A. Proofs

In this section, we provide the proofs of our theoretical results in Section 3.

For ease of presentation, we introduce the following notations and an alternative definition of the ∞-Wasserstein distance. Let $(\mathcal{X}, \mu)$ and $(\mathcal{Y}, \nu)$ be two probability spaces. We say that $T : \mathcal{X} \to \mathcal{Y}$ transports $\mu \in \mathcal{P}(\mathcal{X})$ to $\nu \in \mathcal{P}(\mathcal{Y})$, and we call T a transport map, if $\nu(B) = \mu(T^{-1}(B))$, for all ν-measurable sets B. In addition, for any measurable map $T : \mathcal{X} \to \mathcal{Y}$, we define the *pushforward* of μ through T as $(T_{\#}(\mu))$ given by

$$(T_{\#}(\mu))(B) = \mu(T^{-1}(B)), \quad \text{for any measurable } B \subseteq \mathcal{Y}.$$

Alternative definition of ∞-Wasserstein distance. From the perspective of transportation theory, given two probability measures μ and ν on $(\mathcal{X}, \Delta)$, any joint probability distribution $\gamma \in \Gamma(\mu, \nu)$ corresponds to a specific transport map $T : \mathcal{X} \to \mathcal{X}$ that moves μ to ν. Then, the p-th Wasserstein distance can viewed as finding the optimal transport map to move from μ to ν that minimizes some cost functional depending on p (Kolouri et al., 2017). For the case where $p = \infty$, if we let T be the transport map induced by a given $\gamma \in \Gamma(\mu, \nu)$, then the cost functional can be informally understood as the maximum of all the transport distances $\Delta(T(\boldsymbol{x}), \boldsymbol{x})$. More rigorously, the ∞-Wasserstein distance can be alternatively defined as

$$\begin{aligned}\mathrm{W}_\infty(\mu, \nu) &\coloneqq \inf_{\gamma \in \Gamma(\mu,\nu)} \gamma\text{-ess sup}_{(\boldsymbol{x},\boldsymbol{x}') \in \mathcal{X}^2} \Delta(\boldsymbol{x}, \boldsymbol{x}') \\ &= \inf_{\gamma \in \Gamma(\mu,\nu)} \inf\big\{t \geq 0 \colon \gamma\big(\Delta(\boldsymbol{x}, \boldsymbol{x}') > t\big) = 0\big\}.\end{aligned}$$

A more detailed discussion of ∞-Wasserstein distance can be found in Champion et al. (2008).

A.1. Proof of Theorem 3.2

Theorem 3.2 (restated here) connects the vulnerability of a given representation with the minimum adversarial gap of any classifier based on that representation.

Theorem 3.2. Let $(\mathcal{X}, \|\cdot\|_p)$ be the input metric space and $\mathcal{Y} = \{-1, 1\}$ be the label space. Assume the underlying data are generated according to (3.1). Consider the feature space $\mathcal{Z} = \{-1, 1\}$ and the set of representations,

$$\mathcal{G}_{\text{bin}} = \{g : \boldsymbol{x} \mapsto \operatorname{sgn}(\boldsymbol{w}^\top \boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X} \mid \|\boldsymbol{w}\|_2 = 1\}.$$

Let $\mathcal{H} = \{h : \mathcal{Z} \to \mathcal{Y}\}$ be the set of non-trivial downstream classifiers.[6] Given $\epsilon \geq 0$, for any $g \in \mathcal{G}_{\text{bin}}$, we have

$$\int_{\frac{1}{2} - \mathrm{AG}_\epsilon(f^*)}^{\frac{1}{2}} \mathrm{H}_2'(\theta) d\theta \leq \mathrm{RV}_\epsilon(g) \leq \int_{\frac{1}{2} - \frac{1}{2}\mathrm{AG}_\epsilon(f^*)}^{\frac{1}{2}} \mathrm{H}_2'(\theta) d\theta,$$

where $f^* = \operatorname{argmin}_{h \in \mathcal{H}} \mathrm{AdvRisk}_\epsilon(h \circ g)$ is the optimal classifier based on g, $\mathrm{H}_2(\theta) = -\theta \log \theta - (1-\theta)\log(1-\theta)$ is the binary entropy function and H_2' denotes its derivative.

Proof. Let μ_{XY} be the underlying joint probability distribution of the examples according to (3.1) and μ_X be corresponding the marginal distribution of X. To begin with, we compute the explicit formulation for the defined representation vulnerability in Definition 3.1. Note that $\mathrm{I}(U; V) = \mathrm{H}(U) - \mathrm{H}(U|V) = \mathrm{H}(V) - \mathrm{H}(V|U)$. Thus, for any $g_{\boldsymbol{w}} \in \mathcal{G}$ we have

$$\begin{aligned}\mathrm{RV}_\epsilon(g_{\boldsymbol{w}}) &= \mathrm{H}(g_{\boldsymbol{w}}(X)) - \inf_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \mathrm{H}(g_{\boldsymbol{w}}(X')) \\ &= 1 - \inf_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \Bigg(- \Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' \geq 0) \cdot \log \Big[\Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' \geq 0) \Big] \\ &\qquad - \Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' < 0) \cdot \log \Big[\Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' < 0) \Big] \Bigg),\end{aligned}$$

[6] To be more specific, we do not consider the case where h is a constant function. Under our problem setting, there are two elements in $\mathcal{H}$, namely $h_1(z) = z$, $h_2(z) = -z$, for any $\boldsymbol{z} \in \mathcal{Z}$.

where the first equality holds because $\mathrm{H}(g_{\boldsymbol{w}}(U) \mid U) = 0$ for any random variable U, and the second equality is due to the fact that the distribution of X is symmetric with respect to $\boldsymbol{w}^\top \boldsymbol{x} = 0$. Note that the binary entropy function $\mathrm{H}_2(\theta) = -\theta \log \theta - (1-\theta)\log(1-\theta)$ is monotonically increasing with respect to θ in $[0, 1/2)$ and monotonically decreasing in $(1/2, 1]$. Therefore, the optimal value of $\mathrm{RV}_\epsilon(g_{\boldsymbol{w}})$ is achieved when $\mu_{X'}$ either minimizes $\Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' \geq 0)$ or maximizes $\Pr_{\boldsymbol{x}' \sim \mu_{X'}}(\boldsymbol{w}^\top \boldsymbol{x}' \geq 0)$.

