Contrasting Adversarial Perturbations: The Space of Harmless Perturbations

Lu Chen¹ Shaofeng Li² Benhao Huang¹ Fan Yang¹ Zheng Li¹ Jie Li¹ Yuan Luo¹

¹Shanghai Jiao Tong University

²Peng Cheng Laboratory

{lu.chen,hbh001098hbh,fan-yang,li-zheng,lijiecs}@sjtu.edu.cn

lishf@pcl.ac.cn,luoyuan@cs.sjtu.edu.cn

Abstract

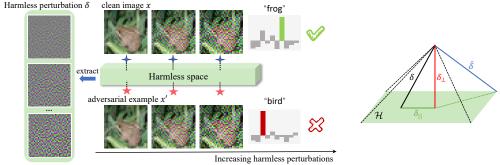
Existing works have extensively studied adversarial examples, which are minimal perturbations that can mislead the output of deep neural networks (DNNs) while remaining imperceptible to humans. However, in this work, we reveal the existence of a harmless perturbation space, in which perturbations drawn from this space, regardless of their magnitudes, leave the network output unchanged when applied to inputs. Essentially, the harmless perturbation space emerges from the usage of non-injective functions (linear or non-linear layers) within DNNs, enabling multiple distinct inputs to be mapped to the same output. For linear layers with input dimensions exceeding output dimensions, any linear combination of the orthogonal bases of the nullspace of the parameter consistently yields no change in their output. For non-linear layers, the harmless perturbation space may expand, depending on the properties of the layers and input samples. Inspired by this property of DNNs, we solve for a family of general perturbation spaces that are redundant for the DNN's decision, and can be used to hide sensitive data and serve as a means of model identification. Our work highlights the distinctive robustness of DNNs (i.e., consistency under large magnitude perturbations) in contrast to adversarial examples (vulnerability for small imperceptible noises).

1 Introduction

The robustness of Deep Neural Networks (DNNs) against structured and unstructured perturbations has attracted significant attention in recent years [Szegedy et al., 2014, Nguyen et al., 2015, Fawzi et al., 2016, Salman et al., 2021]. In particular, deep learning models are shown highly vulnerable to adversarial perturbations [Szegedy et al., 2014]. These well-crafted perturbations, which are imperceptibly small to the human eye, cause DNNs to misclassify with high confidence [Carlini and Wagner, 2017, Madry et al., 2018, Croce and Hein, 2020]. Naturally, an inquiry arises:

Are there perturbations within the input space capable of preserving network output invariance?

Unlike vulnerability against adversarial examples, in this paper, we reveal the robustness of DNNs to specific perturbations that render the network output mathematically strictly invariant. We demonstrate the existence of such *harmless* perturbations that, when introduced onto natural images or embeddings, regardless of their magnitude, will not affect the discrimination of the DNN. Such harmless perturbations arising from the linear layers are universal, as they are instance-independent and solely determined by the parameter space of the DNN. These harmless perturbations span a continuous harmless subspace, embedded within the high-dimensional feature space. The surprising existence of harmless perturbations and their subspaces reveals a distinctive view of DNN robustness.



(a) Harmless perturbations

(b) Decomposition of general perturbations

Figure 1: (a) Harmless perturbations added to images completely do not change the network output of the images, regardless of the magnitude of these harmless perturbations. (b) Illustration of the equivalent effect of any perturbation on the network output. Given any linear layer with a harmless subspace \mathcal{H} , the network outputs of any perturbations δ and $\hat{\delta}$ are equivalent to those of their components δ_{\perp} orthogonal to the harmless subspace.

For the linear layers of DNNs, we find that when its input dimension n exceeds the output dimension m, the harmless perturbation subspace of this layer can be derived by computing the nullspace of its parameter matrix A, i.e., $N(A) = \{v \in \mathbb{R}^n | Av = \mathbf{0}\}$. To this end, the harmless subspace exhibits a dimension of (n-m) and is embedded within an n-dimensional feature space. Furthermore, the harmless perturbation space may expand when involving non-linear layers, depending on the specific non-linear functions and input samples (See Section 3.2). Inspired by the harmless subspace of linear layers, we further investigate the robustness of DNNs against more general perturbations, i.e., random noises or adversarial perturbations. We find that a family of those general perturbations, irrespective of their magnitude, identically influence the DNN's output. This phenomenon stems from the decomposition of arbitrary perturbations into the sum of any harmless and harmful components. Consequently, the network output for general perturbations becomes equivalent to that of harmful perturbations, particularly aligning with that of components orthogonal to the harmless perturbation subspace (Fig. 1(b)). Essentially, for any linear layer with a harmless subspace, the equivalent feature space is characterized by identical orthogonal components, leading to consistent network outputs.

The existence of harmless perturbations and their space promotes several potential benefits. First, capitalizing on the disparity between DNNs and human perception, *i.e.*, significant perturbations perceivable by the human eye may not affect the recognition of DNNs, we delve into the application of harmless perturbations to privacy-preserving data and model fingerprints. Additionally, as demonstrated in Fig. 1(a), there exist equivalent adversarial spaces, ensuring equal attacking capabilities for adversarial perturbations regardless of their magnitude. In other words, the perturbation magnitude is not a decisive factor in attacking the network. Instead, focusing on the attack utility of the "effective component" of the perturbation facilitates a deeper understanding of the robustness of DNNs. In summary, this paper makes the following contributions:

- We demonstrate for the first time the concept of "harmless perturbations" and show the existence of a harmless perturbation space for DNNs. For any linear layer with the input dimension n exceeding the output dimension m, there exists a continuous harmless perturbation subspace of dimension (n-m). The harmless perturbation space may expand when considering non-linear layers, depending on the properties of the layers and input samples.
- We present a novel perspective to decompose any general perturbation (*i.e.*, random noises or adversarial perturbations) into its harmful and harmless counterparts. Given any linear layer with a harmless perturbation subspace, the network output solely depends on its orthogonal (harmful) component, irrespective of its magnitude (innocuous) part.
- We reveal the difference between DNNs and human perception, *i.e.*, significant perturbations captured by humans may not affect the recognition of DNNs, which highlights a distinctive aspect of DNN robustness. Based on this insight, we employ the proposed harmless perturbations with a large magnitude to hide the sensitive image data for DNN usage. As harmless perturbations are usually not transferable across different DNNs, they can also serve as model fingerprints.

2 Related work

Adversarial examples and adversarial robustness. Existing literature extensively explored the impact of adversarial perturbations [Szegedy et al., 2014] on the robustness of DNNs, including their ability to deceive both the digital and physical scenarios [Kurakin et al., 2017], fool both the white-box models [Goodfellow et al., 2015, Madry et al., 2018] and black-box models [Papernot et al., 2017, Chen et al., 2017], and manifest as either image-specific or image-agnostic universal perturbations [Moosavi-Dezfooli et al., 2017]. Many defenses against these adversarial perturbations have been proposed but they were susceptible to being broken by more powerful or adapted attacks [Carlini and Wagner, 2017, Athalye et al., 2018]. Amongst them, adversarial training [Madry et al., 2018] and its variant [Zhang et al., 2019] still indicated their relatively reliable robustness against more powerful attack [Croce and Hein, 2020].

Adversarial space. Previous studies have delved into the vulnerability of DNNs from the perspective of high-dimensional input spaces. Goodfellow et al. [2015] argued that the "highly linear" of DNNs explained their instability to adversarial perturbations. Fawzi et al. [2016] quantified the robustness of classifiers from the dimensionality of subspaces within the semi-random noise regime. Gilmer et al. [2018] suggested that adversarial perturbations arised from the high-dimensional geometry of data manifolds. Tramèr et al. [2017] stated that adversarial transferability arised from the intersection of high-dimensional adversarial subspaces from different models. [Shafahi et al., 2019] empirically discussed that how dimensionality affected the robustness of classifiers to adversarial perturbations.

Unrecognizable features. A series of prior works [Geirhos et al., 2019, Ilyas et al., 2019, Tsipras et al., 2019, Jacobsen et al., 2019, Yin et al., 2019, Wang et al., 2020] have demonstrated that humans and DNNs tend to utilize different features to make decisions. Besides, producing totally unrecognizable images [Nguyen et al., 2015] or introducing visually perceptible patches to images [Salman et al., 2021, Wang et al., 2022, Si et al., 2023] may not alter the classification categories of DNNs.

3 The space of harmless perturbations

We develop a framework to rigorously define "harmless" and "harmful" perturbations w.r.t. the network output. In particular, we formally define and solve for the subspace for harmless perturbations in any linear layer of a given DNN. Subsequently, the definitions and solutions are extended to nonlinear layers by analyzing the properties of the functions.

3.1 Harmless perturbations for a linear layer

Consider a function mapping $\mathcal{L}: \mathbb{R}^n \to \mathbb{R}^m$ on an input sample $x \in \mathbb{R}^n$, the goal is to find a set of input perturbations $\delta \in \mathbb{R}^n$ that rigorously do not change the output of the function. To this end, we first give the definition of harmless perturbations as follows.

Definition 1 (Harmless perturbations). The set of harmless perturbations for a function \mathcal{L} is defined as $\mathcal{S} := \{\delta | \mathcal{L}(x+\delta) = \mathcal{L}(x)\}$ subject to $\|\delta\|_p < \xi, \xi > 0$.

Definition 1 denotes a set of input perturbations that *thoroughly* do not affect the function output. According to Definition 1, the set of harmless perturbations for a linear function $\mathcal{L}(x) = Ax$, where $A \in \mathbb{R}^{m \times n}$ is the parameter matrix, can be formulated as $\mathcal{S} = \{\delta | A(x+\delta) = Ax\} = \{\delta | A\delta = \mathbf{0}\}$. It indicates that the set of harmless perturbations for a single linear layer \mathcal{L} is equivalent to the *nullspace* of the parameter matrix A, *i.e.*, $\mathcal{S} = N(A) = \{v \in \mathbb{R}^n | Av = \mathbf{0}\}$.

Theorem 1 (Dimension of harmless perturbation subspace). Given a linear layer $\mathcal{L}(x) = Ax \in \mathbb{R}^m$ and an input sample $x \in \mathbb{R}^n$, where the parameter matrix $A \in \mathbb{R}^{m \times n}$. The dimension of the subspace for harmless perturbations is $dim(\mathcal{S}) = n - rank(A)$.

Theorem 1 demonstrate that the subspace for harmless perturbations is the span of $dim(\mathcal{S})$ linearly independent vectors $U \subset \mathcal{S}$, i.e., $\mathcal{S} = span(U) = \{\sum_{i=1}^{dim(\mathcal{S})} c_i u_i | c_i \in \mathbb{R}, u_i \in U\}$ (Proof is in Appendix A). As a special case, the parameter matrix A of a linear layer in DNNs learned through an optimization algorithm (e.g., SGD) starting from an arbitrary initialization, usually possesses linearly independent row vectors. So the dimension of the harmless perturbation subspace for a linear layer $\mathcal{L}(x) = Ax \in \mathbb{R}^m$ is $dim(\mathcal{S}) = n - m$.

Remark 1 (Proof in Appendix A). Consider the case that the input dimension of the linear layer is less than or equal to the output dimension, i.e., $n \le m$. In this case, if the column vectors of the parameter matrix A are linearly independent, then the dimension of the subspace for harmless perturbations is dim(S) = 0.

Remark 1 state that there exists no (non-zero) harmless perturbation that does not affect the output of the linear layer when $n \le m$ and rank(A) = n.

3.2 The space of harmless perturbations for DNNs

Extending harmless perturbations from a single linear layer to the entire DNN is challenging. Consider a DNN $f: \mathbb{R}^{n_{\mathrm{in}}} \mapsto \mathbb{R}^{n_{\mathrm{out}}}$ on an input sample $x \in \mathbb{R}^{n_{\mathrm{in}}}$, the goal now is to identify a set of harmless input perturbations $\delta \in \mathbb{R}^{n_{\mathrm{in}}}$ which ultimately do not alter the network output. Notice that harmless perturbations solved for the intermediate layers do not influence subsequent layers. Therefore, we can formally define the set of harmless perturbations layer by layer for a given DNN.

Definition 2 (Set of harmless perturbations for DNNs). The set of harmless perturbations on the (l+1)-th layer of a DNN f is defined as $\mathcal{H}^{(l)} \coloneqq \{\delta^{(l)}|f^{(l+1)}(z^{(l)}+\delta^{(l)})=f^{(l+1)}(z^{(l)})\}.$

 $z^{(l)} \in \mathbb{R}^{n^{(l)}}$ in Definition 2 represents the l-th intermediate-layer features of the input sample x, and $\delta^{(l)}$ denotes the perturbations added to the features $z^{(l)}$. Definition 2 shows that if the set of harmless perturbations on the features can be found, these perturbations leave the network output unaffected. Furthermore, if we identify a set of perturbations on the input $\mathcal{P}^{(l)} \coloneqq \{\delta|z^{(l)} + \delta^{(l)} = (f^{(l)} \circ \cdots \circ f^{(1)})(x+\delta), \forall \delta^{(l)} \in \mathcal{H}^{(l)}\}$ such that $\delta^{(l)} \in \mathcal{H}^{(l)}$, then $\mathcal{P}^{(l)}$ do not alter the network output.

Lemma 1 (Proof in Appendix C). The set of harmless perturbations on the input for a DNN f with L layers is derived as $\mathcal{P} = \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, $\mathcal{P}^{(0)} := \mathcal{H}^{(0)}$, $\mathcal{P} \subset \mathbb{R}^{n_{in}}$.

Lemma 1 suggests that the set of harmless input perturbations for the entire DNN is the union of the corresponding set of harmless input perturbations $\mathcal{P}^{(l)}$ on each layer. Theoretically, Lemma 1 does not restrict whether any layer in the DNN is linear or nonlinear, *i.e.*, given any layer, if $\mathcal{H}^{(l)}$ and $\mathcal{P}^{(l)}$ can be evaluated, then harmless input perturbations for this layer can still be obtained. Based on Lemma 1, we further investigate the effect of a single layer of nonlinearity on the harmless perturbation space. In scenarios involving non-linear layers, the harmless perturbation space may expand, depending on the specific non-linear functions and input samples.

