8. Proofs of our Results

8.1. Proof of Theorem 1

Proof. **Part I.** Suppose the true labeling function h^* is not 1-robust at an $\mathbf{x} \in \mathcal{X}$ with $y = h^\star(\mathbf{x})$. Then, applying Definition 3 with $\epsilon = 1$, there exists a perturbation $\boldsymbol{\delta} \in \mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1) \cap (\mathcal{X} - \{\mathbf{x}\})$ such that $\mathbf{x} + \boldsymbol{\delta} \in \cup_{c \neq y} \mathcal{X}_c$. In other words, there exists an unsafe direction $\mathbf{u} \in \mathcal{U}^*(\mathbf{x})$ and a step size $t > 0$ such that $\boldsymbol{\delta} = t\mathbf{u}$. By the definition of $g^\star$, we see that $t > g^\star(\mathbf{x}, \mathbf{u})$ and hence,

$$d^*_{\mathrm{PD}}(\mathbf{x}, \boldsymbol{\delta}) = d^*_{\mathrm{PD}}(\mathbf{x}, t\mathbf{u}) \geq \frac{1}{g^*(\mathbf{x}, \mathbf{u})} \max\left(\langle t\mathbf{u}, \mathbf{u}\rangle, 0\right) = \frac{t}{g^*(\mathbf{x}, \mathbf{u})} > 1. \tag{2}$$

This is a contradiction since we assumed that $\boldsymbol{\delta} \in \mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1)$, implying $d^*_{\mathrm{PD}}(\mathbf{x}, \boldsymbol{\delta}) \leq 1$. Hence h^* must be 1-robust w.r.t. d_{PD}.

Part II. Note if h is 0-robust at any input $\mathbf{x}$ then h misclassifies $\mathbf{x}$. Let $\mathbf{x} \in \mathcal{X}_y$ be such that h is at most ϵ-robust for some $\epsilon < 1$. Hence there exists a perturbation $\boldsymbol{\delta} \in \mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1) \cap (\mathcal{X} - \{\mathbf{x}\})$ such that $h(\mathbf{x} + \boldsymbol{\delta}) \neq h^\star(\mathbf{x})$. For such a perturbation, there are two cases.

Case 1 : $h(\mathbf{x} + \boldsymbol{\delta}) \neq h^*(\mathbf{x} + \boldsymbol{\delta})$.

In this case, clearly $\mathbf{x} + \boldsymbol{\delta}$ is an adversarial example for h and the conclusion follows.

Case 2 : $h^*(\mathbf{x} + \boldsymbol{\delta}) = h(\mathbf{x} + \boldsymbol{\delta}) = c$ for some $c \neq y$.

Since $\mathbf{x} + \boldsymbol{\delta} \in \mathcal{X}_c$. There necessarily exists an unsafe direction $u \in \mathcal{U}^*(\mathbf{x})$ such that $\boldsymbol{\delta} = t\mathbf{u}$. Further, by definition of the normalization function $t > g^*(\mathbf{x}, \mathbf{u})$. Thus,

$$d^*_{\mathrm{PD}}(\mathbf{x}, \boldsymbol{\delta}) \geq \frac{1}{g^*(\mathbf{x}, \mathbf{u})} \max\left(\langle t\mathbf{u}, \mathbf{u}\rangle, 0\right) = \frac{t}{g^*(\mathbf{x}, \mathbf{u})} > 1.$$

This is a contradiction since $\boldsymbol{\delta} \in \mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1) \cap (\mathcal{X} - \{\mathbf{x}\})$. Hence this case is not possible.

To summarize if there exists an input $\mathbf{x} \in \mathcal{X}_y$ such that h is at most ϵ-robust for some $0 \leq \epsilon < 1$, then every perturbation $\boldsymbol{\delta} \in \mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1) \cap (\mathcal{X} - \{\mathbf{x}\})$ such that $h(\mathbf{x} + \boldsymbol{\delta}) \neq y$ is necessarily such that $\mathbf{x} + \boldsymbol{\delta}$ is misclassified by h, *i.e.* $h(\mathbf{x} + \boldsymbol{\delta}) \neq h^\star(\mathbf{x} + \boldsymbol{\delta})$. Hence the conclusion follows. □

9. Expanded Discussion

9.1. Choosing k-subset $\mathtt{S}_{c,k}$

To obtain a representative subset of unsafe directions $\mathcal{U}(\mathbf{x})$ we find a k-subset $\mathtt{S}_{c,k}$ of $\mathtt{S}_c$ by solving a clustering-type optimization problem,

$$\mathtt{S}_{c,k} := \underset{|A|=k,\ A \subseteq \mathtt{S}_c}{\arg\max} f(A), \tag{3}$$

where

$$f(A) := \min_{\mathbf{x} \in \mathtt{S}_c} \max_{\mathbf{a} \in A} \frac{\langle \mathbf{x}, \mathbf{a}\rangle}{\|\mathbf{x}\|_2 \|\mathbf{a}\|_2}.$$

The objective in Equation (3) defines a discrete k-center problem for which we can find an approximate minimizer using the classical greedy 2-approximation algorithm [21] which starts from a randomly selected element of $\mathtt{S}_c$, and greedily expands the selection k times, each time adding the best element in $\mathtt{S}_c$ according to Equation (3). Algorithm 1 has a computational complexity of $\mathcal{O}(k^2|\mathtt{S}_c|)$. The quality of this greedy approximation depends on the choice of the initial element. Exploring other strategies to select subsets $\mathtt{S}_{c,k}$ remains an interesting future direction.

9.2. Projection onto sub-level sets

Each sublevel set is the intersection of halfspaces,

$$\mathcal{S}(\mathbf{x}, d_{\mathrm{PD}}, \varepsilon) = \bigcap_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \{\boldsymbol{\delta} \in \mathbb{R}^d \mid \langle \boldsymbol{\delta}, \mathbf{u}\rangle \leq \varepsilon \cdot g(\mathbf{x}, \mathbf{u})\}$$

Algorithm 1 Choosing k-subset of $\mathtt{S}_c$: Greedy k-center approximation

Sample uniformly at random $a \sim \mathtt{S}_c$
Initialize : $A \leftarrow \{a\}$
repeat
Find $\mathbf{b} \in \mathtt{S}_c$ with minimal cosine similarity to any $\mathbf{a} \in A$,

$$\mathbf{b} := \arg\min_{\mathbf{x} \in \mathtt{S}_c} \max_{\mathbf{a} \in A} \frac{\langle \mathbf{x}, \mathbf{a} \rangle}{\|\mathbf{x}\|_2 \|\mathbf{a}\|_2}$$