According to the Hölder's inequality, we have $|\langle \mathbf{a}, \boldsymbol{b} \rangle| \leq \|\mathbf{a}\|_p \cdot \|\boldsymbol{b}\|_q$ for any $\mathbf{a}, \boldsymbol{b} \in \mathbb{R}^d$, where $1/p + 1/q = 1$. By the alternative definition of ∞-Wasserstein distance, for any $\mu_{X'}$ that satisfies $\mathrm{W}_\infty(\mu_{X'}, \mu_X) \leq \epsilon$, it induces a transport map $T : \mathcal{X} \to \mathcal{X}$ such that $\mu'_X = T_{\#}(\mu_X)$ and $\|\Delta(T(X), X)\|_p \leq \epsilon$ holds almost surely with respect to the randomness of X and T. Thus, we have

$$\Pr_{\boldsymbol{x} \sim \mu_X} \Big[-\epsilon \cdot \|\boldsymbol{w}\|_q \leq \boldsymbol{w}^\top (T(\boldsymbol{x}) - \boldsymbol{x}) \leq \epsilon \cdot \|\boldsymbol{w}\|_q \Big] \geq \Pr_{\boldsymbol{x} \sim \mu_X} \Big[\|T(\boldsymbol{x}) - \boldsymbol{x}\|_p \leq \epsilon \Big] = 1,$$

which implies

$$\Pr_{\boldsymbol{x} \sim \mu_X} \big(\boldsymbol{w}^\top \boldsymbol{x} - \epsilon \cdot \|\boldsymbol{w}\|_q \geq 0 \big) \leq \Pr_{\boldsymbol{x}' \sim \mu_{X'}} (\boldsymbol{w}^\top \boldsymbol{x}' \geq 0) \leq \Pr_{\boldsymbol{x} \sim \mu_X} \big(\boldsymbol{w}^\top \boldsymbol{x} + \epsilon \cdot \|\boldsymbol{w}\|_q \geq 0 \big).$$

We remark that the equality can be achieved when the the transport map T is constructed by perturbing the i-th element of any sampled $\boldsymbol{x} \sim \mu_X$ by $\epsilon \cdot (w_i^q / \sum_i w_i^q)^{1/p}$, for any $i = 1, 2, \ldots, d$. In addition, according to the assumed Gaussian Mixture model (3.1), we have

$$\begin{aligned}
\Pr_{\boldsymbol{x} \sim \mu_X} \big(\boldsymbol{w}^\top \boldsymbol{x} - \epsilon \cdot \|\boldsymbol{w}\|_q \geq 0 \big) &= \frac{1}{2} \Pr_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*)} \big[\boldsymbol{w}^\top \boldsymbol{x} \geq \epsilon \cdot \|\boldsymbol{w}\|_q \big] + \frac{1}{2} \Pr_{\boldsymbol{x} \sim \mathcal{N}(-\boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*)} \big[\boldsymbol{w}^\top \boldsymbol{x} \geq \epsilon \cdot \|\boldsymbol{w}\|_q \big] \\
&= \frac{1}{2} - \frac{1}{2} \Pr_{Z \sim \mathcal{N}(0,1)} \left[\frac{-\epsilon \cdot \|\boldsymbol{w}\|_q + \boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \leq Z \leq \frac{\epsilon \cdot \|\boldsymbol{w}\|_q + \boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \right].
\end{aligned}$$

Similarly, we have

$$\Pr_{\boldsymbol{x} \sim \mu_X} \big(\boldsymbol{w}^\top \boldsymbol{x} + \epsilon \cdot \|\boldsymbol{w}\|_q \geq 0 \big) = \frac{1}{2} + \frac{1}{2} \Pr_{Z \sim \mathcal{N}(0,1)} \left[\frac{-\epsilon \cdot \|\boldsymbol{w}\|_q + \boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \leq Z \leq \frac{\epsilon \cdot \|\boldsymbol{w}\|_q + \boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \right].$$

Therefore, we derive the explicit formulation for $\mathrm{RV}_\epsilon(g_{\boldsymbol{w}})$

$$\mathrm{RV}_\epsilon(g_{\boldsymbol{w}}) = \mathrm{H}_2\left(\frac{1}{2}\right) - \mathrm{H}_2\left(\frac{1}{2} - \Pr_{Z \sim \mathcal{N}(0,1)} \left[\frac{\boldsymbol{w}^\top \boldsymbol{\theta}^* - \epsilon \cdot \|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \leq Z \leq \frac{\boldsymbol{w}^\top \boldsymbol{\theta}^* + \epsilon \cdot \|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \right] \right), \tag{A.1}$$

where $\mathrm{H}_2(\cdot)$ is denotes binary entropy function.