Lemma 1.1 (Harmless perturbations for injective functions). *If the* (l+1)-th layer $f^{(l+1)}$ is an injective function, the set of harmless perturbations on the (l+1)-th layer of a DNN f is $\mathcal{H}^{(l)} = \{0\}$. Otherwise, $\mathcal{H}^{(l)} \neq \{0\}$. (Proof is in Appendix C)

Lemma 1.2 (Harmless perturbations for ReLU layers). Suppose $f^{(l+1)}$ is the ReLU layer, $\mathcal{H}^{(l)} = \{\delta^{(l)}|\forall i,\delta_i^{(l)} = \begin{cases} 0, & z_i^{(l)} > 0 \\ t(\forall t \leq -z_i^{(l)}), & z_i^{(l)} \leq 0 \end{cases}$, which is determined by intermediate-layer features $z^{(l)}$ and hence the input sample x. (Proof is in Appendix C)

Lemma 1.3 (Harmless perturbations for Softmax layers). Suppose $f^{(l+1)}$ is the Softmax layer, $\mathcal{H}^{(l)} = \{c \cdot 1, c \in \mathbb{R}\}$. (Proof is in Appendix C)

Lemma 1.4 (Harmless perturbations for Average Pooling layers). Suppose $f^{(l+1)}$ is the Average Pooling layer, $\mathcal{H}^{(l)} = N(A_{\text{avg}})$. A_{avg} is a coefficient matrix determined by the constraints that must be satisfied by the perturbations within each averaging region. (Proof is in Appendix C)

Lemma 1.5 (Harmless perturbations for Max Pooling layers). Suppose $f^{(l+1)}$ is the Max Pooling layer, $\mathcal{H}^{(l)} = \{ \forall p, i, \delta_{p,i}^{(l)} \leq c_p - z_{p,i}^{(l)} \} \cap \{ \forall p, \prod_{j=1}^{k \times k} (\delta_{p,j}^{(l)} - c_p + z_{p,j}^{(l)}) = 0 \}$. $c_p \coloneqq \max\{z_{p,1}^{(l)}, z_{p,2}^{(l)}, \cdots, z_{p,k \times k}^{(l)} \}$ is the maximum value of features within the $k \times k$ region of the p-th patch. $\mathcal{H}^{(l)}$ is determined by intermediate-layer features $z^{(l)}$ and hence the input sample x. (Proof is in Appendix C)

Theorem 2 (Harmless perturbations for two-layer neural networks). Given a two-layer neural network $f(x) = \sigma(Ax)$, where σ represents any function. If σ is an injective function, the set of

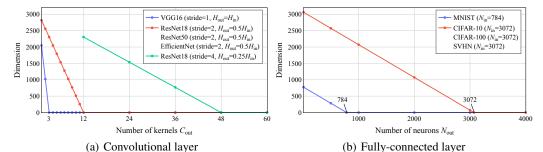


Figure 2: Dimension of harmless perturbation subspace for (a) convolutional layers and (b) fully-connected layers. When the input dimension n of the linear layer is larger than the output dimension m, the dimension of the harmless subspace is (n-m). Otherwise, the dimension is 0.

harmless perturbations on the input \mathcal{P} for f is $\mathcal{P} = \mathcal{P}^{(0)}$. Otherwise, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$. Here, $\mathcal{P}^{(1)} = \{\delta | A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)} \cap C(A)\}^1$ is determined by the specific function σ and the input sample x. (Proof is in Appendix A)

Theorem 2 suggests that the property of the function σ determines whether the set of harmless perturbations for Ax may expand. For instance, if σ is an injective function, such as Sigmoid, Tanh, leaky ReLU [Maas et al., 2013], exponential linear unit (ELU) [Clevert et al., 2016] and scaled exponential linear unit (SeLU) [Klambauer et al., 2017] activation functions, and the linear Batch Normalization (BN) layers at inference time [Ioffe and Szegedy, 2015], the set of harmless perturbations on the input \mathcal{P} remains unchanged, compared to that of Ax. Conversely, if σ is a non-injective function, such as ReLU [Nair and Hinton, 2010], Softmax, Average Pooling [LeCun et al., 1990], and Max Pooling layers [Scherer et al., 2010] (see Lemmas 1.2 to 1.5 and Theorem 2 for their \mathcal{P} , respectively), the set of harmless perturbations on the input \mathcal{P} may expand $\mathcal{P} \supseteq \mathcal{P}^{(0)}$, depending on the specific functions and input samples. Note that, in the above non-linear layers, the harmless perturbation space for the ReLU layer is determined by the input sample x. In an extreme case, if every element of Ax is positive, then its harmless perturbation subspace $\mathcal{P} = \mathcal{P}^{(0)}$. Otherwise, if every element of Ax is not positive, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supset \mathcal{P}^{(0)}$. (For more details, please refer to Lemma 1.2 in Appendix C). In summary, the harmless perturbation space on the input does expand $\mathcal{P} \supseteq \mathcal{P}^{(0)}$ if there exists at least one harmless perturbation $\delta^{(1)} \in \mathcal{H}^{(1)} \cap C(A)(\delta^{(1)} \neq \mathbf{0})$ for the non-injective function σ .

Lemma 1.6 (Harmless perturbations for two-layer linear networks). *Given a two-layer linear network* $f(x) = A_2 A_1 x$, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$. *Here*, $\mathcal{P}^{(0)} = N(A_1)$ and $\mathcal{P}^{(1)} = \{\delta | A_1 \delta = \delta^{(1)}, \forall \delta^{(1)} \in N(A_2) \cap C(A_1)\}$. (*Proof is in Appendix C*)

Furthermore, Lemma 1.6 illustrates the expansion of harmless perturbations on the input \mathcal{P} solely depends on the dimensions of those two linear layers. For two common scenarios in DNNs, where given $A_1 \in \mathbb{R}^{d \times n}$ and $A_2 \in \mathbb{R}^{m \times d}$, when n, m > d, $\mathcal{P} = \mathcal{P}^{(0)}$. Otherwise, when n, m < d, $\mathcal{P} = \mathcal{P}^{(1)}$. (Please see Lemma 1.6 in Appendix C for the details.)

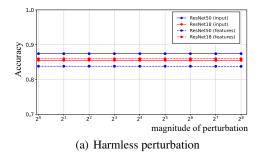
3.3 The subspace of harmless perturbations for linear layers in DNNs

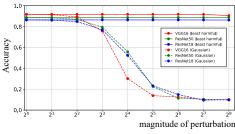
Nevertheless, in this section, we focus on the set of harmless perturbations for two classical linear layers in DNNs, *i.e.*, convolutional layers and fully-connected layers.

Corollary 1 (Harmless perturbation subspace for convolutional layers, proof in Appendix B). Given a convolutional layer $f^{(l+1)}$ with linearly independent vectorized kernels whose kernel size is not smaller than the stride, $z^{(l+1)} = f^{(l+1)}(z^{(l)}) \in \mathbb{R}^{C_{\text{out}} \times H_{\text{out}} \times W_{\text{out}}}$ and $z^{(l)} \in \mathbb{R}^{C_{\text{in}} \times H_{\text{in}} \times W_{\text{in}}}$. If the input dimension is greater than the output dimension, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = C_{\text{in}}H_{\text{in}}W_{\text{in}} - C_{\text{out}}H_{\text{out}}W_{\text{out}}$. Otherwise, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

Corollary 1 demonstrates the subspace for harmless perturbations in a convolutional layer is the span of $dim(\mathcal{H}^{(l)})$ linearly independent vectors $U \subset \mathcal{H}^{(l)}$. Specifically, $\mathcal{H}^{(l)}$ can be obtained by computing the nullspace of a matrix $A \in \mathbb{R}^{(C_{\text{out}}H_{\text{out}}W_{\text{out}}) \times (C_{\text{in}}H_{\text{in}}W_{\text{in}})}$. In practice, A is affected by the padding and the stride of the convolutional layer (see Appendix E for details). Similarly, given a

Note that the equation $A\delta = \delta^{(1)}(\delta \neq \mathbf{0})$ has a solution (meaning at least one solution) if and only if $\delta^{(1)}$ is in the column space of A, i.e., $\delta^{(1)} \in C(A)$.





(b) Least harmful perturbation

Figure 3: The effect of perturbation magnitude on the performance of the network. We trained the CIFAR-10 dataset on various networks and tested the effect of varying magnitudes on (a) harmless perturbations and (b) the least harmful perturbations.

Table 1: Root mean squared errors between the network outputs of the perturbed images and original images on the CIFAR-10 dataset.

	ϵ	2ϵ	4ϵ	8ϵ	16ϵ	32ϵ
Gaussian noise Adversarial perturbation [Madry et al., 2018]	0.1226 6.1994	0.3154 6.3225	0.8112 5.5410	1.7871 5.6122	3.3921 6.6585	5.0009 11.2747
Harmless perturbation Least harmful perturbation	3.63e-15 0.0003	3.70e-15 0.0007	3.77e-15 0.0013	4.20e-15 0.0027	5.38e-15 0.0053	8.55e-15 0.0105

fully-connected layer $z^{(l+1)} = W^\top z^{(l)} \in \mathbb{R}^{N_{\text{out}}}$ and $z^{(l)} \in \mathbb{R}^{N_{\text{in}}}$, the harmless subspace is the span of $dim(\mathcal{H}^{(l)}) = N_{\text{in}} - N_{\text{out}}$ linearly independent vectors $U \subset \mathcal{H}^{(l)}$ (see Corollary 2 in Appendix B). Here, $\mathcal{H}^{(l)}$ is computed as the nullspace of a matrix $A = W^\top$.

Experiments on various DNNs verify Corollaries 1 and 2. In Figure 2, the dimension of the harmless perturbation subspace $dim(\mathcal{H}^{(l)})$ decreased as the output dimension increased. When the output dimension exceeds the input dimension, $dim(\mathcal{H}^{(l)})$ becomes 0. Specifically, we verified the dimension of the harmless perturbation subspace for convolutional layers using various DNNs, including ResNet-18/50 [He et al., 2016], VGG-16 [Simonyan and Zisserman, 2014] and EfficientNet [Tan and Le, 2019], on the CIFAR-10 dataset [Krizhevsky et al., 2009]. Here, we modified the feature size of the output of the first convolutional layer by setting different strides (see Appendix F.1). Furthermore, we verified the dimension of the harmless perturbation subspace for fully-connected layers using the MLP-5 on various datasets, including the MNIST dataset [LeCun and Cortes, 2010], the CIFAR-10/100 dataset [Krizhevsky et al., 2009] and the SHVN dataset [Netzer et al., 2011], to compare the dimension of the subspace under different input dimensions.

Conversely, there exists no (non-zero) perturbation making the network output invariant, if the input dimension of a given linear layer is not greater than the output dimension. However, the least harmful perturbation can be solved for such that the layer output is minimally affected, *i.e.*, given the matrix A with equivalent effect of a linear layer, the least harmful perturbation is $(\delta^{(l)})^* = \operatorname{argmin}_{\delta^{(l)}} \|A\delta^{(l)}\|_2$, s.t., $\|\delta^{(l)}\|_2 = 1$. Hence, the least harmful perturbation $(\delta^{(l)})^*$ is the eigenvector corresponding to the smallest eigenvalue of the matrix A^TA (see Lemma 2 in Appendix C).

We validated the impact of harmless perturbations and the least harmful perturbations on network performance across varying perturbation magnitudes. In Figure 3(a), harmless perturbations, regardless of their magnitude, do not affect the discrimination of DNNs (see Appendix F.2 for details). For the least harmful perturbations in Figure 3(b), they also have negligible effects on the network performance, compared with the Gaussian noise $\mathcal{N}(0,1)$ added to each pixel. Furthermore, we evaluated the root mean squared error (RMSE) between the network outputs of the perturbed images \hat{y}_x and the network outputs of natural images y_x on the ResNet-50, i.e., RMSE= $\mathbb{E}_x[\frac{1}{\sqrt{n}}||\hat{y}_x-y_x||]$. Table 1 further demonstrates that compared to adversarial perturbations and Gaussian noise, harmless perturbations completely did not change the network output with negligible errors, and the least harmful perturbation had a weak impact on the network output as the perturbation magnitude increased.

4 Projection onto the harmless subspace

Inspired by the harmless subspace of linear layers, we can decompose any given perturbation (*i.e.*, random noise, adversarial examples) into its two orthogonal counterparts, namely, harmful and

harmless components. This section extends the harmless subspace to any given perturbations and investigates the projections of these perturbations onto their corresponding harmless subspaces.

Theorem 3 (Arbitrary decomposition of perturbations, proof in Appendix A). Given the (l+1)-th linear layer with harmless subspace $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$ and any perturbation $\forall \delta^{(l)} \notin \mathcal{H}^{(l)}$, it can be arbitrarily decomposed into the sum of a harmless perturbation and a harmful perturbation, i.e., $\delta^{(l)} = \delta_a^{(l)} + \delta_b^{(l)}, \forall \delta_a^{(l)} \in \mathcal{H}^{(l)}$ and $\delta_b^{(l)} \notin \mathcal{H}^{(l)}$. Then, $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta_b^{(l)})$.

Theorem 3 indicates that the network output of any perturbation $\delta^{(l)} \notin \mathcal{H}^{(l)}$ is equivalent to that of its corresponding harmful component $\delta^{(l)}_b \coloneqq (\delta^{(l)} - \delta^{(l)}_a) \notin \mathcal{H}^{(l)}, \forall \delta^{(l)}_a \in \mathcal{H}^{(l)},$ no matter how large the ℓ_p norm of harmful component is. According to Theorem 3, an infinite number of perturbations, regardless of their magnitude, will induce the equivalence of a continuous harmful space². Naturally, an inquiry arises: what is the extent of these perturbations concerning a given DNN? Theorem 4 extends the argument by establishing the existence of a unique perturbation characterized by the smallest ℓ_2 norm (see the proof in Appendix D). This perturbation is orthogonal to the harmless subspace, and exhibits network output consistent with the above infinite number of perturbations embedded in the continuous harmful space (Figure 1(b)).

Theorem 4 (Orthogonal decomposition of perturbations, proof in Appendix A). Given the (l+1)-th linear layer with harmless subspace and any perturbation $\forall \delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a unique decomposition $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$ with the parallel component $\delta^{(l)}_{\parallel} = P\delta^{(l)} \in \mathcal{H}^{(l)}$ and the orthogonal component $\delta^{(l)}_{\perp} = (I-P)\delta^{(l)} \notin \mathcal{H}^{(l)}$. Then, $f^{(l+1)}(\delta^{(l)}_{\parallel}) = \mathbf{0}$ and $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_{\perp})$.

