Exact Feature Collisions in Neural Networks

Supplementary Material

Utku Ozbulak, Shodhan Rao, Wesley De Neve, Joris Vankerschaver, Arnout Van Messem, Manvel Gasparyan

1. Decrease of the smallest eigenvalue with matrix size

We show that a matrix chosen at random is likely to have a smallest eigenvalue that decreases with the size of the matrix. While this does not show that a large neural network must necessarily have colliding features, it demonstrates that having (approximately) colliding features becomes increasingly likely as the network increases in size.

Lemma 1. *(Layer width and eigenvalues) Let $\mathcal{W} \in \mathbb{R}^{n \times n}$ be a matrix of the form $\mathcal{W} = \boldsymbol{W}\boldsymbol{W}^\top$, with $\boldsymbol{W} \in \mathbb{R}^{n \times p}$ a random matrix with standard normally distributed entries. Then the smallest eigenvalue of $\mathcal{W}$ converges to 0 (in probability) as n goes to infinity.*

Proof. Denote the smallest singular value of $\boldsymbol{W}$ by $\sigma_{\min}$. In Edelman [1, Corollary 3.1] it is shown that $n\sigma_{\min}$ converges in probability to a random variable with a known probability density, and more specifically that

$$P(n(\sigma_{\min})^2 \leq \epsilon) = \int_0^\epsilon \frac{1+\sqrt{x}}{2\sqrt{x}} e^{-(x/2+\sqrt{x})}\,\mathrm{d}x + o(1)$$

for all $\epsilon > 0$. Strictly speaking, Corollary 3.1 in Edelman [1] only applies to the case where $\boldsymbol{W}$ is square, but the argument can be extended to the non-square case.

In Tao and Vu [2] it was shown that the integral on the right hand side is equal to $C_\epsilon = 1 - e^{-\epsilon/2-\sqrt{\epsilon}}$. Using the fact that the $o(1)$ term is bounded for large enough n (and thus can be absorbed into C_ϵ), we have that

$$P(n(\sigma_{\min})^2 \leq \epsilon) = C_\epsilon.$$

Given that the smallest eigenvalue, $\lambda_{\min}$, of $\mathcal{W}$ is equal to $(\sigma_{\min})^2$, this shows that $\lambda_{\min}$ converges in probability to 0 as $n \to \infty$. □

2. Additional Experiments

Here, we provide additional experimental results about the properties of images with colliding features.

2.1 Feature collisions and secure models

In the main text, we evaluated feature collision properties of a number of models. In Table 1, we extend this evaluation to adversarially-secure models and report the result obtained with adversarially-

secure models alongside their regular counterparts. These results indicate that feature collisions indeed exist for a majority of the trainable weights (the second and the fourth column) for adversarially-secure models, where properties of adversarially-secure and regular models show significant similarities. Models evaluated in Table 1 are taken as they were released by Salman et al. [3].

Table 1: For the models provided in the second column, the number of trainable weights that satisfy Lemma 4.6 (3^{rd} column) and Lemma 4.9 (5^{th} column) of the main manuscript, with the percentage of those weights compared to all weights (4^{th} and 6^{th} columns) are provided. Moreover, the number of basis vectors in the kernel of all trainable weights (7^{th} column) and the number of basis vectors in only the first trainable weight (8^{th} column) are also given.

Task	Model	$\nu(\theta)$	$\frac{1}{n_\theta}\nu(\theta)$	$\mu(\theta)$	$\frac{1}{n_\theta}\mu(\theta)$	$\sum_{\boldsymbol{W}_i \in \theta}$ nullity($\boldsymbol{W}_i$)	nullity($\boldsymbol{W}_1$)
Secure Classification	ResNet-18 (Regular)	17	80%	17	80%	26,778	90
	ResNet-18 (L_2-secure, $\epsilon = 0.01$)	17	80%	17	80%	26,772	84
	ResNet-18 (L_2-secure, $\epsilon = 0.25$)	17	80%	17	80%	26,771	83
	ResNet-18 (L_2-secure, $\epsilon = 5$)	17	80%	17	80%	26,771	83
	ResNet-50 (Regular)	36	66%	33	61%	40,701	84
	ResNet-50 (L_2-secure, $\epsilon = 0.01$)	33	61%	33	61%	40,684	83
	ResNet-50 (L_2-secure, $\epsilon = 0.25$)	33	61%	33	61%	40,684	83
	ResNet-50 (L_∞-secure, $\epsilon = 0.5$)	33	61%	33	61%	40,684	83
	VGG-16 (Regular)	15	93%	14	87%	53,357	0
	VGG-16 (L_2-secure, $\epsilon = 3$)	15	93%	14	87%	54,549	0

2.2 Shared basis vectors across different models

The existence of an identical basis vector across multiple models would allow its usage to replace the results of feature collisions for those models without the search for additional bases. However, most architectures come with their own unique structure for trainable weights. This implies that basis vectors across models have different lengths, which makes them incomparable to each other. An exception to the aforementioned property are ResNet architectures, which always contain a convolutional layer with a kernel size of 7, a stride of 2, and padding of 3 as the first layer, which makes such models have basis vectors that are of the same length. However, after investigating these architectures in detail, we find that all basis vectors across ResNet models (including the adversarially secure ones) seem to be unique, which prevents the phenomenon described above. This means that data points with colliding features using the Null-space search have to be created for each model individually and cannot be transferred across models.

