References

Abramovich, F. and Grinshtein, V. High-dimensional classification by sparse logistic regression. June 2017. URL http://arxiv.org/abs/1706.08344.

Alvarez-Melis, D. and Jaakkola, T. S. On the robustness of interpretability methods. *arXiv preprint arXiv:1806. 08049*, 2018. URL http://arxiv.org/abs/1806.08049.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. November 2017. URL http://arxiv.org/abs/1711.06104.

Arya, V., Bellamy, R. K. E., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Vera Liao, Q., Luss, R., Mojsilović, A., Mourad, S., Pedemonte, P., Raghavendra, R., Richards, J., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K. R., Wei, D., and Zhang, Y. One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. September 2019. URL http://arxiv.org/abs/1909.03012.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11(Jun):1803–1831, 2010. URL http://www.jmlr.org/papers/volume11/baehrens10a/baehrens10a.pdf.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, pp. 387–402. Springer Berlin Heidelberg, 2013. URL http://dx.doi.org/10.1007/978-3-642-40994-3_25.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. December 2014. URL http://arxiv.org/abs/1412.6572.

Guest, O. and Love, B. C. What the success of brain imaging implies about the neural code. *Elife*, 6, January 2017. URL http://dx.doi.org/10.7554/eLife.21397.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hurley, N. and Rickard, S. Comparing measures of sparsity. *IEEE Trans. Inf. Theory*, 55(10):4723–4741, October 2009. URL http://dx.doi.org/10.1109/TIT.2009.2027527.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. May 2019. URL http://arxiv.org/abs/1905.02175.

Kim, B., Seo, J., and Jeon, T. Bridging adversarial robustness and gradient interpretability. *Safe Machine Learning workshop at ICLR*, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. June 2017.

Molnar, C. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

Noack, A., Ahern, I., Dou, D., and Li, B. Does interpretability of neural networks imply adversarial robustness? *ArXiv*, abs/1912.03430, 2019.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Berkay Celik, Z., and Swami, A. The limitations of deep learning in adversarial settings. November 2015. URL http://arxiv.org/abs/1511.07528.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. February 2016. URL http://arxiv.org/abs/1602.04938.

Sankaranarayanan, S., Jain, A., Chellappa, R., and Lim, S. N. Regularizing deep networks using efficient layerwise adversarial training. May 2017. URL http://arxiv.org/abs/1705.07819.

Shaham, U., Yamada, Y., and Negahban, S. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. November 2015. URL http://arxiv.org/abs/1511.05432.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

Sinha, A., Namkoong, H., and Duchi, J. Certifying some distributional robustness with principled adversarial training. February 2018. URL https://openreview.net/pdf?id=Hk6kPgZA-.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. March 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. December 2013. URL http://arxiv.org/abs/1312.6199.

Tan, M., Tsang, I. W., and Wang, L. Minimax sparse logistic regression for very high-dimensional feature selection. *IEEE Trans Neural Netw Learn Syst*, 24(10):1609–1622, October 2013. URL http://dx.doi.org/10.

1109/TNNLS.2013.2263427.

Tan, M., Tsang, I. W., and Wang, L. Towards ultrahigh dimensional feature selection for big data. *J. Mach. Learn. Res.*, 15(1):1371–1429, 2014. URL http://www.jmlr.org/papers/volume15/tan14a/tan14a.pdf.

Tanay, T. and Griffin, L. D. A new angle on l2 regularization. *CoRR*, abs/1806.11186, 2018.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. May 2018. URL http://arxiv.org/abs/1805.12152.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. August 2017. URL http://arxiv.org/abs/1708.07747.

Xu, H., Caramanis, C., and Mannor, S. Robustness and regularization of support vector machines. *J. Mach. Learn. Res.*, 10(Jul):1485–1510, 2009. URL http://www.jmlr.org/papers/volume10/xu09b/xu09b.pdf.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., and Ravikumar, P. K. On the (in)fidelity and sensitivity of explanations. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 10967–10978. Curran Associates, Inc., 2019.

Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. December 2017. URL http://arxiv.org/abs/1712.07107.

A. Additional Related Work

Section 7 discussed some of the work most directly related to this paper. Here we describe some additional related work.

Adversarial Robustness and Interpretability. Through a very different analysis, (Yeh et al., 2019) show a result closely related to our Theorem 5.1: the show that adversarial training is analogous to making gradient-based explanations more "smooth", which lowers the sensitivity of gradient explanation. The paper of (Noack et al., 2019) considers a question that is the converse of the one we examine in our paper: They show evidence that models that are forced to have interpretable gradients are more robust to adversarial examples than models trained in a standard manner. Another recent paper (Kim et al., 2019) analyzes the effect of adversarial training on the interpretability of neural network loss gradients.

Relation to work on Regularization Benefits of AML. There has been prior work on the *regularization benefits* of adversarial training (Xu et al., 2009; Szegedy et al., 2013; Goodfellow et al., 2014; Shaham et al., 2015; Sankaranarayanan et al., 2017; Tanay & Griffin, 2018), primarily in image-classification applications: when a model is adversarially trained, its classification accuracy on natural (i.e. un-perturbed) test data can improve. All of this prior work has focused on the *performance-improvement* (on natural test data) aspect of regularization, but none have examined the *feature-pruning* benefits explicitly. In contrast to this work, our primary interest is in the explainability benefits of adversarial training, and specifically the ability of adversarial training to significantly improve feature-concentration while maintaining (and often improving) performance on natural test data.

Adversarial Training vs Feature-Selection. Since our results show that adversarial training can effectively shrink the weights of irrelevant or weakly-relevant features (while preserving weights on relevant features), a legitimate counter-proposal might be that one could weed out such features beforehand via a pre-processing step where features with negligible label-correlations can be "removed" from the training process. Besides the fact that this scheme has no guarantees whatsoever with regard to adversarial robustness, there are some practical reasons why correlation-based feature selection is not as effective as adversarial training, in producing pruned models: (a) With adversarial training, one needs to simply try different values of the adversarial strength parameter ε and find a level where accuracy (or other metric such as AUC-ROC) is not impacted much but model-weights are significantly more concentrated; on the other hand with the correlation-based feature-pruning method, one needs to set up an iterative loop with gradually increasing correlation thresholds, and each time the input pre-processing pipeline needs to be re-executed with a reduced set of features. (b) When there are categorical features with large cardinalities, where just some of the categorical values have negligible feature-correlations, it is not even clear how one can "remove" these specific feature *values*, since the feature itself must still be used; at the very least it would require a re-encoding of the categorical feature each time a subset of its values is "dropped" (for example if a one-hot encoding or hashing scheme is used). Thus correlation-based feature-pruning is a much more cumbersome and inefficient process compared to adversarial training.

Adversarial Training vs Other Methods to Train Sparse Logistic Regression Models. (Tan et al., 2013; 2014) propose an approach to train sparse logistic regression models based on a min-max optimization problem that can be solved by the cutting plane algorithm. This requires a specially implemented custom optimization procedure. By contrast, $\ell_\infty(\varepsilon)$-adversarial training can be implemented as a simple and efficient "bolt-on" layer on top of existing ML pipelines based on TensorFlow, PyTorch or SciKit-Learn, which makes it highly practical. Another paper (Abramovich & Grinshtein, 2017) proposes a feature selection procedure based on penalized maximum likelihood with a complexity penalty on the model size, but once again this requires special-purpose optimization code.

B. Discussion of Assumptions

B.1. Loss Functions Satisfying Assumption LOSS-CVX

We show here that several popular loss functions satisfy the Assumption LOSS-CVX.

Logistic NLL (Negative Log Likelihood) Loss.
$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = -\ln(\sigma(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)) = \ln(1 + \exp(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle))$, which can be written as $g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ where $g(z) = \ln(1 + e^z)$ is a non-decreasing and convex function.

Hinge Loss
$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = (1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)^+$, which can be written as $g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ where $g(z) = (1 + z)^+$ is non-decreasing and convex.

Softplus Hinge Loss.
$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = \ln(1 + \exp(1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle))$, which can be written as $g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ where $g(z) = \ln(1 + e^{1+z})$, and clearly g is non-decreasing. Moreover the first derivative of g, $g'(z) = 1/(1 + e^{-1-z})$ is non-decreasing, and therefore g is convex.

B.2. Implications of Assumption FEAT-TRANS

Lemma 1. *Given random variables X', Y where $Y \in \{\pm 1\}$, if we define $X = X' - [\mathbb{E}(X'|Y = 1) + \mathbb{E}(X'|Y = -1)]/2$, then:*

$$\mathbb{E}(X|Y) = aY \tag{B.23}$$

$$\mathbb{E}(YX) = \mathbb{E}[\mathbb{E}(YX|Y)] = \mathbb{E}[Y\mathbb{E}(X|Y)] = \mathbb{E}[Y^2 a] = a \tag{B.24}$$

$$\mathbb{E}(YX|Y) = Y\mathbb{E}[X|Y] = Y^2 a = a, \tag{B.25}$$

where $a = [\mathbb{E}(X'|Y = 1) - \mathbb{E}(X'|Y = -1)]/2$.

Proof. Consider the function $f(Y) = \mathbb{E}(X'|Y)$, and let $b_0 := f(-1)$ and $b_1 := f(1)$. Since there are only *two* values of Y that are of interest, we can represent $f(Y)$ by a *linear* function $aY + c$, and it is trivial to verify that $a = (b_1 - b_0)/2$ and $c = (b_1 + b_0)/2$ are the unique values that are consistent with $f(-1) = b_0$ and $f(1) = b_1$. Thus if $X = X' - c$, then $\mathbb{E}(X|Y) = aY$, proving (B.23), and the other two properties follow trivially. □

C. Expressions for adversarial perturbation and loss-gradient

We show two simple preliminary results for loss functions that satisfy Assumption LOSS-INC: Lemma 2 shows a simple closed form expression for the $\ell_\infty(\varepsilon)$-adversarial perturbation, and we use this result to derive an expression for the *gradient* of the $\ell_\infty(\varepsilon)$-adversarial loss $\mathcal{L}(\boldsymbol{x} + \delta^*, y; \boldsymbol{w})$ with respect to a weight w_i (Lemma 3).

