On the Role of Generalization in Transferability of Adversarial Examples (Supplementary Material)

Yilin Wang[1] **Farzan Farnia**[1]

[1] Department of Computer Science and Engineering,
The Chinese University of Hong Kong,
Hong Kong SAR

A PROOFS

A.1 PROOF OF PROPOSITION 1

To prove the proposition, note that the optimization problem for λ-optimal adversarial attack scheme can be written as

$$\max_{\delta:\mathcal{X}\times\mathcal{Y}\rightarrow\mathbb{R}^d} \; \mathbb{E}\Big[\ell\big(h_{\mathbf{w}}(\mathbf{X}+\delta(\mathbf{X},Y)),Y\big) - \frac{\lambda}{2}\|\delta(\mathbf{X},Y)\|^2\Big].$$

We observe that the above optimization problem decouples into separate problems for every $(\mathbf{x}, y)$, and hence $\delta^*(\mathbf{x}, y)$ is the optimal solution to the following optimization problem

$$\max_{\boldsymbol{\delta}\in\mathbb{R}^d} \; \ell\big(h_{\mathbf{w}}(\mathbf{x}+\boldsymbol{\delta}),y\big) - \frac{\lambda}{2}\|\boldsymbol{\delta}\|^2.$$

Since $\ell\circ h_{\mathbf{w}}$ is assumed to be λ-smooth in $\mathbf{x}$, the objective function in the above optimization problem is a concave function of $\boldsymbol{\delta}$. This is because the Hessian of the above objective function will be negative semi-definite as

$$\nabla^2_{\boldsymbol{\delta}}\Big[\ell\big(h_{\mathbf{w}}(\mathbf{x}+\boldsymbol{\delta}),y\big) - \frac{\lambda}{2}\|\boldsymbol{\delta}\|^2\Big] = \nabla^2_{\boldsymbol{\delta}}\ell\big(h_{\mathbf{w}}(\mathbf{x}+\boldsymbol{\delta}),y\big) - \lambda I_{d\times d} \preceq \mathbf{0}. \tag{1}$$

Therefore, applying the first-order necessary condition implies that a globally optimal solution $\delta^*(\mathbf{x}, y)$ to the above concave objective function will be the solution to

$$\nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x}+\delta^*(\mathbf{x},y)),y\big) - \lambda\delta^*(\mathbf{x},y) = \mathbf{0}.$$

The above necessary and sufficient condition for $\delta^*(\mathbf{x}, y)$ can be rewritten as:

$$\big(\mathbf{x}+\delta^*(\mathbf{x},y)\big) - \frac{1}{\lambda}\nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x}+\delta^*(\mathbf{x},y)),y\big) = \mathbf{x}.$$

Note that this condition is equivalent to the following equation which completes the proof:

$$\delta^*(\mathbf{x},y) \;=\; \Big(\mathrm{Id}_{\mathbf{x}} - \frac{1}{\lambda}\nabla_{\mathbf{x}}\ell\circ h_{\mathbf{w}}\Big)^{-1}(\mathbf{x}) \;-\; \mathbf{x}.$$

A.2 PROOF OF THEOREM 1

Assumption 1. *Loss function $\ell(y, y')$ is a c-bounded, 1-Lipschitz, and 1-smooth function of the input y, i.e. for every $y_1, y_2, y' \in \mathcal{Y}$ we have $|\ell(y_1, y')| \le c$, $|\ell(y_1, y')-\ell(y_2, y')| \le \|y_1-y_2\|_2$, and $\|\nabla_y\ell(y_1, y')-\nabla_y\ell(y_2, y')\|_2 \le \|y_1-y_2\|_2$.*

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Assumption 2. *The set of substitute DNNs in the black-box attack scheme $\mathcal{H}_{\mathcal{W}} = \{h_{\mathbf{w}} : \mathbf{w} \in \mathcal{W}\}$ contains L-layer neural networks $h_{\mathbf{w}}(\mathbf{x}) = W_L\phi_L\big(W_{L-1}\phi_{L-1}(\cdots W_1\phi_1(W_0\mathbf{x})\cdot)\big)$. We suppose that the dimensions of matrices $W_0, \ldots, W_k$ is bounded by D, and assume every activation ϕ_i satisfies $\phi_i(0) = 0$ and is γ_i-Lipschitz and γ_i-smooth, i.e. $\max\{|\phi_i'(z)|, |\phi_i''(z)|\} \le \gamma_i$ holds for every $z \in \mathbb{R}$.*

Assumption 3. *The class of target classifiers $\mathcal{F}_{\mathcal{V}} = \{f_{\mathbf{v}} : \mathbf{v} \in \mathcal{V}\}$ consists of K-layer neural network functions $f_{\mathbf{v}}(\mathbf{x}) = V_K\psi_L\big(V_{L-1}\psi_{L-1}(\cdots V_1\psi_1(V_0\mathbf{x})\cdot)\big)$ with activation function ψ_i's. We suppose that the dimensions of matrices $V_0, \ldots, V_k$ is bounded by D. Also, we assume every ψ_i satisfies $\psi_i(0) = 0$ and is ξ_i-Lipschitz, i.e. $\max_z\, |\psi_i'(z)| \le \xi_i$. Also, we define the capacity $R_{\mathcal{V}}$ as*

$$R_{\mathcal{V}} := \sup_{\mathbf{v}\in\mathcal{V}} \left\{ \Big(\prod_{i=0}^{K} \xi_i \|V_i\|_2\Big) \Big(\sum_{i=0}^{K} \frac{\|V_i^\top\|_{2,1}^{2/3}}{\|V_i\|_2^{2/3}}\Big)^{3/2} \right\}.$$

To show Theorem 1, we first present the following lemmas.

