Supplementary Material for "Which Models have Percptually Aligned Gradients? An Explanation via Off-Manifold Robustness"

A Broader Impact

This work studies the impact of robust training objectives on the perceptual alignment of gradients, and does not propose any new tools or methods. As such, this work is foundational in nature and does not have any direct societal impact.

B Additional Proofs

Theorem 3 (Equivalence between off-manifold robustness and on-manifold alignment)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ exhibits on-manifold gradient alignment if and only if it is off-manifold robust wrt normal noise $\mathbf{u} \sim \mathcal{N}(0, \sigma^2)$ for $\sigma \to 0$ (with $\rho_1 = \rho_2$).*

Proof. We proceed by observing that we can decompose the input-gradient into on-manifold and off-manifold components by projecting onto the tangent space and its orthogonal component respectively, i.e., $\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbb{P}_x \nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbb{P}_x^{\perp} \nabla_{\mathbf{x}} f(\mathbf{x})$.

We also observe that we can write the gradient norm in terms of an expected dot product, i.e., $\frac{1}{\sigma^2} \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2)} (\nabla_{\mathbf{x}} f(\mathbf{x})^\top \mathbf{u})^2 = \frac{1}{\sigma^2} \nabla_{\mathbf{x}} f(\mathbf{x})^\top \mathbb{E}(\mathbf{u}\mathbf{u}^\top) \nabla_{\mathbf{x}} f(\mathbf{x}) = \|\nabla_{\mathbf{x}} f(\mathbf{x})\|^2$.

Using these facts we can compute the norm of the off-manifold component as follows,

$$\underbrace{\frac{\|\nabla_{\mathbf{x}} f(\mathbf{x}) - \mathbb{P}_x \nabla_{\mathbf{x}} f(\mathbf{x})\|^2}{\|\nabla_{\mathbf{x}} f(\mathbf{x})\|^2}}_{\text{On-manifold gradient alignment}} = \frac{\|\mathbb{P}_x^{\perp} \nabla_{\mathbf{x}} f(\mathbf{x})\|^2}{\|\nabla_{\mathbf{x}} f(\mathbf{x})\|^2}$$

$$= \frac{\frac{1}{\sigma^2} \mathbb{E}_{\mathbf{u}_{\text{off}} \sim \mathcal{N}(0, \sigma^2 \Sigma)} (\nabla_{\mathbf{x}} f(\mathbf{x})^\top \mathbf{u}_{\text{off}})^2}{\frac{1}{\sigma^2} \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2)} (\nabla_{\mathbf{x}} f(\mathbf{x})^\top \mathbf{u})^2} \;\; ; \;\; \Sigma = \text{Cov}(\mathbf{u}_{\text{off}}) = \mathbb{P}_x^{\perp} (\mathbb{P}_x^{\perp})^\top$$

$$= \lim_{\sigma \to 0} \underbrace{\frac{\mathbb{E}_{\mathbf{u}_{\text{off}} \sim \mathcal{N}(0, \sigma^2 \Sigma)} (f(\mathbf{x} + \mathbf{u}_{\text{off}}) - f(\mathbf{x}))^2}{\mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2)} (f(\mathbf{x} + \mathbf{u}) - f(\mathbf{x}))^2}}_{\text{Off-manifold robustness}}$$

The second line is obtained by using the fact above regarding re-writing the gradient norm in terms of the expected dot product, and the final line is obtained by using a first order Taylor expansion, which is exact in the limit of small sigma. From the equality of first and last terms, we have that the on-manifold gradient alignment $\Leftrightarrow$ the off-manifold robustness. □

Theorem 4. *The input-gradients of Bayes optimal classifiers lie on the signal manifold $\Leftrightarrow$ Bayes optimal classifiers are relative off-manifold robust.*

Proof. From definition 3, it is clear that given a classification problem, there exists a single distractor distribution $d(\mathbf{x})$. Now, we take gradients of log probabilities of the Bayes optimal classifiers, which results in:

$$\nabla_{\mathbf{x}} \log p(y = i \mid \mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x} \mid y = i) - \sum_j p(y = j \mid \mathbf{x}) \nabla_{\mathbf{x}} \log p(\mathbf{x} \mid y = j)$$