Next, given $g_{\boldsymbol{w}} \in \mathcal{G}_{\text{bin}}$, we are going to compute the adversarial gap of $f \circ g_{\boldsymbol{w}}$ for each $h \in \mathcal{H}$. To begin with, we consider the first case $h_1(z) = z$ for any $z \in \mathcal{Z}$. According to the definition of adversarial risk, we have

$$\begin{aligned}
\mathrm{AdvRisk}_\epsilon(h_1 \circ g_{\boldsymbol{w}}) &= \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}} \big[\exists\, \boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon) \text{ s.t. } \mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x}') \neq y \big] \\
&= \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}} \left[\min_{\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon)} y \cdot \boldsymbol{w}^\top \boldsymbol{x}' \leq 0 \right] \\
&= \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}} \left[y \cdot \boldsymbol{w}^\top \boldsymbol{x} \leq - \min_{\boldsymbol{\Delta} \in \mathcal{B}(\mathbf{0}, \epsilon)} \boldsymbol{w}^\top \boldsymbol{\Delta} \right] \\
&= \Pr_{Z \sim \mathcal{N}(0,1)} \left[Z \leq \frac{\epsilon \|\boldsymbol{w}\|_q - \boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \right],
\end{aligned}$$

where the equality is due to the fact that $\mathcal{B}(\mathbf{0}, \epsilon)$ is symmetric with respect to $\mathbf{0}$, and the last equality holds because of the Hölder's inequality: for any $\mathbf{a}, \boldsymbol{b} \in \mathbb{R}^n$, it holds that $\mathbf{a}^\top \boldsymbol{b} \geq -\|\mathbf{a}\|_p \cdot \|\boldsymbol{b}\|_q$ and the equality is achieved when $(a_i / \|\boldsymbol{a}\|_p)^p = (b_i / \|\boldsymbol{b}\|_q)^q$ for any $i \in \{1, 2, \ldots, d\}$.

Similarly, the standard risk can be computed as:

$$\mathrm{Risk}(h_1 \circ g_{\boldsymbol{w}}) = \Pr_{(\boldsymbol{x}, y) \sim \mathcal{D}} \big[\mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x}) \neq y \big] = \Pr_{(\boldsymbol{x}, y) \sim \mathcal{D}} \big[y \cdot \boldsymbol{w}^\top \boldsymbol{x} \leq 0 \big] = \Pr_{Z \sim \mathcal{N}(0,1)} \left[Z \leq \frac{-\boldsymbol{w}^\top \boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}^* \boldsymbol{w}}} \right].$$

Thus, we derive the gap between standard and adversarial risk with respect to $h_1 \circ g_{\boldsymbol{w}}$:

$$\mathrm{AG}_\epsilon(h_1 \circ g_{\boldsymbol{w}}) = \Pr_{Z\sim\mathcal{N}(0,1)}\left[\frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* - \epsilon\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}} \le Z \le \frac{\boldsymbol{w}^\top\boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right].$$

For the other case where $h_2(z) = -z$ for any $z \in \mathcal{Z}$, note that $h_1 \circ g_{\boldsymbol{w}} = h_2 \circ g_{-\boldsymbol{w}}$ for any $g_{\boldsymbol{w}} \in \mathcal{G}_{\text{bin}}$. Thus, a similar proof technique can be applied to compute the adversarial risk,

$$\mathrm{AdvRisk}_\epsilon(h_2 \circ g_{\boldsymbol{w}}) = \Pr_{Z\sim\mathcal{N}(0,1)}\left[Z \le \frac{\epsilon\|\boldsymbol{w}\|_q + \boldsymbol{w}^\top\boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right],$$

and the adversarial gap,

$$\mathrm{AG}_\epsilon(h_2 \circ g_{\boldsymbol{w}}) = \Pr_{Z\sim\mathcal{N}(0,1)}\left[\frac{\boldsymbol{w}^\top\boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}} \le Z \le \frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* + \epsilon\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right].$$

Note that $f^* : \mathcal{X} \to \mathcal{Y}$ is the optimal classifier based on $g_{\boldsymbol{w}}$ that minimizes the adversarial risk of $h \circ g_{\boldsymbol{w}}$ for any $h \in \mathcal{H}$. By comparing the adversarial risk of $h_1 \circ g_{\boldsymbol{w}}$ and $h_2 \circ g_{\boldsymbol{w}}$, we have $f^* = h_1 \circ g_{\boldsymbol{w}}$, if $\boldsymbol{w}^\top\boldsymbol{\theta}^* \ge 0$; $f^* = h_2 \circ g_{\boldsymbol{w}}$, otherwise. Thus, we derive the adversarial gap with respect to f^* as follows

$$\mathrm{AG}_\epsilon(f^*) = \begin{cases} \Pr_{Z\sim\mathcal{N}(0,1)}\left(\frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* - \epsilon\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}} \le Z \le \frac{\boldsymbol{w}^\top\boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right), & \text{if } \boldsymbol{w}^\top\boldsymbol{\theta}^* \ge 0; \\ \Pr_{Z\sim\mathcal{N}(0,1)}\left(\frac{\boldsymbol{w}^\top\boldsymbol{\theta}^*}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}} \le Z \le \frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* + \epsilon\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right), & \text{otherwise.} \end{cases} \tag{A.2}$$

Based on (A.2), we further obtain the following inequality

$$\mathrm{AG}_\epsilon(f^*) \le \Pr_{Z\sim\mathcal{N}(0,1)}\left[\frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* - \epsilon\cdot\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}} \le Z \le \frac{\boldsymbol{w}^\top\boldsymbol{\theta}^* + \epsilon\cdot\|\boldsymbol{w}\|_q}{\sqrt{\boldsymbol{w}^\top\boldsymbol{\Sigma}^*\boldsymbol{w}}}\right] \le 2\cdot\mathrm{AG}_\epsilon(f^*).$$

Finally, according to the formulation of representation vulnerability (A.1), we have

$$\int_{\frac{1}{2}-\mathrm{AG}_\epsilon(f^*\circ g_{\boldsymbol{w}})}^{\frac{1}{2}} \mathrm{H}_2'(\theta)d\theta \le \mathrm{RV}_\epsilon(g_{\boldsymbol{w}}) \le \int_{\frac{1}{2}-\frac{1}{2}\mathrm{AG}_\epsilon(f^*\circ g_{\boldsymbol{w}})}^{\frac{1}{2}} \mathrm{H}_2'(\theta)d\theta,$$

which completes the proof.

