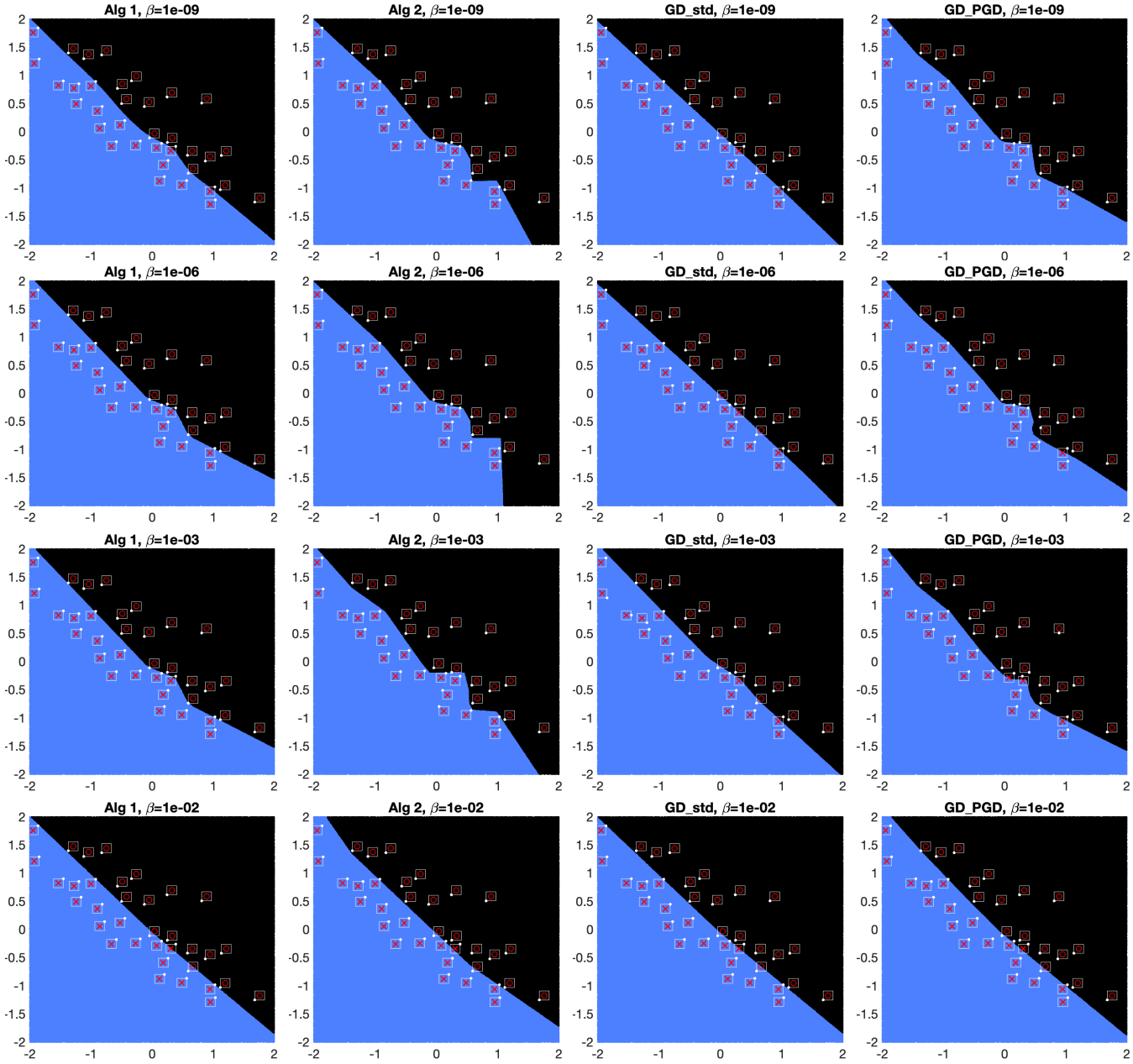

Fig. 5: Decision boundaries obtained from various methods with β set to 10^{-9}, 10^{-6}, 10^{-3}, and 10^{-2}