 $P = U(U^{\top}U)^{-1}U^{\top}$ represents the projection matrix onto the harmless subspace $\mathcal{H}^{(l)} \subset \mathbb{R}^{n^{(l)}}$, and $U \in \mathbb{R}^{n^{(l)} \times dim(\mathcal{H}^{(l)})}$ denotes a set of $dim(\mathcal{H}^{(l)})$ orthogonal bases for the subspace $\mathcal{H}^{(l)}$.

As a special case of Theorem 3, Theorem 4 demonstrates that the network output of a family of features/perturbations is equivalent to that of the component of this perturbation family, which is orthogonal to the subspace. In essence, as expounded in Theorem 5, a collection of perturbations can be categorized as a perturbation family with identical impact on the network output, if their orthogonal components exhibit congruence in both magnitude and direction.

Theorem 5 (Identical impact of a family of perturbations, proof in Appendix A). Given the (l+1)-th linear layer with harmless subspace and two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)}$ and $\delta^{(l)}, \hat{\delta}^{(l)} \notin \mathcal{H}^{(l)}$, if their orthogonal components are the same, i.e., $\delta^{(l)}_{\perp} = \hat{\delta}^{(l)}_{\perp}$, then $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$.

Theorem 5 posits that when a set of features/perturbations lies equidistant to the harmless subspace and exhibits the same direction in their orthogonal components, these perturbations form a family that induces uniform network effects. These perturbations can be analogized to form contour lines in a topographic map, as these perturbations with the same orthogonal components yield the same network output (Figure 6). Notably, this effect remains consistent irrespective of the perturbation magnitude. Furthermore, when the orthogonal components of any two perturbations have different directions, i.e., $\delta_{\perp}^{(l)} \neq \alpha \cdot \hat{\delta}_{\perp}^{(l)} (\alpha \in \mathbb{R})$, then the layer outputs are inconsistent $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$ (Lemma 4 in Appendix C). It is also implied that orthogonal components with the same magnitude but different directions do not necessarily corrupt the network to the same output. We believe that the perturbation decomposition approach presented in this work allows us to re-examine the intriguing properties of adversarial examples by decomposing the perturbations into their harmful and harmless counterparts.

5 Applications of harmless perturbations

5.1 Privacy protection

We first consider a scenario where users may require employing a pre-trained model on a third-party server to analyze data containing sensitive information (*e.g.*, facial, medical, and credit data) [Schick et al., 2023, Shen et al., 2023, Wu et al., 2023, Liang et al., 2023]. Specifically, either the third-party server or the user provides a pre-trained model, enabling the user to access the network parameters. Subsequently, the user locally generates privacy-preserving data using the available parameters, and then deploys the protected data, along with the network, to the third-party server. To alleviate

²Note that the harmful space is not a linear subspace of $\mathbb{R}^{n^{(l)}}$, since it does not contain $\mathbf{0} \in \mathbb{R}^{n^{(l)}}$.

information leakage from sensitive data, harmless perturbations with sufficiently large magnitudes can be incorporated to original samples. This process renders the generated samples unrecognizable to humans, effectively obscuring sensitive information within the images, without compromising network performance.

To be specific, our goal is to generate a visually unrecognizable image, denoted as $\hat{x} \in \mathbb{R}^{n_{\text{in}}}$, to substitute the original image x, ensuring that its network output is identical with that of the original image x. Specifically, given a DNN with a harmless perturbation subspace $\mathcal{H}^{(0)} \subset \mathbb{R}^{n_{\text{in}}}$ in its first linear layer, and a set of orthonormal bases $\{u_1, u_2, \cdots, u_d\}$ of the subspace $\mathcal{H}^{(0)}$ ($d = dim(\mathcal{H}^{(0)})$), visually unrecognizable harmless perturbations can simply be generated by maximizing the dissimilarity between the original image x and the generated image $\hat{x} \coloneqq x + \sum_{i=1}^d c_i u_i, c_i \in \mathbb{R}, u_i \in \mathbb{R}^{n_{\text{in}}}$. Without loss of generality, we quantify the difference between the two images using the Mean Squared Error (MSE), i.e., $\max_{\{c_1,c_2,\cdots,c_d\}} \frac{1}{n_{\text{in}}} \|\hat{x}-x\|_2^2$. To make the pixels of the generated image in the range [0,1], we add two penalties on the pixels out of bounds, i.e., $\sum_i |\mathbb{I}(\hat{x}_i < 0) \cdot \hat{x}_i| + |\mathbb{I}(\hat{x}_i > 1) \cdot \hat{x}_i|$, as shown in Figure 1(a).

Recovering original images. Reconstructing the original image x from the generated image \hat{x} is a challenging task even if the attacker can access network parameters. Since the parameter matrix A uniquely determine the harmless perturbation subspace, it is equivalent to specifying the subspace $\mathcal{H}^{(0)}$. However, according to Theorem 3, the generated image $\hat{x} \notin \mathcal{H}^{(0)}$ can be decomposed into the sum of an *infinite* number harmless components $\hat{\delta} \in \mathcal{H}^{(0)}$ (Figure 1(b)) and reconstructed images $x^{\text{recon}} := \hat{x} - \hat{\delta}, \forall \hat{\delta} \in \mathcal{H}^{(0)}$. Therefore, the original image cannot be uniquely determined when the magnitude and direction of the harmless perturbation are unknown.

Visual indistinguishability. To quantify the capability of the generated images in preserving privacy for human perception, we evaluated the per-turbed images and the original images on the ceptual similarity using two similarity metrics, i.e., CIFAR-10 dataset. the Structural Similarity Index (SSIM) [Wang et al., 2004] and the Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al., 2018] metrics. Besides, we evaluated the degradation in classification performance of the generated images, compared to the

Table 2: Perceptual similarity between the per-

	Harmless perturbation	Gaussian noise		
SSIM (\lambda)	0.4719	0.6825		
LPIPS (†)	0.2031	0.3007		
Δ accuracy (\downarrow)	$\boldsymbol{0.00\%}$	32.46%		

original images. We compared the privacy-preserving capability of the generated harmless perturbations with the Gaussian noise $\mathcal{N}(0,0.1^2)$ added to each pixel on the ResNet-50. Table 2 shows that the generated harmless perturbations achieved a similar level of privacy preservation as Gaussian noise, but the harmless perturbations completely did not change the DNN's discrimination of images.

Model fingerprint 5.2

Harmless perturbations also can be used for model fingerprints [Finlayson et al., 2024, Zeng et al., 2024] to faithfully reflect the model's changes, as they are determined by the parameter space of the DNN. We demonstrate this usage by considering the simple case of establishing identity fingerprints for two DNNs. Figure 4 illustrates the network's response when adding two different harmless perturbations extracted from two distinct models on input images. Incorporating significant harmless perturbations generated by one model into various input samples preserves the outputs of that model, while applying them to input samples of another

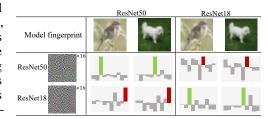


Figure 4: Harmless perturbations (magnified by 16x) can serve as identity fingerprint for models, allowing for tracking changes in closedsource models.

model leads to significant changes in the outputs of another model. This demonstrates harmless perturbations can potentially serve as model fingerprints.

Transferability of harmless perturbations. Typically, given two DNNs with different parameters, their harmless perturbation spaces are not equal, i.e., $\mathcal{P}_1 \neq \mathcal{P}_2$. However, there may exist few harmless perturbations that are transferable and serve as harmless perturbations for both DNNs, i.e., $\delta \in \mathcal{P}_1 \cap \mathcal{P}_2$. To avoid choosing those rare transferable harmless perturbations as model fingerprints, we constraint

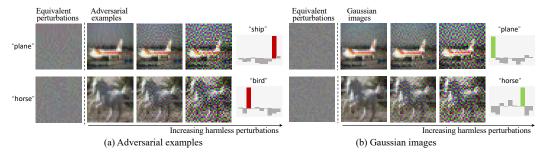


Figure 5: Adding Harmless perturbations to images with different noises ((a) adversarial perturbations (b) Gaussian noises) completely do not change the network output of the perturbed images, regardless of the magnitude of these harmless perturbations. Perturbed images incorporating harmless perturbations of arbitrary magnitude, drawn from the equivalent perturbation space, exhibit an effect on the network output that is equivalent to the impact of images with equivalent (orthogonal) perturbations.

the sampling of model fingerprints solely from the non-intersecting harmless perturbation spaces of the two DNNs, satisfying $\delta_1 \in \mathcal{P}_1 - (\mathcal{P}_1 \cap \mathcal{P}_2)$ and $\delta_2 \in \mathcal{P}_2 - (\mathcal{P}_1 \cap \mathcal{P}_2)$.

5.3 The intriguing properties of DNNs from harmless perturbations

Seeing is not always believing. Surprisingly, we find that distances within the feature space may exhibit considerable variation between DNNs and human perception. Human perceptual systems tend to discern non-equivalence when the magnitude of the perturbation added to a feature significantly exceeds the feature's magnitude. In contrast, DNNs tend to disregard the magnitude of features/perturbations. DNNs are completely unaffected by such harmless perturbations, highlighting a distinctive aspect of DNN robustness. Furthermore, harmless perturbations invalidate distance-based similarity metrics, such as the widely used Euclidean and cosine distances [Mensink et al., 2013, Zhang et al., 2018]. For example, two vectors sampled from harmless/equivalent space, regardless of their magnitude or direction, may deemed dissimilar through these similarity metrics, yet deep networks still regard them as identical. Consequently, there arises a necessity to reassess whether these similarity metrics faithfully reflect the true modelling of similarity by deep networks.

Equivalent adversarial spaces. We revisit the impact of perturbation range/magnitude [Madry et al., 2018, Wang et al., 2019] of adversarial examples on the DNNs. Theorems 3 and 4 demonstrate that *infinitely large, infinitely numerous* features/perturbations are equivalent to their components orthogonal to the harmless subspace. Therefore, for any perturbation, *there exist equivalent (adversarial) perturbation spaces*, ensuring equal attacking capabilities for perturbations. Compared to the Gaussian noises in Figure 5(b), the adversarial perturbations which have similar perturbation magnitudes in Figure 5(a), lead to completely different attack utilities. Interestingly, all adversarial perturbations for each sample in Figure 5(a) have the same attack utility, irrespective of their magnitudes. The equivalent adversarial spaces imply that: 1) *the perturbation magnitude is not a decisive factor attacking the network*, and 2) attention should be paid to the "effective components" of perturbations, *i.e.*, we can further decompose the perturbation in a more fine-grained way. We believe that further exploration of the equivalent space helps to understand the robustness of DNNs.

6 Conclusion

In this paper, we show the existence of harmless perturbations and their subspaces. Such harmless perturbations, regardless of their magnitude, render the network output completely unaltered. Essentially, the harmless perturbation space arises from the usage of non-injective functions within DNNs. We prove that for any linear layer in a DNN where the input dimension n exceeds the output dimension m, there exists a continuous harmless subspace with a dimension of (n-m). We further show the existence of the feature/perturbation space characterized by identical orthogonal components, consistently influencing the network output. Besides, the harmless perturbation space may expand when involving non-linear layers. Our work reveals that DNNs tend to disregard the magnitude of features/perturbations, which highlights a distinctive aspect of DNN robustness. Based on this insight, we utilize the proposed harmless perturbations for hiding sentitive data and model fingerprints.

References

- A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International Conference on Machine Learning (ICML)*, 2018.
- N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. IEEE Symposium on Security and Privacy (SP), 2017.
- P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.
- D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations (ICLR)*, 2016.
- F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *International Conference on Machine Learning (ICML)*, 2020.
- A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. Robustness of classifiers: from adversarial to random noise. *Neural Information Processing Systems (NeurIPS)*, 2016.
- M. Finlayson, X. Ren, and S. Swayamdipta. Logits of api-protected llms leak proprietary information. arXiv preprint arXiv:2403.09539, 2024.
- R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*, 2019.
- J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres. *Workshop of International Conference on Learning Representations (ICLR)*, 2018.
- I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Neural Information Processing Systems (NeurIPS)*, 2019.
- S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)*, 2015.
- J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge. Excessive invariance causes adversarial vulnerability. International Conference on Learning Representations (ICLR), 2019.
- G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. Neural Information Processing Systems (NeurIPS), 2017.
- A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations (ICLR) Workshop*, 2017.
- Y. LeCun and C. Cortes. Mnist handwritten digit database, 2010. URL http://yann.lecun.com/exdb/mnist/.
- Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. *Neural Information Processing Systems (NeurIPS)*, 1990.
- Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao, Y. Wang, L. Shou, M. Gong, and N. Duan. Taskmatrix.ai: Completing tasks by connecting foundation models with millions of apis. *arXiv* preprint arXiv:2303.16434, 2023.
- A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. International Conference on Machine Learning (ICML), 2013.
- A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations (ICLR)*, 2018.

- T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11): 2624–2637, 2013. doi: 10.1109/TPAMI.2013.83.
- S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. Computer Vision and Pattern Recognition (CVPR), 2017.
- V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. *International Conference on Machine Learning (ICML)*, 2010.
- Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *In Neural Information Processing Systems (NeurIPS) Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
- A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. The ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS), 2017.
- H. Salman, A. Ilyas, L. Engstrom, S. Vemprala, A. Madry, and A. Kapoor. Unadversarial examples: Designing objects for robust vision. Neural Information Processing Systems (NeurIPS), 2021.
- D. Scherer, A. Muller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. *International Conference on Artificial Neural Networks (ICANN)*, 2010.
- T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *Neural Information Processing Systems* (*NeurIPS*), 2023.
- A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. Are adversarial examples inevitable? *International Conference on Learning Representations (ICLR)*, 2019.
- Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Neural Information Processing Systems (NeurIPS)*, 2023.
- W. Si, S. Li, S. Park, I. Lee, and O. Bastani. Angelic patches for improving third-party object detector performance. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014.
- M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning (ICML)*, 2019.
- F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. The space of transferable adversarial examples. *ArXiv preprint arXiv:1704.03453*, 2017.
- D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representations (ICLR)*, 2019.
- H. Wang, X. Wu, Z. Huang, and E. P. Xing. High frequency component helps explain the generalization of convolutional neural networks. Computer Vision and Pattern Recognition (CVPR), 2020.
- J. Wang, Z. Yin, P. Hu, A. Liu, R. Tao, H. Qin, X. Liu, and D. Tao. Defensive patches for robust recognition in the physical world. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2022.
- Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, and Q. Gu. On the convergence and robustness of adversarial training. International Conference on Machine Learning (ICML), 2019.
- Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 2004.

