A. QuickShift Segmentation

QuickShift is a mode seeking clustering algorithm proposed by [37]. QuickShift creates segments by repeatedly moving each data point to its closest neighbor point that has higher density calculated by a Parzen Estimator. The Kernel size argument in the QuickShift function controls the width of the gaussian kernel of the estimator. The path of moving points can be seen as a tree that connects data points. Eventually, the algorithm connects all data points into a single tree. To balance between under and over fragmentation of the image, a threshold, τ, is served as a breaking point that limits the length of the branches in the QuickShift trees. The threshold, τ, is the Max distance argument in the QuickShift function. Finally, the pre-processing step of QuickShift projects a given image into a 5D space, including color space (r, g, b) and location (x, y). A hyper-parameter, λ, takes a value between 0 and 1 and serves as a weight assigned to the color space, such that the feature space can be presented as $\{\lambda r, \lambda g, \lambda b, \lambda x, \lambda y\}$.

LIME uses QuickShift for image segmentation where the default Kernel Size is 4, the Max distance is 200, and the threshold τ is 0.2. This combination prevents generating too many image segments. Even-though the image segmentation process is only performed once per image, we would like to point out that the parameter selection does change the explanation results slightly. First, increasing the kernel size increases the computation time while decreasing the number of image segments, making this parameter the major computational bottleneck in image segmentation. Second, extra care should be taken when it comes to low-resolution images, when the image is coarse and the number of image segments are low, because important and unimportant features can easily be merged together, as demonstrated in Fig. 11. From the perspective of explainability, both accuracy and human-readability are needed. This is achieved as long as the important segments are not merged with unimportant ones. This problem can be solved by selecting a small kernel size. In our algorithm, we introduce a user tunable hyper-parameter, called explainabilty length, K, that allows users to decide the number of explainable segments. Human-readability is subjective, so we let the user decide the explainable length, Fig. 12. We see that in Fig. 12, the wall of the castle on the left most side of the image is merged with the sky due to the similarity between colors. In both case, we picked the top 10 segments as explanations, i.e., explainabilty length=10. It is important to note that unlike LIME and other explainability algorithms, the choice of a longer explainabilty length (more segments) does not increase the computational time of our algorithm.

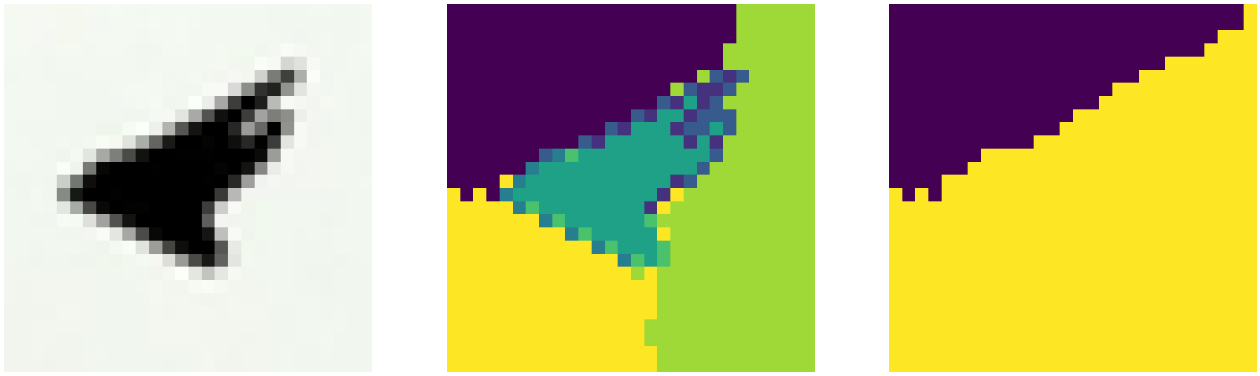
(a) Original image from CIFAR10 (b) Preferable segmentation (c) Under segmentation

Figure 11: Segmentation in low-resolution images.

Deciding the tradeoff between the importance of the color (r,g,b) and spatial components (x,y) of the feature space, is especially important for high resolution images. Take a castle image in the ImgeNet dataset as an example (given in Fig. 13). We choose two different parameter combinations for comparison. The only difference between the two combinations is the λ parameter. For the first combination, we used 0.2 (Fig. 13b), for the second combination, we used 0.8 (Fig. 13c). One can see that using a lower λ prevents details from merging with irrelevant background information. In Fig. 13b and Fig. 13c, the total number of segments are nearly the same (73 and 81) but the explanations have different qualities.

(a) Original image from ImageNet (b) Explainabilty length = 1 (c) Explainabilty length = 2

Figure 12: Relationship between produced explanations and explainabilty length.



(a) Original image from ImageNet (b) λ=0.2 (c) λ=0.8

Figure 13: The effects of λ on the explanations produced.

