Analysis of Dominant Classes in Universal Adversarial Perturbations

Jon Vadillo[1,*], Roberto Santana[1], and Jose A. Lozano[1,2]

[1]University of the Basque Country UPV/EHU,
Faculty of Informatics,
Department of Computer Science and Artificial Intelligence,
Manuel Lardizabal 1, Donostia-San Sebastián, 20018 Gipuzkoa,
Spain.
[2]Basque Center for Applied Mathematics (BCAM),
Alameda de Mazarredo 14, 48009 Bilbao, Spain.
[*]Corresponding author.
Email Address: jon.vadillo@ehu.eus (Jon Vadillo)

Abstract

The reasons why Deep Neural Networks are susceptible to being fooled by adversarial examples remains an open discussion. Indeed, many different strategies can be employed to efficiently generate adversarial attacks, some of them relying on different theoretical justifications. Among these strategies, universal (input-agnostic) perturbations are of particular interest, due to their capability to fool a network independently of the input in which the perturbation is applied. In this work, we investigate an intriguing phenomenon of universal perturbations, which has been reported previously in the literature, yet without a proven justification: universal perturbations change the predicted classes for most inputs into one particular (dominant) class, even if this behavior is not specified during the creation of the perturbation. In order to justify the cause of this phenomenon, we propose a number of hypotheses and experimentally test them using a speech command classification problem in the audio domain as a testbed. Our analyses reveal interesting properties of universal perturbations, suggest new methods to generate such attacks and provide an explanation of dominant classes, under both a geometric and a data-feature perspective.

Keywords: Adversarial Examples, Universal Adversarial Perturbations, Deep Neural Networks, Robust Speech Classification

1 Introduction

Universal adversarial perturbations [1] are input-agnostic perturbations capable of fooling a Deep Neural Network (DNN) while remaining imperceptible for humans. These perturbations are generally created as *untargeted* attacks, so that no preference over the (incorrect) output class is assumed [1, 2, 3, 4]. However, previous work [1, 5, 6, 7] has reported a phenomenon regarding the effect of universal perturbations in the attacked model: the preference of the perturbation to change the class of the inputs into a particular *dominant* class, without this being specified or imposed in the generation of the perturbation. Thus, some classes (or class regions in the decision space) act as *attractors* under the effect of universal perturbations.

In this paper, we carry out, for the first time, an in-depth study of this phenomenon with the aim of sheding light on the (still misunderstood) vulnerability of DNNs to universal perturbations. The main contributions of our paper are the following:

- First, we propose a number of hypotheses to explain and characterize the existence of dominant classes linked to universal adversarial perturbations, and revisit previous hypotheses and open questions in the related work.

- We experimentally test the proposed hypotheses using a speech command classification task in the audio domain as a testbed. To the best of our knowledge, this is the first work in which the analysis of dominant classes is studied for the audio domain. Apart from providing evidence of the validity of the proposed hypotheses, our results reveal interesting properties of the DNN sensitivity to novel types of perturbations, such as perturbations optimized to prevent the main dominant classes.

- Overall, our study exposes the connection between the dominant classes and the sensitivity of the model to (I) patterns in the data distribution that the model recognizes as each class with high confidence, and (II) to *vulnerable* directions in the decision space learned by the model. Our findings also suggest novel approaches to generate universal perturbations, opening the venue for future research on more effective attacks and defenses.

- Finally, we highlight a number of differences between the image domain and the audio domain regarding the analysis of adversarial examples, contributing to a more general understanding of adversarial machine learning.

2 Related work

Universal adversarial perturbations for DNNs were introduced in [1] for image classification tasks. The goal of such perturbations is to fool a DNN for "most" natural inputs when they are applied to them, and, at the same time, to be imperceptible for humans. Formally, following the notation used in [8], a perturbation v is said to be (ξ, δ)-universal if the following conditions are satisfied:

$$||v||_2 \leq \xi, \tag{1}$$

$$\mathbb{P}_{x \sim \mu}\left[f(x+v) \neq f(x)\right] \geq 1 - \delta, \tag{2}$$

being μ the distribution of natural inputs in the d-dimensional input space $\mathbb{R}^d$, and $f(x)$ the output class assigned to an input x by a classifier $f : \mathbb{R}^d \rightarrow \{y_1, ..., y_k\}$. Thus, universal perturbations generalize *individual* (i.e., *input dependent*) adversarial perturbations [9, 10, 11, 12, 13], which are optimized to fool a DNN for one particular input of interest.

In the seminal work of Moosavi-Dezfooli et al. [1], an iterative procedure is proposed to generate the universal perturbations. This procedure accumulates *input dependent* perturbations [11] generated for a set of inputs, and projects the universal perturbation after every update in order to bound its norm. Subsequent works have proposed alternative approaches to generate universal adversarial perturbations, such as training generative networks to learn a distribution of universal adversarial perturbations (which, therefore, can be used to sample universal perturbations) [14, 15, 16], or *data-free* approaches capable of generating universal perturbations without any access to the data used to train the target models [2, 17, 18, 19]. Other works pursue more particular objectives, such as generating targeted universal perturbations which change the classification of the model to one predefined label [15, 19, 20], or perturbations that only fool the model for inputs of one particular class [21]. Finally, although image classification tasks have been the main focus of study, universal perturbations have also been reported for tasks such as image segmentation [18, 22], speaker recognition [23], speech recognition [4, 24] or text classification [7, 25].

The discovery of such attacks for state-of-the-art DNNs has led to a deeper study of their properties. In [1], the vulnerability of DNNs to universal perturbations is empirically studied in the image domain, which is attributed in part to the geometry of the decision boundaries learned by the DNNs. In particular, it is shown that, in the vicinity of natural inputs, perturbations normal to the decision boundaries are *correlated*, in the sense that they approximately span a low dimensional subspace (in comparison to the dimensionality of the input space). Thus, being

$$v_x = \arg\min_v ||v||_2 \quad \text{s.t.} \quad f(x) \neq f(x+v) \tag{3}$$

the minimal perturbation capable of changing the output of an input x (hence *normal* to the decision boundary at $x + v_x$), it is possible to find a subspace $S \subset X$, with $dim(S) \ll dim(X)$, so that $v_x \in S$ for $x \sim \mu$. The existence of such a subspace implies that even random perturbations (with small norms) sampled from S are likely to cause a misclassification for a large number of inputs [1]. This hypothesis is further developed in [8], also for the image domain, where the vulnerability of classifiers to universal perturbations is formalized, under the assumption of locally linear decision boundaries in the vicinity of natural inputs. An illustration of a linear approximation of the decision boundary is shown in Figure 1 (left).

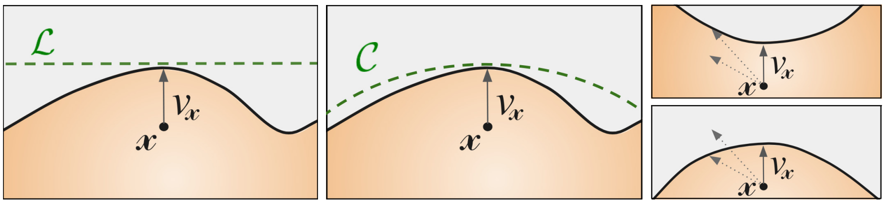


Figure 1: Illustration of the decision boundary approximations introduced in [8]. The left image illustrates the locally linear (flat) decision boundary model, and the middle figure the locally curved decision boundary model. The solid curve corresponds to the actual boundary, and the dashed lines to the approximations. Note that in both cases the approximations are estimated at $x + v_x$, being x an input sample and v_x a vector *normal* to the decision boundary (see Equation 3). The right images compare a positively curved boundary (bottom) with a negatively curved boundary (top) along v_x. Two dashed arrows have been included as reference in both images, to highlight that positively curved boundaries require smaller norms to be surpassed.

However, the assumption of locally linear decision boundaries becomes insufficient to comprehensively formalize the vulnerability of DNNs to universal perturbations. Indeed, there is a crucial connection between that vulnerability and the curvature of the decision boundaries [8]: there exist common perturbation directions (i.e., span a low-dimensional subspace) in the input space for which, starting from natural inputs, the decision boundaries are positively curved along these directions. See Figure 1 (right) for a comparison between a positively curved boundary and a negatively curved boundary. The positive curvature of the decision boundaries implies small upper bounds for the amount of perturbation required to surpass the decision boundaries, as depicted in Figure 1 (right). Thus, those positive curvatures increase the vulnerability of DNNs, as smaller perturbations are required to fool the model. At the same time, the fact that those directions are also *common* for multiple inputs implies the existence of small *input-agnostic* adversarial perturbations.

In a further analysis developed in [26], it is shown that the directions in the input space for which the decision boundaries are highly curved are indeed associated by the DNN with class identities (the further we move in one of such directions, the higher - or lower- the confidence of the model in one particular class is). Moreover, it is shown that the class *features*[1] associated to such directions are, indeed, the most relevant ones as far as the classification performance of the model is concerned, what links the accuracy of DNNs with their vulner-

[1] In this paper, unless specified, *features* are assumed to be abstract representations derived from *patterns* in the data distribution (e.g., how round the objects in an image are), rather than the set of individual *attributes* that characterize the data (e.g., the set of pixels of an image).

ability to adversarial attacks. A feature-perspective is also employed in [19] to justify the vulnerability of the models to universal perturbations, experimentally showing that universal perturbations contain features which predominate over the features of natural images. Thus, in the presence of universal perturbations, natural images act like noise, despite being visually predominant.

The aforementioned theoretical frameworks focus, in particular, on the vulnerability to universal perturbations. In this paper, we focus instead on one particular property of universal perturbations: the existence of *dominant* classes that are significantly more frequently predicted for the perturbed (and misclassified) inputs. This phenomenon was first reported in [1] for image classification tasks. Subsequent works have also reported the existence of dominant classes in image classification tasks [5, 6], and in text classification tasks [7]. In this paper, we show that this happens also for other domains, such as speech command classification tasks in the audio domain. Although it is hypothesized in [1] that a possible explanation for the *dominant* classes is that they occupy a larger region in the decision space, it is left as an open research question. In this paper, we tackle this research question and test multiple hypotheses in the search for a deeper understanding of this phenomenon.