- C. Wu, S.-K. Yin, W. Oi, X. Wang, Z. Tang, and N. Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671, 2023.
- D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and G. Justin. A fourier perspective on model robustness in computer vision. Neural Information Processing Systems (NeurIPS), 2019.
- B. Zeng, C. Zhou, X. Wang, and Z. Lin. Human-readable fingerprint for large language models. arXiv preprint arXiv:2312.04828, 2024.
- H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. International Conference on Machine Learning (ICML), 2019.
- R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Proofs of Theorems and Remarks

In this section, we prove the theorems and remarks in the paper.

Theorem 1 (Dimension of harmless perturbation subspace for a linear layer) Given a linear layer $\mathcal{L}(x) = Ax \in \mathbb{R}^m$ and an input sample $x \in \mathbb{R}^n$, where the parameter matrix $A \in \mathbb{R}^{m \times n}$. The dimension of the subspace for harmless perturbations is dim(S) = n - rank(A).

Proof. The rank-nullity theorem in linear algebra states that, for an $m \times n$ matrix A, which represents a linear map $\mathcal{L}: \mathbb{R}^n \to \mathbb{R}^m$, the number of columns of a matrix A is the sum of the rank of A and the nullity of A, *i.e.*,

$$rank(A) + nullity(A) = n (1)$$

where the rank of A is the dimension of the column space of A, i.e., rank(A) = dim(C(A)), and the nullity of A is the dimension of the nullspace of A, i.e., nullity(A) = dim(N(A)).

Here, the harmless perturbation subspace for a linear layer \mathcal{L} is $\mathcal{S} = \{\delta \in \mathbb{R}^n | A(x+\delta) = Ax\} =$ $\{\delta|A\delta=\mathbf{0}\}=N(A)$, where the nullspace of the matrix A is $N(A):=\{v\in\mathbb{R}^n|Av=\mathbf{0}\}.$

Thus, the dimension of the subspace for harmless perturbations is dim(S) = dim(N(A)) =nullity(A) = n - rank(A).

Remark 1 Consider the case that the input dimension of the linear layer is less than or equal to the output dimension, i.e., $n \le m$. In this case, if the column vectors of the parameter matrix A are linearly independent, then the dimension of the subspace for harmless perturbations is dim(S) = 0.

Proof. According to Theorem 1, the dimension of the subspace for harmless perturbations is $dim(\mathcal{S}) = n - rank(A)$.

The rank of $A \in \mathbb{R}^{m \times n}$ satisfies $rank(A) \leq \min(m,n)$. Consider the case that $n \leq m$, then $rank(A) \leq n$.

If the column vectors of A are linearly independent, then rank(A) = n.

Thus, the dimension of the subspace for harmless perturbations is dim(S) = n - n = 0.

Remark 2 Consider the case that the input dimension of the linear layer is greater than the output dimension, i.e., n > m. In this case, if the row vectors of the parameter matrix A are linearly independent, then the dimension of the subspace for harmless perturbations is dim(S) = n - m.

Proof. According to Theorem 1, the dimension of the subspace for harmless perturbations is $dim(\mathcal{S}) = n - rank(A)$.

The rank of $A \in \mathbb{R}^{m \times n}$ satisfies $rank(A) \leq \min(m,n)$. Consider the case that n > m, then rank(A) < m.

12

If the row vectors of A are linearly independent, then rank(A) = m.

Thus, the dimension of the subspace for harmless perturbations is dim(S) = n - m.

Theorem 2 (Set of harmless perturbations for two-layer neural networks) Given a two-layer neural network $f(x) = \sigma(Ax)$, where σ represents any function. If σ is an injective function, the set of harmless perturbations on the input \mathcal{P} for f is $\mathcal{P} = \mathcal{P}^{(0)}$. Otherwise, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$. Here, $\mathcal{P}^{(1)} = \{\delta | A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)} \cap C(A)\}$ is determined by the specific function σ and the input sample x.

Proof. The set of input perturbations for the (l+1)-th layer is defined as $\mathcal{P}^{(l)} \coloneqq \{\delta \in \mathbb{R}^{n_{\text{in}}} | z^{(l)} + \delta^{(l)} = (f^{(l)} \circ \cdots \circ f^{(1)})(x+\delta), \forall \delta^{(l)} \in \mathcal{H}^{(l)}\}$ such that the perturbations on the l-th intermediate-layer features have no effect on the network output, i.e., $\delta^{(l)} \in \mathcal{H}^{(l)}$.

According to Lemma 1, given a two-layer neural network $f = \sigma(Ax)$, the set of harmless input perturbations for the network f with two layers is $\mathcal{P} = \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}, \mathcal{P}^{(0)} \coloneqq \mathcal{H}^{(0)}$.

Suppose σ is an injective function, then $\mathcal{H}^{(1)} = \{\mathbf{0}\}$, according to Lemma 1.1. Thus, $\mathcal{P}^{(1)} = \{\delta | z^{(1)} + \mathbf{0} = A(x + \delta)\} = \{\delta | A\delta = \mathbf{0}\} = N(A)$, where $z^{(1)} = Ax$. Therefore, the set of harmless perturbations on the input \mathcal{P} for $\sigma(Ax)$ is $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} = \mathcal{P}^{(0)}$.

Otherwise, if σ is not an injective function, then $\mathcal{H}^{(1)} \neq \{\mathbf{0}\}$, according to Lemma 1.1. Therefore, the harmless perturbation space may expand, $\mathit{i.e.}$, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$. Then, $\mathcal{P}^{(1)} = \{\delta|z^{(1)} + \delta^{(1)} = A(x+\delta), \forall \delta^{(1)} \in \mathcal{H}^{(1)}\} = \{\delta|A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)}\}$, where $z^{(1)} = Ax$. Notice that the equation $A\delta = \delta^{(1)}(\delta \neq \mathbf{0})$ has a solution (meaning at least one solution) if and only if $\delta^{(1)}$ is in the column space of A, $\mathit{i.e.}$, $\delta^{(1)} \in C(A)$. Then, $\mathcal{P}^{(1)} = \{\delta|A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)}, \delta^{(1)} \in \mathcal{H}^{(1)}, \delta^{(1)} \in \mathcal{H}^{(1)}, \delta^{(1)} \in \mathcal{H}^{(1)}, \delta^{(1)} \in \mathcal{H}^{(1)}$ is determined by the specific function σ and the input sample x. Please see Lemmas 1.2 to 1.4 for specific functions.

Theorem 3 (Arbitrary decomposition of perturbations) If there exists a harmless perturbation subspace in the (l+1)-th linear layer, *i.e.*, $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$, given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it can be arbitrarily decomposed into the sum of a harmless perturbation and a harmful perturbation, *i.e.*, $\delta^{(l)} = \delta_a^{(l)} + \delta_b^{(l)}$, $\forall \delta_a^{(l)} \in \mathcal{H}^{(l)}$ and $\delta_b^{(l)} \notin \mathcal{H}^{(l)}$. Then, $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta_b^{(l)})$.

Proof. Let the matrix $A \in \mathbb{R}^{n^{(l+1)} \times n^{(l)}}$ with linearly independent rows/columns have an equivalent effect with the parameters of the linear layer, i.e., $z^{(l+1)} = f^{(l+1)}(z^{(l)}) = Az^{(l)} \in \mathbb{R}^{n^{(l+1)}}$.

$$\text{Let } \mathcal{H}^{(l)} = \{ \delta^{(l)} \in \mathbb{R}^{n^{(l)}} | A(z^{(l)} + \delta^{(l)}) = Az^{(l)} \} = \{ \delta^{(l)} | A\delta^{(l)} = \mathbf{0} \} = N(A) \neq \{ \mathbf{0} \}.$$

(1) First, we will prove that given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it can be arbitrarily decomposed into the sum of a harmless perturbation and a harmful perturbation, *i.e.*, $\delta^{(l)} = \delta_a^{(l)} + \delta_b^{(l)}$, $\forall \delta_a^{(l)} \in \mathcal{H}^{(l)}$ and $\delta_b^{(l)} \notin \mathcal{H}^{(l)}$. In other words, it is equivalent to proving that $\forall \delta_a^{(l)} \in \mathcal{H}^{(l)}$, $\delta_b^{(l)} \coloneqq \delta^{(l)} - \delta_a^{(l)} \notin \mathcal{H}^{(l)}$.

To achieve this, we prove $\delta_b^{(l)} \notin \mathcal{H}^{(l)}$ by contradiction. Assume that $\delta_b^{(l)} \in \mathcal{H}^{(l)}$, then we can obtain $\forall \delta_a^{(l)} \in \mathcal{H}^{(l)}, f^{(l+1)}(\delta_a^{(l)}) = A\delta_a^{(l)} = \mathbf{0}$ and $\forall \delta_b^{(l)} \coloneqq \delta^{(l)} - \delta_a^{(l)} \in \mathcal{H}^{(l)}, f^{(l+1)}(\delta_b^{(l)}) = A\delta_b^{(l)} = \mathbf{0}$. Then, $\forall \delta^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta^{(l)}) = A\delta^{(l)} = A(\delta_a^{(l)} + \delta_b^{(l)}) = A\delta_a^{(l)} + A\delta_b^{(l)} = \mathbf{0}$, which contradicts $\forall \delta^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta^{(l)}) = A\delta^{(l)} \neq \mathbf{0}$.

Thus, given any perturbation $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it can be arbitrarily decomposed into the sum of a harmless perturbation $\delta^{(l)}_a \in \mathcal{H}^{(l)}$ and a harmful perturbation $\delta^{(l)}_b \coloneqq \delta^{(l)} - \delta^{(l)}_a \notin \mathcal{H}^{(l)}$.

(2) Second, we will prove that $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_b)$.

We have $\forall \delta_a^{(l)} \in \mathcal{H}^{(l)}, f^{(l+1)}(\delta_a^{(l)}) = A\delta_a^{(l)} = \mathbf{0}$, and $\forall \delta_b^{(l)} := \delta^{(l)} - \delta_a^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta_b^{(l)}) = A\delta_b^{(l)} \neq \mathbf{0}$.

$$\begin{split} f^{(l+1)}(\delta^{(l)}) &= A\delta^{(l)} \\ &= A(\delta_a^{(l)} + \delta_b^{(l)}) \\ &= A\delta_a^{(l)} + A\delta_b^{(l)} \\ &= A\delta_b^{(l)} = f^{(l+1)}(\delta_b^{(l)}) \end{split} \tag{2}$$

Thus, the layer output of any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$ is equivalent to the layer output of its corresponding harmful component $\delta^{(l)}_b \coloneqq \delta^{(l)} - \delta^{(l)}_a \notin \mathcal{H}^{(l)}, \forall \delta^{(l)}_a \in \mathcal{H}^{(l)}$.

Theorem 4 (Orthogonal decomposition of perturbations) If there exists a harmless perturbation subspace in the (l+1)-th linear layer, i.e., $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$, given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a unique decomposition $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$ such that the parallel component $\delta^{(l)}_{\parallel} = P\delta^{(l)} \in \mathcal{H}^{(l)}$ and the orthogonal component $\delta^{(l)}_{\perp} = (I-P)\delta^{(l)} \notin \mathcal{H}^{(l)}$. Then, $f^{(l+1)}(\delta^{(l)}_{\parallel}) = \mathbf{0}$ and $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_{\perp})$.

Proof. Let the matrix $A \in \mathbb{R}^{n^{(l+1)} \times n^{(l)}}$ with linearly independent rows/columns have an equivalent effect with the parameters of the linear layer, i.e., $z^{(l+1)} = f^{(l+1)}(z^{(l)}) = Az^{(l)} \in \mathbb{R}^{n^{(l+1)}}$.

Let
$$\mathcal{H}^{(l)} = \{ \delta^{(l)} \in \mathbb{R}^{n^{(l)}} | A(z^{(l)} + \delta^{(l)}) = Az^{(l)} \} = \{ \delta^{(l)} | A\delta^{(l)} = \mathbf{0} \} = N(A) \neq \{ \mathbf{0} \}.$$

(1) First, we will prove that given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a *unique* decomposition $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$ such that $\delta^{(l)}_{\parallel} = P\delta^{(l)} \in \mathcal{H}^{(l)}$ and $\delta^{(l)}_{\perp} = (I - P)\delta^{(l)} \notin \mathcal{H}^{(l)}$.

Here, $P = U(U^{\top}U)^{-1}U^{\top}$ represents the projection matrix onto the subspace $\mathcal{H}^{(l)}$ of $\mathbb{R}^{n^{(l)}}$, and the matrix $U \in \mathbb{R}^{n^{(l)} \times dim(\mathcal{H}^{(l)})}$ denotes a set of $dim(\mathcal{H}^{(l)})$ orthogonal bases for the subspace $\mathcal{H}^{(l)}$.

(a) Orthogonal decomposition. We will prove that $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a decomposition $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$.

According to Theorem 3, given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it can be arbitrarily decomposed into the sum of a harmless perturbation and a harmful perturbation. Then, let $\delta^{(l)}_{\parallel} \in \mathcal{H}^{(l)}$ and $\delta^{(l)}_{\perp} \notin \mathcal{H}^{(l)}$.

Let $U = [u_1, u_2, \cdots, u_d] \in \mathbb{R}^{n^{(l)} \times dim(\mathcal{H}^{(l)})}$, where $d = dim(\mathcal{H}^{(l)})$. Here, $u_1, u_2, \cdots, u_d \in \mathbb{R}^{n^{(l)}}$ are a set of orthonormal bases that spans the subspace $\mathcal{H}^{(l)}$.

Since $\delta_{\parallel}^{(l)} \in \mathcal{H}^{(l)}$, it can be represented as a linear combination of orthonormal bases, *i.e.*, $\delta_{\parallel}^{(l)} = \sum_{i=1}^{\dim(\mathcal{H}^{(l)})} c_i u_i, c_i \in \mathbb{R}$. Rewrite $\delta_{\parallel}^{(l)}$ into matrix form, *i.e.*, $\delta_{\parallel}^{(l)} = Uc, c = [c_1, c_2, \cdots, c_d]^{\top}$.

