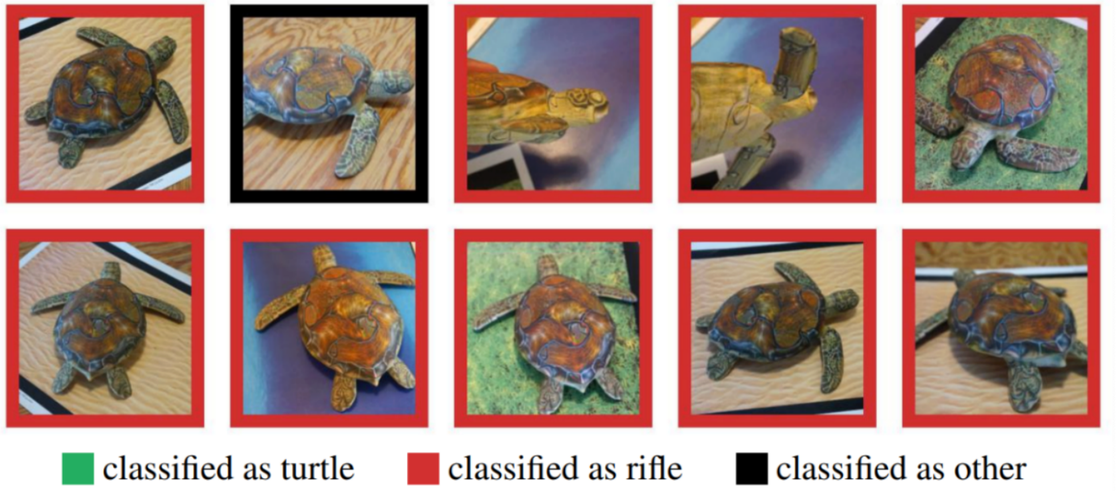
- **Adversarial пример** — вектор, подающийся на вход алгоритму, на котором алгоритм выдает некорректный выход.
- **Adversarial атака** — алгоритм действий, целью которого является получение Adversarial примера.