2.3 Image representations of basis vectors and visual examples

Image representations of basis vectors – As described in Section 4.2 of the main manuscript, the proposed Null-space search leverages basis vectors of the first trainable weight to create perturbations. We observe that these basis vectors themselves come in distinct visual forms when individually visualised. In Figure 1, we visualize the first 36 basis vectors of the first trainable weight of ResNet-50. These basis vectors are put into the form of an image by repetition, hence the single-color nature of the examples. Note that these perturbations (with varying intensities) can be added as is to the images in order to create colliding images.

Variety of perturbations – Instead of just repeating a single basis vector to create the whole perturbing image, one can also use a variety of basis vectors together. In Figure 2, we provide perturbation samples created using a random selection of basis vectors. Although these perturbation samples look no different than random noise, their application successfully creates colliding images.

Instead of random basis selection, it is also possible to repeat and create perturbations with distinct visual properties. In Figure 3, we provide examples for this case where we create perturbations using repetitions.

Perturbation multiplier (β) and visual distortion – In order to put into perspective the perturbation multiplier and its effect on visual distortion, in Figure 4, we provide perturbations created with random basis selection and their application on an image with progressively increasing values of β, starting with $\beta = 8$ (leftmost image) and increasing with a factor of two until $\beta = 8192$ (rightmost image).

Feature collision and interpretability maps – In Section 4.3 of the main manuscript, we mentioned that the application of interpretability methods that only rely on a forward pass will result in an identical output for colliding images if the target layer for visualization is the layer of collision or another layer after that. As for interpretability methods that make use of both a forward and a backward pass (i.e., the gradient), if the target layer is the layer of collision, or any layer before, the interpretability output will be different. However, any target layer after that will result in the same interpretability map. This observation is not immediately obvious since, given two inputs $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, $f(\boldsymbol{x}_1) = f(\boldsymbol{x}_2)$ does not guarantee that $f'(\boldsymbol{x}_1) = f'(\boldsymbol{x}_2)$ (e.g., $f(x) = x^2$ with $x_1 = 2$ and $x_2 = -2$). However, since neural networks are built in a composite fashion, as given in Eq. (1) of the main manuscript, gradients at layers after the layer of collision end up being the same for colliding images since they are calculated the with respect to the target layer.

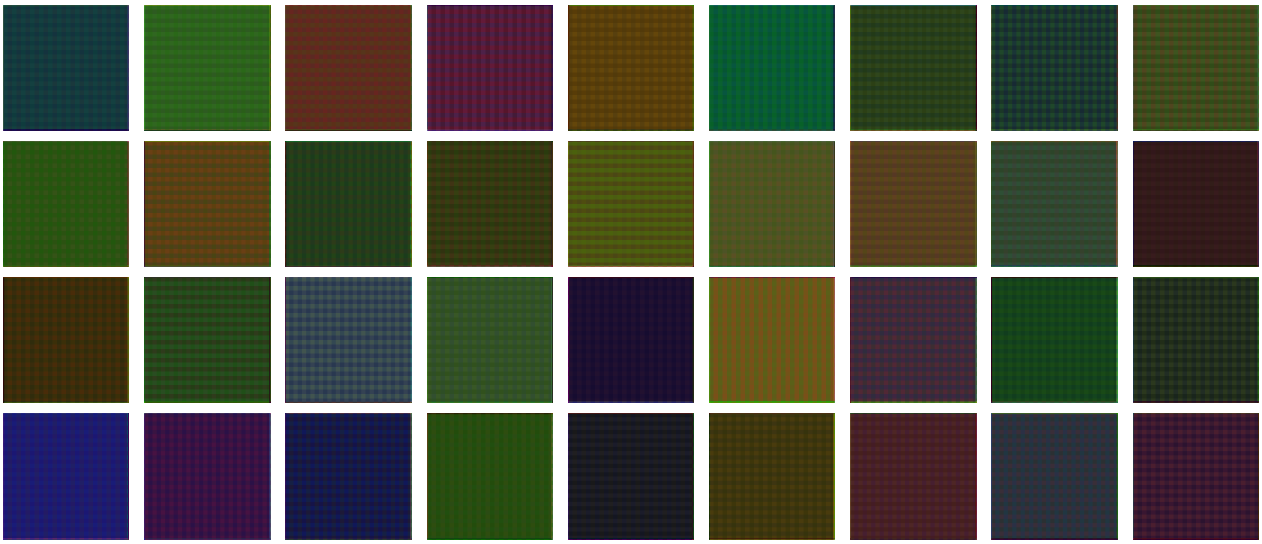

Figure 1: Image representations of the first 36 basis vectors of ResNet-50. For all of the illustrations, a single basis vector is repeated throughout the image.