□

A.2. Proof of Lemma 3.3

Lemma 3.3, restated below, connects adversarial risk and input distribution perturbations bounded in an ∞-Wasserstein ball.

Lemma 3.3. Let $(\mathcal{X}, \Delta)$ be the input metric space and $\mathcal{Y}$ be the set of labels. Assume all the examples are generated from a joint probability distribution $(X, Y) \sim \mu_{XY}$. Let μ_X be the marginal distribution of X. Then, for any classifier $f : \mathcal{X} \to \mathcal{Y}$ and $\epsilon \ge 0$, we have

$$\mathrm{AdvRisk}_\epsilon(f) = \sup_{\mu_{X'}\in\mathcal{B}_{\mathrm{W}_\infty}(\mu_X,\epsilon)} \Pr\big[f(X') \ne Y\big],$$

where X' denotes the random variable that follows $\mu_{X'}$.

Proof. Our proof proves the equality by proving $\le$ inequalities in both directions. First, we prove

$$\mathrm{AdvRisk}_\epsilon(f) \le \sup_{\mu_{X'}\in\mathcal{B}_{\mathrm{W}_\infty}(\mu_X,\epsilon)} \Pr\big[f(X') \ne Y\big]. \tag{A.3}$$

For any classifier $f : \mathcal{X} \to \mathcal{Y}$, according to Definition 2.3, we have

$$\mathrm{AdvRisk}_\epsilon(f) = \Pr_{(\boldsymbol{x},y)\sim\mu_{XY}}\big[\exists\, \boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x},\epsilon) \text{ s.t. } f(\boldsymbol{x}') \ne y\big].$$

Since f is a given deterministic function, the optimal perturbation scheme that achieves $\mathrm{AdvRisk}_\epsilon(f)$ essentially defines a transport map $T : \mathcal{X} \to \mathcal{X}$. More specifically, let $\mathcal{C}_y(f) = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \neq y\}$. Then, for any sampled pair $(\boldsymbol{x}, y) \sim \mu_{XY}$, we can construct T such that

$$T(\boldsymbol{x}) = \begin{cases} \operatorname{argmin}_{\boldsymbol{x}' \in \mathcal{C}_y(f)} \Delta(\boldsymbol{x}', \boldsymbol{x}), & \text{if } \mathcal{C}_y(f) \cap \mathcal{B}(\boldsymbol{x}, \epsilon) \neq \varnothing; \\ \boldsymbol{x}, & \text{otherwise.} \end{cases}$$

Let (X, Y) be the random variable that follows μ_{XY}. By construction, it can be easily verified that $T_\#(\mu_X) \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)$ and $\mathrm{AdvRisk}_\epsilon(f) = \Pr\left[f(T(X)) \neq Y\right]$. Therefore, we have proven (A.3).

It remains to prove the other direction of the inequality:

$$\mathrm{AdvRisk}_\epsilon(f) \geq \sup_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \Pr\left[f(X') \neq Y\right]. \tag{A.4}$$

According to the alternative definition of ∞-Wasserstein distance, the optimal solution $\mu^*_{X'}$ that achieves the supremum of the right hand side of (A.4) can be captured by a transport map $T^* : \mathcal{X} \to \mathcal{X}$ such that $\mu^*_{X'} = T^*_\#(\mu_X)$ and $\Delta(T^*(X), X) \leq \epsilon$ holds almost surely with respect to the randomness of X and T^*. Thus, we have

$$\begin{aligned} \Pr\left[f(T^*(X)) \neq Y\right] &= \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}}\left[f(T^*(\boldsymbol{x})) \neq y\right] \\ &= \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}}\left[\Delta(T^*(\boldsymbol{x}), \boldsymbol{x}) \leq \epsilon \text{ and } f(T^*(\boldsymbol{x})) \neq y\right] \\ &\leq 1 - \Pr_{(\boldsymbol{x}, y) \sim \mu_{XY}}\left[\forall\, \boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon) \text{ s.t. } f(\boldsymbol{x}') = y\right] = \mathrm{AdvRisk}_\epsilon(f). \end{aligned}$$

Therefore, we have proven the second direction and completed the proof. □

A.3. Proof of Theorem 3.4

Theorem 3.4, restated below, gives a lower bound for the adversarial risk for any downstream classifier in terms of the worst-case mutual information between the representation's input and output distributions.

Theorem 3.4. Let $(\mathcal{X}, \Delta)$ be the input metric space, $\mathcal{Y}$ be the set of labels and μ_{XY} be the underlying joint probability distribution. Assume the marginal distribution of labels μ_Y is a uniform distribution over $\mathcal{Y}$. Consider the feature space $\mathcal{Z}$ and the set of downstream classifiers $\mathcal{H} = \{h : \mathcal{Z} \to \mathcal{Y}\}$. Given $\epsilon \geq 0$, for any $g : \mathcal{X} \to \mathcal{Z}$, we have

$$\inf_{h \in \mathcal{H}} \mathrm{AdvRisk}_\epsilon(h \circ g) \geq 1 - \frac{\mathrm{I}(X; Z) - \mathrm{RV}_\epsilon(g) + \log 2}{\log |\mathcal{Y}|},$$

where X is the random variable that follows the marginal distribution of inputs μ_X and $Z = g(X)$.