B. *Proof of Theorem 2*

We start by recasting the constraint of (6) as $\max_{\|u\|_2\leq 1} |v^\top (Xu)_+| \leq \beta$, and obtain

$$\max_{\|u\|_2\leq 1} \left|v^\top (Xu)_+\right| = \max_{\|u\|_2\leq 1} \left|v^\top \operatorname{diag}([Xu \geq 0])Xu\right| = \max_{i\in[P]} \left(\max_{\substack{\|u\|_2\leq 1 \\ (2D_i - I_n)Xu\geq 0}} \left|v^\top D_i Xu\right| \right),$$

where the last equality holds by the definition of the D_i matrices: $D_1 \ldots, D_P$ are all distinct matrices that can be formed by $\operatorname{diag}([Xu \geq 0])$ for some $u \in \mathbb{R}^d$. The constraint $(2D_i - I_n)Xu \geq 0$ is equivalent to $D_i Xu \geq 0$ and $(I_n - D_i)Xu \leq 0$, which forces $D_i = \operatorname{diag}([Xu \geq 0])$ to hold.

Therefore, (6) can be recast as

$$\max_{v} -\ell^*(v) \qquad \text{s. t.} \quad \max_{\substack{\|u\|_2\leq 1 \\ (2D_i - I_n)Xu\geq 0}} \left|v^\top D_i Xu\right| \leq \beta, \ \forall i \in [P]. \tag{19}$$

To form a tractable convex program that provides an approximation to (19), one can independently sample a subset of the diagonal matrices. One possible sampling procedure is presented in Alg 1. The sampled matrices, denoted as $D_1, \ldots, D_{P_s}$,

TABLE II: Optimal objective, CPU time, and prediction accuracy on clean and adversarial data with different neural network widths on the CIFAR-10 dataset.

$P_s = 24$ AND $m = 48$				
METHOD	CLEAN	FGSM ADV.	PGD ADV.	OBJECTIVE
GD-STD	81.40 %	54.72 %	54.66 %	.1486
GD-FGSM	77.75 %	64.53 %	64.46 %	.7038
GD-PGD	76.49 %	64.70 %	64.64 %	.7363
ALG 1	80.51 %	.2500 %	.1357 %	.007516
ALG 2	78.54 %	66.91 %	66.75 %	.7123

$P_s = 18$ AND $m = 36$				
METHOD	CLEAN	FGSM ADV.	PGD ADV.	OBJECTIVE
GD-STD	81.04 %	54.86 %	54.82 %	.1550
GD-FGSM	77.29 %	64.69 %	64.56 %	.7131
GD-PGD	76.44 %	64.76 %	64.74 %	.7365
ALG 1	79.71 %	.3571 %	.2714 %	.008953
ALG 2	78.71 %	63.89 %	63.67 %	.8049

can be used to construct the relaxed problem:

$$d_{s1}^\star = \max_v -\ell^*(v) \qquad \text{s.t.} \max_{\substack{\|u\|_2\le 1 \\ (2D-I_n)Xu\ge 0}} \left|v^\top D_i X u\right| \le \beta,\ \forall i \in [P_s]. \tag{20}$$

The optimization problem (20) is convex with respect to v. [21] has shown that (19) has the same optimal objective as its dual problem (5). By following precisely the same derivation, it can be shown that (20) has the same optimal objective as (8) and $p_{s1}^\star = d_{s1}^\star$. Moreover, if an additional diagonal matrix D_{P_s+1} is independently randomly sampled to form (9), then we also have $p_{s2}^\star = d_{s2}^\star$, where

$$d_{s2}^\star = \max_v -\ell^*(v) \qquad \text{s.t.} \max_{\substack{\|u\|_2\le 1 \\ (2D-I_n)Xu\ge 0}} \left|v^\top D_i X u\right| \le \beta,\ \forall i \in [P_s+1].$$

Thus, the level of suboptimality of (20) compared with (19) is the level of suboptimality of (8) compared with (5). It can be concluded from [23, Theorem 1] that if $P_s \ge \frac{n+1}{\psi\xi} - 1$, then with probability no smaller than $1-\xi$, the solution $v^\star$ to the sampled convex problem (20) satisfies

$$\mathbb{P}\Big\{D \in \mathcal{D} \ : \max_{\substack{\|u\|_2\le 1 \\ (2D-I_n)Xu\ge 0}} \left|v^{\star\top} D X u\right| > \beta\Big\} \le \psi.$$

where $\mathcal{D}$ denotes the set of all diagonal matrices that can be formed by $\mathrm{diag}([Xu \ge 0])$ for some $u \in \mathbb{R}^d$, which is the set formed by $D_1, \ldots, D_P$.