B. Convergence of Explanations across Adversarial Attacks

As a tool for explainability, efficiency, accuracy and consistency are of top priority. Our experiments show that ℓ_2 PGD attacks with different iterations create explanations similar to ℓ_2 FGM attack. This points to consistency in explanations produced by our algorithm. PGD attack is an iterative version of FGM, while both attacks are subjected to an ℓ_2 norm. Note that the distribution of the attacks can influence the explanation results. This also means that since the attack distributions of the first iteration and later iterations of the PGD attack are nearly identical, the overall explanations remain the same. In Fig. 14, we provide an example from the ImageNet dataset to show the convergence of the attacks and consistency of our explanations. Fig. 14b shows the explanation results for an FGM based algorithm. Fig. 14c and Fig. 14d show the explanation results based on the PGD attack with different number of iterations. They both look exactly the same. This is because the slight changes on the attack distribution for different number of iterations, do not affect the overall density of pixel changes in each segment, thus the final explainability results do not change. This point to stability and consistency of our algorithm. To further explore the stability and consistency of our approach, we can segment the image into much smaller segments, as given in Fig. 14e and Fig. 14f, in this case using 50 times more segments than the previous case and then produce the explanations. In this case, we do see small differences between an explanation produced with a PGM attack with 10 iterations and one based on a PGM attack with 40 iterations. These small differences are caused by small differences in the attack distributions in each segment. While it is interesting to further explore how different types of attacks can lead to more "suitable" explanations, it is important to note that one could explain the outcomes using our algorithm and with both types of attacks. Further, we can conclude that using FGM or PGD attacks in our algorithm satisfies consistency, accuracy

and efficiency conditions for producing explanations.

(a) Original Image from ImageNet (b) ℓ_2 FGM (c) PGD with 10 iterations

(d) PGD with 40 iterations (e) PGD with 10 iterations (f) PGD with 40 iterations

Figure 14: Convergence of explanations for different adversarial attacks and number of segments (Architecture: ResNet34, Dataset: ImageNet).

C. Further Details on the Statistical Analyses given in Subsection 3.2

C.1. Further details on the statistical tests

The Fisher-Pearson coefficient g_1 of a distribution x with a sample size N is calculated using the third moment m_3 and the second moment m_2 of the distribution,

$$g_1 = \frac{m_3}{m_2^{\frac{3}{2}}}, \tag{2}$$

where,

$$m_i = \frac{1}{N}\sum_{n=1}^{N}(x[n] - \bar{x})^i \tag{3}$$

If skewness is 0, the data is perfectly symmetrical, if skewness is positive, then one interprets the distribution as skewed right, if skewness is negative, then the distribution is skewed left. [6] pointed out that there are three levels of symmetricity, a) when skewness is between -0.5 to 0.5, the distribution is "approximately symmetric," b) when skewness is within -1 and -0.5 or 0.5 and +1, the distribution is "moderately skewed," c) when skewness falls out of the mentioned range, then the distribution is highly skewed. The Fisher-Pearson coefficient of all attack magnitudes are shown in Fig. 19. It is seen that the skewness of all attack magnitudes falls within -0.5 an 0.5 showing the strong evidence that the distributions are approximately symmetric.

The t-statistic test is represented as follows,

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p\sqrt{\frac{2}{n}}} \tag{4}$$

where,

$$s_p = \sqrt{\frac{s^2_{X_1} + s^2_{X_2}}{2}} \tag{5}$$

Here $\bar{X}_1$, $\bar{X}_2$ and $s^2_{X_1}$, $s^2_{X_2}$ are the means and variances of the two distributions with size n. The t-statistic can be interpreted as a kind of measurement for the ratio of the "difference between groups" over the "difference within groups." Carrying out pair t-tests on all samples allows us to further be conservative on the similarity on means between the distributions. The results are shown in Table 4. Overall, there is no significant differences between the distributions.

To show the similarity between the distributions produced for a dataset, we also use the one-way ANOVA test on all the samples to show that the means across different distributions are the same. Samples here are defined as intensity vs. frequency distributions for all adversarial test samples created by attacking a model trained on a specific dataset. For CIFAR10, we get the p-value of 0.9, and for a random subset of ImageNet test dataset we get the p-value of 0.94, indicating no significant differences between the distribution means. Similarly, a two-sample location t-test is used to determine if there is a significant difference between two groups where the null hypothesis is the equality of the means. Even-though ANOVA and t-tests are known for being robust on non-normal data, we further performed pair wise Mann–Whitney U test on all pair of distributions to test whether the mean ranks are similar.

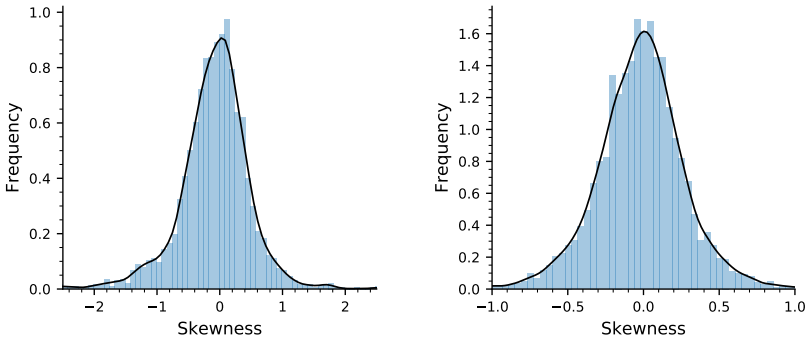
Mann–Whitney U test is a nonparametric test of the null hypothesis that two independent samples selected from population have the same distribution. The statistic U is calculated as following,