$A \leftarrow A \cup \{b\}$.
until $|A| = k$.
Return A

We let $\mathcal{C}_{\mathbf{x},\mathbf{u}} := \{\boldsymbol{\delta} \in \mathbb{R}^d \mid \langle \boldsymbol{\delta}, \mathbf{u} \rangle \leq \varepsilon \cdot g(\mathbf{x}, \mathbf{u})\}$ denote the half space defined by the unsafe direction $\mathbf{u}$ at input $\mathbf{x}$. For each individual halfspace we denote $P_{\mathcal{C}_{\mathbf{x},\mathbf{u}}}$ the projection operator defined as,

$$P_{\mathcal{C}_{\mathbf{x},\mathbf{u}}}(\boldsymbol{\delta}) := \boldsymbol{\delta} - \max\left(\langle \boldsymbol{\delta}, \mathbf{u} \rangle - \varepsilon g(\mathbf{x}, \mathbf{u}), 0\right) \mathbf{u}.$$

To obtain a projection operator for the sublevel sets one can employ a greedy procedure (see Algorithm 2) that alternates between projection onto the halfspaces $\mathcal{C}_{\mathbf{x},\mathbf{u}}$.

Algorithm 2 Greedy Projection

Require: Nonempty closed convex sets C_i for $1 \leq i \leq T$.
Require: Input $\mathbf{a} \in \mathbb{R}^d$.
Require: Iteration hyper-parameter $N \geq 1$.
Ensure: Projection onto intersection of convex sets $\cap_i C_i$.
repeat N **times**
Select farthest convex set C_j and project to C_j,

$$C_j \leftarrow \arg\max_{C_i} \|\mathbf{a} - P_{C_i}(\mathbf{a})\|_2 \tag{4}$$

$$\mathbf{a} \leftarrow P_{C_j}(\mathbf{a}). \tag{5}$$

end
Return $\mathbf{a}$.

In practice, we leverage the linearity of the PD-threat by utilizing the lazy projection algorithm $\boldsymbol{\delta} \to \frac{\varepsilon}{d_{\mathrm{PD}}(\mathbf{x},\boldsymbol{\delta})}\boldsymbol{\delta}$ in time-and compute-constrained settings.

9.3. Quality of Approximation

The quality of the approximation depends on the choice of representative unsafe directions $\mathcal{U}_k(\cdot)$ via the k-subset $\mathtt{S}_{c,k}$ and on the heuristic choice of the approximate normalization g_β, via the scaling hyper-parameter β. For approximating the normalization, we note that for $\beta = 1$, $d_{\mathrm{PD,k},1}(\mathbf{x}, \boldsymbol{\delta}) \leq d^*_{\mathrm{PD,k},\beta}(\mathbf{x}, \boldsymbol{\delta})$ for all inputs $\mathbf{x}$ and perturbations $\boldsymbol{\delta}$. However, the permissible set $\mathcal{S}(\mathbf{x}, d_{\mathrm{PD,k},\beta}, \varepsilon)$ for threshold $\varepsilon = 1$ is likely to include unsafe perturbations. To see this, let $\mathbf{u} \in \mathcal{U}_k(\mathbf{x})$ and let $\tilde{\mathbf{x}} \in \mathtt{S}_{c,k}$ be the corresponding point such that $\mathbf{u} := \frac{\tilde{\mathbf{x}}-\mathbf{x}}{\|\tilde{\mathbf{x}}-\mathbf{x}\|_2}$. The perturbation $\boldsymbol{\delta} := \tilde{\mathbf{x}} - \mathbf{x}$ has threat $d_{\mathrm{PD,k},1}(\mathbf{x}, \boldsymbol{\delta}) = 1$. Yet, for some $t \leq 1$, the scaled perturbation $t\boldsymbol{\delta}$ has threat less than 1 but $\mathbf{x} + t\boldsymbol{\delta}$ still has true label c and hence $t\boldsymbol{\delta}$ is an unsafe perturbation. Hence, in practice, we make a heuristic choice of $\beta = \frac{1}{2}$ to compute the approximate normalization.

Next we note that a larger k trades-off computational efficiency of evaluating $d_{\mathrm{PD,k},\beta}$ to how well it approximates d^*_{PD}. Given training data $\mathtt{S}$, we first recommend finding the minimum k such that PD threat $d_{\mathrm{PD,k},\beta}$ rates unsafe corruptions on training data as a sufficiently large threat,

$$k_{\min} := \min_{k \in [1, \frac{|S|}{C}]} \text{ s.t. } \min_{(\mathbf{x},y),\ (\tilde{\mathbf{x}},c) \in \mathtt{S}} d_{\mathrm{PD,k},\beta}(\mathbf{x}, \tilde{\mathbf{x}} - \mathbf{x}) > 1$$

We let $k_{\max} \in [k_{\min}, \frac{|S|}{C}]$ be the maximum k subject to a practitioner's memory constraints and desired throughput on evaluation of threat function. We then recommend a equi-spaced grid search over the interval $[k_{\min}, k_{\max}]$ to determine an appropriate heuristic choice. In our experiments on Imagenet-1k, we observed that $k_{\min} = 20$ and $k_{\max} = 50$ with our computing resources. In this range, a grid search showed that $k = 50$ is sufficiently fast, and capable of discriminating safe and unsafe perturbations on validation data (more on this in Section 4).

9.4. Additional Technical characteristics of PD

Scope of Design. We emphasize that the PD threat is designed to disentangle safe and unsafe perturbations. By construction, the unsafe directions $\mathcal{U}(\mathbf{x})$ only contain perturbations that alter the class label. Thus, the PD threat is not expected to differentiate two safe perturbations. The threat of two different safe perturbations that retain the class label need not be ordered by the perceptual similarity which requires a fine-grained inference on image features at multiple levels of resolution. As such the PD threat is not a suitable replacement for neural perceptual distance metrics, and human-evaluation studies such as two-alternative forced choice testing (2AFC) are out of scope for the proposed design. In a similar vein, unsafe directions only correspond to observations within the data domain. Along a safe direction, the threat is not normalized to the boundaries of the data domain. Hence the PD threat (even exact threat d^*_{PD}) is not designed to identify out-of-distribution (OOD) data.

Attribution. In Figure 6, the perturbation $\boldsymbol{\delta}$ is assessed to have large threat at input $\mathbf{x}$ with label y. Let $\bar{u} := \underset{\mathbf{u} \in \mathcal{U}(\mathbf{x})}{\arg\max} \frac{1}{g_\beta(\mathbf{x},\mathbf{u})} \text{ReLU}\left(\langle \boldsymbol{\delta}, \mathbf{u} \rangle\right)$. be the unsafe direction most aligned with the perturbation $\boldsymbol{\delta}$. Since $\bar{u} \in \mathcal{U}(\mathbf{x})$, there exists a point $\bar{\mathbf{x}} \in \mathtt{S}_{c,k}$ where $c \neq y$ such that $\bar{u} = \frac{\bar{\mathbf{x}}-\mathbf{x}}{\|\bar{\mathbf{x}}-\mathbf{x}\|_2}$. Hence for each perturbation $\boldsymbol{\delta}$ and input $\mathbf{x}$, the PD-threat identifies a point $\bar{x}$ that is most aligned with the direction of perturbation. This feature enables a direct attribution of threat to observed training data points.