Since D_{P_s+1} is randomly sampled from $\mathcal{D}$, we have

$$\mathbb{P}\Big\{D \in \mathcal{D} \ : \max_{\substack{\|u\|_2\le 1 \\ (2D-I_n)Xu\ge 0}} \left|v^{\star\top} D X u\right| > \beta\Big\} = \mathbb{P}\Big\{\max_{\substack{\|u\|_2\le 1 \\ (2D_{P_s+1}-I_n)Xu\ge 0}} \left|v^{\star\top} D_{P_s+1} X u\right| > \beta\Big\}$$

Thus, with probability no smaller than $1-\xi$,

$$\mathbb{P}\Big\{\max_{\substack{\|u\|_2\le 1 \\ (2D_{P_s+1}-I_n)Xu\ge 0}} \left|v^{\star\top} D_{P_s+1} X u\right| > \beta\Big\} \le \psi.$$

Moreover, $d_{s2}^\star < d_{s1}^\star$ if and only if $\left|v^{\star\top} D_{P_s+1} X u\right| > \beta$ with $d_{s2}^\star = d_{s1}^\star$ otherwise. The proof is completed by noting that $p_{s1}^\star = d_{s1}^\star$ and $p_{s2}^\star = d_{s2}^\star$. ■

C. Proof of Theorem 3

Before proceeding with the proof, we first present the following result borrowed from [21].

Lemma 6. *For a given data matrix X and $(v_i, w_i)_{i=1}^P$, if $(2D_i - I_n)Xv_i \ge 0$ and $(2D_i - I_n)Xw_i \ge 0$ for all $i \in [P]$, then we can recover the corresponding neural network weights $(u_{v,w_j}, \alpha_{v,w_j})_{j=1}^{m^\star}$ using the formulas in (7), and*

$$\ell\Big(\sum_{i=1}^P D_i X(v_i - w_i), y\Big) + \beta\sum_{i=1}^P \left(\|v_i\|_2 + \|w_i\|_2\right) = \ell\Big(\sum_{j=1}^{m^\star}(Xu_{v,w_j})_+\alpha_{v,w_j}, y\Big) + \frac{\beta}{2}\sum_{j=1}^{m^\star}\left(\|u_{v,w_j}\|_2^2 + \alpha_{v,w_j}^2\right).$$

Theorem 1 implies that (4) has the same objective value as the following finite-dimensional convex optimization problem:

$$\begin{aligned} q^\star = \min_{(v_i,w_i)_{i=1}^P} \ell\Big(\sum_{i=1}^P D_i X(v_i - w_i), y\Big) + \beta \sum_{i=1}^P \big(\|v_i\|_2 + \|w_i\|_2\big) \\ \text{s.t.} \quad (2D_i - I_n)Xv_i \geq 0,\ (2D_i - I_n)Xw_i \geq 0,\ \ \forall i \in [P] \end{aligned} \tag{21}$$

where $D_1, \ldots, D_P$ are all elements in $\mathcal{D}$, defined as the set of all distinct diagonal matrices $\operatorname{diag}([Xu \geq 0])$ that can be obtained for all possible $u \in \mathbb{R}^d$. We recall that the optimal network weights can be recovered using (7).

Consider the optimization problem

$$\begin{aligned} \widetilde{q}^\star = \min_{(v_i,w_i)_{i=1}^{\widetilde{P}}} \ell\Big(\sum_{i=1}^{\widetilde{P}} D_i X(v_i - w_i), y\Big) + \beta \sum_{i=1}^{\widetilde{P}} \big(\|v_i\|_2 + \|w_i\|_2\big) \\ \text{s.t.} \quad (2D_i - I_n)Xv_i \geq 0,\ (2D_i - I_n)Xw_i \geq 0,\ \ \forall i \in [\widetilde{P}] \end{aligned} \tag{22}$$

where additional D matrices, denoted as $D_{P+1}, \ldots, D_{\widetilde{P}}$, are introduced. These additional matrices are still diagonal with each entry being either 0 or 1, while they do not belong to $\mathcal{D}$. They represent "infeasible hyperplanes" that cannot be achieved by the sign pattern of Xu for any $u \in \mathbb{R}^d$.

Lemma 7. *It holds that $\widetilde{q}^\star = q^\star$, meaning that the optimization problem (22) has the same optimal objective as (21).*

The proof of Lemma 7 is given in Section VIII-G.