$$U_1 = R_1 - \frac{n_1(n_1-1)}{2},\ U_2 = R_2 - \frac{n_2(n_2)}{2} \tag{6}$$

Where subscripts "1" and "2" denote the two distributions being compared. In the case of comparing two distributions "sample 1" and "sample 2." One first combines "sample 1" and "sample 2" together to form an ordered set, and then one assigns ranks to the members of this set. Next, one adds up the ranks for the members of the set coming from "sample 1" and "sample 2" respectively. This is called the rank sum of R_1 and R_2. Once the rank sums are calculated The U statistic of the two distributions (U_1 and U_2) are calculated as above. Finally, the U statistic is determined by the lower value between U_1 and U_2. If U_1 is lower than U_2, then U_1 is the U statistic of the Mann Whitney test between "sample1" and "sample 2"" and vice versa. We further perform the pair-wise Mann–Whitney U test on all pair of distributions to test whether the mean ranks are similar as well. If U is 0, it means that the two distributions are far away from each other where there are no overlaps between them. If the Rank sums are close enough, one can say the two distributions are highly overlapped. Thus, one can say the Mann–Whitney U test is a test comparing the Rank sums (or the mean ranks, calculated by dividing the Rank sums over the size of samples) of two distributions. The smaller values of U_1 and U_2 is the one used when consulting significance tables.

C.2. Quantile-Quantile plot

Quantile-Quantile (Q-Q) plot allows us to show how the quantiles of a distribution deviates from a specified theoretical distribution. The theoretical distribution selected here is the normal distribution. Quantiles are cut points dividing the range of a probability distribution into continuous intervals with equal probabilities. A Q-Q plot is then a scatter-plot showing two sets of quantiles (a sample distribution and a theoretical distribution) against one another. The x-axis are the quantile values of the theoretical distribution while the y-axis are the quantile values of the sample distribution, i.e., the distribution of attack intensities vs. pixel frequencies. One can see that if the quantiles of the sample distribution perfectly match the theoretical quantiles, then one can see all the quantiles located on a straight line. While it is unlikely to have identical distributions that perfectly match the theoretical distribution, one can look at different sections of the Q-Q curves to distinguish the parts that two distributions share similarity and parts that they differ. Compared to a normal distribution, if the sample distribution has heavy or light tails, the Q-Q curve bends at the upper or lower portion based on side of the tails that deviates from the normal distribution. One can say that one purpose of Q-Q plots is to look at the "straightness" of the Q-Q curve. We took a subset that contains 1000 images from both ImageNet and CIFAR10 and plotted the distributions against a normal distribution as given in Fig. 3. It is seen that all attack distributions plotted against the normal distribution have fairly straight lines at the middle portion of the Q-Q curve, while the curve bends at the upper part and the lower part. One can interpret this result as



(a) PGD, LeNet, SpeechCommands (b) PGD, CNN, Sentence Polarity

Figure 15: The Fisher-Pearson coefficient of attack magnitudes vs. frequency distributions.

the attack magnitudes are similar to a normal distribution but differ in a way that the distributions have"heavy tails" thus the upper part of the curve bends "up" and the lower part of the curve bends"down."

C.3. The beta distribution

The beta distribution is a family of distributions defined on the interval [a, b] parametrized by two positive shape parameters, denoted by p and q. The general formula for the probability density function of the beta distribution can be written as,

$$f(x) = \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-x)^{p+q-1}} \tag{7}$$

where,

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{8}$$

The beta distribution is often used to describe different types of data, such as rainfall, traffic and financial data. In this paper, estimate the parameters of a beta distribution for our distributions. The method of moments estimation is employed to calculate the shape parameters, p,q, of the two-parameter beta distribution. As the interval [a, b] is known, the method of moments estimates of p and q are

$$p = \bar{x}(\frac{\bar{x}(1-\bar{x})}{s^2} - 1) \tag{9}$$

$$q = (1-\bar{x})(\frac{\bar{x}(1-\bar{x})}{s^2} - 1) \tag{10}$$

When the interval [a, b] is [0, 1]. This is called the standard beta distribution. Since in most cases the interval [a, b] is not bounded between [0, 1], one can replace $\bar{x}$ with $\frac{\bar{x}-a}{b-a}$ and s^2 with $\frac{s^2}{(b-a)^2}$. Finally the estimated p and q of the beta distribution is listed in Table 3.

C.4. Statistical analysis of distributions for DNNs with text or audio input types

We test the symmetricity of distributions by calculating the Fisher-Pearson coefficient of skewness for LeNet trained on Speech Commands dataset, and a convolutional neural network (CNN) given in [16] on Polarity dataset. The Fisher-Pearson coefficients of the attack magnitudes vs. frequency distributions for all 3 cases are shown in Fig. 15. It is seen that the skewness of all distributions falls within the $[-0.5, 0.5]$ range showing strong evidence that they are approximately symmetric [6].