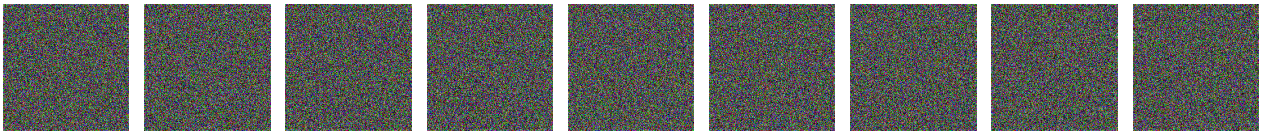

Figure 2: Perturbations created with random usage of basis vectors obtained from ResNet-50.

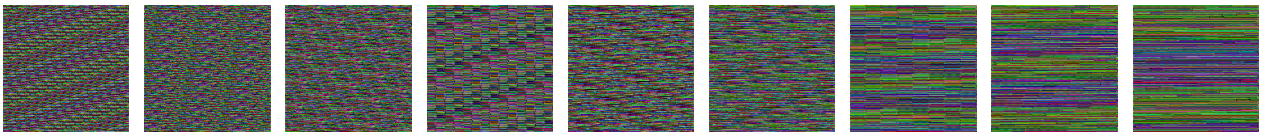

Figure 3: Perturbations created with the repetition of certain basis vectors obtained from ResNet-50.

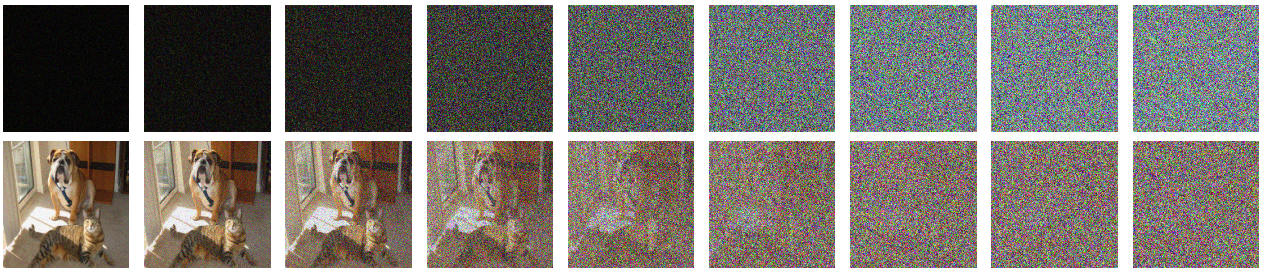

Figure 4: (top) Random perturbations created with the Null-space attack using ResNet-50 with progressively amplified β values starting with $\beta = 8$ and ending with $\beta = 2048$ (increased by a factor of 2 at each step) and (bottom) the application of the perturbation to an image. Note that all perturbed images below lead to the same prediction using ResNet-50.

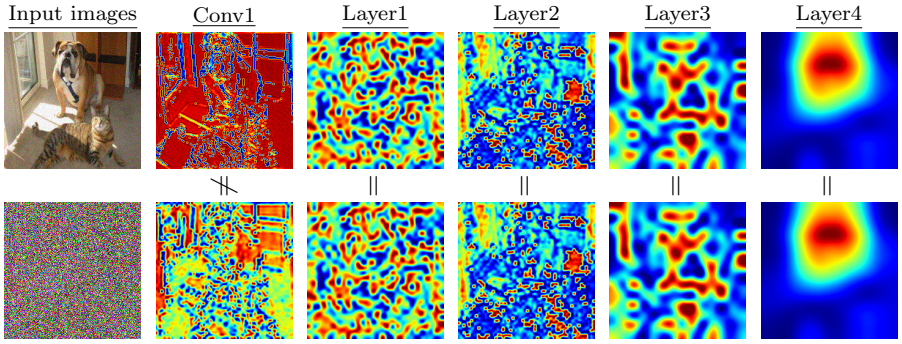

Figure 5: Interpretability maps obtained from ResNet-50 using GradCam with an input image and its (unrecognizable) colliding counterpart created with the Null-space search. Note that the collision occurs in the first layer of the network (i.e., Conv1) which causes the interpretability map obtained from that layer to be different for both images. Conversely, when targeting the forthcoming layers, interpretability map becomes the same since the forward and the backward pass (i.e., the gradient) is the same.

References

[1] Alan Edelman. Eigenvalues and Condition Numbers of Random Matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, 1988. doi: 10.1137/0609045.

[2] Terence Tao and Van Vu. Random matrices: The distribution of the smallest singular values. *Geometric and Functional Analysis*, 20:260–297, 2010.

[3] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.