Adversarial attacks
and
anns implicit bias.

Content

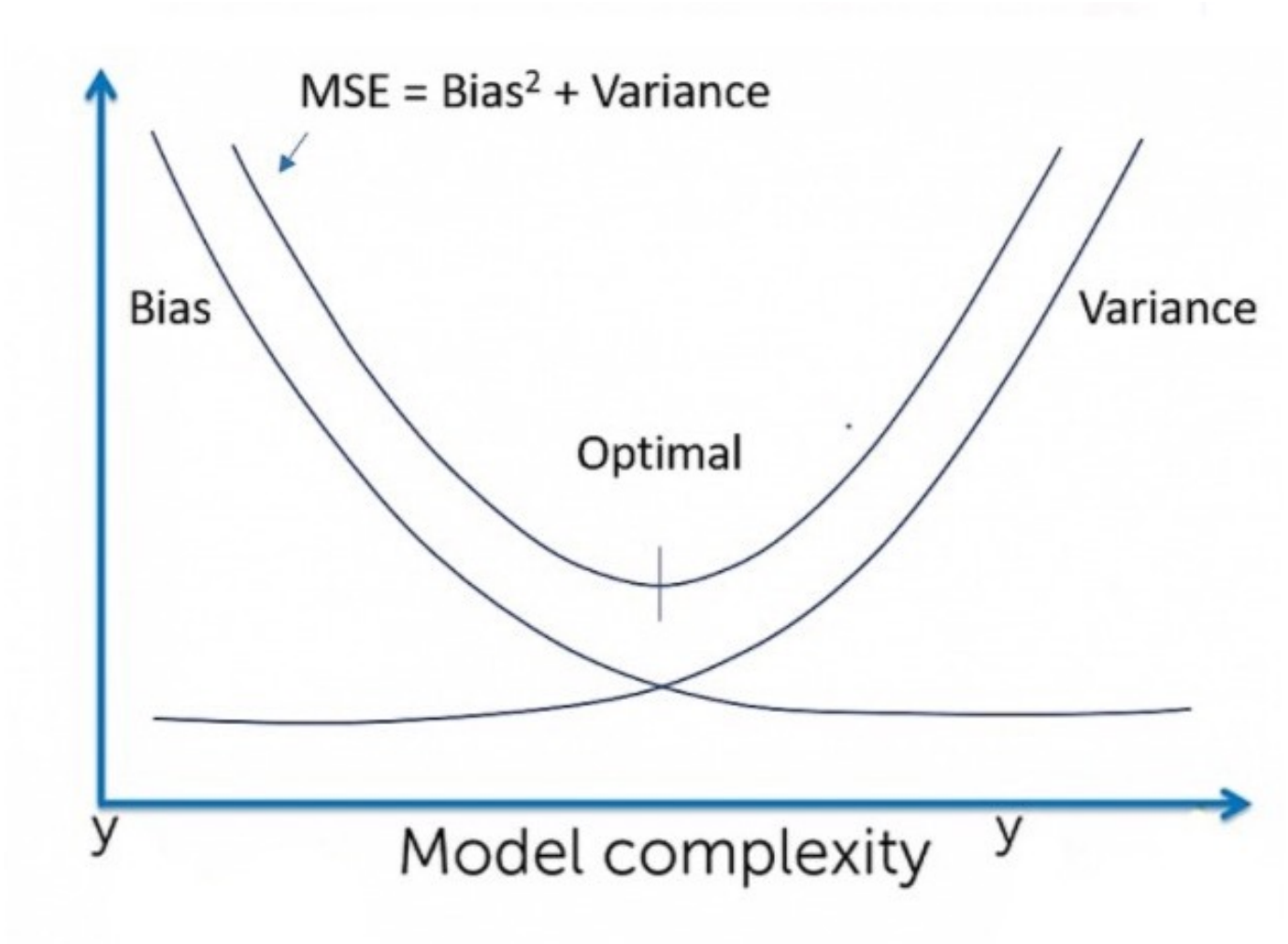
1. More on Implicit Bias in neural networks:

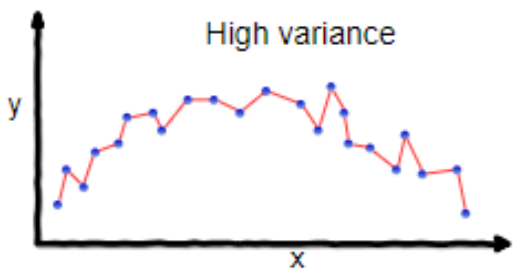
- The Implicit Bias of Gradient Descent on Separable Data: linear case
- Implicit bias for 1 neuron ReLu Networks
- Implicit Bias for infinite-width ReLu shallow network
- Implicit Bias for linear convolutional networks

2. Adversarsarial attacks (short), Robustness and implicit bias.

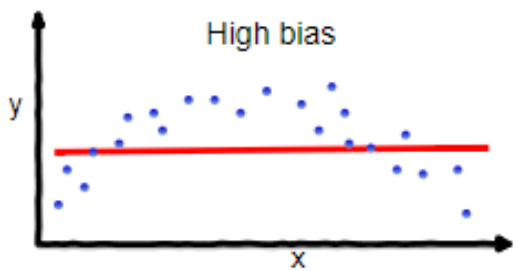
Fourier bias and robustness.

Recap IB

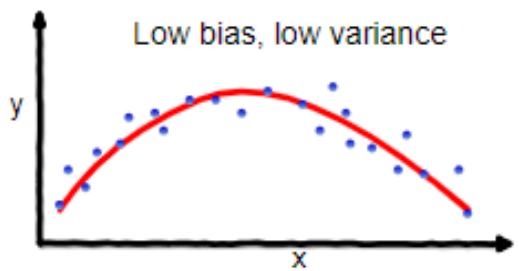




overfitting



underfitting



Good balance

Implicit regularization

Number of parameters grows but the effective learnable parameters are bounded

Implicit regularization

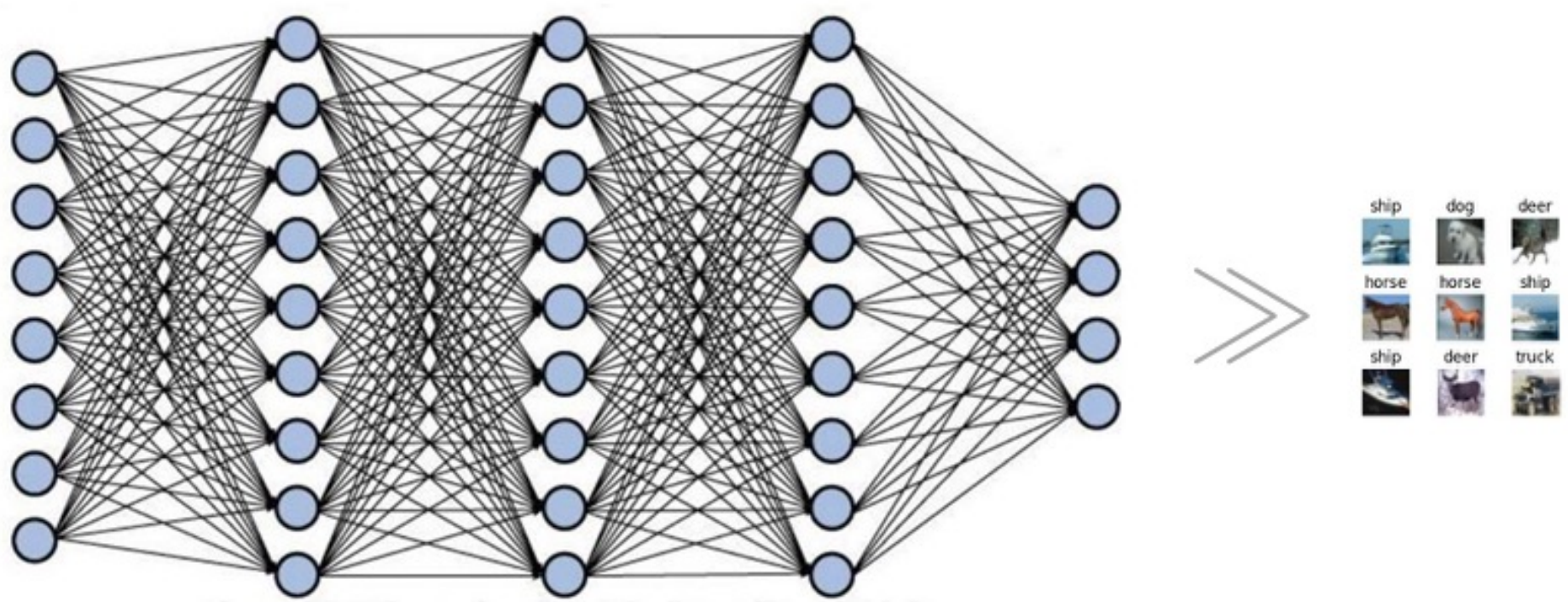
- Explicit regularization imposes an explicit penalty on the parameters of the model (typically some measure of complexity or sensitivity)

- Implicit regularization occurs when the dynamics of training lead to certain minima rather than others

- Algorithm
- Initialization
- Architecture

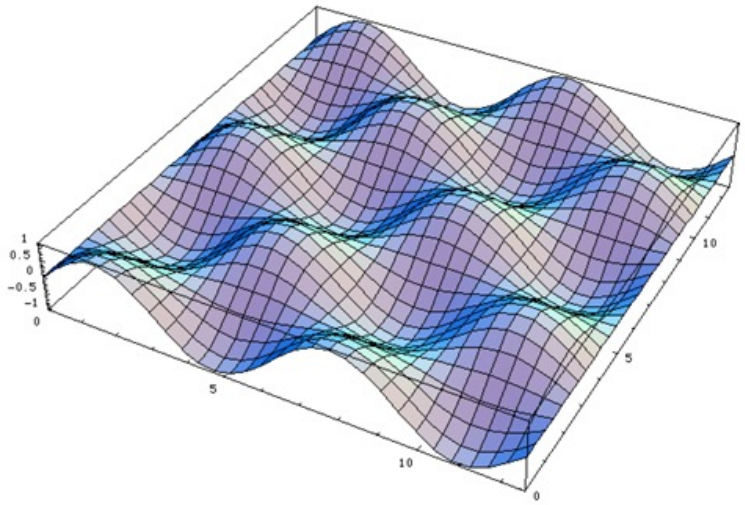
Geometric picture: many global minima

of learned weights $\gg$ # of training examples



Geometric picture: many global minima

Multiple global minima: some generalize well, others don't



Solution found by Gradient Descent (GD) often generalizes well

Conventional Wisdom

Gradient-based optimization induces an implicit regularization

1-neuron ReLu implicit bias

Notations. We use bold-faced letters to denote vectors, e.g., $\mathbf{x} = (x_1, \ldots, x_d)$. For $\mathbf{x} \in \mathbb{R}^d$ we denote by $\|\mathbf{x}\|$ the Euclidean norm.

Single-neuron networks. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a training dataset, where for every i we have $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We consider empirical-loss minimization of a single-neuron network, with respect to the square loss. Thus, the objective is given by

$$\mathcal{L}(\mathbf{w}) := \frac{1}{2} \sum_{i=1}^{n} (\sigma(\langle \mathbf{x}_i, \mathbf{w} \rangle) - y_i)^2 \ , \tag{1}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function. Let $X \in \mathbb{R}^{n \times d}$ denote the data matrix, i.e., the rows of X are $\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top$, and let $\mathbf{y} = (y_1, \ldots, y_n)$. We have

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \|\sigma(X\mathbf{w}) - \mathbf{y}\|^2 \ ,$$

where σ is applied component-wise. We assume that the data is realizable, that is, $\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = 0$. Moreover, we focus on settings where the network is *overparameterized*, in the sense that $\mathcal{L}$ has multiple (or even infinitely many) global minima.

We analyze the implicit regularization of gradient flow on the objective given by Eq. 1. This setting captures the behaviour of gradient descent with infinitesimally small step size. Let $\mathbf{w}(t)$ be the trajectory of gradient flow, where $\mathbf{w}(0)$ is the initial point. The dynamics of $\mathbf{w}(t)$ is given by the differential equation

$$\dot{\mathbf{w}}(t) := \frac{d\mathbf{w}(t)}{dt} = -\nabla \mathcal{L}(\mathbf{w}(t)) \ .$$

If $\lim_{t \to \infty} \mathbf{w}(t)$ exists then we denote it by $\mathbf{w}(\infty)$. We note that gradient flow is not guaranteed to converge to a global minimum (cf. Yehudai and Shamir [2020]). In the overparameterized setting there can be infinitely many global minima, and we study to which one gradient flow converges, assuming that it converges to a global minimum.

The implicit regularization of a ReLU neuron is approximately the ℓ_2 norm

While the implicit regularization of ReLU neuron cannot be expressed as a function $\mathcal{R}(\mathbf{w})$, in this section we show that it can be expressed approximately, within a factor of 2, by the ℓ_2 norm. This implies that even without early stopping, if the data can be labeled by a ReLU neuron with small ℓ_2 norm, then gradient flow will converge to a ReLU neuron whose ℓ_2 norm is not much larger. Since a ReLU neuron is just a linear function composed with a fixed nonlinearity, this can be used to derive good statistical generalization guarantees, via standard techniques (cf. Shalev-Shwartz and Ben-David [2014]).