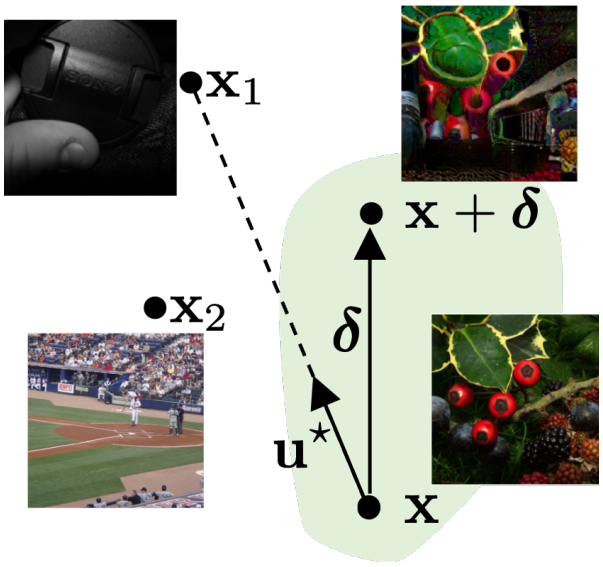
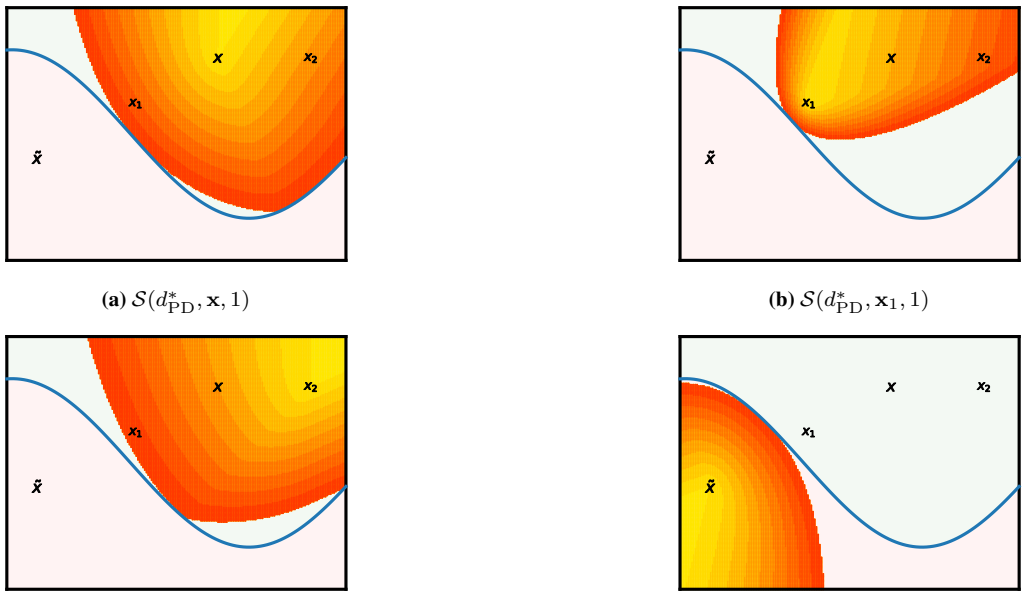


Figure 6. Adversarial attack on benchmark model from [33]

Bottom-up Perspective. We first note that the threat function at each input $\mathbf{x}$, can be re-arranged as a particular large-width single layer feed-forward neural network $h_{\text{PD},\mathbf{x}}$. Here the neural networks $h_{\text{PD},\mathbf{x}}$ have parameters $\{\mathtt{S}_{c,k}\}_{c \in [C]}$. On the Imagenet dataset with $C = 1000$ labels, and $k = 50$, this amounts to a neural network with ≈ 7.5 billion parameters. However, unlike the pretraining required to compute vision-language embeddings, the PD threat describes a randomized neural network based on observed data that does not require an iterative gradient based learning of parameters. The only computation needed is the selection of representative subset $\mathtt{S}_{c,k}$ for each label c. Thus the PD-threat can be viewed as a bottom-up definition of neural embedding that requires no training and is motivated instead as a heuristic empirical approximation of a principled exact non-parametric threat function d^*_{PD}. The value of this approach is shown in promising experimental evidence in Section 4.

Illustration of exact threat d^*_{PD} on synthetic 2D data Consider a binary classification task where inputs $\mathbf{x} \in \mathbb{R}^2$ from a bounded domain $\mathcal{X}$ are assigned labels $y \in \mathcal{Y} := \{-1, +1\}$ by a true labeling function h^*. In Figure 7, the solid black rectangular regions indicate the bounded domain $\mathcal{X}$, and the solid blue lines are the decision boundary of the true labeling

function h^*. A point within the domain is labeled 1 if it is above the blue line and -1 otherwise. Figure 7 presents four inputs, three points $(\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2)$ with label 1 and $\tilde{\mathbf{x}}$ with label -1. The points are chosen such that $\mathbf{x}_1$ and $\mathbf{x}_2$ are equidistant from $\mathbf{x}$, i.e., $\|\mathbf{x}_1 - \mathbf{x}\|_2 = \|\mathbf{x}_2 - \mathbf{x}\|_2 = \varepsilon$. In this toy example, the true labeling function h^* and the corresponding partition sets $\mathcal{X}_1$ and $\mathcal{X}_{-1}$ are known, and thus the exact PD-threat d^*_{PD} can be computed[13]. At $\mathbf{x}$, clearly $\mathbf{u}_1 \coloneqq \frac{\mathbf{x}_1 - \mathbf{x}}{\|\mathbf{x}_1 - \mathbf{x}\|_2}$ is an unsafe direction while $\mathbf{u}_2 \coloneqq \frac{\mathbf{x}_2 - \mathbf{x}}{\|\mathbf{x}_2 - \mathbf{x}\|_2}$ is a safe direction and hence $d^*_{\mathrm{PD}}(\mathbf{x}, \varepsilon\mathbf{u}_1) \geq d^*_{\mathrm{PD}}(\mathbf{x}, \varepsilon\mathbf{u}_2)$. Figure 7 presents a visualization of the 1-sublevel sets at each marked point. Figure 7 demonstrates the anisotropy and locality of the sublevel sets $\mathcal{S}(d^\star_{\mathrm{PD}}, \cdot, 1)$. A large value of threat $d^\star_{\mathrm{PD}}(\mathbf{x}, \boldsymbol{\delta})$ indicates the proximity of the input $\mathbf{x}$ to the boundary of the partition sets. This behaviour is intuitively captured in Figure 3 where x_1 is closer to the boundary indicated by the blue line than the other marked points.



