Supplementary Material
"Closer Look at the Transferability of Adversarial Examples: How They Fool Different Models Differently"

1. Details of datasets

In Table 1, we provide the details of the datasets we used in the main paper.

Dataset	Class num.	Image size	Train	Test
Fashion-MNIST	10	(1,28,28)	60,000	10,000
CIFAR-10	10	(3,32,32)	50,000	10,000
STL-10	10	(3,96,96)	5,000	8,000

Table 1. Details of datasets we used in the main paper. Image size represents (channel, height, width) of images.

2. Adversarial Transferability Analysis

In this section, we provide supplementary results for our analysis of the class-aware transferability of AEs.

2.1. Details of evaluated models

All models were trained using the stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and weight decay of 0.0005.

For Fashion-MNIST, we trained models at an initial learning rate of 0.01, which decayed 0.1 times at the 20^{th} epoch with 40 epochs in total. Details of model architectures of FC-2/-4 and Conv-2/-4 are described in Table 2.

For CIFAR-10 and STL-10, we trained models at an initial learning rate of 0.01, which decayed 0.1 times at the 50^{th} epoch with 100 epochs in total. Additionally, we used data augmentation techniques to train models for CIFAR-10 and STL-10 to prevent strong overfit.

2.2. Model Similarity Analysis

Here, we show the supplementary results of class-aware transferability of AEs. The results for Fashion-MNIST, CIFAR-10, and STL-10 are shown in Figure 1, Figure 2, and Figure 3, respectively. These results include the analysis of various models, which are not in the main paper. Furthermore, the analysis of the AEs generated by Momentum Iterative Method (MIM) [2] is added. We confirm the consistency of our findings for various models and attacks: the fact that AEs tend to cause same mistakes, but a non-trivial proportion of different mistakes exist is consistent.

We also evaluated two optimization-based adversarial attacks, CW [1] and Deepfool [7]. Figure 4 shows the results for CIFAR-10. Since these optimization-based attacks try to find minimum perturbations that are enough to fool the source model F1, they hardly transferred between models. Interestingly, AEs generated by the optimization-based attacks do not transfer even to the source models at 80^{th} epochs. Therefore, fooled ratios were too small to analyze the class-aware transferability of the AEs.

2.3. Correlation Between Decision Boundaries' Distance and Class-aware Transferability

The correlations between $Dist(F1, F2)$, which is the quantitative measurement of the distance between models' decision boundaries, and the fooled ratio in Figure 5a. We confirmed that the distance metric of decision boundaries $Dist(F1, F2)$ is directly related to the non-target transferability, as stated by Tramer et al.[8]. In addition, the correlations between $Dist(F1, F2)$ and the same mistake ratio for all evaluated models and adversarial attacks are shown in Figure 5b. These results show that non-targeted transferability and same mistakes are strongly associated with each other.

2.4. Perturbation Size Analysis

The class-aware transferability of AEs when the perturbation size was gradually changed is shown in Figure 6a for the ResNet-18 source model and Figure 6b for the VGG-16 source model.

2.5. Decision Boundary Analysis

The visualization of the decision boundary for several different images is shown in Figure 7 for the ResNet-18 source model and Figure 8 for the VGG-16 source model.

2.6. Additional Adversarial Transferability Analysis on FGVC-Aircraft dataset

We additionally analyze the class-aware transferability of AEs generated for FGVC-Aircraft dataset [6] to under-

stand the effect of class similarity and the number of classes. FGVC Aircraft dataset contains 10,000 images, which are split into 6,667 images for train set and 3,333 images for test set. It is composed of only images of aircrafts, which are labeled hierarchically: For example, the label level of "variant", e.g. "Boeing 737-700", has 100 classes which are finest visually distinguishable classes. The label level of "manufacturer", e.g. "Boeing", has 40 classes of different manufacturers. We trained all models at the initial learning rate of 0.01, which decayed 0.1 times at the 100^{th} and 150^{th} epoch with 200 epochs in total.

Since FGVC-Aircraft contains only aircraft images, images for different classes are visually more similar than, e.g., "cat" and "truck" images in CIFAR-10. Therefore, it is more likely that AEs cause different mistakes unless the AEs have a substantial effect on fooling models towards a specific class.

Figure 9 shows the class-aware transferability of AEs generated for FGVC-Aircraft ("variants") dataset. Note that if AEs fool target models towards random directions, the proportion of same mistake ratio out of fooled ratio is 1% for a 100-class dataset.

We observe that non-targeted attacks caused same mistakes at a high rate. On the other hand, targeted attacks did not cause same mistakes as many as non-targeted attacks; however, still the proportions of same mistake ratio out of fooled ratio were more than 1%.

It is intriguing that, although FGVC-Aircraft ("variant") has 100 classes, the same mistake ratio is high with non-targeted attacks. It indicates that the AEs generated by non-targeted attacks had strong effects on fooling models towards specific classes, which suggests the existence of non-robust features of the specific classes.

For targeted attacks, although targeted attacks still cause a moderate number of same mistakes, they are not as much as non-targeted attacks. For example, the proportions of same mistake ratio out of fooled ratio when {F1, F2}={ResNet-18,VGG-16} were 9.4%, 6.0%, and 9.8% for targeted FGM, PGD, and MIM, respectively (the leftmost column of Figure 9).

To understand how different mistakes occur with the FGVC-Aircraft dataset, we further analyzed different mistakes at a class-wise level (Figure 10). For non-targeted FGM (Figure 10a), it is observed that different mistakes tend to occur within the same "manufacturer". It indicates that in the different mistake cases in non-targeted AEs, the non-robust features of a specific class were recognized as a different but similar class. On the other hand, targeted FGM (Figure 10b) caused different mistakes for other "manufacturers" more than non-targeted FGM. In general, targeted attacks are harder to perform than non-targeted attacks since targeted attacks are forced to aim at a specific class. Therefore, we think that this difficulty of targeted attacks can result in targeted attacks generating AEs with model-specific non-robust features for the source model, which are not likely to be perceived similarly by a target model. However, the differences between the mechanism and nature of targeted and non-targeted attacks are still not fully understood, which should be future work.

3. Non-robust Feature Analysis

3.1. Theory: The Difference in Learned Features Causes Different Behavior on Adversarial Attacks

In the paper, we showed that different models might classify AEs differently due to the different usage of non-robust features. In this section, we show a mathematical example of this phenomenon using a simple mathematical model proposed by Tspiras et al. [9].

3.1.1 Setup

As in Tspiras et al. [9], we consider a binary classification task in which the data consists of input-label pairs (x, y) sampled from a distribution D as follows:

$$y \overset{u.a.r}{\sim} \{-1, 1\}, \quad x_1 = \begin{cases} +y & w.p.\ \ p \\ -y & w.p.\ \ 1-p \end{cases},$$
$$x_2, ..., x_{d+1} \overset{i.i.d}{\sim} N(\eta y, 1),$$

where $N(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, and $p \leq 0.5$. Features x include strongly correlated feature x_1 and weakly correlated features $x_2, ..., x_{d+1}$ with small coefficient η. Here, the features $x_2, ..., x_{d+1}$ are non-robust to perturbations with size η.

3.1.2 Weakly-correlated features suffice standard classification accuracy

Although $x_1, ..., x_{d+1}$ only weakly correlate, and each cannot be predictive individually, they can be used to acquire good standard accuracy. As shown in [9], a simple linear classifier

$$f_{avg}(x) := sign(w_{unif}^T x),$$
$$\text{where } \ w_{unif} := \left[0, \frac{1}{d}, ..., \frac{1}{d}\right]$$

can achieve standard accuracy over 99% when $\eta \geq 3/\sqrt{d}$ (e.g. if d=1000, $\eta \geq 0.095$). Proof is shown below.

$$\begin{aligned} Pr\left[f_{avg}(x) = y\right] &= Pr\left[sign(w_{unif}x) = y\right] \\ &= Pr\left[\frac{y}{d}\sum_{i=1}^{d} N\left(\eta y, 1\right) > 0\right] \\ &= Pr\left[N\left(\eta, \frac{1}{d}\right) > 0\right] \\ &> 99\% \text{ (when } \eta \geq 3/\sqrt{d}) \end{aligned}$$

This means that even when features are weakly correlated, their collection could be predictable enough for classification.