Lemma 1 ([Farnia et al., 2018], Lemma 7)**.** *Under Assumptions 1, 2, the substitute neural network's loss function $\ell(h_{\mathbf{w}}(\mathbf{x}), y)$ is κ-smooth in input vector $\mathbf{x}$, i.e its gradient with respect to $\mathbf{x}$ is κ-Lipschitz and satisfies*

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X},\, y \in \mathcal{Y}: \quad \big\|\nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x}), y\big) - \nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x}'), y\big)\big\| \le \kappa\|\mathbf{x} - \mathbf{x}'\|,$$

where $\kappa = \Big(\sum_{i=0}^{L}\prod_{j=0}^{i}\gamma_j\|W_j\|\Big)\Big(\prod_{i=0}^{L}\gamma_i\|W_i\|\Big)$.

Lemma 2. *Under Theorem 1's assumptions, the λ-optimal attack scheme $\delta^*_{\mathbf{w}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ satisfies the following output norm constraint for every $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$:*

$$\big\|\delta^*_{\mathbf{w}}(\mathbf{x}, y)\big\| \le \frac{\prod_{i=0}^{L}\gamma_i\|W_i\|_2}{\lambda} = \frac{L_{\mathbf{w}}}{\lambda}.$$

Proof. Note that since $\lambda > \Big(\sum_{i=0}^{L}\prod_{j=0}^{i}\gamma_j\|W_j\|_2\Big)\prod_{i=0}^{L}\gamma_i\|W_i\|_2$ holds according to Theorem 1's assumptions, the smoothness condition of Proposition **??** will hold according to Lemma 1. As a result, we have

$$\delta^*_{\mathbf{w}}(\mathbf{x}, y) = \frac{1}{\lambda}\nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\mathbf{w}}(\mathbf{x}, y)), y\big).$$

Therefore,

$$\big\|\delta^*_{\mathbf{w}}(\mathbf{x}, y)\big\| \le \frac{\|\nabla_{\mathbf{x}}\ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\mathbf{w}}(\mathbf{x}, y)), y\big)\|}{\lambda} \le \frac{\prod_{i=0}^{L}\gamma_i\|W_i\|_2}{\lambda}.$$

The final inequality follows from the Lipschitz coefficient of DNN function $h_{\mathbf{w}}$ which is the composition of linear transformation W_i's (with Lipschitz constant $\|W_i\|_2$) and activation non-linearity ϕ_i's (with Lipschitz constant γ_i). Therefore, the proof is complete. ☐

Lemma 3. *Under Assumption 2, the substitute neural network $h_{\mathbf{w}}$'s gradient satisfies the following error bound under a perturbation matrix Δ_k with L_2-operator norm $\|\Delta_k\|_2 \le t$ to wight matrix W_k, where we define $\widetilde{\mathbf{w}} = \mathrm{vec}(W_0, \ldots, W_{k-1}, W_k + \Delta_k, W_{k+1}, \ldots, W_L)$:*

$$\big\|\nabla_{\mathbf{x}}h_{\mathbf{w}}(\mathbf{x}) - \nabla_{\mathbf{x}}h_{\widetilde{\mathbf{w}}}(\mathbf{x})\big\| \le \frac{L_{\mathbf{w}}\sum_{i=k}^{L}\prod_{j=0}^{i}\gamma_j\|W_j\|}{\|W_k\|_2}\|\Delta_k\|_2$$

Proof. Note that the neural network's Jacobian with respect to the input follows from:

$$\mathrm{J}_{h_{\mathbf{w}}}(\mathbf{x}) = \prod_{i=0}^{L} W_i^\top \,\mathrm{diag}\big(\phi_i'(h_{\mathbf{w_{0:i}}}(\mathbf{x}))\big).$$

In the above, $h_{\mathbf{w_{0:i}}}(\mathbf{x})$ denotes the neural net's output at layer i. Therefore, for $\widetilde{\mathbf{w}}$ which is different from $\mathbf{w}$ only at layer k we will have:

$$\big\|\mathrm{J}_{h_{\mathbf{w}}}(\mathbf{x}) - \mathrm{J}_{h_{\widetilde{\mathbf{w}}}}(\mathbf{x})\big\|_2 \le \sum_{i=k}^{L}\Big[\Big(\prod_{j=0}^{L}\gamma_j\|W_j\|_2\Big)\Big(\prod_{j=0}^{i}\gamma_j\|W_j\|_2\Big)\Big]\frac{\|\Delta_k\|_2}{\|W_k\|_2}$$

$$= \left(\prod_{j=0}^{L} \gamma_j \|W_j\|_2\right) \sum_{i=k}^{L} \left[\prod_{j=0}^{i} \gamma_j \|W_j\|_2\right] \frac{\|\Delta_k\|_2}{\|W_k\|_2}$$
$$= \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\|W_k\|_2} \|\Delta_k\|_2.$$

The proof is hence finished. □

Lemma 4. *Under Theorem 1's assumptions, the λ-optimal attack scheme $\delta^*_{\mathbf{w}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ satisfies the following error bound under a norm-bounded perturbation $\Delta_k : \|\Delta_k\|_2 \le t$: to wight matrix W_k where we define $\widetilde{\mathbf{w}} = \mathrm{vec}(W_0, \dots, W_{k-1}, W_k + \Delta_j, W_{k+1}, \dots, W_L)$:*

$$\left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| \le \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\tau\lambda \|W_k\|_2} \|\Delta_k\|_2$$