- Fast Sign Gradient Method

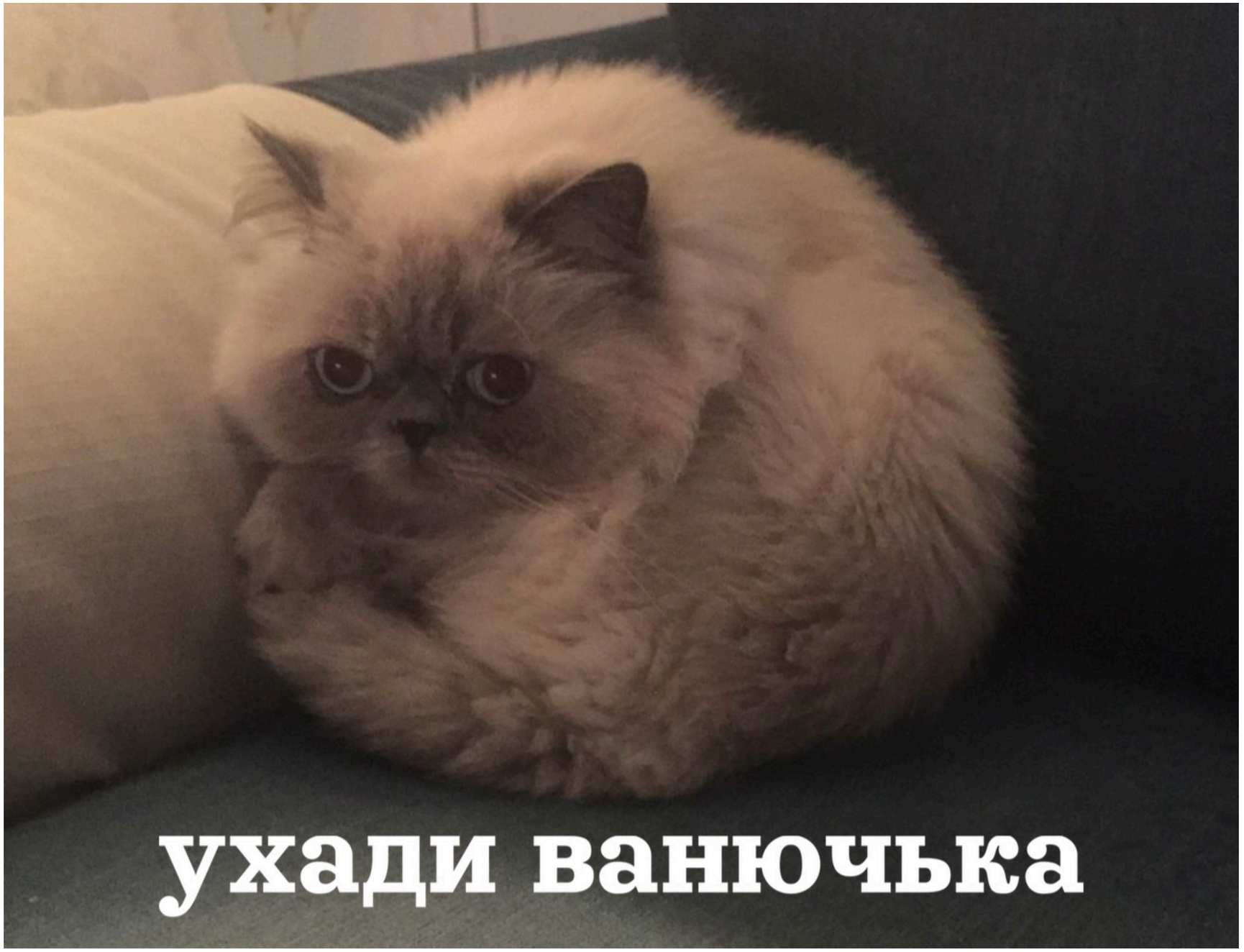
$$X' = X + \epsilon * sign(\nabla_x J(X, ytrue))$$

- DeepFool атака (про минимальную шумовую карту)
- GAN-based методы (встраивание BlackBox модели в архитектуру генеративно-состязательной сети)

Adversarial Training

- Добавлять adversarial примеры в обучающую выборку
- Добавлять немного зашумленные объекты обучающей выборки
- Label Smoothing
- Построение ансамблей
- Feature squeezing

Немного про тестирование


ухади ванючька

Adversarial Examples Are Not Bugs, They Are Features

Setup

- binary classification $(x, y) \in \mathcal{X} \times \{\pm 1\}$ $C : \mathcal{X} \to \{\pm 1\}$
- $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$
- $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)] = 0$
- $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)^2] = 1)$

Categorize features

- ***ρ*-useful features** if it is correlated with the true label in expectation

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot f(x)] \geq \rho.$$

- ***γ*-robustly useful features** if *ρ*-useful feature remains *γ*-usefulunder adversarial perturbation

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\inf_{\delta\in\Delta(x)} y \cdot f(x+\delta)\right] \geq \gamma$$

- **Useful, non-robust features** is a feature which is *ρ*-useful for some *ρ* bounded away from zero, but is not a *γ*-robust feature for any $\gamma \geq 0$

Classification

$$C(x) = \operatorname{sgn}\left(b + \sum_{f \in F} w_f \cdot f(x)\right)$$

Standard Training

- performed by minimizing a loss function (via *empirical risk minimization* (ERM)) that decreases with the correlation between the weighted combination of the features and the label

.