3.1.3 Different usage of weakly-correlated features can cause different predictions

Next we think of classifiers f_A, f_B which have weights w_A, w_B as below.

$$\begin{aligned} f_A(x) &:= sign(w_A^T x), \\ \text{where } w_A &:= \frac{2}{d(d+1)}\left[0, 1, 2, ..., d\right] \\ f_B(x) &:= sign(w_B^T x), \\ \text{where } w_B &:= \frac{2}{d(d+1)}\left[0, d,\ d-1, ..., 1\right] \end{aligned}$$

These classifiers only use the weakly-correlated features, but they have a bias on weights, different from w_{unif}. The difference between these two classifiers is that the preference for using weakly correlated features is the opposite. These classifiers achieve a standard accuracy of over 99% when $\eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}$ (e.g. if d=1000, $\eta \geq 0.11$). The proof for f_A is shown below (the same calculation also proves for f_B).

$$\begin{aligned} Pr\left[f_A(x) = y\right] &= Pr\left[sign(w_A x) = y\right] \\ &= Pr\left[\frac{2y}{d(d+1)}\sum_{i=1}^{d} i \cdot N\left(\eta y, 1\right) > 0\right] \\ &= Pr\left[N\left(\eta\frac{d(d+1)}{2}, \frac{d(d+1)(2d+1)}{6}\right) > 0\right] \\ &> 99\% \left(\text{when } \eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}\right) \end{aligned}$$

Now we think of an adversarial attack that perturbs each feature x_i by a moderate ϵ. For instance, if $\epsilon = 2\eta$, adversary can shift each weakly-correlated feature towards $-y$. Here, we consider the case in which only the first half of the weakly-correlated features are perturbed by $\epsilon = 2\eta$: we consider perturbed features $x'_2, ..., x'_{k+1}$ are sampled i.i.d. from $N(-\eta y, 1)$, where $k = d/2$ (for simplicity, suppose d is an even number and $d \gg 2$). Then the probability of f_A correctly predicting y is over 90% when $\eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}$ (e.g. if d=1000, $\eta \geq 0.11$).

$$\begin{aligned} Pr\left[f_A(x') = y\right] &= Pr\left[sign(w_A x') = y\right] \\ &= Pr\Big[\frac{2y}{d(d+1)}\Big(\sum_{i=1}^{k} i \cdot N\left(-\eta y, 1\right) \\ &\qquad + \sum_{i=k+1}^{2k} i \cdot N\left(\eta y, 1\right)\Big) > 0\Big] \\ &= Pr\left[N\left(\eta k^2, \frac{d(d+1)(2d+1)}{6}\right) > 0\right] \\ &= Pr\left[N\left(\eta\frac{d^2}{4}\sqrt{\frac{6}{d(d+1)(2d+1)}}, 1\right) > 0\right] \\ &> 90\% \left(\text{when } \eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}\right) \end{aligned}$$

In the same way, the probability of f_B correctly predicting y is less than 10% when $\eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}$ (e.g. if d=1000, $\eta \geq 0.11$).

$$\begin{aligned} Pr\left[f_B(x') = y\right] &= Pr\left[sign(w_B x') = y\right] \\ &= Pr\left[N\left(-\eta\frac{d^2}{4}\sqrt{\frac{6}{d(d+1)(2d+1)}}, 1\right) > 0\right] \\ &< 10\% \left(\text{when } \eta \geq \sqrt{\frac{6(2d+1)}{d(d+1)}}\right) \end{aligned}$$

Therefore, it is proved that there exists a case in which the perturbed input x' is correctly predicted by f_A while incorrectly predicted by f_B. This analysis shows that how each model puts weights on weakly-correlated features can determine the transferability of adversarial examples. Similarly, simply extending this analysis to a multi-class setting can theoretically show that there is a possibility to attack different models to cause different mistakes when the models use features differently.

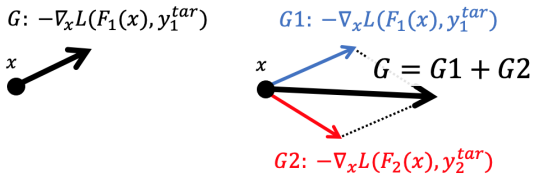
3.2. Supplementary Results

Here, we provide supplementary results and details of the non-robust feature analysis.

3.2.1 N-targeted attack

Figure 11 describes the difference between vanilla targeted attack and the N-targeted attack. N-targeted attack aims to fool multiple models towards each specified target class. It simply adds up the gradients for all target models.

Table 3 shows the accuracy of models on the AEs generated by the N-targeted attack, which constructs non-robust sets.



(a) Targeted attack (b) N-targeted attack (N=2)

Figure 11. Difference between (a) targeted attack and (b) proposed N-targeted attack (N=2), which sums up all gradients for all target models ($G = G1 + G2$) and aims to mislead model $F1$ towards class y_1^{tar} and model $F2$ towards class y_2^{tar}.

3.2.2 Full Results and Optimized Hyperparameters

For CIFAR-10 and STL-10, we conducted a grid search to obtain the best hyperparameters for training models on the constructed non-robust sets. The grid search area of hyperparameters is shown in Figure 12. Initial learning rate, batch size, and level of data augmentation were optimized. The results and corresponding hyperparameters are shown in Figure 4, Figure 5, and Figure 6 for Fashion-MNIST, CIFAR-10, and STL-10, respectively.

3.2.3 Accuracy Curves

Train and test accuracy curves for training models on the constructed non-robust sets are shown in Figure 13 (CIFAR-10). Note that train accuracy represents accuracy on the constructed non-robust sets, which seem completely mislabeled for humans, and test accuracy represents accuracy on the original test set that is correctly labeled.

Following the experiment from Ilyas et al. [4], the accuracy numbers reported correspond to the last iteration since we cannot do meaningful early-stopping as the validation set itself comes from the constructed non-robust set and not from the true data distribution.

4. Potential Application of Our Findings

In this paper, we have mainly focused on the theoretical understanding of adversarial transferability. This section lists some potential applications to use our main findings.

4.1. Attack-side perspective

Using the N-targeted attack concept is one potential application. It can be used to attack systems with the primary classifier model and an AE detection model. Experiments showed that it might be possible to generate AEs with non-robust features that are recognized by the primary classifier but not by the AE detection model. Another potential application of our paper is to generate transferable AEs. Our paper suggests that AEs transfer when they have non-robust features that DNNs commonly recognize. Therefore, the promising direction to generate transferable AEs is to investigate how to find "commonly perceived" non-robust features by different DNNs.

4.2. Defense-side perspective

In general, our work further supports viewing adversarial vulnerability as a feature learning problem, as asserted by Ilyas et al. [4]: to reduce the adversarial vulnerability of DNNs, it is necessary to restrict DNNs from learning non-robust features that humans do not use. Our contribution is to support this view by showing that non-robust features can explain the transferability of AEs, even from the more detailed perspective of class-aware transferability. One specific approach our paper suggests is to ensemble models: it can alleviate the sensitivity to non-robust features learned by a particular model and become only sensitive to the non-robust features commonly learned by all models to be ensembled. In other words, the ensemble model may rely less on specific non-robust features than a single model, which can reduce adversarial vulnerability.

FC-2	FC-4	Conv-2	Conv-4
Linear: (784, 500) ReLU Linear: (500, 10)	Linear: (784, 500) ReLU Linear: (500, 200) ReLU Linear: (200, 100) ReLU Linear: (100, 10)	Conv2d: (1, 32, 3, 1) ReLU Conv2d: (32, 64, 3, 1) ReLU Maxpool Linear: (9216, 128) ReLU Dropout Linear: (128, 10)	Conv2d: (1, 32, 3, 1) ReLU Conv2d: (32, 64, 3, 1) ReLU Conv2d: (64, 128, 3, 1) ReLU Maxpool Conv2d: (128, 128, 3, 1) ReLU Maxpool Linear: (9216, 128) ReLU Dropout Linear: (128, 10)

Table 2. Model architectures used for Fashion-MNIST. "Linear: (i, j)" is a fully connected layer with input size i and output size j. "Conv2d: (C_i, C_o, k, s)" is a convolution layer with input channel size C_i, output channel size C_o, kernel size k, and stride s.