Theorem 5.1. *Consider gradient flow on the objective given by Eq. 1, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a monotonically non-decreasing activation function (e.g., ReLU). Assume that $\mathbf{w}(\infty)$ exists and $\mathcal{L}(\mathbf{w}(\infty)) = 0$. Let $\mathbf{w}^* \in \operatorname{argmin}_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(0)\|$ s.t. $\mathcal{L}(\mathbf{w}) = 0$. Then, $\|\mathbf{w}(\infty) - \mathbf{w}(0)\| \leq 2 \cdot \|\mathbf{w}^* - \mathbf{w}(0)\|$.*

Proof. First, note that

$$\begin{aligned}\langle \nabla \mathcal{L}(\mathbf{w}(t)), \mathbf{w}(t) - \mathbf{w}^* \rangle &= \left\langle \sum_{i=1}^{n} \left(\sigma(\mathbf{x}_i^\top \mathbf{w}(t)) - \sigma(\mathbf{x}_i^\top \mathbf{w}^*)\right) \sigma'(\mathbf{x}_i^\top \mathbf{w}(t)) \mathbf{x}_i, \mathbf{w}(t) - \mathbf{w}^* \right\rangle \\ &= \sum_{i=1}^{n} \left(\sigma(\mathbf{x}_i^\top \mathbf{w}(t)) - \sigma(\mathbf{x}_i^\top \mathbf{w}^*)\right) \sigma'(\mathbf{x}_i^\top \mathbf{w}(t)) (\mathbf{x}_i^\top \mathbf{w}(t) - \mathbf{x}_i^\top \mathbf{w}^*) \,.\end{aligned}$$

Since σ is monotonically non-decreasing then

$$\left(\sigma(\mathbf{x}_i^\top \mathbf{w}(t)) - \sigma(\mathbf{x}_i^\top \mathbf{w}^*)\right) (\mathbf{x}_i^\top \mathbf{w}(t) - \mathbf{x}_i^\top \mathbf{w}^*) \geq 0 \,,$$

and $\sigma'(\mathbf{x}_i^\top \mathbf{w}(t)) \geq 0$. Hence,

$$\langle \nabla \mathcal{L}(\mathbf{w}(t)), \mathbf{w}(t) - \mathbf{w}^* \rangle \geq 0 \,.$$

Therefore, we have

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{w}(t) - \mathbf{w}^*\|^2\right) = \langle \mathbf{w}(t) - \mathbf{w}^*, \dot{\mathbf{w}}(t) \rangle = \langle \mathbf{w}(t) - \mathbf{w}^*, -\nabla \mathcal{L}(\mathbf{w}(t)) \rangle \leq 0 \,.$$

Thus, $\|\mathbf{w}(\infty) - \mathbf{w}^*\| \leq \|\mathbf{w}(0) - \mathbf{w}^*\|$.

Hence,

$$\begin{aligned} \|\mathbf{w}(\infty) - \mathbf{w}(0)\| &= \|\mathbf{w}(\infty) - \mathbf{w}^* + \mathbf{w}^* - \mathbf{w}(0)\| \\ &\leq \|\mathbf{w}(\infty) - \mathbf{w}^*\| + \|\mathbf{w}^* - \mathbf{w}(0)\| \\ &= 2 \cdot \|\mathbf{w}^* - \mathbf{w}(0)\| \,. \end{aligned}$$

□

Remark 5.1. *Note that Theorem 5.1 implies that if* $\mathbf{w}(0) = \mathbf{0}$ *then* $\|\mathbf{w}(\infty)\| \leq 2 \cdot \|\mathbf{w}^*\|$. *By Theorem 4.2, the implicit regularization cannot be expressed as a function of* $\mathbf{w}$. *Hence, while the implicit regularization of a ReLU neuron cannot be expressed exactly as a function of* $\mathbf{w}$, *it can be expressed approximately, within a factor of* 2, *by the* ℓ_2 *norm.*

Non-linearity choice and implicit bias

For the ReLu expansion of a function $f : \mathbb{R} \to \mathbb{R}$ we have:

$$f(x) = \int d\beta \; c(\beta) ReLu(x - \beta) \tag{6}$$

with

$$f'(x) = \int d\beta \; c(\beta) Step(x - \beta) \tag{7}$$

Taking into account of saturation we can use instead of a ReLu the following:

$$SReLu(x) = \frac{1}{w}\Big(ReLu(x - \beta) - ReLu(x - (\beta + w))\Big) \tag{8}$$

where the factor $1/w$ is a normalization factor. Indeed

$$f'(x) = \int d\beta \; d(\beta) \frac{1}{w} Box_w(x - \beta) = \int d\beta \; d(\beta) \bar{Box}_w(x - \beta). \tag{9}$$

where $(1/w)Box = \bar{Box}$ is the Box function at $\beta + w$ with integral normalized to unit.
How can we pass from one representation to the other? Note that:

$$Box_{\beta,w}(x) = Step_\beta(x) - Step_{\beta+w}(x) = (Id - T_w)Step_\beta(x) \equiv \Delta_w Step_\beta(x) \tag{10}$$

where $T : \mathbb{R} \to \mathbb{R}$, $T_w f(x) = f(x + w)$ is a translation operator.
Thus we have

$$d(\beta) = \Delta_w^{-1} c(\beta) \tag{11}$$

where the operator can be explicitly written as a sum of translation operators since:

$$\Delta_w^{-1} = (Id - T_w)^{-1} = \sum_{n \in \mathbb{N}} T_w^n. \tag{12}$$

Indeed:

$$Step_\beta(x) = \Delta_w^{-1} Box_\beta(x) = \sum_{n \in \mathbb{N}} T_w^n Box_{\beta,w}(x). \tag{13}$$

Explicitly:

$$d(\beta) = \sum_{n \in \mathbb{N}} c(\beta - nw) = \sum_{n \in \mathbb{N}} \frac{1}{w} f''(\beta - nw). \tag{14}$$

Check for constants.

Non-linearity choice and implicit bias

Supposing an $N = ceil(x_{max}/w)$ maximum (finite range of neural activations):

$$\left\|f''\right\|_2^2 = \int d\beta\, c^2(\beta) = \int d\beta \left|\sum_{n=0}^{N-1} f''(\beta - nw)\right|^2 = \int dk \left|\mathcal{F}(\sum_{n=0}^{N-1} \frac{1}{w} T_{nw} \partial_\beta^2 f(\beta))\right|^2$$

$$= \int dk\, k^4 \left|\sum_{n=0}^{N-1} \frac{e^{inwk}}{w}\right|^2 \tilde{f}^2(k).$$

Using

$$\left|\sum_{n=0}^{N-1} \frac{e^{inwk}}{w}\right|^2 = \frac{\sin^2(\frac{1}{2}Nwk)}{w^2 \sin^2(\frac{1}{2}wk)}$$

we finally have:

$$\boxed{\left\|f''\right\|_2^2 = \int dk\, \frac{k^4}{w^2} \frac{\sin^2(\frac{1}{2}Nwk)}{\sin^2(\frac{1}{2}wk)} \tilde{f}^2(k)}$$

To compare in the ReLu case a similar calculation would lead to:

$$\left\|f''\right\|_2^2 = \int dk\, k^4\, \tilde{f}^2(k).$$

Suppose $f(t) = \cos(w_1 t) + \cos(w_2 t)$ is a sound wave and let $\sigma(t) = t^2$. Then

$$(\sigma \circ f)(t) = \frac{1}{2}[2 + \cos(2w_1 t) + \cos(2w_2 t) + 2\cos((w_1 + w_2)t) + 2\cos((w_1 - w_2)t)].$$

$$FT\phi(x) = F \sum_n a_n \underbrace{x \odot \cdots \odot x}_{\text{n-times}} = \sum_n a_n \underbrace{\hat{x} * \cdots * \hat{x}}_{\text{n-times}},$$

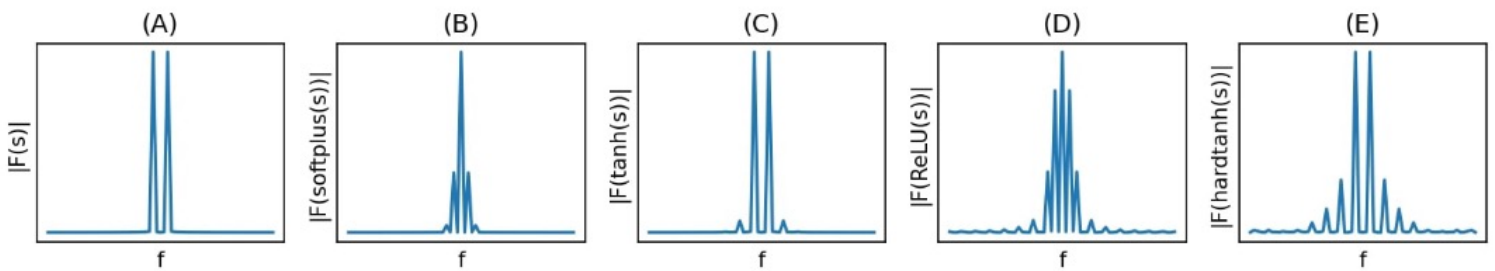
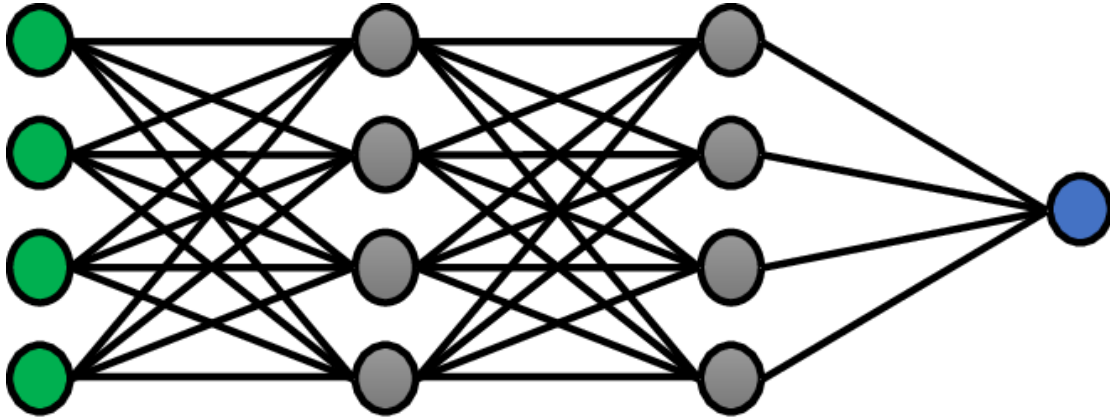


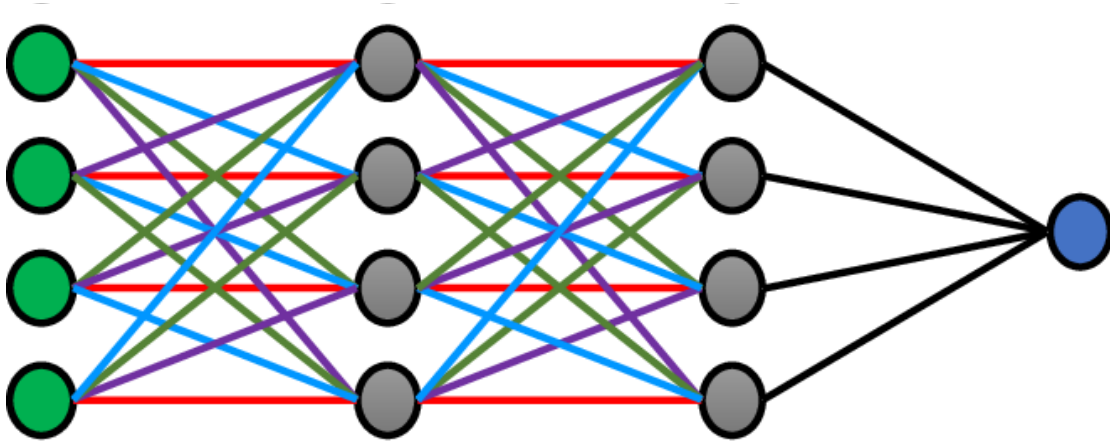
Figure 1. Non-linear distortions in the frequency domain due to the application of (B) softplus, (C) tanh, (D) ReLU and (E) hardtanh non-linear activations on $s(\cdot) = \sin(\cdot)$ (A).