Lemma 2 (Closed form for $\ell_\infty(\varepsilon)$-adversarial perturbation)**.** *For a data point $(\boldsymbol{x}, y)$, given model weights $\boldsymbol{w}$, if the loss function $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$ satisfies Assumption LOSS-INC, the $\ell_\infty(\varepsilon)$-adversarial perturbation δ^* is given by:*

$$\delta^* = -y\,\mathrm{sgn}(\boldsymbol{w})\varepsilon, \tag{7}$$

and the corresponding $\ell_\infty(\varepsilon)$-adversarial loss is

$$\mathcal{L}(\boldsymbol{x} + \delta^*, y;\ \boldsymbol{w}) = g(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle) \tag{8}$$

Proof. Assumption LOSS-INC implies that the loss is non-increasing in $y\langle \boldsymbol{w}, x\rangle$, and therefore the $\ell_\infty(\varepsilon)$-perturbation δ^* of x that maximizes the loss would be such that, for each $i \in [d]$, x_i is changed by an amount ε in the direction of $-y\,\mathrm{sgn}(w_i)$, and the result immediately follows. □

Lemma 3 (Gradient of adversarial loss)**.** *For any loss function satisfying Assumption LOSS-INC, for a given data point $(\boldsymbol{x}, y)$, the gradient of the $\ell_\infty(\varepsilon)$-adversarial loss is given by:*

$$\frac{\partial \mathcal{L}(\boldsymbol{x} + \delta^*, y;\ \boldsymbol{w})}{\partial w_i} = -g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\ (yx_i - \mathrm{sgn}(w_i)\varepsilon) \tag{9}$$

Proof. This is straightforward by substituting the expression (7) for δ^* in $g(-y\langle \boldsymbol{w}, \boldsymbol{x} + \delta^*\rangle)$, and applying the chain rule. □

D. Expectation of SGD Weight Update

The following Lemma will be used to prove Theorem 3.1.

D.1. Upper bound on $\mathbb{E}[Zf(Z, V)]$

Lemma 4 (Upper Bound on expectation of $Zf(Z, V)$ when f is non-increasing in Z, $(Z \perp V)|Y$, and $\mathbb{E}(Z|Y) = \mathbb{E}(Z)$)**.** *For any random variables Z, V, if:*

- *$f(Z, V)$ is non-increasing in Z,*

- *Z, V are conditionally independent given a third r.v. Y, and*
- *$\mathbb{E}(Z|Y) = \mathbb{E}(Z)$,*

then

$$\mathbb{E}[Zf(Z,V)] \;\leq\; \mathbb{E}(Z)\mathbb{E}[f(Z,V)] \tag{D.26}$$

Proof. Let $\overline{z} = \mathbb{E}(Z) = \mathbb{E}(Z|Y)$ and note that

$$\begin{aligned}\mathbb{E}[Zf(Z,V)] - \mathbb{E}[Z]\mathbb{E}[f(Z,V)] &= \mathbb{E}[Zf(Z,V)] - \overline{z}\mathbb{E}[f(Z,V)] &\text{(D.27)}\\ &= \mathbb{E}[(Z-\overline{z})f(Z,V)] &\text{(D.28)}\end{aligned}$$

We want to now argue that $\mathbb{E}[(Z-\overline{z})f(\overline{z},V)] = 0$. To see this, apply the Law of Total Expectation by conditioning on Y:

$$\begin{aligned}\mathbb{E}[(Z-\overline{z})f(\overline{z},V)] &= \mathbb{E}\Big[\mathbb{E}\big[(Z-\overline{z})f(\overline{z},V)|Y\big]\Big] & & \\ &= \mathbb{E}\Big[\mathbb{E}\big[(Z-\overline{z})|Y\big]\mathbb{E}\big[f(\overline{z},V)|Y\big]\Big] & \text{(since } (Z\perp V)|Y) & \quad\text{(D.29)}\\ &= 0. & \text{(since } \mathbb{E}(Z|Y) = \mathbb{E}(Z) = \overline{z}) & \quad\text{(D.30)}\end{aligned}$$

Since $\mathbb{E}[(Z-\overline{z})f(\overline{z},V)] = 0$, we can subtract it from the last expectation in (D.28), and by linearity of expectations the RHS of (D.28) can be replaced by

$$\mathbb{E}\big[(Z-\overline{z})(f(Z,V) - f(\overline{z},V))\big]. \tag{D.31}$$

That fact that $f(Z,V)$ is non-increasing in Z implies that $(Z-\overline{z})(f(Z,V) - f(\overline{z},V)) \leq 0$ for any value of Z and V, with equality when $Z = \overline{z}$. Therefore the expectation (D.31) is bounded above by zero, which implies the desired result. □

Theorem 3.1 (Expected SGD Update in Adversarial Training)**.** *For any loss function $\mathcal{L}$ satisfying Assumption LOSS-CVX, a dataset $\mathcal{D}$ satisfying Assumption FEAT-TRANS, a subset S of features that are conditionally independent of the rest given the label y, if a data point $(\boldsymbol{x}, y)$ is randomly drawn from $\mathcal{D}$, and $\boldsymbol{x}$ is perturbed to $\boldsymbol{x}' = \boldsymbol{x} + \delta^*$, where δ^* is an $\ell_\infty(\varepsilon)$-adversarial perturbation, then during SGD using the $\ell_\infty(\varepsilon)$-adversarial loss $\mathcal{L}(\boldsymbol{x}', y; \boldsymbol{w})$, the expected weight-updates $\overline{\Delta}_i := \mathbb{E}\Delta w_i$ for $i \in S$ and the corresponding $\mathbf{w}$-weighted average $\overline{\Delta}_S^{\mathbf{w}}$ satisfy the following properties:*

1. *If $w_i = 0 \; \forall i \in S$, then for each $i \in S$,*
$$\overline{\Delta}_i = \overline{g'}\, a_i, \tag{11}$$
2. *and otherwise,*
$$\overline{\Delta}_S^{\mathbf{w}} \leq \overline{g'}(a_S^{\mathbf{w}} - \varepsilon), \tag{12}$$
and equality holds in the limit as $w_i \to 0 \; \forall i \in S$,

where $\overline{g'}$ is the expectation in (10), $a_i = \mathbb{E}(x_i y)$ is the directed strength of feature x_i from Eq. (4), and $a_S^{\mathbf{w}}$ is the corresponding $\mathbf{w}$-weighted average over S.

Proof. Consider the adversarial loss gradient expression (9) from Lemma 3. For the case where $w_i = 0$ for all $i \in S$, for any given $i \in S$, the negative expectation of the adversarial loss gradient is

$$\begin{aligned}\overline{\Delta}_i &= \mathbb{E}\big[yx_i\, g'(\varepsilon||\boldsymbol{w}||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\big] & \\ &= \mathbb{E}\Big[\mathbb{E}\big[yx_i\, g'(\varepsilon||\boldsymbol{w}||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\; |y\big]\Big] & \text{(Law of Total Expectation)}\\ &= \mathbb{E}\Big[y\,\mathbb{E}\big[x_i\, g'(\varepsilon||\boldsymbol{w}||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\; |y\big]\Big], &\end{aligned}$$

and in the last expectation above, we note that since $w_i = 0 \; \forall i \in S$, the argument of g' does not depend on x_i for any $i \in S$, and since the features in S are conditionally independent of the rest given the label y, x_i is independent of the g' term in the inner conditional expectation. Therefore the inner conditional expectation can be factored as a product of conditional expectations, which gives

$$\begin{aligned}
\overline{\Delta}_i &= \mathbb{E}\Big[y\mathbb{E}(x_i|y)\mathbb{E}\big[g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w},\, \boldsymbol{x}\rangle)\ |y\big]\Big] \\
&= \mathbb{E}\Big[y^2 a_i \mathbb{E}\big[g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w},\, \boldsymbol{x}\rangle)\ |y\big]\Big] && \text{(Assumption FEAT-TRANS, Eq B.23)} \\
&= a_i \mathbb{E}\Big[\mathbb{E}\big[g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w},\, \boldsymbol{x}\rangle)\ |y\big]\Big] && \text{(since } y = \pm 1\text{)} \\
&= a_i \overline{g'}, && \text{(D.32)}
\end{aligned}$$

which establishes the first result.

Now consider the case where $w_i \neq 0$ for at least one $i \in S$. Starting with the adversarial loss gradient expression (9) from Lemma 3, for any $i \in S$, multiplying throughout by $-\operatorname{sgn}(w_i)$ and taking expectations results in

$$\operatorname{sgn}(w_i)\overline{\Delta}_i = \mathbb{E}\Big[\big[yx_i \operatorname{sgn}(w_i) - \varepsilon\big]\ g'(\varepsilon||w||_1 - y\langle \boldsymbol{w},\, \boldsymbol{x}\rangle)\Big] \tag{D.33}$$

where the expectation is with respect to a random choice of data-point $(\boldsymbol{x}, y)$. The argument of g' can be written as

$$\varepsilon||w||_1 - y\langle \boldsymbol{w},\, \boldsymbol{x}\rangle = -\sum_{j=1}^{d} |w_j|(yx_j \operatorname{sgn}(w_j) - \varepsilon),$$

and for $j \in [d]$ if we let Z_j denote the random variable corresponding to $yx_j \operatorname{sgn}(w_j) - \varepsilon$, then (D.33) can be written as

$$\operatorname{sgn}(w_i)\overline{\Delta}_i = \mathbb{E}\left[Z_i\ g'\left(-\sum_{j=1}^{d} |w_j| Z_j\right)\right]. \tag{D.34}$$

Taking the $|w_i|$-weighted average of both sides of (D.34) over $i \in S$ yields

$$\overline{\Delta}^{\mathbf{w}}_S = \frac{1}{\sum_{i\in S}|w_i|}\mathbb{E}\left[g'\left(-\sum_{j=1}^{d}|w_j|Z_j\right)\sum_{i\in S}(|w_i|Z_i)\right]. \tag{D.35}$$

If we now define $Z_S := \sum_{i\in S}(|w_i|Z_i)$, the argument of g' in the expectation above can be written as $V_S - Z_S$ where V_S denotes the negative sum of $|w_j|Z_j$ terms over all $j \notin S$, and thus (D.35) can be written as

$$\overline{\Delta}^{\mathbf{w}}_S = \frac{1}{\sum_{i\in S}|w_i|}\mathbb{E}\left[g'\left(V_S - Z_S\right)Z_S\right]. \tag{D.36}$$

Note that Z_S is a function of Y and the features in S, and V_S is a function of Y and the features in the *complement* of S. Since the features in S are conditionally independent of the rest given the label Y (this is a condition of the Theorem), it follows that $(V_S \perp Z_S)|Y$. Since by Assumption LOSS-CVX, g' is a non-decreasing function, $g'(V_S - Z_S)$ is *non-increasing* in Z_S. Thus all three conditions of Lemma 4 are satisfied, with the random variables Z, V, Y and function f in the Lemma corresponding to random variables Z_S, V_S, Y and function g' respectively in the present Theorem. It then follows from Lemma 4 that

$$\overline{\Delta}^{\mathbf{w}}_S \leq \frac{1}{\sum_{i\in S}|w_i|}E(Z_S)\overline{g'}. \tag{D.37}$$

The definition of Z_S, and the fact that $\mathbb{E}(Z_i) = \operatorname{sgn}(w_i)\mathbb{E}(yx_i) - \varepsilon = a_i \operatorname{sgn}(w_i) - \varepsilon$ (property (4)), imply

$$\mathbb{E}(Z_S) = \sum_{i\in S}\left[|w_i|\operatorname{sgn}(w_i)a_i\right] - \varepsilon\sum_{i\in S}|w_i|,$$

and the definition of $a^{\mathbf{w}}_S$ allows us to simplify (D.37) to

$$\overline{\Delta}^{\mathbf{w}}_S \leq \overline{g'}(a^{\mathbf{w}}_S - \varepsilon),$$

which establishes the upper bound (12).