Outside the particular field of universal perturbations, multiple theoretical frameworks have been proposed for the explanation of adversarial examples. Whereas most of them focus on the properties of the DNNs [9, 10, 27, 28], other alternative explanations have also been proposed. In this paper, special attention is paid to the one introduced in [29], in which adversarial examples are explained in terms of the *robustness* of the features in the data. In particular, it it shown that datasets contain non-robust features which, although being highly discriminative (i.e., that the data is well described by these features), are uncorrelated with the ground-truth classes when they are perturbed by small (adversarial) perturbations. Thus, when a classifier learns to rely on such non-robust features to accurately classify the data, it becomes vulnerable to adversarial perturbations. The small robustness of such features to small perturbations also implies their lack of meaning for humans, which explains the imperceptibility of the attacks. In our paper (Section 5.2), we hypothesize that the higher sensitivity of the model to certain features might explain the existence of dominant classes.

3 Proposed Framework

Let us consider a machine learning model $f : X \to Y$, with $X \subseteq \mathbb{R}^d$ and $Y = \{y_1, \dots, y_k\}$, trained to classify inputs $x \in X$ coming from a data distribution $x \sim \mu$ among one of the k possible classes in Y. To formally describe *dominant classes*, let us denote p_j^v the probability of misclassifying an input as the class y_j when a universal perturbation v is added to the inputs:

$$p_j^v = \mathbb{P}_{\substack{x \sim \mu \\ f(x) \neq y_j}} \left[f(x+v) = y_j \right]. \tag{4}$$

Similarly, let $t^v_{i,j}$ represent the probability that, departing from an input of ground-truth y_i, the model incorrectly predicts the class y_j for the perturbed inputs:

$$t^v_{i,j} = \mathbb{P}_{\substack{x\sim\mu \\ f(x)=y_i}} [f(x + v) = y_j]. \tag{5}$$

In practice, if the distribution μ is unknown, these probabilities can be estimated using a finite set of input samples $\mathcal{X}$.

Definition 1. *y_a is an attractor class for another class y_i ($i \neq a$), under a perturbation v, which will be denoted as $y_i \xrightarrow{v} y_a$, if at least the $\alpha > \frac{1}{k-1}$ proportion of the inputs corresponding to the class y_i are predicted as y_a when they are perturbed with v, that is:*

$$t^v_{i,a} \geq \alpha. \tag{6}$$

Notice that the threshold $\frac{1}{k-1}$ represents the proportion that would be achieved if the inputs were evenly distributed among the $k-1$ possible incorrect classes.

Definition 2. *y_b is a dominant class for the universal perturbation v if at least the $\beta > \frac{1}{k-1}$ proportion of the inputs are wrongly classified as y_b when they are perturbed with v, that is:*

$$p^v_b \geq \beta. \tag{7}$$

Alternatively, y_b can be defined also in terms of the number of classes that it attracts. Let $Y^v_b = \{y_i \in Y \mid y_i \xrightarrow{v} y_b\}$ represent the set of classes attracted by y_b with the perturbation v, and $|Y^v_b|$ the cardinality of the set Y^v_b. Precisely, y_b is dominant if it is an attractor class for at least the $\zeta > \frac{1}{k-1}$ proportion of the remaining classes:

$$\frac{|Y^v_b|}{k-1} \geq \zeta. \tag{8}$$

The choice of the parameters α, β and ζ can determine the existence of multiple attractor and dominant classes. In this paper, we assume $\alpha, \beta, \zeta \geq \frac{1}{3}$ since we are interested in those classes which are incorrectly predicted for a significant proportion of inputs, or which attract a significant proportion of other classes.

To explain the relationship between universal perturbations and dominant classes, we use a speech command classification problem in the audio domain as a testbed. We selected the Speech Command Dataset [30], in which the underlying task consists of classifying audio signals, of fixed length, into one of the following classes: *silence, unknown, yes, no, up, down, left, right, on, off, stop* and *go*.

We trained a convolutional neural network as a classifier, based on the architecture proposed in [31], which is composed of two convolutional layers with ReLU activations, a fully connected layer and a final softmax layer. This architecture has been used in a number of related works [30, 32, 33, 34]. The audio waveforms (in the time-domain) from the input space $\mathbb{R}^{16000}$, which take values in the range $[-1, 1]$, are first converted into spectrograms by dividing the audios into frames of 20ms, with a stride of 10ms, and applying the real-valued fast Fourier transform (retrieving 512 components) for each frame. As the frequency spectrum of a real signal is Hermitian symmetric, only the first 257 components are retained. The dimension of the resulting spectrogram is 99×257. Finally, the Mel-Frequency Cepstrum Coefficients (MFCCs) [35] are extracted from the spectrogram, in the space $\mathbb{R}^{99\times 40}$, before being sent to the network. It is worth pointing out that the adversarial perturbations that are generated for this model are optimized in an end-to-end fashion, directly in the audio waveform representation of the signal.

We selected the UAP-HC algorithm introduced in [4] to create the universal perturbations. This algorithm, which is a reformulation for the audio domain of the one proposed in [1], consists of iteratively accumulating individual untargeted adversarial perturbations, generated using the DeepFool algorithm [11]. The pseudocodes for both the UAP-HC and DeepFool algorithms can be found in Algorithm 1 and Algorithm 2, respectively. These algorithms have been generalized to (optionally) prevent them from reaching certain adversarial classes. This generalization will be further described and motivated in Section 4.

Finally, we highlight that the rationale of the DeepFool algorithm relies on a geometric approach. In particular, a first-order approximation of the decision boundaries is used to move the input towards the estimated closest boundary, being, therefore, an untargeted attack. Thus, the optimization process of the UAP-HC algorithm is not biased towards any particular class, although, in practice, different universal perturbations lead in most of the cases to the same dominant classes.

4 Dominant classes in speech command classification

In this section, we generate different universal adversarial perturbation for the speech command classification task described in Section 3, in order to investigate whether in this domain dominant classes are also produced.

We start by generating 10 different universal perturbations using the UAP-HC algorithm, without restricting any class ($\mathcal{R} = \varnothing$). We set $\xi = 0.1$ as threshold for the perturbation ℓ_2 norm, and restricted the UAP-HC algorithm to a maximum of five iterations. To generate the perturbations, we used a *training* set of 100 inputs per class, which makes a total of 1200 inputs. Once the perturbations are generated, their effectiveness will be measured in a *test* set, containing samples that were not used during the generation of the perturbations. The initial accuracy of the model in this set is 85.52%.[2]

According to the results, the algorithm led to universal perturbations with *left* and *unknown* as dominant classes for almost all the experiments. This can be seen in Figure 2 (top), which shows the frequency with which each class is

[2] The number of samples per class in the test set and the accuracy of the model in each class is reported in Table A.1.

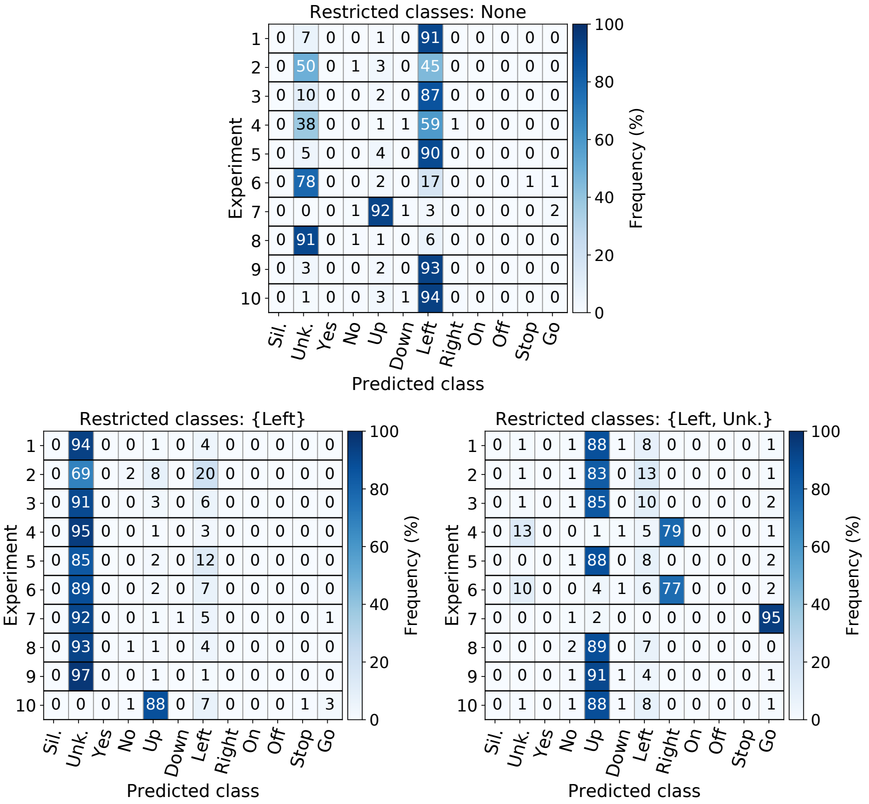
Algorithm 1 UAP-HC [4]

```
Input: A classification model $f$, a set of input samples $\mathcal{X}$, a projection operator
    $\mathcal{P}_{p,\xi}$, a fooling rate threshold $\delta$, a maximum number of iterations $I_{\max}$, a
    set of restricted classes $\mathcal{R} \subset Y$
Output: A universal perturbation $v$
v ← initialize with zeros
$FR \leftarrow 0$    ▷ Fooling rate.
$iter \leftarrow 0$    ▷ Iteration number.
while $FR < 1 - \delta \wedge iter < I_{\max}$ do
    $\mathcal{X} \leftarrow$ randomly shuffle $\mathcal{X}$
    for $x_i \in \mathcal{X}$ do
        ▷ Check that $x_i$ is not already fooled by $v$:
        if $f(x_i + v) = f(x_i)$ then
            $\triangle v_i \leftarrow \text{DeepFool}(x_i + v, f, \mathcal{R})$
            $v' \leftarrow \mathcal{P}_{p,\xi}(v + \triangle v_i)$    ▷ Project $(v + \triangle v_i)$ in the $\ell_p$ ball of radius $\xi$
            and centered at 0.
            $FR' \leftarrow \mathbb{P}_{x \in \mathcal{X}} [f(x) \neq f(x + v')]$
            ▷ Update $v$ only if adding $\triangle v_i$ increases the FR and if the current
            class is not in $\mathcal{R}$:
            if $FR < FR' \wedge f(x_i + v + \triangle v_i) \notin \mathcal{R}$ then
                $v \leftarrow v'$
                $FR \leftarrow FR'$
            end if
        end if
    end for
    $iter \leftarrow iter + 1$
end while
```

wrongly predicted when the perturbation is applied to the audios in the test set. We only considered those inputs that were initially correctly classified by the model, but misclassified when the perturbation is applied. The frequencies are shown individually for the ten universal perturbations, with each row corresponding to one perturbation. As can be seen, both *left* and *unknown* arise as dominant classes in 9 of the 10 experiments, sometimes even at the same time.