Implicit bias of convolutional and fully connected networks



(a) Fully connected network of depth L

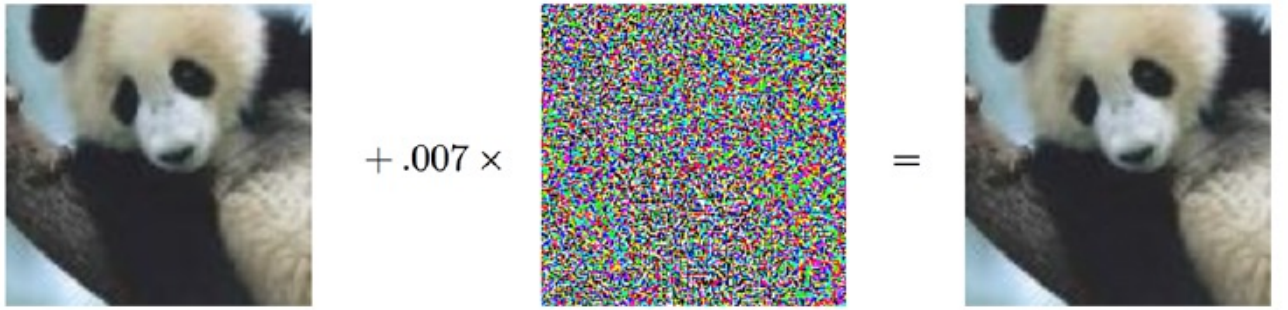
$\overline{\boldsymbol{\beta}}^{\infty} \propto \underset{\forall n,\, y_n \langle \mathbf{x}_n, \boldsymbol{\beta} \rangle \geq 1}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_2$ (independent of L)



(b) Convolutional network of depth L

$\overline{\boldsymbol{\beta}}^{\infty} \propto$ first order stationary point of $\underset{\forall n,\, y_n \langle \mathbf{x}_n, \boldsymbol{\beta} \rangle \geq 1}{\operatorname{argmin}} \|\widehat{\boldsymbol{\beta}}\|_{2/L}$

Adversarial attacks



Definition 1 (Adversarial Attack)**.** *Let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be a data point belong to class $\mathcal{C}_i$. Define a target class $\mathcal{C}_t$. An **adversarial attack** is a mapping $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$ such that the perturbed data*

$$\boldsymbol{x} = \mathcal{A}(\boldsymbol{x}_0)$$

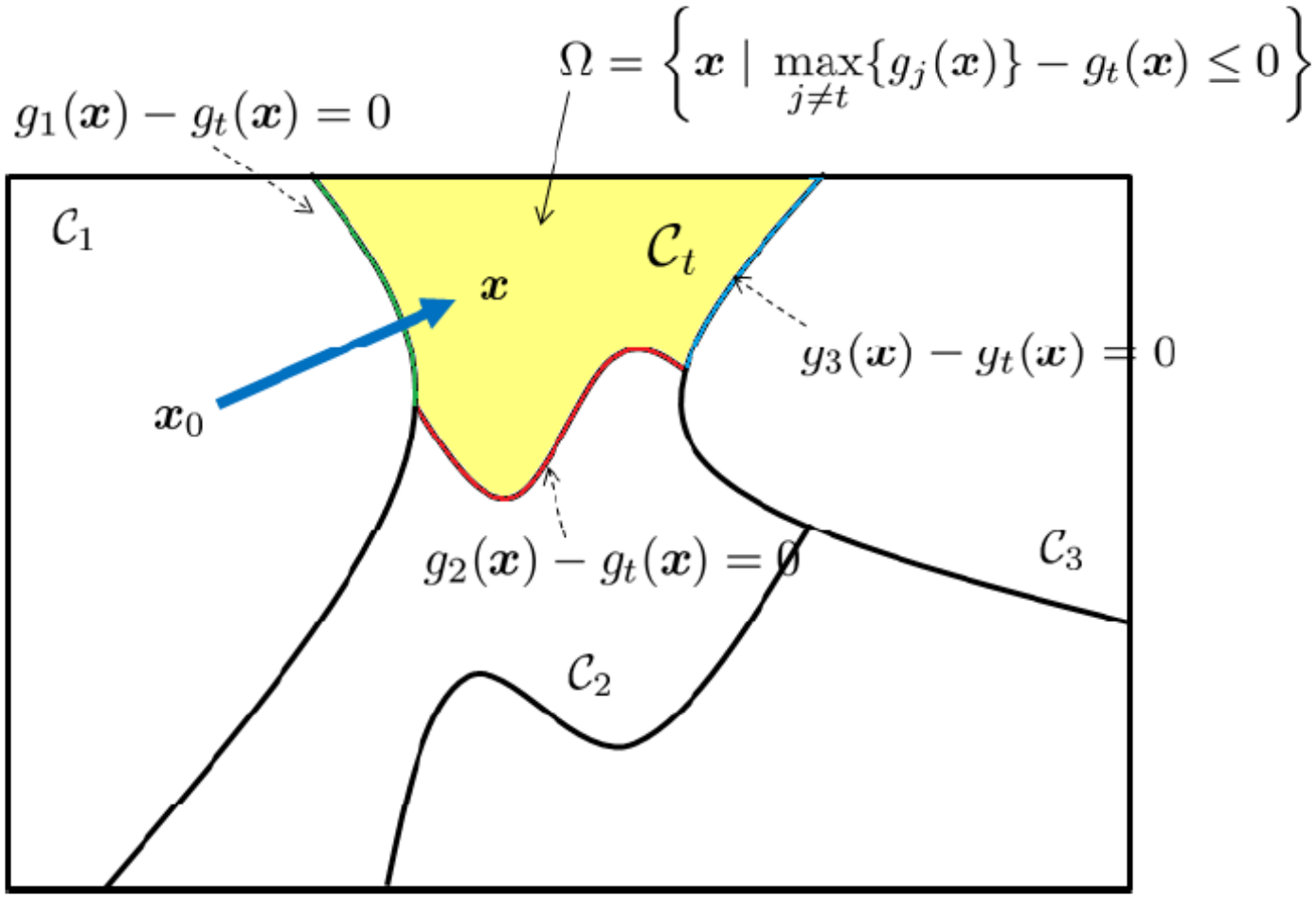
is misclassified as $\mathcal{C}_t$.

Definition 2 (Additive Adversarial Attack)**.** *Let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be a data point belong to class $\mathcal{C}_i$. Define a target class $\mathcal{C}_t$. An **additive** adversarial attack is an addition of a perturbation $\boldsymbol{r} \in \mathbb{R}^d$ such that the perturbed data*

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{r}$$

is misclassified as $\mathcal{C}_t$.

Geometry



$$\Omega = \left\{ \boldsymbol{x} \;\middle|\; \begin{array}{ll} g_1(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \\ g_2(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \\ \vdots & \\ g_k(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \end{array} \right\}$$

Attacks

Definition 3 (Minimum Norm Attack). *The* ***minimum norm attack*** *finds a perturbed data* $\boldsymbol{x}$ *by solving the optimization*

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \|\boldsymbol{x} - \boldsymbol{x}_0\| \\ \text{subject to} \quad & \max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \leq 0, \end{aligned} \tag{3.3}$$

where $\|\cdot\|$ *can be any norm specified by the user.*

Definition 4 (Maximum Allowable Attack). *The* ***maximum allowable attack*** *finds a perturbed data* $\boldsymbol{x}$ *by solving the optimization*

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \\ \text{subject to} \quad & \|\boldsymbol{x} - \boldsymbol{x}_0\| \leq \eta, \end{aligned} \tag{3.4}$$

where $\|\cdot\|$ *can be any norm specified by the user, and* $\eta > 0$ *denotes the magnitude of the attack.*

Definition 5 (Regularization-based Attack). *The* ***regularization-based attack*** *finds a perturbed data* $\boldsymbol{x}$ *by solving the optimization*

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\| + \lambda \left(\max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x})\right) \tag{3.5}$$

where $\|\cdot\|$ *can be any norm specified by the user, and* $\lambda > 0$ *is a regularization parameter.*

Linear attacks

To gain insights about these constraints, we consider a k-class linear classifier. In this case, each discriminant function takes the form

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i,0}. \tag{3.9}$$

The decision boundary between the i-th class and the t-th class is therefore

$$g(\boldsymbol{x}) = (\boldsymbol{w}_i - \boldsymbol{w}_t)^T \boldsymbol{x} + w_{i,0} - w_{t,0} = 0. \tag{3.10}$$

Since we want the perturbed data $\boldsymbol{x}$ to live in $\mathcal{C}_t$, the constraint set becomes

$$\begin{bmatrix} \boldsymbol{w}_1^T - \boldsymbol{w}_t^T \\ \vdots \\ \boldsymbol{w}_{t-1}^T - \boldsymbol{w}_t^T \\ \boldsymbol{w}_{t+1}^T - \boldsymbol{w}_t^T \\ \vdots \\ \boldsymbol{w}_k^T - \boldsymbol{w}_t^T \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} w_{1,0} - w_{t,0} \\ \vdots \\ w_{t-1,0} - w_{t,0} \\ w_{t+1,0} - w_{t,0} \\ \vdots \\ w_{k,0} - w_{t,0} \end{bmatrix} \leq \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{A}^T \boldsymbol{x} \leq \boldsymbol{b} \tag{3.11}$$

where $\boldsymbol{A} = [\boldsymbol{w}_1 - \boldsymbol{w}_t, \ldots, \boldsymbol{w}_k - \boldsymbol{w}_t] \in \mathbb{R}^{d \times (k-1)}$, and $\boldsymbol{b} = [w_{t,0} - w_{1,0}, \ldots, w_{t,0} - w_{k,0}]^T$. This is summarized in the following Lemma.

Lemma 1 (Constraint Set of Linear Classifier). *Consider a k-class linear classifier with discriminant function $g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$ for $i = 1, \ldots, k$. Define*

$$\boldsymbol{A} = [\boldsymbol{w}_1 - \boldsymbol{w}_t, \ldots, \boldsymbol{w}_k - \boldsymbol{w}_t] \in \mathbb{R}^{d \times (k-1)}, \quad \text{and} \quad \boldsymbol{b} = [w_{t,0} - w_{1,0}, \ldots, w_{t,0} - w_{k,0}]^T \in \mathbb{R}^{k-1},$$

Then, the constraint set is

$$\Omega = \{\boldsymbol{x} \mid \boldsymbol{A}^T \boldsymbol{x} \leq \boldsymbol{b}\}. \tag{3.12}$$

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \quad \text{subject to} \quad \boldsymbol{A}^T\boldsymbol{x} \leq \boldsymbol{b}$$

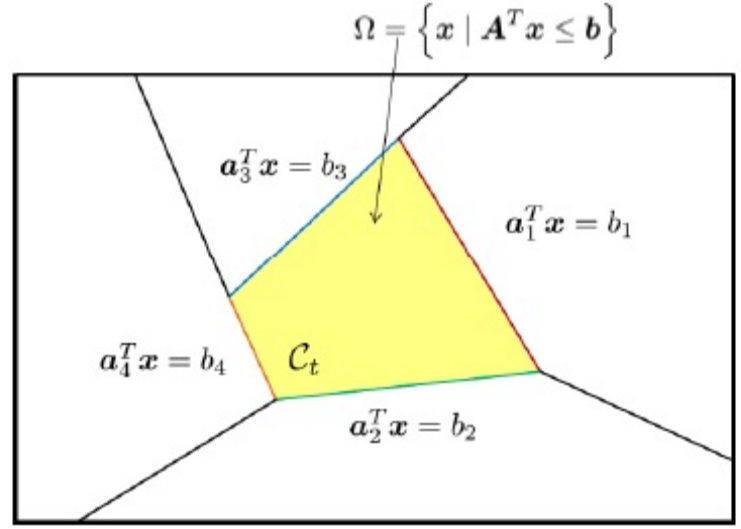


Figure 3.4: Geometry of the constraint set Ω for a linear classifier. The constraint set Ω is now a polygon with decision boundaries defined by $\boldsymbol{a}_i^T\boldsymbol{x} = b_i$, where $\boldsymbol{a}_i = \boldsymbol{w}_i - \boldsymbol{w}_t$ and $b_i = w_{i0} - w_{t0}$.

Linear attacks

Minimum norm attack

A. Minimum-Norm Attack

In order to gain insight of the adversarial attack, it is useful to consider a simple linear classifier with only two classes. A linear classifier has a discriminant function

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}, \tag{3.22}$$

and therefore by defining $\boldsymbol{w} \stackrel{\text{def}}{=} \boldsymbol{w}_i - \boldsymbol{w}_t$ and $w_0 \stackrel{\text{def}}{=} w_{i0} - w_{t0}$, we can show that

$$g_i(\boldsymbol{x}) - g_t(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0. \tag{3.23}$$

Recalling Equation (3.3), the adversarial attack can be formulated as

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \\ \text{subject to} & g_i(\boldsymbol{x}) - g_t(\boldsymbol{x}) = 0. \end{array} \tag{3.24}$$

In this optimization, we simplify $\max_{j \neq t} \{g_j(\boldsymbol{x})\}$ to $g_i(\boldsymbol{x})$ because there are only two classes in our problem. If the index is not t, then it has to be i. We also replaced the inequality $g_i(\boldsymbol{x}) - g_t(\boldsymbol{x}) \leq 0$ as an equality, because the projection theorem in Equation (3.17) suggests that the solution must be on the decision boundary.

Theorem 2 (Minimum ℓ_2 Norm Attack for Two-Class Linear Classifier). *The adversarial attack to a two-class linear classifier is the solution of*

$$\underset{\boldsymbol{x}}{\textit{minimize}} \ \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \ \textit{subject to} \ \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0, \tag{3.25}$$

which is given by

$$\boldsymbol{x}^* = \boldsymbol{x}_0 - \left(\frac{\boldsymbol{w}^T \boldsymbol{x}_0 + w_0}{\|\boldsymbol{w}\|_2} \right) \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}. \tag{3.26}$$

Proof. The Lagrange multiplier of the constrained optimization is given by

$$\mathcal{L}(\boldsymbol{x}, \lambda) = \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + \lambda(\boldsymbol{w}^T \boldsymbol{x} + w_0).$$

The solution of the optimization is the saddle point $(\boldsymbol{x}^*, \lambda^*)$ such that $\nabla_{\boldsymbol{x}} \mathcal{L} = 0$ and $\nabla_\lambda \mathcal{L} = 0$. Taking derivative with respect to $\boldsymbol{x}$ and λ yields

$$\begin{aligned} \nabla_{\boldsymbol{x}} \mathcal{L} &= \boldsymbol{x} - \boldsymbol{x}_0 + \lambda \boldsymbol{w} = 0, \\ \nabla_\lambda \mathcal{L} &= \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0. \end{aligned}$$

Regularization norm attack

C. Regularization-based Attack

In both minimum norm attack and maximum allowable attack, the optimizations are constrained. For certain simple cases, e.g., binary linear classifiers, closed-form solutions can be derived. However, for advanced classifiers such as deep neural networks, solving an optimization involving constraints are typically very difficult. The regularization-based attack alleviates the problem by considering

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x}-\boldsymbol{x}_0\|^2+\lambda\left(\max_{j\neq t}\{g_j(\boldsymbol{x})\}-g_t(\boldsymbol{x})\right).$$

For two-class linear classifier, this is simplified to

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x}-\boldsymbol{x}_0\|^2+\lambda(\boldsymbol{w}^T\boldsymbol{x}+w_0).$$

Since the regularization-based attack is an unconstrained version of the minimum norm attack and the maximum allowable attack, one should expect to have a very similar solution.