The robust minimax training problem (2) considers an uncertain data matrix $X + \Delta$. Different values of $X + \Delta$ within the uncertainty set $\mathcal{U}$ can result in different D matrices. Now, we define $\widehat{\mathcal{D}} = \bigcup_\Delta \mathcal{D}_\Delta$, where $\mathcal{D}_\Delta$ is the set of diagonal matrices for a particular Δ such that $X + \Delta \in \mathcal{U}$. By construction, we have $\mathcal{D}_\Delta \subseteq \widehat{\mathcal{D}}$ for every Δ such that $X + \Delta \in \mathcal{U}$. Thus, if we define $D_1, \ldots, D_{\widehat{P}}$ as all matrices in $\widehat{\mathcal{D}}$, then for every Δ with the property $X + \Delta \in \mathcal{U}$, the optimization problem

$$\begin{aligned} \min_{(v_i,w_i)_{i=1}^{\widehat{P}}} \ell\Big(\sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_i - w_i), y\Big) + \beta \sum_{i=1}^{\widehat{P}} (\|v_i\|_2 + \|w_i\|_2) \\ \text{s.t.} \quad (2D_i - I_n)(X + \Delta)v_i \geq 0,\ (2D_i - I_n)(X + \Delta)w_i \geq 0,\ \ \forall i \in [\widehat{P}] \end{aligned} \tag{23}$$

is equivalent to

$$\min_{(u_j,\alpha_j)_{j=1}^m} \ell\Big(\sum_{j=1}^m \big((X + \Delta)u_j\big)_+ \alpha_j, y\Big) + \frac{\beta}{2} \sum_{j=1}^m \big(\|u_j\|_2^2 + \alpha_j^2\big)$$

as long as $m \geq \widehat{m}^\star$ with $\widehat{m}^\star = |\{i : v_i^\star(\Delta) \neq 0\}| + |\{i : w_i^\star(\Delta) \neq 0\}|$, where $(v_i^\star(\Delta), w_i^\star(\Delta))_{i=1}^{\widehat{P}}$ denotes an optimal point to (23). Since this equivalence holds for all Δ, it holds for $\Delta_{v,w}^\star$, the optimizer of

$$\max_{\Delta : X + \Delta \in \mathcal{U}} \ell\Big(\sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_i - w_i), y\Big).$$

Due to the reason discussed in Section IV, we apply robust optimization techniques ([24] Section 4.4.2) and replace the constraints in (23) with robust convex constraints that covers all allowed perturbed input data. Then, substituting $\Delta_{v,w}^\star$ into the objective of (23) leads to (10). Let $\big((v_{\text{rob}_i}^\star, w_{\text{rob}_i}^\star)_{i=1}^{\widehat{P}}, \Delta_{\text{rob}}^\star\big)$ denote an optimal point of (10) and let $(u_{\text{rob}_j}^\star, \alpha_{\text{rob}_j}^\star)_{j=1}^{\widehat{m}^\star}$ be the neural network weights recovered from $(v_{\text{rob}_i}^\star, w_{\text{rob}_i}^\star)_{i=1}^{\widehat{P}}$ with (7), where $\widehat{m}^\star$ is the number of nonzero weights. In light of Lemma 6, since the constraints $(2D_i - I_n)(X + \Delta)v_{\text{rob}_i}^\star \geq 0$ and $(2D_i - I_n)(X + \Delta)w_{\text{rob}_i}^\star \geq 0$ for all $i \in [\widehat{P}]$ apply to all $X + \Delta \in \mathcal{U}$, all $X + \Delta \in \mathcal{U}$ satisfy the equality

$$\begin{aligned} &\ell\Big(\sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_{\text{rob}_i}^\star - w_{\text{rob}_i}^\star), y\Big) + \beta \sum_{i=1}^{\widehat{P}} \big(\|v_{\text{rob}_i}^\star\|_2 + \|w_{\text{rob}_i}^\star\|_2\big) \\ &\qquad = \ell\Big(\sum_{j=1}^{\widehat{m}^\star} \big((X + \Delta)u_{\text{rob}_j}^\star\big)_+ \alpha_{\text{rob}_j}^\star, y\Big) + \frac{\beta}{2} \sum_{j=1}^{\widehat{m}^\star} \big(\|u_{\text{rob}_j}^\star\|_2^2 + \alpha_{\text{rob}_j}^{\star 2}\big). \end{aligned}$$