We perform the two-sample location t-test and Mann-Whitney U test to determine if there is a significant difference between two groups where the null hypothesis is the equality of the means. The results reported in Table 6 indicate no significant difference between the means. Further, the Mann-Whitney U test results indicate that all pairs are similar to each other on the mean ranks. Under the assumption of two distributions having similar shapes, one could further state that Mann-Whitney test can be considered as a test of medians [22]. Since, we have shown that the shapes are similar, we can conclude that there are no significant difference between the medians of the distributions.

Dataset	LeNet, SpeechCommands, PGD		CNN, Sentence Polarity, PGD	
Test	t-test	Mann-Whitney	t-test	Mann-Whitne
p-value	0.30	0.25	0.47	0.42

Table 6: p-values for the mean similarity statistical tests at significance level 0.05.

	LeNet, SpeechCommands, PGD	CNN, Sentence Polarity, PGD
15th Quantile	$(-4.110e-3, -4.049e-3)$	$(-2.753e-1, -2.673e-1)$
25th Quantile	$(-1.150e-3, -1.109e-3)$	$(-1.472e-1, -1.414e-1)$
Mean	$(1.749e-5, 2.245e-5)$	$(-4.165e-3, -2.492e-3)$
Median	$(-4.181e-09, 1.356e-09)$	$(-2.142e-3, -6.219e-4)$
75th Quantile	$(1.145e-3, 1.204e-3)$	$(1.365e-1, 1.421e-1)$
85th Quantile	$(4.153e-3, 4.220e-3)$	$(2.599e-1, 2.677e-1)$

Table 7: Estimations for mean, median, 15th , 25th, 75th and 85th quantiles at 95% confidence level.

	LeNet, SpeechCommands, PGD	CNN, Sentence Polarity, PGD
p	$(5.282e+1, 5.451e+1)$	$(1.322e+1, 1.368e+1)$
q	$(5.144e+1, 5.309e+1)$	$(1.346e+1, 1.393e+1)$

Table 8: Statistical estimations for parameters of beta distribution at 95% confidence level.

Next, to show consistency across distributions for a given model, dataset and attack, we estimate the values of quantiles, means and medians. We do this by estimating the statistics of the distributions and constructing confidences intervals. For each experiment, we estimate the mean, median, 15th, 25th, 75th and 85th quantiles of each attack magnitude vs. frequency distribution for the entire test dataset. The statistical confidence interval estimations at confidence level of 95% are reported in Table 7. Our results show that the confidence intervals have narrow ranges and the estimations are consistent. The estimates for the 15th, 25th, 75th and 85th quantiles indicate a strong symmetricity with respect to the origin in all cases. Another observation is that the confidence interval of the mean and medians are pretty narrow, supporting the results of the t-tests and Mann-Whitney U test. Finally, we can show with high confidence that the distributions consistently follow a beta distribution. The beta distribution is a family of distributions defined by two positive shape parameters, denoted by p and q. The estimated p and q of the beta distribution are reported in Table 8.

D. Explanations and Class Boundaries

Explaining how important features affect the predictions made by the model depends on the set of classes the model was trained to predict. Un-targeted attacks change the prediction label of an input to the label of its closest neighbor. Based on the different datasets that a model may have been trained on, the label changes after attack may be significantly different. For example, given an image of a "Beagle" and a model that is trained on a dataset consisting of labels {Cats and Dogs}, after attacking the model, the label of the image can change from "Dog" to "Cat." But if the same model is trained on a dataset composing of "Beagle, Golden retriever, and Egyptian Cat", the label of the image can change from "Beagle" to "Golden retriever," which is a more granule change. When an image is attacked, the features of the image will be directed to the nearest class with a similar probability distribution in the decision layer. Let's look at an example from ImageNet where the input image is classified as a "convertible" by ResNet34 trained on ImageNet (given in Fig. 16). There are multiple classes such as minivan, sports car, race car etc., under the "car" category in ImageNet. After attacking the model, the label changes from "convertible" to "sports car." This indicates that "sports car" may be the nearest neighbor class to the "convertible" class. If we look at the produced explanations we see that segments including the door are intensely attacked as given in Fig. 16b. The fact is that the model thinks that the doors are the 'most' important features for switching the label from "convertible" to "sports car." Both classes, "convertible" and "sports car," have similar wheels but different doors. In order to fool the model, attacking the wheels is not of top priority, it's the doors that makes the difference between two classes. The fact is that the model thinks that the doors are the most important features for classifying the original image as "convertible" and not "sports car." Both classes, have similar wheels but different doors. In order to fool the model, attacking the wheels is not of top priority, it's the doors that make the difference between two classes. After bluring the segments of interest to the model, i.e.

the door segment—Fig. 16c, and feeding the image to the model, the predicted label changes from "convertible" to "sports car" which proves that the doors are the major features supporting the predictions made by the model. Using adversarial attacks as the force behind producing the explanations helps with finding the important features that are not only globally important to the model (doors are important features of cars, other classes do not have doors similar to cars), but also locally important to the model (within the car class, doors are the important features that make a difference between a convertible and a sports car).