Theorem 6 (Regularization-based Attack for Two-Class Linear Classifier). *The regularization-based attack for a two-class linear classifier generates the attack by solving*

$$\underset{\boldsymbol{x}}{\textit{minimize}} \quad \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2+\lambda(\boldsymbol{w}^T\boldsymbol{x}+w_0), \tag{3.46}$$

of which the solution is given by

$$\boldsymbol{x}=\boldsymbol{x}_0-\lambda\boldsymbol{w}. \tag{3.47}$$

Proof. Let $\varphi(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2+\lambda(\boldsymbol{w}^T\boldsymbol{x}+w_0)$. Taking the first order derivative and sending to zero yields

$$0=\nabla\varphi(\boldsymbol{x})=(\boldsymbol{x}-\boldsymbol{x}_0)+\lambda\boldsymbol{w}. \tag{3.48}$$

Therefore, we have $\boldsymbol{x}=\boldsymbol{x}_0-\lambda\boldsymbol{w}$. □

Non-linear attacks

Definition 6 (DeepFool Attack by Moosavi-Dezfooli et al. 2016). *The DeepFool attack for a two-class classification generates the attack by solving the optimization*

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x}-\boldsymbol{x}_0\|^2 \quad \text{subject to} \quad g(\boldsymbol{x})=0, \tag{3.29}$$

where $g(\boldsymbol{x})=0$ is the nonlinear decision boundary separating the two classes.

Due to the nonlinearity of $g(\boldsymbol{x})$, it is generally very difficult to derive a closed-form solution. The numerical procedure to compute the solution can be derived by iteratively updating $\boldsymbol{x}$, where each iteration minimizes $\boldsymbol{x}$ over the first order approximation of $g(\boldsymbol{x})$:

$$g(\boldsymbol{x}) \approx g(\boldsymbol{x}^{(k)}) + \nabla_x g(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}-\boldsymbol{x}^{(k)}),$$

where $\boldsymbol{x}^{(k)}$ is the k-th iterate of the solution. In other words, we are defining a procedure

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad \|\boldsymbol{x}-\boldsymbol{x}^{(k)}\|^2 \quad \text{subject to} \quad g(\boldsymbol{x}^{(k)}) + \nabla_x g(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}-\boldsymbol{x}^{(k)}) = 0. \tag{3.30}$$

By identifying $\boldsymbol{w}^{(k)} = \nabla_x g(\boldsymbol{x}^{(k)})$ and $w_0^{(k)} = g(\boldsymbol{x}^{(k)}) - \nabla_x g(\boldsymbol{x}^{(k)})^T\boldsymbol{x}^{(k)}$, the linearized problem Equation (3.30) becomes

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad \|\boldsymbol{x}-\boldsymbol{x}^{(k)}\|^2 \quad \text{subject to} \quad (\boldsymbol{w}^{(k)})^T\boldsymbol{x} + w_0^{(k)} = 0$$

The solution can thus be found by using Equation (3.26), yielding

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \left(\frac{(\boldsymbol{w}^{(k)})^T\boldsymbol{x}^{(k)} + w_0^{(k)}}{\|\boldsymbol{w}^{(k)}\|^2}\right)\boldsymbol{w}^{(k)} \\ &= \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_x g(\boldsymbol{x}^{(k)})\|^2}\right)\nabla_x g(\boldsymbol{x}^{(k)}).\end{aligned}$$

This result is summarized in the following corollary.

Corollary 1 (DeepFool Algorithm for Two-Class Problem). *An iterative procedure to obtain the DeepFool attack solution is*

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \underset{\boldsymbol{x}}{\textit{argmin}} \quad \|\boldsymbol{x}-\boldsymbol{x}^{(k)}\|^2 \quad \textit{subject to} \quad g(\boldsymbol{x}^{(k)}) + \nabla_x g(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}-\boldsymbol{x}^{(k)}) = 0 \\ &= \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_x g(\boldsymbol{x}^{(k)})\|^2}\right)\nabla_x g(\boldsymbol{x}^{(k)}).\end{aligned} \tag{3.31}$$

Can we attack with random noise?

3.4 Can Random Noise Attack?

Adversarial attack works because the perturbation is carefully designed. What if we perturb the data by pure i.i.d. Gaussian noise?

Recall that when launching attacks for a linear classifier, the perturbation always takes the form of $\boldsymbol{x} = \boldsymbol{x}_0 + \lambda \boldsymbol{w}$. That is, we want to move $\boldsymbol{x}_0$ along $\boldsymbol{w}$ by an amount λ so that $\boldsymbol{x}$ is misclassified. Now, if we do not move along $\boldsymbol{w}$ but along a random vector $\boldsymbol{r}$ such that

$$\boldsymbol{x} = \boldsymbol{x}_0 + \sigma_r \boldsymbol{r}, \tag{3.57}$$

where $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, then we can ask if we can still misclassify the data point $\boldsymbol{x}$. Clearly, this requires us to check whether $\boldsymbol{w}^T \boldsymbol{r} > 0$. If $\boldsymbol{w}^T \boldsymbol{r} > 0$, then $\boldsymbol{r}$ and $\boldsymbol{w}$ will form an acute angle and so for sufficient step size we will be able to move $\boldsymbol{x}_0$ to another class. If $\boldsymbol{w}^T \boldsymbol{r} < 0$, then $\boldsymbol{w}$ and $\boldsymbol{r}$ are form an obtuse angle and so $\boldsymbol{r}$ will move $\boldsymbol{x}_0$ to an opposite direction.

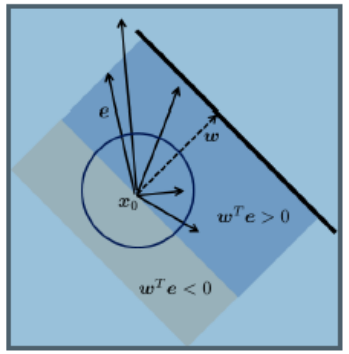


Figure 3.10: Attacking the linear classifier with i.i.d. noise is equivalent to putting an uncertainty circle around $\boldsymbol{x}_0$ with radius σ_r. The possible attack directions are those that form acute angle with the normal vector $\boldsymbol{w}$. Therefore, among all the possible $\boldsymbol{r}$'s, we are only interested in those such that $\boldsymbol{w}^T \boldsymbol{r} > 0$, which occupy half of the space in 2D.

Figure 3.10 provides a pictorial illustration of the noise attack. From this figure, we are tempted to think that i.i.d. noise can achieve 50% attack rate because $\boldsymbol{w}^T \boldsymbol{r} > 0$ occupies half of the space. The problem of such argument is that in high dimension, the probability of $\boldsymbol{w}^T \boldsymbol{r} > 0$ is diminishing very quickly as the dimensionality of $\boldsymbol{r}$ grows. This is a well-known

Can we attack with random noise?

phenomenon called **curse of dimensionality**. To illustrate the idea, let us evaluate the probability of $\boldsymbol{w}^T\boldsymbol{r} \geq \epsilon$ for some $\epsilon > 0$. To this end, let us consider

$$\mathbb{P}\left[\frac{1}{d}\boldsymbol{w}^T\boldsymbol{r} \geq \epsilon\right] = \mathbb{P}\left[\frac{1}{d}\sum_{j=1}^{d} w_j r_j \geq \epsilon\right],$$

where d is the dimensionality of $\boldsymbol{w}$, i.e., $\boldsymbol{w} \in \mathbb{R}^d$. The tolerance level ϵ is a small positive constant that stays away from 0.

Example. Before we discuss the theoretical results, it will be useful to do a quick numerical experiment. Consider a special case where $\boldsymbol{w} = \mathbf{1}_{d\times 1}$, i.e., a d-dimensional all-one vector, and $\boldsymbol{r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. In this case, we define the average as

$$Y \overset{\text{def}}{=} \frac{1}{d}\sum_{j=1}^{d} r_j. \tag{3.58}$$

It follows that Y is a Gaussian random variable because linear combination of Gaussian remains a Gaussian. The mean and variance are

$$\mathbb{E}[Y] = 0, \quad \text{and} \quad \text{Var}[Y] = \frac{1}{d}. \tag{3.59}$$

Therefore, the probability of the event $\{Y > \epsilon\}$ is

$$\begin{aligned}\mathbb{P}[Y > \epsilon] &= \int_{\epsilon}^{\infty} \frac{1}{\sqrt{2\pi/d}} \exp\left\{-\frac{t^2}{2/d}\right\} dt \\ &= \int_{\epsilon\sqrt{\frac{d}{2}}}^{\infty} \frac{1}{\sqrt{\pi}} \exp\left\{-t^2\right\} dt \\ &= \frac{1}{2}\text{erfc}\left(\epsilon\sqrt{d/2}\right),\end{aligned} \tag{3.60}$$

where erfc is the complementary error function defined as $\text{erfc}(z) \overset{\text{def}}{=} \frac{2}{\sqrt{\pi}}\int_z^{\infty} e^{-t^2} dt$. As we can see in Figure 3.11, the probability drops rapidly as d increases.

Let us discuss the general case. One classical result to derive the probability is the tail bound on Gaussian by Feller 1968. [1] If Y is a Gaussian random variable with $Y \sim \mathcal{N}(\mu, \sigma)$, then

$$\mathbb{P}[Y \geq \mu + \sigma\epsilon] \leq \frac{1}{\epsilon}\frac{e^{-\epsilon^2/2}}{\sqrt{2\pi}}. \tag{3.61}$$

Can we attack with random noise?

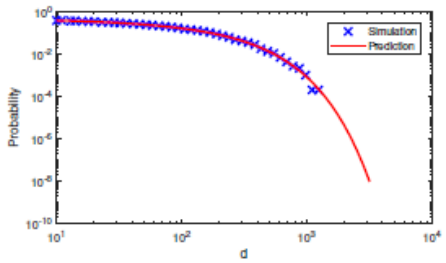


Figure 3.11: Empirical probability and the theoretically predicted value. In this experiment, we let $Y = \frac{1}{d}\sum_{j=1}^{d} r_j$ where $r_j \sim \mathcal{N}(0,1)$ is an iid Gaussian random variable. We plot the probability $\mathbb{P}[Y > \epsilon]$ as a function of d. Note the tightness of the prediction, and the rapid decaying behavior of the probability. The result shows that when d grows, the value Y is concentrated at 0.

Now, let $Y = \frac{1}{d}\sum_{j=1}^{d} w_j r_j$. Since a linear combination of Gaussian remains a Gaussian, it holds that Y is Gaussian and

$$\mu = \mathbb{E}[Y] = 0, \quad \text{and} \quad \sigma^2 = \mathrm{Var}[Y] = \frac{1}{d^2}\sum_{j=1}^{d} w_j^2 = \frac{\|\boldsymbol{w}\|^2}{d^2}. \tag{3.62}$$

Therefore, by substituting $\varepsilon = \sigma\epsilon$ we can show that

$$\mathbb{P}\left[\frac{1}{d}\sum_{j=1}^{d} w_j r_j \geq \varepsilon\right] = \mathbb{P}[Y \geq \varepsilon] \leq \frac{\sigma}{\varepsilon}\frac{e^{-\frac{\varepsilon^2}{2\sigma^2}}}{\sqrt{2\pi}} = \frac{\|\boldsymbol{w}\|}{\varepsilon d\sqrt{2\pi}} \exp\left\{-d^2\frac{\varepsilon^2}{2\|\boldsymbol{w}\|^2}\right\}. \tag{3.63}$$

As $d \to \infty$, it holds that $\mathbb{P}\left[\frac{1}{d}\sum_{j=1}^{d} w_j r_j \geq \varepsilon\right] \to 0$. That means, the probability of getting a "good attack direction" is diminishing to zero exponentially. Putting everything together we have the following theorem.

Theorem 7. *Let $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$ be the decision boundary of a linear classifier, and let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be an input data point. Suppose we attack the classifier by adding i.i.d. Gaussian noise $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ to $\boldsymbol{x}_0$. The probability of a successful attack at a tolerance level ε is $\mathbb{P}\left[\frac{1}{d}\sum_{j=1}^{d} w_j r_j \geq \varepsilon\right]$, and such probability is upper bounded by*

$$\mathbb{P}\left[\frac{1}{d}\sum_{j=1}^{d} w_j r_j \geq \varepsilon\right] \leq \frac{\|\boldsymbol{w}\|}{\varepsilon d\sqrt{2\pi}} \exp\left\{-d^2\frac{\varepsilon^2}{2\|\boldsymbol{w}\|^2}\right\}. \tag{3.64}$$

Therefore, as $d \to \infty$ it becomes increasingly more difficult for i.i.d. Gaussian noise to succeed in attacking.

Class 2: IB and adv attacks

Equivalence robust classifier/maximum margin

Definition 2.2 (Dual norm)**.** Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. The associated *dual norm*, denoted $\|\cdot\|_*$, is defined as $\|\delta\|_* = \sup_{\boldsymbol{x}}\{|\langle \delta, x\rangle| \mid \|x\| \le 1\}$.

Definition 2.3 (Linear Separability)**.** We say a dataset is linearly separable if there exists w, b such that $y_i(w^\top x_i + b) > 0$ for all i.

Lemma 2.1 (Maximally Robust Linear Classifier (Ben-Tal et al. [14], §12))**.** *For linear models and linearly separable data, the following problems are equivalent; i.e., from a solution of one, a solution of the other is readily found.*

$$\textit{Maximally robust classifier:} \quad \arg\max_{w,b}\{\varepsilon \mid y_i(w^\top(x_i+\delta)+b) > 0,\ \forall i, \|\delta\| \le \varepsilon\}\,, \tag{4}$$

$$\textit{Maximum margin classifier:} \quad \arg\max_{w,b:\|w\|_*\le 1} \{\varepsilon \mid y_i(w^\top x_i + b) \ge \varepsilon,\ \forall i\}\,, \tag{5}$$

$$\textit{Minimum norm classifier:} \quad \arg\min_{w,b}\{\|w\|_* \mid y_i(w^\top x_i + b) \ge 1,\ \forall i\}\,. \tag{6}$$

The expression $\min_i y_i(w^\top x_i + b)/\|w\|$ *is the margin of a classifier* w *that is the distance of the nearest training point to the classification boundary, i.e. the line* $\{v : w^\top v = -b\}$.