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}_\theta(x,y)\right] = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y\cdot\left(b+\sum_{f\in F} w_f\cdot f(x)\right)\right]$$

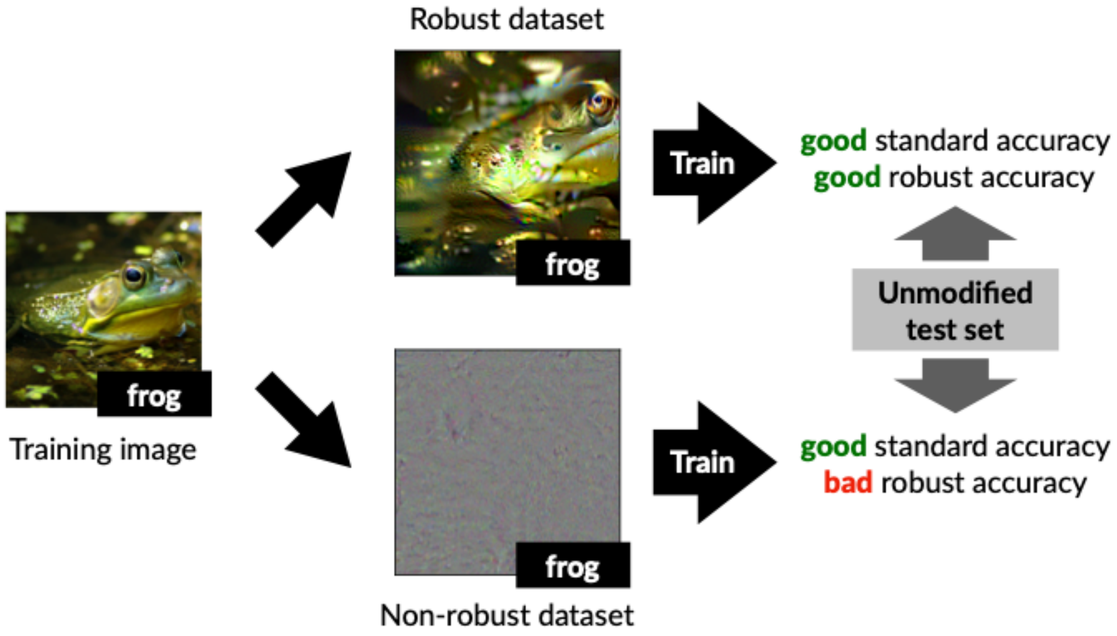
Robust training

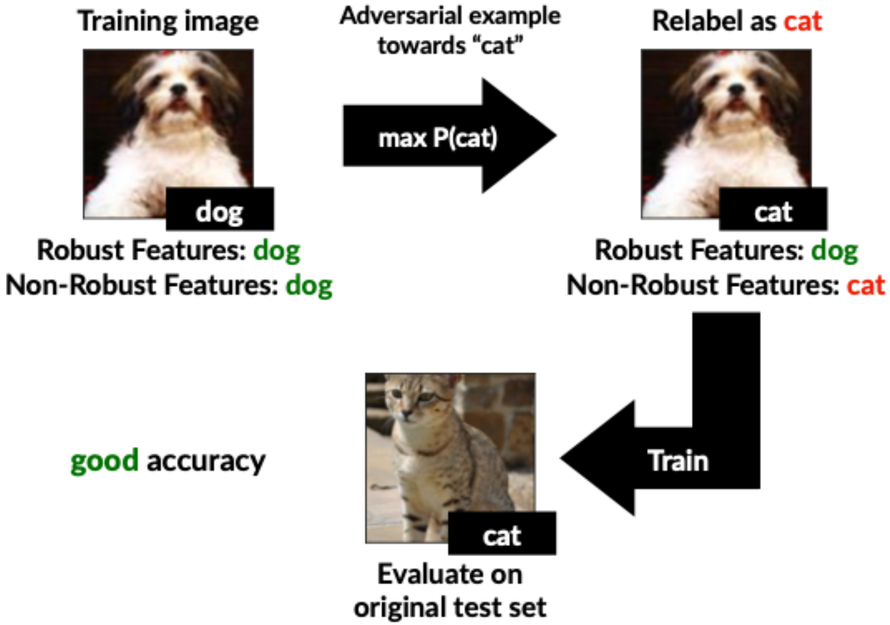
- use an *adversarial* loss function that can discern between robust and non-robust features:

.