It is important to highlight that dominant classes arise without being imposed in the universal perturbation crafting procedure. For this reason, an interesting property to study is whether dominant classes remain dominant even if we explicitly avoid them during the optimization process (see Algorithms 1 and 2). To shed light on this question, we start by preventing the algorithm from considering those directions that point to the decision boundaries of the class *left*. The results obtained for ten new perturbations generated with this restriction are shown in Figure 2 (bottom left). As can be seen, the most frequent adversarial class is now *unknown* for 9 of the 10 perturbations created.

We went another step further and repeated the experiment, this time, how-



Restricted classes: None

Experiment	Sil.	Unk.	Yes	No	Up	Down	Left	Right	On	Off	Stop	Go
1	0	7	0	0	1	0	91	0	0	0	0	0
2	0	50	0	1	3	0	45	0	0	0	0	0
3	0	10	0	0	2	0	87	0	0	0	0	0
4	0	38	0	0	1	1	59	1	0	0	0	0
5	0	5	0	0	4	0	90	0	0	0	0	0
6	0	78	0	0	2	0	17	0	0	0	1	1
7	0	0	0	1	92	1	3	0	0	0	0	2
8	0	91	0	1	1	0	6	0	0	0	0	0
9	0	3	0	0	2	0	93	0	0	0	0	0
10	0	1	0	0	3	1	94	0	0	0	0	0

Restricted classes: {Left}

Experiment	Sil.	Unk.	Yes	No	Up	Down	Left	Right	On	Off	Stop	Go
1	0	94	0	0	1	0	4	0	0	0	0	0
2	0	69	0	2	8	0	20	0	0	0	0	0
3	0	91	0	0	3	0	6	0	0	0	0	0
4	0	95	0	0	1	0	3	0	0	0	0	0
5	0	85	0	0	2	0	12	0	0	0	0	0
6	0	89	0	0	2	0	7	0	0	0	0	0
7	0	92	0	0	1	1	5	0	0	0	0	1
8	0	93	0	1	1	0	4	0	0	0	0	0
9	0	97	0	0	1	0	1	0	0	0	0	0
10	0	0	0	1	88	0	7	0	0	0	1	3

Restricted classes: {Left, Unk.}

Experiment	Sil.	Unk.	Yes	No	Up	Down	Left	Right	On	Off	Stop	Go
1	0	1	0	1	88	1	8	0	0	0	0	1
2	0	1	0	1	83	0	13	0	0	0	0	1
3	0	1	0	1	85	0	10	0	0	0	0	2
4	0	13	0	0	1	1	5	79	0	0	0	1
5	0	0	0	1	88	0	8	0	0	0	0	2
6	0	10	0	0	4	1	6	77	0	0	0	2
7	0	0	0	1	2	0	0	0	0	0	0	95
8	0	0	0	2	89	0	7	0	0	0	0	0
9	0	0	0	1	91	1	4	0	0	0	0	1
10	0	1	0	1	88	1	8	0	0	0	0	1

Figure 2: Overview of the frequency with which each class was assigned to the inputs misclassified as a consequence of universal perturbations. The frequencies have been computed individually (row-wise) for the 10 perturbations generated in each of the following configurations of the UAP-HC algorithm: default algorithm (top), restricting the algorithm to follow the class *left* (bottom left) and restricting the algorithm to follow the classes *left* and *unknown* (bottom right).

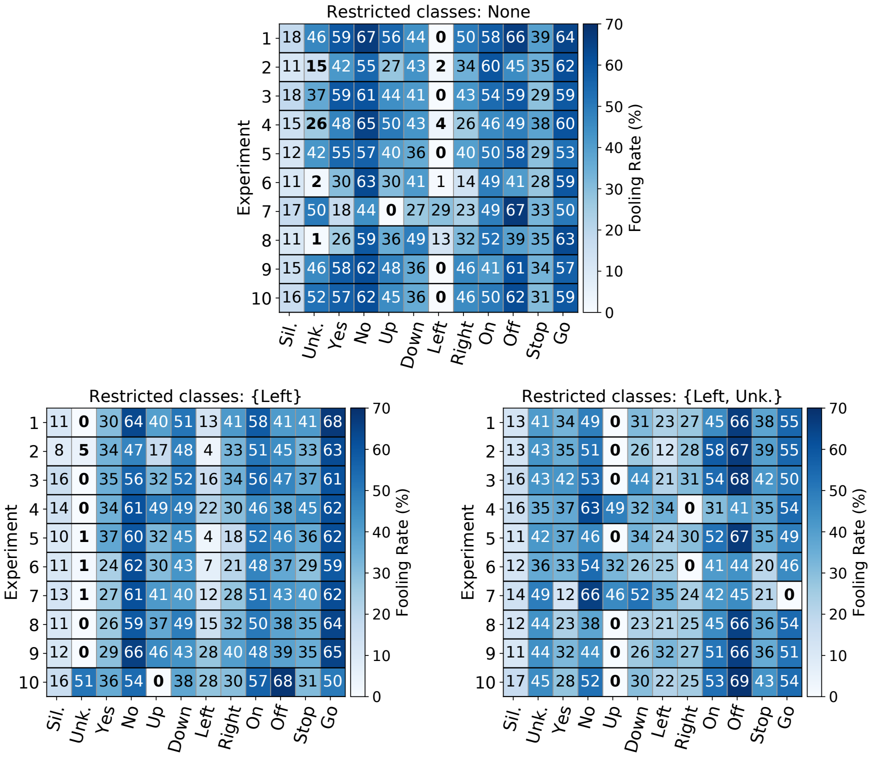


Figure 3: Fooling rate percentage, computed individually for each class, of the 10 perturbations generated in each of the following configurations of the UAP-HC algorithm: default algorithm (top), restricting the algorithm to follow the class *left* (bottom left) and restricting the algorithm to follow the classes *left* and *unknown* (bottom right). In the three figures, the results corresponding to the dominant classes (for each experiment) have been highlighted using bold text.

Algorithm 2 DeepFool [11]

Input: An input sample x of class y_i, a classifier f, a set of restricted classes $\mathcal{R} \subset Y$.

Output: An individual perturbation r.

```
$x' \leftarrow x$
$r \leftarrow$ initialize with zeros
$Y' \leftarrow Y - (\mathcal{R} \cup \{y_i\})$
while $f(x') = y_i$ do
    for $y_j \in Y'$ do
        $f'_j \leftarrow f_j(x') - f_i(x')$
        $w'_j \leftarrow \bigtriangledown f_j(x') - \bigtriangledown f_i(x')$
    end for
    $l \leftarrow \arg\min_{j \in Y'} \frac{|f'_j|}{||w'_j||}$
    $r \leftarrow r + \frac{|f'_l|}{||w'_l||_2^2} w'_l$
    $x' \leftarrow x + r$
end while
```

ever, restricting the boundaries corresponding to both *left* and *unknown* classes. The results are shown in Figure 2 (bottom right). In this case, the two restricted classes were no longer dominant classes, but different dominant classes were obtained, precisely, *up*, *right* and *go*. It is also worth emphasizing that, although dominant classes were obtained in all the experiments, they were different depending on which other classes were restricted. For instance, whereas the class *up* rarely appeared as dominant without restrictions, it is the most frequent dominant class when both *left* and *unknown* classes are restricted.

Regarding the effectiveness of the attacks, the fooling rate of every perturbation (i.e., the percentage of inputs that are misclassified when the perturbation is applied) is shown in Figure 3, for each class independently. The fooling rates have been computed considering the inputs that were initially correctly classified. As can be seen, the effectiveness of each perturbation is higher in some classes than in others, achieving up to ≈69% in some cases. The fooling rates corresponding to the dominant classes, which have been highlighted in the figure, are practically zero for most of the perturbations, which reveals that the perturbation does not change the prediction of the model for those inputs.

For more informative results, the mean and maximum fooling rate of all the perturbations are shown in Table 1. To avoid biases, these aggregated fooling rates have been computed in three different ways: (I) considering all the inputs, (II) without considering the inputs corresponding to the dominant classes, and (III) without considering the dominant classes and the class *silence*. The reason for not considering the inputs belonging to the dominant classes is because the perturbation reinforces the confidence on those classes, and, as a consequence, there are practically no misclassifications in those inputs. On the contrary, the results for the class *silence* are clearly lower than for the rest of the classes, which biases the results. Comparing the average effectiveness of the

Restricted classes in UAP-HC	Fooling Rate					
	Considering all the classes		w/o considering dominant classes		w/o considering dominant & *Silence*	
	Mean	Max.	Mean	Max.	Mean	Max.
None	37.94	46.34	41.68	50.84	44.97	54.76
{*Left*}	34.90	37.73	37.39	40.60	40.32	43.71
{*Left*, *Unk.*}	33.75	37.49	37.08	41.36	39.90	44.37

Table 1: Effectiveness of the UAP-HC algorithm in a set of *test* samples, not seen during the generation of the perturbations.

universal perturbations, we can notice that the average fooling rate achieved by the perturbations decreases when the dominant classes are restricted in the UAP-HC algorithm. We confirmed using the Wilcoxon signed-rank test [36] (with a significance level of 0.05) that, in comparison to the results obtained when no class is restricted (i.e., $\mathcal{R} = \varnothing$), the decrease is significant when the set of classes $\mathcal{R} = \{Left\}$ or $\mathcal{R} = \{Left, Unknown\}$ is restricted. According to the same test, the differences observed between the cases in which the sets of restricted classes are $\mathcal{R} = \{Left\}$ and $\mathcal{R} = \{Left, Unknown\}$ were not statistically significant.