(a) $\mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}, 1)$ **(b)** $\mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}_1, 1)$

(c) $\mathcal{S}(d^*_{\mathrm{PD}}, \mathbf{x}_2, 1)$ **(d)** $\mathcal{S}(d^*_{\mathrm{PD}}, \tilde{\mathbf{x}}, 1)$

Figure 7. Shape of 1-sublevel sets at different inputs

[13] Unsafe directions $\mathcal{U}^\star(\mathbf{x})$ and normalization $g^\star(\mathbf{x}, \mathbf{u})$ are computed via a 2D discretization grid over the domain $\mathcal{X}$

9.5. Weighted Threat Specification

Definition 9 requires a relative distance between class labels $W : [C] \times [C] \to [0, 1]$. We note that $W(y, c)$ is the weight assigned to threats where y is the class label of the original input under perturbation and c is the class label associated with the unsafe direction. In this section we identify 3 distinct approaches to computing such a relative distance between class labels. For each approach, the relative weights are computed by scaling w.r.t minimum and maximum for any fixed class y and varying class c' of the unsafe directions. This additional normalization ensures comparability of weights across different approaches. The final weights $W(y, c)$ used to define the weighted threat specification combines all 3 approaches.

Definition 10 (Euclidean Relative Weights)**.** We define the class distance based on Euclidean norm as the average ℓ_2 distance between the selected subsets[14] of training data $\{\mathtt{S}_{c,k}\}_{c=1}^{C}$,

$$\mathrm{L2}(y, c) := \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathrm{Unif}(\mathtt{S}_{y,k}),\\ \tilde{\mathbf{x}}\sim\mathrm{Unif}(\mathtt{S}_{c,k})}} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2 ,$$

where $\mathrm{Unif}(\cdot)$ denotes sampling uniformly at random. The relative class distance $W_{\text{Euclidean}} : [C] \times [C] \to [0, 1]$ is the defined as,

$$W_{\text{Euclidean}}(y, c) := \frac{\mathrm{L2}(y, c) - \min_{c_1} \mathrm{L2}(y, c_1)}{\max_{c_2} \mathrm{L2}(y, c_2) - \min_{c_1} \mathrm{L2}(y, c_1)}.$$

As explained in the motivation ℓ_p norms are insufficient to measure perceptual similarity between images of distinct class labels. Hence we propose to account for perceptual similarity using DreamSim.

Definition 11 (DreamSim Relative Weights)**.** We define the class distance based on DreamSim as,

$$\mathrm{DS}(y, c) := \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathrm{Unif}(\mathtt{S}_{y,k}),\\ \tilde{\mathbf{x}}\sim\mathrm{Unif}(\mathtt{S}_{c,k})}} \mathrm{DreamSim}(\mathbf{x}, \tilde{\mathbf{x}}).$$

The relative class distance $W_{\mathrm{DS}} : [C] \times [C] \to [0, 1]$ is the defined as,

$$W_{\text{DreamSim}}(y, c) := \frac{\mathrm{DS}(y, c) - \min_{c_1} \mathrm{DS}(y, c_1)}{\max_{c_2} \mathrm{DS}(y, c_2) - \min_{c_1} \mathrm{DS}(y, c_1)}$$

DreamSim is finetuned to match human perceptual similarity judgements however it is unclear if DreamSim can explicitly account for the concept hierarchy of Imagenet-1k class labels provided by WordNet. For e.g. images of class labels HEN and Ostrich can be deemed perceptually distinct but are closer semantically as they both correspond to bird categories.

Next, we discuss a relative distance that explicitly accounts for semantic similarity based on class hierarchy. For Imagenet-1k class labels $[C]$, the associated WordNet hierarchy can be represented as the tree $\mathrm{WordNet}(V, E)$ where $[C] \subset V$ and V is the set of WordNet classes and the edge set E contains an edge (v_1, v_2) if v_1 is a sub-class of v_2 or vice-versa. For any pair of classes (v_1, v_2), the lowest common ancestor $\mathrm{LCA} : V \times V \to V$ function outputs the lowest (i.e. deepest) class $\mathrm{LCA}(v_1, v_2)$ that has both v_1 and v_2 as descendants. Let v_{root} denote the root class (ENTITY for Imagenet-1k) such that all classes in $V \backslash \{v_{\text{root}}\}$ are descendants of v_{root}. Let d_v denote the length of the minimal path from root v_{root} to class v.

Definition 12 (WordNet Relative Weights)**.** For any two pairs of classes (v_1, v_2), we define the class distance based on the WordNet hierarchy as the length of the minimal path connecting the two classes v_1 and v_2 (through $\mathrm{LCA}(v_1, v_2)$),

$$\mathrm{dist}_{\mathrm{LCA}}(v_1, v_2) := d_{v_1} + d_{v_2} - 2d_{\mathrm{LCA}(v_1,v_2)}$$

The relative class distance based on the WordNet class hierarchy, $W_{\text{WordNet}} : [C] \times [C] \to [0, 1]$ is defined as,

$$W_{\text{WordNet}}(y, c) := \frac{\mathrm{dist}_{\mathrm{LCA}}(y, c) - \min_{c_1} \mathrm{dist}_{\mathrm{LCA}}(y, c_1)}{\max_{c_2} \mathrm{dist}_{\mathrm{LCA}}(y, c_2) - \min_{c_1} \mathrm{dist}_{\mathrm{LCA}}(y, c_1)}$$

Definition 13 (Threat Specification Relative Weights)**.** The relative weights $W : [C] \times [C] \to [0, 1]$ used to define the weighted threat specification are explicitly defined as,

$$W(y, c) := \Big(\min \{W_{\text{Euclidean}}(y, c), W_{\text{DreamSim}}(y, c), W_{\text{WordNet}}(y, c)\} \Big)^2.$$

[14] Recall, $\mathtt{S}_{c,k}$ is the representative subset chosen to formulate the threat specification.

A smaller value of $W(y, c) \approx 0$ indicates that the class labels y and c are *nearby* by at least one of the 3 relative distances based on Euclidean norm, DreamSim or WordNet Hierarchy. A smaller value of $W(y, c)$ indicates that perturbations $\boldsymbol{\delta}$ on inputs $\mathbf{x}$ with label y that are aligned with unsafe directions $\mathbf{u}$ of label c have a larger threat $d_{\text{PD}-\text{W}}(\mathbf{x}, \boldsymbol{\delta})$, thus the threat of perturbations between nearby classes is amplified.