Before starting the proof, we state two useful lemmas on Markov chains. A Markov chain is defined to be a collection of random variables $\{X_t\}_{t \in \mathbb{Z}}$ with the property that given the present, the future is conditionally independent of the past. Namely,

$$\Pr(X_t = j | X_0 = i_0, X_1 = i_1, ..., X_{(t-1)} = i_{(t-1)}) = \Pr(X_t = j | X_{(t-1)} = i_{(t-1)}).$$

Lemma A.1 (Fano's Inequality)**.** Let X be a random variable uniformly distributed over a finite set of outcomes $\mathcal{X}$. For any estimator $\widehat{X}$ such that $X \to Y \to \widehat{X}$ forms a Markov chain, we have

$$\Pr(\widehat{X} \neq X) \geq 1 - \frac{\mathrm{I}(X; \widehat{X}) - \log 2}{\log |\mathcal{X}|}.$$

Lemma A.2 (Data-Processing Inequality)**.** For any Markov chain $X \to Y \to Z$, we have

$$\mathrm{I}(X; Y) \geq I(X; Z) \quad \text{and} \quad \mathrm{I}(Y; Z) \geq \mathrm{I}(X; Z).$$

Chapter 2 in Cover & Thomas (2012) provides proofs of Lemmas A.1 and A.2.

Proof of Theorem 3.4. For any classifier $h : \mathcal{Z} \to \mathcal{Y}$, according to Lemma 3.3, we have

$$\mathrm{AdvRisk}_\epsilon(h \circ g) = \sup_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \Pr\big[h(g(X')) \neq Y\big]. \tag{A.5}$$

Let $\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)$ be a probability measure over $(\mathcal{X}, \Delta)$. According to the alternative definition of ∞-Wasserstein distance using optimal transport, $\mu_{X'}$ corresponds to a transport map $T : \mathcal{X} \to \mathcal{X}$ such that $\mu_{X'} = T_\#(\mu_X)$. Thus, for any given $\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)$ and $h \in \mathcal{H}$, we have the Markov chain

$$Y \to X \xrightarrow{T} X' \xrightarrow{g} g(X') \xrightarrow{h} (h \circ g)(X').$$

where X, Y are random variables for input and label distributions respectively. The first Markov chain $Y \to X$ can be understood as a generative model for generating inputs according to the conditional probability distribution $\mu_{X|Y}$. Therefore, applying Lemmas A.1 and A.2, we obtain the inequality,

$$\Pr\big[h(g(X')) \neq Y\big] \geq 1 - \frac{\mathrm{I}\big(Y; (h \circ g)(X')\big) + \log 2}{\log |\mathcal{Y}|} \geq 1 - \frac{\mathrm{I}\big(X'; g(X')\big) + \log 2}{\log |\mathcal{Y}|}. \tag{A.6}$$

Taking the supremum over the distribution of X' in $\mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)$ and infimum over $h \in \mathcal{H}$ on both sides of (A.6) yields

$$\begin{aligned}
\inf_{h \in \mathcal{H}} \big[\mathrm{AdvRisk}_\epsilon(h \circ g)\big] &= \inf_{h \in \mathcal{H}} \sup_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \Pr\big[h(g(X')) \neq Y\big] \\
&\geq 1 - \frac{\inf_{\mu_{X'} \in \mathcal{B}_{\mathrm{W}_\infty}(\mu_X, \epsilon)} \mathrm{I}\big(X'; g(X')\big) + \log 2}{\log |\mathcal{Y}|} \\
&= 1 - \frac{\mathrm{I}(X; g(X)) - \mathrm{RV}_\epsilon(g) + \log 2}{\log |\mathcal{Y}|},
\end{aligned}$$

where the first equality is due to (A.5) and the inequality holds because of (A.6). Thus, we completed the proof. □

B. Algorithm for Estimating the Worst-Case Mutual Information

This section presents the pseudocode of our heuristic algorithm for solving the empirical estimation problem (4.3). More specifically, given a training sample set $\mathcal{S}_{\text{train}}$, our algorithm alternatively optimizes for the worst-case input perturbations using projected gradient descent (Algorithm 1) and conducts gradient ascent for the network parameters θ (training phase in Algorithm 2). Based on the best parameter θ_{opt} selected from the training phase, our algorithm then estimates the worst-case mutual information with respect to the given representation g using a testing sample set $\mathcal{S}_{\text{test}}$ (testing phase in Algorithm 2). Since we only have assess to a finite set of data sampled from μ_X, we use an additional testing phase in Algorithm 2 to minimize the overfitting effect of the training samples on the optimal network parameter θ_{opt} for mutual information neural estimation (MINE).

Moreover, we adopt the negative sampling scheme (Hjelm et al., 2018) to estimate the expectation term with respect to $\widehat{\mu}_X^{(m)} \otimes \widehat{\mu}_Z^{(m)}$ in mutual information neural estimation for better performance. Here, the pairing scheme defines a correspondence from each input to a set of inputs for a given sample set. To be more specific, given a set of samples $\{\boldsymbol{x}_i\}_{i \in [B]}$, a pairing scheme with negative sampling size $N \leq B$ corresponds to a set of vectors $\{\boldsymbol{\pi}_i\}_{i \in [B]}$ such that each $\boldsymbol{\pi}_i$ is a randomly selected subset from $\{1, 2, \ldots, B\}$ with size N, and π_{ij} denotes the j-th element of $\boldsymbol{\pi}_i$. Compared with the algorithm in Hjelm et al. (2018) for estimating standard mutual information, Algorithm 2 requires additional $B \cdot S$ steps of forward and backward propagations with respect to the input for finding the worst-case input perturbations in each iteration.