To analyze the limiting case where $w_i \to 0$ for all $i \in S$, write Eq. (D.36) as follows:

$$\overline{\Delta}_S^{\mathbf{w}} = \mathbb{E}\left[g'\left(V_S - Z_S\right)\frac{Z_S}{\sum_{i\in S}|w_i|}\right]. \tag{D.38}$$

If we let $|w_i| \to 0$ for all $i \in S$, the Z_S in the argument of g' can be set to 0, and we can write the RHS of (D.38) as

$$\overline{\Delta}_S^{\mathbf{w}} = \mathbb{E}\left[g'(V_S)\frac{Z_S}{\sum_{i\in S}|w_i|}\right] = \mathbb{E}\left[\mathbb{E}\left[g'(V_S)\frac{Z_S}{\sum_{i\in S}|w_i|}\;\middle|\; Y\right]\right], \tag{D.39}$$

where the inner conditional expectation can be factored as a product of conditional expectations since $(Z_S \perp V_S | Y)$:

$$\overline{\Delta}_S^{\mathbf{w}} = \mathbb{E}\left[\mathbb{E}\left[g'(V_S)\;\middle|\; Y\right]\mathbb{E}\left[\frac{Z_S}{\sum_{i\in S}|w_i|}\;\middle|\; Y\right]\right]. \tag{D.40}$$

Now notice that

$$\mathbb{E}(Z_S|Y) = \mathbb{E}\left[\sum_{i\in S}(|w_i|Z_i)\;\middle|\; Y\right] = \sum_{i\in S}|w_i|\,\mathbb{E}\left[\operatorname{sgn}(w_i)Yx_i - \varepsilon \mid Y\right]. \tag{D.41}$$

From Property (5) of datasets satisfying Assumption FEAT-TRANS, $\mathbb{E}[Yx_i|Y] = \mathbb{E}(Yx_i) = a_i$, and so the second inner expectation in (D.40) simplifies to a constant:

$$\mathbb{E}\left[\frac{Z_S}{\sum_{i\in S}|w_i|}\;\middle|\; Y\right] = \frac{\sum_{i\in S}\left[|w_i|\operatorname{sgn}(w_i)a_i\right]}{\sum_{i\in S}|w_i|} - \varepsilon = a_S^{\mathbf{w}} - \varepsilon. \tag{D.42}$$

Eq. (D.40) can therefore be simplified to

$$\overline{\Delta}_S^{\mathbf{w}} = \mathbb{E}\left[\mathbb{E}\left[g'(V_S)\;\middle|\; Y\right]\right](a_S^{\mathbf{w}} - \varepsilon) = \overline{g'}(a_S^{\mathbf{w}} - \varepsilon), \tag{D.43}$$

which shows the final statement of the Theorem, namely, that if $w_i \to 0$ for all $i \in S$, then (12) holds with equality. □

D.2. Implications of Theorem 3.1

Keeping in mind the interpretations of the $\mathbf{w}$-weighted average quantities $a_S^{\mathbf{w}}$ and $\overline{\Delta}_S^{\mathbf{w}}$ described in the paragraph after the statement of Theorem 3.1, we can state the following implications of this result:

If all weights of S are zero, then they grow in the correct direction. When $w_i = 0$ for all $i \in S$ (recall that S is a subset of features, conditionally independent of the rest given the label y), the expected SGD update $\overline{\Delta}_i$ for each $i \in S$ is proportional to the directed strength a_i of feature x_i, and if $\overline{g'} \neq 0$, this means that on average the SGD update causes the weight w_i to grow from zero in the *correct direction*. This is what one would expect from an SGD training procedure.

If the weights of S are mis-aligned weights on average, then they shrink at a rate proportional to $\varepsilon + |a_S^{\mathbf{w}}|$. Suppose for at least one $i \in S$, $w_i \neq 0$, and $a_S^{\mathbf{w}} < 0$, i.e. the weights of the features in S are mis-aligned on average. In this case by (12), $\overline{\Delta}_S^{\mathbf{w}} < 0$, i.e. the weights of the features in S, in aggregate (i.e. in the $|w_i|$-weighted sense) shrink toward zero in expectation. The aggregate rate of this shrinkage is proportional to $\varepsilon + |a_S^{\mathbf{w}}|$. In other words, all other factors remaining the same, adversarial training (i.e. with $\varepsilon > 0$) shrinks mis-aligned faster than natural training (i.e. with $\varepsilon = 0$).

If the weights of S are aligned on average, and $\varepsilon > |a_S^{\mathbf{w}}|$ then they shrink at a rate proportional to $\varepsilon - |a_S^{\mathbf{w}}|$. Suppose that $w_i \neq 0$ for at least one $i \in S$, and the weights of S are aligned on average, i.e. $a_S^{\mathbf{w}} > 0$. Even in this case, the weights of S shrink on average, *provided* the alignment strength $a_S^{\mathbf{w}}$ is dominated by the adversarial ε; the rate of shrinkage is proportional to $\varepsilon - |a_S^{\mathbf{w}}|$, by Eq. (12). Thus adversarial training with a *sufficiently large* ε that dominates the average strength of the features in S, will cause the weights of these features to shrink on average. This observation is key to explaining the "feature-pruning" behavior of adversarial training: "weak" features (relative to ε) are weeded out by the SGD updates.

If the weights of S are aligned, $\varepsilon < |a_S^{\mathbf{w}}|$, then the weights of S expand up to a certain point. Consider the case where at least one of the S weights is non-zero, and the adversarial ε does not dominate the average strength $a_S^{\mathbf{w}}$. Again from Eq.

(12), if $a_S^{\mathbf{w}} > 0$ and $\varepsilon < |a_S^{\mathbf{w}}|$, then the upper bound (12) on $\overline{\Delta}_S^{\mathbf{w}}$ is non-negative. Since the Theorem states that equality holds in the limit as $w_i \to 0$ for all $i \in S$, this means if all $|w_i|$ for $i \in S$ are sufficiently small, the expected SGD update $\overline{\Delta}_S^{\mathbf{w}}$ is non-negative, i.e., the S weights expand on average. In other words, the weights of a conditionally independent feature-subset S, if they are aligned on average, then their aggregate weights expand on average up to a certain point, if ε does not dominate their strength.

Note that Assumption LOSS-CVX implies that $\overline{g'} \geq 0$, and when the model $\boldsymbol{w}$ is "far" from the optimum, the values of $-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle$ will tend to be large, and since g' is a non-decreasing function (Assumption LOSS-CVX), $\overline{g'}$ will be large as well. So we can interpret $\overline{g'}$ as being a proxy for "average model error". Thus during the initial iterations of SGD, this quantity will tend to be large and positive, and shrinks toward zero as the model approaches optimality. Since $\overline{g'}$ appears as a factor in (11) and (12), we can conclude that the above effects will be more pronounced in the initial stages of SGD and less so in the later stages. The experimental results described in Section 6 are consistent with several of the above effects.

E. Generalization of Theorem 3.1 for the multi-class setting

E.1. Setting and Assumptions

Let there be $k \geq 3$ classes. For a given data point $\mathbf{x} \in \mathbb{R}^d$, its true label, $i \in [k]$, is represented by a vector $\mathbf{y} = [\underbrace{-1 \cdots -1}_{\text{i-1}}\, 1 \underbrace{-1 \cdots -1}_{\text{k-i}}]$. We assume that the input $(\mathbf{x}, \mathbf{y})$ is drawn from the distribution $\mathcal{D}$. For this multi-class classification problem, we assume the usage of the standard one-vs-all classifiers, i.e., there are k different classifiers with the i-th classifier (ideally) predicting $+1$ iff the true label of $\mathbf{x}$ is i, else it predicts -1. Let $\mathbf{w}$ represent the $k \times d$ weight matrix where $\mathbf{w_i}$ represents the $1 \times d$ weight vector for the i-th classifier. w_{ij} represents the j-th entry of $\mathbf{w_i}$. Let y_i represent the i-th entry of $\mathbf{y}$.

The assumptions presented in the main paper (Sec. 2) are slightly tweaked as follows and hold true for each of the k one-vs-all classifiers:

Assumption LOSS-INC: *The loss function for each of the one-vs-all classifier is of the form $\mathcal{L}(\mathbf{x}, y_i; \mathbf{w_i}) = g(-y_i\langle \mathbf{w_i}, \mathbf{x}\rangle)$ where g is a non-decreasing function.*

Assumption LOSS-CVX: *The loss function for each of the one-vs-all classifier is of the form $\mathcal{L}(\mathbf{x}, y_i; \mathbf{w_i}) = g(-y_i\langle \mathbf{w_i}, \mathbf{x}\rangle)$ where g is non-decreasing, almost-everywhere differentiable and convex.*

Assumption FEAT-INDEP: *The features $\mathbf{x}$ are conditionally independent given the label y_i for the i-th one-vs-all classifier, i.e., for any two distinct induces s, t, x_s is independent of x_t given y_i, or more compactly, $(x_s \perp x_t) \,|\, y_i$.*

Assumption FEAT-EXP: *For each feature $x_j, j \in [d]$ and the i-th one-vs-all classifier $\mathbb{E}(x_j|y_i) = a_{ij}.y_i$ for some constant a_{ij}.*

Additionally, we introduce a new assumption on the distribution $\mathcal{D}$ as follows.

Assumption DIST-EXPC: *The input distribution $\mathcal{D}$ satisfies the following expectation for a function $h_i, i \in [k]$ (defined by Eqs. (E.47),(E.48), (E.49)) and constant g^* (defined by Eq. (E.46))*

$$\mathbb{E}\Big[h_i(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon)\Big] = 0 \tag{E.44}$$

$$Pr_{\mathcal{D}}[\epsilon < x_j < \epsilon + \rho] = 0, \quad \rho \text{ is a small constant} \tag{E.45}$$

$$x_j \geq \epsilon + \rho \implies (x_j - \epsilon)g_i^* \geq (x_j + \epsilon)g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle) \tag{E.46}$$

If $y_i \text{sgn}(w_{ij}) = -1$, then

$$h(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon) \geq (x_j + \epsilon)g_i^* - (x_j - \epsilon)g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle) \tag{E.47}$$

If $y_i \text{sgn}(w_{ij}) = 1 \wedge x_j > \epsilon$, then

$$-\Big((x_j - \epsilon)g_i^* - (x_j + \epsilon)g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle)\Big) \leq h(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon) \leq 0 \tag{E.48}$$

If $y_i \text{sgn}(w_{ij}) = 1 \wedge x_j \leq \epsilon$, then

$$h(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon) \geq (x_j + \epsilon)g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle) - (x_j - \epsilon)g_i^* \tag{E.49}$$

This assumption is not as restrictive as it may appear. Eq. E.45 can be satisfied naturally for discrete domains. For example, for images $x_j \in \{0, 1, 2, \cdots, 254, 255\}$; thus $\rho \in (0, 1)$. For continuous domains, ρ can be set to a small value and the values of x_j can be appropriately rounded in the input dataset.