Overall, these results confirm the existence of dominant classes in audio tasks, and reveal a number of properties that, to the best of our knowledge, have not been reported before in related works. First, we have shown that it is possible to prevent one class from being dominant during the optimization of the universal perturbation. However, doing so leads to different dominant classes. Moreover, the fact that the effectiveness of the universal perturbations decreases when the *most frequent* dominant classes are restricted might suggest that some classes are more *dominant* than others. All these findings and properties will serve as a basis to further study the cause of this phenomenon in the following sections.

5 Hypotheses about the existence dominant classes

In this section, we propose a number of hypotheses to explain and characterize the relationship between universal adversarial perturbations and dominant classes. The proposed hypotheses are also experimentally tested using the framework described in Section 3.

5.1 Dominant classes occupy a larger region in the input space

In [1], the existence of dominant classes is attributed to a larger region of such classes in the image space. Nevertheless, due to the high dimensionality of the input spaces in current machine learning problems, exploring the volume that each decision region occupies in the whole input space is intractable in practice.

Even so, to test this hypothesis, we randomly sampled and classified 1,000,000 inputs from the input space. The values of the inputs were sampled uniformly at random in the range $[-1, 1]$. We found that all the samples were classified as the class *silence*, which is not a dominant class in our experiments, as shown in Section 4 (see Figure 2). Therefore, our results suggest that there is not necessarily a connection between the *volume* occupied by the decision regions of different classes and the frequency with which inputs perturbed by universal perturbations reach the regions corresponding to the dominant classes.

5.2 Class properties of universal perturbations

Universal perturbations are capable of changing the output class of a large number of inputs, and the majority of the misclassified inputs are moved unintentionally towards a dominant class. In this section, we show that the perturbation itself is predicted by the model as the dominant class with high confidence.

In fact, we noticed that the following three factors are positively correlated during the generation process of a universal perturbation v: the fooling rate ($\mathcal{F}_1$), the percentage of inputs misclassified as the dominant class y_b ($\mathcal{F}_2$), and the confidence with which the model considers that the perturbation belongs to the dominant class ($\mathcal{F}_3$):[3]

$$\mathcal{F}_1(v) = \mathbb{P}_{x \in \mathcal{X}} \left[f(x) \neq f(x+v) \right], \tag{9}$$

$$\mathcal{F}_2(v) = \mathbb{P}_{x \in \mathcal{X}} \left[f(x+v) = y_b \right], \tag{10}$$

$$\mathcal{F}_3(v) = f_b(v), \tag{11}$$

where $\mathcal{X}$ is a set of inputs and $f_j : X \to \mathbb{R}$ represents the output confidence of the classifier f corresponding to the class y_j. An example of the evolution of these factors during the optimization process of a universal perturbation, using the UAP-HC algorithm, is shown in Figure 4. These results correspond to the first experiment of Section 4, for the case in which no class was restricted. In particular, the left figure shows the evolution of the frequency with which each class is (wrongly) predicted for the misclassified inputs, and the right figure shows the output confidences of the model when the universal perturbation is classified. The fooling ratio of the perturbation has been included in both figures as a reference, represented by a dashed line.

More generally, for the 10 different universal perturbations generated in Section 4 (without restricting any class), the average Pearson correlation coefficient between $\mathcal{F}_1$ and $\mathcal{F}_3$ during the first iteration of Algorithm 1 is 0.79. Similarly, the average correlation between $\mathcal{F}_1$ and $\mathcal{F}_2$ is 0.87, and the average correlation between $\mathcal{F}_2$ and $\mathcal{F}_3$ is 0.91. These results confirm that the three factors

[3] For those perturbations in which there are two dominant classes at the same time, the class $f(v)$ has been considered as the dominant (i.e., the class assigned to the perturbation by the model).

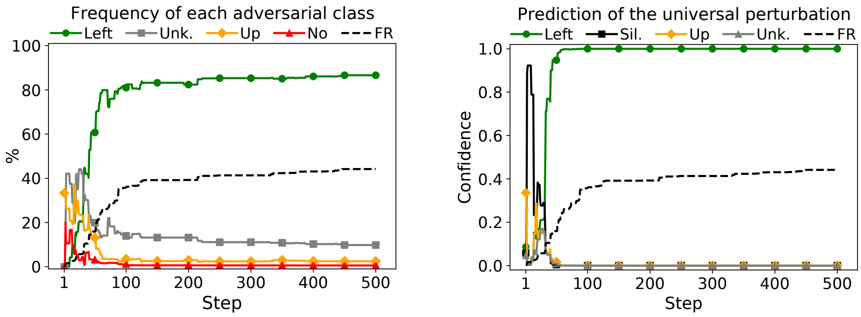


Figure 4: Evolution of three different factors during the optimization process of a universal perturbation using the UAP-HC algorithm: the frequency with which the inputs are classified as the dominant class (left), the confidence of the model in the dominant class when the perturbation is predicted (right), and the evolution of the fooling ratio (FR), which is shown in both plots as a reference. These results have been computed on the training set, and correspond to the first experiment reported in Section 4, for the case in which no class was restricted. For the sake of clarity, only the information of the four most relevant classes are plotted in each plot.

are being maximized jointly during the optimization process of the universal perturbation, even if such behavior is not specified by the model.

Motivated by this finding, we studied whether any perturbation v that is classified by the model as one particular class with high confidence is capable of producing the same effect as a universal perturbation, that is, to force the misclassification of a large number of inputs by pushing them to the class $f(v)$. For this purpose, we defined the following optimization problem, in which the objective is to find a perturbation v, with a constrained norm, that maximizes the confidence of the model in one particular class y_t, $f_t(v)$, that is:

$$\max_{v} \quad f_t(v) \quad \text{s.t.} \quad ||v||_2 \leq \xi. \tag{12}$$

We launched 100 trials for each possible target class, starting from random perturbations.[4] We used a gradient descent approach to optimize the perturbation, restricting the search to 100 gradient descent iterations, and setting a threshold of $\xi = 0.1$ for the perturbation norm.

The mean and maximum fooling rates obtained with the generated perturbations are shown in Table 2, computed independently for each target class. The fooling rate for each class individually is shown in Figure 5 (left). As can be seen in Table 2, for the classes *left* and *unknown*, both the most frequent dominant classes associated to the universal perturbations generated using the UAP-HC algorithm (see Figure 2), a significantly higher effectiveness is achieved than for

[4] The initial perturbations were randomly sampled from the input space $\mathbb{R}^{16000}$, where each value was sampled uniformly at random in the range $[-10^{-3}, 10^{-3}]$.

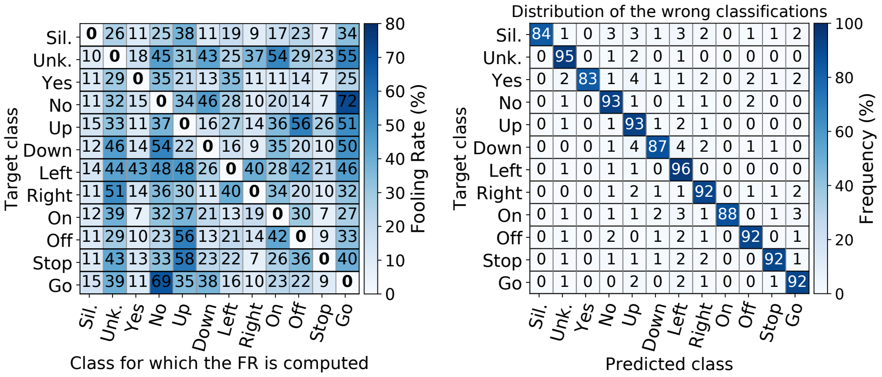


Figure 5: Overview of the effectiveness of the perturbations found by solving the optimization problem defined in (12). In both figures, the results are reported independently for each target class (row-wise), and are averaged for the 100 trials generated for each target class. Left: average fooling rate obtained by the 100 perturbations found for each target class, computed for each class individually. Right: Average frequency with which each class is wrongly assigned to the fooled inputs by the model.

the rest of classes. We confirmed this using the Wilcoxon signed-rank statistical test [36], under a significance level of 0.05. Apart from that, with independence of the target class, the majority of the samples fooled by these perturbations were classified as the target class. This is shown in Figure 5 (right), in which the average frequency with which each class is predicted under the effect of the perturbations is computed, independently for each target class.

These results reveal that a perturbation which is optimized only to maximize the confidence of a model into one class can behave as a universal perturbation, and, more relevantly, that their effectiveness is maximized when the target class is a dominant class. Based on these findings, we can hypothesize that the model is more sensitive to some class *features* than to others, and that, ultimately, the sensitivity degree to each class feature is what determines the dominant classes. In other words, a class y_j will have a greater dominance the more sensitive the model is to the patterns in the data distribution that are associated to y_j (by the model itself)[5].

[5] These results are consistent with previous explanations proposed for the vulnerability of universal adversarial perturbations. For instance, these results could be related to the non-robust data-feature framework introduced in [29], to the predominance of the features of universal perturbations over the features of natural inputs [19], or to the link between the class-identity associations of the model and the most vulnerable directions in the input space studied in [26] (see Section 2 for more details).

Target class	Fooling Rate					
	Considering all the classes		w/o considering dominant classes		w/o considering dominant & *Silence*	
	Mean	Max.	Mean	Max.	Mean	Max.
Sil.	17.85	21.71	19.77	24.05	19.77	24.05
Unk.	30.31	33.88	32.40	36.21	35.00	39.14
Yes	16.91	20.40	18.67	22.52	19.59	23.89
No	23.46	25.82	25.28	27.84	26.91	29.74
Up	25.53	28.19	28.16	31.10	29.79	32.97
Down	22.56	24.68	24.45	26.75	25.95	28.28
Left	32.57	37.25	35.73	40.87	38.37	44.22
Right	23.25	27.28	25.38	29.78	27.07	31.88
On	19.50	22.43	21.25	24.45	22.40	25.94
Off	21.56	24.46	23.39	26.54	24.83	28.48
Stop	25.07	27.21	27.61	29.97	29.64	32.32
Go	22.99	25.66	24.84	27.72	26.03	29.24

Table 2: Effectiveness of the perturbations generated using Algorithm 12, averaged for the 100 perturbations generated for each target class.