Thus, since

$$\Delta_{\text{rob}}^\star = \operatorname*{arg\,max}_{\Delta : X + \Delta \in \mathcal{U}} \ell\Big(\sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_{\text{rob}_i}^\star - w_{\text{rob}_i}^\star), y\Big) + \beta \sum_{i=1}^{\widehat{P}} \big(\|v_{\text{rob}_i}^\star\|_2 + \|w_{\text{rob}_i}^\star\|_2\big),$$

we have

$$\Delta^\star_{\text{rob}} = \underset{\Delta: X+\Delta\in\mathcal{U}}{\arg\max}\ \ell\Big(\sum_{j=1}^{\widehat{m}^\star}\big((X+\Delta)u^\star_{\text{rob}_j}\big)_+\alpha^\star_{\text{rob}_j}, y\Big) + \frac{\beta}{2}\sum_{j=1}^{\widehat{m}^\star}\big(\|u^\star_{\text{rob}_j}\|_2^2 + \alpha^{\star 2}_{\text{rob}_j}\big),$$

giving rise:

$$\begin{aligned}
&\ell\Big(\sum_{i=1}^{\widehat{P}} D_i(X+\Delta^\star_{\text{rob}})(v^\star_{\text{rob}_i} - w^\star_{\text{rob}_i}), y\Big) + \beta\sum_{i=1}^{\widehat{P}}\big(\|v^\star_{\text{rob}_i}\|_2 + \|w^\star_{\text{rob}_i}\|_2\big)\\
&\qquad = \ell\Big(\sum_{j=1}^{\widehat{m}^\star}\big((X+\Delta^\star_{\text{rob}})u^\star_{\text{rob}_j}\big)_+\alpha^\star_{\text{rob}_j}, y\Big) + \frac{\beta}{2}\sum_{j=1}^{\widehat{m}^\star}\big(\|u^\star_{\text{rob}_j}\|_2^2 + \alpha^{\star 2}_{\text{rob}_j}\big)\\
&\qquad = \max_{\Delta: X+\Delta\in\mathcal{U}}\ell\Big(\sum_{j=1}^{\widehat{m}^\star}\big((X+\Delta)u^\star_{\text{rob}_j}\big)_+\alpha^\star_{\text{rob}_j}, y\Big) + \frac{\beta}{2}\sum_{j=1}^{\widehat{m}^\star}\big(\|u^\star_{\text{rob}_j}\|_2^2 + \alpha^{\star 2}_{\text{rob}_j}\big)\\
&\qquad \geq \min_{(u_j,\alpha_j)_{j=1}^{\widehat{m}^\star}}\Big(\max_{\Delta: X+\Delta\in\mathcal{U}}\ell\Big(\sum_{j=1}^{\widehat{m}^\star}\big((X+\Delta)u_j\big)_+\alpha_j, y\Big) + \frac{\beta}{2}\sum_{j=1}^{\widehat{m}^\star}\big(\|u_j\|_2^2 + \alpha_j^2\big).\Big)
\end{aligned}$$

Therefore, (10) is an upper bound to (2).

D. Proof of (11)

Define $E_i = 2D_i - I_n$ for all $i \in [\widehat{P}]$. Note that each E_i is a diagonal matrix, and its diagonal elements are either -1 or 1. Therefore, for each $i \in [\widehat{P}]$, we can analyze the robust constraint $\min_{\Delta: X+\Delta\in\mathcal{U}} E_i(X+\Delta)v_i \geq 0$ element-wise (for each data point). Let e_{ik} denote the k^{th} diagonal element of E_i and $\delta_{ik}^\top$ denote the k^{th} element of Δ that appears in the i^{th} constraint. We then have:

$$\Big(\min_{\|\delta_{ik}\|_\infty\leq\epsilon} e_{ik}(x_k^\top + \delta_{ik}^\top)v_i\Big) = \Big(e_{ik}x_k^\top v_i + \min_{\|\delta_{ik}\|_\infty\leq\epsilon} e_{ik}\delta_{ik}^\top v_i\Big) \geq 0 \tag{24}$$

The minima of the above optimization problems are achieved at $\delta^{\star\star}_{ik} = \epsilon\cdot\text{sgn}(e_{ik}v_i) = \epsilon\cdot e_{ik}\cdot\text{sgn}(v_i)$.