We notice first that the vectors $\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid y)$ all lie tangent to the data manifold by definition, as this data generating process $p(\mathbf{x} \mid y)$ itself defines the data manifold. As $\nabla_{\mathbf{x}} \log p(y \mid \mathbf{x})$ is a *linear combination* of the class-conditional generative model gradients, it follows that the input-gradient of the Bayes optimal model also lie tangent to the data manifold. Now, like any vector on the tangent space at $\mathbf{x}$, it can be decomposed into signal and distractor components. Computing the distractor, we find that

$$\nabla_{\mathbf{x}} \log p(y \mid \mathbf{x}) \odot (1 - \mathbf{m}^*(\mathbf{x})) = d(\mathbf{x}) - \sum_j p(y = j \mid \mathbf{x}) d(\mathbf{x}) = 0$$

This happens because the distractor is independent of the label, thus the distractor component is zero, and the input-gradient of the Bayes optimal model lies entirely on the signal manifold. From Theorem 3, it follows that when a model gradients lie on a manifold, it is also off-manifold robust wrt that manifold.

□

C Experimental Details

C.1 Robust Training Objectives

We consider the following robust training objectives, where $l(x, y)$ denotes the cross-entropy loss function.

1. Gradient norm regularization: $l(f(x), y) + \lambda \|\nabla_{\mathbf{x}} f(\mathbf{x})\|_2^2$ with a regularization constant λ.
2. A smoothness penalty: $l(f(x), y) + \lambda \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2)} \|f(x + \epsilon) - f(x)\|_2^2$ with a fixed noise level σ^2 and a varying regularization constant λ.
3. Randomized Smoothing: $\mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2)} l(f(x + \epsilon), y)$ with a noise level σ^2.
4. Adversarial Robust Training: $l(f(\tilde{x}), y)$ where $\tilde{x} = \arg\max_{\tilde{x} \in B_\epsilon(x)} l(f(\tilde{x}), y)$ and $\tilde{x}$ was obtained from the ϵ-ball around x using projected gradient descent.

C.2 Training Details

On CIFAR-10, we trained Resnet18 models for 200 epochs with an initial learning rate of 0.025. When training with gradient norm regularization or the smoothness penalty and large regularization constants we reduced the learning rate proportional to the increase in the regularization constant. After 150 and 175 epochs, we decayed the learning rate by a factor of 10.

On ImageNet-64x64, we trained Resnet18 models for 90 epochs with a batch size of 4096 and an initial learning rate of 0.1 that was decayed after 30 and 60 epochs, respectively. We used the same parameters for projected gradient descent (PGD) as in [29], that is we took 3 steps with a step size of $2\epsilon/3$.

On the MNIST dataset with a distractor, we trained a Resnet18 model for 9 epochs with an initial learning rate of 0.1 that was decayed after 3 and 6 epochs, respectively. We also trained an l_2-adversarially robust Resenet18 with projected gradient descent (PGD). We randomly chose the perturbation budget $\epsilon \in \{1, 4, 8\}$ and took 10 steps with a step size of $\alpha = 2.5\epsilon/10$.

C.3 Diffusion Models

On CIFAR-10, we use the unconditional diffusion model `edm-cifar10-32x32-uncond-vp`. On ImageNet-64x64, we use the conditional diffusion model `edm-imagenet-64x64-cond-adm`. Both models are available at `https://github.com/NVlabs/edm`.

C.4 Model Gradients

With the unconditional diffusion model, we sum the input gradients across all classes. With the conditional diffusion model, we consider the input gradient with respect to the predicted class. We consider input gradients before the softmax [20].

C.5 CIFAR-10 Autoencoder

We use `https://github.com/clementchadebec/benchmark_VAE` to train an autoeoncoder on CIFAR-10 with a latent dimension $k = 128$. We use a default architecture and training schedule. We

Figure 6: **Left:** Images from CIFAR10. **Middle:** Random perturbations on the data manifold. **Right:** Random perturbations off the data manifold.

then use the autoencoder to estimate, at each data point, a 128-dimensional tangent space. Figure 6 depicts random directions within the estimated tangent spaces.