C. Worst-case Mutual Information for Individual Neuron Features

The following tensorization inequality (Scarlett & Cevher, 2019) characterizes the connection between the mutual information of individual neuron features and that of the whole representation. According to Theorem 3.4, such connection suggests the necessity of enough worst-case mutual information for each individual neuron.

Algorithm 1 Heuristic Search for Worst-Case Input Perturbations

Input: samples $\{\boldsymbol{x}_i\}_{i\in[B]}$, representation g, MINE estimator T_θ, paring scheme $\{\boldsymbol{\pi}_i\}_{i\in[B]}$, perturbation strength ϵ in ℓ_p
Hyperparameters: negative sampling size N, number of iterations S, step size η_a

Initialize $\{\boldsymbol{x}'_i\}_{i\in[B]} \leftarrow \{\boldsymbol{x}_i\}_{i\in[B]}$
for $s = 1, 2, \ldots, S$ **do**
  $J(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_B, \theta) \leftarrow \frac{1}{B}\sum_{i=1}^{B} T_\theta\big(\boldsymbol{x}'_i, g(\boldsymbol{x}'_i)\big) - \log\big(\frac{1}{BN}\sum_{i=1}^{B}\sum_{j=1}^{N}\exp\big[T_\theta\big(\boldsymbol{x}'_i, g(\boldsymbol{x}'_{\pi_{ij}})\big)\big]\big)$
  for $i = 1, 2, \ldots, B$ **do**
    $\boldsymbol{x}'_i \leftarrow \mathcal{P}_{\mathcal{B}(\boldsymbol{x}_i,\epsilon)}\big[\boldsymbol{x}'_i - \eta_a \cdot \nabla_{\boldsymbol{x}'_i} J(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_B, \theta)\big]$ // $\mathcal{P}_{\mathcal{B}(\boldsymbol{x}_i,\epsilon)}$ denotes the projection operator onto $\mathcal{B}(\boldsymbol{x}_i, \epsilon)$
  end for
end for
$V_1 \leftarrow J(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_B, \theta)$
$V_2 \leftarrow \nabla_\theta J(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_B, \theta)$

Output: $\{V_1, V_2\}$

Algorithm 2 Empirical Estimation of Worst-Case Mutual Information

Input: training and testing sample sets $(\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{test}})$ sampled from μ_X, representation g, perturbation strength ϵ in ℓ_p
Hyperparameters: number of training epochs T, step size η_e, number of testing mini-batches K

// **Training Phase**
$\theta_1 \leftarrow$ initialize network parameter for MINE estimator
for $t = 1, 2, \ldots, T$ **do**
  $\{\boldsymbol{x}_i\}_{i\in[B]}, \{\boldsymbol{\pi}_i\}_{i\in[B]} \leftarrow$ randomly generate a batch of B training samples and a pairing scheme
  $\{V_1(t), V_2(t)\} \leftarrow$ Algorithm 1$\big(\{\boldsymbol{x}_i\}_{i\in[B]}, g, T_{\theta_t}, \{\boldsymbol{\pi}_i\}_{i\in[B]}, \epsilon\big)$
  $\theta_{t+1} \leftarrow \theta_t + \eta_e \cdot V_2(t)$
end for
$\theta_{\text{opt}} \leftarrow \operatorname{argmax}\big\{t \in [T] : V_1(t)\big\}$ // choose the best parameter θ_{opt} based on history
// **Testing Phase**
Randomly split the testing set $\mathcal{S}_{\text{test}}$ into K mini-batches $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ with equal size
for $k = 1, 2, \ldots, K$ **do**
  $\{\boldsymbol{\pi}(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{S}_k} \leftarrow$ randomly generate a pairing scheme with respect to $\mathcal{S}_k$
  $\{V_1(k), V_2(k)\} \leftarrow$ Algorithm 1$\big(\mathcal{S}_k, g, T_{\theta_{\text{opt}}}, \{\boldsymbol{\pi}(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{S}_k}, \epsilon\big)$
end for
$\widehat{I}_{\text{worst}} \leftarrow \frac{1}{K}\sum_{k=1}^{K} V_1(k)$

Output: $\widehat{I}_{\text{worst}}$

Lemma C.1 (Tensorization of Mutual Information)**.** Let $\boldsymbol{Z} = (Z_1, ..., Z_n)$ be a product distributions over random variables. If Z_1, ..., Z_n are mutually independent conditioned on X, then

$$I(X; \boldsymbol{Z}) \leq \sum_{i=1}^{n} I(X; Z_i)$$

Suppose neurons within a single layer have no interconnection, then each neuron's output is mutually independent conditioned on the model input. If a perturbation imposed on the input distribution makes the perturbed mutual information $I(X'; Z'_i)$ relatively low for each neuron, then the perturbed mutual information with respect to the entire layer $I(X'; \boldsymbol{Z}')$ will also be low, which further implies a low adversarial accuracy for any downstream classifier based on Theorem 3.4.

D. Experiments

D.1. Implementation Details

Here, we provide additional implementation details of our experiments presented in Section 6.

Model architectures. For all experiments, we follow Hjelm et al. (2018) in implementing the MINE estimator. We adopt the *encode-and-dot-product* model architecture in Hjelm et al. (2018) which maps $\boldsymbol{x}$ and $\boldsymbol{z}$ respectively to two high-dimensional vectors and then takes the dot-product to calculate the output. The basic modules used in our experiments are listed in Table 2. A slight difference in training the feature (encoder) is that Hjelm et al. (2018) shares parameters between parts of the mutual information estimator and the encoder, while we separate the two parts completely to be consistent with our mutual information estimation experiments.