For the rest of the discussion let us consider the case where $g'(z) = c$ (for example for hinge loss function $c = 1$ for $z > -1$) and $x_j \in [0, 1]$. Now consider,

$$g^* = (1 + \epsilon)c/\rho, \;\; \rho = 0.01$$
$$f_1(x) := ((x + \epsilon)g^* - (x - \epsilon)c)$$
$$f_2(x) := ((x + \epsilon)c - (x - \epsilon)g^*)$$
$$h_i(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon) = \begin{cases} f_1(x_j) & \text{if sgn}(w_{ij})y_i = +1 \\ f_2(x_j) & \text{otherwise} \end{cases}$$
$$\mathbb{E}[h_i(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon)] = \int_0^1 Pr[x_j | y_i \text{sgn}(w_{ij}) = -1] \cdot f_1(x_j)dx +$$
$$\int_\epsilon^1 Pr[x_j | y_i \text{sgn}(w_{ij}) = +1] \cdot f_2(x)dx + \int_0^\epsilon Pr[x_j | y_i \text{sgn}(w_{ij}) = +1] \cdot f_2(x)dx$$

We observe that $f_1(x)$ is increasing in $x \in [0, 1]$, and $x \geq \epsilon + \delta \implies f_2(x) \leq 0$ and $f_2(x)$ is decreasing in $x \in [\epsilon + \delta, 1]$. Thus intuitively for $\mathbb{E}(h_i)$ to be zero, $Pr[x_j | \text{sgn}(w_{ij})y_i = -1]$ must have high values for lower magnitudes of x_j (say $x_j < 0.5$), and $Pr[x_j | \text{sgn}(w_{ij})y_i = -1]$ has low values for $x_j \leq \epsilon$ and high values for $x_j \geq \epsilon + \delta$. For example, let us assume $\epsilon = 0.1$ and that the distributions $Pr[x_i | \text{sgn}(w_{ij})y_i = +1]$ and $Pr[x_i | \text{sgn}(w_{ij})y_i = -1]$ can be approximated by truncated Gaussian distributions (with appropriate adjustments to ensure $Pr[\epsilon < x_j < \epsilon + \delta] = 0]$) with means m_1 and m_2 respectively. Then, it can be seen that there exists h_i for $m_1 < 0.3$ and $m_2 > 0.6$ such that $\mathbb{E}[h_i] = 0$.

The overall loss function, $\mathcal{L}_T$ for the multi-class classifier is the sum of the loss functions of each individual one-vs-all classifiers and is given by

$$\mathcal{L}_T(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \sum_{i=1}^{k} \mathcal{L}(\mathbf{x}, y_i; \mathbf{w_i})$$
$$= \sum_{i=1}^{k} g(-y_i\langle \mathbf{w_i}, \mathbf{x} \rangle) \qquad \text{[From Assumption } \textbf{LOSS-CVX}]$$

The **expected SGD update** $\overline{\Delta w_{ij}}$ is defined as follows:

$$\Delta w_{ij} = \begin{cases} \mathbb{E}\frac{\partial \mathcal{L}_T(\mathbf{x} + \delta^*, \mathbf{y}; \mathbf{w})}{\partial w_{ij}} & \text{when } w_i = 0 \\ -\text{sgn}(w_{ij})\mathbb{E}\frac{\partial \mathcal{L}_T(\mathbf{x} + \delta^*, \mathbf{y}; \mathbf{w})}{\partial w_{ij}} & \text{when } w_i \neq 0 \end{cases} \tag{E.50}$$

Also let

$$\overline{g_i'} := \mathbb{E}[g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle)], i \in [k] \tag{E.51}$$

Theorem E.1 (Expected SGD Update in Adversarial Training for Multi-Class Classification)**.** *For any loss function $\mathcal{L}$ satisfying assumptions **LOSS-CVX**, **FEAT-INDEP** and **FEAT-EXP**, if a data point $(\mathbf{x}, \mathbf{y})$ is randomly drawn from $\mathcal{D}$ that satisfies Assumption **DIST-EXPC**, and $\mathbf{x}$ is perturbed to $\mathbf{x}' = \mathbf{x} + \delta^*$, where δ^* is an $l_\infty(\epsilon)$-adversarial perturbation, then under the $l_\infty(\epsilon)$-adversarial loss $\mathcal{L}_T(\mathbf{x}, \mathbf{y}; \mathbf{w})$ the expected SGD-update of weight w_{ij}, namely $\overline{\Delta w_{ij}}$ satisfies the following properties*

1. *If $w_{ij} = 0$, then*

$$\overline{\Delta w_{ij}} = a_{ij}\overline{g_i'}$$

2. *If $w_{ij} \neq 0$, then*

$$\overline{\Delta w_{ij}} \leq \tilde{g}[a_{ij}\, sgn(w_{ij}) - \epsilon]$$
$$\tilde{g} \in \{\overline{g_i'}, g_i^*\}$$

Proof. For

$$\delta^* \in \mathbb{R}^d \text{ s.t} \tag{E.52}$$
$$\mathcal{L}_T(\mathbf{x} + \delta^*, \mathbf{y}; \mathbf{w}) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}}[\max_{||\delta||_\infty \leq \epsilon} \mathcal{L}_T(\mathbf{x} + \delta, \mathbf{y}; \mathbf{w})] \tag{E.53}$$

the shift by ϵ in x_j can be in either of the two directions, $y_i\text{sgn}(w_ij)$ or $-y_i\text{sgn}(w_ij)$. We have,

$$\frac{\partial \mathcal{L}_T(\mathbf{x}, \mathbf{y}; \mathbf{w}))}{\partial w_{ij}} = \sum_{s=1}^{k} \frac{\partial\big(g(-y_s\langle \mathbf{w}_s, \mathbf{x} + \delta^*\rangle)\big)}{\partial w_{ij}} = \frac{\partial g(-y_i\langle \mathbf{w}_i, \mathbf{x} + \delta^*\rangle)}{\partial w_{ij}} \tag{E.54}$$

Thus by Assumption **LOSS-CVX** either of the following two equations hold true

$$\frac{\partial \mathcal{L}_T(\mathbf{x}, \mathbf{y}; \mathbf{w}))}{\partial w_{ij}} = g'(-y_i\langle \mathbf{w}_i, \mathbf{x} + \delta^*\rangle)(-y_i x_j + \text{sgn}(w_{ij})\epsilon) \tag{E.55}$$
$$\frac{\partial \mathcal{L}_T(\mathbf{x}, \mathbf{y}; \mathbf{w}))}{\partial w_{ij}} = g'(-y_i\langle \mathbf{w}_i, \mathbf{x} + \delta^*\rangle)(-y_i x_j - \text{sgn}(w_{ij})\epsilon) \tag{E.56}$$

Thus when $w_{ij} = 0$,

$$\overline{\Delta w_{ij}} = \mathbb{E}\Big[y_i x_j g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^*\rangle)\Big]$$
$$= a_{ij}\overline{g_i'}[\text{Follows from the proof in Theorem 3.1 in the paper}]$$

Now let us consider the case when $w_{ij} \neq 0$.
Case I: Eq. E.55 is satisfied
This means that for δ^*, x_j is changed by ϵ in the direction of $-y_i\text{sgn}(w_{ij})$. Thus after multiplying throughout with $-\text{sgn}(w_{ij})$ and taking expectations, we have

$$\overline{\Delta w_{ij}} = \mathbb{E}\Big[[y_i x_j \text{sgn}(w_{ij}) - \epsilon]g'(\sum_{l=1, l\neq j}^{d} s_l\epsilon|w_{il}| + \epsilon|w_{ij}| - y_i\langle \mathbf{w_i}, \mathbf{x}\rangle)\Big] \tag{E.57}$$

where s_l represents the corresponding sign for the value $\epsilon|w_l|$ (based on which direction is x_l perturbed in) and the expectation is with respect to a random choice of data point $(\mathbf{x}, \mathbf{y})$. Let us define two random variables V and Z as follows

$$Z = y_i x_j \text{sgn}(w_{ij}) - \epsilon \tag{E.58}$$
$$V = -\sum_{l=1, l\neq j}^{d} |w_{il}|\Big(y_i x_l \text{sgn}(w_{il}) - s_l\epsilon\Big) \tag{E.59}$$

Thus,

$$\overline{\Delta w_{ij}} = \mathbb{E}[Zg'(V - |w_{ij}|Z)] \tag{E.60}$$

Let random variable Y correspond to the label y_i of the data point. Since Z is a function of feature x_j and Y , and V is a function of the remaining features and Y, Assumption **FEAT-INDEP** implies $(V \perp Z)|Y$. Additionally by Assumption **FEAT-EXP**

$$\mathbb{E}(Z) = \mathbb{E}(Z|Y) = a_j \text{sgn}(w_{ij}) - \epsilon \tag{E.61}$$

Since by Assumption **LOSS-CVX**, g' is a non-decreasing function, $g'(V - |w_{ij}|Z)$ is non-increasing in Z. Thus all three conditions of Lemma 4 are satisfied, with the random variables Z, V, Y and function f in the Lemma corresponding to random variables Z, V, Y and function g' respectively in the present theorem. Following an analysis similar to the one presented in the proof for Theorem 3.1, we have

$$\overline{\Delta w_{ij}} \leq \mathbb{E}(Z)\overline{g'_i} = \overline{g'_i}[a_{ij}\text{sgn}(w_{ij}) - \epsilon] \tag{E.62}$$

Case II: Eq. E.56 is satisfied
In this case, for δ^* x_j is perturbed by ϵ in the direction $y_i \text{sgn}(w_{ij})$. Now multiplying both sides by $-\text{sgn}(w_{ij})$

$$\Delta w_{ij} = (y_i x_j \text{sgn}(w_{ij}) + \epsilon)g'(-y_i\langle \mathbf{w_i}, \mathbf{x} + \delta^* \rangle) \tag{E.63}$$

Now let us consider the case when $y_i \text{sgn}(w_{ij}) = -1$. From Assumption **DIST-EXPC** (Eqs. (E.63),(E.46) and (E.47)), we have

$$\Delta w_{ij} \leq (y_i x_j \text{sgn}(w_{ij}) - \epsilon)g_i^* + h_i(\text{sgn}(w_{ij})y_i, \mathbf{x}, \mathbf{w_i}, \epsilon) \tag{E.64}$$

For the case when $y_i \text{sgn}(w_{ij}) = -1 \wedge x_j > \epsilon$, again from Eqs. (E.63),(E.46) and (E.48) Eq. (E.64) holds true. Similarly, for the case of $y_i \text{sgn}(w_{ij}) = -1 \wedge x_j \leq \epsilon$, the validity of Eq. (E.64) can be verified from Eqs. (E.63),(E.46) and (E.49). Now taking expectations over both sides of Eq. (E.64) results in

$$\overline{\Delta w_{ij}} \leq (a_{ij}\text{sgn}(w_{ij}) - \epsilon)g_i^* \qquad \text{[From Eq. (E.44)]} \tag{E.65}$$

This concludes our proof. □

F. Proof of Lemma 4.1

Lemma 4.1 (IG Attribution for 1-layer Networks). *If $F(\boldsymbol{x})$ is computed by a 1-layer network (14) with weights vector $\boldsymbol{w}$, then the Integrated Gradients for all dimensions of $\boldsymbol{x}$ relative to a baseline $\boldsymbol{u}$ are given by:*

$$\text{IG}^F(\boldsymbol{x}, \boldsymbol{u}) = [F(\boldsymbol{x}) - F(\boldsymbol{u})]\frac{(\boldsymbol{x} - \boldsymbol{u}) \odot \boldsymbol{w}}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w}\rangle}, \tag{15}$$

where the $\odot$ operator denotes the entry-wise product of vectors.