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(x)}\mathcal{L}_\theta(x+\delta,y)\right]$$

Finding Robust (and Non-Robust) Features





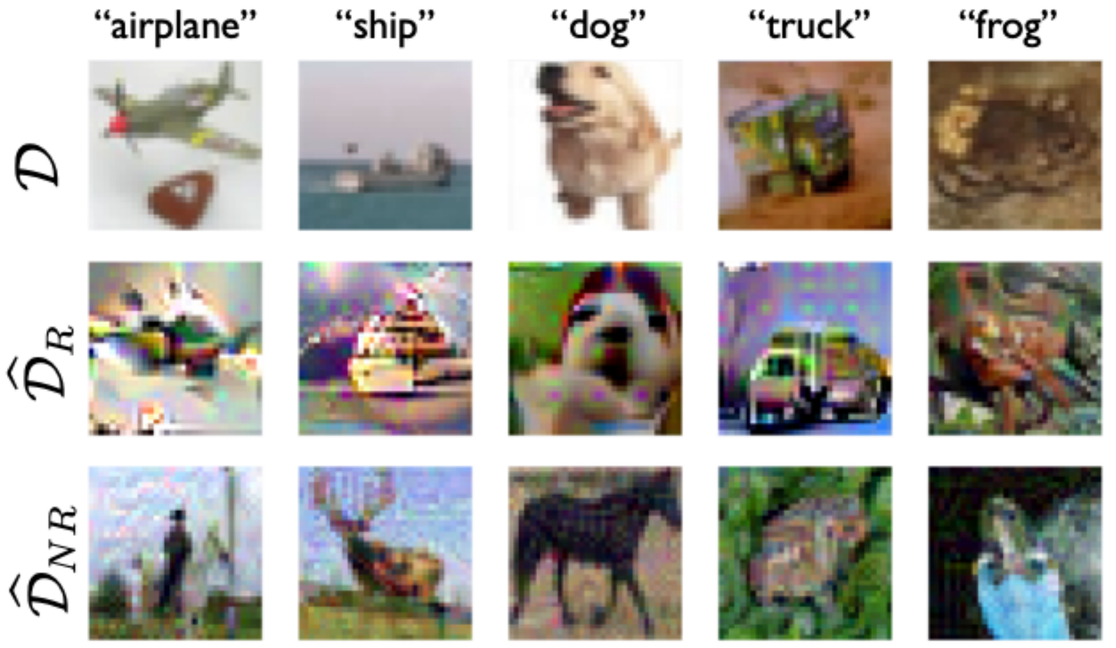
Disentangling robust and non-robust features