Since $\delta_{\perp}^{(l)} \coloneqq \delta^{(l)} - \delta_{\parallel}^{(l)} = \delta^{(l)} - Uc$ is orthogonal to the subspace $\mathcal{H}^{(l)}$, $\delta_{\perp}^{(l)}$ is orthogonal to the orthonormal bases u_1, u_2, \cdots, u_d , respectively. Then, it can derived that $u_1^{\top}(\delta^{(l)} - Uc) = 0, \cdots, u_d^{\top}(\delta^{(l)} - Uc) = 0$. Rewrite these equations into matrix form, *i.e.*, $U^{\top}(\delta^{(l)} - Uc) = 0$. Thus, $c = (U^{\top}U)^{-1}U^{\top}\delta^{(l)}$ ($U^{\top}U$ is invertible, since U has full column rank).

Thus, $\delta_{\parallel}^{(l)} = Uc = U(U^{\top}U)^{-1}U^{\top}\delta^{(l)} = P\delta^{(l)} \in \mathcal{H}^{(l)}$ and $\delta_{\perp}^{(l)} = \delta^{(l)} - \delta_{\parallel}^{(l)} = (I - P)\delta^{(l)} \notin \mathcal{H}^{(l)}$. We have proved that $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a decomposition $\delta^{(l)} = \delta_{\parallel}^{(l)} + \delta_{\parallel}^{(l)}$.

(b) Uniqueness of orthogonal decomposition. We will prove that the orthogonal decomposition on a subspace is *unique*.

Let $\mathcal{H}^{(l)} \subset \mathbb{R}^{n^{(l)}}$ be a subspace of $\mathbb{R}^{n^{(l)}}$, and let P_1 , P_2 be arbitrary projection matrices onto $\mathcal{H}^{(l)}$, we prove that the orthogonal projector onto $\mathcal{H}^{(l)}$ is unique, *i.e.*, $P_1 = P_2$.

For $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$, the parallel components are $\delta^{(l)}_{\parallel,1} = P_1 \delta^{(l)} \in \mathcal{H}^{(l)}$ and $\delta^{(l)}_{\parallel,2} = P_2 \delta^{(l)} \in \mathcal{H}^{(l)}$. The corresponding orthogonal components satisfy $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$, $(\delta^{(l)} - P_1 \delta^{(l)}) \perp \mathcal{H}^{(l)}$ and $(\delta^{(l)} - P_2 \delta^{(l)}) \perp \mathcal{H}^{(l)}$, thus, $(P_1 - P_2) \delta^{(l)} \perp \mathcal{H}^{(l)}$. However, $(P_1 - P_2) \delta^{(l)} = \delta^{(l)}_{\parallel,1} - \delta^{(l)}_{\parallel,2} \in \mathcal{H}^{(l)}$, then we have $(P_1 - P_2) \delta^{(l)} = \mathbf{0}$ for every $\delta^{(l)}$. Therefore, $P_1 = P_2$.

Thus, given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, the orthogonal decomposition of perturbation $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$ on a subspace $\mathcal{H}^{(l)} \subset \mathbb{R}^{n^{(l)}}$ is unique.

(2) Second, we will prove that $f^{(l+1)}(\delta^{(l)}_{\parallel})=\mathbf{0}$ and $f^{(l+1)}(\delta^{(l)})=f^{(l+1)}(\delta^{(l)}_{\perp})$.

We have $\delta_{\parallel}^{(l)} \in \mathcal{H}^{(l)}, f^{(l+1)}(\delta_{\parallel}^{(l)}) = A\delta_{\parallel}^{(l)} = \mathbf{0}$, and $\delta_{\perp}^{(l)} := \delta^{(l)} - \delta_{\parallel}^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta_{\perp}^{(l)}) = A\delta_{\perp}^{(l)} \neq \mathbf{0}$.

$$\begin{split} f^{(l+1)}(\delta^{(l)}) &= A\delta^{(l)} \\ &= A(\delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}) \\ &= A\delta^{(l)}_{\parallel} + A\delta^{(l)}_{\perp} \\ &= A\delta^{(l)}_{\perp} = f^{(l+1)}(\delta^{(l)}_{\perp}) \end{split} \tag{3}$$

Thus, the layer output of any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$ is equivalent to the layer output of its unique orthogonal component $\delta^{(l)}_{\perp} := \delta^{(l)} - \delta^{(l)}_{\parallel} = (I - P)\delta^{(l)} \notin \mathcal{H}^{(l)}$.

Theorem 5 (Identical impact of a family of perturbations) If there exists a harmless perturbation subspace in the (l+1)-th linear layer, i.e., $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$, given two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)}$, $\hat{\delta}^{(l)} \notin \mathcal{H}^{(l)}$, if their orthogonal components are the same, i.e., $\delta^{(l)}_{\perp} = \hat{\delta}^{(l)}_{\perp}$, then $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$.

$$\begin{split} &\textit{Proof.} \ \, \text{According to Theorem 4,} \ \forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}} \ \, \text{and} \ \, \delta^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_{\perp}). \\ &\text{Similarly,} \ \, \forall \hat{\delta}^{(l)} \neq \delta^{(l)} \in \mathbb{R}^{n^{(l)}} \ \, \text{and} \ \, \hat{\delta}^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\hat{\delta}^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)}_{\perp}). \\ &\text{Thus, given } \delta^{(l)}_{\perp} = \hat{\delta}^{(l)}_{\perp}, \text{ then } f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_{\perp}) = f^{(l+1)}(\hat{\delta}^{(l)}_{\perp}) = f^{(l+1)}(\hat{\delta}^{(l)}). \end{split}$$

B Proofs of Corollaries

In this section, we prove the corollaries in the paper.

Corollary 1 (Dimension of harmless perturbation subspace for a convolutional layer) Given a convolutional layer $f^{(l+1)}$ with linearly independent vectorized kernels whose kernel size is larger than or equal to the stride, the input feature is $z^{(l)} \in \mathbb{R}^{C_{\text{in}} \times H_{\text{in}} \times W_{\text{in}}}$ and the output feature is $z^{(l+1)} = f^{(l+1)}(z^{(l)}) \in \mathbb{R}^{C_{\text{out}} \times H_{\text{out}} \times W_{\text{out}}}$. If the input dimension of the convolutional layer is greater than the output dimension, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = C_{\text{in}}H_{\text{in}}W_{\text{in}} - C_{\text{out}}H_{\text{out}}W_{\text{out}}$. Otherwise, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

Proof. Let the matrix $A \in \mathbb{R}^{(C_{\text{out}}H_{\text{out}}W_{\text{out}})\times(C_{\text{in}}H_{\text{in}}W_{\text{in}})}$ with linearly independent rows/columns (see Appendix E for the generation of matrix A) have an equivalent effect with the parameters of the

convolutional layer, i.e., $\hat{z}^{(l+1)} = A\hat{z}^{(l)}$, $\hat{z}^{(l)} \in \mathbb{R}^{C_{\text{in}}H_{\text{in}}W_{\text{in}}}$ is the vectorized feature $z^{(l)}$, and $\hat{z}^{(l+1)} \in \mathbb{R}^{C_{\text{out}}H_{\text{out}}W_{\text{out}}}$ is the vectorized feature $z^{(l+1)}$.

Let
$$\mathcal{H}^{(l)} = \{\delta^{(l)} \in \mathbb{R}^{C_{\text{in}}H_{\text{in}}W_{\text{in}}} | A(\hat{z}^{(l)} + \delta^{(l)}) = A\hat{z}^{(l)}\} = \{\delta^{(l)} | A\delta^{(l)} = \mathbf{0}\} = N(A).$$

According to Remark 1, if $C_{\text{in}}H_{\text{in}}W_{\text{in}} \leq C_{\text{out}}H_{\text{out}}W_{\text{out}}$ and the column vectors of A are linearly independent, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = C_{\text{in}}H_{\text{in}}W_{\text{in}} - C_{\text{in}}H_{\text{in}}W_{\text{in}} = 0$. Thus, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

According to Remark 2 in Appendix A, if $C_{\rm in}H_{\rm in}W_{\rm in}>C_{\rm out}H_{\rm out}W_{\rm out}$ and the row vectors of A are linearly independent, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)})=C_{\rm in}H_{\rm in}W_{\rm in}-C_{\rm out}H_{\rm out}W_{\rm out}$.

Thus, if the input dimension of the convolutional layer is greater than the output dimension, *i.e.*, $C_{\text{in}}H_{\text{in}}W_{\text{in}} > C_{\text{out}}H_{\text{out}}W_{\text{out}}$, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = C_{\text{in}}H_{\text{in}}W_{\text{in}} - C_{\text{out}}H_{\text{out}}W_{\text{out}}$. Otherwise, $\mathcal{H}^{(l)} = \{0\}$.

Corollary 2 (Dimension of harmless perturbation subspace for a fully-connected layer). Given a fully-connected layer $z^{(l+1)} = f^{(l+1)}(z^{(l)}) = W^{\top}z^{(l)} \in \mathbb{R}^{N_{\text{out}}}$ with linearly independent rows/columns in the matrix W, the input feature is $z^{(l)} \in \mathbb{R}^{N_{\text{in}}}$. If the input dimension of the fully-connected layer is greater than the output dimension, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = N_{\text{in}} - N_{\text{out}}$. Otherwise, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

Proof. Let the matrix $A = W^{\top} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{in}}}$ with linearly independent rows/columns have an equivalent effect with the parameters of the fully-connected layer. Let $\mathcal{H}^{(l)} = \{\delta^{(l)} \in \mathbb{R}^{N_{\text{in}}} | A(z^{(l)} + \delta^{(l)}) = Az^{(l)}\} = \{\delta^{(l)} | A\delta^{(l)} = \mathbf{0}\} = N(A)$.

According to Remark 1, if $N_{\rm in} \leq N_{\rm out}$ and the column vectors of A are linearly independent, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = N_{\rm in} - N_{\rm in} = 0$. Thus, $\mathcal{H}^{(l)} = \{0\}$.

According to Remark 2 in Appendix A, if $N_{\rm in} > N_{\rm out}$ and the row vectors of A are linearly independent, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = N_{\rm in} - N_{\rm out}$.

Thus, if the input dimension of the fully-connected layer is greater than the output dimension, *i.e.*, $N_{\rm in} > N_{\rm out}$, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = N_{\rm in} - N_{\rm out}$. Otherwise, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

C Proofs of Lemmas

In this section, we prove the lemmas in the paper.

Lemma 1 (Set of harmless input perturbations for a DNN) The set of harmless input perturbations for a DNN f with L layers is derived as $\mathcal{P} = \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, $\mathcal{P}^{(0)} := \mathcal{H}^{(0)}$, $\mathcal{P} \subset \mathbb{R}^{n_{\text{in}}}$.

Proof. The set of harmelss perturbations for a DNN f is defined as $\mathcal{P} := \{\delta \in \mathbb{R}^{n_{\text{in}}} | f(x+\delta) = f(x) \}$.

The set of harmless perturbations on the (l+1)-th layer of a DNN f is defined as $\mathcal{H}^{(l)} \coloneqq \{\delta^{(l)} \in \mathbb{R}^{n^{(l)}} | f^{(l+1)}(z^{(l)} + \delta^{(l)}) = f^{(l+1)}(z^{(l)})\}$. Here, $z^{(l)} \in \mathbb{R}^{n^{(l)}}$ represents the features of the l-th intermediate layer of the input sample x, and $\delta^{(l)}$ denotes the perturbations added to the features $z^{(l)}$.

The set of input perturbations for the (l+1)-th layer is defined as $\mathcal{P}^{(l)} := \{\delta \in \mathbb{R}^{n_{\text{in}}} | z^{(l)} + \delta^{(l)} = (f^{(l)} \circ \cdots \circ f^{(1)})(x+\delta), \forall \delta^{(l)} \in \mathcal{H}^{(l)}\}$ such that the perturbations on the l-th intermediate-layer features have no effect on the network output, i.e., $\delta^{(l)} \in \mathcal{H}^{(l)}$.

To prove that $\mathcal{P} = \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, We will prove that $(1) \ \forall \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, $\delta \in \mathcal{P}$, and $(2) \ \forall \delta \in \mathcal{P}$, $\delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, respectively. Notice that the (l+1)-th layer can be arbitrary layer (linear or non-linear contents).

layer), \mathcal{P} , $\mathcal{H}^{(l)}$ and $\mathcal{P}^{(l)}$ are instance-specific, given a specific input sample x and the corresponding feature $z^{(l)}$ ($x := z^{(0)}$).

(1) We will prove that $\forall \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}, \delta \in \mathcal{P}.$

Given an input sample $x, \forall \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$ means that $\forall \delta, \exists l \in \{0,1,\cdots,L-1\}$ such that $\delta \in \mathcal{P}^{(l)} \coloneqq \{\delta \in \mathbb{R}^{n_{\text{in}}}|z^{(l)} + \delta^{(l)} = (f^{(l)} \circ \cdots \circ f^{(1)})(x+\delta), \forall \delta^{(l)} \in \mathcal{H}^{(l)}\}, \mathcal{P}^{(0)} \coloneqq \mathcal{H}^{(0)}$. Then, the corresponding harmless perturbations $\delta^{(l)}$ on the l-th intermediate-layer features satisfy $f^{(l+1)}(z^{(l)} + \delta^{(l)}) = f^{(l+1)}(z^{(l)})$ due to $\delta^{(l)} \in \mathcal{H}^{(l)}$. Therefore, the layer outputs of the subsequent layers are all equal, $\forall l' \in \{l, l+1, \cdots, L-1\}, f^{(l'+1)}(z^{(l')} + \delta^{(l')}) = f^{(l'+1)}(z^{(l')})$ and finally $f^{(L)}(z^{(L-1)} + \delta^{(L-1)}) = f^{(L)}(z^{(L-1)})$. Thus, $f(x+\delta) = f(x)$ can be derived and hence $\delta \in \mathcal{P} \coloneqq \{\delta \in \mathbb{R}^{n_{\text{in}}} | f(x+\delta) = f(x)\}$.

Thus, we have proved that $\forall \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}, \delta \in \mathcal{P}$.

(2) We will prove that $\forall \delta \in \mathcal{P}, \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$.