Proof. Since the function F, the baseline input $\boldsymbol{u}$ and weight vector $\boldsymbol{w}$ are fixed, we omit them from $\text{IG}^F(\boldsymbol{x}, \boldsymbol{u})$ and $\text{IG}_i^F(\boldsymbol{x}, \boldsymbol{u})$ for brevity. Consider the partial derivative $\partial_i F(\boldsymbol{u} + \alpha(\boldsymbol{x} - \boldsymbol{u}))$ in the definition (13) of $\text{IG}_i(\boldsymbol{x})$. For a given $\boldsymbol{x}, \boldsymbol{u}$ and α, let $\boldsymbol{v}$ denote the vector $\boldsymbol{u} + \alpha(\boldsymbol{x} - \boldsymbol{u})$. Then $\partial_i F(\boldsymbol{v}) = \partial F(\boldsymbol{v})/\partial v_i$, and by applying the chain rule we get:

$$\partial_i F(\boldsymbol{v}) := \frac{\partial F(\boldsymbol{v})}{\partial v_i} = \frac{\partial A(\langle \boldsymbol{w}, \boldsymbol{v}\rangle)}{\partial v_i} = A'(z)\frac{\partial \langle \boldsymbol{w}, \boldsymbol{v}\rangle}{\partial v_i} = w_i A'(z),$$

where $A'(z)$ is the gradient of the activation A at $z = \langle \boldsymbol{w}, \boldsymbol{v}\rangle$. This implies that:

$$\begin{aligned}
\frac{\partial F(\boldsymbol{v})}{\partial \alpha} &= \sum_{i=1}^{d}\left(\frac{\partial F(\boldsymbol{v})}{\partial v_i}\frac{\partial v_i}{\partial \alpha}\right) \\
&= \sum_{i=1}^{d}[w_i A'(z)(x_i - u_i)] \\
&= \langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w}\rangle A'(z)
\end{aligned}$$

We can therefore write

$$dF(\boldsymbol{v}) = \langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle A'(z) d\alpha,$$

and since $\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle$ is a scalar, this yields

$$A'(z) d\alpha = \frac{dF(\boldsymbol{v})}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle}$$

Using this equation the integral in the definition of $IG_i(x)$ can be written as

$$\begin{aligned} \int_{\alpha=0}^{1} \partial_i F(\boldsymbol{v}) d\alpha &= \int_{\alpha=0}^{1} w_i A'(z) d\alpha \\ &= \int_{\alpha=0}^{1} w_i \frac{dF(\boldsymbol{v})}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle} \\ &= \frac{w_i}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle} \int_{\alpha=0}^{1} dF(\boldsymbol{v}) \qquad \text{(F.66)} \\ &= \frac{w_i}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle} [F(\boldsymbol{x}) - F(\boldsymbol{u})], \end{aligned}$$

where (F.66) follows from the fact that $(\boldsymbol{x} - \boldsymbol{u})$ and $\boldsymbol{w}$ do not depend on α. Therefore from the definition (13) of $\mathrm{IG}_i(\boldsymbol{x})$:

$$\mathrm{IG}_i(\boldsymbol{x}) = [F(\boldsymbol{x}) - F(\boldsymbol{u})] \frac{(x_i - u_i) w_i}{\langle \boldsymbol{x} - \boldsymbol{u},\, \boldsymbol{w} \rangle},$$

and this yields the expression (15) for $\mathrm{IG}(\boldsymbol{x})$. □

G. Proof of Theorem 5.1

Theorem 5.1 (Equivalence of Stable IG and Adversarial Robustness)**.** *For loss functions $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$ satisfying Assumption LOSS-CVX, the augmented loss inside the expectation (17) equals the $\ell_\infty(\varepsilon)$-adversarial loss inside the expectation (2), i.e.*

$$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) \;+ \max_{||\,\boldsymbol{x}' - \boldsymbol{x}\,||_\infty \le \varepsilon} ||\,\mathrm{IG}^{\mathcal{L}_y}(\boldsymbol{x}, \boldsymbol{x}')||_1 \;= \max_{||\delta||_\infty \le \varepsilon} \mathcal{L}(\boldsymbol{x} + \delta, y; \boldsymbol{w}) \tag{18}$$

Proof. Recall that Assumption LOSS-CVX implies $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = g(-y\langle \boldsymbol{w},\, \boldsymbol{x} \rangle)$ for some non-decreasing, differentiable, convex function g. Due to this special form of $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$, the function $\mathcal{L}_y$ is a differential function of $\langle \boldsymbol{w},\, \boldsymbol{x} \rangle$, and by Lemma 4.1 the i'th component of the IG term in (18) is

$$\mathrm{IG}_i^{\mathcal{L}_y}(\boldsymbol{x}, \boldsymbol{x}';\; \boldsymbol{w}) = \frac{\boldsymbol{w}_i (\boldsymbol{x}' - \boldsymbol{x})_i}{\langle \boldsymbol{w}, \boldsymbol{x}' - \boldsymbol{x} \rangle} \cdot \big(g(-y\langle \boldsymbol{w}, \boldsymbol{x}' \rangle) - g(-y\langle \boldsymbol{w}, \boldsymbol{x} \rangle), \big)$$

and if we let $\Delta = \boldsymbol{x}' - \boldsymbol{x}$ (which satisfies that $\|\Delta\|_\infty \le \varepsilon$), its absolute value can be written as

$$\frac{\big| g(-y\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y\langle \boldsymbol{w}, \Delta \rangle) \;-\; g(-y\langle \boldsymbol{w}, \boldsymbol{x} \rangle)\big|}{|\langle \boldsymbol{w}, \Delta \rangle|} \cdot |\,\boldsymbol{w}_i\, \Delta_i|$$

Let $z = -y\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ and $\delta = -y\langle \boldsymbol{w}, \Delta \rangle$, this is further simplified as $\frac{|g(z+\delta) - g(z)|}{|\delta|} |w_i \Delta_i|$. By Assumption LOSS-CVX, g is convex, and therefore the "chord slope" $[g(z+\delta) - g(z)]/\delta$ cannot decrease as δ is increased. In particular to maximize the ℓ_1-norm of the IG term in Eq (18), we can set δ to be largest possible value subject to the constraint $||\Delta||_\infty \le \varepsilon$, and we achieve this by setting $\Delta_i = -y\, \mathrm{sgn}(\boldsymbol{w}_i)\varepsilon$, for each dimension i. This yields $\delta = \|\,\boldsymbol{w}\,\|_1 \varepsilon$, and the second term on the LHS of (18) becomes

$$\begin{aligned} |g(z+\delta) - g(z)| \cdot \frac{\sum_i |\,\boldsymbol{w}_i\, \Delta_i|}{|\delta|} &= |g(z + \varepsilon\|\,\boldsymbol{w}\,\|_1) - g(z)| \cdot \frac{\sum_i |\,\boldsymbol{w}_i\,|\varepsilon}{\|\,\boldsymbol{w}\,\|_1 \varepsilon} \\ &= |g(z + \varepsilon\|\,\boldsymbol{w}\,\|_1) - g(z)| \\ &= g(z + \varepsilon\|\,\boldsymbol{w}\,\|_1) - g(z) \end{aligned}$$

where the last equality follows because g is nondecreasing. Since $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = g(z)$ by Assumption LOSS-CVX, the LHS of (18) simplifies to

$$g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle + \varepsilon \| \boldsymbol{w} \|_1),$$

and by Eq. (7), this is exactly the $\ell_\infty(\varepsilon)$-adversarial loss on the RHS of (18). □

H. Aggregate IG Attribution over a Dataset

Recall that in Section 4 we defined $\mathrm{IG}^F(\boldsymbol{x}, \boldsymbol{u})$ in Eq. (13) for a *single* input $\boldsymbol{x}$ (relative to a baseline input $\boldsymbol{u}$). This gives us a sense of the "importance" of each input feature in explaining a *specific* model prediction $F(\boldsymbol{x})$. Now we describe some ways to produce *aggregate* importance metrics over an entire dataset. For brevity let us simply write $\mathrm{IG}(\boldsymbol{x})$ and $\mathrm{IG}_i(\boldsymbol{x})$ and omit F and $\boldsymbol{u}$ since these are fixed for a given model and a given dataset.

Note that in Eq. 13, $\boldsymbol{x}$ is assumed to be an input vector in "exploded" space, i.e., all categorical features are (explicitly or implicitly) one-hot encoded, and i is the position-index corresponding to either a specific numerical feature, or a categorical feature-*value*. Thus if i corresponds to a categorical feature-value, then for any input $\boldsymbol{x}$ where $x_i = 0$ (i.e. the corresponding categorical feature-value is not "active" for that input), $\mathrm{IG}_i(\boldsymbol{x}) = 0$. A natural definition of the overall importance of a feature (or feature-value) i for a given model F and dataset $\mathcal{D}$, is the average of $|\mathrm{IG}_i(\boldsymbol{x})|$ over all inputs $\boldsymbol{x} \in \mathcal{D}$, which we refer to as the **Feature Value Impact** $FV_i[\mathcal{D}]$. For a categorical feature with m possible values, we can further define its **Feature-Impact** (FI) as the *sum* of $FV_i[\mathcal{D}]$ over all i corresponding to possible values of this categorical feature.

The FI metric is particularly useful in tabular datasets to gain an understanding of the aggregate importance of high-cardinality categorical features.