(a) (b) (c)

Figure 16: Left: Original image sample from ImageNet, Middle: The intensely attacked segments. Right: The original image with the explainable parts, i.e., the doors, blurred.

There are also some explainable features that humans hardly understand but models do, these can be called "non-robust features." [35] introduced the concept of robust and non-robust features, where the authors indicated that there are features that humans ignore but the models are sensitive to. They call these the non-robust features. Non-robust features are the features can easily be manipulated by the attacker in order to fool the model. Robust features are features that are both important to the model and also humans and at the same time invincible to small adversarial manipulations.

E. Further Experiment Results

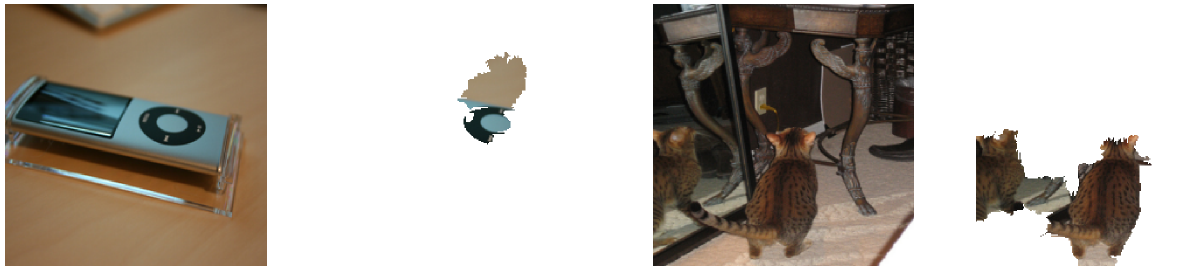
E.1. Explaining an image classification model

Fig. 17 shows two examples of the explanations produced using AXAI for image samples from ImageNet [8] test dataset for a Resnet34 trained on ImageNet training dataset. In the first example, Fig. 17a, the explanation results clearly show that the round control panel on an iPod is an important feature that helps the model identify an IPod in the image. The second example, Fig. 17c, shows how the model recognizes that there are two cats in the image (one is the reflection of the cat in the mirror).

CIFAR10 dataset [15] consists of images of size 32×32 pixels, compared to ImageNet, these images are low-resolution images. Fig. 18 shows the explanations produced by AXAI for sample images from CIFAR10 dataset for an AlexNet image classification model trained on CIFAR10 training dataset. For CIFAR10, our explanations clearly separate the background and capture the target object. The explanation given in Fig. 18b shows that the head of the horse with the leather halter is recognized by the model, and the white fence behind the horse is completely ignored by the model. This indicates that the model is well-trained. Similarly in Fig. 18d the ear and head of deer in the image helps the model to classify the image correctly into the deer class. Images from CIFAR10 dataset are easily explained due to the nature of the dataset with most objects in the images being located in the middle of the image and the lack of noisy background in most images.

E.2. Explaining an object detection model

We present two examples of explanations produced by our algorithm for a YOLOv3 object detection model trained on the SpaceNet Building Dataset [36] to detect buildings in overhead imagery. The produced explanation are clearly focused on areas where buildings are located and ignore empty spaces in the images such as the top left corner of Fig. 19b. Further,



(a) An image of an iPod (b) Explanation (c) An image of a cat (d) Explanation

Figure 17: The explanation results for a ResNet34 image classification model trained on ImageNet.

(a) An image of a horse (b) Explanation (c) An image of a deer (d) Explanation

Figure 18: The explanation results for an AlexNet image classification model trained on CIFAR10.

as seen in Fig. 19d, the roads are ignored and only buildings and their contours affect the predictions made by the object detector.

(a) Image sample 1 (b) Explanation (c) Image sample 2 (d) Explanation

Figure 19: The explanation results for a YOLOv3 object detection model trained on SpaceNet Building Dataset .

E.3. Further details on the speech recognition experiment

The Speech Commands Dataset [38] is an audio dataset of short spoken words, such as “Right,” “Three,” “Bed,” etc. The audio files are converted to spectograms and are used to train a LeNet for a command recognition task. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. Fig. 5a is an example of a spectrogram. The y-axis, the frequency, of the spectograms are presented on a log-scale, the x-axis represent the time-scale, and the color bar shows the magnitude. Fig. 5a is the frequency spectrum of a human speaking the word “Right.” It is seen that in the time interval 0.4s to 1.1s, high magnitude is presented in the spectrum. In other words, the speaker pronounces the word ”Right” around 0.4s to 1.1s into the recorded audio file. This is how one reads a spectrogram. Our explainable solution uses audio files as input, converts them into spectrograms, and then generates the corresponding explanations. So if one feeds

AXAI with an audio file of a human speaking "Right," AXAI first transforms the audio into a spectrogram shown in Fig. 5a, and produces the explanations in Fig. 5b. The explanation will have the exact same scale as the input, and simply masks out the unimportant parts of the spectrogram. To read the explanations, one can refer to the original spectrogram input Fig. 5a and find where the audio is located in the spectrogram (for example looking at the magnitudes), and then look at the corresponding location of the explanations in Fig. 5b.