Proof. We first show that the maximally robust classifier is equivalent to a robust counterpart by removing δ from the problem,

$$\arg\max_{w,b}\{\varepsilon \mid y_i(w^\top(x_i+\delta)+b) > 0,\ \forall i,\ \|\delta\| \le \varepsilon\}$$

(homogeneity of p-norm)

$$= \arg\max_{w,b}\{\varepsilon \mid y_i(w^\top(x_i+\varepsilon\delta)+b) > 0,\ \forall i,\ \|\delta\| \le 1\}$$

(if it is true for all δ it is true for the worst of them)

$$= \arg\max_{w,b}\{\varepsilon \mid \inf_{\|\delta\|\le 1} y_i(w^\top(x_i+\varepsilon\delta)+b) > 0,\ \forall i\}$$

$$= \arg\max_{w,b}\{\varepsilon \mid y_i(w^\top x_i+b) + \varepsilon \inf_{\|\delta\|\le 1} w^\top\delta > 0,\ \forall i\}$$

(definition of dual norm)

$$= \arg\max_{w,b}\{\varepsilon \mid y_i(w^\top x_i+b) > \varepsilon\|w\|_*,\ \forall i\}$$

Assuming $w \neq 0$, which is a result of linear separability assumption, we can divide both sides by $\|w\|_*$ and change variables,

$$= \arg\max_{w,b}\{\varepsilon \mid y_i(w^\top x_i+b) \ge \varepsilon,\ \forall i, \|w\|_* \le 1\}\,,$$

where we are also allowed to change $>$ to $\ge$ because any solution to one problem gives an equivalent solution to the other given $w \neq 0$.

Proof continues

Now we show that the robust counterpart is equivalent to the minimum norm classification problem by removing ε. When the data is linearly separable there exists a solution with $\varepsilon > 0$,

$$\begin{aligned}
&\arg\max_{\boldsymbol{w},b} \{\varepsilon \mid y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) > \varepsilon \|w\|_*,\ \forall i\} \\
&= \arg\max_{\boldsymbol{w},b} \left\{\varepsilon \mid y_i \left(\frac{\boldsymbol{w}^\top}{\varepsilon\|\boldsymbol{w}\|_*}\boldsymbol{x}_i + \frac{b}{\varepsilon\|\boldsymbol{w}\|_*}\right) \geq 1,\ \forall i\right\}
\end{aligned}$$

This problem is invariant to any non-zero scaling of (w, b), so with no loss of generality we set $\|w\|_* = 1$.

$$= \arg\max_{\boldsymbol{w},b} \left\{\varepsilon \mid y_i \left(\frac{\boldsymbol{w}^\top}{\varepsilon}\boldsymbol{x}_i + b\right) \geq 1,\ \forall i, \|\boldsymbol{w}\|_* = 1\right\}$$

Let $w' = w/\epsilon$, then the solution to the following problem gives a solution for w,

$$\begin{aligned}
&\arg\max_{\boldsymbol{w}',b} \left\{\frac{1}{\|\boldsymbol{w}'\|_*} \mid y_i(\boldsymbol{w}'^\top \boldsymbol{x}_i + b) \geq 1,\ \forall i\right\} \\
&= \arg\min_{\boldsymbol{w}',b} \left\{\|\boldsymbol{w}'\|_* \mid y_i(\boldsymbol{w}'^\top \boldsymbol{x}_i + b) \geq 1,\ \forall i\right\}.
\end{aligned}$$

Robustness and implicit Bias

Theorem 3.1 (Implicit Bias of Steepest Descent (Gunasekar et al. [12] (Theorem 5))). *For any separable dataset* $\{x_i, y_i\}$ *and any norm* $\|\cdot\|$, *consider the steepest descent updates from* (7) *for minimizing the empirical risk* $\mathcal{L}(w)$ *(defined in Section 2) with the exponential loss,* $\zeta(z) = \exp(-z)$. *For all initializations* w_0, *and all bounded step-sizes satisfying a known upper bound, the iterates* w_t *satisfy*

$$\lim_{t\to\infty} \min_i \frac{y_i w_t^\top x_i}{\|w_t\|} = \max_{w:\|w\|\le 1} \min_i y_i w^\top x_i \,. \tag{8}$$

In particular, if a unique maximum margin classifier $w^*_{\|\cdot\|} = \arg\max_{w:\|w\|\le 1} \min_i y_i w^\top x_i$ *exists, the limit direction converges to it, i.e.* $\lim_{t\to\infty} \frac{w_t}{\|w_t\|} = w^*_{\|\cdot\|}$.

Robustness and implicit Bias

Corollary 1 (Implicit Robustness of Steepest Descent). *For any linearly separable dataset and any norm* $\|\cdot\|$*, steepest descent iterates minimizing the empirical risk,* $\mathcal{L}(\boldsymbol{w})$*, satisfying the conditions of Theorem 3.1, converge in direction to a maximally robust classifier,*

$$\arg\max_{\boldsymbol{w}}\{\varepsilon \,|\, y_i \boldsymbol{w}^\top (\boldsymbol{x}_i + \boldsymbol{\delta}) > 0,\ \ \forall i,\ \|\boldsymbol{\delta}\|_* \leq \varepsilon\}\,.$$

In particular, a maximally robust classifier against ℓ_1*,* ℓ_2*, and* ℓ_∞ *is reached, respectively, by sign gradient descent, gradient descent, and coordinate descent.*

Proof. By Theorem 3.1, the margin of the steepest descent iterates, $\min_i \frac{y_i \boldsymbol{w}_t^\top \boldsymbol{x}_i}{\|\boldsymbol{w}_t\|}$, converges as $t \to \infty$ to the maximum margin, $\max_{\boldsymbol{w}:\|\boldsymbol{w}\|\leq 1} \min_i y_i \boldsymbol{w}^\top \boldsymbol{x}_i$. By Lemma 2.1, any maximum margin classifier w.r.t. $\|\cdot\|$ gives a maximally robust classifier w.r.t. $\|\cdot\|_*$. □

IB Fourier and maximal robustness

Definition 3.3 (Discrete Fourier Transform). $\mathcal{F}(\boldsymbol{w}) \in \mathbb{C}^d$ denotes the Fourier coefficients of $\boldsymbol{w}$ where $[\mathcal{F}(\boldsymbol{w})]_d = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} [\boldsymbol{w}]_k \exp(-\frac{2\pi j}{d} kd)$ and $j^2 = -1$.

Theorem 3.2 (Implicit Bias towards Fourier Sparsity (Gunasekar et al. [22], Theorem 2, 2.a)). *Consider the family of L-layer linear convolutional networks and the sequence of gradient descent iterates, $\boldsymbol{w}_t$, minimizing the empirical risk, $\mathcal{L}(\boldsymbol{w})$, with the exponential loss, $\exp(-z)$. For almost all linearly separable datasets under known conditions on the step size and convergence of iterates, $\boldsymbol{w}_t$ converges in direction to the classifier minimizing the norm of the Fourier coefficients given by*

$$\arg\min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_L} \{\|\mathcal{F}(\boldsymbol{w})\|_{2/L} \mid y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq 1, \forall i\}. \tag{9}$$

In particular, for two-layer linear convolutional networks the implicit bias is towards the solution with minimum ℓ_1 norm of the Fourier coefficients, $\|\mathcal{F}(\boldsymbol{w})\|_1$. For $L > 2$, the convergence is to a first-order stationary point.

IB Fourier and maximal robustness

Corollary 2 (Maximally Robust to Perturbations with Bounded Fourier Coefficients). *Consider the family of two-layer linear convolutional networks and the gradient descent iterates, $\boldsymbol{w}_t$, minimizing the empirical risk. For almost all linearly separable datasets under conditions of Theorem 3.2, $\boldsymbol{w}_t$ converges in direction to a maximally robust classifier,*

$$\arg\max_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_L} \{\varepsilon \,|\, y_i \varphi_{conv}(\boldsymbol{x}_i + \boldsymbol{\delta}; \{\boldsymbol{w}_l\}_{l=1}^{L}) > 0, \ \forall i, \ \|\mathcal{F}(\boldsymbol{\delta})\|_\infty \leq \varepsilon\} .$$

Fourier, proof

B.2 Proof of Maximally Robust to Perturbations Bounded in Fourier Domain (Corollary 2)

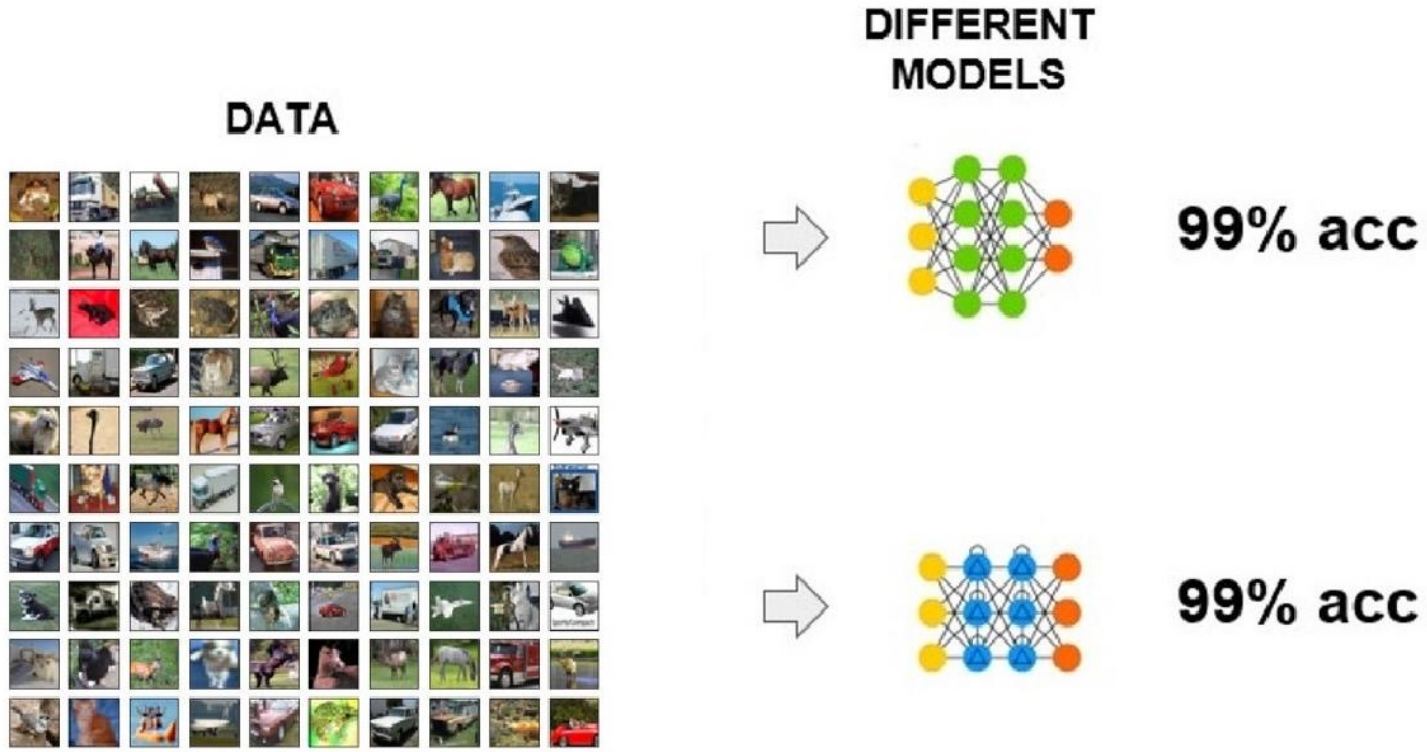
The proof mostly follows from the equivalence for linear models in Appendix B.1 by substituting the dual norm of Fourier-ℓ_1. Here, $\boldsymbol{A}^*$ denotes the complex conjugate transpose, $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}^*$ is the complex inner product, $[\boldsymbol{F}]_{ik} = \frac{1}{\sqrt{D}} \omega_D^{ik}$ the DFT matrix where $\omega_D = e^{-j2\pi/D}$, $j = \sqrt{-1}$.

Let $\| \cdot \|$ be a norm on $\mathbb{C}^n$ and $\langle \cdot, \cdot \rangle$ be the complex inner product. Similar to $\mathbb{R}^n$, the associated dual norm is defined as $\|\boldsymbol{\delta}\|_* = \sup_{\boldsymbol{x}} \{ |\langle \boldsymbol{\delta}, \boldsymbol{x} \rangle| \mid \|\boldsymbol{x}\| \leq 1 \}$.

$$
\begin{aligned}
&\|\mathcal{F}(\boldsymbol{w})\|_1 \\
&= \sup_{\|\boldsymbol{\delta}\|_\infty \leq 1} |\langle \mathcal{F}(\boldsymbol{w}), \boldsymbol{\delta} \rangle| \\
&\qquad \text{(Expressing DFT as a linear transformation.)} \\
&= \sup_{\|\boldsymbol{\delta}\|_\infty \leq 1} |\langle \boldsymbol{F}\boldsymbol{w}, \boldsymbol{\delta} \rangle| \\
&= \sup_{\|\boldsymbol{\delta}\|_\infty \leq 1} |\langle \boldsymbol{w}, \boldsymbol{F}^*\boldsymbol{\delta} \rangle| \\
&\qquad \text{(Change of variables and } \boldsymbol{F}^{-1} = \boldsymbol{F}^*\text{.)} \\
&= \sup_{\|\boldsymbol{F}\boldsymbol{\delta}\|_\infty \leq 1} |\langle \boldsymbol{w}, \boldsymbol{\delta} \rangle| \\
&= \sup_{\|\mathcal{F}(\boldsymbol{\delta})\|_\infty \leq 1} |\langle \boldsymbol{w}, \boldsymbol{\delta} \rangle| \,.
\end{aligned}
$$

Fourier Bias and adversarial training

Multiple solutions for the same computational problem



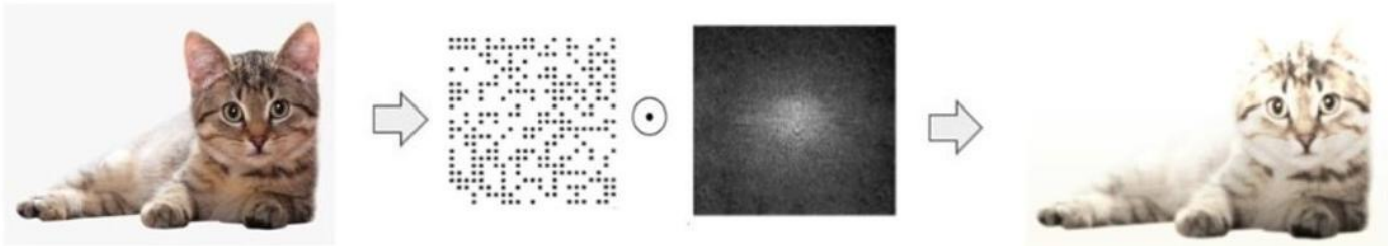
How is the network choosing a particular solution?