Note that as ϵ approaches 0, $\delta^{\star\star}_{ik}$ and $\Delta^\star_{\text{rob}}$ in Theorem 3 both approach 0, which means that the gap between the convex robust problem (15) and the non-convex adversarial training problem (13) diminishes. Plugging $\delta^{\star\star}_k$ into (24) yields that

$$\Big(e_{ik}x_k^\top v_i - \epsilon\|e_{ik}v_i\|_1\Big) = \Big(e_{ik}x_k^\top v_i - \epsilon\|v_i\|_1\Big) \geq 0.$$

Vertically concatenating $e_{ik}x_k^\top v_i - \epsilon\|v_i\|_1 \geq 0$ for all $i \in [\widehat{P}]$ gives the vectorized representation $E_iXv_i - \epsilon\|v_i\|_1 \geq 0$, which leads to (11). Since the constraints on w are exactly the same, we also have that $\min_{\Delta: X+\Delta\in\mathcal{U}} E_i(X+\Delta)w_i \geq 0$ is equivalent to $E_iXw_i - \epsilon\|w_i\|_1 \geq 0$ for every $i \in [\widehat{P}]$.

E. Proof of Theorem 4

The regularization term is independent from Δ. Thus, it can be ignored for the purpose of analyzing the inner maximization. Note that each D_i is diagonal, and its diagonal elements are either 0 or 1. Therefore, the inner maximization of (5) can be analyzed element-wise (cost of each data point).

The maximization problem of the loss at each data point is:

$$\max_{\|\delta_k\|_\infty\leq\epsilon}\Big(1 - y_k\sum_{i=1}^{P} d_{ik}(x_k^\top + \delta_k^\top)(v_i - w_i)\Big)_+ \tag{25}$$

where d_{ik} is the k^{th} diagonal element of D_i and $\delta_k^\top$ is the k^{th} row of Δ. One can write:

$$\begin{aligned}
&\max_{\|\delta_k\|_\infty\leq\epsilon}\Big(1 - y_k\sum_{i=1}^{P} d_{ik}(x_k^\top + \delta_k^\top)(v_i - w_i)\Big)_+\\
&\qquad = \Big(\max_{\|\delta_k\|_\infty\leq\epsilon} 1 - y_k\sum_{i=1}^{P} d_{ik}(x_k^\top + \delta_k^\top)(v_i - w_i)\Big)_+\\
&\qquad = \Big(1 - y_k\sum_{i=1}^{P} d_{ik}x_k^\top(v_i - w_i) - \min_{\|\delta_k\|_\infty\leq\epsilon}\delta_k^\top y_k\sum_{i=1}^{P} d_{ik}(v_i - w_i)\Big)_+.
\end{aligned}$$

The optimal solution to $\min_{\|\delta_k\|_\infty \le \epsilon} \delta_k^\top y_k \sum_{i=1}^P d_{ik}(v_i - w_i)$ is $\delta^\star_{\text{hinge}_k} = -\epsilon \cdot \operatorname{sgn}\left(y_k \sum_{i=1}^P d_{ik}(v_i - w_i)^\top\right)$, or equivalently, $\Delta^\star_{\text{hinge}} = -\epsilon \cdot \operatorname{sgn}\left(\sum_{i=1}^P D_i y (v_i - w_i)^\top\right)$.

By substituting $\delta^\star_{\text{hinge}_k}$ into (25), the optimization (25) reduces to:

$$\left(1 - y_k \sum_{i=1}^P d_{ik} x_k^\top (v_i - w_i) + \epsilon \left\| y_k \sum_{i=1}^P d_{ik}(v_i - w_i)\right\|_1\right)_+ = \left(1 - y_k \sum_{i=1}^P d_{ik} x_k^\top (v_i - w_i) + \epsilon |y_k| \left\| \sum_{i=1}^P d_{ik}(v_i - w_i)\right\|_1\right)_+.$$

Therefore, the overall loss function is:

$$\frac{1}{n}\sum_{k=1}^n \left(1 - y_k \sum_{i=1}^P d_{ik} x_k^\top (v_i - w_i) + \epsilon |y_k| \left\| \sum_{i=1}^P d_{ik}(v_i - w_i)\right\|_1\right)_+.$$

In the case of binary classification, $y = \{-1, 1\}^n$, and thus $|y_k| = 1$ for all $k \in [n]$. Therefore, the above is equivalent to

$$\frac{1}{n}\sum_{k=1}^n \left(1 - y_k \sum_{i=1}^P d_{ik} x_k^\top (v_i - w_i) + \epsilon \left\| \sum_{i=1}^P d_{ik}(v_i - w_i)\right\|_1\right)_+ \tag{26}$$

which is the objective of (15). This completes the proof.