10. Illustrative Examples

Following Mintun et al. [36]'s protocol, each corruption style is computed in-memory to avoid additional noise incurred from compression quality of images saved to disk. Figure 8 and Figure 9 show the corruptions of an original image $\mathbf{x}$ with label LIONFISH. Table 4 and Table 5 show the amount of threat assesed for each corruption by 6 threat models - (1) the ℓ_p threat models - d_∞ and d_2, (2) the proposed PD threat models - d_{PD}, $d_{\text{PD}-\text{W}}$ and $d_{\text{PD}-\text{S}}$ and finally (3) the Dreamsim threat model d_{DS}. In Table 4 and Table 5 the threats across different threat models are not comparable due to different scalings (for e.g. d_∞, d_{DS} range between $[0, 1]$) but the others are not limited to an interval. Note, the threats d_2 and $d_{\text{PD}-\text{W}}$ are scaled by a constant factor for readability.

The corruptions are sourced from Imagenet-C [25] and Imagenet-$\bar{C}$ [36]. The Imagenet-C corruptions are at severity 5 (maximum 5). The Imagenet-$\bar{C}$ corruptions includes the full list of 30 corruptions at severity 5 (maximum 10). We note that the larger experiments on comparison of average threat (for eg. Figure 3) only include the subset of 10 Imagenet-$\bar{C}$ corruptions that are considered semantically distinct from Imagenet-C [36]. For any fixed threat model, the threats for different corruption styles vary. Selective entries are colored *red* if the threat assessed for the corruption style is as large as the threat of the unsafe perturbation. Entries are colored *orange* if the threat of the corruption style is at least half of the threat of the perturbation. The colors are meant only for illustrative purposes to highlight the ability of each threat model to separate safe and unsafe perturbations. Evidently most of the safe corruptions are rated as high threat by d_∞. We note that both d_2 and d_{DS} rate noise corruptions as high threat but $d_{\text{PD}}, d_{\text{PD}-\text{W}}$ and $d_{\text{PD}-\text{S}}$ do not. Weather corruptions are uniformly rated as high threat. Implementation of PD-threat and other necessary files can be found at our github repo.

CATEGORY	STYLE	d_∞	d_2	d_{PD}	$d_{\text{PD}-\text{W}}$	$d_{\text{PD}-\text{S}}$	d_{DS}
Unsafe	Unsafe	0.90	0.40	3.30	1.79	2.51	0.64
Noise	Gaussian Noise	0.88	0.28	0.51	0.28	0.49	0.36
	Shot Noise	0.89	0.30	0.61	0.30	0.46	0.35
	Impulse Noise	0.90	0.27	0.57	0.31	0.52	0.35
	Speckle Noise	0.91	0.24	0.46	0.24	0.35	0.32
Blur	Defocus Blur	0.64	0.08	0.40	0.15	1.28	0.23
	Glass Blur	0.59	0.07	0.38	0.15	1.24	0.16
	Motion Blur	0.66	0.09	0.43	0.19	1.29	0.18
	Zoom Blur	0.51	0.08	0.37	0.18	0.96	0.14
	Gaussian Blur	0.63	0.07	0.42	0.17	1.29	0.26
Weather	Snow	0.82	0.38	2.53	0.79	2.19	0.53
	Frost	0.58	0.34	2.31	0.68	2.40	0.44
	Fog	0.61	0.22	1.87	0.99	1.76	0.49
Compression	Pixelate	0.54	0.06	0.22	0.09	0.76	0.11
	JPEG	0.51	0.06	0.12	0.05	0.39	0.10
Digital	Brightness	0.47	0.26	1.41	0.48	2.07	0.20
	Contrast	0.60	0.16	1.54	0.74	2.13	0.48
	Elastic Transform	0.69	0.16	0.25	0.11	0.89	0.01
	Saturate	0.87	0.18	0.71	0.27	1.15	0.24
Occlusion	Spatter	0.81	0.16	0.63	0.31	0.49	0.25