We will prove $\forall \delta \in \mathcal{P}, \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$ by contradiction. Assume that $\exists \delta \in \mathcal{P}, \delta \notin \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, $\mathcal{P}^{(0)} \coloneqq \mathcal{H}^{(0)}$. Here, $\delta \notin \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$ means that $\forall l \in \{0,1,\cdots,L-1\}, \delta \notin \mathcal{P}^{(l)}$. Then, the corresponding harmless perturbations $\delta^{(l)}$ on the l-th intermediate-layer features satisfy $\forall l \in \{0,1,\cdots,L-1\}, \delta^{(l)} \notin \mathcal{H}^{(l)} \coloneqq \{\delta^{(l)} \in \mathbb{R}^{n^{(l)}} | f^{(l+1)}(z^{(l)} + \delta^{(l)}) = f^{(l+1)}(z^{(l)}) \}$. Therefore, the outputs of each layer are not equal, $i.e., \forall l \in \{0,1,\cdots,L-1\}, f^{(l+1)}(z^{(l)} + \delta^{(l)}) \neq f^{(l+1)}(z^{(l)})$ and finally $f^{(L)}(z^{(L-1)} + \delta^{(L-1)}) \neq f^{(L)}(z^{(L-1)})$. Thus, $f(x+\delta) \neq f(x)$, which is in contradiction to $\delta \in \mathcal{P} \coloneqq \{\delta \in \mathbb{R}^{n_{in}} | f(x+\delta) = f(x) \}$.

Thus, we have proved that $\forall \delta \in \mathcal{P}, \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$.

Since (1) $\forall \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}, \delta \in \mathcal{P}$ and (2) $\forall \delta \in \mathcal{P}, \delta \in \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}$, we have proved that $\mathcal{P} = \bigcup_{l=0}^{L-1} \mathcal{P}^{(l)}, \mathcal{P} \subset \mathbb{R}^{n_{\text{in}}}$.

Lemma 1.1 (Set of harmless perturbations for injective functions) If the (l+1)-th layer $f^{(l+1)}$ is an injective function, the set of harmless perturbations on the (l+1)-th layer of a DNN f is $\mathcal{H}^{(l)} = \{\mathbf{0}\}$. Otherwise, $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$.

Proof. According to Definition 2, the set of harmless perturbations on the (l+1)-th layer of a DNN f is defined as $\mathcal{H}^{(l)} := \{\delta^{(l)} \in \mathbb{R}^{n^{(l)}} | f^{(l+1)}(z^{(l)} + \delta^{(l)}) = f^{(l+1)}(z^{(l)}) \}.$

If $f^{(l+1)}$ is an injective function that maps distinct elements of its domain to distinct elements, that is, $x_1 \neq x_2$ implies $f^{(l+1)}(x_1) \neq f^{(l+1)}(x_2)$. Since $\forall \delta^{(l)} \neq \mathbf{0}, z^{(l)} + \delta^{(l)} \neq z^{(l)}$, the function output satisfies $f^{(l+1)}(z^{(l)} + \delta^{(l)}) \neq f^{(l+1)}(z^{(l)})$, then, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$.

Otherwise, if $f^{(l+1)}$ is not an injective function, that is, $\exists x_1 \neq x_2$ satisfies $f^{(l+1)}(x_1) = f^{(l+1)}(x_2)$. Since $\exists \delta^{(l)} \neq \mathbf{0}, z^{(l)} + \delta^{(l)} \neq z^{(l)}$, the function output satisfies $f^{(l+1)}(z^{(l)} + \delta^{(l)}) = f^{(l+1)}(z^{(l)})$, then, $\mathcal{H}^{(l)} \neq \{\mathbf{0}\}$.

Lemma 1.2 (Set of harmless perturbations for ReLU layers) Suppose $f^{(l+1)}$ is the ReLU layer, $\mathcal{H}^{(l)} = \left\{ \delta^{(l)} | \forall i, \delta_i^{(l)} = \begin{cases} 0, & z_i^{(l)} > 0 \\ t(t \leq -z_i^{(l)}), & z_i^{(l)} \leq 0 \end{cases} \right\}, \text{ which is determined by intermediate-layer features } z^{(l)} \text{ and hence the input sample } x.$

Proof. Suppose $f^{(l+1)}$ is the ReLU layer, which is not an injective function. The set of harmless perturbations for the ReLU layer $f^{(l+1)}$ is $\mathcal{H}^{(l)} = \{\delta^{(l)}|\text{ReLU}(z^{(l)} + \delta^{(l)}) = \text{ReLU}(z^{(l)})\}$. Since the ReLU layer outputs 0 for all inputs that are not positive, then the i-th element (dimension) of $\delta^{(l)}$ in $\mathcal{H}^{(l)}$ satisfies,

$$\forall i, \delta_i^{(l)} = \begin{cases} 0, & z_i^{(l)} > 0 \\ t(t \le -z_i^{(l)}), & z_i^{(l)} \le 0 \end{cases} \tag{4}$$

Here, t denotes any real number not greater than $-z_i^{(l)}$. In other words, each element of $\delta^{(l)}$ in $\mathcal{H}^{(l)}$ is determined by the value of the corresponding dimension of the intermediate-layer feature $z^{(l)}$. Some dimensions of $\delta^{(l)}$ are zero, while others are not greater than the corresponding dimensions of the intermediate-layer feature's (negative) value.

Therefore, suppose
$$f^{(l+1)}$$
 is the ReLU layer, $\mathcal{H}^{(l)} = \{\delta^{(l)} | \forall i, \delta_i^{(l)} = \begin{cases} 0, & z_i^{(l)} > 0 \\ t(t \leq -z_i^{(l)}), & z_i^{(l)} \leq 0 \end{cases}$,

which is determined by intermediate-layer features $z^{(l)}$ and hence the input sample x.

Besides, let us further consider the impact of the input sample x on the harmless perturbation space of the ReLU layer. Specifically, let us consider the case of the harmless perturbation space for a two-layer neural network with the ReLU layer, i.e., f(x) = ReLU(Ax).

According to Lemma 1.2 and Theorem 2, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$ and $\mathcal{P}^{(1)} = \{\delta | A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)} \cap C(A)\}$. Note that $\mathcal{P}^{(1)}$ is determined by the input sample x. In an extreme case, if every element of $z^{(1)} = Ax$ is positive, $\mathcal{H}^{(1)} = \{\mathbf{0}\}$ and $\mathcal{P}^{(1)} = \{\delta | A\delta = \mathbf{0}\} = N(A)$. Thus, the set of harmless perturbations on the input \mathcal{P} for ReLU(Ax) is $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} = \mathcal{P}^{(0)}$. Otherwise, if every element of $z^{(1)} = Ax$ is not positive, then $\mathcal{H}^{(1)} = \{\delta^{(1)} | \forall i, \delta_i^{(1)} \leq -z_i^{(1)}\}$ and $\mathcal{P}^{(1)} = \{\delta | A\delta = \delta^{(1)}, \forall \delta^{(1)} \in \mathcal{H}^{(1)} \cap C(A)\}$. Thus, the set of harmless perturbations on the input \mathcal{P} for ReLU(Ax) is $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$.

Lemma 1.3 (Set of harmless perturbations for Softmax layers) Suppose $f^{(l+1)}$ is the Softmax layer, $\mathcal{H}^{(l)} = \{c \cdot \mathbf{1}, c \in \mathbb{R}\}.$

Proof. Suppose $f^{(l+1)}$ is the Softmax layer, which is not an injective function. The set of harmless perturbations for the Softmax layer $f^{(l+1)}$ is $\mathcal{H}^{(l)} = \{\delta^{(l)}|\operatorname{Softmax}(z^{(l)} + \delta^{(l)}) = \operatorname{Softmax}(z^{(l)})\}$. Since the Softmax layer is invariant when translating the same value in each coordinate, that is, adding $\mathbf{c} = (c, c, \cdots, c)$ to the input yields $\operatorname{Softmax}(\mathbf{x} + \mathbf{c}) = \operatorname{Softmax}(\mathbf{x})$, because the i-th element (dimension) satisfies,

$$\forall i, \text{Softmax}(\mathbf{x} + \mathbf{c})_i = \frac{e^{x_i + c}}{\sum_{k=1}^K e^{x_k + c}} = \frac{e^{x_i} \cdot e^c}{\sum_{k=1}^K e^{x_k} \cdot e^c} = \text{Softmax}(\mathbf{x})_i$$
 (5)

Therefore, suppose $f^{(l+1)}$ is the Softmax layer, $\mathcal{H}^{(l)} = \{c \cdot \mathbf{1}, c \in \mathbb{R}\}.$

Lemma 1.4 (Set of harmless perturbations for Average Pooling layers) Suppose $f^{(l+1)}$ is the Average Pooling layer, $\mathcal{H}^{(l)} = N(A_{\text{avg}})$. A_{avg} is a coefficient matrix determined by the constraints that must be satisfied by the perturbations within each averaging region.

Proof. Suppose $f^{(l+1)}$ is the Average Pooling layer, which is not an injective function. The set of harmless perturbations for the Average Pooling layer $f^{(l+1)}$ is $\mathcal{H}^{(l)} = \{\delta^{(l)}|\operatorname{AvgPool}(z^{(l)}+\delta^{(l)}) = \operatorname{AvgPool}(z^{(l)})\}$.

Notice that the Average Pooling layer computes the average over each $k \times k$ region of features. Thus, within this $k \times k$ region, each element (dimension) of $\delta^{(l)}$ in $\mathcal{H}^{(l)}$ must satisfy the equation:

$$z_1^{(l)} + z_2^{(l)} + \ldots + z_{k \times k}^{(l)} = (z_1^{(l)} + \delta_1^{(l)}) + (z_2^{(l)} + \delta_2^{(l)}) + \ldots + (z_{k \times k}^{(l)} + \delta_{k \times k}^{(l)})$$
 (6)

 \Box

Consequently, the perturbations within each $k \times k$ region must satisfy: $\delta_1^{(l)} + \delta_2^{(l)} + \cdots + \delta_{k \times k}^{(l)} = 0$. Therefore, harmless perturbations within all $K \times K$ region (i.e., shape of feature is $K \times K$) can be represented as $A_{\text{avg}} \delta^{(l)} = \mathbf{0}$, where each row of A_{avg} represents the constraint that perturbations within each $k \times k$ region must satisfy. Thus, $\mathcal{H}^{(l)} = \{\delta^{(l)} | A_{\text{avg}} \delta^{(l)} = \mathbf{0}\} = N(A_{\text{avg}})$.

Lemma 1.5 (Set of harmless perturbations for Max Pooling layers) Suppose $f^{(l+1)}$ is the Max Pooling layer, $\mathcal{H}^{(l)} = \{\forall p, i, \delta_{p,i}^{(l)} \leq c_p - z_{p,i}^{(l)}\} \cap \{\forall p, \prod_{j=1}^{k \times k} (\delta_{p,j}^{(l)} - c_p + z_{p,j}^{(l)}) = 0\}.$ $c_p \coloneqq \max\{z_{p,1}^{(l)}, z_{p,2}^{(l)}, \cdots, z_{p,k \times k}^{(l)}\}$ is the maximum value of features within the $k \times k$ region of the p-th patch. $\mathcal{H}^{(l)}$ is determined by intermediate-layer features $z^{(l)}$ and hence the input sample x.

Proof. Suppose $f^{(l+1)}$ is the Max Pooling layer, which is not an injective function. The set of harmless perturbations for the Max Pooling layer $f^{(l+1)}$ is $\mathcal{H}^{(l)} = \{\delta^{(l)} | \text{MaxPool}(z^{(l)} + \delta^{(l)}) = \text{MaxPool}(z^{(l)})\}$.

Notice that the Max Pooling layer computes the maximum value of the feature $z^{(l)}$ within each $k \times k$ region. Let us divide the feature $z^{(l)}$ into P patches. For the p-th patch of the feature $z^{(l)}(p=\{1,2,\cdots,P\})$, we define the maximum value within the $k \times k$ region as $c_p \coloneqq \max\{z_{p,1}^{(l)}, z_{p,2}^{(l)}, \cdots, z_{p,k \times k}^{(l)}\}$, which depends on the specific feature $z^{(l)}$ and the input sample x. Thus, for the p-th patch with $k \times k$ region, each element (dimension) of $\delta^{(l)}$ in $\mathcal{H}^{(l)}$ must satisfy the equation:

$$\begin{cases} z_{p,1}^{(l)} + \delta_{p,1}^{(l)} \le c_p \\ z_{p,2}^{(l)} + \delta_{p,2}^{(l)} \le c_p \\ \dots \\ z_{p,k \times k}^{(l)} + \delta_{p,k \times k}^{(l)} \le c_p \end{cases}$$
(7)

Note that for the above $k \times k$ inequalities, it needs to be satisfied that at least one of the inequalities takes equality, such that the maximum value of perturbations added to the features within this patch is still c_p , i.e., $\max\{z_{p,1}^{(l)} + \delta_{p,1}^{(l)}, z_{p,2}^{(l)} + \delta_{p,2}^{(l)}, \cdots, z_{p,k\times k}^{(l)} + \delta_{p,k\times k}^{(l)}\} = c_p$. Hence, an additional equality constraint $\prod_{j=1}^{k\times k} [\delta_{p,j}^{(l)} - (c_p - z_{p,j}^{(l)})] = 0$ needs to be satisfied.

Therefore, for the p-th patch with $k \times k$ region, the harmless perturbations satisfy $\{\forall i, \delta_{p,i}^{(l)} \leq c_p - z_{p,i}^{(l)}\} \cap \{\prod_{j=1}^{k \times k} (\delta_{p,j}^{(l)} - c_p + z_{p,j}^{(l)}) = 0\}$. For harmless perturbations within all P patches, $\mathcal{H}^{(l)} = \{\forall p, i, \delta_{p,i}^{(l)} \leq c_p - z_{p,i}^{(l)}\} \cap \{\forall p, \prod_{j=1}^{k \times k} (\delta_{p,j}^{(l)} - c_p + z_{p,j}^{(l)}) = 0\}$.

Lemma 1.6 (Set of harmless perturbations for two-layer linear networks) Given a two-layer linear network $f(x) = A_2 A_1 x$, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$. Here, $\mathcal{P}^{(0)} = N(A_1)$ and $\mathcal{P}^{(1)} = \{\delta | A_1 \delta = \delta^{(1)}, \forall \delta^{(1)} \in N(A_2) \cap C(A_1)\}$.

