A. Implementation Details

A.1. Datasets and Classifiers

MNIST (LeCun & Cortes, 2010) The classifier consists of two 6x6 CNN layers with a stride of 2, followed by a 256-unit fully connected layer, a dropout layer with $p = 0.5$, and the 10 output neurons. As shown in (Springenberg et al., 2015) the stride>1 CNN achieved comparable performance with pooling layers. The classifier was trained for 50 epochs and achieve a test accuracy of 99.3%.

Street-View House Numbers (SVHN) (Netzer et al., 2011) We tested our models on the cropped version of SVHN and used the same model architecture with that of MNIST and achieved a test accuracy of 90.3% after 50 epochs of training.

CIFAR10 (Krizhevsky, 2009) We trained a classifier consist of 4 repetitive units, with each unit constructed by two 3x3 CNN layers and a 2x2 average pooling layer, with each CNN layer followed by a batch normalization layer. The classifer achieved 87.8% test accuracy after 100 epochs of training.

apple2orange (Zhu et al., 2017) We trained a classifier taking the original 256x256 image as input. The classifier was constructed by adding a global average pooling layer on top of MobileNet (Howard et al., 2017), and then followed by a dense layer of 1024 neurons and a dropout layer of $p = 0.5$ before the output neurons. The classifier was trained for 50 epochs and achieve a test accuracy of 87.7%.

BAM (Benchmarking Attribution Methods) (Yang & Kim, 2019) BAM dataset was originally designed for evaluating explainable models in one-vs-one settings. It was constructed by positioning objects from MS COCO dataset (Lin et al., 2014) on to background images from miniplaces dataset (Zhou et al., 2018). Here we treated different combinations of objects and backgrounds as different classes and considered four classes in our experiment: $(pizza, bedroom)$, $(pizza, bamboo forest)$, $(stopsign, bedroom)$, $(stopsign, bamboo forest)$.

Similar to the apple2orange dataset, we constructed a classifier consist of MobileNet (Howard et al., 2017), a dense layer, a global average pooling layer, and a dropout layer and trained for 50 epochs to achieve 96.6% test accuracy.

A.2. Baseline Generation with GANMEX

Our baseline generation process is based on StarGAN (Choi et al., 2017). We used the Tensorflow-GAN implementation (`https://github.com/tensorflow/gan`) and made the following two modifications (Equation 9):

1. The class discriminator D_{cls} is replace by the target classifier S to be explained.
2. A similarity loss $\mathcal{L}_{sim}$ is added to the training objective function.

We train the GANMEX model for 100k steps for the MNIST and apple2orange datasets, 300k steps for the SVHN dataset, and 400k steps for the CIFAR10 dataset. Only the train split is used for training, and the attribution results and evaluation were done on the test split of the dataset.

We released our source code at `https://github.com/pinjutien/GANMEX`.

A.3. Attribution Methods

We used DeepExplain (`https://github.com/marcoancona/DeepExplain`) for generating saliency maps with IG, DeepLIFT, and Occlusion. We modified the code base to use the score delta ($S_{c_o} - S_{c_t}$) instead of the original class score (S_{c_t}) and allowing replacing the zero baseline (see Section 2.1) by custom baselines from GANMEX and MDTS. EG was separately implemented according to the formulation in (Erion et al., 2019). We set the number of sampling steps to 200 for both IG and EG, and used Occlusion-1 that only perturb the pixel itself (as supposed to perturbing the whole neighboring patch of pixels).

The DeepSHAP saliency maps were calculated using SHAP (https://github.com/slundberg/shap). We made similar modification to replace the original class score by the score delta and feed in the custom baseline instances.

In all saliency maps shown in the paper, blue color in indicates positive values and red color indicates negative values. We skipped Occlusion for large images (apple2orange) and also skipped SHAP for full dataset evaluations due to the computation resource constraints.

B. Baseline Distance Analysis

To measure how various baseline selection approaches satisfy the minimum distance requirements in Equation 1, we calculated $D(x, \tilde{x}) = \|x - \tilde{x}\|$ for (1) GANMEX, (2) MDTS, (3) a randomly selected sample in the target class as baseline and (4) zero baseline. GANMEX was on-par with MDTS on the MNIST dataset, but on SVHN and CIFAR10 dataset that have more degrees of freedom (object size, color, orientation, background, ...), GANMEX was significantly better in identifying minimum distance baselines compared to the in-sample search. The high dataset complexity of was supported by the average intra-class distance, the average distance between any two instances within the same class, which was higher than that of MNIST. Note that the resulting sample to baseline distance $D(x, \tilde{x}_{\text{GANMEX}})$ is much higher in MNIST than in SVHN, because there were more boundary values (0s and 1s) in MNIST.