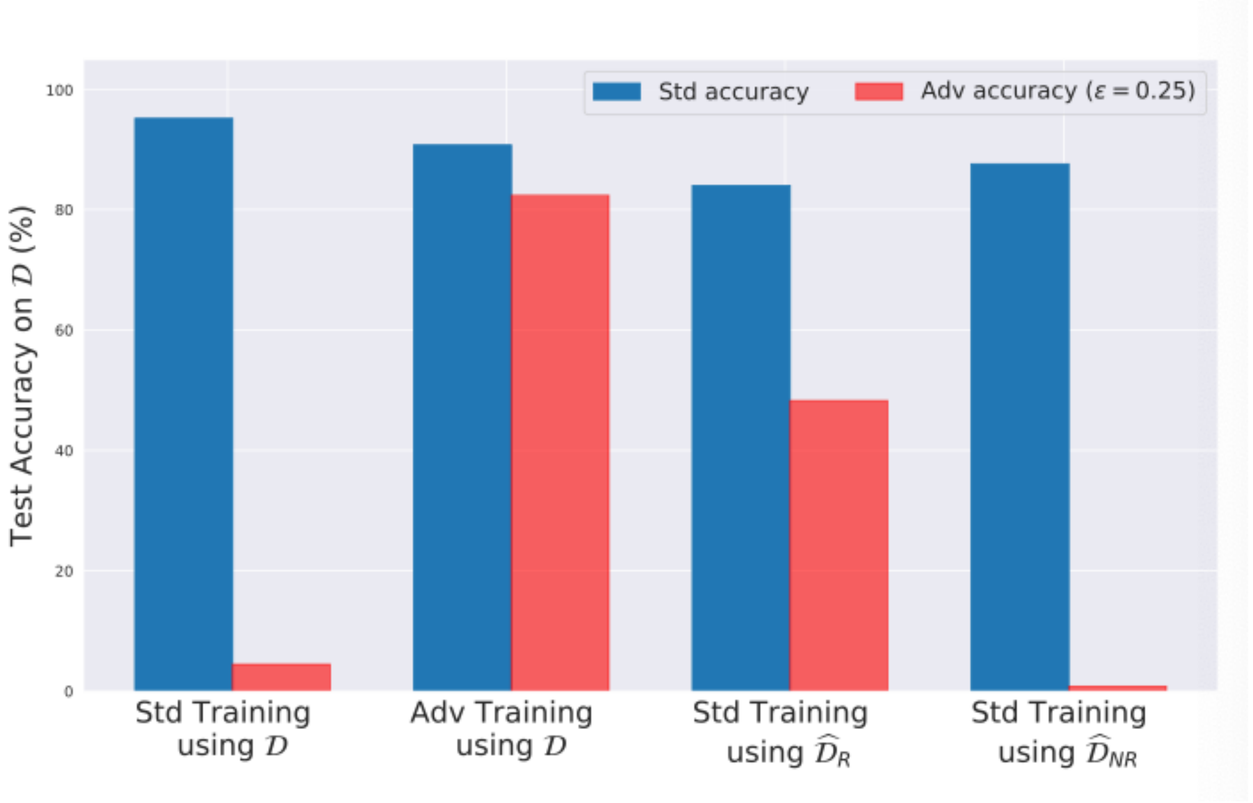
- Хочу: гарантировать, что полезны только надежные функции, тогда бы обучение привело к созданию надежного классификатора
- Могу: использовать надежную модель и модифицировать набор данных, чтобы он содержал только те функции, которые имеют отношение к этой модели

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_R}\left[f(x)\cdot y\right] = \begin{cases} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f(x)\cdot y\right] & \text{if } f \in F_C \\ 0 & \text{otherwise,} \end{cases}$$

GETROBUSTDATASET(D)

1. $C_R \leftarrow$ ADVERSARIALTRAINING(D)
   $g_R \leftarrow$ mapping learned by C_R from the input to the representation layer
2. $D_R \leftarrow \{\}$
3. For $(x, y) \in D$
   $x' \sim D$
   $x_R \leftarrow \arg\min_{z\in[0,1]^d} \|g_R(z) - g_R(x)\|_2$ # Solved using ℓ_2-PGD starting from x'
   $D_R \leftarrow D_R \cup \{(x_R, y)\}$
4. Return D_R


"airplane"
"ship"
"dog"
"truck"
"frog"
$\mathcal{D}$
$\widehat{\mathcal{D}}_R$
$\widehat{\mathcal{D}}_{NR}$


Std accuracy
Adv accuracy ($\varepsilon = 0.25$)
Test Accuracy on $\mathcal{D}$ (%)
0
20
40
60
80
100
Std Training using $\mathcal{D}$
Adv Training using $\mathcal{D}$
Std Training using $\widehat{\mathcal{D}}_R$
Std Training using $\widehat{\mathcal{D}}_{NR}$

Non-robust features suffice for standard classification

GETNONROBUSTDATASET(D, ε)