Module	Structure
Feature Extractor	$\text{Conv}(64, 4\times4, 2) \to \text{Conv}(128, 4\times4, 2) \to \text{Conv}(256, 4\times4, 2) \to \text{FC}(1024) \to \text{FC}(64)$
Top Classifier (MLP)	$\text{FC}(200) \to \text{FC}(10)$
Top Classifier (Linear)	$\text{FC}(10)$
Baseline-H	Feature Extractor $\to$ Top Classifier (MLP)
Estimator Part 1	$\text{Conv}(64, 4\times4, 2) \to \text{Conv}(128, 4\times4, 2) \to \text{Conv}(256, 4\times4, 2)$
Estimator Part 2	$\text{Conv}(2048, 1\times1, 1) \to \text{Conv}(2048, 1\times1, 1)$
Estimator	$(\boldsymbol{x} \to \text{Estimator Part 1} \to \text{Estimator Part 2}) \cdot (\boldsymbol{z} \to \text{Estimator Part 2})$

Table 2. Basic model structures used in our experiments. Batch-normalization and ReLU activation are used between layers (not including the output of each module). Shortcut-connection is omitted for Estimator Part 2. For scalar feature z, Estimator Part 2 is replaced by an identity mapping. Average operation is needed in the dot-product operation of Estimator. For more details, see Hjelm et al. (2018)

Hyperparameters. We use simple hyperparameter settings to control their effect on our various ablation experiments. We use l_∞ constrained perturbations and PGD attack (Mądry et al., 2018) on all datasets. For CIFAR-10, we set the radius $\epsilon = 8/255$ and use 7 attack steps with step size 0.01. For MNIST, we set the radius $\epsilon = 0.3$ and use 10 attack steps with step size 0.1. For Fashion-MNIST, we set the radius $\epsilon = 0.1$ and use 10 attack steps with step size 0.02. For SVHN, we set the radius $\epsilon = 4/255$ and use 10 attack steps with step size 0.005. The batch size is set as 128 for both datasets, and our results are consistent with different batch sizes between 128 to 512 (we did not test other sizes). A total training epochs of 200 is set for VGG, ResNet, and DenseNet, with an initial learning rate of 0.1 which decays by a factor of 10 every 50 epochs. For the Baseline-H model and the similar mutual information estimator, we set the training epoch to 300 and use a fixed learning rate of 0.0001 as in Hjelm et al. (2018).

D.2. Additional Results

Results for MNIST, Fashion-MNIST, and SVHN. We present the downstream classification results for MNIST, Fashion-MNIST, and SVHN in Table 3, 4, 5. These results support similar conclusions as those drawn from CIFAR-10 dataset in Table 1. That is, our training principle always produces representations that have significantly better adversarial accuracy for downstream adversarial classification. In many cases, our training principle also produces representations that have better natural accuracy, despite the worst-case situation that our training principle considers.

Saliency maps of internal features. In section 6.1, we evaluated the internal feature vulnerability of all the convolutional kernels in the second layer of Baseline-H. Here, we further visualize the saliency maps of the those internal features to evaluate the underlying correlations. As shown in Figure 5, features in robust model have less noisy saliency maps, which is consistent with the observations of lower representation vulnerability shown in Figure 3.

Saliency maps of learned representations. More comparison results of saliency maps are given in Figure 6. The saliency maps of representations learned using our unsupervised training method shows comparable interpretability results to the models learned using fully-supervised adversarial training. Saliency maps computed by different losses also show consistent interpretability results. This indicates that our training principle indeed produces adversarially robust representations.

Representation (g)	Classifier (h)	MLP h		Linear h	
		Natural	Adversarial	Natural	Adversarial
Hjelm et al. (2018)	Standard	$\mathbf{96.96 \pm 0.35}$	0.00 ± 0.00	$\mathbf{84.88 \pm 1.11}$	0.00 ± 0.00
Hjelm et al. (2018)	Robust	44.99 ± 14.49	16.70 ± 2.22	11.35 ± 0.00	11.35 ± 0.00
Ours	Standard	96.67 ± 0.12	9.97 ± 1.88	82.10 ± 0.37	3.69 ± 0.55
Ours	Standard (E.S.)	94.68 ± 0.93	12.79 ± 2.24	77.72 ± 2.47	4.73 ± 1.29
Ours	Robust	95.05 ± 0.19	$\mathbf{60.64 \pm 1.82}$	73.99 ± 1.16	$\mathbf{30.55 \pm 1.34}$
Fully-Supervised	Standard	99.13 ± 0.23	0.45 ± 0.33	99.13 ± 0.04	0.00 ± 0.00
Fully-Supervised	Robust	99.25 ± 0.05	95.73 ± 0.09	99.21 ± 0.06	95.29 ± 0.18

Table 3. Comparisons of different representation learning methods for downstream classification on MNIST. *E.S.* denotes early stopping under the criterion of the best adversarial accuracy. We present the mean accuracy and the standard deviation over 4 repeated trials.

Representation (g)	Classifier (h)	MLP h		Linear h	
		Natural	Adversarial	Natural	Adversarial
Hjelm et al. (2018)	Standard	89.58 ± 0.13	0.00 ± 0.00	85.93 ± 0.26	0.00 ± 0.00
Hjelm et al. (2018)	Robust	48.61 ± 4.96	14.95 ± 0.79	10.00 ± 0.00	10.00 ± 0.00
Ours	Standard	$\mathbf{90.45 \pm 0.19}$	5.38 ± 1.00	$\mathbf{87.37 \pm 0.10}$	18.20 ± 2.87
Ours	Standard (E.S.)	81.66 ± 0.18	29.71 ± 2.00	86.27 ± 0.64	23.40 ± 2.65
Ours	Robust	84.31 ± 0.29	$\mathbf{70.44 \pm 3.62}$	81.05 ± 0.30	$\mathbf{61.33 \pm 0.49}$
Fully-Supervised	Standard	92.09 ± 0.23	0.00 ± 0.00	85.93 ± 0.26	0.00 ± 0.00
Fully-Supervised	Robust	87.94 ± 0.18	77.59 ± 0.38	88.05 ± 0.46	77.15 ± 0.24

Table 4. Comparisons of different representation learning methods for downstream classification on Fashion-MNIST. *E.S.* denotes early stopping under the criterion of the best adversarial accuracy. We present the mean accuracy and the standard deviation over 4 repeated trials.