Table 4. Threats evaluated on Imagenet-C corruptions in Figure 8

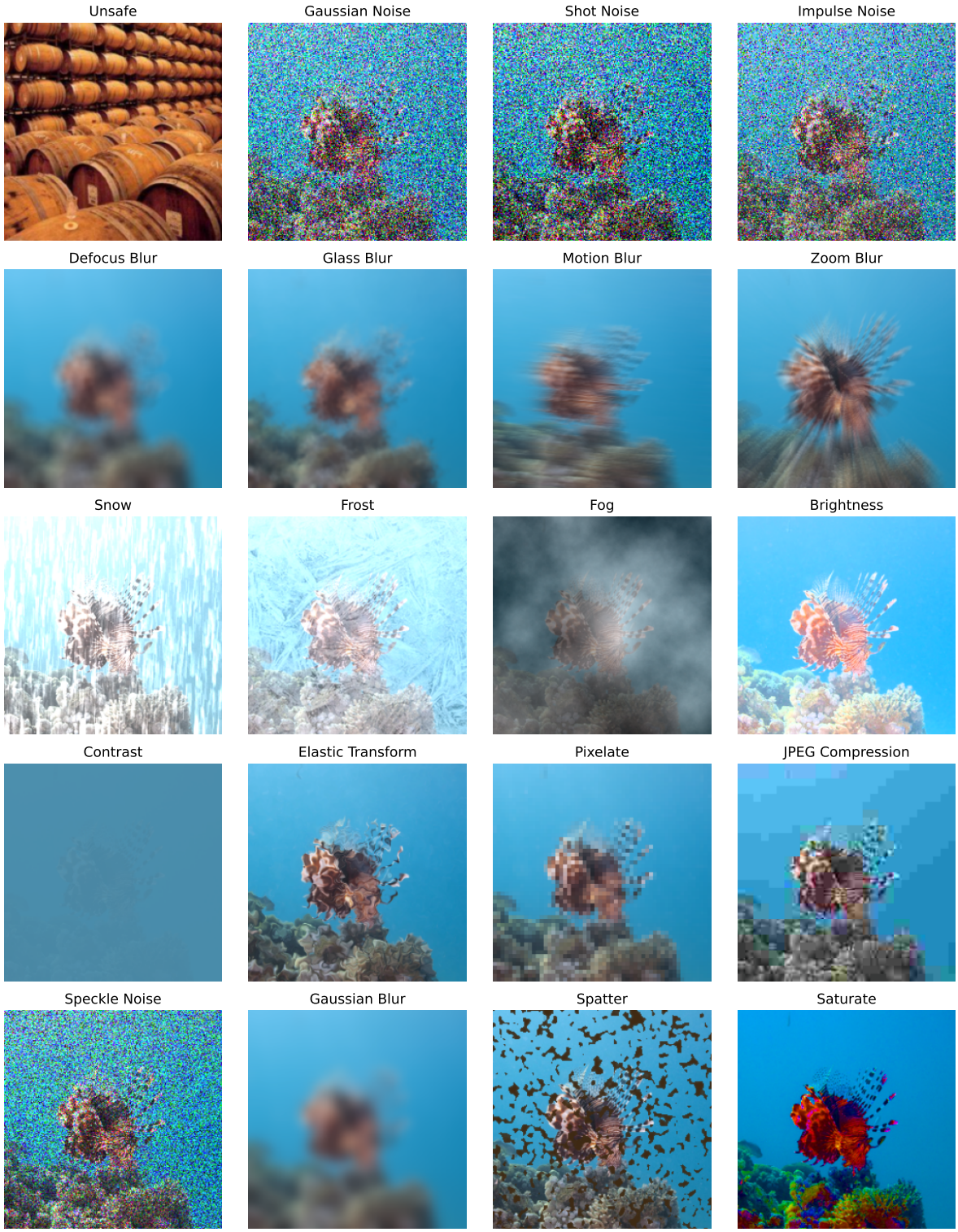


Figure 8. Imagenet-C corruptions on a sample image of class LIONFISH.

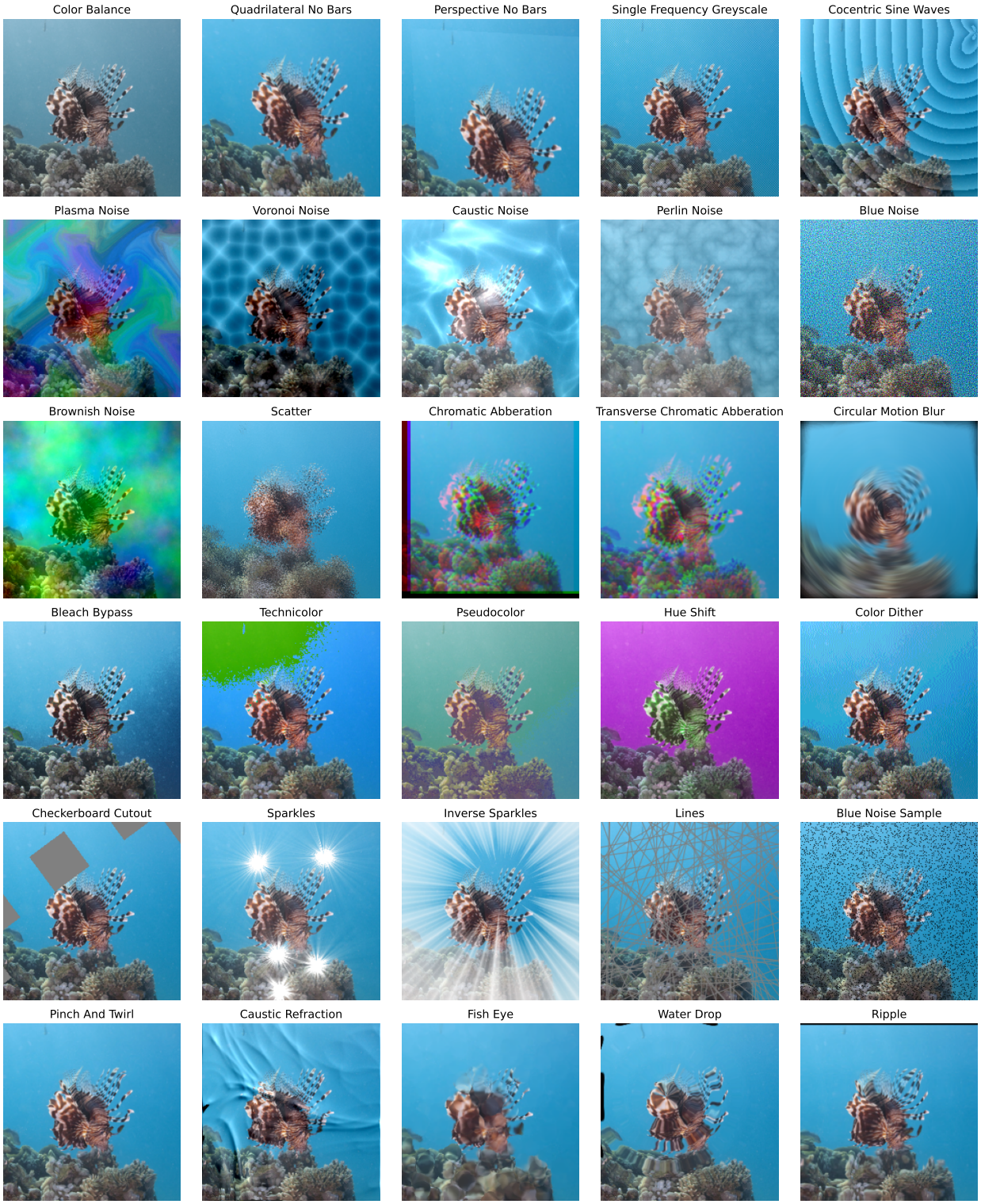


Figure 9. Imagenet-$\bar{C}$ corruptions on a sample image of class LIONFISH.