5.3 Singular Value Decomposition

In [1], the existence of universal perturbations for image classification DNNs is attributed, in part, to the presence of similar patterns in the geometry of decision boundaries around different points of the decision space. In particular, as described in Section 2, perturbations *normal* to the decision boundaries in the vicinity of natural inputs approximately span a very low-dimensional subspace, revealing that *similar* perturbations are capable of changing the output class of different input samples. This was assessed experimentally for state-of-the-art DNNs, by computing the Singular Value Decomposition (SVD) of a matrix A collecting normalized individual untargeted perturbations generated using the DeepFool algorithm. The SVD provides a set of *singular vectors* $\{s_1, s_2, \ldots, s_r\}$, which represent a basis for the subspace spanned by the adversarial perturbations in A. Each s_i is related to a *singular value* σ_i, which indicates the *importance* or contribution of that singular vector. As shown in [1], considering that the singular values are arranged in decreasing order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, the decay of the singular values was considerably faster in comparison to the decay obtained from the SVD of random perturbations (sampled from the unit sphere). This implies that the subspace spanned just by the first $d' \ll d$ singular vectors (i.e., those corresponding to the highest singular values) contained vectors normal to the decision boundaries in the vicinity of natural samples. Indeed, random perturbations sampled from such a subspace were capable of achieving a fooling rate of nearly 38% on unseen inputs, whereas random perturbations (of the same norm) in the input space only achieved a fooling rate of approximately 10% [1].

In this section, we take this approach as a framework to study the existence of dominant classes. First, we will replicate the previous experiment to assess whether, in the audio domain, it is also possible to find a low-dimensional subspace of the input space collecting vectors normal to the decision boundaries of DNNs. The existence of such a subspace would allow us to test a number of hypotheses, for example, whether the directions in such subspaces mainly point towards the decision boundaries corresponding to the dominant classes. This would explain why most of the inputs are (incorrectly) classified as the dominant class when they are adversarially perturbed.

Nevertheless, due to the input transformation process required to convert the raw audio signal into the MFCC representation (see Section 3), the results might differ depending on the data representation in which the analysis is done. For this reason, we need to assess first which audio representation is the most informative one in our case. Thus, we computed the SVD for a set of individual perturbations and different sets of random perturbations, under the three main representations for audio signals: raw audio waveform, spectrogram and MFCC coefficients.

5.3.1 Analysis of the SVD of audio perturbations

Let us consider a set of n natural input samples $\mathcal{X} = \{x_1, \dots, x_n\}$. The individual perturbations were generated using the DeepFool algorithm, in the raw audio waveform representation:

$$\mathcal{V} = \{v_i \mid v_i = \text{DeepFool}(x_i),\ i = 1, \dots, n\}\,. \tag{13}$$

The perturbations that these raw waveforms produce in both the spectrogram and MFCC representations are computed as $v_i' = g(x_i + v_i) - g(x_i)$, being g the input transform function, which maps the raw audio waveforms into either a spectrogram or the MFCC features:

$$\mathcal{V}_{\text{SPEC}} = \{v_i^{\text{spec}} \mid v_i^{\text{spec}} = g_{\text{SPEC}}(x_i + v_i) - g_{\text{SPEC}}(x_i),\ i = 1, \dots, n\}\,, \tag{14}$$

$$\mathcal{V}_{\text{MFCC}} = \left\{v_i^{\text{mfcc}} \mid v_i^{\text{mfcc}} = g_{\text{MFCC}}(x_i + v_i) - g_{\text{MFCC}}(x_i),\ i = 1, \dots, n\right\}. \tag{15}$$

The random perturbations were sampled uniformly at random from the raw input space:

$$\mathcal{R} = \left\{r_i \mid\ r_i \text{ is sampled u.a.r. from } [-1, 1]^{16000},\ i = 1, \dots, n\right\}. \tag{16}$$

As in the case of adversarial perturbations, the corresponding perturbations in the frequency-domain representation are computed as:

$$\mathcal{R}_{\text{SPEC}} = \{r_i^{\text{spec}} \mid r_i^{\text{spec}} = g_{\text{SPEC}}(x_i + r_i) - g_{\text{SPEC}}(x_i),\ i = 1, \dots, n\}\,, \tag{17}$$

$$\mathcal{R}_{\text{MFCC}} = \left\{r_i^{\text{mfcc}} \mid r_i^{\text{mfcc}} = g_{\text{MFCC}}(x_i + r_i) - g_{\text{MFCC}}(x_i),\ i = 1, \dots, n\right\}. \tag{18}$$

In this case, the random perturbations were scaled to have a fixed ℓ_2 norm of 0.1 before being applied to the inputs in Equations (17) and (18).

Finally, for a more representative analysis, we considered two additional sets of random perturbations, sampled uniformly at random from the space of spectrograms and the space of MFCC coefficients:

$$\mathfrak{R}_{\text{SPEC}} = \left\{ \mathfrak{r}_i \mid \mathfrak{r}_i \text{ is sampled u.a.r. from } [-1,1]^{99\times 257},\ i = 1,\dots,n \right\}, \tag{19}$$

$$\mathfrak{R}_{\text{MFCC}} = \left\{ \mathfrak{r}_i \mid \mathfrak{r}_i \text{ is sampled u.a.r. from } [-1,1]^{99\times 40},\quad i = 1,\dots,n \right\}. \tag{20}$$

All the perturbations described in Equations (13)-(20) were normalized before computing the SVD. It is worth highlighting the key difference between the random perturbations defined in (17) and (18) and those defined in (19) and (20). The former represent the changes that randomly perturbing a raw signal produces on the spectrogram (or MFCC) space. In contrast, the random perturbations in (19) and (20) are directly generated in the spectrogram space or in the MFCC space, respectively. In other words, the perturbations considered in (19) and (20) are analogous to those in (13), but in the spaces corresponding to the spectrograms or to the MFCC coefficients instead of the space of raw audio waveforms. Considering all these types of perturbations and representations is important to better study which of them are the most informative ones in the audio domain, and to ensure that our subsequent analyses will be carried out using the most appropriate representation.

Figure 6 compares the decay of the singular values (sorted in decreasing order), for all the sets of perturbations considered in Equations (13)-(20). The results corresponding to the raw waveform, spectrogram and MFCC representations are shown in the first, second and third row of the figure, respectively. Whereas the left column shows the singular values obtained with the SVD for each data representation, in the right column the decays are characterized by fitting exponential curves (depicted as dashed lines) with the following form:[6]

$$y = \rho \cdot e^{-x\lambda} + \omega \quad , \quad \rho, \lambda, \omega \in \mathbb{R}. \tag{21}$$

A higher value of the decay factor λ represents a faster decay, as is illustrated in Figure 7, which shows the behavior of the exponential curves for different values of the decay factor λ. As can be seen in the figure, for low values of λ (e.g., $\lambda \leq 1$) the obtained curves are close to a *constant* or *linear* decay (i.e., $y = 1 - x$), whereas for $\lambda > 1$ the values decay much faster (i.e., exponentially).

Regarding the results in the raw waveform representation (i.e., $\mathcal{V}$ and $\mathcal{R}$), the decay of the singular values is mainly linear for both individual and random perturbations, which can be assessed by their decay factor λ (see Figure 6), since in both cases $\lambda < 1$ is obtained. This means that, in both cases, there is not a set of singular vectors that is considerably more *informative* than the rest, and, as a consequence, a large set of vectors would be needed to provide an approximate basis for the perturbations. Thus, the perturbations do not show meaningful correlations in this representation. The same conclusion can be drawn from the perturbations sampled uniformly at random in the space of

[6] Note that the singular values have been scaled in the range $[0, 1]$ before fitting the exponential curves, for a more uniform comparison.

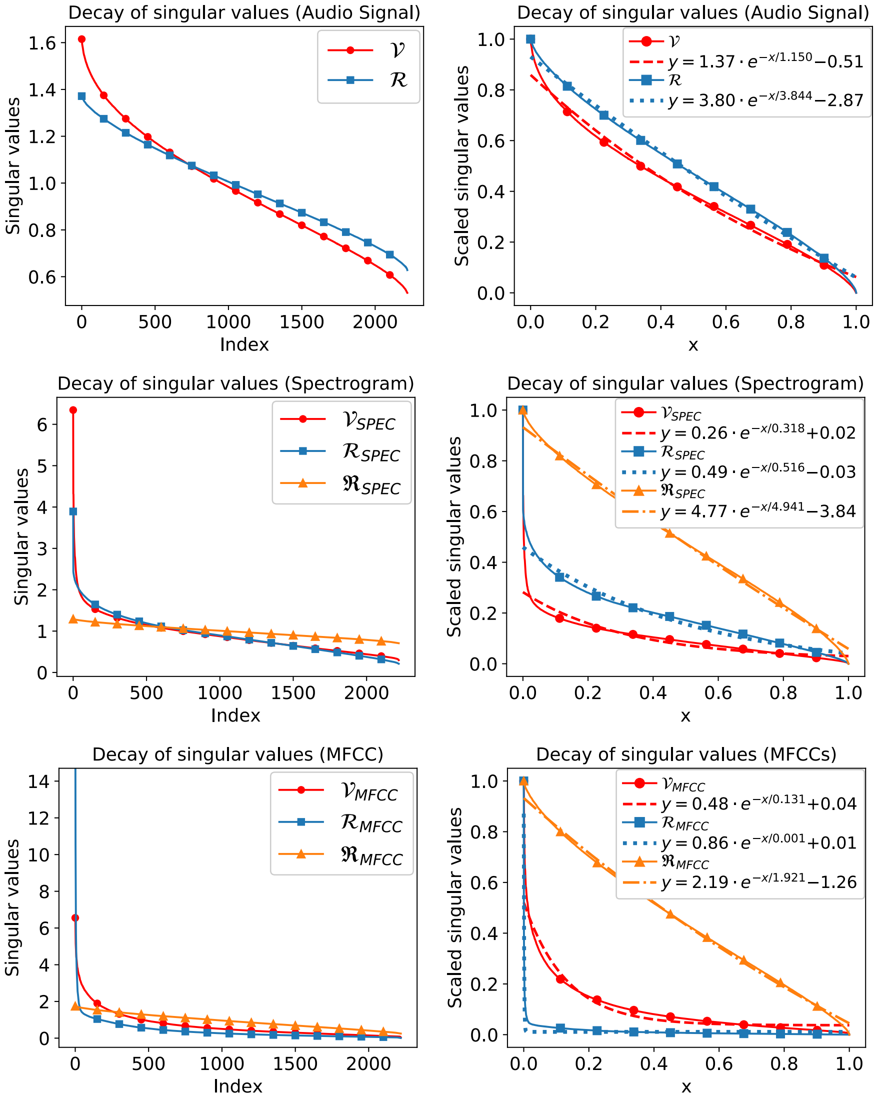


Figure 6: Left column: singular values obtained in the SVD of individual adversarial perturbations and random perturbations, computed in three feature representations: raw audio waveforms (top), spectrograms (center) and MFCCs (bottom). Right column: characterization of the decay of the singular values by fitting an exponential curve (the values in both axes have been scaled in the range [0,1]).