I. Definition of the Gini Index

The definition is adapted from (Hurley & Rickard, 2009): Suppose we are given a vector of non-negative values $\boldsymbol{v} = [v_1, v_2, v_3, \ldots, v_d]$. The vector is first *sorted* in non-decreasing order, so that the resulting indices after sorting are $(1), (2), (3), \ldots, (d)$, i.e., $v_{(k)}$ denotes the k'th value in this sequence. Then the Gini Index is given by:

$$G(\boldsymbol{v}) = 1 - 2\sum_{k=1}^{d} \frac{v_{(k)}}{|| \boldsymbol{v} ||_1} \Big(\frac{d-k+0.5}{d}\Big). \tag{I.67}$$

Another equivalent definition of the Gini Index is based on plotting the cumulative fractional contribution of the sorted values. In particular if the sorted non-negative values are $[v_{(1)}, v_{(2)}, \ldots, v_{(d)}]$, and for $k \in [d]$, we plot k/d (the fraction of dimensions up to k) vs $\frac{\sum_{i=1}^{k} v_{(i)}}{|| \boldsymbol{v} ||_1}$ (the fraction of values until the k'th dimension), then the Gini Index $G(\boldsymbol{v})$ is 0.5 minus the area under this curve

The Gini Index by definition lies in [0,1], and a higher value indicates more sparseness. For example if just one of the $v_i > 0$ and all the rest are 0, then $G(v) = 1.0$, indicating perfect sparseness. At the other extreme, if all v_i are equal to some positive constant, then $G(v) = 0$.

J. Experiments

J.1. Experiment Datasets and Methodology

We experiment with 5 public benchmark datasets. Below we briefly describe each dataset and model-training details.

MNIST. This is a classic image benchmark dataset consisting of grayscale images of handwritten digits 0 to 9 in the form of 28 x 28 pixels, along with the correct class label (0 to 9) (LeCun & Cortes, 2010). We train a Deep Neural Network consisting of two convolutional layers with 32 and 64 filters respectively, each followed by 2x2 max-pooling, and a fully connected layer of size 1024. Note that this is identical to the state-of-the-art adversarially trained model used by (Madry et al., 2017). We use 50,000 images for training, and 10,000 images for testing. When computing the IG vector for an input image, we use the predicted probability of the *true class* as the function F in the definition (13) of IG. For training each of the model types on MNIST, we use the Adam optimizer with a learning rate 10^{-4}, with a batch size of 50. For the naturally-trained model (with or without ℓ_1-regularization) we use 25,000 training steps. For adversarial training, we use 100,000 training steps overall, and to generate adversarial examples we use Projected Gradient Descent (PGD) with random start. The PGD hyperparameters depend on the specific ε bound on the ℓ_∞-norm of the adversarial perturbations: the number of PGD steps was set as $\varepsilon * 100 + 10$, and the PGD step size was set to 0.01.

Fashion-MNIST. This is another image benchmark dataset which is a drop-in replacement for MNIST (Xiao et al., 2017). Images in this dataset depict wearables such as shirts and boots instead of digits. The image format, the number of classes, as well as the number of train/test examples are all identical to MNIST. We use the same model and training details as for MNIST.

CIFAR-10. The CIFAR-10 dataset (Krizhevsky et al., 2009) is a dataset of 32x32 color images with ten classes, each consisting of 5,000 training images and 1,000 test images. The classes correspond to dogs, frogs, ships, trucks, etc. The pixel values are in range of $[0, 255]$. We use a wide Residual Network (He et al., 2016), which is identical to the state-of-art adversarially trained model on CIFAR-10 used by (Madry et al., 2017). When computing the IG vector for an input image, we use the predicted probability of the *true class* as the function F in the definition (13) of IG. For training each of the model types on CIFAR-10, we use Momentum Optimizer with weight decay. We set momentum rate as 0.9, weight decay rate as 0.0002, batch size as 128, and training steps as 70,000. We use learning rate schedule: the first 40000 steps, we use learning rate of 10^{-1}; after 40000 steps and before 60,000 steps, we use learning rate of 10^{-2}; after 60,000 steps, we use learning rate of 10^{-3}. We use Projected Gradient Descent (PGD) with random start to generate adversarial examples. The PGD hyperparameters depend on the specific ε bound on the ℓ_∞-norm of the adversarial perturbations: the number of PGD steps was set as $\varepsilon + 1$, and the PGD step size was set to 1.

Mushroom. This is a standard tabular public dataset from the UCI Data Repository (Dheeru & Karra Taniskidou, 2017). The dataset consists of 8142 instances, each of which corresponds to a different mushroom species, and has 22 categorical features (and no numerical features), whose cardinalities are all under 10. The task is to classify an instance as edible (label=1) or not (label=0). We train a simple *logistic regression* model to predict the probability that the mushroom is edible, with a 70/30 train/test split, and use a 0.5 threshold to make the final classification. We train the models on 1-hot encoded feature vectors, and the IG computation is on these (sparse) 1-hot vectors, with the output function F being the final predicted probability. We train logistic regression models for this dataset, and for natural model training (with or without ℓ_1-regularization) we use the Adam optimizer with a learning rate of 0.01, batch size of 32, and 30 training epochs. Adversarial training is similar, except that each example batch is perturbed using the closed-form expression (7).

Spambase. This is another tabular dataset from the UCI Repository, consisting of 4601 instances with 57 numerical attributes (and no categorical ones). The instances are various numerical features of a specific email, and the task is classify the email as spam (label = 1) or not (label = 0). The model and training details are similar to those for the mushroom dataset.

The code for all experiments (included along with the supplement) was written using Tensorflow 2.0. The following subsections contain results that were left out of the main body of the paper due to space constraints.

J.2. Mushroom Dataset: Average IG-based Feature Impact

We contrast between the weights learned by natural training and adversarial training with $\varepsilon = 0.1$. Since all features in this dataset are categorical, many with cardinalities close to 10, there are too many features in the "exploded" space to allow a clean display, so we instead look at the average Feature Impact (FI, defined in Section H) over the (natural, unperturbed) test dataset, see Figure J.4. It is worth noting that several features that have a significant impact on the naturally-trained model have essentially no impact on the adversarially trained model.

J.3. Spambase: Average IG-based Feature Impact

We fix $\varepsilon = 0.1$ for adversarial training and show in Figure J.5 a bar-plot comparing the average Feature-Impacts (FI), between naturally-trained and adversarially-trained models. Note how the adversarially trained model has significantly fewer features with non-negligible impacts, compared to a naturally trained model.

J.4. MNIST, Fashion-MNIST and CIFAR-10: examples

Figs. J.6, J.7 and J.8 below show IG-based saliency maps of images correctly classified by three model types: Naturally trained un-regularized model, naturally trained model with ℓ_1-regularization, and an $\ell_\infty(\varepsilon)$-adversarially trained model. The values of λ and ε are those indicated in Table 1. In each example, all three models predict the correct class with high probability, and we compare the Gini Indices of the IG-vectors (with respect to the predicted probability of the true class).

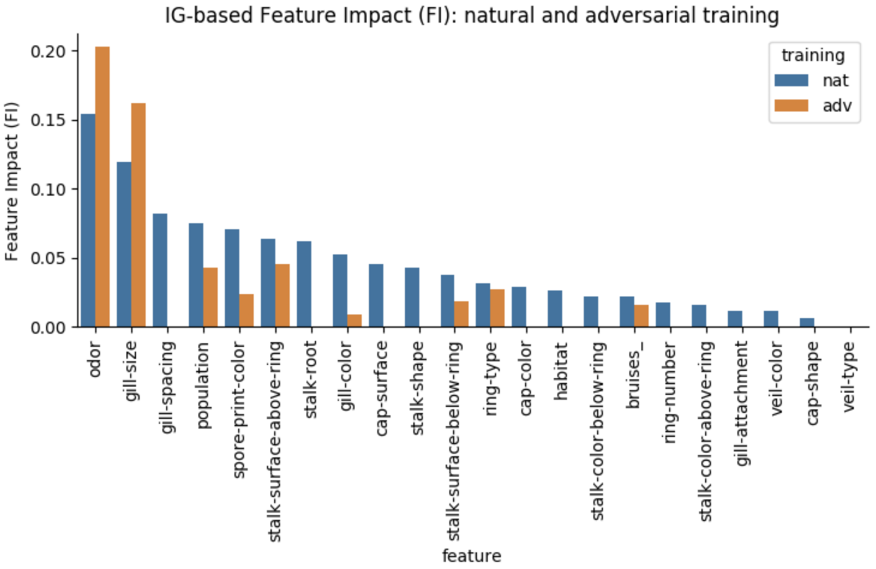


Figure J.4: Comparison of aggregate Feature Impact (FI) for a naturally-trained model, and an adversarially-trained model with $\varepsilon = 0.1$, on the mushroom dataset. The features are arranged left to right in decreasing order of the FI value in the naturally-trained model.

The sparseness of the saliency maps of the adversarially-trained models is visually striking compared to those of the other two models, and this is reflected in the Gini Indices as well. Figs. J.9 and J.10 show analogous results, but using the DeepSHAP (Lundberg & Lee, 2017) attribution method instead of IG. The effect of adversarial training on the sparseness of the saliency maps is even more visually striking when using DeepSHAP, compared to IG (We had difficulty running DeepSHAP on CIFAR-10 data, so we are only able to show results for DeepSHAP on MNIST and Fashion-MNIST).