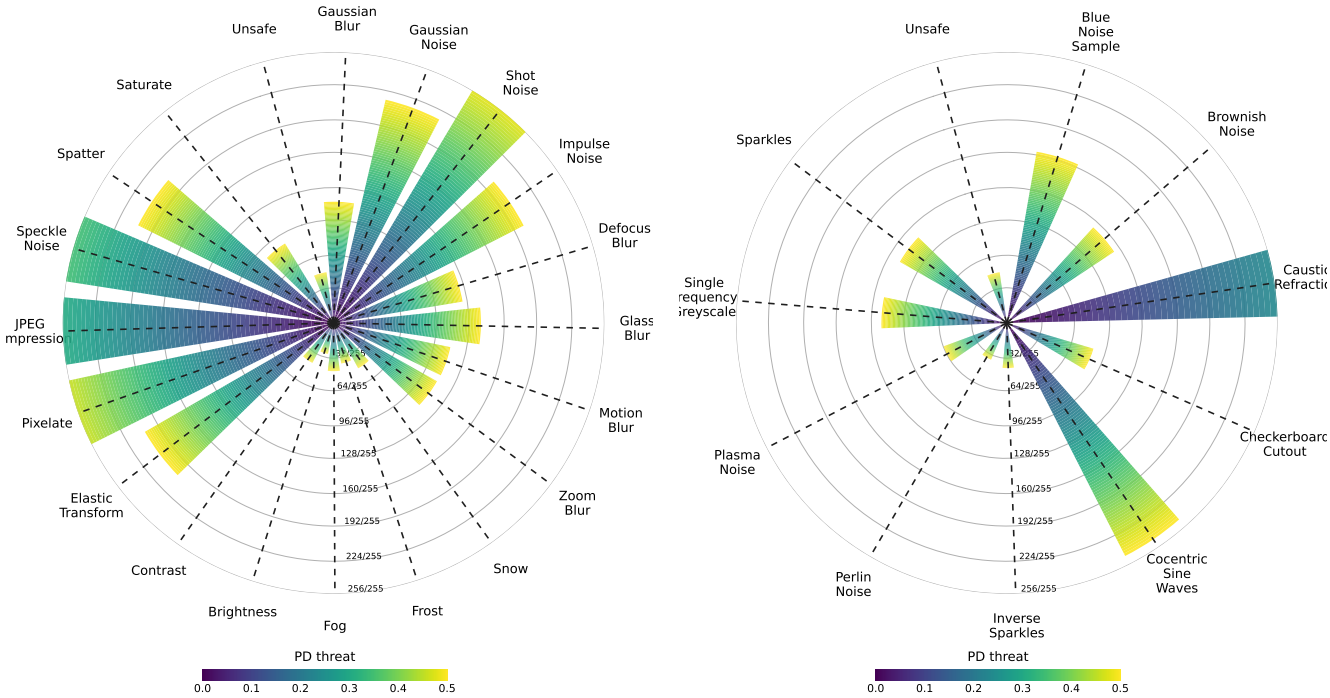
10.1. Anisotropy of Threat on Imagenet-1k

We illustrate a key feature of the PD threat - anisotropy. At each input $\mathbf{x}$, along any direction $\mathbf{u}$, we can compare two threat functions d_1 and d_2 by measuring the largest d_1 threat for perturbations within $\mathcal{S}(d_2, \mathbf{x}, \varepsilon)$. Such a measurement is feasible

CATEGORY	STYLE	d_∞	d_2	d_{PD}	$d_{\text{PD}-\text{W}}$	$d_{\text{PD}-\text{S}}$	d_{DS}
Unsafe	Unsafe	0.90	0.40	3.30	1.79	2.51	0.64
Noise	Plasma Noise	0.29	0.13	0.62	0.25	0.72	0.17
	Voronoi Noise	0.30	0.18	1.04	0.37	0.91	0.18
	Caustic Noise	0.59	0.14	0.71	0.20	1.31	0.09
	Perlin Noise	0.28	0.11	0.88	0.37	0.95	0.31
	Blue Noise	0.87	0.22	0.26	0.15	0.24	0.25
	Brownish Noise	0.48	0.16	0.56	0.22	0.91	0.29
	Blue Noise Sample	0.90	0.16	0.44	0.19	0.30	0.29
Blur	Cocentric Sine Waves	0.12	0.07	0.03	0.01	0.07	0.16
	Scatter	0.70	0.10	0.34	0.14	1.13	0.12
	Chromatic Abberations	0.89	0.17	0.76	0.31	1.48	0.22
	Transverse Chromatic Abberation	0.70	0.10	0.38	0.18	0.94	0.20
	Circular Motion Blur	0.81	0.08	0.35	0.15	0.94	0.19
	Pinch and Twirl	0.73	0.06	0.15	0.06	0.51	0.00
	Caustic Refraction	0.88	0.09	0.18	0.07	0.48	0.09
	Fish Eye	0.68	0.08	0.25	0.10	0.88	0.05
	Water Drop	0.89	0.10	0.28	0.13	0.97	0.03
	Ripple	0.89	0.12	0.40	0.18	1.08	0.07
Digital	Color Balance	0.18	0.09	0.74	0.45	0.20	0.29
	Quadrilateral No Bars	0.77	0.13	0.57	0.26	1.62	0.02
	Perspective No Bars	0.75	0.16	0.80	0.30	2.08	0.03
	Bleach Bypass	0.25	0.08	0.33	0.13	0.94	0.07
	Technicolor	0.90	0.21	0.75	0.45	0.94	0.24
	Pseudocolor	0.21	0.11	0.93	0.53	0.94	0.41
	Hue Shift	0.55	0.26	1.53	0.92	0.94	0.36
	Color Dither	0.17	0.06	0.02	0.00	0.00	0.04
Occlusion	Single Frequency GreyScale	0.18	0.12	0.04	0.02	0.03	0.14
	Checkerboard Cutout	0.42	0.07	0.24	0.14	0.08	0.05
	Sparkles	0.77	0.14	0.77	0.30	0.59	0.13
	Inverse Sparkles	0.72	0.26	1.75	0.61	0.50	0.33
	Lines	0.43	0.10	0.65	0.35	0.67	0.20

Table 5. Threats evaluated on Imagenet-$\bar{C}$ corruptions in Figure 9

for PD threat and ℓ_p threat due to linear growth, since, for any perturbation, evaluating the threat at $d(\mathbf{x}, \boldsymbol{\delta})$, immediately provides the threat at scaled perturbations since $d(\mathbf{x}, t\boldsymbol{\delta}) = td(\mathbf{x}, \boldsymbol{\delta})$ for each $d \in \{d_\infty, d_2, d_{\text{PD}}\}$. Figure 10 is a radial bar plot of the corruptions $\omega \in \Omega_5$ (with severity level 5) where the radial axis is d_∞ threat. The heights of each radial bar is $\frac{1}{2 \cdot \text{avg}(d_{\text{PD}}, \omega)} \cdot \text{avg}(d_\infty, \omega)$. A larger height indicates a corruption ω where the growth of PD threat per unit ℓ_∞ threat is lower (on average). Figure 10 indicates that if corruptions $\omega(\mathbf{x}) - \mathbf{x}$ are scaled to a fixed d_∞ threat, the resulting PD threat varies across directions reflecting the anisotropy of the PD threat model. We emphasize such a plot is not possible for the DreamSim threat model since the growth is non-linear and hence threat at each scaled perturbation $t\boldsymbol{\delta}$ needs to be evaluated separately.