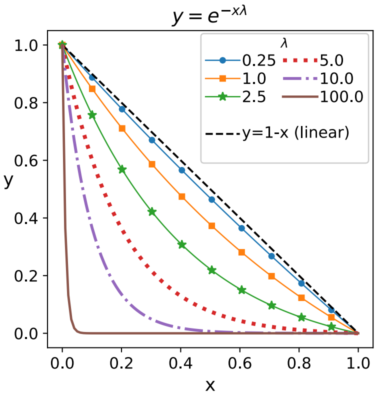


Figure 7: Illustration of an exponential decay $y = \rho e^{-x\lambda} + \omega$ for different values of the *decay factor* λ. For a more uniform comparison, the values $\rho = 1$ and $\omega = 0$ were used in all the cases, and the curves were normalized in the range $[0, 1]$.

spectrograms ($\mathfrak{R}_{\text{SPEC}}$) and in the space of MFCC coefficients ($\mathfrak{R}_{\text{MFCC}}$). However, considering the perturbations in the frequency domain produced by the raw waveform perturbations, either random or adversarial (i.e., $\mathcal{V}_{\text{SPEC}}$, $\mathcal{R}_{\text{SPEC}}$, $\mathcal{V}_{\text{MFCC}}$ and $\mathcal{R}_{\text{MFCC}}$), the singular values decay exponentially, achieving decay factors which are at least of one order of magnitude greater than for the previous cases. For instance, in the MFCC representation (i.e., $\mathcal{V}_{\text{MFCC}}$ and $\mathcal{R}_{\text{MFCC}}$), the values obtained are $\lambda = \frac{1}{0.131}$ and $\lambda = \frac{1}{0.001}$, respectively.

These results indicate, first, that even if the perturbations are generated in the raw audio waveform representation, it is necessary to go to the frequency-domain to observe informative patterns. This might be a fundamental difference between the image domain and the audio domain, as most of the analyses done in the former can be done directly in the *raw* image space. Secondly, the effect of audio perturbations in the frequency-domain can be characterized by just a small (in comparison to the dimensionality of the corresponding spaces) number of singular vectors. For instance, for the MFCC representation, the most relevant information is captured in less than the ∼150 first singular vectors (that is, those corresponding to the highest singular values). The fact that this happens for both random or adversarial perturbations could imply, however, that the captured correlations are uninformative about the geometry of the decision boundaries around natural inputs, or, alternatively, about the vulnerability of the network to adversarial attacks. Nevertheless, in the reminder of this section we show that the SVD of individual adversarial perturbations not only provides a representative basis for input-agnostic perturbations, but also that this basis is strongly connected with the dominant classes. For the previous reasons, the rest of the analysis will focus on the MFCC feature space.

We start evaluating the fooling rate of randomly sampled perturbations in the subspace spanned by the first $N = \{10, 50, 100, 200, 500\}$ singular vectors, for the cases in which the SVD is computed for individual perturbations ($\mathcal{V}_{\text{MFCC}}$) and random perturbations ($\mathcal{R}_{\text{MFCC}}$). Given a value of N, the sampled perturbations will be produced as:

$$v' = \begin{bmatrix} | & | & & | \\ s_1 & s_2 & \dots & s_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad x_1, \dots, x_N \sim \mathcal{U}(0, 1), \tag{22}$$

that is, as a linear combination of the first N singular vectors $s_1, \dots, s_N$ (computed for either $\mathcal{V}_{\text{MFCC}}$ or $\mathcal{R}_{\text{MFCC}}$). All the sampled perturbations were normalized, and the fooling rate was evaluated for different scaling factors under the ℓ_2 norm, in the range $[-200, 200]$. Note that, given a unit vector v, for any scalar $c \in \mathbb{R}$, $||v \cdot c||_2 = |c|$. For reference, the median ℓ_2 norm of the perturbations (in the MFCC) produced by the 10 universal attacks generated in Section 4, measured in the test set, is approximately 100.

Figure 8 shows, for each value of N, the average fooling rates obtained for 100 trials (i.e., 100 random perturbations). The fooling rates have been computed in the test set. The results clearly show that, when the SVD is computed for individual perturbations ($\mathcal{V}_{\text{MFCC}}$), the fooling rates are remarkably higher than for the case of random perturbations ($\mathcal{R}_{\text{MFCC}}$), even for norms close to zero. For instance, taking as reference the results corresponding to an ℓ_2 norm of 100, the average fooling rate is approximately 48% for the case of individual perturbations, when $N \leq 100$. For the case of random perturbations, in the same conditions, the average fooling rate is only 17%.

However, the fooling rate corresponding to individual perturbations considerably decreases when a large number of singular vectors are considered. Indeed, for $N \geq 200$, the fooling rates get closer to those obtained for random perturbations. For instance, when $N = 500$, the average fooling rate (with an ℓ_2 norm of 100) is approximately 18%. This reveals that, whereas the singular vectors corresponding to the highest singular values are capturing directions normal to the decision boundaries around natural inputs (being, therefore, effective in fooling the model for a large number of inputs), the remaining singular vectors do not provide additional or relevant information.

5.3.2 Connection with dominant classes

In the previous section, we have shown that, also for speech command classification models, it is possible to find a low dimensional subspace S containing (*input-agnostic*) vectors normal to the decision boundaries in the vicinity of natural inputs. Therefore, a reasonable hypothesis is that dominant classes can be explained in terms of the geometric similarity of the decision boundaries in regions surrounding natural inputs, information that is captured by the basis of S, that is, by the singular vectors obtained from the SVD of individual perturbations.

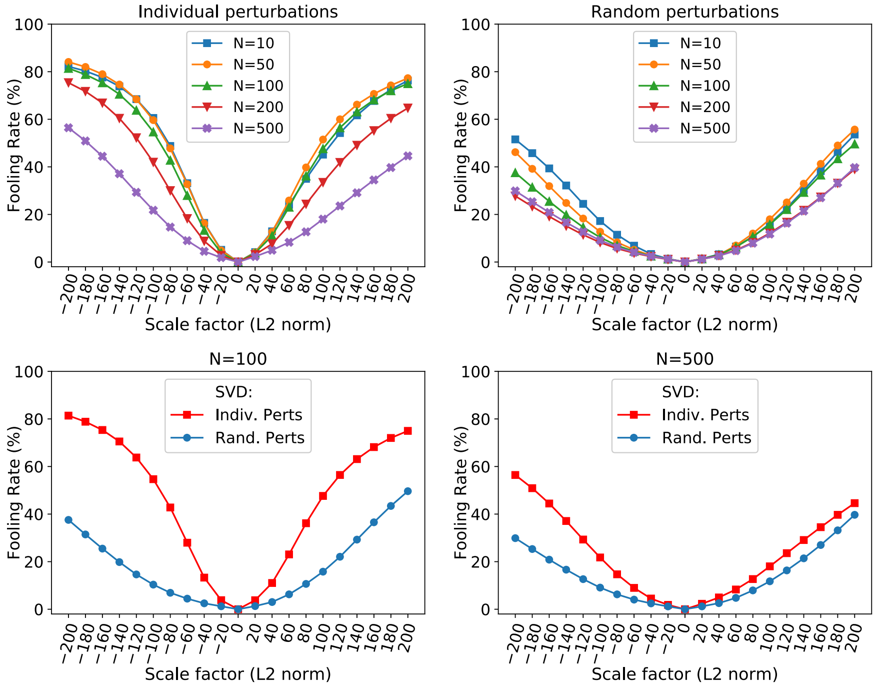


Figure 8: Fooling rate produced by random perturbations sampled from the subspace spanned by the first N singular vectors. The results are averaged for 100 random perturbations. Each perturbation v was normalized and multiplied by different scale factors s_f (horizontal axis), so that $||v||_2 = |s_f|$. The SVD is computed for individual perturbations (top left) and for random perturbations (top right), in the MFCC feature space. The bottom row shows a direct comparison between the average effectiveness of individual and random perturbations for $N = 100$ (bottom left) and $N = 500$ (bottom right).

The first hypothesis is that the first singular vectors are also normal to decision boundaries corresponding to the dominant classes. To validate this hypothesis, we first computed the fooling rate that each singular vector can achieve individually. This is shown in Figure 9 (top left), in which the fooling rate of the first 250 singular vectors is reported for different ℓ_2 norms. For reference, the results corresponding to a norm of 100 are also shown independently in the bottom-left part of the figure. The results clearly show that the first singular vectors are capable of fooling the model for a considerable number of test inputs, particularly for the first 50 vectors (approximately), for which an average fooling rate of 56.3% is achieved. These fooling rates are also remarkably higher than the ones obtained when the SVD is computed for random perturbations, which are also shown in Figure 9 (right column). Indeed, the average fooling rate obtained in the latter case (considering the first 50 vectors) is 18.7%, which represents a difference of 37.6%.

To continue with the analysis, we computed the frequency with which each class is (wrongly) predicted, considering only the inputs that were misclassified when the singular vectors were used as perturbations. The aim of this analysis is to assess if there exists a direct connection with the dominant classes. The results are shown in Figure 10, considering the first 100 singular vectors, scaled to have an Euclidean norm of 100. As can be seen, considering the singular vectors with the highest fooling rate (those corresponding to the vectors approximately in the range [1,50]), the most frequent wrong classes are *unknown* and *left*. Indeed, for 84% of the singular vectors in [1,50], the sum of the frequency corresponding to those two classes exceeds 50%, that is, at least 50% of the misclassified inputs are classified as *left* or as *unknown*. Moreover, for 62% of the singular vectors, the total frequency corresponding to those two classes exceeds 80%. Therefore, we now know that the singular vectors (with a high fooling rate) not only point towards decision boundaries in the close vicinity of natural inputs, but also that those decision boundaries correspond mainly to the dominant classes.