F. Proof of Theorem 5

We first exploit the structure of (17) and reformulate it as the following robust second-order cone program (SOCP):

$$\begin{aligned}
&\min_{(v_i, w_i, b_i, c_i)_{i=1}^{\widehat{P}}, a} \; a + \beta \sum_{i=1}^{\widehat{P}} (b_i + c_i) \\
&\text{s.t. } (2D_i - I_n) X v_i \ge \epsilon \|v_i\|_1, \;\; (2D_i - I_n) X w_i \ge \epsilon \|w_i\|_1, \;\; \|v_i\|_2 \le b_i, \;\; \|w_i\|_2 \le c_i, \;\; \forall i \in [\widehat{P}] \\
&\max_{\Delta : X + \Delta \in \mathcal{X}} \left\| \begin{bmatrix} \sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_i - w_i) - y \\ 2a - \frac{1}{4} \end{bmatrix} \right\|_2 \le 2a + \tfrac{1}{4}, \qquad \forall i \in [\widehat{P}].
\end{aligned} \tag{27}$$

Then, we need to establish the equivalence between (27) and (18). To this end, we consider the constraints of (27) and argue that these can be recast as the constraints given in (18). One can write:

$$\begin{aligned}
&\max_{\Delta : X + \Delta \in \mathcal{X}} \left\| \begin{bmatrix} \sum_{i=1}^{\widehat{P}} D_i (X + \Delta)(v_i - w_i) - y \\ 2a - \frac{1}{4} \end{bmatrix} \right\|_2 \le 2a + \frac{1}{4} \\
\iff &\max_{\|\delta_k\|_\infty \le \epsilon, \; \forall k \in [n]} \left\| \begin{bmatrix} \sum_{i=1}^{\widehat{P}} d_{i1}(x_1^\top - \delta_1^\top)(v_i - w_i) - y_1 \\ \sum_{i=1}^{\widehat{P}} d_{i2}(x_2^\top - \delta_2^\top)(v_i - w_i) - y_2 \\ \vdots \\ \sum_{i=1}^{\widehat{P}} d_{in}(x_n^\top - \delta_n^\top)(v_i - w_i) - y_n \\ 2a - \frac{1}{4} \end{bmatrix} \right\|_2 \le 2a + \frac{1}{4} \\
\iff &\max_{\|\delta_k\|_\infty \le \epsilon, \; \forall k \in [n]} \left(\sum_{k=1}^n \left(\sum_{i=1}^{\widehat{P}} d_{ik}(x_k^\top - \delta_k^\top)(v_i - w_i) - y_k \right)^2 + \left(2a - \frac{1}{4}\right)^2 \right)^{\frac{1}{2}} \le 2a + \frac{1}{4}
\end{aligned}$$

where d_{ik} is the k^{th} diagonal element of D_i and $\delta_k^\top$ is the k^{th} row of Δ. The above constraints can be rewritten by introducing slack variables $z \in \mathbb{R}^{n+1}$ as

$$\begin{aligned}
&z_k \ge \left| \sum_{i=1}^{\widehat{P}} d_{ik} x_k^\top (v_i - w_i) - y_k \right| + \epsilon \left\| \sum_{i=1}^{\widehat{P}} d_{ik}(v_i - w_i) \right\|_1, \;\; \forall k \in [n] \\
&z_{n+1} \ge \left| 2a - \tfrac{1}{4} \right|, \quad \|z\|_2 \le 2a + \tfrac{1}{4}.
\end{aligned}$$

■

G. *Proof of Lemma 7*

According to [21], recovering the neural network weights by plugging (7) in (21) leads to

$$\begin{aligned} q^\star &= \min_{(v_i,w_i)_{i=1}^P} \ell\left(\sum_{i=1}^P D_i X(v_i - w_i), y\right) + \beta \sum_{i=1}^P \left(\|v_i\|_2 + \|w_i\|_2\right) \\ &= \min_{(u_j,\alpha_j)_{j=1}^{m^\star}} \ell\left(\sum_{j=1}^{m^\star} (Xu_j)_+ \alpha_j, y\right) + \frac{\beta}{2} \sum_{j=1}^{m^\star} \left(\|u_j\|_2^2 + \alpha_j^2\right) \end{aligned}$$