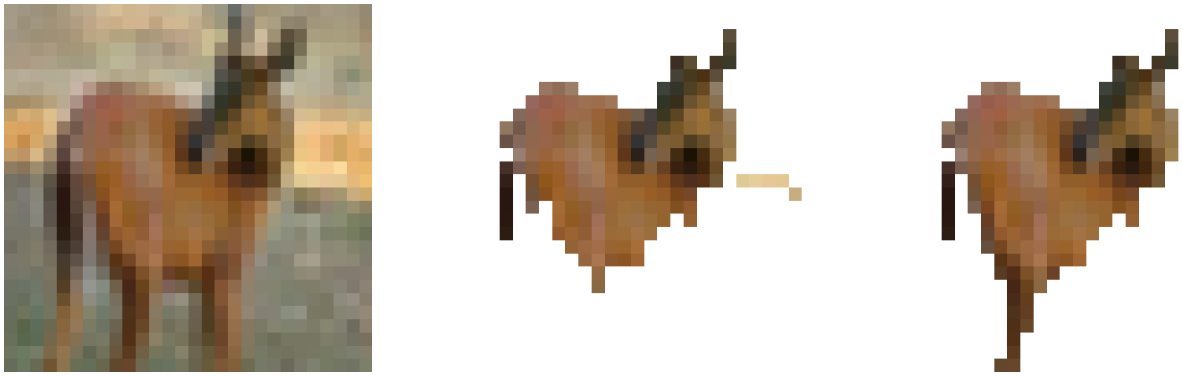
The explanations of two examples are presented in Fig. 5. The spectrogram of the first example "Right" and its explanation are shown in Fig. 5a and Fig. 5b. One can see from Fig. 5a that the spoken word "'Right" appears between 0.4s to 1.1s in the spectrogram of the audio file. If one looks at its corresponding explanation, it is seen that only time-intervals of 0.4s to 0.5s, 0.5s to 0.6s and 1.0s to 1.2s are not masked out by AXAI. This means that these intervals in the audio have great importance for the prediction made by the model. if we look back at Fig. 5a, one then realizes that the explanation shows that the first few and the last few seconds of the spoken word "Right" are important to the model, and the middle part is not. Why is that? The neighboring class of "Right" is "Five." "Right" and "Five" differ in how "R" & "F" and "t" & "ve" are pronounced. The middle part of "Five" and "Right" is highly similar and does not affect the model's prediction on deciding whether the spoken word is "Five" or "Right." The second example is "Three." As seen in the spectrogram, Fig. 5c, "Three" is expressed around the time-interval 1.4s to 2.2s in the spectrogram of the audio file. The corresponding explanation is shown in Fig. 5d. The explanation masks out almost everywhere except 1.4s to 1.6s and a small part in 1.6s to 1.7s and 1.9s to 2.2s. Now, let's look at the original spectrgram of "Three" and understand what the explanation means. Since The explanation highlights 1.4s to 1.6s, which is the first few seconds of the spoken word. To understand why, one can learn that if we attack the model, then "Three" is miss-classified as "Tree." This indicates that the model has learned to recognize "Three" and not "Three" by learning the difference between "Thr" and "Tr." The explanation tells us that the first few seconds of the audio are important (the utterance of "Thr").

E.4. Ablation study

If a feature or a group of features is important to a model, then completely removing those features from the input would decrease the probability of a correct prediction. Accordingly, we performed an ablation study confirming that the explanations produced by AXAI contain important features. This ablation method can be used to test the accuracy of an explainability solution. If the generated explanation is faithful to the model, then removing the explanations would decrease the accuracy of the predictions. In this section, we demonstrate a simple experiment to validate our algorithm. Our experiment is performed as follows: 1) Generate the explanation of a targeted image X via AXAI, where the explanation length $K = 10$ is selected in this experiment, 2) Blur the top 5 explanations/segments of the targeted image according to the produced explanations, feed the modified image to the model and obtain its label, 3) repeat this process throughout the test dataset 4) Calculate the total decrease in accuracy. We use a ResNet34 training on ImageNet for this experiment and report the results for the entire ImageNet test dataset. Our results show that the prediction accuracy of the DNN decreases to $\%43$ after blurring the top 5 explanation/segments. To further investigate, instead of blurring the top 5 explanations, we blur only the 6th to10th explanations. This results in a $\%22$ drop in total accuracy. Hence, we can conclude 1) AXAI generates faithful explanations so that blurring the top explanations (the 1st-5th explanations) lead to a strong decrease in model prediction accuracy, and 2) AXAI generates faithful explanations in order of importance, i.e., the generated 6th to 10th explanations are also important to the model but their influence on model predictions is relatively less than the first 5 generated explanations.