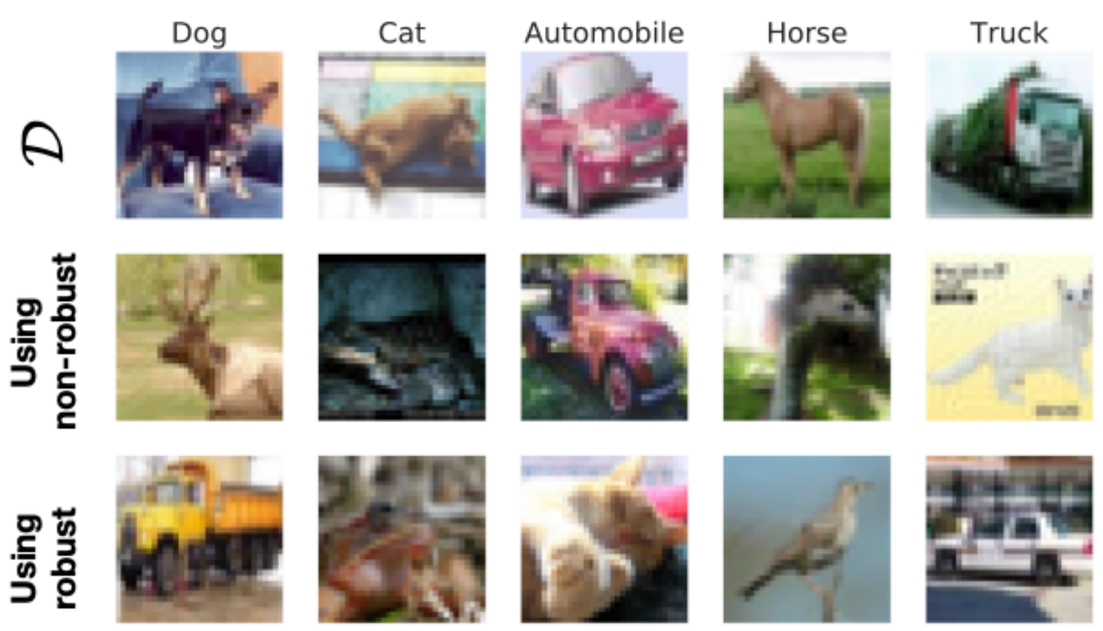
1. $D_{NR} \leftarrow \{\}$
2. $C \leftarrow$ STANDARDTRAINING(D)
3. For $(x, y) \in D$

   $t \overset{\text{uar}}{\sim} [C]$ # or $t \leftarrow (y+1) \mod C$

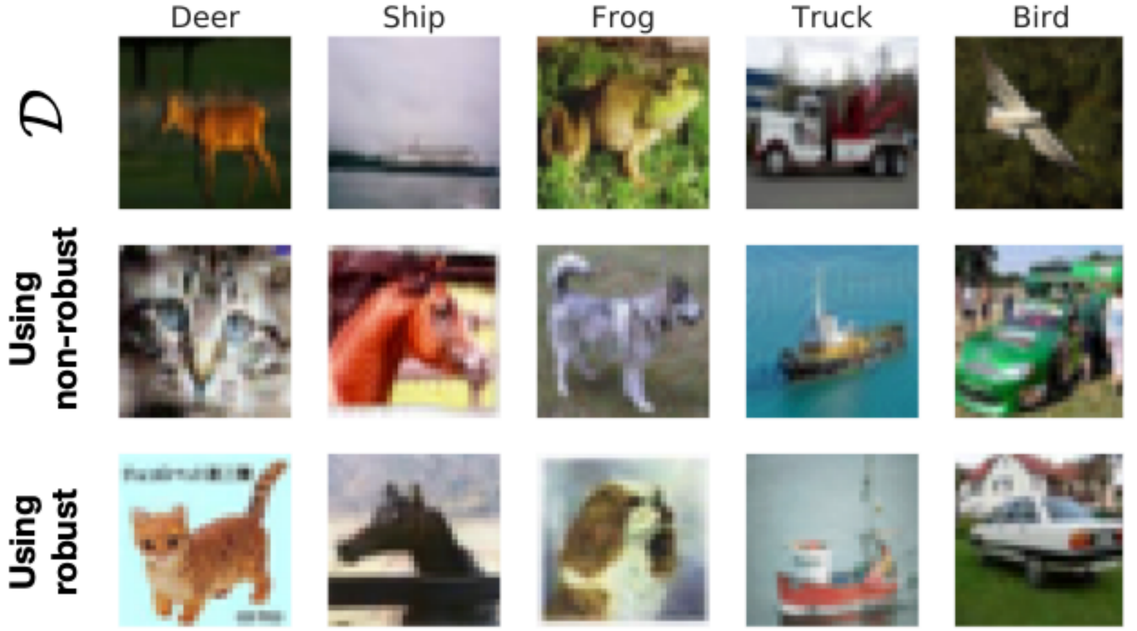
   $x_{NR} \leftarrow \min_{||x'-x|| \leq \varepsilon} L_C(x', t)$ # Solved using ℓ_2 PGD