C.6 Pre-Trained Robust Models on ImageNet

On ImageNet, we use the pre-trained robust Resnet18 models form `https://github.com/microsoft/robust-models-transfer`. To load these models, we use the robustness library `https://github.com/MadryLab/robustness`.

C.7 Estimating the Score on ImageNet

We estimate the score on ImageNet using the diffusion model for ImageNet-64x64. To estimate the score, we simply down-scale an image to 64x64.

C.8 MNIST with a Distractor

The MNIST data set with a distractor is inspired by [11]. The data set consists of gray-scale images of size 56x28. Every image contains a single MNIST digit and the distractor. We choose the fixed letter "A" as the distractor. On every image, we randomly place the distractor on top or below the MNIST digit. In order to estimate the relative noise robustness, we separately add different levels of noise to the signal or distractor. Figure 12 depicts images and models gradients on this data set.

C.9 The LPIPS metric

The LPIPS metric measures the perceptual similarity between two different images. The metric itself corresponds to a loss, meaning that lower values correspond to more similar images [31]. The figures in the main paper depict 1-LPIPS, that is higher values correspond to more similar images.

C.10 Code Availability

Code that allows to replicate all the results in this paper is part of the Supplementary material.

C.11 Resources Used

All computations were done on an internal cluster using Nvidia 2080 Ti GPUs. In total, this project required 6 GPU months.

D Additional Plots

The figures below depict the model gradients of different types of models, ranging from weakly robust to excessively robust. The figures depict the relationship between model gradients and the score qualitatively. This complements the quantitative results in the main paper.