E.5. AXAI explanations for a robust model trained with adversarial training

In this subsection, we compare the explanations produced for a robust model to explanations produced for a non-robust model. In our experiment, a robust model is a model trained on an adversarial dataset in addition to the training dataset so that the final trained model is more robust against adversarial attacks. Hypothetically, a robust model should focus more on robust important input features when making predictions. We have trained a non-robust AlexNet and a robust AlexNet on CIFAR10 and produced the explanations using AXAI for test inputs. Fig. 20 shows the AXAI produced explanations for the DNN given a sample input. It is seen that a small part of the background is included in the explanations produced for the non-robust AlexNet. However, the AXAI generated explanations for the robust model includes only the important features pertaining to the object in the image. In addition, the leg of the deer is now included in the explanations as well. It is concluded that explanations produced for the robust DNN are sharper, clearer and more robust than the ones generated for the regularly trained DNN.



(a) Image of a deer (b) Non-robust AlexNet (c) Robust AlexNet

Figure 20: Comparison between explanations produced by AXAI ($K = 10$) for AlexNet trained on CIFAR10 with and without adversarial training

F. Benchmark Tests

We test our algorithm against LIME and SHAP. We use "Gradient Explainer" in SHAP, which integrates the f Integrated gradients algorithm with SHAP. Fig. 21 shows some sample comparisons among the 3 algorithms for 3 cases: 1) AlexNet trained on CIFAR10, 2) ResNet34 trained on ImageNet, 3) VGG16 trained on CIFAR100. PGDM with 20 iterations is used in our algorithm. For ImageNet, explanations for a sample test picture belonging to "Egyptian cat" are shown in Fig. 21a, Fig. 21b, and Fig. 21c. One can see the similarity between the explanations. The explanations produced by the 3 algorithms focused on the upper left of the image which contains the eyes of the "Egyptian cat." Both LIME and our algorithms point to the same segment as explanations. SHAP (Gradient Explainer) locates pixels of interest. The important pixels shown in this case aligns with the results of LIME and AXAI. Since the default image segmentation parameters LIME chooses do not allow for a suitable number of segments for explanation for CIFAR10 and CIFAR100 due to the resolutions of images, we lowered the Kernel size parameter to 1. The default Kernel size parameters LIME uses for QuickShift is too large for low-resolution images. As we mentioned before, this leads to a few very large segments in the image and neglects all the granular details in the image. For CIFAR10, both our approach and LIME capture the upper portion of the head of the horse including the ears and eyes (Fig. 21d, Fig. 21e). The results of SHAP point out the important pixels located on the head, the nose and some pixels in the background (Fig. 21f). For CIFAR100, the explanations produced by the 3 algorithms are once again highly similar (Fig. 21g, Fig. 21h, and Fig. 21i). One can see that in many cases, pixel explanations do not serve as the best solution. Without the segments, it is hard to grasp the meaning behind explanations, this is because the human brain tends to comprehend image segments better than individual pixels.

G. Additional Examples

In this section we provide additional explanation results of the Alexnet image classification model,the VGG16 image classification model, the LeNet speech recognition model, and the sentence classification model in Fig. 22, Fig. 23, Fig. 24, and Fig. 25. For the ResNet34 image classification model, the results are shown in the main paper, Fig. 10.

(a) (b) (c)

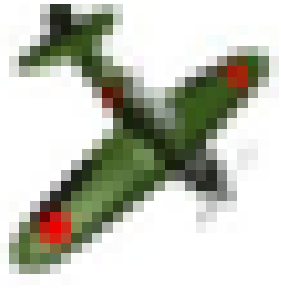
(d) (e) (f)

(g) (h) (i)

Figure 21: Comparisons between our adversarial explainability approach (Left Column), LIME (Middle Column), and SHAP (Right Column). LIME parameters: number of perturbed samples $N = 1000$, number of features $M = 5$. First row: ResNet34 trained on ImageNet, Second row: AlexNet trained on CIFAR10, Third row: VGG16 trained on CIFAR100

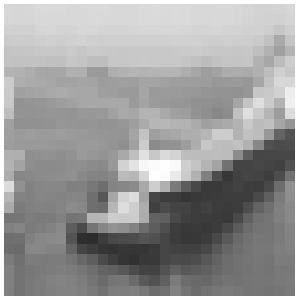
(a) An image of a car

(b) Explanation



(c) An image of a plane

(d) Explanation



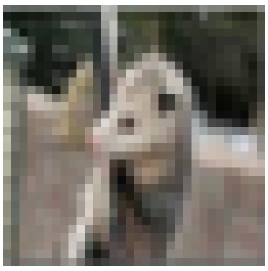
(e) An image of a ship

(f) Explanation

(g) An image of a bird

(h) Explanation

Figure 22: Additional explanation results for an AlexNet image classification model trained on CIFAR10.