 $\begin{array}{l} \textit{Proof.} \ \ \text{In general, } \mathcal{H}^{(1)} = \{\delta^{(1)} | A_2(z^{(1)} + \delta^{(1)}) = A_2 z^{(1)}\} = N(A_2) \\ \mathcal{P}^{(1)} = \{\delta | z^{(1)} + \delta^{(1)} = A_1(x + \delta), \forall \delta^{(1)} \in \mathcal{H}^{(1)}\} = \{\delta | A_1 \delta = \delta^{(1)}, \forall \delta^{(1)} \in N(A_2)\} = \{\delta | A_1 \delta = \delta^{(1)}, \forall \delta^{(1)} \in N(A_2) \cap C(A_1)\}, \text{ where } z^{(1)} = A_1 x. \text{ Note that the equation } A_1 \delta = \delta^{(1)} \text{ has a solution (meaning at least one solution) if and only if } \delta^{(1)} \text{ is in the column space of } A_1, \textit{i.e., } \delta^{(1)} \in C(A_1). \end{array}$

$$\mathcal{H}^{(0)} = \{\delta | A_1(x+\delta) = A_1 x\} = N(A_1)$$

 $\mathcal{P}^{(0)} := \mathcal{H}^{(0)}$

Therefore, $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \supseteq \mathcal{P}^{(0)}$.

Specifically, let us consider two common cases in DNNs. Given $A_1 \in \mathbb{R}^{d \times n}$ and $A_2 \in \mathbb{R}^{m \times d}$, where $y = A_2 A_1 x \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$.

(1) n > d and d < m, which means the input dimension is greater than the first layer's feature dimension, and the first layer's feature dimension is smaller than the second layer's feature dimension.

According to Theorem 1, $\mathcal{H}^{(1)} = \{\mathbf{0}\}$ and $\mathcal{H}^{(0)} = N(A_1)$. Then, $\mathcal{P}^{(1)} = \mathcal{H}^{(0)}$ and $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} = \mathcal{P}^{(0)}$.

(2) n < d and d > m, which means the input dimension is smaller than the first layer's feature dimension, and the first layer's feature dimension is greater than the second layer's feature dimension.

According to Theorem 1, $\mathcal{H}^{(1)} = N(A_2)$ and $\mathcal{H}^{(0)} = \{\mathbf{0}\}$. Then, $\mathcal{P}^{(1)} = \{\delta | A_1 \delta = \delta^{(1)}, \forall \delta^{(1)} \in N(A_2) \cap C(A_1)\}$ and $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} = \mathcal{P}^{(1)}$.

Lemma 2 (The least harmful perturbation for a linear layer). If the input dimension of a given linear layer $f^{(l+1)}(z^{(l)}) = Az^{(l)}$ is less than or equal to the output dimension, and the column vectors of A are linearly independent, i.e., $\mathcal{H}^{(l)} = \{\mathbf{0}\}$, then the least harmful perturbation $(\delta^{(l)})^*$ is the eigenvector corresponding to the smallest eigenvalue of the matrix $A^{\top}A$.

Proof. According to Remark 1, if the input dimension of the linear layer is less than or equal to the output dimension, and the column vectors of A are linearly independent, then the dimension of the subspace for harmless perturbations is $dim(\mathcal{H}^{(l)}) = 0$. Thus, $\mathcal{H}^{(l)} = \{\mathbf{0}\}$, which means that there exists no (non-zero) harmless perturbations.

However, in this case, we can solve for the least harmful perturbation such that the output of the given linear layer is minimally affected. Given the matrix A with equivalent effect of a linear layer, the least harmful perturbation is

$$(\delta^{(l)})^* = \operatorname{argmin}_{\delta^{(l)}} ||A\delta^{(l)}||_2, s.t., ||\delta^{(l)}||_2 = 1.$$
(8)

which can be rewritten as a quadratically constrained quadratic program,

$$(\delta^{(l)})^* = \operatorname{argmin}_{\delta^{(l)}} ||A\delta^{(l)}||_2^2, s.t., ||\delta^{(l)}||_2^2 = 1.$$
(9)

We solve above equation by introducing Lagrange function,

$$L(\delta^{(l)}, \gamma) = ||A\delta^{(l)}||_2^2 + \lambda(1 - ||\delta^{(l)}||_2^2)$$

= $(\delta^{(l)})^{\mathsf{T}} A^{\mathsf{T}} A \delta^{(l)} + \lambda(1 - (\delta^{(l)})^{\mathsf{T}} \delta^{(l)})$ (10)

Then the critical points of the Lagrange function can be found by,

$$\frac{\partial L(\delta^{(l)}, \gamma)}{\partial \delta^{(l)}} = 2A^{\top} A \delta^{(l)} - 2\lambda \delta^{(l)} = \mathbf{0}$$
(11)

Then, Equation (11) can be further written as $(A^{\top}A - \lambda I)\delta^{(l)} = \mathbf{0}$. That is, each eigenvector $\delta^{(l)}$ of $A^{\top}A$ with corresponding eigenvalue λ is a critical point. To obtain the smallest $\|A\delta^{(l)}\|_2^2$, the smallest function value $\|A\delta^{(l)}\|_2^2$ at a critical point should be chosen. The solution of Equation (9) is,

$$||A\delta^{(l)}||_2^2 = (\delta^{(l)})^\top A^\top A \delta^{(l)} = (\delta^{(l)})^\top \lambda \delta^{(l)} = \lambda ||\delta^{(l)}||_2^2 = \lambda$$
(12)

Thus, the least harmful perturbation $(\delta^{(l)})^*$ is the eigenvector corresponding to the smallest eigenvalue of the matrix $A^{\top}A$.

Lemma 3. If there exists no harmless perturbations in the (l+1)-th linear layer, i.e., $\mathcal{H}^{(l)} = \{\mathbf{0}\}$, then there exist no two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$ that produce the the same output of the layer, i.e., $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$.

Proof. Let the matrix $A \in \mathbb{R}^{n^{(l+1)} \times n^{(l)}}$ with linearly independent rows/columns have an equivalent effect with the parameters of the linear layer, *i.e.*, $z^{(l+1)} = f^{(l+1)}(z^{(l)}) = Az^{(l)} \in \mathbb{R}^{n^{(l+1)}}$.

$$\text{Let } \mathcal{H}^{(l)} = \{\delta^{(l)} \in \mathbb{R}^{n^{(l)}} | A(z^{(l)} + \delta^{(l)}) = Az^{(l)} \} = \{\delta^{(l)} | A\delta^{(l)} = \mathbf{0} \} = N(A) = \{\mathbf{0}\}.$$

We will prove $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}, f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$ by contradiction. For any two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$, it has $\delta^{(l)} - \hat{\delta}^{(l)} \neq \mathbf{0}$.

Assume that $\exists \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}, f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$. Then, it can be derived that $A\delta^{(l)} = A\hat{\delta}^{(l)}$ and is equivalent to $A(\delta^{(l)} - \hat{\delta}^{(l)}) = \mathbf{0}$, which is in contradiction to $N(A) = \{\mathbf{0}\}$ since $\delta^{(l)} - \hat{\delta}^{(l)} \neq \mathbf{0}$.

Thus, if $\mathcal{H}^{(l)} = \{\mathbf{0}\}$, then there exist no two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$ that produce the same output of the layer, *i.e.*, $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$.

Lemma 4. Given two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)}, \hat{\delta}^{(l)} \notin \mathcal{H}^{(l)}, \mathcal{H}^{(l)} \neq \{\mathbf{0}\}$, if their corresponding orthogonal components have different directions, i.e., $\delta^{(l)}_{\perp} \neq \alpha \cdot \hat{\delta}^{(l)}_{\perp}, \alpha \in \mathbb{R}$, then $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$.

Proof. The linear layer is $z^{(l+1)} = f^{(l+1)}(z^{(l)}) = Az^{(l)} \in \mathbb{R}^{n^{(l+1)}}$.

According to Theorem 4, $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}_{\perp}).$

Similarly, $\forall \hat{\delta}^{(l)} \neq \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\hat{\delta}^{(l)} \notin \mathcal{H}^{(l)}, f^{(l+1)}(\hat{\delta}^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)}).$

We will prove that if $\delta_{\perp}^{(l)} \neq \alpha \cdot \hat{\delta}_{\perp}^{(l)}$, $\alpha \in \mathbb{R}$, then $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$ by contradiction.

Assume that $\exists \delta_{\perp}^{(l)} \neq \alpha \cdot \hat{\delta}_{\perp}^{(l)}$ such that $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$. In this way, if $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$, it has $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$, therefore, $A\delta_{\perp}^{(l)} = A\hat{\delta}^{(l)} = A\hat{\delta}^{(l)} = A\hat{\delta}^{(l)}$. Furthermore, it can be derived that $A(\delta_{\perp}^{(l)} - \hat{\delta}^{(l)}_{\perp}) = \mathbf{0}$ and $\delta_{\perp}^{(l)} - \hat{\delta}^{(l)}_{\perp} \in N(A)$.

However, according to Theorem 4, these two orthogonal components are orthogonal to $\mathcal{H}^{(l)}=N(A)\neq\{\mathbf{0}\}$, respectively, *i.e.*, $\delta_{\perp}^{(l)}\bot N(A)$ and $\hat{\delta}_{\perp}^{(l)}\bot N(A)$. In other words, $\delta_{\perp}^{(l)}\in N^{\perp}(A)$ and $\hat{\delta}_{\perp}^{(l)}\in N^{\perp}(A)$. Linear algebra states that, the row space $C(A^{\top})$ of a matrix A is orthogonal to the nullspace N(A) of the matrix A, *i.e.*, $C(A^{\top})=N^{\perp}(A)$. Then, it has $\delta_{\perp}^{(l)}\in C(A^{\top})$ and $\hat{\delta}_{\perp}^{(l)}\in C(A^{\top})$. Thus, $\delta_{\perp}^{(l)}-\hat{\delta}_{\perp}^{(l)}\in C(A^{\top})$.

Since (1) $\delta_{\perp}^{(l)} - \hat{\delta}_{\perp}^{(l)} \in N(A)$ and (2) $\delta_{\perp}^{(l)} - \hat{\delta}_{\perp}^{(l)} \in C(A^{\top}) = N^{\perp}(A)$, we can derive that $\delta_{\perp}^{(l)} - \hat{\delta}_{\perp}^{(l)} = \mathbf{0}$, which is in contradiction to $\delta_{\perp}^{(l)} \neq \alpha \cdot \hat{\delta}_{\perp}^{(l)}$, $\alpha \in \mathbb{R}$.

Thus, we have proved that given two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$, if their corresponding orthogonal components have different directions, *i.e.*, $\delta^{(l)}_{\perp} \neq \alpha \cdot \hat{\delta}^{(l)}_{\perp}$, then $f^{(l+1)}(\delta^{(l)}) \neq f^{(l+1)}(\hat{\delta}^{(l)})$.

D Proof in Section 4

In the fifth paragraph of Section 4, we state that among all the perturbations leading to identical layer outputs, there exists a unique perturbation characterized by the smallest ℓ_2 norm, which is orthogonal to the harmless perturbation subspace.

Proof. Lemma 4 has proved that given two different perturbations $\forall \delta^{(l)} \neq \hat{\delta}^{(l)} \in \mathbb{R}^{n^{(l)}}$, if they lead to the same layer output, *i.e.*, $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\hat{\delta}^{(l)})$, then they share the same orthogonal component $\delta^{(l)}_{\perp} = \hat{\delta}^{(l)}_{\perp}$.

Next, we will prove that among all the perturbations that lead to the same layer output, the orthogonal perturbation $\delta_{\perp}^{(l)}$ has the smallest ℓ_2 norm.

According to Theorem 4, $\forall \delta^{(l)} \notin \mathcal{H}^{(l)}$, given any perturbation $\forall \delta^{(l)} \in \mathbb{R}^{n^{(l)}}$ and $\delta^{(l)} \notin \mathcal{H}^{(l)}$, it has a unique decomposition $\delta^{(l)} = \delta^{(l)}_{\parallel} + \delta^{(l)}_{\perp}$, where $\delta^{(l)}_{\perp} \notin \mathcal{H}^{(l)}$ denotes the component that is orthogonal to the harmless perturbation subspace $\mathcal{H}^{(l)}$. Since $\delta^{(l)}_{\parallel} \in \mathcal{H}^{(l)}$, then $\delta^{(l)}_{\perp} \perp \delta^{(l)}_{\parallel}$. Thus, it has $\|\delta^{(l)}\|^2 = \|\delta^{(l)}_{\parallel}\|^2 + \|\delta^{(l)}_{\parallel}\|^2$.

Furthermore, Theorem 4 states that $f^{(l+1)}(\delta_{\parallel}^{(l)}) = \mathbf{0}$ and $f^{(l+1)}(\delta^{(l)}) = f^{(l+1)}(\delta_{\perp}^{(l)})$. That is, $\|\delta_{\parallel}^{(l)}\|^2$ does not affect the network output. Thus, $\|\delta^{(l)}\|_{\min}^2 = \|\delta_{\parallel}^{(l)}\|_{\min}^2 + \|\delta_{\perp}^{(l)}\|^2 = 0 + \|\delta_{\perp}^{(l)}\|^2 = \|\delta_{\perp}^{(l)}\|^2$.

Thus, among all the perturbations that lead to the same layer output, the orthogonal perturbation $\delta_{\perp}^{(l)}$ has the smallest ℓ_2 norm.

E Generation of an equivalence matrix A for a convolutional layer

In this section, we represent the convolutional layer as an equivalent matrix A, which is a sparse matrix.

Formally, let the matrix $A \in \mathbb{R}^{(C_{\text{out}}H_{\text{out}}W_{\text{out}})\times(C_{\text{in}}H_{\text{in}}W_{\text{in}})}$ with linearly independent rows/columns have an equivalent effect with the parameters of the convolutional layer $z^{(l+1)} = f^{(l+1)}(z^{(l)})$, i.e., $\hat{z}^{(l+1)} = A\hat{z}^{(l)}$, $\hat{z}^{(l)} \in \mathbb{R}^{C_{\text{in}}H_{\text{in}}W_{\text{in}}}$ is the vectorized feature $z^{(l)}$, and $\hat{z}^{(l+1)} \in \mathbb{R}^{C_{\text{out}}H_{\text{out}}W_{\text{out}}}$ is the vectorized feature $z^{(l+1)}$.

To compute the equivalent matrix A of the convolutional layer, we divide the generation process into three steps. (1) Consider a convolution kernel acting on a local receptive field as one equation. (2) Consider a convolution kernel acting on the whole input as $H_{\text{out}}W_{\text{out}}$ equations. (3) Consider C_{out} convolution kernels acting on the whole input as $C_{\text{out}}H_{\text{out}}W_{\text{out}}$ equations.

(1) Consider a convolution kernel acting on a local receptive field as one equation.