Similarly, we can recover the neural network weights from the solution $(\widetilde{v}_i^\star, \widetilde{w}_i^\star)_{i=1}^{\widetilde{P}}$ of (22) using:

$$(\widetilde{u}_{j_{1i}}, \widetilde{\alpha}_{j_{1i}}) = \left(\frac{\widetilde{v}_i^\star}{\sqrt{\|\widetilde{v}_i^\star\|_2}}, \sqrt{\|\widetilde{v}_i^\star\|_2}\right), \quad (\widetilde{u}_{j_{2i}}, \widetilde{\alpha}_{j_{2i}}) = \left(\frac{\widetilde{w}_i^\star}{\sqrt{\|\widetilde{w}_i^\star\|_2}}, -\sqrt{\|\widetilde{w}_i^\star\|_2}\right), \quad \forall i \in [\widetilde{P}]. \tag{28}$$

Unlike (7), zero weights are not discarded in (28). For simplicity, we use $\widetilde{u}_1, \ldots, \widetilde{u}_{\widetilde{m}^\star}$ to refer to the hidden layer weights and use $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_{\widetilde{m}^\star}$ to refer to the output layer weights recovered using (28). Since $(\widetilde{v}_i^\star, \widetilde{w}_i^\star)_{i=1}^{\widetilde{P}}$ is a solution to (22), it satisfies $(2D_i - I_n)X\widetilde{v}_i^\star \geq 0$ and $(2D_i - I_n)X\widetilde{w}_i^\star \geq 0$ for all $i \in [\widetilde{P}]$. Thus, we can apply Lemma 6 to obtain:

$$\begin{aligned} \widetilde{q}^\star =& \ell\left(\sum_{i=1}^{\widetilde{P}} D_i X(\widetilde{v}_i^\star - \widetilde{w}_i^\star), y\right) + \beta \sum_{i=1}^{\widetilde{P}} \left(\|\widetilde{v}_i^\star\|_2 + \|\widetilde{w}_i^\star\|_2\right) \\ =& \ell\left(\sum_{j=1}^{\widetilde{m}^\star} (X\widetilde{u}_j^\star)_+ \alpha_j, y\right) + \frac{\beta}{2} \sum_{j=1}^{\widetilde{m}^\star} \left(\|\widetilde{u}_j^\star\|_2^2 + \widetilde{\alpha}_j^{\star 2}\right) \\ \geq & \min_{(u_j,\alpha_j)_{j=1}^{\widetilde{m}^\star}} \ell\left(\sum_{j=1}^{\widetilde{m}^\star} (Xu_j)_+ \alpha_j, y\right) + \frac{\beta}{2} \sum_{j=1}^{\widetilde{m}^\star} \left(\|u_j\|_2^2 + \alpha_j^2\right) \end{aligned}$$

Since $\widetilde{P} \geq P$, $m^\star \leq 2P$ and $\widetilde{m}^\star = 2\widetilde{P}$, we have $\widetilde{m}^\star \geq m^\star$. Therefore, according to Section 2 and Theorem 6 of [21], we have

$$\begin{aligned} q^\star &= \min_{(u_j,\alpha_j)_{j=1}^{m^\star}} \ell\left(\sum_{j=1}^{m^\star} (Xu_j)_+ \alpha_j, y\right) + \frac{\beta}{2} \sum_{j=1}^{m^\star} \left(\|u_j\|_2^2 + \alpha_j^2\right) \\ &= \min_{(u_j,\alpha_j)_{j=1}^{\widetilde{m}^\star}} \ell\left(\sum_{j=1}^{\widetilde{m}^\star} (Xu_j)_+ \alpha_j, y\right) + \frac{\beta}{2} \sum_{j=1}^{\widetilde{m}^\star} \left(\|u_j\|_2^2 + \alpha_j^2\right) \\ &\leq \widetilde{q}^\star \end{aligned}$$

The above inequality shows that a neural network with more than m neurons in the hidden layer will yield the same loss as the neural network with m neurons when optimized.

Note that (22) can always attain $q^\star$ by simply plugging in the optimal solution of (21) and assigning 0 to all other additional v_i and w_i, implying that $q^\star \geq \widetilde{q}^\star$. Since $q^\star$ is both an upper bound and a lower bound on $\widetilde{q}^\star$, we have $\widetilde{q}^\star = q^\star$, proving that as long as all matrices in $\mathcal{D}$ are included, the existence of redundant matrices does not change the optimal objective value.