(a) An image of a possum

(b) Explanation

(c) An image of a squirrel

(d) Explanation

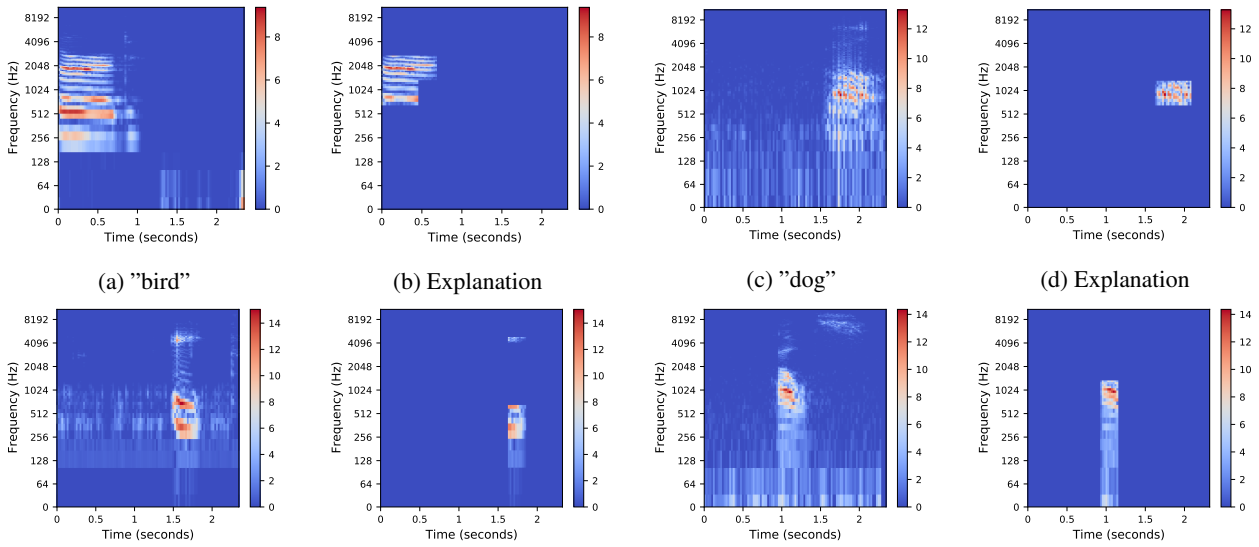
(e) An image of a road

(f) Explanation

(g) An image of a wardrobe

(h) Explanation

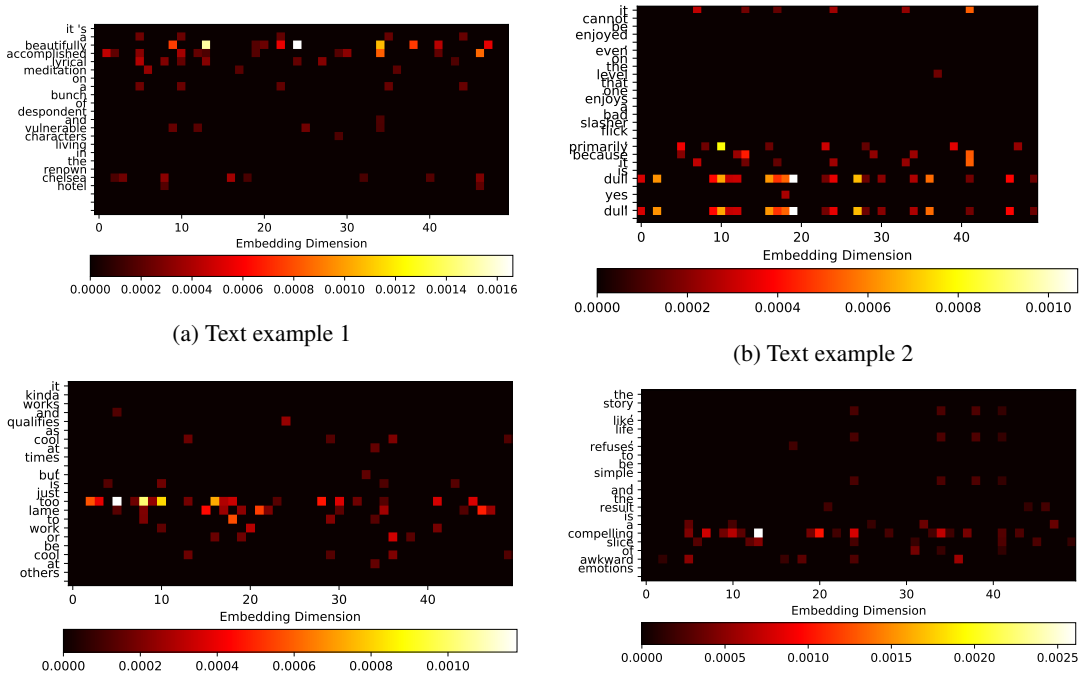
Figure 23: Additional explanation results for a VGG16 image classification model trained on CIFAR100.



(a) ”bird” (b) Explanation (c) ”dog” (d) Explanation

(e) ”eight” (f) Explanation (g) ”house” (h) Explanation

Figure 24: Additional explanation results for a LeNet speech recognition model trained on the Speech Commands Dataset.



(a) Text example 1 (b) Text example 2

(c) Text example 3 (d) Text example 3

Figure 25: Additional explanation results for a sentence classification model trained on the Sentence Polarity Dataset.