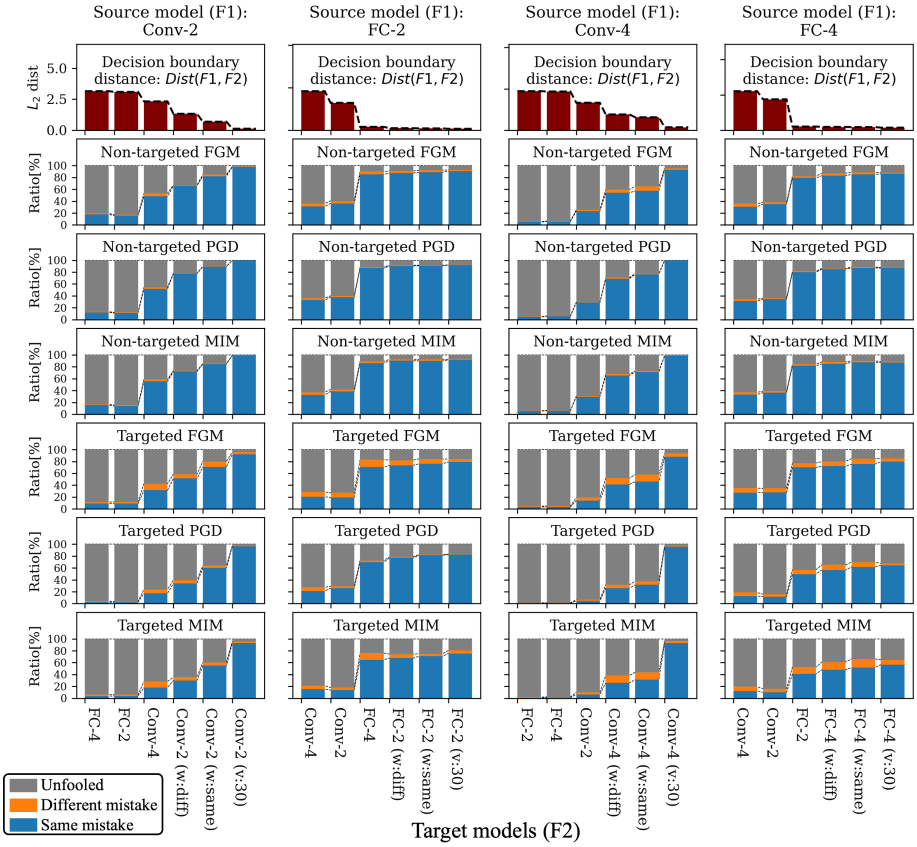


Figure 1. Class-aware transferability of adversarial attacks for Fashion-MNIST. We evaluate FGM [3], PGD [5], and MIM [2] with both non-targeted and targeted objectives. AEs were l2-bounded by ϵ=1.0. Order of F2 is sorted by $Dist(F1, F2)$ (1st row) for each F1 so rightmost F2 was estimated to be more similar to F1.

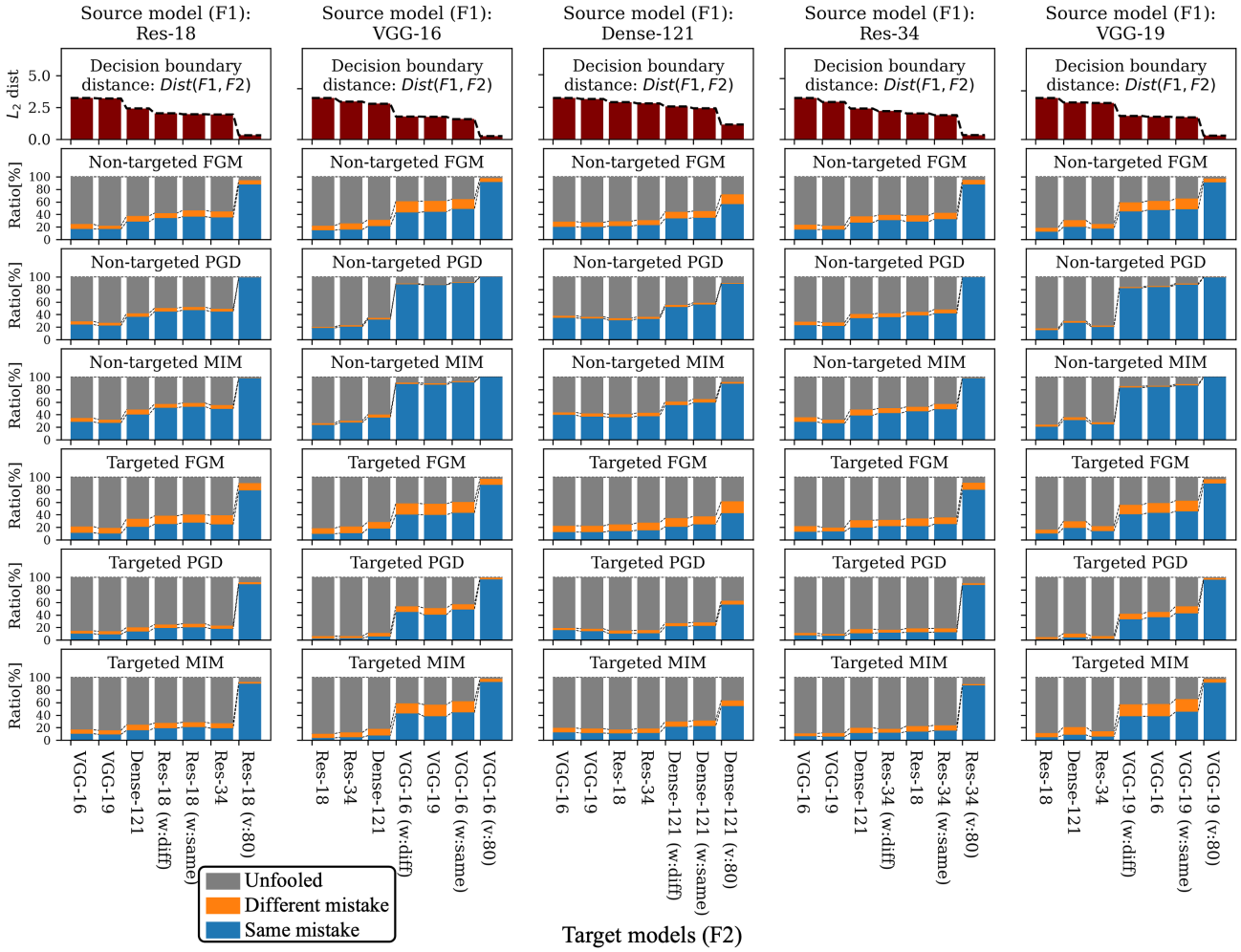


Figure 2. Class-aware transferability of adversarial attacks for CIFAR-10. We evaluate FGM [3], PGD [5], and MIM [2] with both non-targeted and targeted objectives. AEs were l2-bounded by ϵ=1.0. Order of F2 is sorted by $Dist(F1, F2)$ (1st row) for each F1 so rightmost F2 was estimated to be more similar to F1..

Dataset	Non-robust set constructed for		$F1(X') = Y1$	$F2(X') = Y2$	$F1(X') = Y1$ & $F2(X') = Y2$
	F1	F2			
Fashion-MNIST	Conv-2	FC-2	92.3	60.0	60.0
	Conv-2	Conv-2 (w:same)	94.6	93.8	93.0
	FC-2	FC-2 (w:same)	58.7	58.5	46.0
CIFAR-10	ResNet-18	VGG-16	95.6	99.0	95.4
	ResNet-18	ResNet-18 (w:same)	94.1	94.1	92.0
	VGG-16	VGG-16 (w:same)	99.5	99.5	99.2
STL-10	ResNet-18	VGG-16	99.2	99.7	99.0
	ResNet-18	ResNet-18 (w:same)	99.2	99.3	99.0
	VGG-16	VGG-16 (w:same)	99.8	99.5	99.2

Table 3. Accuracy of models attacked using AEs X' generated by N-targeted attack, which constructs non-robust sets. $Y1$ is target classes for N-targeted attack for model $F1$, and $Y2$ is that for model $F2$. These results are particularly interesting: in a white-box setting, it is easy to generate AEs that lead to different sequences of classes $Y1$ and $Y2$ (success rate $> 90\%$ for CIFAR-10 and STL-10).