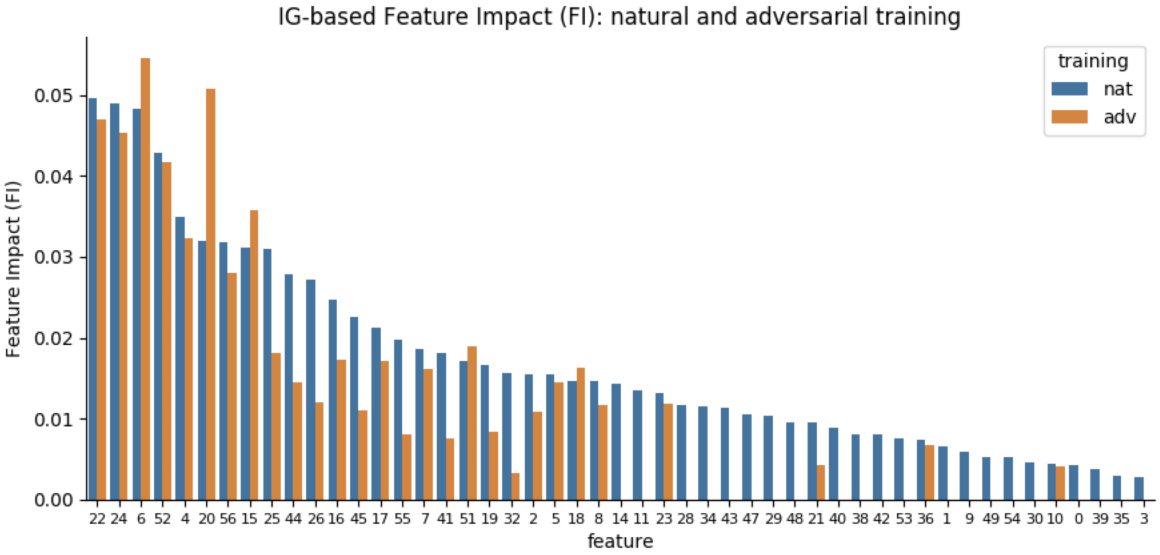
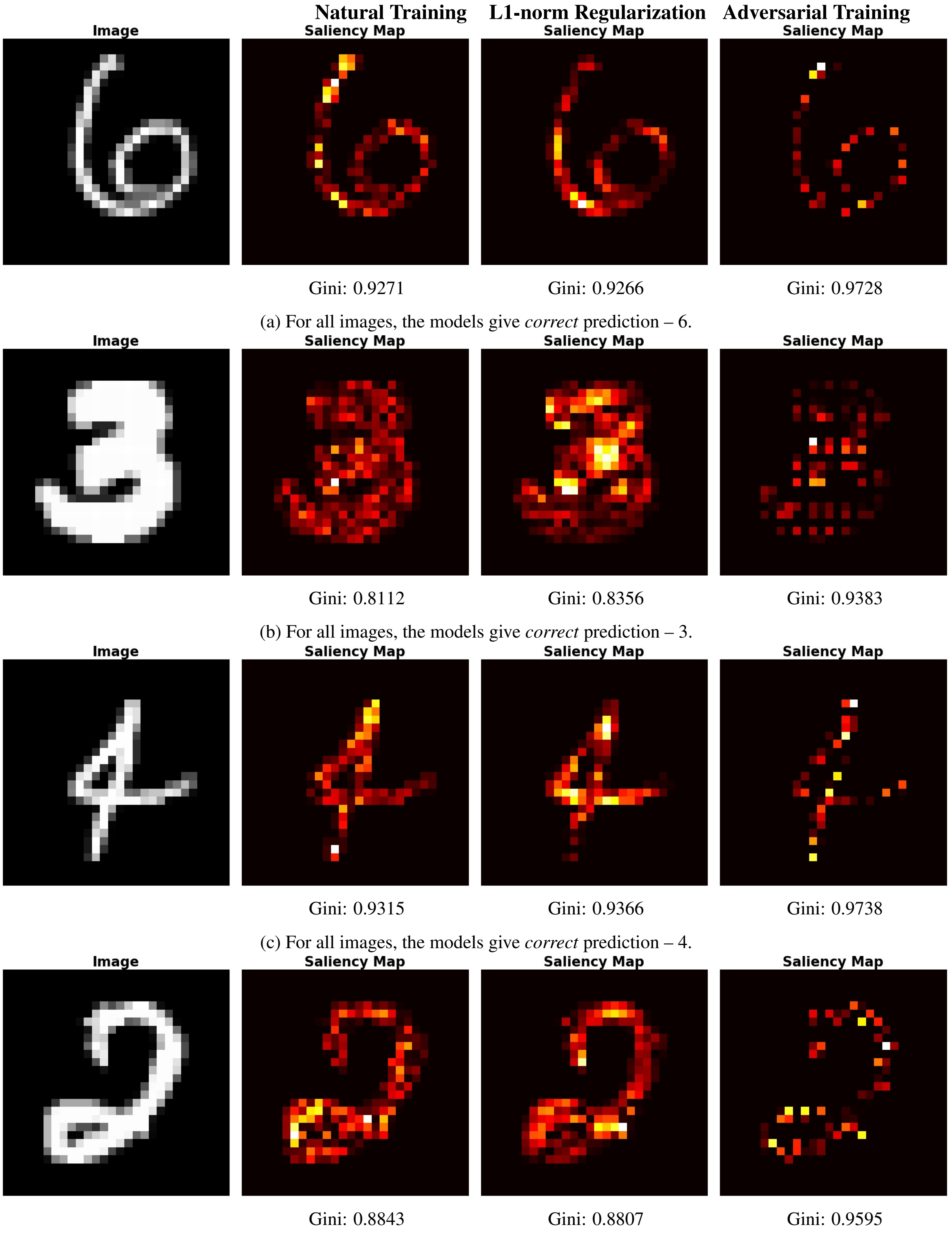


Figure J.5: Comparison of aggregate Feature Impact (FI) for a naturally-trained model, and an adversarially-trained model with $\varepsilon = 0.1$, on the spambase dataset. The features are arranged left to right in decreasing order of their FI in the naturally-trained model. To avoid clutter, we show only features that have an FI at least 5% of the highest FI (across both models).



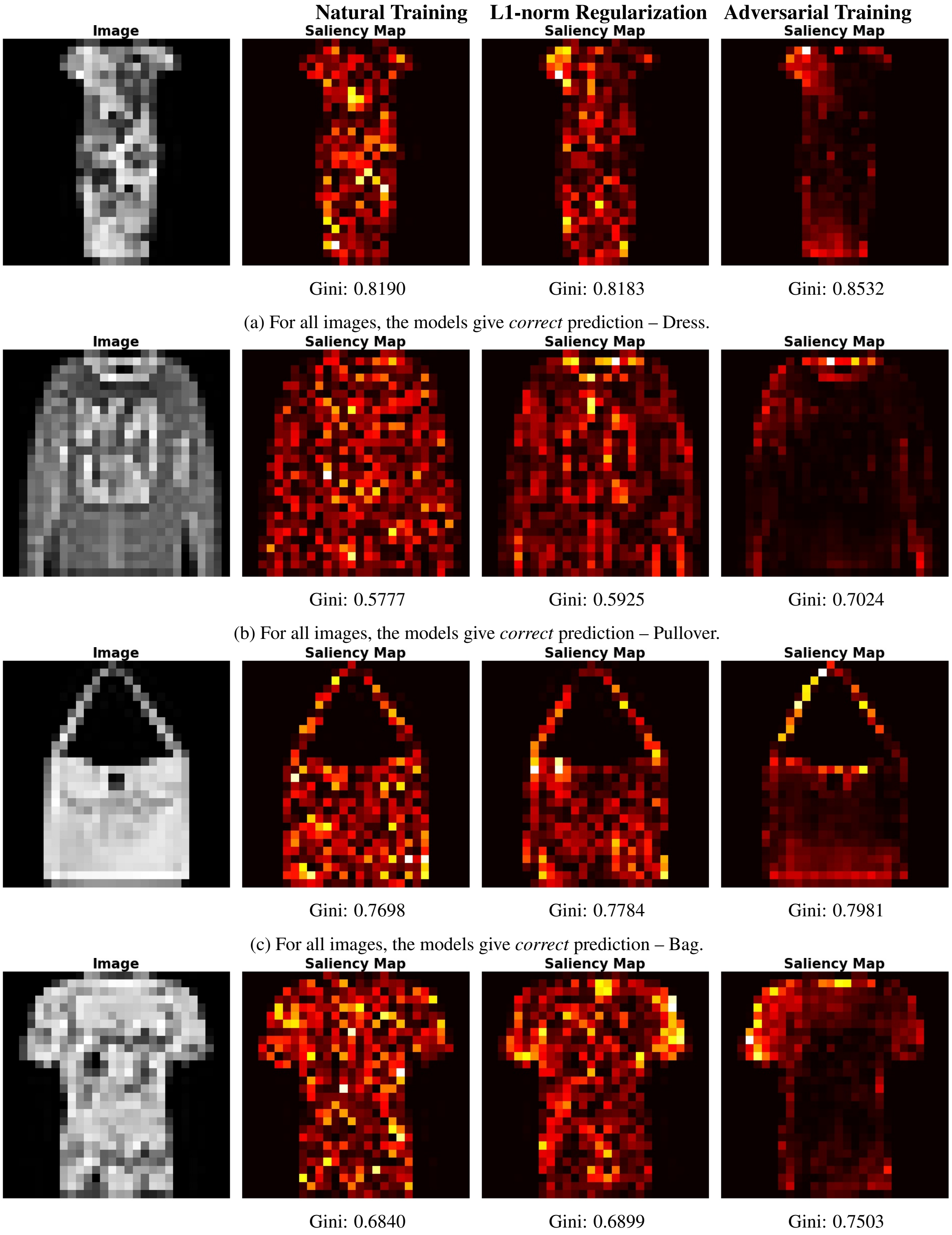
(a) For all images, the models give *correct* prediction – 6.

(b) For all images, the models give *correct* prediction – 3.

(c) For all images, the models give *correct* prediction – 4.

(d) For all images, the models give *correct* prediction – 2.

Figure J.6: Some examples on MNIST. We can see the saliency maps (also called feature importance maps), computed via IG, of adversarially trained model are much sparser compared to other models.



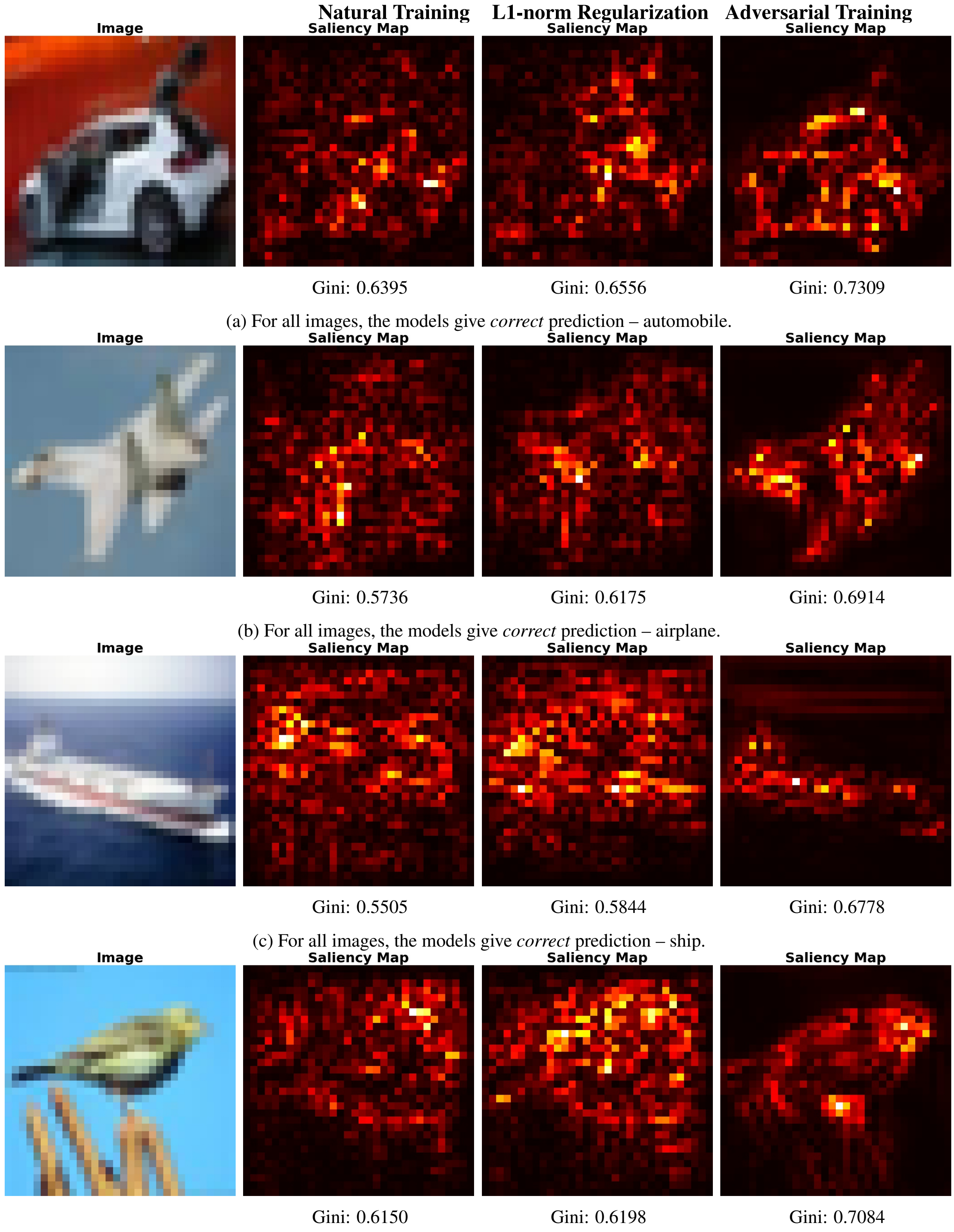
(a) For all images, the models give *correct* prediction – Dress.