Proof. Since $\lambda > \left(\sum_{i=0}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|_2\right) \prod_{i=0}^{L} \gamma_i \|W_i\|_2$ follows from Theorem 1's assumption, Proposition **??** will hold according to Lemma 1 and implies that

$$\delta^*_{\mathbf{w}}(\mathbf{x}, y) = \frac{1}{\lambda} \nabla_{\mathbf{x}} \ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\mathbf{w}}(\mathbf{x}, y)), y\big).$$

As a result,

$$\begin{aligned}
&\left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| \\
&= \frac{1}{\lambda} \left\| \nabla_{\mathbf{x}} \ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\mathbf{w}}(\mathbf{x}, y)), y\big) - \nabla_{\mathbf{x}} \ell\big(h_{\widetilde{\mathbf{w}}}(\mathbf{x} + \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)), y\big) \right\| \\
&\le \frac{1}{\lambda} \left\| \nabla_{\mathbf{x}} \ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\mathbf{w}}(\mathbf{x}, y)), y\big) - \nabla_{\mathbf{x}} \ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)), y\big) \right\| \\
&\quad + \frac{1}{\lambda} \left\| \nabla_{\mathbf{x}} \ell\big(h_{\mathbf{w}}(\mathbf{x} + \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)), y\big) - \nabla_{\mathbf{x}} \ell\big(h_{\widetilde{\mathbf{w}}}(\mathbf{x} + \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)), y\big) \right\| \\
&\overset{(a)}{\le} \frac{\mathrm{Lip}(\nabla_{\mathbf{x}} \ell \circ h_{\mathbf{w}})}{\lambda} \left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| + \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\lambda \|W_k\|_2} \|\Delta_k\|_2 \\
&\overset{(b)}{\le} \frac{L_{\mathbf{w}} \sum_{i=0}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\lambda} \left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| + \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\lambda \|W_k\|_2} \|\Delta_k\|_2 \\
&\overset{(c)}{\le} (1 - \tau) \left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| + \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\lambda \|W_k\|_2} \|\Delta_k\|_2
\end{aligned}$$

Here, (a) follows from the definition of Lipschitz constant and Lemma 3's weight perturbation bound. (b) comes from a direct application of Lemma 1, and (c) follows from Theorem 1' assumption. Therefore, the above inequalities collectively lead to the following bound, which completes the proof:

$$\tau \left\|\delta^*_{\mathbf{w}}(\mathbf{x}, y) - \delta^*_{\widetilde{\mathbf{w}}}(\mathbf{x}, y)\right\| \le \frac{L_{\mathbf{w}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\lambda \|W_k\|_2} \|\Delta_k\|_2.$$

□

To prove Theorem 1, we follow a covering-number-based approach similar to the generalization analysis in [Bartlett et al., 2017] for the standard non-adversarial deep supervised learning problem. To do this, we consider the norm constraints $\|W_i\|_2 \le a'_i$, $\|W_i\|_{2,1} \le b'_i$ for every $i = 0, \dots, L$, and $\|V_j\|_2 \le a_i$, $\|V_j\|_{2,1} \le b_i$ for every $j = 0, \dots, K$. Now, we define the following covering resolution parameters for the classifier and substitute DNNs' different layers:

$$\epsilon'_k = \frac{\tau\lambda a'_k \alpha'_k \epsilon}{2(\prod_{i=0}^{K} \xi_i a_i)(\prod_{i=0}^{L} \gamma_i a'_i)(\sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j a'_j)}, \text{ where } \alpha'_k = \frac{1}{A'} \frac{{b'_k}^{2/3}}{{a'_k}^{2/3}}, \; A' = \sum_{i=0}^{L} \frac{{b'_i}^{2/3}}{{a'_i}^{2/3}}$$

$$\epsilon_j = \frac{a_j \alpha_j \epsilon}{2\prod_{i=j}^{K} \gamma_i a_i}, \text{ where } \alpha_j = \frac{1}{A}\frac{b_j^{2/3}}{a_j^{2/3}},\ A = \sum_{i=0}^{K} \frac{b_i^{2/3}}{a_i^{2/3}}.$$

Note that Lemma 4 implies that by finding an ϵ_k'-covering for each W_k and ϵ_j-covering for each V_j, the covering resolution for $\mathcal{F} \circ \Delta_H|_S$ will be upper-bounded by

$$\sum_{k=0}^{L}\Big[\frac{L_{\mathbf{w}} L_{\mathbf{v}} \sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_j \|W_j\|}{\tau\lambda \|W_k\|_2}\epsilon_k'\Big] + \sum_{k=0}^{K}\Big[\frac{\prod_{i=k}^{K} \gamma_i \|V_i\|_2}{\|V_k\|_2}\epsilon_k\Big] = \epsilon.$$