Let $K \in \mathbb{R}^{c \times h \times w}$ denote a convolutional kernel and $k = [k_1, k_2, \cdots, k_{chw}]^{\top} \in \mathbb{R}^{chw}$ denote the vectorized kernel. Let $X \in \mathbb{R}^{c \times h \times w}$ denote an input region/patch covered by a neuron's receptive field, and $x = [x_1, x_2, \cdots, x_{chw}]^{\top} \in \mathbb{R}^{chw}$ denote the vectorized input patch. Then, the scalar output of the convolutional kernel acting on this input patch is $k^{\top}x \in \mathbb{R}$.

To compute a harmless perturbation patch x, it is equivalent to compute an equation $k^{\top}x=0$, which represents the input patch satisfying the equation has no effect on the output, after passing through this convolutional kernel. In this way, the equivalent matrix $A=k^{\top}$.

(2) Consider a convolution kernel acting on the whole input as $H_{\text{out}}W_{\text{out}}$ equations.

Let $K \in \mathbb{R}^{c \times h \times w}$ denote a convolutional kernel and $k = [k_1, k_2, \cdots, k_{chw}]^{\top} \in \mathbb{R}^{chw}$ denote the vectorized kernel. Let $z^{(l)} \in \mathbb{R}^{C_{\text{in}} \times H_{\text{in}} \times W_{\text{in}}}$ denote the whole input feature, and $x \coloneqq \hat{z}^{(l)} = [x_1, x_2, \cdots, x_{C_{\text{in}}H_{\text{in}}W_{\text{in}}}]^{\top} \in \mathbb{R}^{C_{\text{in}}H_{\text{in}}W_{\text{in}}}$ denote the vectorized feature $z^{(l)}$. Here, $C_{\text{in}} = c$.

Given a convolutional layer with the stride S and the zero padding O, the convolutional kernel K acts on the whole input feature $z^{(l)}$ to yield $H_{\text{out}}W_{\text{out}}$ scalar outputs. Here, $H_{\text{out}} = \lfloor \frac{H_{\text{in}} - h + 2O}{S} \rfloor + 1$ and $W_{\text{out}} = \lfloor \frac{W_{\text{in}} - w + 2O}{S} \rfloor + 1$. In other words, there are a total of $H_{\text{out}}W_{\text{out}}$ input patches covered by the receptive fields of a total of $H_{\text{out}}W_{\text{out}}$ neurons.

Let the i-th input patch $P^{(i)} \in \mathbb{R}^{c \times h \times w}$ denote the region covered by the receptive field of the i-th neuron, which has a total of chw units. Within the input region, each unit of $P^{(i)}$ is an unknown variable $x_j, j \in \{1, 2, \cdots, C_{\text{in}} H_{\text{in}} W_{\text{in}}\}$ of the input feature x, or 0 if the unit is a zero padding unit in the receptive field. Let $p^{(i)} \in \mathbb{R}^{chw}$ denote the vectorized input patch $P^{(i)}$.

To compute the harmless perturbation x, let the outputs of the $H_{\text{out}}W_{\text{out}}$ neurons all be zero. That is, the scalar output of each neuron needs to satisfy $k^\top p^{(i)} = 0, \forall i \in \{1, 2, \cdots, H_{\text{out}}W_{\text{out}}\}$. Thus, there are a total of $H_{\text{out}}W_{\text{out}}$ equations with $C_{\text{in}}H_{\text{in}}W_{\text{in}}$ unknown variables. Rewriting the $H_{\text{out}}W_{\text{out}}$

equations in matrix form yields an equivalent matrix $A \in \mathbb{R}^{(H_{\text{out}}W_{\text{out}}) \times (C_{\text{in}}H_{\text{in}}W_{\text{in}})}$ that satisfies $Ax = \mathbf{0}$. Here, the equivalent matrix A is a sparse matrix, since $chw \ll C_{\text{in}}H_{\text{in}}W_{\text{in}}$ is usually satisfied.

It is worth noting that when the kernel size of the convolutional layer is greater than or equal to the stride, the convolutional kernel acts on all $C_{\rm in}H_{\rm in}W_{\rm in}$ input variables, which means that each unknown variable is constrained by at least one equation. In this case, the matrix $A \in \mathbb{R}^{(H_{\rm out}W_{\rm out})\times(C_{\rm in}H_{\rm in}W_{\rm in})}$ has linearly independent rows/columns. When the kernel size of the convolutional layer is smaller than the stride, the convolutional kernel only acts on some of input variables, which means that there are some unknown variables that are not constrained by any of the equations (variables not constrained by equations can take any value). In this case, the column vectors of the matrix $A \in \mathbb{R}^{(H_{\rm out}W_{\rm out})\times(C_{\rm in}H_{\rm in}W_{\rm in})}$ are linearly dependent.

(3) Consider C_{out} convolution kernels acting on the whole input as $C_{\text{out}}H_{\text{out}}W_{\text{out}}$ equations.

A convolutional kernel acting on the whole input yields $H_{\text{out}}W_{\text{out}}$ equations $(H_{\text{out}}W_{\text{out}})$ linearly independent row vectors in the matrix $A \in \mathbb{R}^{(H_{\text{out}}W_{\text{out}}) \times (C_{\text{in}}H_{\text{in}}W_{\text{in}})}$.

Given a convolutional layer with linearly independent vectorized kernels, a total of C_{out} convolutional kernels acting on the whole input can yield $C_{\text{out}}H_{\text{out}}W_{\text{out}}$ equations $(C_{\text{out}}H_{\text{out}}W_{\text{out}})$ independent row vectors in the matrix $A \in \mathbb{R}^{(C_{\text{out}}H_{\text{out}}W_{\text{out}}) \times (C_{\text{in}}H_{\text{in}}W_{\text{in}})}$.

F Experimental details

F.1 Experiment details for verifying the dimension of harmless perturbation subspace in Figure 2

To verify the dimension of harmless perturbation subspace in Section 3.3, we fixed the input dimension and increased the output dimension of the first linear layer of each DNN trained on different dataset. To increase the output dimension of the linear layer, we increased the number of convolutional kernels $C_{\rm out}$ for the convolutional layer, and the number of neurons $N_{\rm out}$ for the fully-connected layer, respectively.

For convolutional layers, if the input dimension is greater than the output dimension, then the dimension of the harmless perturbation subspace for a convolutional layer is $dim(\mathcal{H}^{(l)}) = C_{\rm in}H_{\rm in}W_{\rm in} - C_{\rm out}H_{\rm out}W_{\rm out}$. We modified the feature size of the output $H_{\rm out}$ and $W_{\rm out}$ of the first convolutional layer by setting different strides S and kernel sizes K (K > S). Here, $H_{\rm out} = \lfloor \frac{H_{\rm in} - K + 2P}{S} \rfloor + 1$ and $W_{\rm out} = \lfloor \frac{W_{\rm in} - K + 2P}{S} \rfloor + 1$.

Specifically, (1) when the stride S=1, the kernel size K=3, and the zero padding P=1, it has $H_{\rm out}=H_{\rm in}$ and $W_{\rm out}=W_{\rm in}$. In this case, for the VGG-16 (stride S=1) on the CIFAR-10 dataset, when $C_{\rm out}=C_{\rm in}=3$, $dim(\mathcal{H}^{(l)})=0$.

- (2) Similarly, when S=2, K=3, and P=1, it has $H_{\rm out}=0.5H_{\rm in}$ and $W_{\rm out}=0.5W_{\rm in}$. In this case, for the ResNet-18/50 and EfficientNet (stride S=2), when $C_{\rm out}=4C_{\rm in}=12, dim(\mathcal{H}^{(l)})=0$.
- (3) When S=4, K=5, and P=1, it has $H_{\rm out}=0.25H_{\rm in}$ and $W_{\rm out}=0.25W_{\rm in}$. In this case, for the ResNet-18 (stride S=4), when $C_{\rm out}=16C_{\rm in}=48, dim(\mathcal{H}^{(l)})=0$.

For fully-connected layers, if the input dimension is greater than the output dimension, then the dimension of the harmless perturbation subspace for a fully-connected layer is $dim(\mathcal{H}^{(l)}) = N_{\rm in} - N_{\rm out}$. We increased the number of neurons $N_{\rm out}$ of the first fully-connected layer. In this case, for the MLP-5 on the MNIST dataset, when $N_{\rm out} = N_{\rm in} = 28 \times 28 = 784$, $dim(\mathcal{H}^{(l)}) = 0$. For the MLP-5 on the CIFAR-10/100 and SVHN datasets, when $N_{\rm out} = N_{\rm in} = 3 \times 32 \times 32 = 3072$, $dim(\mathcal{H}^{(l)}) = 0$.

F.2 Experiment details for evaluating the network performance in Figure 3

To evaluate the impact of harmless perturbations on network performance in Section 3.3, we trained the CIFAR-10 dataset on different networks and tested the effects of harmless perturbations on the network performance across varying perturbation magnitudes.

To ensure the existence of harmless perturbation subspace, it is necessary to guarantee that the input dimension of the convolutional layer exceeds the output dimension. Specifically, when computing

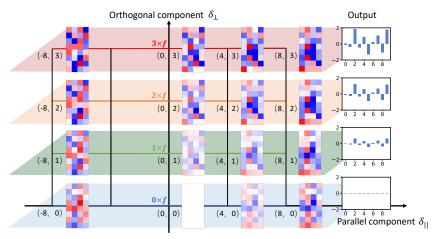


Figure 6: Contour map of the actual impact of any perturbations on the output of the last linear layer. Perturbations in the same row share identical orthogonal components, leading to identical network outputs. Perturbations in different rows exhibit proportional orthogonal components, leading to proportional network outputs.

harmless perturbations on images, we set the number of convolutional kernels to 10 in the first convolutional layer of the DNNs. When computing harmless perturbations on intermediate-layer features, we selected the first linear layer whose input dimension exceeded the output dimension.

When computing harmless perturbations on images, we replaced the original convolutional layer in the ResNet18/50, which had 64 kernels with the kernel size 7, the stride 2, and the zero padding 3, with two convolutional layers. Specifically, the first convolutional layer had 10 kernels with the kernel size 7, the stride 2, and the zero padding 3. The second convolutional layer had 64 kernels with the kernel size 3, the stride 1, and the zero padding 1. Thus, the dimension of the harmless perturbation subspace of the first convolutional layer after the replacement was $dim(\mathcal{H}^{(l)}) = C_{\rm in}H_{\rm in}W_{\rm in} - C_{\rm out}H_{\rm out}W_{\rm out} = 3\times32\times32-10\times16\times16=512$.

When computing harmless perturbations on intermediate-layer features, we selected the first linear layer whose input dimension exceeds the output dimension. We removed the skip connections of selected convolutional layers. Specifically, for the ResNet-18, we computed the harmless perturbation subspace of the first convolutional layer of the 0-th block of the second layer. The dimension of the harmless perturbation subspace of the chosen convolutional layer was $dim(\mathcal{H}^{(l)}) = C_{\rm in}H_{\rm in}W_{\rm in} - C_{\rm out}H_{\rm out}W_{\rm out} = 64\times8\times8-128\times4\times4=2048$. For the ResNet-50, we computed the harmless perturbation subspace of the first convolutional layer of the second block of the first layer. The dimension of the harmless perturbation subspace of the chosen convolutional layer was $dim(\mathcal{H}^{(l)}) = C_{\rm in}H_{\rm in}W_{\rm in} - C_{\rm out}H_{\rm out}W_{\rm out} = 256\times8\times8-64\times8\times8=12288$.

Since there were infinite harmless perturbations in the harmless perturbation subspace, we chose a harmless perturbation direction to verify the network performance. Without loss of generality, we employed parallel components of adversarial perturbations (see Theorem 4) as the chosen directions of harmless perturbations. According to Theorem 4, the parallel component of an arbitrary perturbation is a harmless perturbation.

To achieve this, we first generated adversarial perturbations $\delta^{\rm adv}$ by PGD-20 for PGD with 20 steps, the maximum perturbation was set to $\epsilon=8/255$ and the step size was set to 1/255 [Madry et al., 2018]. Then, we produced the corresponding parallel components of the adversarial perturbations $\delta^{\rm adv}_{\parallel}=P\delta^{\rm adv}$ according to Theorem 4. Finally, we scaled the parallel components $\delta^{\rm adv}_{\parallel}$ to obtain new perturbations, i.e., $\hat{\delta}^{\rm adv}_{\parallel}=\frac{\epsilon}{\|\delta^{\rm adv}_{\parallel}\|_{\infty}}\cdot\delta^{\rm adv}_{\parallel}$, such that the generated perturbations statisfied $\|\hat{\delta}^{\rm adv}_{\parallel}\|_{\infty}=8/255$. In Figure 3(a), we increased the magnitude of the generated perturbations by $\alpha\cdot\hat{\delta}^{\rm adv}_{\parallel}$, where $\alpha=\{2^0,2^1,\cdots,2^8\}$.

F.3 Experiment details for decomposing arbitrary perturbations in Figure 6

To verify the decomposition of arbitrary perturbations in Theorem 4 and Theorem 5, we plotted the contour map of the output of the last linear layer. According to Theorem 5, given a linear layer with a harmless perturbation subspace, the contour map of the layer output can be plotted along the direction of the chosen orthogonal component. We conducted experiments on MLP-5 with 32 neurons in the penultimate layer on the CIFAR-10 dataset. Figure 6 illustrates the contour map of the actual impact of perturbations on the network output, in which perturbations were generated through linear combinations of orthogonal and parallel components of an arbitrary perturbation.

F.4 Experiment details for privacy protection

To achieve the application of the harmless perturbation space for privacy protection in Section 5.1, we have to obtain the harmless perturbation subspace in the first linear layer.

To ensure the existence of harmless perturbation subspace, we modified the number of convolution kernels to 10 in the first convolutional layer in the ResNet-50. To achieve this, we replaced the original convolutional layer in the ResNet-50, which had 64 kernels with the kernel size 7, the stride 2, and the zero padding 3, with two convolutional layers. Specifically, the first convolutional layer had 10 kernels with the kernel size 7, the stride 2, and the zero padding 3. The second convolutional layer had 64 kernels with the kernel size 3, the stride 1, and the zero padding 1. Thus, the dimension of the harmless perturbation subspace of the first convolutional layer after the replacement was $dim(\mathcal{H}^{(l)}) = C_{\text{in}}H_{\text{in}}W_{\text{in}} - C_{\text{out}}H_{\text{out}}W_{\text{out}} = 3\times32\times32 - 10\times16\times16 = 512.$