   $D_{NR} \leftarrow D_{NR} \cup \{(x_{NR}, t)\}$
4. Return D_{NR}

Source Dataset	Dataset	
	CIFAR-10	ImageNet$_R$
$\mathcal{D}$	95.3%	96.6%
$\widehat{\mathcal{D}}_{rand}$	63.3%	87.9%
$\widehat{\mathcal{D}}_{det}$	43.7%	64.4%

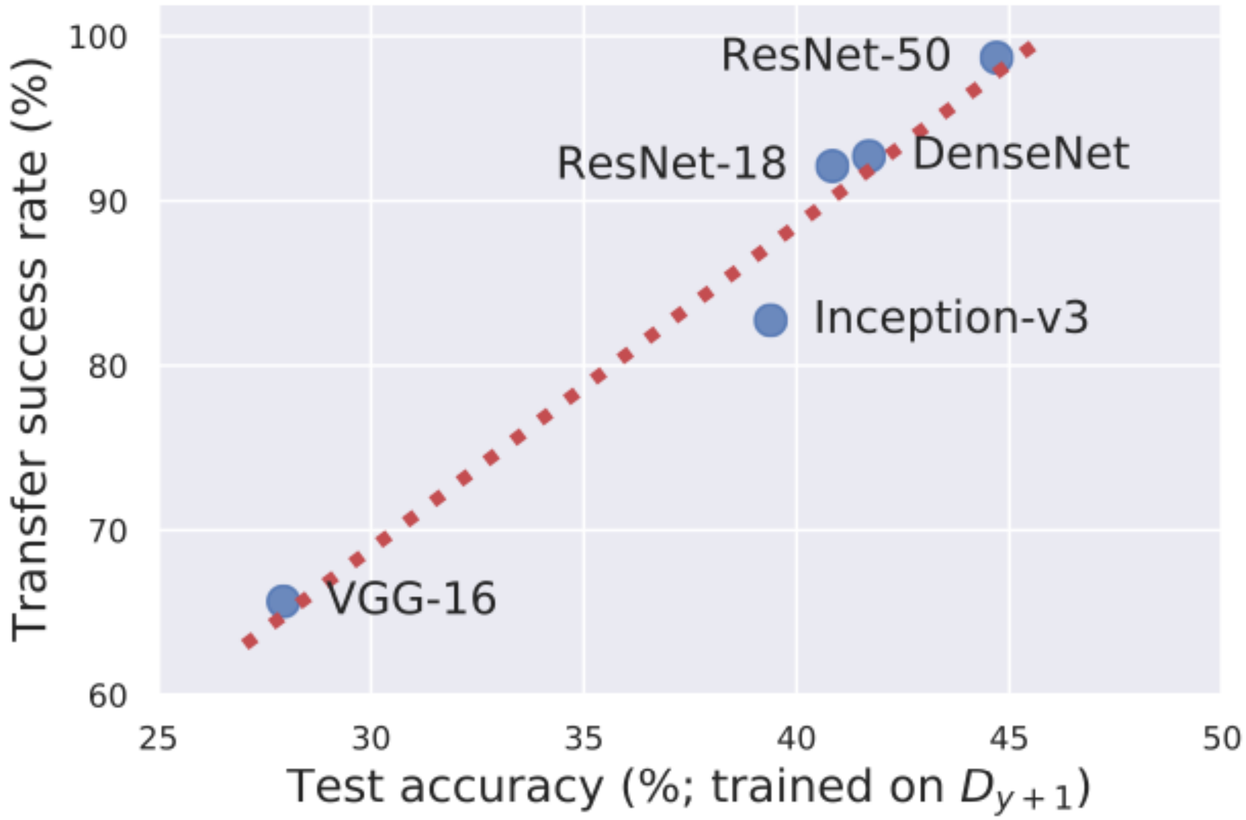


(a) $\widehat{\mathcal{D}}_{rand}$



(b) $\widehat{\mathcal{D}}_{det}$

Transferability can arise from non-robust features



A Theoretical Framework for Studying (Non)-Robust Features

Setup

- maximum likelihood classification between two Gaussian distributions
- $$y \overset{\text{u.a.r.}}{\sim} \{-1, +1\}, \qquad x \sim \mathcal{N}(y \cdot \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$
- Goal: $$\Theta = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(x; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma})\right]$$
- $$y = \arg\max_y \ell(x; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \operatorname{sign}\left(x^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right)$$
- $$\Theta_r = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|_2 \leq \varepsilon} \ell(x + \delta; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma})\right]$$

Vulnerability from metric misalignment (non-robust features)

Theorem 1 (Adversarial vulnerability from misalignment). *Consider an adversary whose perturbation is determined by the "Lagrangian penalty" form of (12), i.e.*

$$\max_{\delta} \ell(x+\delta; y\cdot\boldsymbol{\mu}, \boldsymbol{\Sigma}) - C\cdot\|\delta\|_2,$$

where $C \geq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma}_*)}$ *is a constant trading off NLL minimization and the adversarial constraint*[14]*. Then, the adversarial loss* $\mathcal{L}_{adv}$ *incurred by the non-robustly learned* $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *is given by:*