Representation (g)	Classifier (h)	MLP h		Linear h	
		Natural	Adversarial	Natural	Adversarial
Hjelm et al. (2018)	Standard	50.15 ± 0.89	0.00 ± 0.00	38.94 ± 1.52	0.00 ± 0.00
Hjelm et al. (2018)	Robust	19.59 ± 0.00	19.59 ± 0.00	19.59 ± 0.00	19.59 ± 0.00
Ours	Standard	$\mathbf{74.32 \pm 0.49}$	26.29 ± 1.41	$\mathbf{58.37 \pm 0.54}$	21.62 ± 0.91
Ours	Standard (E.S.)	71.85 ± 0.59	29.59 ± 0.83	54.76 ± 0.86	25.00 ± 0.53
Ours	Robust	68.25 ± 0.83	$\mathbf{40.23 \pm 0.83}$	49.04 ± 0.79	$\mathbf{30.56 \pm 0.38}$
Fully-Supervised	Standard	91.97 ± 0.13	9.77 ± 1.58	91.33 ± 0.15	9.29 ± 1.73
Fully-Supervised	Robust	90.14 ± 0.83	65.35 ± 0.44	89.60 ± 0.54	64.48 ± 1.06

Table 5. Comparisons of different representation learning methods for downstream classification on SVHN. *E.S.* denotes early stopping under the criterion of the best adversarial accuracy. We present the mean accuracy and the standard deviation over 4 repeated trials.

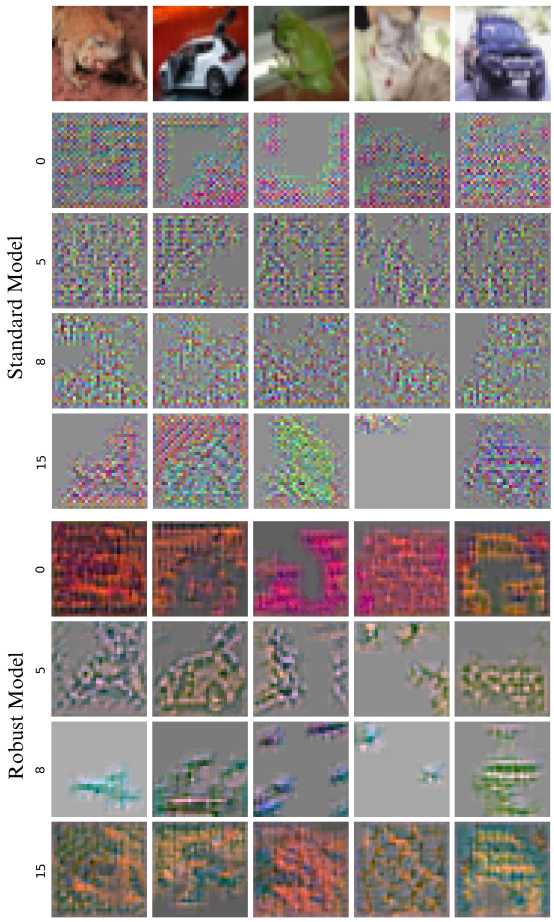

Figure 5. Saliency maps of four arbitrarily selected features in the second convolutional layer of Baseline-H. The feature saliency map is computed by the gradient of a kernel's averaged activation over a input image. Each row presents saliency maps of a specific convolutional kernel.

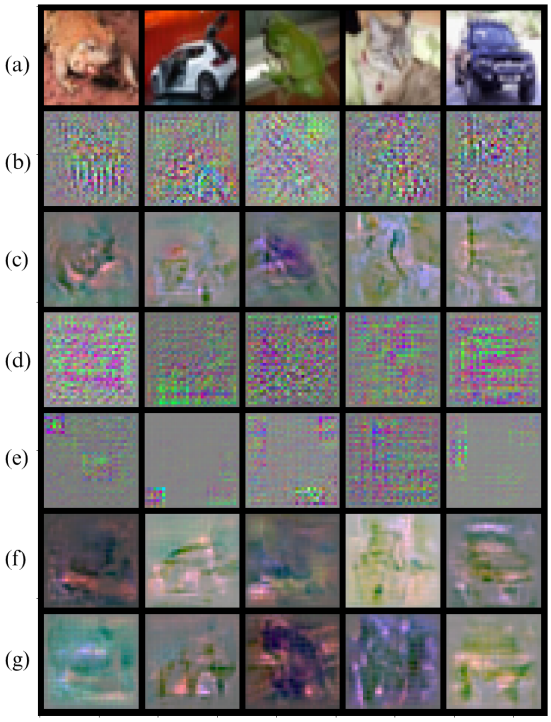

Figure 6. Saliency maps of different models on CIFAR-10: (a) original images (b) fully-supervised standard model (c) fully-supervised robust model (d) representations learned using Hjelm et al. (2018) with cross-entropy loss (e) representations learned using Hjelm et al. (2018) with mutual information maximization loss (f) representations learned using our method with cross-entropy loss (g) representations learned using our method with mutual information maximization loss.