(b) For all images, the models give *correct* prediction – Pullover.

(c) For all images, the models give *correct* prediction – Bag.

(d) For all images, the models give *correct* prediction – T-shirt.

Figure J.7: Some examples on Fashion-MNIST. We can see the saliency maps (also called feature importance maps), computed via IG, of adversarially trained model are much sparser compared to other models.



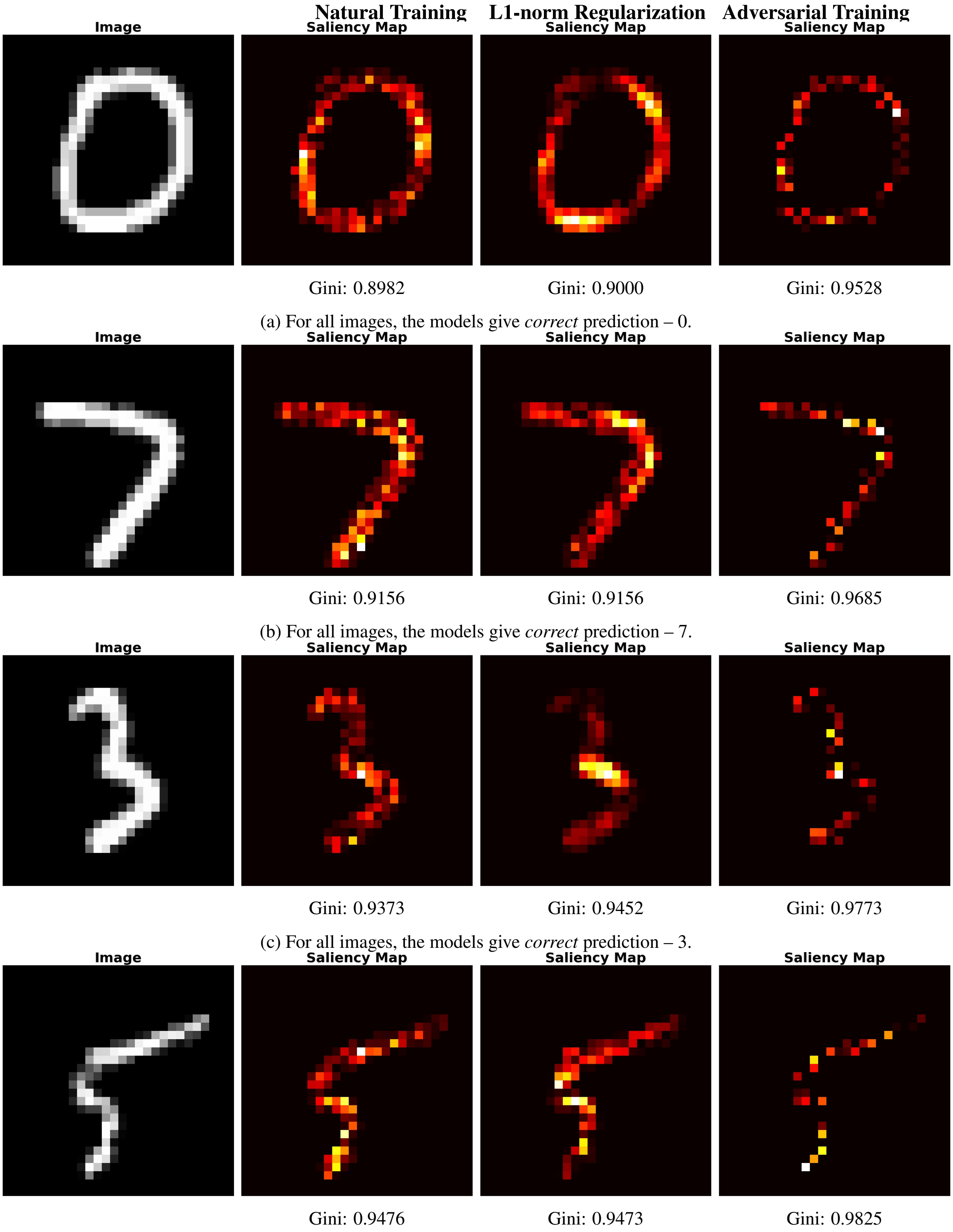
(a) For all images, the models give *correct* prediction – automobile.

(b) For all images, the models give *correct* prediction – airplane.

(c) For all images, the models give *correct* prediction – ship.

(d) For all images, the models give *correct* prediction – bird.

Figure J.8: Some examples on CIFAR-10. We can see the saliency maps (also called feature importance maps), computed via IG, of adversarially trained model are much sparser compared to other models.



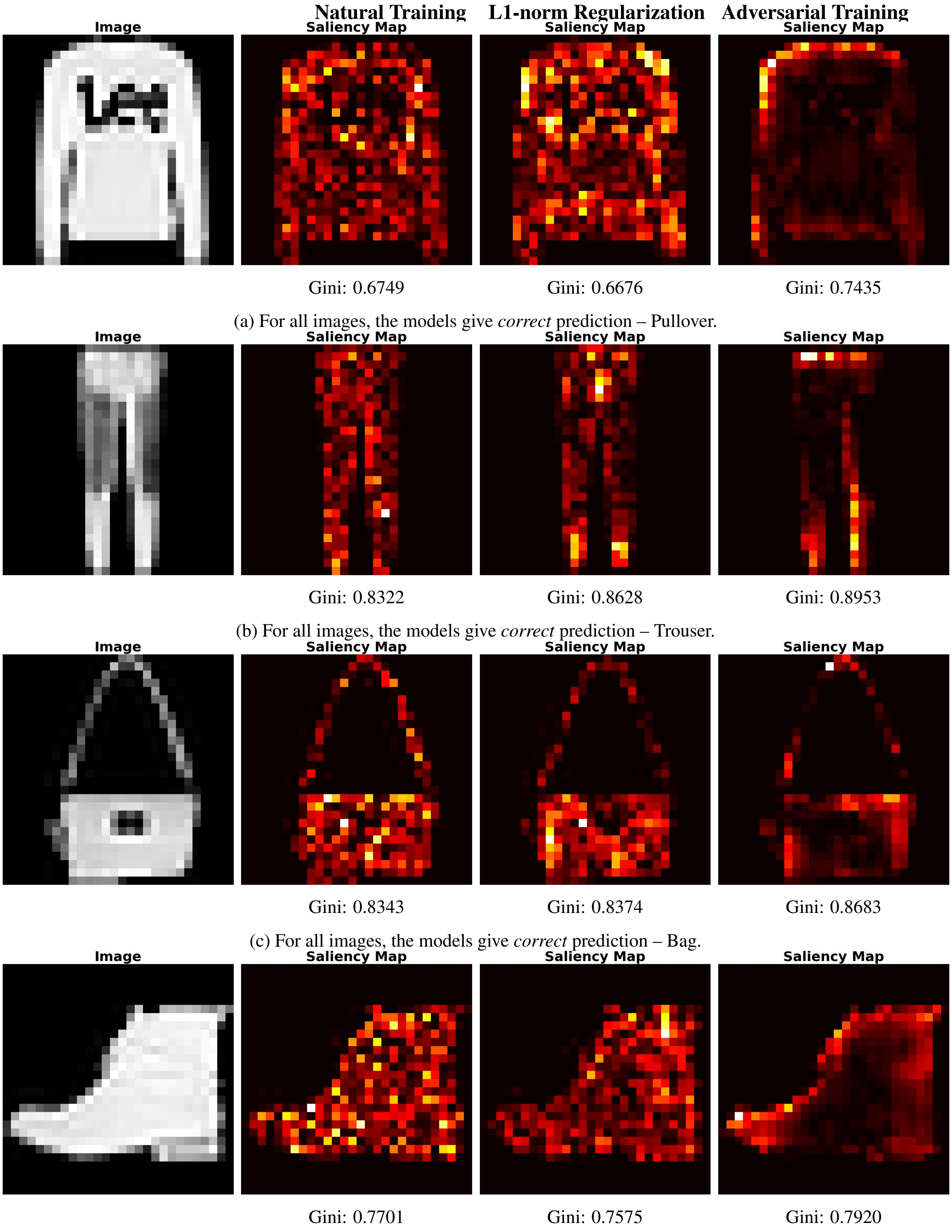
(a) For all images, the models give *correct* prediction – 0.

(b) For all images, the models give *correct* prediction – 7.

(c) For all images, the models give *correct* prediction – 3.

(d) For all images, the models give *correct* prediction – 5.

Figure J.9: Some examples on MNIST. We can see the saliency maps (also called feature importance maps), computed via DeepSHAP, of adversarially trained model are much sparser compared to other models.



(a) For all images, the models give *correct* prediction – Pullover.

(b) For all images, the models give *correct* prediction – Trouser.

(c) For all images, the models give *correct* prediction – Bag.

(d) For all images, the models give *correct* prediction – Ankle boot.

Figure J.10: Some examples on Fashion-MNIST. We can see the saliency maps (also called feature importance maps), computed via DeepSHAP, of adversarially trained model are much sparser compared to other models.