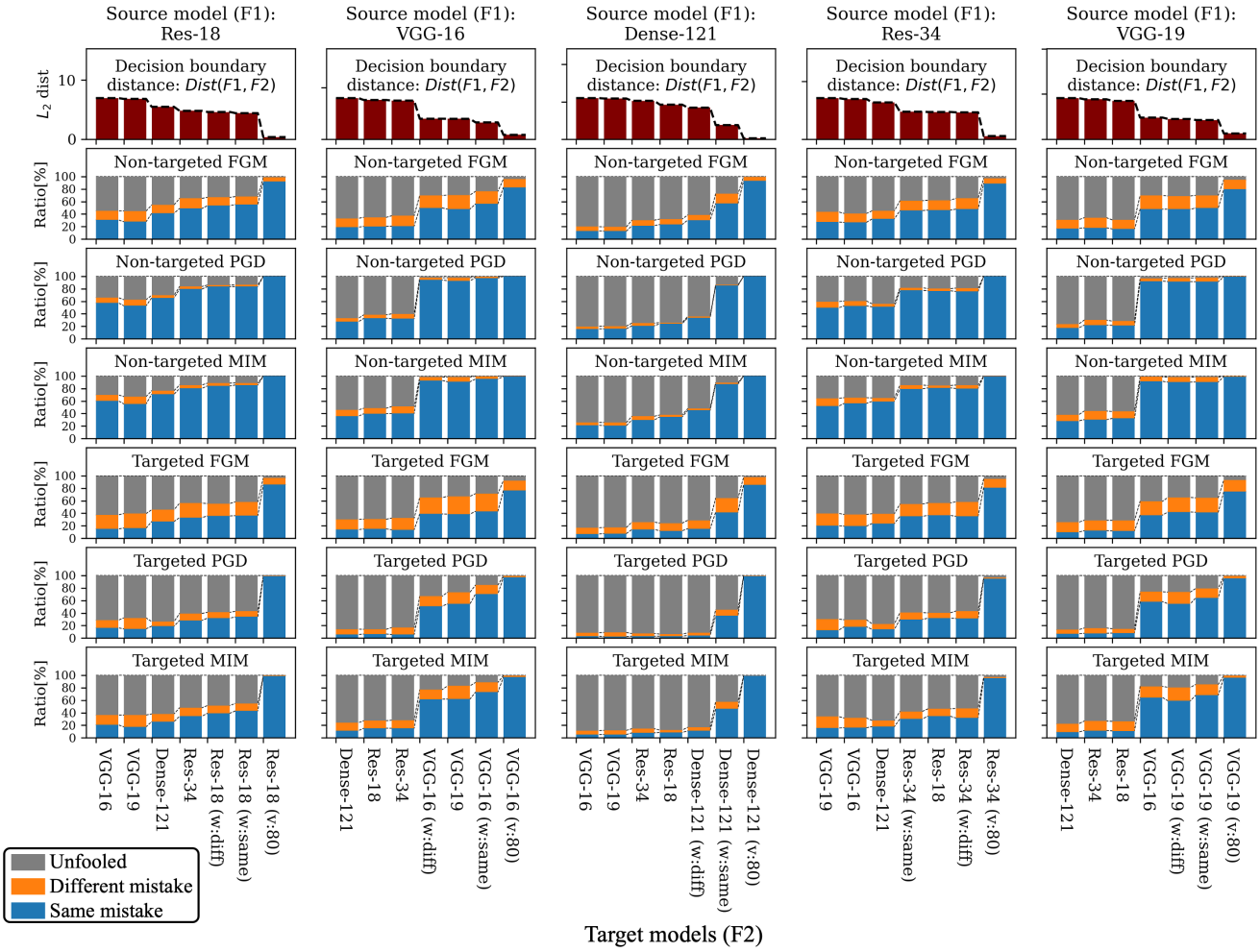


Figure 3. Class-aware transferability of adversarial attacks for STL-10. We evaluate FGM [3], PGD [5], and MIM [2] with both non-targeted and targeted objectives. AEs were l2-bounded by ϵ=5.0. Order of F2 is sorted by $Dist(F1, F2)$ (1st row) for each F1 so rightmost F2 was estimated to be more similar to F1.

.

Dataset	Non-robust set constructed for	Train set	Trained model	Test acc (X,Y)	Initial learning rate	Batch size	Data aug.
Fashion-MNIST	F1: Conv-2 F2: FC-2	$D'_1 : (X', Y1)$	Conv-2	82.9	0.01	256	None
			FC-2	62.1			
		$D'_2 : (X', Y2)$	Conv-2	80.3			
			FC-2	75.4			
	F1: Conv-2 F2: Conv-2 (w:same)	$D'_1 : (X', Y1)$	Conv-2	81.9			
			FC-2	66.2			
		$D'_2 : (X', Y2)$	Conv-2	82.4			
			FC-2	67.1			
	F1: FC-2 F2: FC-2 (w:same)	$D'_1 : (X', Y1)$	Conv-2	79.0			
			FC-2	80.5			
		$D'_2 : (X', Y2)$	Conv-2	77.6			
			FC-2	81.4			

Table 4. Non-robust features analysis for Fashion-MNIST. Initial learning rate, batch size, and data augmentations were fixed.

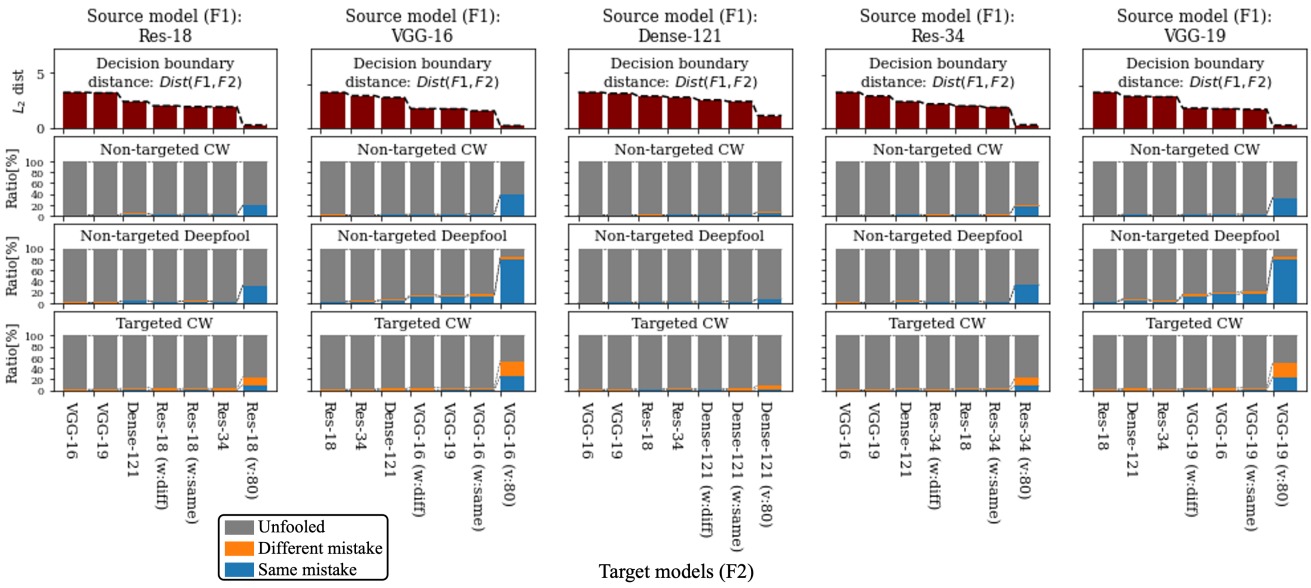
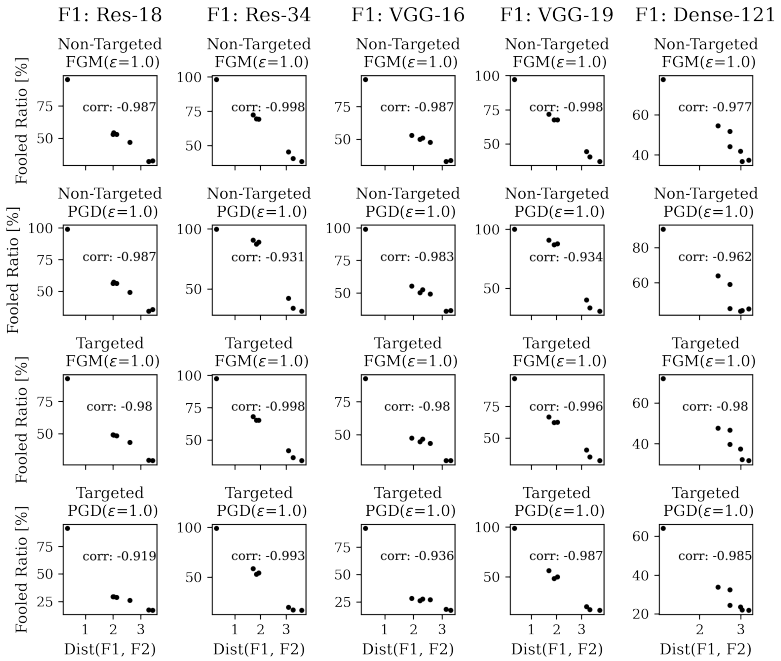