We repeated the experiment using the singular vectors obtained when the SVD is computed for random perturbations. The results are shown in Figure 11. In this case, it is evident that the results are more uniform along all the singular vectors, particularly for those singular vectors with a higher fooling rate (precisely, those in the range [1,50], as shown in Figure 9). For reference, in this case, only for 32% of the singular vectors in the range [1,50] the total frequency corresponding to *unknown* or *left* exceeds 50%, and only for 2% of the singular vectors the total frequency exceeds 70%.

Overall, the SVD of individual perturbations has shown that the obtained singular vectors are input-agnostic perturbations directions for which the model is highly vulnerable: even when the inputs are slightly pushed in those directions, they surpass the decision boundary of the model. This reveals that the geometry of the decision boundary has *patterns* that are repeated in the vicinity of multiple natural inputs. Apart from that, we have shown that such *adversarial* directions mainly point towards the decision boundaries corresponding to the dominant classes. Therefore, it can be concluded that the universal pertur-

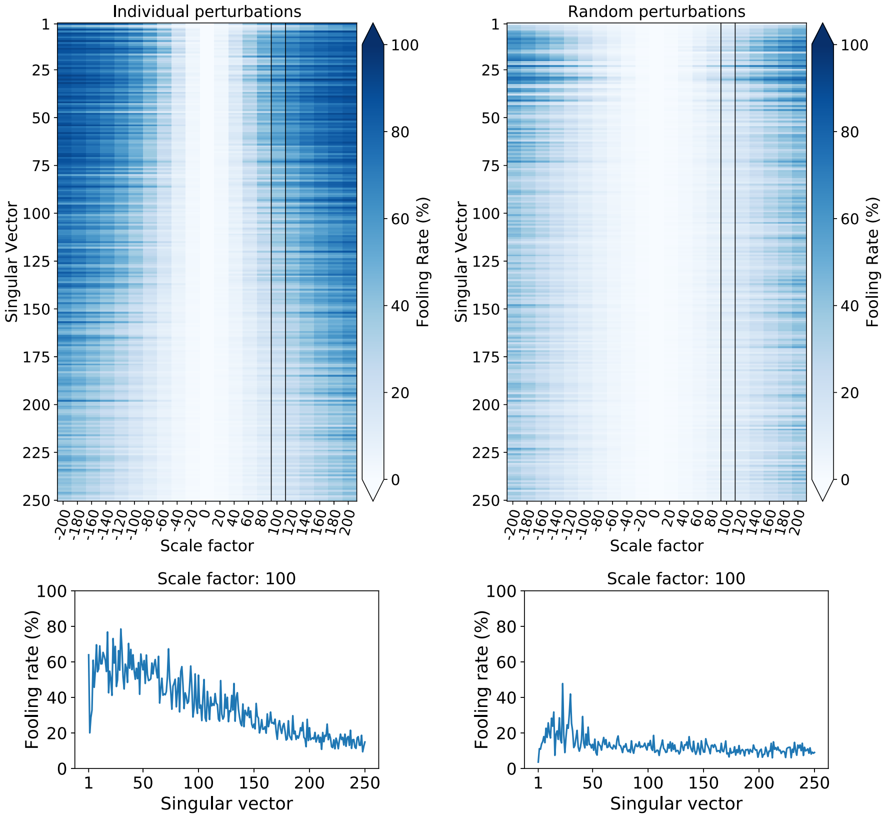


Figure 9: Fooling rate percentage achieved when the inputs are perturbed with the first singular vectors computed for individual perturbations (left column) and for random perturbations (right column), in the MFCC feature space.

bation optimization algorithms implicitly exploit the *shared* geometric patterns of decision boundaries to increase the effectiveness of the perturbations, leading to the same dominant classes in the majority of the cases.

6 Conclusion

In this paper, we have proposed and experimentally validated a number of hypotheses to justify the intriguing phenomenon of why universal adversarial perturbations for DNNs are capable of sending the majority of inputs towards the same wrong class (i.e., dominant classes), even if such behaviour is not specified during the optimization of the perturbations. These hypotheses were studied in the audio domain, using a speech command classification task as a testbed. To the best of our knowledge, previous work has examined this effect only in the image domain, proposing open explanations that we revisit. The results obtained from our analysis revealed multiple interesting facts regarding the vulnerability of DNNs to adversarial perturbations. On the one hand, we have shown that universal perturbations can be created just by optimizing a perturbation to be recognized by the model as one particular class with high confidence. This establishes a new perspective to create universal perturbations, while explains that a class is dominant if it contains patterns in the data distribution for which the model has a higher sensitivity. On the other hand, we demonstrated that the geometry of the decision boundaries of audio DNNs contains similar patterns in the vicinity of natural inputs, and that the most *vulnerable* directions in the decision space point to the regions corresponding to the dominant classes. Finally, our work highlights a number of differences between the image domain and the audio domain, which contribute to a better and more general understanding of the field of adversarial machine learning.

7 Future research lines

Whereas the frameworks proposed in this paper have shown to be effective in revealing the connections between dominant classes and universal perturbations, there are a number of open lines that could be further investigated in order to achieve a deeper understanding of the behavior of universal perturbations.

First, focusing on the framework proposed in Section 5.2 an interesting future line of research could be trying to identify the data-features that the model recognizes as each class with high confidence, for instance, following the methodologies proposed in recent related works [29]. Similarly, the analysis of the geometry of the decision space carried out in Section 5.3 could be further extended by considering the curvature of the decision boundaries, which has proven to be highly informative for the analysis of universal perturbations [8, 26]. Moreover, it could be interesting trying to unify the data-feature perspective used in Section 5.2 and the one used in Section 5.3, relying on the geometry of the decision space of the DNN. Finally, a deeper understanding of the decision

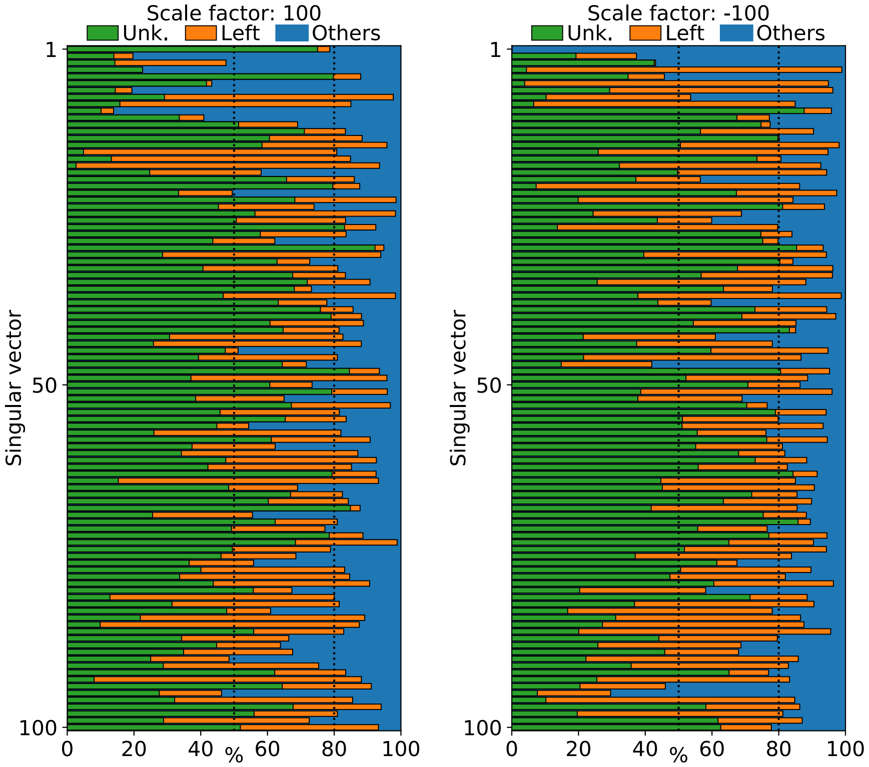


Figure 10: Frequency with which each class is assigned to the misclassified inputs under the effect of singular vectors (computed for **individual perturbations**, see Equation (15)). The (unit) singular vectors have been scaled using two different scale factors: 100 (left) and −100 (right). For the sake of clarity, the frequencies are shown individually for the classes *unknown* and *left*, while the total frequency corresponding to the rest of classes has been grouped (*others*).

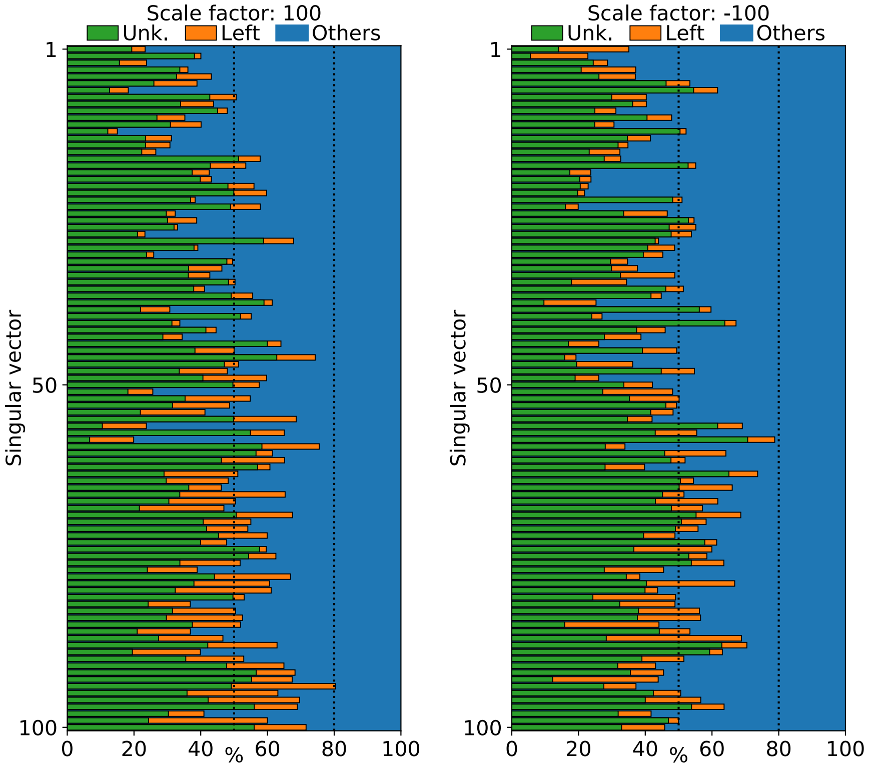


Figure 11: Frequency with which each class is assigned to the misclassified inputs under the effect of singular vectors (computed for **random perturbations**, see Equation (18)). The (unit) singular vectors have been scaled using two different scale factors: 100 (left) and −100 (right). For the sake of clarity, the frequencies are shown individually for the classes *unknown* and *left*, while the total frequency corresponding to the rest of classes has been grouped (*others*).

spaces of DNNs is necessary to comprehensively explain why decision boundaries contain large geometric correlations around natural inputs, as well as many other fundamental questions regarding the learning process of DNNs.