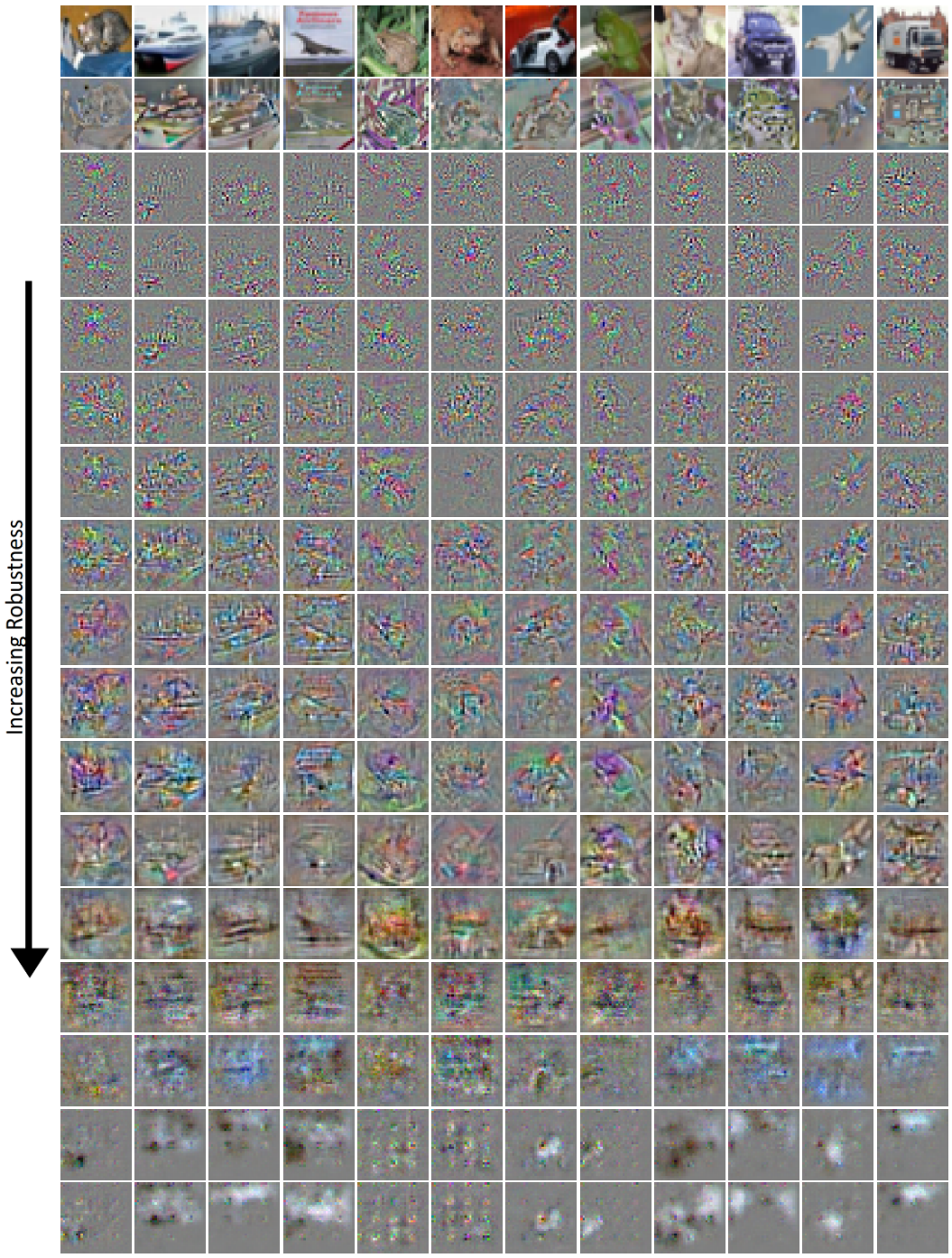


Figure 7: The input gradients of different models trained with **gradient norm regularization on CIFAR-10**. The top rows depict the image, the score, and the input gradients of unrobust models. The middle rows depict the perceptually aligned input gradients of robust models. The bottom rows depict the input gradients of excessively robust models. Best viewed in digital format.