Figure 4. Class-aware transferability of optimization-based adversarial attacks for CIFAR-10. We evaluate CW [1] and Deepfool [7] (Deepfool is defined only for a non-targeted objective). Order of F2 is sorted by $Dist(F1, F2)$ (1st row) for each F1 so rightmost F2 was estimated to be more similar to F1. Since these optimization-based attacks try to find minimum perturbations that are enough to fool the source model F1, they hardly transfer between models.

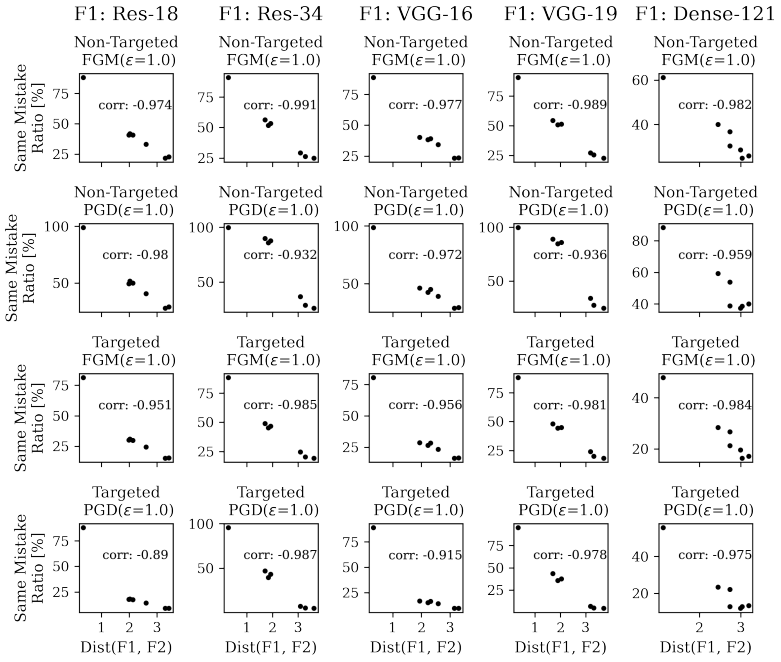
.

Dataset	Non-robust set constructed for	Train set	Trained model	Test acc (X,Y)	Initial learning rate	Batch size	Data aug.
CIFAR-10	F1: ResNet-18 F2: VGG-16	$D'_1 : (X', Y1)$	ResNet-18	51.3	0.005	128	Level 3
			VGG-16_bn	53.9	0.001	128	Level 2
		$D'_2 : (X', Y2)$	ResNet-18	10.2	0.05	128	Level 1
			VGG-16_bn	71.0	0.01	128	Level 1
	F1: ResNet-18 F2: ResNet-18 (w:same)	$D'_1 : (X', Y1)$	ResNet-18	50.1	0.05	128	Level 3
			VGG-16_bn	54.1	0.005	256	Level 3
		$D'_2 : (X', Y2)$	ResNet-18	59.2	0.05	128	Level 1
			VGG-16_bn	58.9	0.005	128	Level 3
	F1: VGG-16 F2: VGG-16 (w:same)	$D'_1 : (X', Y1)$	ResNet-18	63.5	0.05	128	Level 2
			VGG-16	68.8	0.01	256	Level 3
		$D'_2 : (X', Y2)$	ResNet-18	11.0	0.01	128	Level 1
			VGG-16	73.1	0.01	128	Level 1

Table 5. Non-robust features analysis for CIFAR-10. Optimized hyperparameters are shown besides the test accuracy.

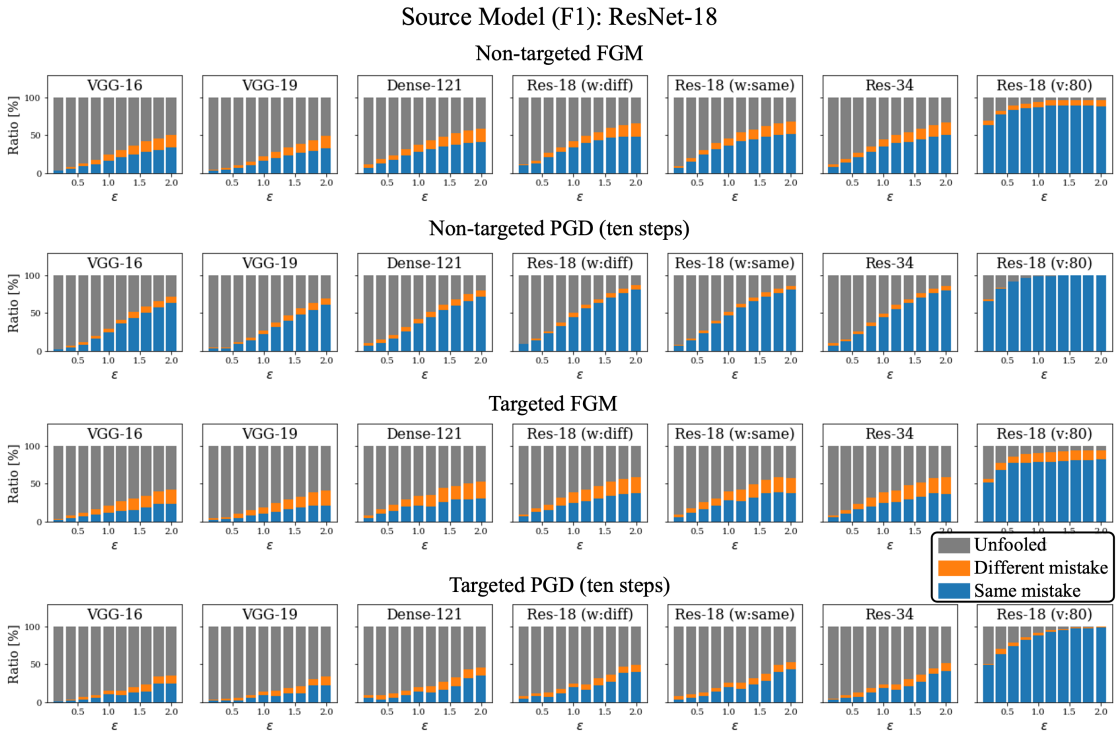


(a) Correlation between $Dist(F1, F2)$ and fooled ratio for CIFAR-10.