Therefore, by applying Lemma A.7 from [Bartlett et al., 2017] we will have the following bound on the ϵ-covering-number for the set $\mathcal{F} \circ \Delta_H|_S = \{\ell(f_{\mathbf{v}}(\mathbf{X} + \delta_{\mathbf{w}}^*(\mathbf{X}, Y)) :\ \forall 0 \le i \le K : \|V_i\|_2 \le a_i,\ \|V_i\|_{2,1} \le b_i,\ \forall 0 \le j \le L :\ \|W_j\|_2 \le a_j',\ \|W_j\|_{2,1} \le b_j'\}$

$$\begin{aligned}
&\log \mathcal{N}\big(\mathcal{F} \circ \Delta_H|_S, \|\cdot\|_2, \epsilon\big) \\
&\le \sum_{i=0}^{L} \sup_{\mathbf{w}_{-i},\mathbf{v} \in \mathcal{W},\mathcal{V}} \Big[\log \mathcal{N}\big(\big\{\delta_{\mathbf{w}}^*(\mathbf{X}, Y) :\ \|\mathbf{W}_i\|_2 \le a_i',\ \|\mathbf{W}_i\|_{2,1} \le b_i'\big\}, \|\cdot\|_2, \epsilon_i'\big)\Big] \\
&\quad + \sum_{i=0}^{K} \sup_{\mathbf{w},\mathbf{v}_{-i} \in \mathcal{W},\mathcal{V}} \Big[\log \mathcal{N}\big(\big\{h_{\mathbf{v}_{0:i}}(\mathbf{X}), Y) :\ \|\mathbf{V}_i\|_2 \le a_i,\ \|\mathbf{V}_i\|_{2,1} \le b_i\big\}, \|\cdot\|_2, \epsilon_i\big)\Big] \\
&\le \sum_{i=0}^{L} \sup_{\mathbf{w}_{-i},\mathbf{v} \in \mathcal{W},\mathcal{V}} \Big[\log \mathcal{N}\big(\big\{\delta_{\mathbf{w}}^*(\mathbf{X}, Y) :\ \|\mathbf{W}_i\|_{2,1} \le b_i'\big\}, \|\cdot\|_2, \epsilon_i'\big)\Big] \\
&\quad + \sum_{i=0}^{K} \sup_{\mathbf{w},\mathbf{v}_{-i} \in \mathcal{W},\mathcal{V}} \Big[\log \mathcal{N}\big(\big\{h_{\mathbf{v}_{0:i}}(\mathbf{X}), Y) :\ \|\mathbf{V}_i\|_{2,1} \le b_i\big\}, \|\cdot\|_2, \epsilon_i\big)\Big] \\
&\le \sum_{i=0}^{L} \Big[\sup_{\mathbf{w}_{-i},\mathbf{v} \in \mathcal{W},\mathcal{V}} \frac{{b_i'}^2 \|\delta_{\mathbf{w}}^*(\mathbf{X}, Y)\|_2^2}{{\epsilon_i'}^2} \log(2W^2)\Big] \\
&\quad + \sum_{i=0}^{L} \Big[\sup_{\mathbf{w},\mathbf{v}_{-i} \in \mathcal{W},\mathcal{V}} \frac{b_i^2 \|h_{\mathbf{v}_{0:i}}(\mathbf{X})\|_2^2}{\epsilon_i^2} \log(2W^2)\Big] \\
&\le \sum_{i=0}^{K} \big[\sup_{\mathbf{w}_{-i},\mathbf{v} \in \mathcal{W},\mathcal{V}} \frac{L_{\mathbf{w}}^2 \log(2W^2)}{\lambda^2 \epsilon^2} \frac{{b_i'}^2}{{\epsilon_i'}^2}\big] \\
&\quad + \sum_{i=0}^{L} \Big[\frac{4b_i^2 (B + \frac{\prod_{i=0}^{L} \gamma_i a_i'}{\lambda})^2 \prod_{i=0}^{K} \xi_i^2 a_i^2}{\epsilon^2} \sum_{i=0}^{K} \frac{b_i^2}{\alpha_i^2 a_i^2}\Big] \\
&\le \frac{\log(2W^2) \prod_{i=0}^{L} \gamma_i^2 {a_i'}^2}{\lambda^2 \epsilon^2} \sum_{k=0}^{K} \big[\frac{{b_i'}^2 (\sum_{i=k}^{L} \prod_{j=0}^{i} \gamma_i a_i)^2}{{\alpha_i'}^2 {a_i'}^2}\big] \\
&\quad + \sum_{i=0}^{L} \Big[\frac{4b_i^2 (B + \frac{\prod_{i=0}^{L} \gamma_i a_i'}{\lambda})^2 \prod_{i=0}^{K} \xi_i^2 a_i^2}{\epsilon^2} \sum_{i=0}^{K} \frac{b_i^2}{\alpha_i^2 a_i^2}\Big] \\
&\le \frac{\log(2W^2) \prod_{i=0}^{L} \gamma_i^2 {a_i'}^2 (\sum_{i=0}^{L} \prod_{j=0}^{i} \gamma_i a_i)^2}{\lambda^2 \epsilon^2} \sum_{k=0}^{K} \big[\frac{{b_k'}^2}{{\alpha_k'}^2 {a_k'}^2}\big] \\
&\quad + \sum_{i=0}^{L} \Big[\frac{4b_i^2 (B + \frac{\prod_{i=0}^{L} \gamma_i a_i'}{\lambda})^2 \prod_{j=0}^{K} \xi_j^2 a_j^2}{\epsilon^2} \sum_{k=0}^{K} \frac{b_k^2}{\alpha_k^2 a_k^2}\Big] \\
&\le \frac{4\log(2W^2) \prod_{i=0}^{L} \gamma_i^2 {a_i'}^2 \big(\sum_{i=0}^{L} \prod_{j=0}^{i} \gamma_j a_j\big)^2}{\lambda^2 \epsilon^2} \sum_{i=0}^{L} \big[\frac{{b_i'}^2}{{\alpha_i'}^2 {a_i'}^2}\big] \\
&\quad + \frac{4\log(2W^2)(B + \frac{\prod_{i=0}^{L} \gamma_i a_i'}{\lambda})^2 \prod_{i=0}^{K} \xi_i^2 a_i^2}{\epsilon^2} \sum_{i=0}^{K} \big[\frac{b_i^2}{\alpha_i^2 a_i^2}\big]
\end{aligned}$$