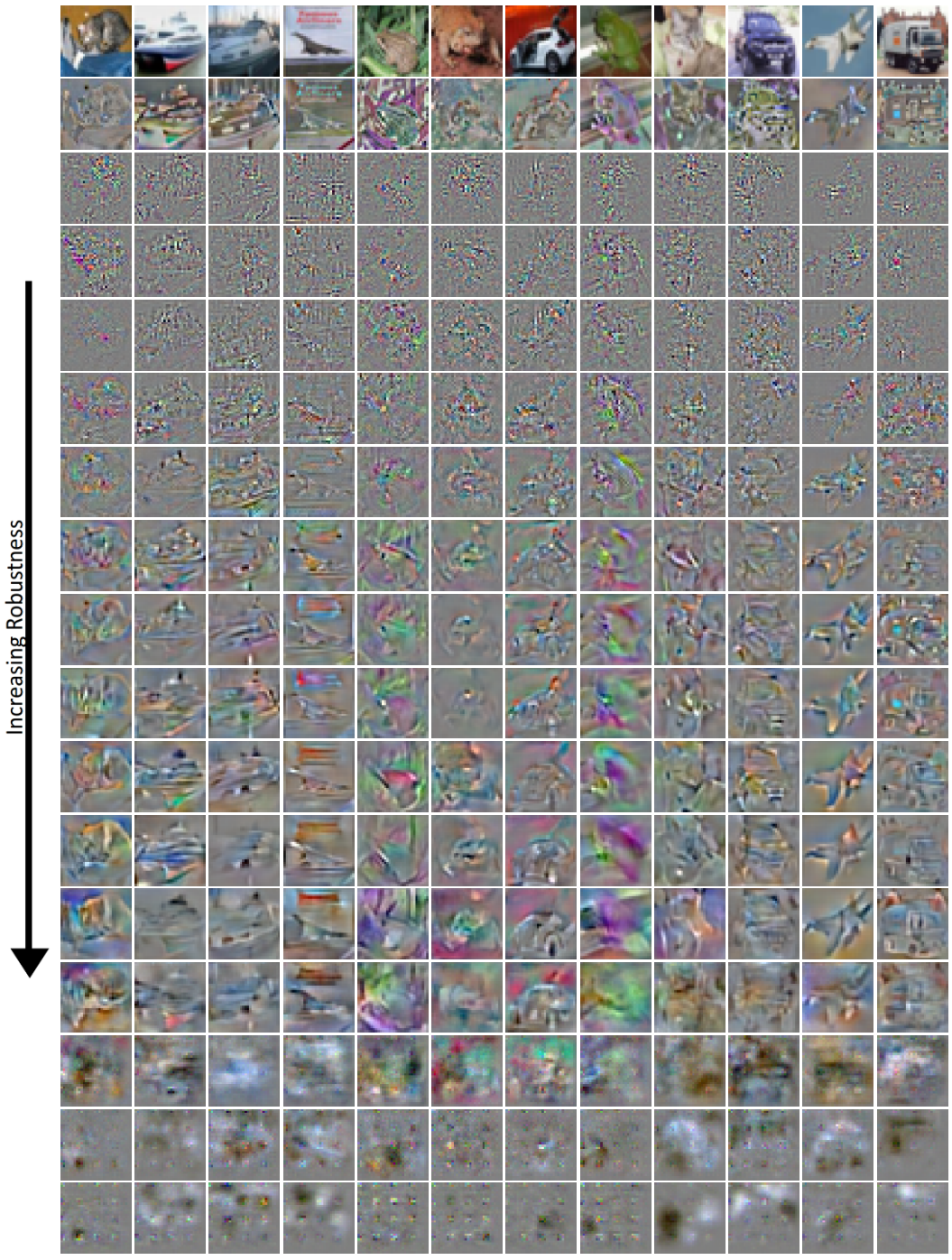


Figure 8: The input gradients of different models trained with a **smoothness penalty on CIFAR-10**. The top rows depict the image, the score, and the input gradients of unrobust models. The middle rows depict the perceptually aligned input gradients of robust models. The bottom rows depict the input gradients of excessively robust models. Best viewed in digital format.