$$\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) = tr\left[\left(I + (C\cdot\boldsymbol{\Sigma}_* - I)^{-1}\right)^2\right] - d,$$

and, for a fixed $tr(\boldsymbol{\Sigma}_*) = k$ *the above is minimized by* $\boldsymbol{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$.

Robust Learning

Theorem 2 (Robustly Learned Parameters). *Just as in the non-robust case, $\boldsymbol{\mu}_r = \boldsymbol{\mu}^*$, i.e. the true mean is learned. For the robust covariance $\boldsymbol{\Sigma}_r$, there exists an $\varepsilon_0 > 0$, such that for any $\varepsilon \in [0, \varepsilon_0)$,*

$$\boldsymbol{\Sigma}_r = \frac{1}{2}\boldsymbol{\Sigma}_* + \frac{1}{\lambda} \cdot \boldsymbol{I} + \sqrt{\frac{1}{\lambda} \cdot \boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}, \qquad \textit{where} \qquad \Omega\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}+\varepsilon^{3/2}}\right) \leq \lambda \leq O\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}}\right).$$

The effect of robust optimization under an ℓ_2-constrained adversary is visualized in Figure 4. As ϵ grows, the learned covariance becomes more aligned with identity. For instance, we can see that the classifier learns to be less sensitive in certain directions, despite their usefulness for natural classification.

Gradient Interpretability

Theorem 3 (Gradient alignment). *Let $f(x)$ and $f_r(x)$ be monotonic classifiers based on the linear separator induced by standard and ℓ_2-robust maximum likelihood classification, respectively. The maximum angle formed between the gradient of the classifier (wrt input) and the vector connecting the classes can be smaller for the robust model:*

$$\min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f_r(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f_r(x)\|} > \min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f(x)\|}.$$

Конец