$$= \frac{C}{\epsilon^2}$$

where we define

$$C := 4\log(2W^2)\Big[\frac{\prod_{i=0}^{L}\gamma_i^2 {a'_i}^2\big(\sum_{i=0}^{L}\prod_{j=0}^{i}\gamma_j a_j\big)^2}{\lambda^2}\left[\sum_{i=0}^{L}\frac{{b'_i}^{2/3}}{{a'_i}^{2/3}}\right]^3 + \big(B+\frac{\prod_{i=0}^{L}\gamma_i a'_i}{\lambda}\big)^2\prod_{i=0}^{K}\xi_i^2 a_i^2\left[\sum_{i=0}^{K}\frac{b_i^{2/3}}{a_i^{2/3}}\right]^3\Big].$$

Now, based on the above covering-number bound, we use the Dudley entropy integral bound [Bartlett et al., 2017] which bounds the empirical Rademacher complexity of $\mathcal{F}\circ\Delta_{\mathcal{H}}|_S$ as

$$\begin{aligned}\mathcal{R}(\mathcal{F}\circ\Delta_{\mathcal{H}}|_S) &\le \inf_{\alpha\ge 0}\left\{\frac{4\alpha}{\sqrt{n}}+\frac{12}{n}\int_{\alpha}^{\sqrt{n}}\sqrt{\log\mathcal{N}\big(\mathcal{F}\circ\Delta_H|_S,\|\cdot\|_2,\epsilon\big)}d\epsilon\right\}\\ &\le \inf_{\alpha\ge 0}\left\{\frac{4\alpha}{\sqrt{n}}+\frac{12\sqrt{C}}{n}\log(\frac{\sqrt{n}}{\alpha})\right\}\\ &\le \frac{4}{n^{3/2}}+\frac{18\log(n)\sqrt{C}}{n}\end{aligned}$$

where the last line follows from choosing $\alpha = 1/n$. Also, note that since for every non-negative constants $a,b\ge 0$ we have $\sqrt{a+b}\le\sqrt{a}+\sqrt{b}$, then

$$\sqrt{C} \le 2\sqrt{\log(2D^2)}\Big[\frac{\prod_{i=0}^{L}\gamma_i a'_i\big(\sum_{i=0}^{L}\prod_{j=0}^{i}\gamma_j a_j\big)}{\lambda}\left[\sum_{i=0}^{L}\frac{{b'_i}^{2/3}}{{a'_i}^{2/3}}\right]^{3/2} + \big(B+\frac{\prod_{i=0}^{L}\gamma_i a'_i}{\lambda}\big)\prod_{i=0}^{K}\xi_i a_i\left[\sum_{i=0}^{K}\frac{b_i^{2/3}}{a_i^{2/3}}\right]^{3/2}\Big].$$

Consequently, we have the following bound where $R_{\mathcal{V}}$ and $R_{\mathcal{W}}$ are defined as in Theorem 1:

$$\mathcal{R}(\mathcal{F}\circ\Delta_{\mathcal{H}}|_S)\le\mathcal{O}\Big(\frac{(B+\frac{L_{\mathbf{w}}}{\lambda})\big(R_{\mathcal{V}}+\frac{1}{\tau^2}L_{\mathbf{w}}R_{\mathcal{W}}\big)\log(n)}{n}\log(D)\Big)$$

Therefore, according to standard Rademacher complexity-based generalization bounds [Bartlett and Mendelson, 2002], for every $\omega>0$ with probability at least $1-\omega$ we have for every $\mathbf{v},\mathbf{w}\in\mathcal{V},\mathcal{W}$:

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{x}_i+\delta^*_{\mathbf{w}}(\mathbf{x}_i,y_i)),y_i\big)\Big]-\mathbb{E}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{X}+\delta^*_{\mathbf{w}}(\mathbf{X},Y)),Y\big)\Big]\\ &\le\mathcal{O}\Big(c\sqrt{\frac{\log(1/\omega)}{n}}+\frac{(B+\frac{L_{\mathbf{w}}}{\lambda})\big(R_{\mathcal{V}}+\frac{1}{\tau^2}L_{\mathbf{w}}R_{\mathbf{w}}\big)\log(n)}{n}\log(D)\Big),\end{aligned}$$

which implies that

$$\begin{aligned}\epsilon_{\text{gen}}(\delta^*_{\mathbf{w}}) &= \min_{\mathbf{v}\in\mathcal{V}}\Big\{\frac{1}{n}\sum_{i=1}^{n}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{x}_i+\delta^*_{\mathbf{w}}(\mathbf{x}_i,y_i)),y_i\big)\Big]\Big\}-\min_{\mathbf{v}\in\mathcal{V}}\Big\{\mathbb{E}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{X}+\delta^*_{\mathbf{w}}(\mathbf{X},Y)),Y\big)\Big]\Big\}\\ &\le\max_{\mathbf{v}\in\mathcal{V}}\Big\{\frac{1}{n}\sum_{i=1}^{n}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{x}_i+\delta^*_{\mathbf{w}}(\mathbf{x}_i,y_i)),y_i\big)\Big]-\mathbb{E}\Big[\ell\big(f_{\mathbf{v}}(\mathbf{X}+\delta^*_{\mathbf{w}}(\mathbf{X},Y)),Y\big)\Big]\Big\}\\ &\le\mathcal{O}\Big(c\sqrt{\frac{\log(1/\omega)}{n}}+\frac{(B+\frac{L_{\mathbf{w}}}{\lambda})\big(R_{\mathcal{V}}+\frac{1}{\tau^2}L_{\mathbf{w}}R_{\mathbf{w}}\big)\log(n)}{n}\log(D)\Big).\end{aligned}$$

Therefore, the theorem's proof is complete.