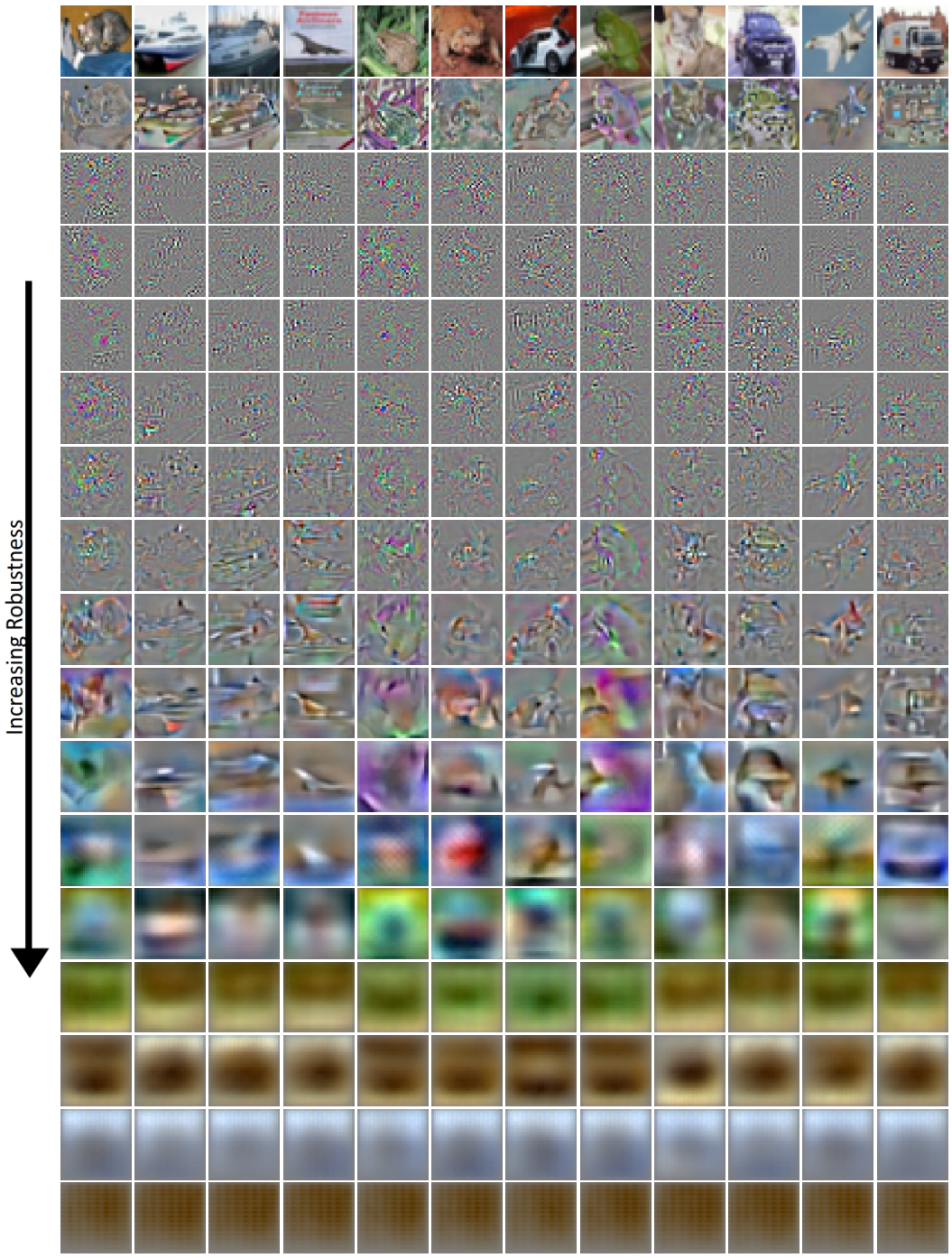


Figure 9: The input gradients of different models trained with **randomized smoothing on CIFAR-10**. The top rows depict the image, the score, and the input gradients of unrobust models. The middle rows depict the perceptually aligned input gradients of robust models. The bottom rows depict the input gradients of excessively robust models. Best viewed in digital format.