(a) Imagenet-C **(b)** Imagenet-$\bar{C}$

Figure 10. Anisotropy of PD-Threat Model

11. Comparison of Average Threat Statistics

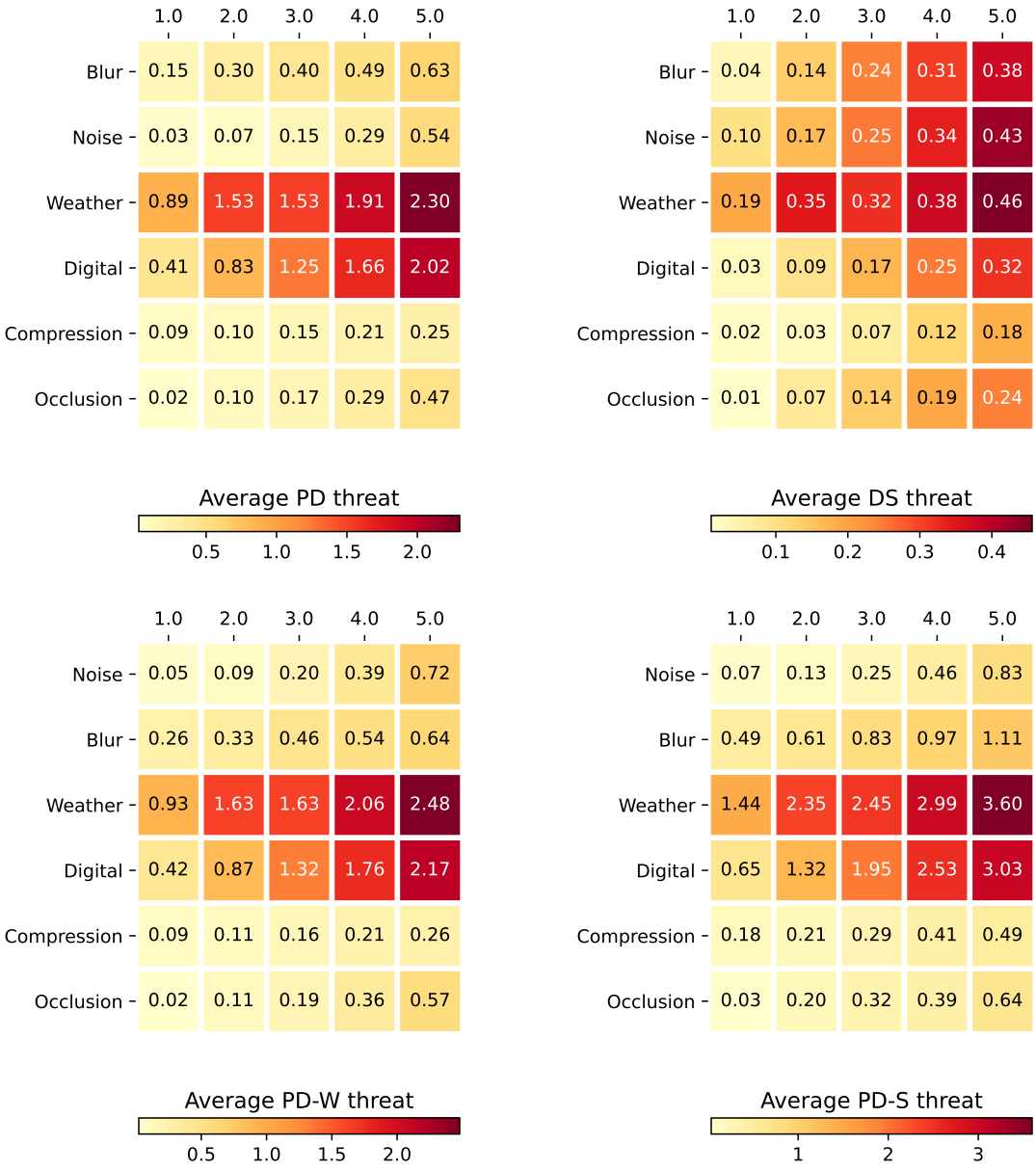
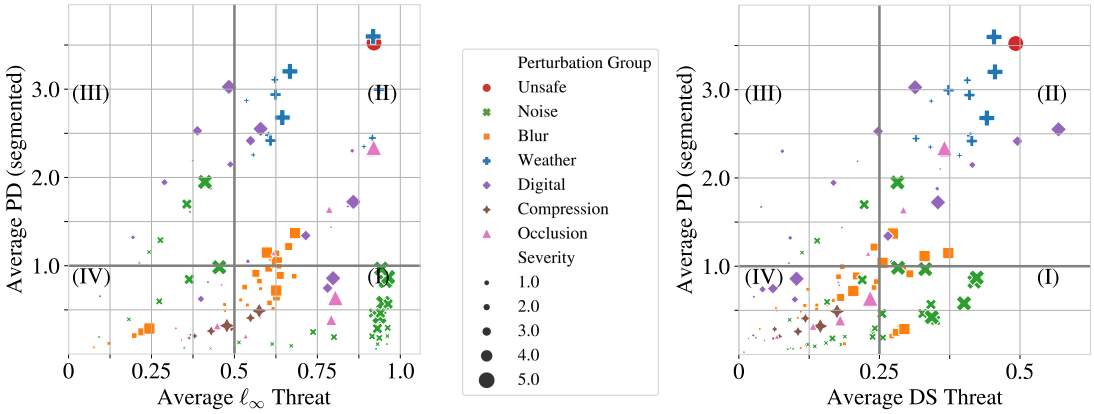


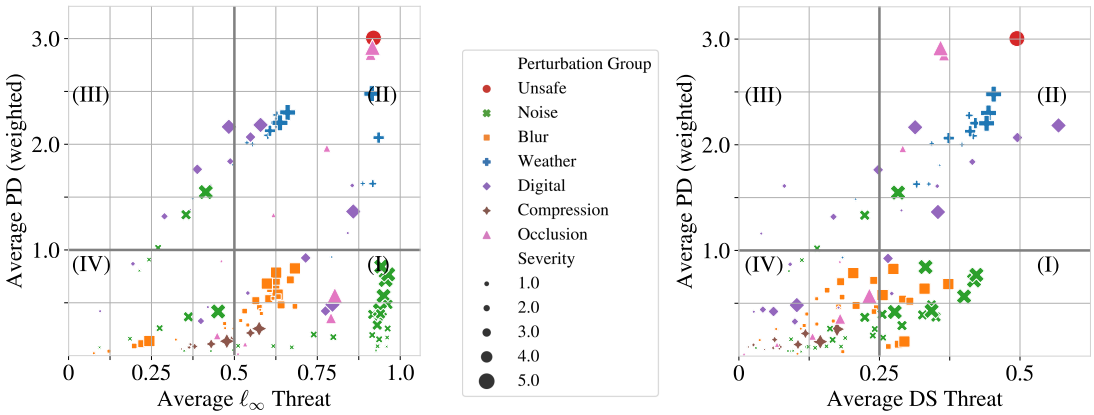
Figure 11. Heatmap of threat models vs corruption severity



(a) PD-S vs ℓ_∞

(b) PD-S vs DS

Figure 12. Comparison of PD-S threat



(a) PD-W vs ℓ_∞

(b) PD-W vs DS

Figure 13. Comparison of PD-W threat