B ADDITIONAL NUMERICAL EXPERIMENTS

In this section, we report the complete set of our numerical results. Table 1 shows the complete results for DNNs regularized by early stopping, i.e. the complete version of Table **??**. Moreover, Tables 2,3,4 are the full versions of Table **??**, demonstrating the relationship between generalization errors and transferability rates of the discussed datasets and DNN architectures. In addition to our earlier results, the accuracies for adversarially-perturbed training and test samples are also included in the tables.

In addition to the previous results, we also present the transferability rates averaged over test samples whose adversarial examples are predicted correctly by both the regularized and unregularized substitute DNNs. The transferability rates over those test samples intersecting the correctly predicted samples by regularized and unregularized substitute DNNs are shown in Tables 5 and 6 under the title Transferability Rate-Int. Our numerical results suggest that over the test samples correctly predicted by both regularzied and unregularized DNNs, a better generalization score again results in higher transferability rates for designed adversarial examples.

Last, Tables 7 and 8 show the numerical results when substitute model and target are trained with the same training set.

Table 1: Generalization and transferability rates for different DNN architectures and image datasets with and without early stopping (ES)

Dataset	Model	Method	Train Acc	Test Acc	Gen.Err.	Transferability Rate(VGG16)	Transferability Rate(ResNet18)
Cifar10	Inception	PGM	0.970	0.453	0.517	0.127	0.104
		PGM-ES	0.591	0.518	0.073	0.198	0.172
		FGM	0.997	0.529	0.467	0.100	0.089
		FGM-ES	0.657	0.530	0.126	0.170	0.147
	Alexnet	PGM	1.000	0.421	0.579	0.098	0.077
		PGM-ES	0.548	0.487	0.061	0.154	0.136
		FGM	1.000	0.480	0.520	0.100	0.087
		FGM-ES	0.594	0.501	0.092	0.152	0.127
Cifar100	Inception	PGM	0.877	0.231	0.646	0.283	0.258
		PGM-ES	0.408	0.271	0.137	0.330	0.286
		FGM	0.984	0.272	0.711	0.270	0.239
		FGM-ES	0.457	0.312	0.146	0.327	0.289
	Alexnet	PGM	0.966	0.202	0.764	0.252	0.227
		PGM-ES	0.338	0.248	0.091	0.294	0.266
		FGM	0.990	0.234	0.756	0.261	0.232
		FGM-ES	0.399	0.278	0.122	0.291	0.259
SVHN	Inception	PGM	0.925	0.585	0.341	0.207	0.220
		PGM-ES	0.711	0.654	0.057	0.298	0.322
		FGM	0.998	0.619	0.380	0.136	0.129
		FGM-ES	0.898	0.718	0.180	0.213	0.219
	Alexnet	PGM	0.848	0.541	0.307	0.211	0.228
		PGM-ES	0.624	0.594	0.030	0.256	0.278
		FGM	0.949	0.576	0.373	0.157	0.170
		FGM-ES	0.691	0.627	0.064	0.241	0.260

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

Farzan Farnia, Jesse M Zhang, and David Tse. Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457*, 2018.

Table 2: Generalization and transferability rates on CIFAR-10 data, with and without using spectral regularization