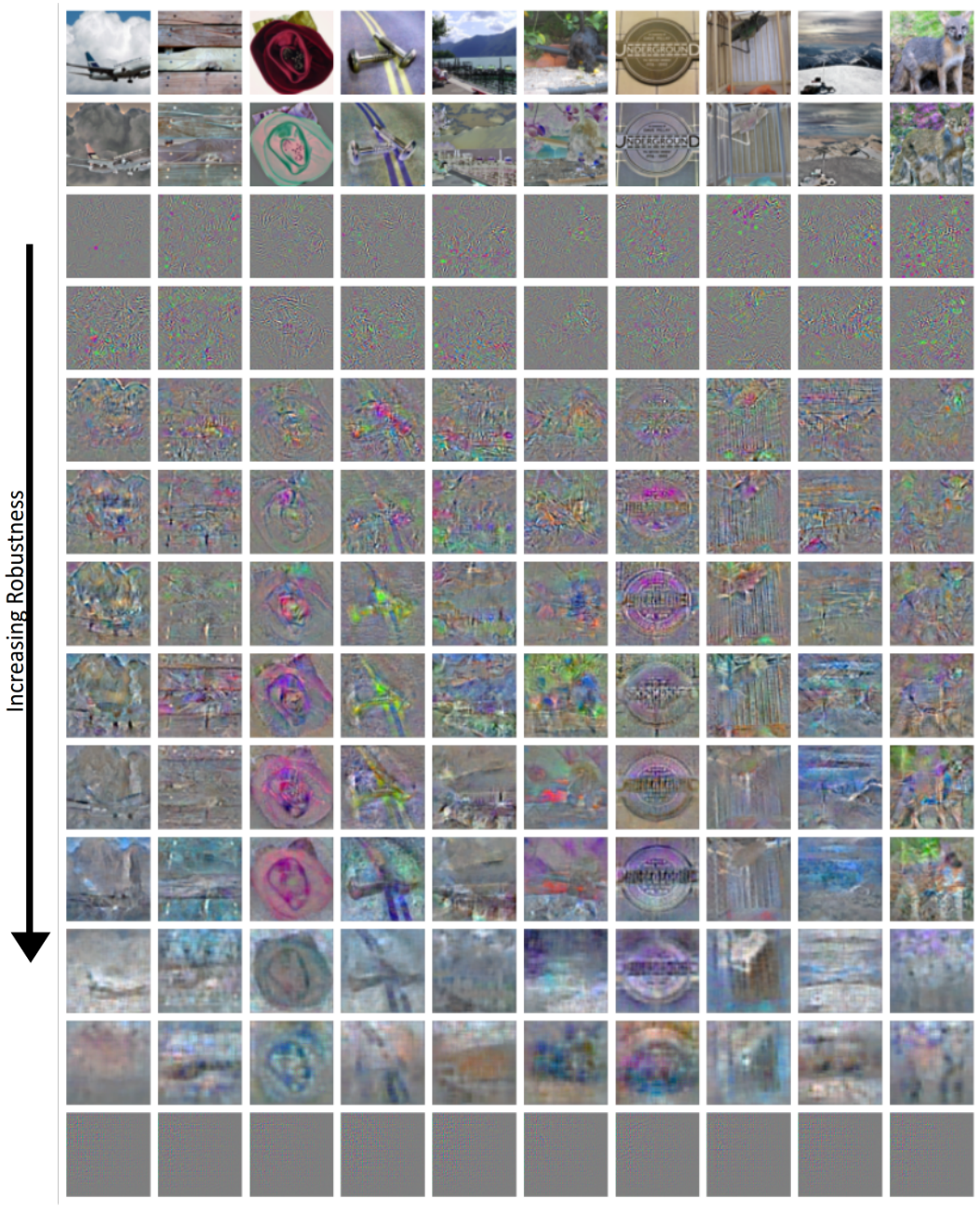


Figure 10: The input gradients of different models trained with **projected gradient descent on ImageNet-64x64**. The top rows depict the image, the score, and the input gradients of unrobust models. The middle rows depict the perceptually aligned input gradients of robust models. The bottom rows depict the input gradients of excessively robust models. Best viewed in digital format.

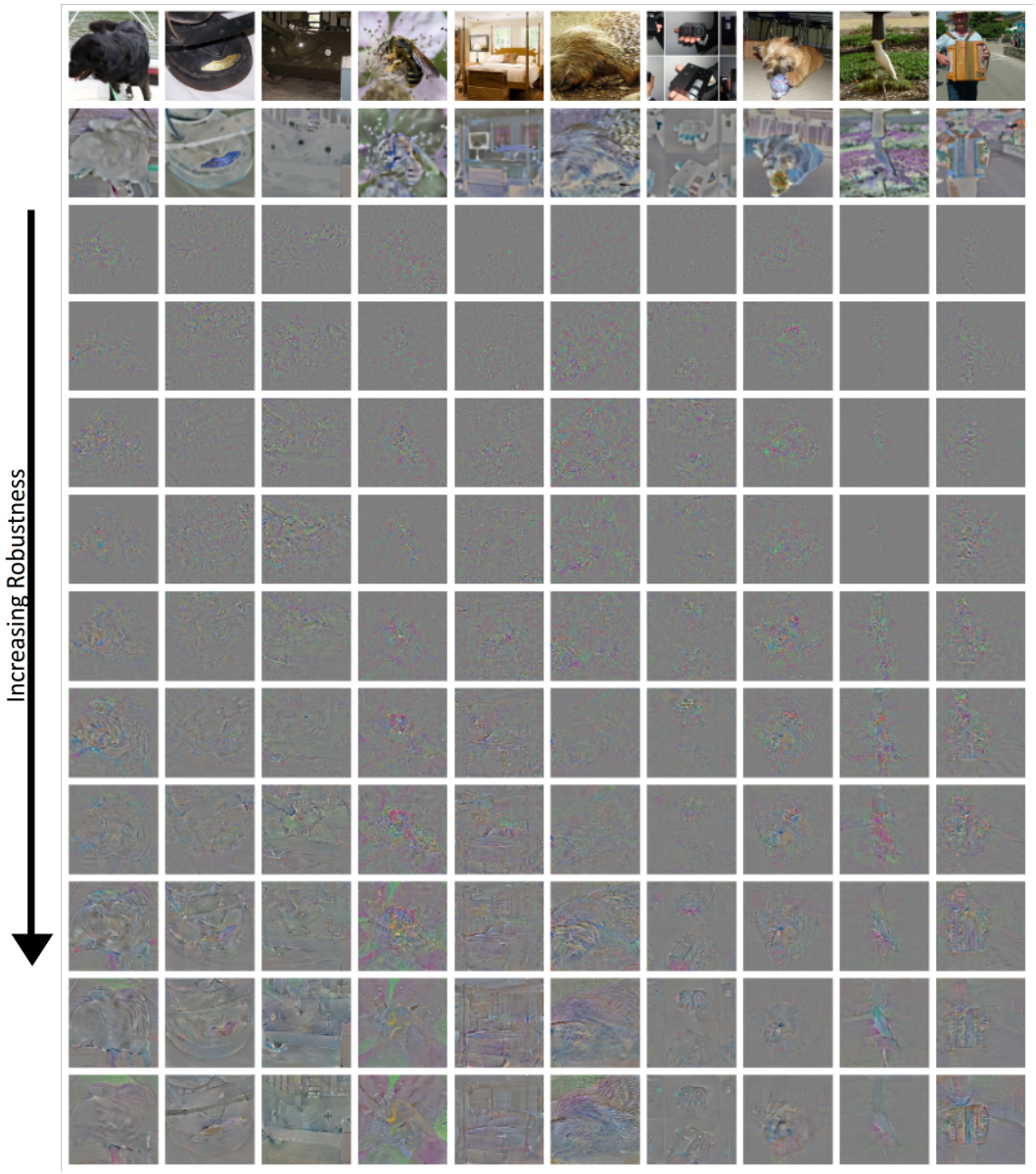
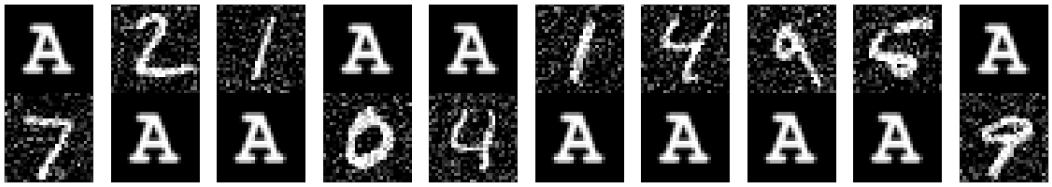
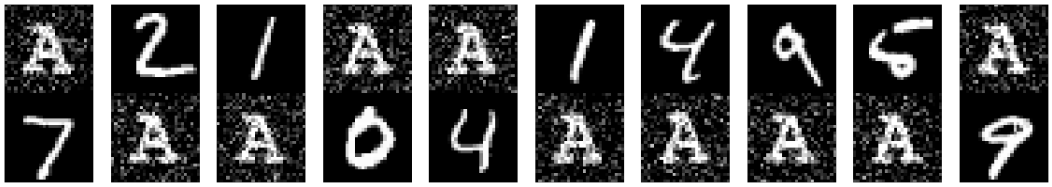