An aspect of the implicit bias: important Fourier features. A step towards testing for interpretability

Question

Which Fourier features are essential to the network to solve the task?

x $\quad M_\Phi \odot \mathcal{F}x$ $\quad \mathcal{F}^{-1}(M_\Phi \odot \mathcal{F}x)$



$$M_\Phi(\lambda, p) = \text{argmin}_{M_\Phi} \sum_{x \in \mathcal{X}_V} e^{[\mathcal{L}(\Phi(\bar{x}),y) - \mathcal{L}(\Phi(x),y)]^2} + \lambda \| M_\Phi \|_p, \quad \lambda \in \mathbb{R}_+,$$

Result: M_Φ highlights the important Fourier features.

The study hasn't been done in the context of neuroscience but the technique can be applied.

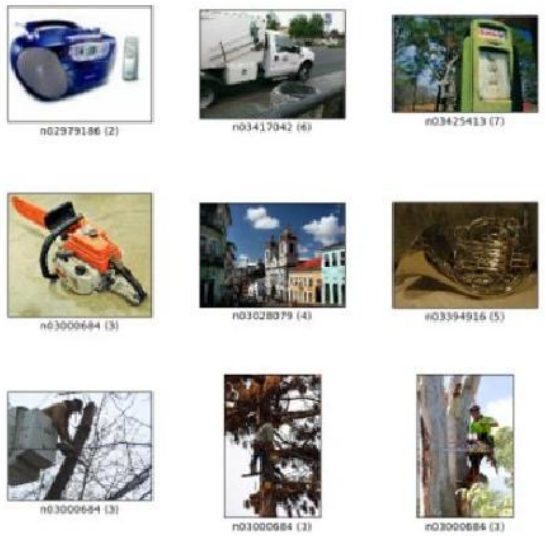
Algorithm

- Train the model
- Freeze the weights and add a mask layer
- Train the mask to
  1. preserve the score
  2. maximize the frequency sparsity

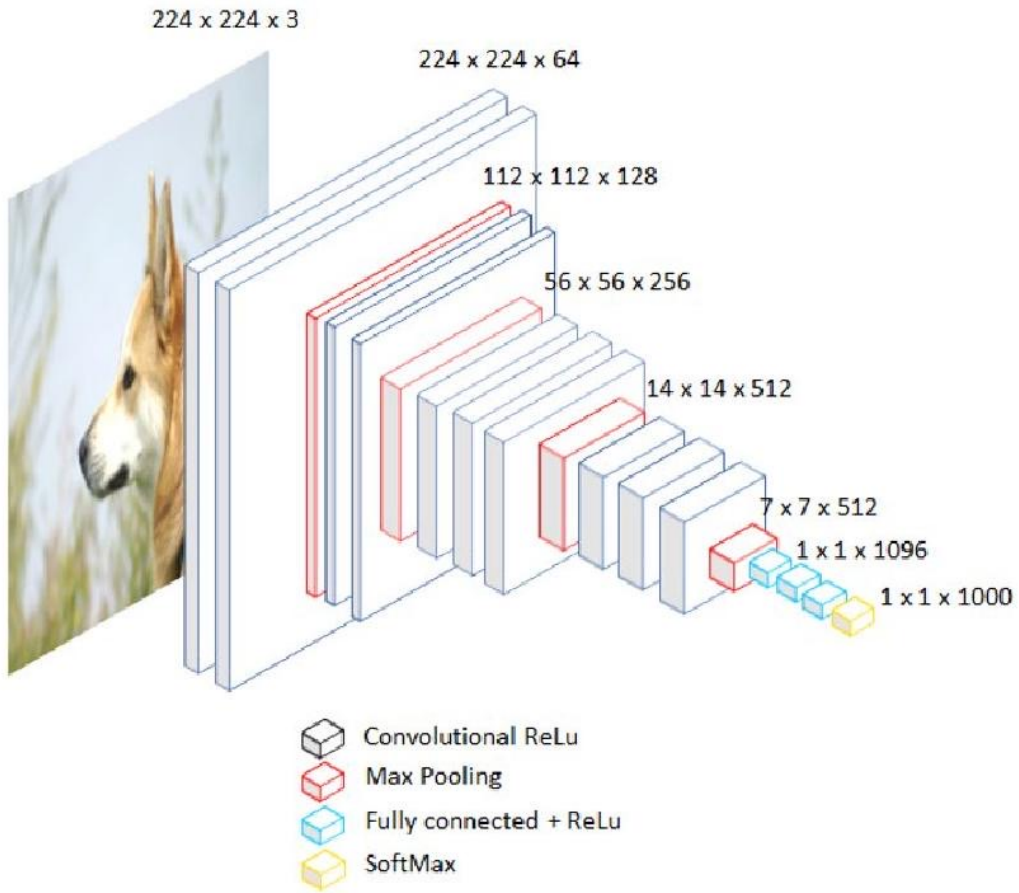
Why Fourier? Many successful stories

- Abello et al. (2021). Dissecting the high-frequency bias in convolutional neural In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- Caro et al. (2020). Local convolutions cause an implicit bias towards high frequency adversarial examples.
- Christian et al. (2021). Ringing relus: Harmonic distortion analysis of nonlinear feedforward networks. ICLR
- Geirhos, R.et al. Imagenet trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. ICLR. Li, Z. et al. (2022). Robust deep learning object recognition models rely on low frequency information in natural images.
- Ortiz-Jimenez, G. at al. (2020). Neural anisotropy directions.
- Sharma, Y., Ding, G. W., and Brubaker, M. A. (2019). On the Effectiveness of Low Frequency Perturbations. IJCAI

Data and Model: Imagenette and VGG11

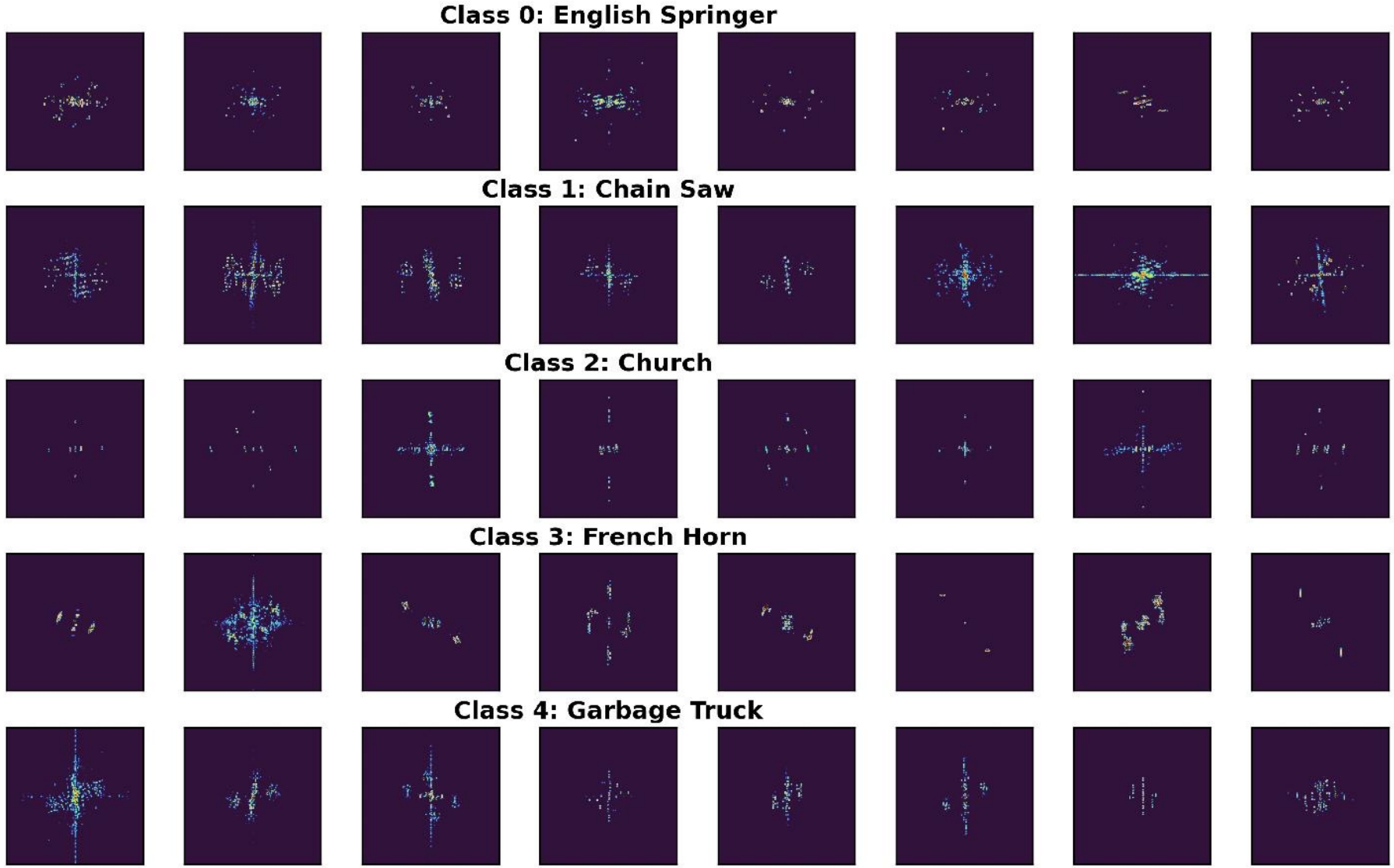


```
lbl_dict = dict(
    n01440764='tench',
    n02102040='English springer',
    n02979186='cassette player',
    n03000684='chain saw',
    n03028079='church',
    n03394916='French horn',
    n03417042='garbage truck',
    n03425413='gas pump',
    n03445777='golf ball',
    n03888257='parachute'
)
```

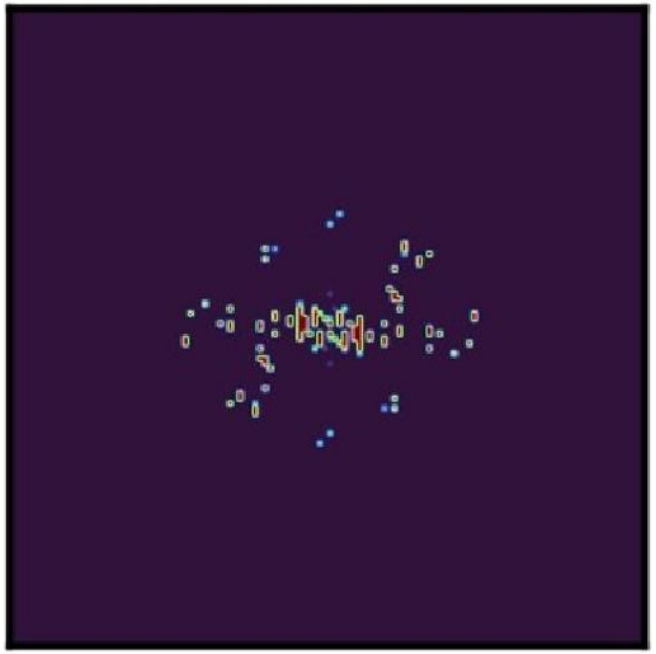


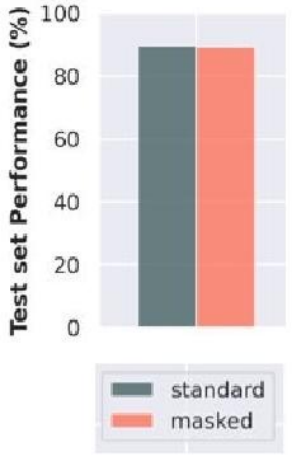
6, 644 image/label pairs: training set. 1, 934 pairs: validation set. For simplicity, we used grayscale versions.