Dataset	Model	Method	β	Train Acc	Test Acc	Gen.Err.	Transferability Rate(VGG16)	Transferability Rate(ResNet18)
Cifar10	Inception	FGM(L_2)	∞	0.998	0.532	0.466	0.092	0.078
			1	0.830	0.511	0.320	0.136	0.113
			1.3	0.936	0.479	0.456	0.124	0.097
			1.6	0.997	0.509	0.488	0.099	0.084
			2	0.999	0.541	0.458	0.087	0.073
			3	1.000	0.554	0.445	0.090	0.077
		PGM(L_2)	∞	0.951	0.443	0.508	0.104	0.084
			1	0.759	0.502	0.258	0.150	0.134
			1.3	0.913	0.459	0.454	0.125	0.099
			1.6	0.945	0.447	0.499	0.115	0.095
			2	0.979	0.451	0.529	0.110	0.092
			3	0.988	0.468	0.520	0.103	0.082
		FGM(L_∞)	∞	0.946	0.442	0.504	0.124	0.101
			1	0.694	0.451	0.243	0.191	0.153
			1.3	0.842	0.421	0.421	0.165	0.129
			1.6	0.951	0.419	0.531	0.129	0.106
			2	0.980	0.453	0.527	0.121	0.103
			3	0.988	0.470	0.517	0.112	0.091
		PGM(L_∞)	∞	0.708	0.525	0.184	0.419	0.393
			1	0.627	0.545	0.082	0.497	0.474
			1.3	0.674	0.534	0.140	0.466	0.443
			1.6	0.710	0.520	0.190	0.442	0.422
			2	0.739	0.505	0.234	0.408	0.389
			3	0.745	0.504	0.241	0.398	0.380
	Alexnet	FGM(L_2)	∞	1.000	0.495	0.505	0.089	0.070
			1	0.977	0.526	0.451	0.147	0.122
			1.3	1.000	0.496	0.504	0.132	0.105
			1.6	1.000	0.474	0.526	0.115	0.090
			2	1.000	0.477	0.523	0.108	0.088
			3	1.000	0.496	0.504	0.098	0.079
		PGM(L_2)	∞	0.999	0.454	0.545	0.105	0.087
			1	0.861	0.519	0.342	0.162	0.139
		FGM(L_∞)	∞	0.996	0.400	0.596	0.102	0.089
			1	0.914	0.426	0.487	0.199	0.169
			1.3	0.998	0.387	0.610	0.169	0.142
			1.6	1.000	0.368	0.632	0.146	0.120
			2	1.000	0.420	0.580	0.128	0.106
			3	1.000	0.435	0.565	0.115	0.096
		PGM(L_∞)	∞	0.685	0.474	0.211	0.446	0.423
			1	0.628	0.473	0.156	0.472	0.457
			1.3	0.680	0.436	0.245	0.441	0.423
			1.6	0.697	0.420	0.277	0.421	0.403
			2	0.679	0.408	0.271	0.404	0.388
			3	0.628	0.439	0.189	0.426	0.405

Table 3: Generalization and transferability rates on CIFAR-100 dataset, with and without spectral regularization.

Dataset	Model	Method	β	Train Acc	Test Acc	Gen.Err.	Transferability Rate(VGG16)	Transferability Rate(ResNet18)
Cifar100	Inception	FGM(L_2)	∞	0.996	0.279	0.717	0.268	0.236
			1	0.761	0.277	0.484	0.250	0.290
			1.3	0.853	0.294	0.558	0.313	0.275
			1.6	0.998	0.260	0.738	0.236	0.263
			2	1.000	0.284	0.715	0.238	0.272
			3	0.999	0.262	0.736	0.221	0.257
		PGM(L_2)	∞	0.857	0.255	0.602	0.303	0.270
			1.3	0.750	0.256	0.494	0.330	0.301
	Alexnet	FGM(L_2)	∞	1.000	0.242	0.758	0.258	0.232
			1	0.925	0.315	0.611	0.342	0.310
			1.3	1.000	0.289	0.710	0.304	0.266
			1.6	1.000	0.285	0.715	0.291	0.248
			2	1.000	0.284	0.716	0.288	0.253
			3	1.000	0.269	0.730	0.265	0.240
		PGM(L_2)	∞	1.000	0.210	0.789	0.229	0.260
			1	0.889	0.288	0.601	0.323	0.353

Table 4: Generalization and transferability rates on SVHN dataset, with and without spectral regularization.

Dataset	Model	Method	β	Train Acc	Test Acc	Gen.Err.	Transferability Rate(VGG16)	Transferability Rate(ResNet18)
SVHN	Inception	FGM(L_2)	∞	0.991	0.618	0.373	0.134	0.126
			1	0.848	0.645	0.203	0.277	0.257
			1.3	0.967	0.605	0.362	0.223	0.209
			1.6	0.998	0.573	0.426	0.202	0.187
			2	1.000	0.571	0.429	0.158	0.149
			3	1.000	0.642	0.358	0.121	0.118
		PGM(L_2)	∞	0.934	0.592	0.342	0.193	0.177
			1	0.767	0.652	0.115	0.339	0.313
	Alexnet	FGM(L_2)	∞	0.991	0.618	0.373	0.134	0.126
			1	0.848	0.645	0.203	0.277	0.257
			1.3	0.967	0.605	0.362	0.223	0.209
			1.6	0.998	0.573	0.426	0.202	0.187
			2	1.000	0.571	0.429	0.158	0.149
			3	0.999	0.581	0.418	0.145	0.133
		PGM(L_2)	∞	0.844	0.546	0.298	0.211	0.225
			1	0.817	0.618	0.199	0.276	0.292

Table 5: Generalization and transferability rates for different DNN architectures and image datasets with and without spectral regularization. Transferability Rate-Int. means averaged transferability rate on adversarial examples correctly labeled by both the regularized and unregularized DNNs.

Dataset	Model	Method	β	Gen.Err.	Transferability Rate-Int(VGG16)	Transferability Rate-Int(ResNet18)
Cifar10	Inception	FGM(L_2)	∞	0.466	0.032	0.026
			1	0.320	0.077	0.058
		PGM(L_2)	∞	0.508	0.030	0.026
			1	0.258	0.063	0.052
		FGM(L_∞)	∞	0.504	0.029	0.028
			1	0.243	0.070	0.090
		PGM(L_∞)	∞	0.184	0.136	0.181
			1	0.082	0.182	0.162
	Alexnet	FGM(L_2)	∞	0.505	0.035	0.029
			1	0.451	0.076	0.068
		PGM(L_2)	∞	0.545	0.037	0.030
			1	0.342	0.077	0.063
		FGM(L_∞)	∞	0.596	0.039	0.019
			1	0.487	0.089	0.070
		PGM(L_∞)	∞	0.211	0.227	0.222
			1	0.156	0.271	0.248
Cifar100	Inception	FGM(L_2)	∞	0.717	0.126	0.112
			1.3	0.558	0.154	0.131
		PGM(L_2)	∞	0.602	0.141	0.123
			1.3	0.494	0.160	0.141
	Alexnet	FGM(L_2)	∞	0.758	0.127	0.101
			1	0.611	0.180	0.159
		PGM(L_2)	∞	0.789	0.114	0.108
			1	0.601	0.159	0.151
SVHN	Inception	FGM(L_2)	∞	0.373	0.010	0.013
			1	0.203	0.103	0.125
		PGM(L_2)	∞	0.342	0.024	0.031
			1	0.115	0.102	0.125
	Alexnet	FGM(L_2)	∞	0.373	0.013	0.014
			1	0.203	0.092	0.112
		PGM(L_2)	∞	0.298	0.044	0.053
			1	0.199	0.080	0.101