Figure 11: The input gradients of different models trained with **projected gradient descent on ImageNet**. The models are from [29]. The top rows depict the image, the score, and the input gradients of unrobust models. The bottom rows depict the perceptually aligned input gradients of robust models. Best viewed in digital format.

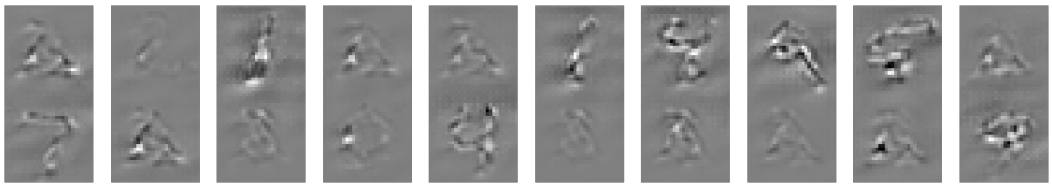
(a) Images from the data set.



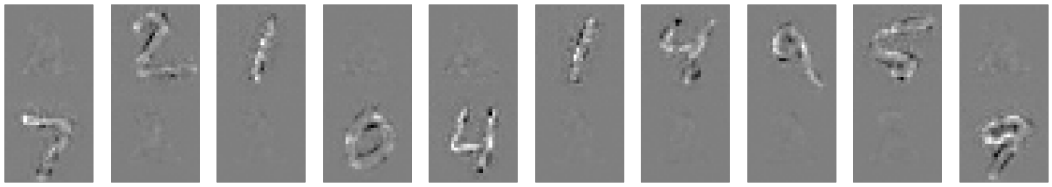
(b) Noise on the signal.



(c) Noise on the distractor.



(d) Input gradients of a Resnet18.



(e) Input gradients of an adversarially robust Resnet18.

Figure 12: The MNIST dataset with a distractor used to create Figure 4 in the main paper.