Important Fourier features for single images



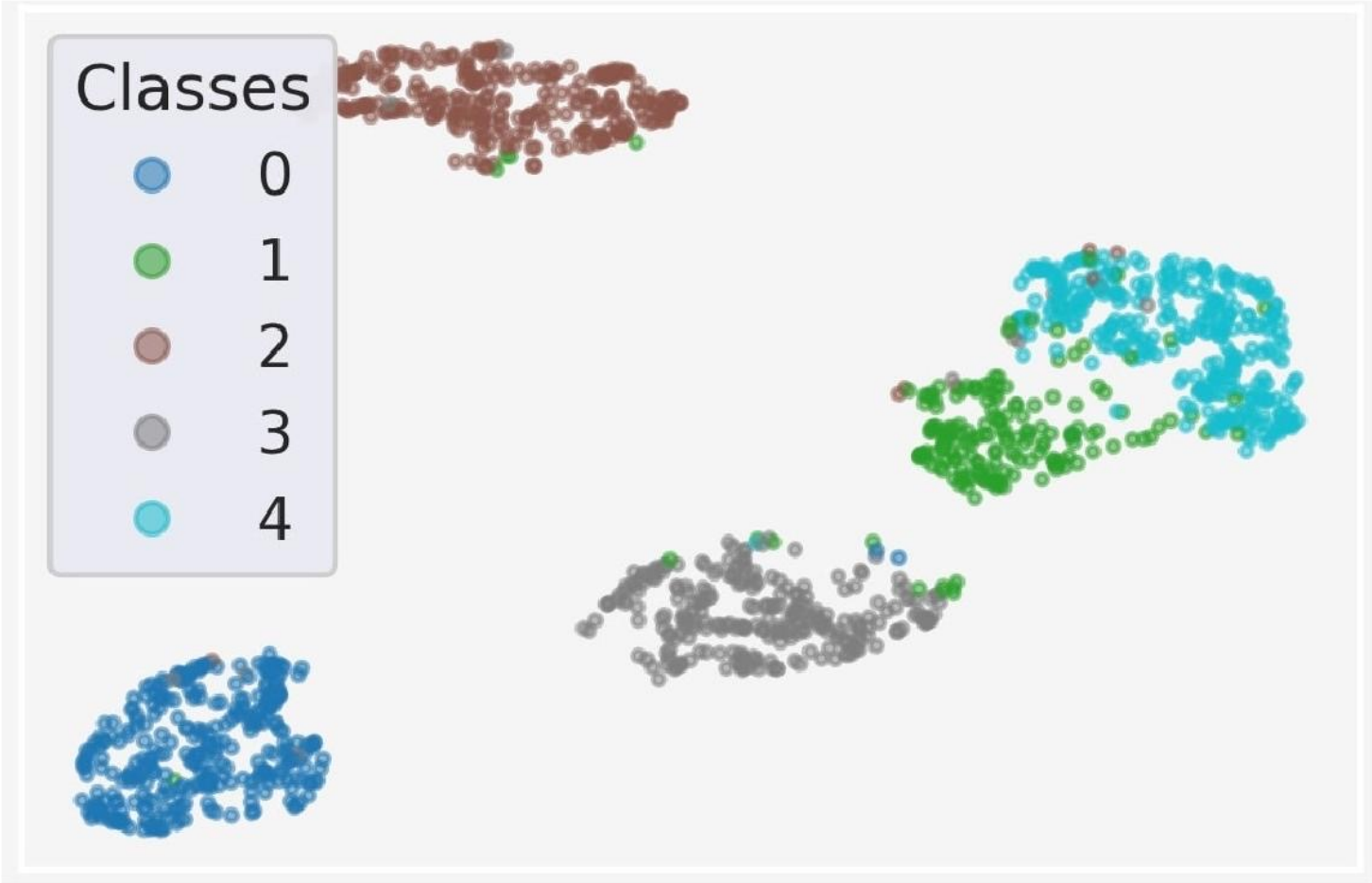
Single image Fourier masks





- Important frequencies are very sparse.
- The performance is preserved with the mask
- The performance drops 50% with complementary mask

Frequency clusters, UMAP



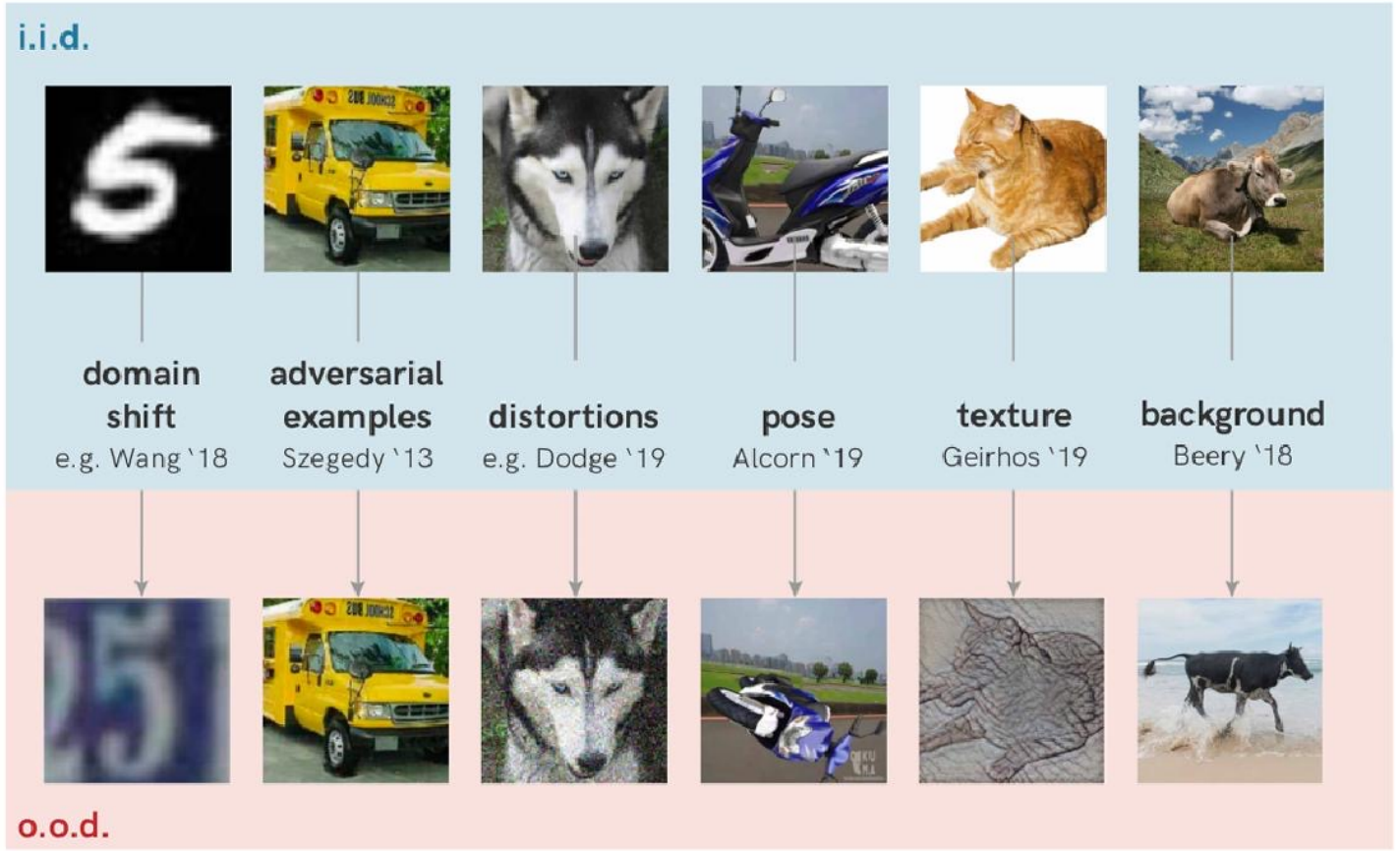
The Fourier representation separates well the data.

What is the network really looking at when solving a task? Filtered images

A: original images
B: images filtered with $J - M_\Phi$
C: images filtered with M_Φ

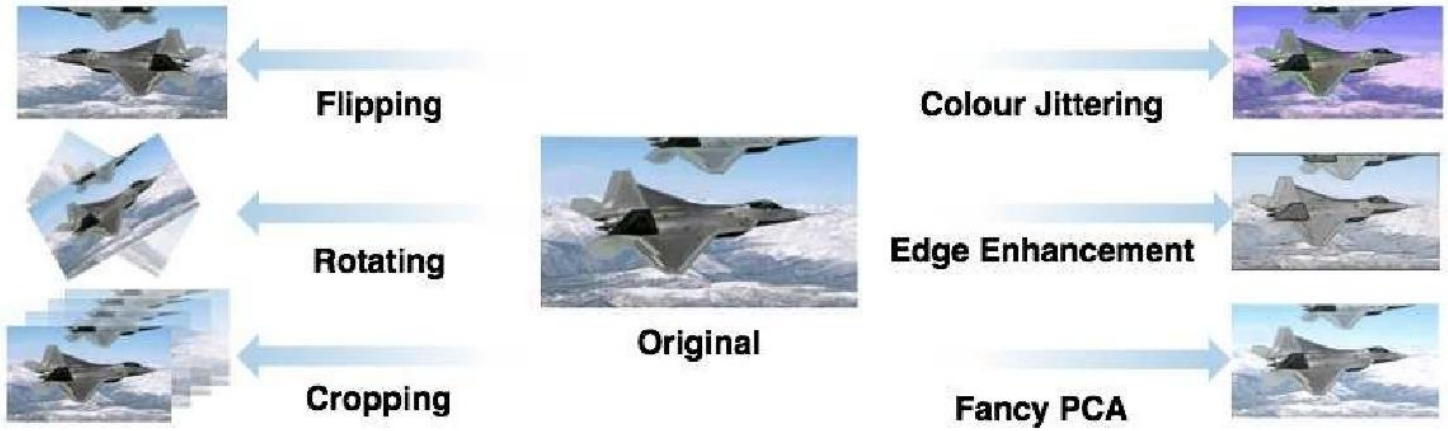
Filtered images are **not interpretable**. This will be another test for the Brain-score: what is the human/monkey/mouse performance on those images?

Human recognition is robust. Ann recognition is not.



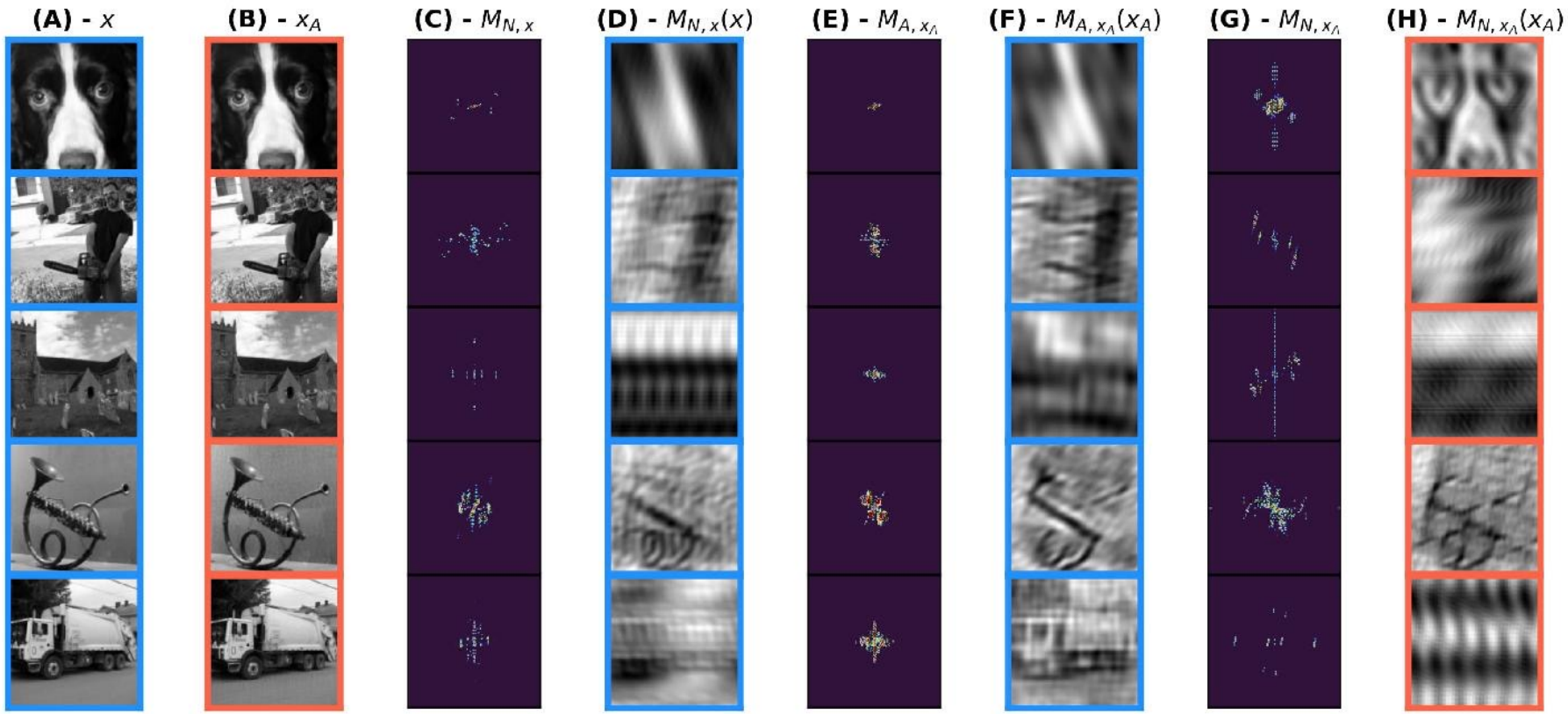
"Shortcut Learning in Deep Neural Networks" Robert Geirhos et al. 2021