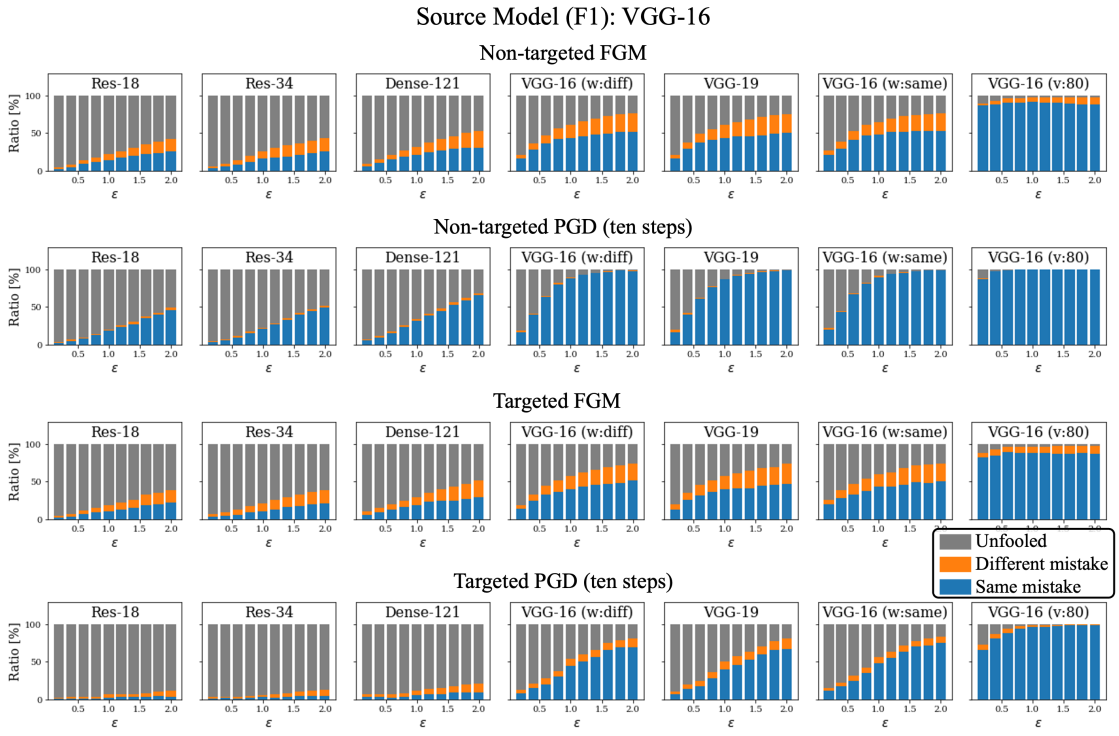


(b) Correlation between $Dist(F1, F2)$ and same mistake ratio for CIFAR-10.

Figure 5. Correlations (1) between $Dist(F1, F2)$ and fooled ratio ratio (Figure 5a), and (2) between $Dist(F1, F2)$ and same mistake (Figure 5b) for CIFAR-10. ResNet-18, ResNet-34, VGG-16, VGG-19, and DenseNet-121 source models (1st to 5th columns, respectively) were attacked by non-targeted and targeted attack (1-2nd and 3-4th rows, respectively) using FGM and PGD (ten-step) (1,3rd and 2,4th rows, respectively) methods. For each source model, the results of the corresponding seven target models (shown in Figure 2) are displayed in a scatter plot.



(a) Class-aware transferability of AEs generated for ResNet-18 source model.



(b) Class-aware transferability of AEs generated for VGG-16 source model.

Figure 6. Class-aware transferability of AEs when the perturbation size ϵ is gradually changed (CIFAR-10). Here we show the results of attacking the source model of ResNet-18 (Figure 6a) and VGG-16 (Figure 6b) with non-targeted attack (1st and 2nd rows) and targeted attack (3rd and 4th rows), using FGM (1st and 3rd rows) and PGD (ten-step) (2nd and 4th rows) methods.

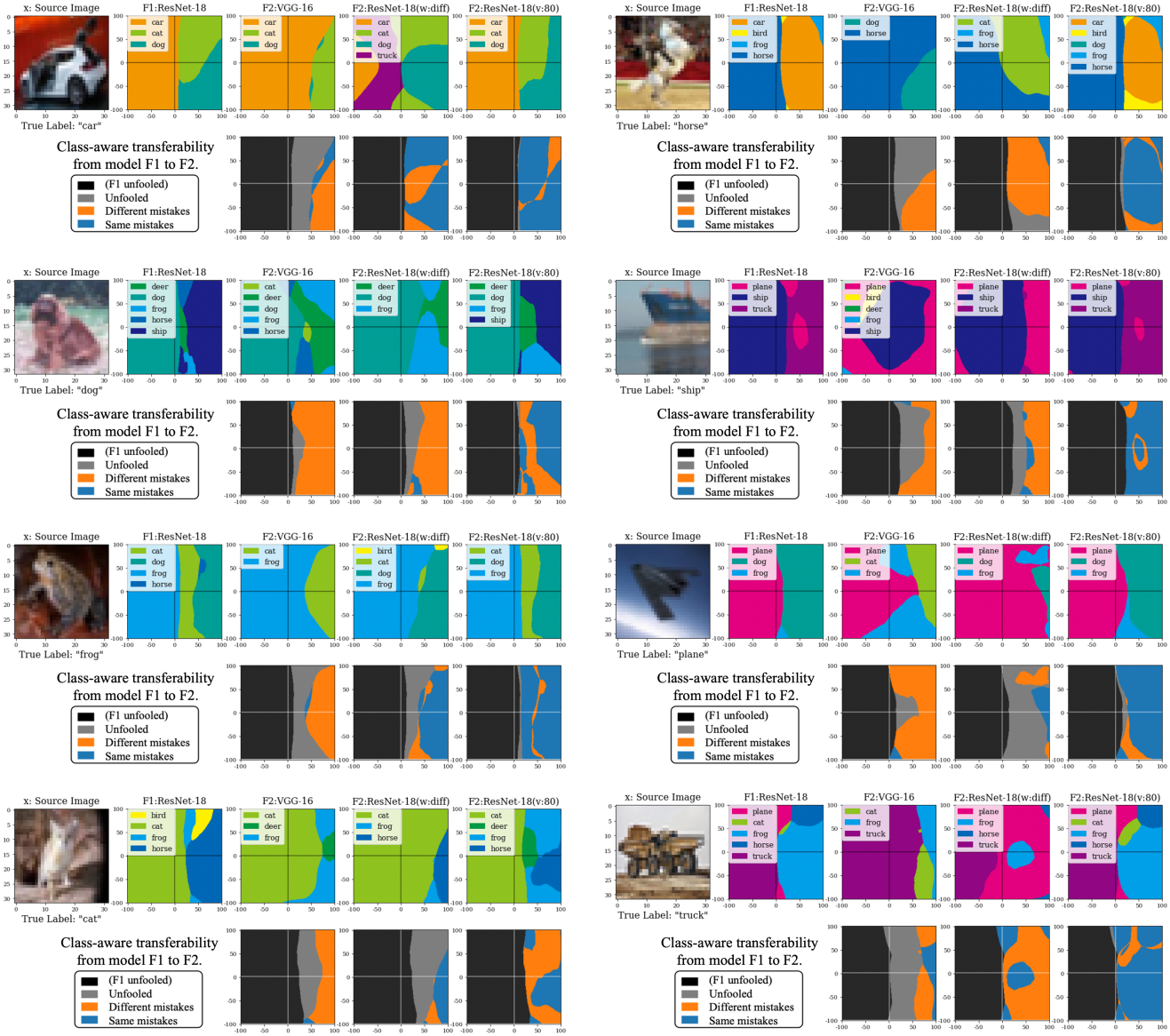


Figure 7. Visualization of decision boundaries when the source model is ResNet-18 (CIFAR-10). For each image, the first row shows the classification results, where each color represents a certain class. The second row shows to which area the three cases of class-aware transferability correspond. The distance from (0, 0) point to the closest decision boundary along the x-axis corresponds to the metric $d(F1, x)$ for each image x. The unit of each axis is 0.02 in l2 distance.