Table 6: Generalization and transferability rates for different DNN architectures and image datasets with and without early stopping (ES), Transferability Rate-Int. means averaged transferability rate on adversarial examples correctly labeled by both the regularized and unregularized DNNs.

Dataset	Model	Method	Gen.Err.	Transferability Rate-Int.(VGG16)	Transferability Rate-Int.(ResNet18)
Cifar10	Inception	PGM	0.517	0.030	0.127
		PGM-ES	0.073	0.063	0.198
		FGM	0.467	0.032	0.100
		FGM-ES	0.126	0.077	0.170
	Alexnet	PGM	0.579	0.037	0.098
		PGM-ES	0.061	0.077	0.154
		FGM	0.520	0.035	0.100
		FGM-ES	0.092	0.076	0.152
Cifar100	Inception	PGM	0.646	0.141	0.283
		PGM-ES	0.137	0.160	0.330
		FGM	0.711	0.126	0.270
		FGM-ES	0.146	0.154	0.327
	Alexnet	PGM	0.764	0.114	0.252
		PGM-ES	0.091	0.159	0.294
		FGM	0.756	0.127	0.261
		FGM-ES	0.122	0.180	0.291
SVHN	Inception	PGM	0.341	0.024	0.207
		PGM-ES	0.057	0.102	0.298
		FGM	0.380	0.010	0.136
		FGM-ES	0.180	0.103	0.213
	Alexnet	PGM	0.307	0.044	0.211
		PGM-ES	0.030	0.080	0.256
		FGM	0.373	0.013	0.157
		FGM-ES	0.064	0.092	0.241

Table 7: Generalization error and transferability rates when the target and substitute models are trained using the same training set, with and without spectral normalization.

Dataset	Model	Method	β	Train Acc	Test Acc	Gen. Err.	Transferability Rate (VGG16)	Transferability Rate (R18)
Cifar10	Inception	FGM(L_2)	∞	0.983	0.563	0.420	0.995	0.108
			1	0.801	0.565	0.236	0.176	0.174
		PGM(L_2)	∞	0.896	0.498	0.398	0.129	0.134
			1	0.757	0.544	0.213	0.183	0.183
Cifar100	Inception	FGM(L_2)	∞	0.822	0.281	0.541	0.211	0.213
			1.3	0.619	0.309	0.310	0.253	0.254
		PGM(L_2)	∞	0.669	0.292	0.377	0.253	0.254
			1.3	0.559	0.310	0.249	0.285	0.286
SVHN	Alexnet	FGM(L_2)	∞	0.901	0.629	0.272	0.160	0.164
			1	0.813	0.667	0.146	0.262	0.267
		PGM(L_2)	∞	0.792	0.578	0.214	0.234	0.235
			1	0.775	0.637	0.139	0.281	0.286

Table 8: Generalization error and transferability rates when the target and substitute models are trained using the same training set, with and without early stopping.

Dataset	Model	Attack	Train Acc	Test Acc	Gen. Err.	Transferability Rate (VGG16)	Transferability Rate (R18)
Cifar10	Inception	FGM(L_2)	0.984	0.553	0.431	0.010	0.108
		FGM(L_2)-ES	0.622	0.583	0.039	0.139	0.143
	Alexnet	FGM(L_2)	1.000	0.525	0.475	0.084	0.094
		FGM(L_2)-ES	0.667	0.548	0.119	0.132	0.137
Cifar100	Inception	FGM(L_2)	0.939	0.290	0.649	0.224	0.263
		FGM(L_2)-ES	0.429	0.340	0.088	0.261	0.315
	Alexnet	FGM(L_2)	0.912	0.248	0.664	0.246	0.271
		FGM(L_2)-ES	0.338	0.307	0.032	0.283	0.306
SVHN	Inception	FGM(L_2)	0.997	0.621	0.376	0.130	0.136
		FGM(L_2)-ES	0.883	0.667	0.216	0.226	0.224
	Alexnet	FGM(L_2)	0.948	0.578	0.370	0.171	0.160
		FGM(L_2)-ES	0.685	0.628	0.057	0.250	0.253

Table 9: Transferability rates when adversarial examples are generated by different methods and from different substitute models.

substitute DNN	regularization method of DNN	target DNN	Transferability Rate of FGM	Transferability Rate of I-FGSM	Transferability Rate of PGD
PGD-trained Inception	spectral normalization	ERM-trained VGG16	0.134	0.148	0.151
	early stop		0.172	0.177	0.198
	None		0.092	0.097	0.106
ERM-trained Inception	spectral normalization	ERM-trained VGG16	0.071	0.070	0.062
	early stop		0.088	0.099	0.113
	None		0.030	0.044	0.024