Data augmentation



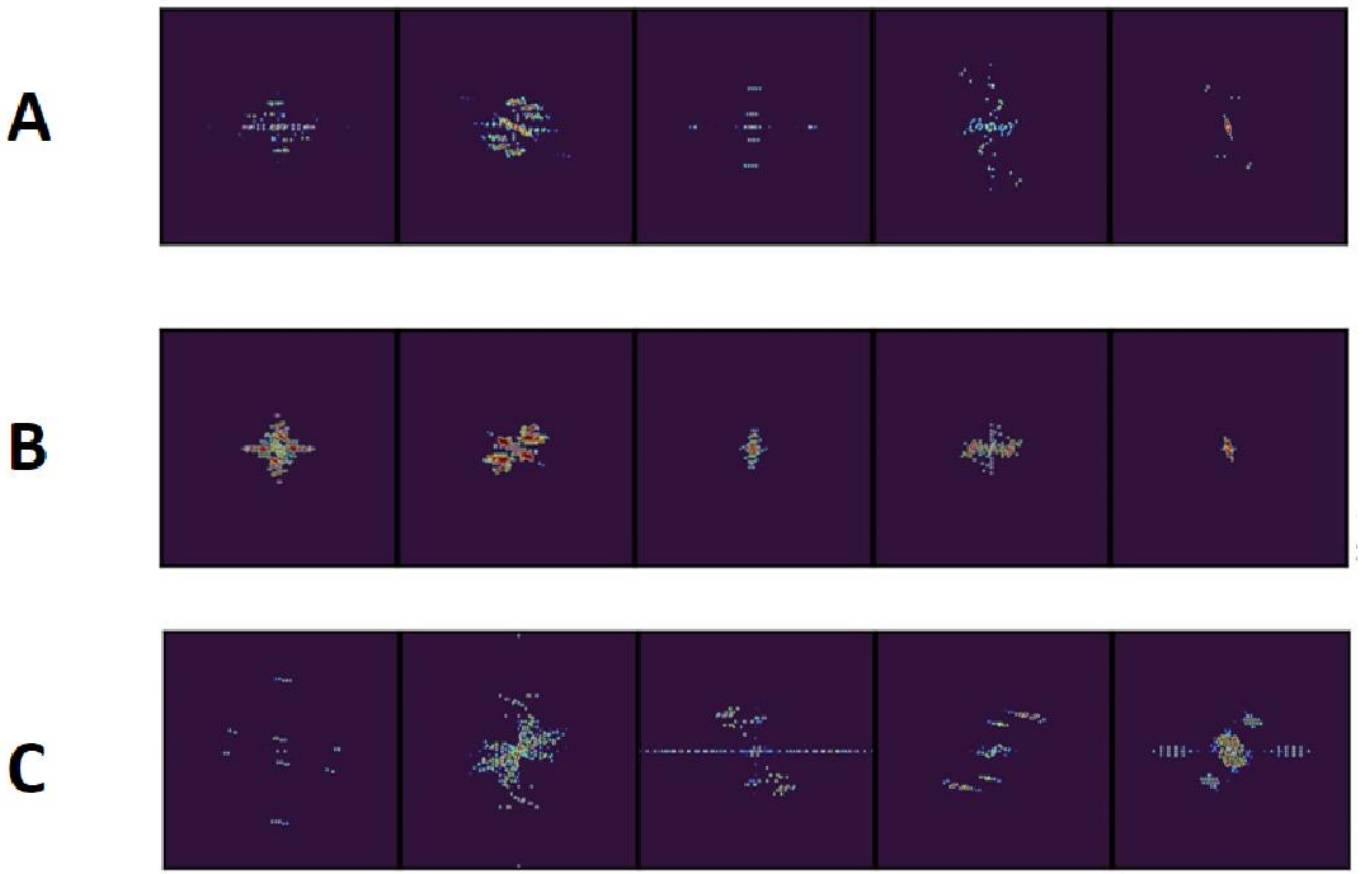
- What is the effect of data augmentation and adversarial training on Fourier important features?
- Common hypothesis: **adversarial training decrease high frequency features**.

Adversarial attacks: single image analysis



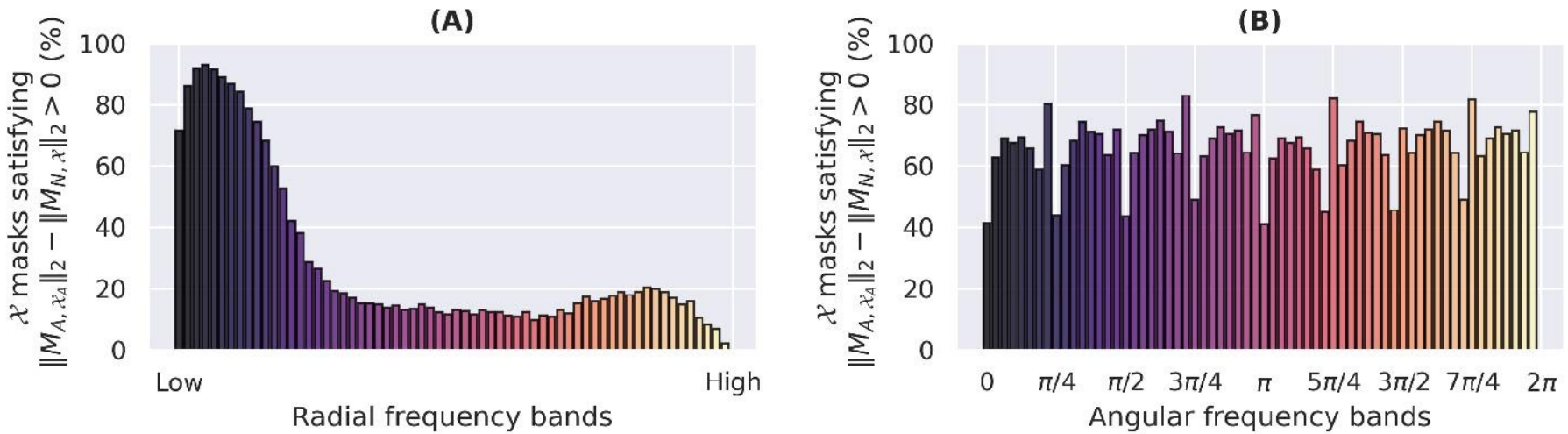
- A,B: original x and perturbed image x_A
- C, D: mask for the naive net on x and mask filtered image
- E, F: mask for the adversarially trained net on x_A and mask filtered image
- G, H: mask for the naive net on x and mask filtered image

Adversarial attacks and training: single image analysis



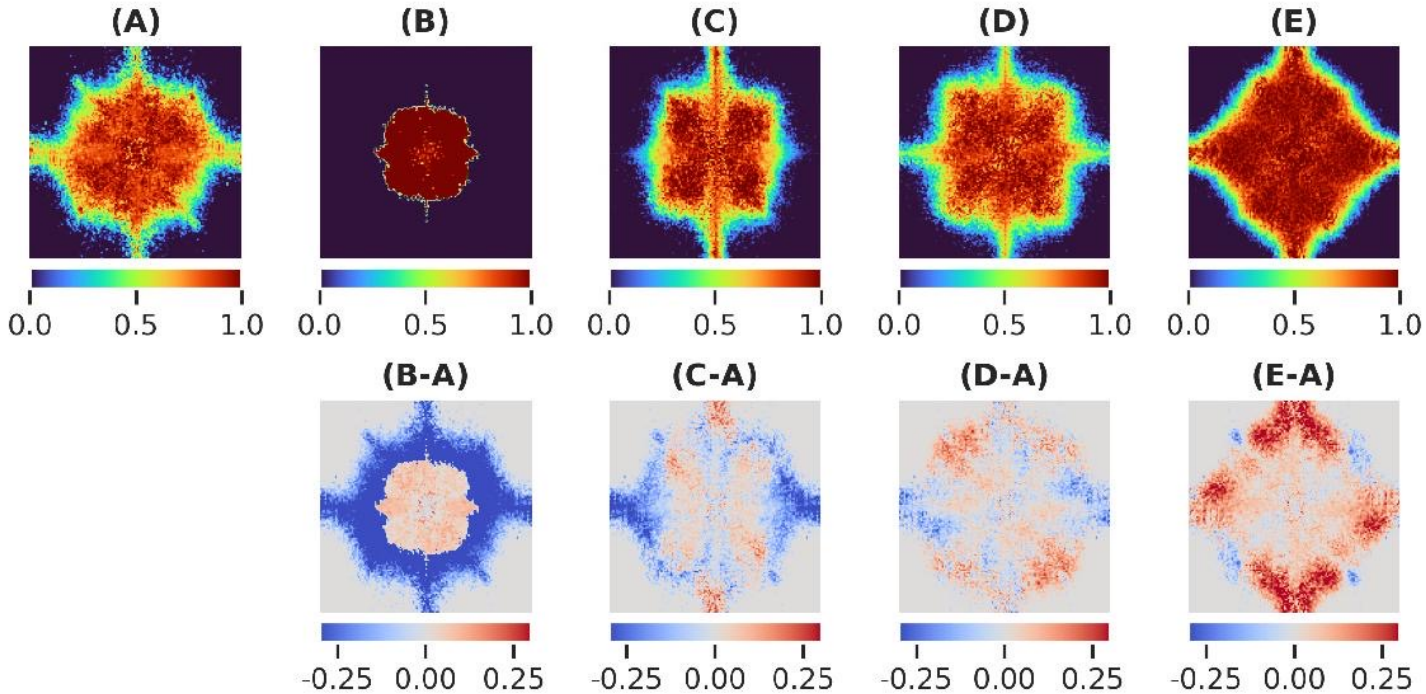
A: Frequencies for correct classification of the image
B: Frequencies for correct classification of the adversarial image with adversarially trained net
C: Frequencies for wrong classification of the adversarial image with naive net

Energy distribution



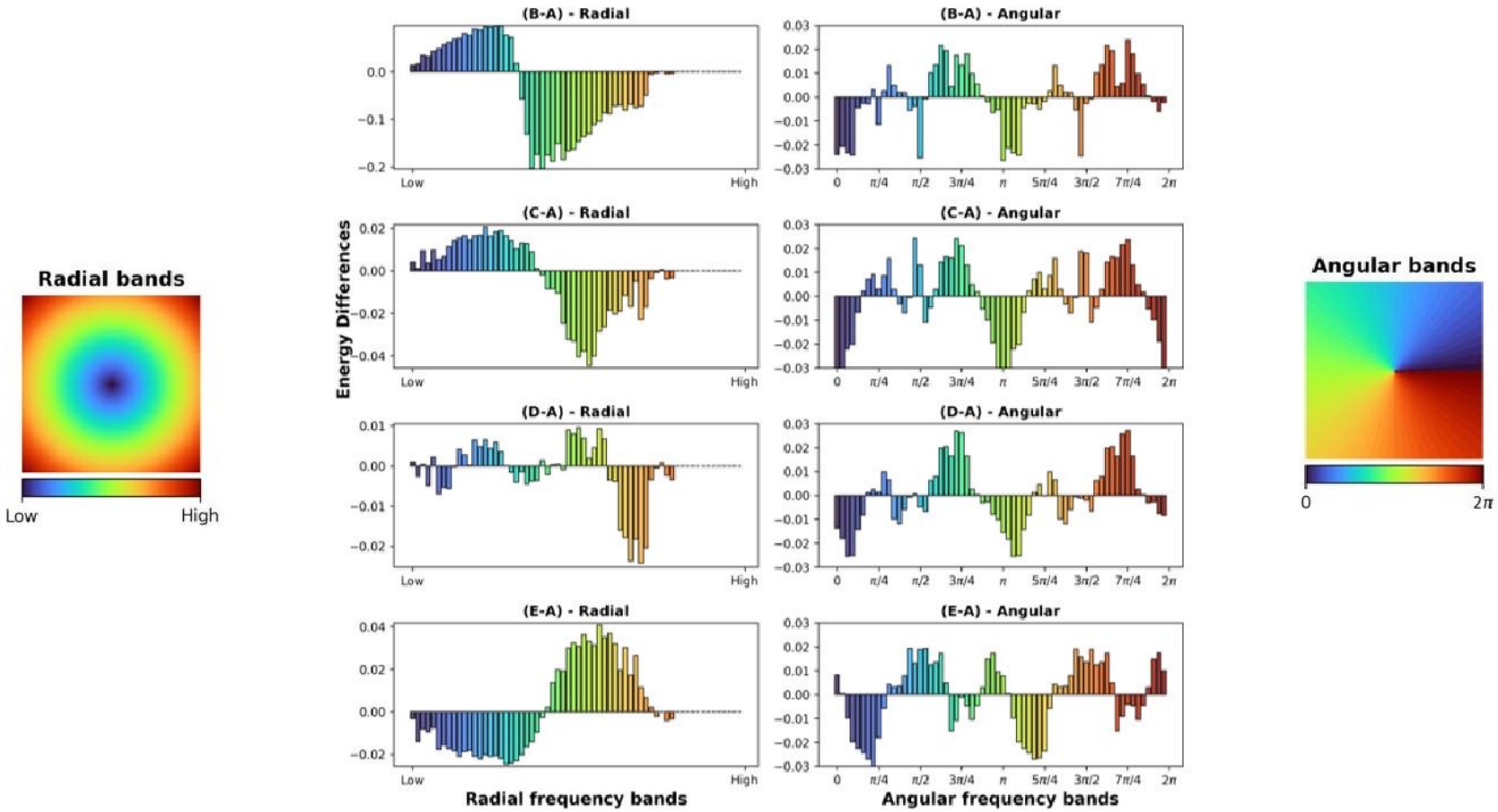
Lower frequencies are more important in the adversarially trained net w.r.t. the naive net.

Data augmentation (full dataset): testing the low-frequency bias



A: Naive
B: Adversarially trained
C: Scale augmentation
D: Translation augmentation
E: Rotation augmentation

Data augmentation (full dataset): testing the low-frequency bias



*Adversarial training and scale are biasing towards lower frequencies.
*The bias is highly direction dependent.

Where does this leave us?

- Although reaching good performance ANNs are relying on features that do not seem to be those the brain is relying on.
- Test "brain-like" Anns:
  1. Train an ANN to a recognition task that an animal also performs well.
  2. Find the essential frequencies and produce filtered images.
  3. Show back to the animal the filtered images and test for the animal performance.
- Generate "brain-like" Anns:
  1. Online learning of a frequency filter that maximize real neurons activity (how?)
  2. Learn a network to do e.g. recognition on mask-filtered images.