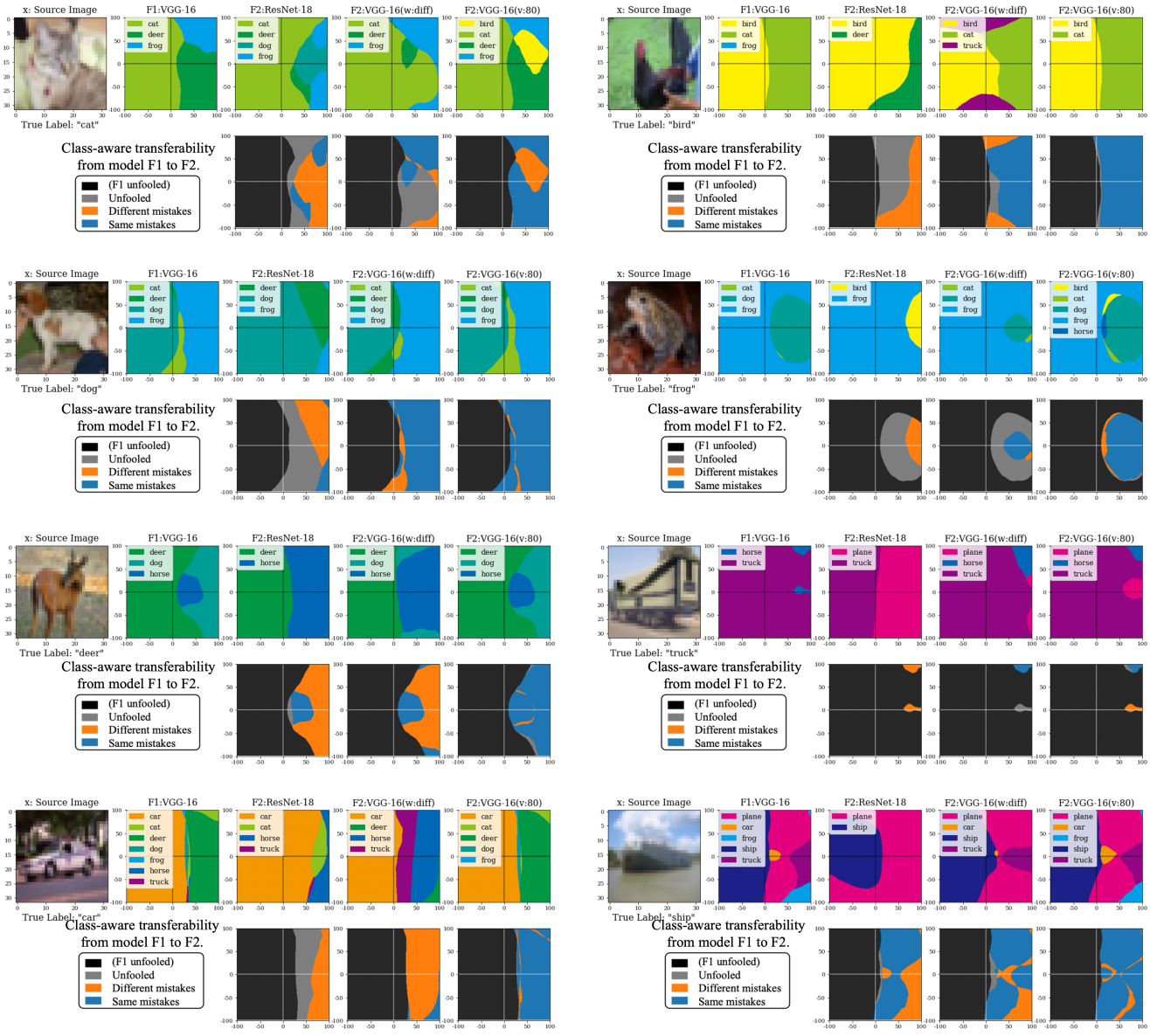


Figure 8. Visualization of decision boundaries when the source model is VGG-16 (CIFAR-10). For each image, the first row shows the classification results, where each color represents a certain class. The second row shows to which area the three cases of class-aware transferability correspond. The distance from (0, 0) point to the closest decision boundary along the x-axis corresponds to the metric $d(F1, x)$ for each image x. The unit of each axis is 0.02 in l2 distance.

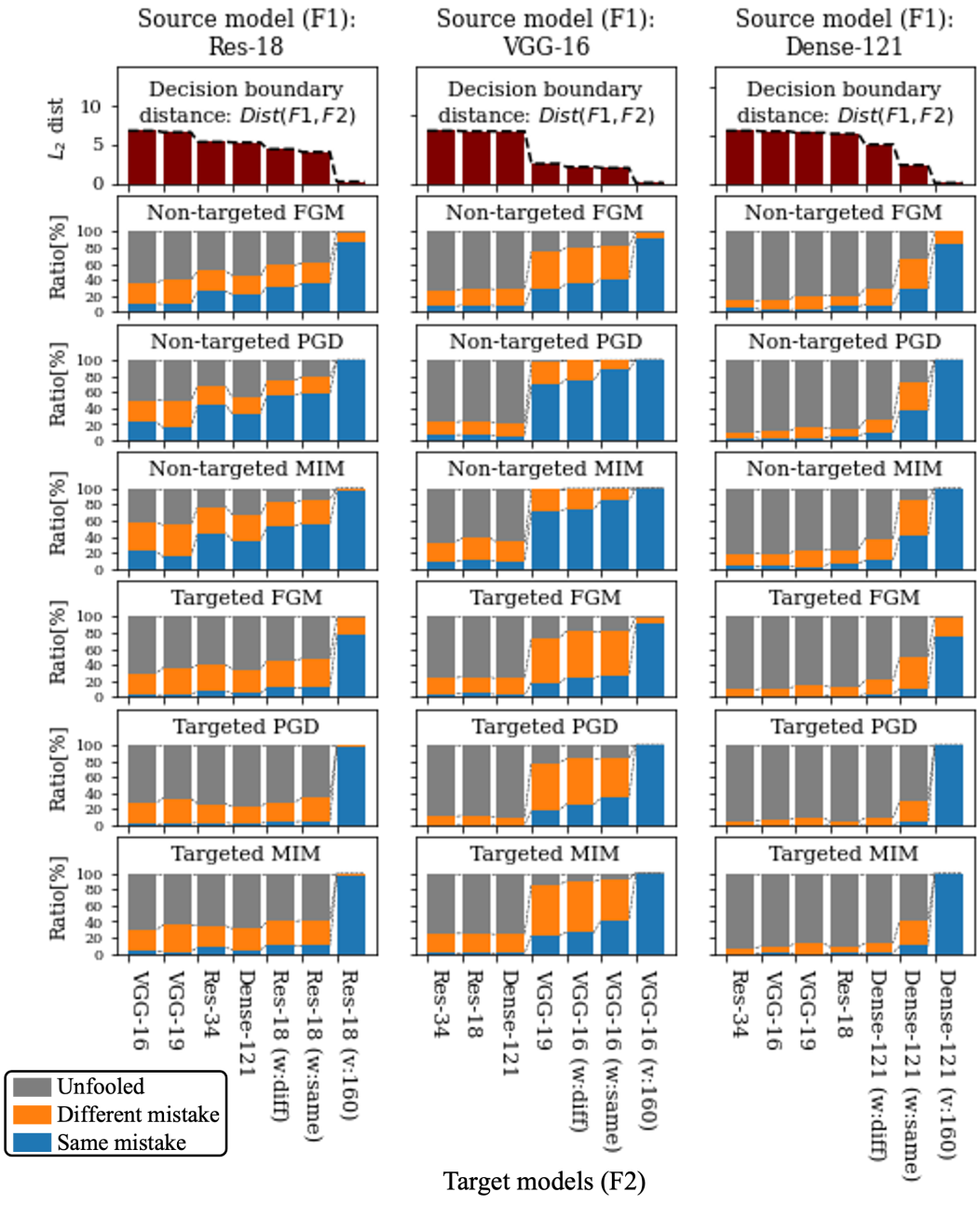
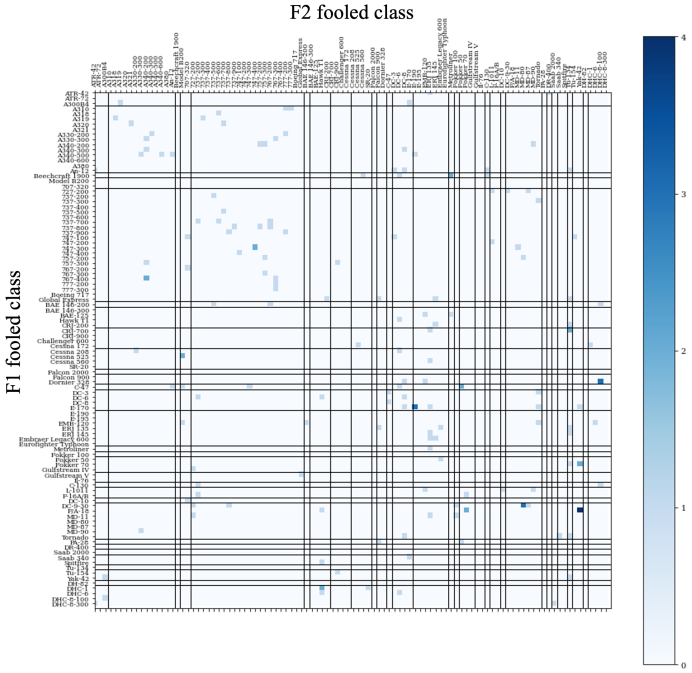
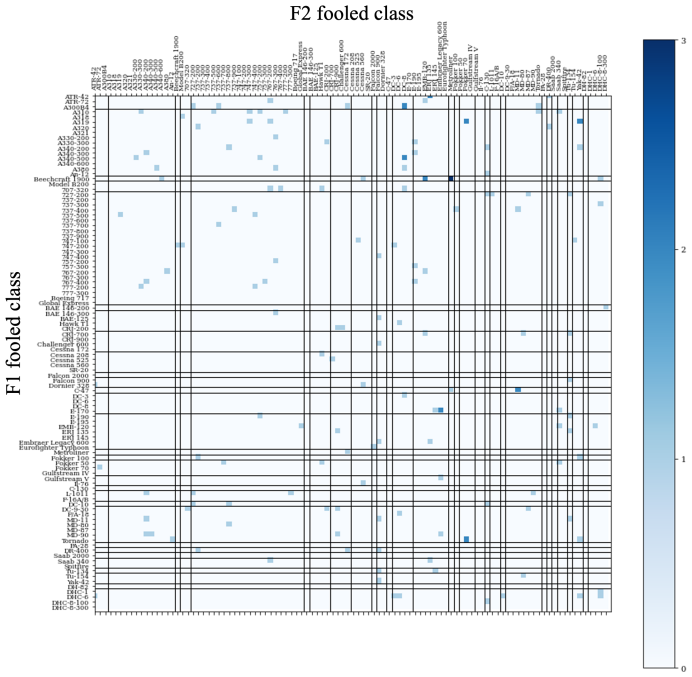