Advances in all these research lines could bring a deeper understanding of the vulnerability of DNNs to adversarial attacks, which can be used, for instance, to create more effective attacks. Indeed, as shown in Section 4, the existence of dominant classes reduces the effectiveness of universal perturbations, since the fooling rate in the inputs of those classes is practically zero. Therefore, preventing the appearance of dominant classes during the generation of the perturbation can lead to more effective attacks. At the same time, understanding the vulnerabilities of DNNs to adversarial attacks also contributes to the generation of more effective defensive strategies, and, ultimately, more robust models.

8 Acknowledgments

This work is supported by the Basque Government (BERC 2018-2021 and ELKARTEK programs, IT1244-19, and PRE_2019_1_0128 predoctoral grant), by the Spanish Ministry of Economy and Competitiveness MINECO (projects TIN2016-78365-R and PID2019-104966GB-I00) and by the Spanish Ministry of Science, Innovation and Universities (FPU19/03231 predoctoral grant). Jose A. Lozano acknowledges support by the Spanish Ministry of Science, Innovation and Universities through BCAM Severo Ochoa accreditation (SEV-2017-0718).

References

[1] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal Adversarial Perturbations, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 86–94. doi:10.1109/CVPR.2017.17

[2] K. R. Mopuri, U. Garg, R. V. Babu, Fast Feature Fool: A Data Independent Approach to Universal Adversarial Perturbations, in: Proceedings of the British Machine Vision Conference 2017 (BMVC), 2017, pp. 1–12. doi:10.5244/C.31.30

[3] V. Khrulkov, I. Oseledets, Art of Singular Vectors and Universal Adversarial Perturbations, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8562–8570. doi:10.1109/CVPR.2018.00893.

[4] J. Vadillo, R. Santana, Universal Adversarial Examples in Speech Command Classification, in: Proceedings of the XIX Conference of the Spanish Association for Artificial Intelligence (CAEPIA), 2021, pp. 642–647.

[5] K. T. Co, L. Muñoz-González, L. Kanthan, B. Glocker, E. C. Lupu, Universal Adversarial Perturbations to Understand Robustness of Texture vs. Shape-biased Training, arXiv preprint arXiv:1911.10364.

[6] H. Hirano, A. Minagi, K. Takemoto, Universal Adversarial Attacks on Deep Neural Networks for Medical Image Classification, available at Research Square (2020). doi:10.21203/rs.3.rs-70727/v2.

[7] M. Behjati, S.-M. Moosavi-Dezfooli, M. S. Baghshah, P. Frossard, Universal Adversarial Attacks on Text Classifiers, in: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 7345–7349. doi:10.1109/ICASSP.2019.8682430.

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, S. Soatto, Analysis of Universal Adversarial Perturbations, arXiv preprint arXiv:1705.09554.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing Properties of Neural Networks, in: International Conference on Learning Representations (ICLR), 2014, pp. 1–10.

[10] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, in: International Conference on Learning Representations (ICLR), 2015, pp. 1–11.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574–2582. doi:10.1109/CVPR.2016.282.

[12] A. Kurakin, I. J. Goodfellow, S. Bengio, Adversarial Machine Learning at Scale, in: International Conference on Learning Representations (ICLR), 2017, pp. 1–17.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, in: International Conference on Learning Representations (ICLR), 2018, pp. 1–23.

[14] K. R. Mopuri, U. Ojha, U. Garg, R. V. Babu, NAG: Network for Adversary Generation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 742–751. doi:10.1109/CVPR.2018.00084.

[15] O. Poursaeed, I. Katsman, B. Gao, S. Belongie, Generative Adversarial Perturbations, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4422–4431.

[16] J. Hayes, G. Danezis, Learning Universal Adversarial Perturbations with Generative Models, in: 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 43–49. doi:10.1109/SPW.2018.00015.

[17] K. R. Mopuri, P. K. Uppala, R. V. Babu, Ask, Acquire, and Attack: Data-Free UAP Generation Using Class Impressions, in: Computer Vision - European Conference on Computer Vision (ECCV), 2018, pp. 20–35. doi:10.1007/978-3-030-01240-3_2.

[18] K. R. Mopuri, A. Ganeshan, R. V. Babu, Generalizable Data-Free Objective for Crafting Universal Adversarial Perturbations, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (10) (2019) 2452–2465. doi:10.1109/TPAMI.2018.2861800.

[19] C. Zhang, P. Benz, T. Imtiaz, I. S. Kweon, Understanding Adversarial Examples From the Mutual Influence of Images and Perturbations, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 14509–14518. doi:10.1109/CVPR42600.2020.01453.

[20] T. Gupta, A. Sinha, N. Kumari, M. Singh, B. Krishnamurthy, A Method for Computing Class-wise Universal Adversarial Perturbations, arXiv preprint arXiv:1912.00466.

[21] C. Zhang, P. Benz, T. Imtiaz, I.-S. Kweon, CD-UAP: Class Discriminative Universal Adversarial Perturbation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 6754–6761. doi:10.1609/aaai.v34i04.6154.

[22] J. H. Metzen, M. C. Kumar, T. Brox, V. Fischer, Universal Adversarial Perturbations Against Semantic Image Segmentation, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2774–2783. doi:10.1109/ICCV.2017.300.

[23] J. Li, X. Zhang, C. Jia, J. Xu, L. Zhang, Y. Wang, S. Ma, W. Gao, Universal Adversarial Perturbations Generative Network For Speaker Recognition, in: 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1–6. doi:10.1109/ICME46284.2020.9102886.

[24] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, F. Koushanfar, Universal Adversarial Perturbations for Speech Recognition Systems, in: Interspeech 2019, 2019, pp. 481–485. doi:10.21437/Interspeech.2019-1353.

[25] E. Wallace, S. Feng, N. Kandpal, M. Gardner, S. Singh, Universal Adversarial Triggers for Attacking and Analyzing NLP, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2153–2162. doi:10.18653/v1/D19-1221.

[26] S. Jetley, N. Lord, P. Torr, With Friends Like These, Who Needs Adversaries?, in: Advances in Neural Information Processing Systems 31, 2018, pp. 10749–10759.

[27] T. Tanay, L. Griffin, A Boundary Tilting Perspective on the Phenomenon of Adversarial Examples, arXiv preprint arXiv:1608.07690.

[28] D. Stutz, M. Hein, B. Schiele, Disentangling Adversarial Robustness and Generalization, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6969–6980. doi:10.1109/CVPR.2019.00714.

[29] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial Examples Are Not Bugs, They Are Features, in: Advances in Neural Information Processing Systems 32, 2019, pp. 125–136.

[30] P. Warden, Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition, arXiv preprint arXiv:1804.03209.

[31] T. N. Sainath, C. Parada, Convolutional Neural Networks for Small-Footprint Keyword Spotting, in: Sixteenth Annual Conference of the International Speech Communication Association (Interspeech 2015), 2015, pp. 1478–1482.

[32] Z. Li, Y. Wu, J. Liu, Y. Chen, B. Yuan, AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20, 2020, pp. 1121–1134. doi:10.1145/3372297.3423348.

[33] M. Alzantot, B. Balaji, M. Srivastava, Did you hear that? Adversarial Examples Against Automatic Speech Recognition, arXiv preprint arXiv:1801.00554.

[34] F. Yu, Z. Xu, Y. Wang, C. Liu, X. Chen, Towards Robust Training of Neural Networks by Regularizing Adversarial Gradients, arXiv preprint arXiv:1805.09370.

[35] L. Muda, M. Begam, I. Elamvazuthi, Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques, Journal of Computing 2 (3) (2010) 138–143.

[36] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83. doi:https://doi.org/10.2307/3001968.

Appendix A Clean accuracy of the model in the test set

Class	Accuracy	Samples
Silence	99.51	408
Unknown	66.42	408
Yes	94.03	419
No	74.57	405
Up	92.00	425
Down	80.79	406
Left	89.81	412
Right	88.64	396
On	87.12	396
Off	81.59	402
Stop	93.67	411
Go	77.36	402
Average	85.52	-

Table A.1: Initial accuracy percentage of the DNN on the test set.

Appendix B Detailed analysis of the effectiveness of universal perturbations (UAP-HC)

Table B.1 shows the effectiveness of each universal adversarial perturbation generated in Section 4, using Algorithm 1.

Experiment	Restricted class		
	None	{*Left*}	{*Left*, *Unk.*}
1	46.34	37.73	33.88
2	35.29	31.56	34.24
3	41.25	36.35	37.49
4	38.47	37.42	34.91
5	38.35	32.86	34.31
6	30.13	30.30	29.84
7	32.52	34.55	32.88
8	33.98	34.29	30.94
9	41.08	37.14	33.86
10	41.94	36.80	35.15
Mean	37.94	34.90	33.75
Mean[1]	41.68	37.39	37.08
Mean[2]	44.97	40.32	39.90

[1] Without considering dominant classes.

[2] Without considering dominant classes and *Silence*.

Table B.1: Fooling rate percentage of the universal adversarial perturbations generated using Algorithm 1. The results are computed for a set of *test* samples, which were not seen during the generation of the universal perturbations.