Figure 9. Class-aware transferability of adversarial attacks for FGVC-Aircraft ("variant"). AEs were l_2-bounded by ϵ=5.0. Order of F2 is sorted by $Dist(F1, F2)$ (1st row) for each F1 so rightmost F2 was estimated to be more similar to F1.



(a) Class-wise analysis of different mistakes caused by AEs generated by non-targeted FGM.



(b) Class-wise analysis of different mistakes caused by AEs generated by targeted FGM.

Figure 10. Class-wise analysis of "different mistakes" for FGVC-Aircraft ("variant"). The y-axis shows the classes to which the source model F1 misclassified the AEs and the x-axis shows the classes to which the target model F2 misclassified the AEs. Each value represents the number of each case. The source model F1 is ResNet-18 and the target model F2 is VGG-16. The classes are sorted by the "manufacturer" labels and the black lines separates "variant" classes for each "manufacturer". For non-targeted FGM (Figure 10a), it is observed that different mistakes tend to occur within the same "manufacturer". On the other hand, targeted FGM (Figure 10b) caused different mistakes for other "manufacturers" more than non-targeted FGM.

Initial learning rate	**Batch size**	**Data augmentation**
0.05, 0.01, 0.005, 0.001	128, 256	**Level 1:** Random crop by padding=4 **Level 2:** Random crop by padding=4 + Random horizontal flip **Level 3:** Random crop by padding=4 + Random horizontal flip + Random affine augmentation

Figure 12. Grid search area to obtain hyperparameters for training models on the constructed non-robust sets (used for CIFAR-10 and STL-10).

Dataset	Non-robust set constructed for	Train set	Trained model	Test acc (X,Y)	Initial learning rate	Batch size	Data aug.
STL-10	F1: ResNet-18 F2: VGG-16	$D'_1 : (X', Y1)$	ResNet-18	24.0	0.001	256	Level 3
			VGG-16_bn	25.4	0.001	256	Level 2
		$D'_2 : (X', Y2)$	ResNet-18	53.7	0.005	128	Level 1
			VGG-16_bn	57.2	0.01	128	Level 2
	F1: ResNet-18 F2: ResNet-18 (w:same)	$D'_1 : (X', Y1)$	ResNet-18	18.6	0.001	256	Level 3
			VGG-16_bn	20.1	0.001	256	Level 3
		$D'_2 : (X', Y2)$	ResNet-18	32.5	0.01	128	Level 1
			VGG-16_bn	32.4	0.01	256	Level 1
	F1: VGG-16 F2: VGG-16 (w:same)	$D'_1 : (X', Y1)$	ResNet-18	38.5	0.001	256	Level 3
			VGG-16_bn	51.8	0.01	256	Level 3
		$D'_2 : (X', Y2)$	ResNet-18	52.2	0.01	128	Level 2
			VGG-16_bn	52.2	0.01	256	Level 2

Table 6. Non-robust features analysis for STL-10. Optimized hyperparameters are shown besides the test accuracy.

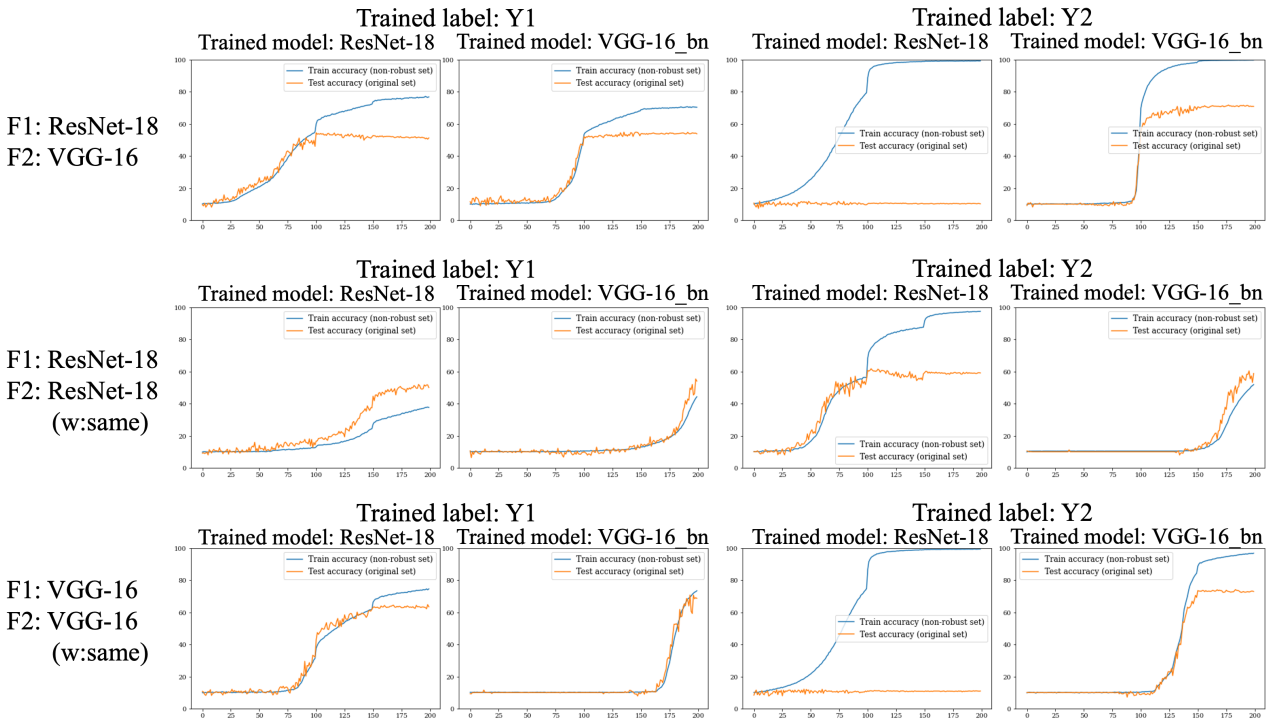


Figure 13. Accuracy curves when models were trained on the constructed non-robust sets (CIFAR-10). Each figure plots training accuracy on the constructed non-robust set (blue line) and test accuracy on the original test set (orange line).

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[4] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS Reproducibility Challenge*, 2019.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[6] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[7] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[8] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

[9] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.