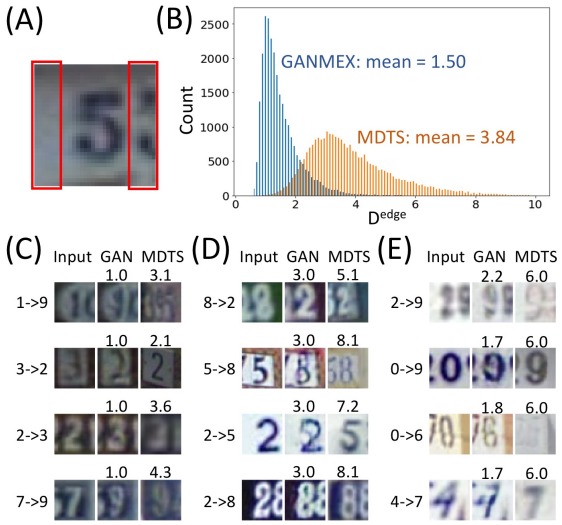

Figure 6. (A) Vertical edge area in an SVHN image. (B) Histogram of sample to baseline distance ($D^{\text{edge}}(x, \tilde{x})$) in the vertical edge area. (C-E) Samples comparing GANMEX and MDTS baselines with $D^{\text{edge}}(x, \tilde{x})$ indicated on the top of the baseline images. (C) Easy cases for GANMEX ($D^{\text{edge}}(x, \tilde{x}) \approx 1$). (D) Difficult cases for GANMEX ($D^{\text{edge}}(x, \tilde{x}) \approx 3$). (E) Difficult cases for GANMEX ($D^{\text{edge}}(x, \tilde{x}) \approx 6$).

We further evaluated the similarity distance on the vertical edge area ($D^{\text{edge}}(x, \tilde{x})$) of the SVHN images (Figure6.A). Empirically, we observed that the digits of interest were rarely present in the vertical edge, and therefore, we would expect a closest baseline choice will lead to minimal changes in the edge area $D^{\text{edge}}(x, \tilde{x})$ under the minimum distance requirements. We provided a histogram in Figure 6.B for comparing the distribution of $D^{\text{edge}}(x, \tilde{x})$ for MDTS and GANMEX, and we presented sampled success/failure cases in Figure 6.C-E. Overall, GANMEX leads to baselines that are closer to the original samples.

C. Additional Metrics

C.1. Perturbation-based evaluation

We followed the perturbation-based evaluation suggested by (Bach et al., 2015) that flips input features starting from the ones with the highest saliency values and evaluates the cumulative impacts on the score delta $S_{c_o} - S_{c_t}$ as proposed by (Shrikumar et al., 2017). Flipping a feature means to provide with a value of $1 - x$ where x is its original value, assuming all features are normalized to $x \in [0, 1]$. A wanted behavior from the attribution map is that the score delta will decrease as rapidly as possible as we flip the features one by one. We provide in Figure 7.A-C the perturbation curves for both MNIST, SVHN and CIFAR10, plotting the score delta as a function of the number of flipped features. It is

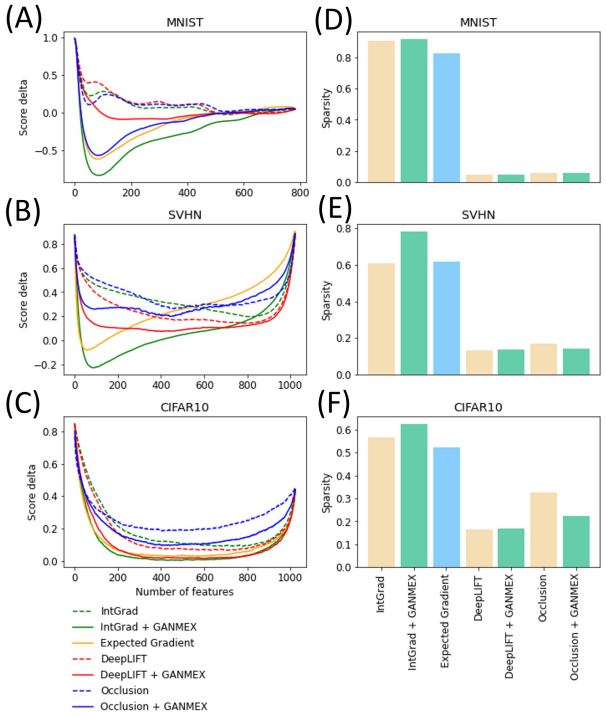

Figure 7. (A-C) Perturbation-based evaluation plots for MNIST and SVHN, respectively. The dashed lines represent the non-class-targeted baselines and the solid lines represent class-targeted baselines. (D-F) Gini indices, with the yellow bars represent saliency maps with zero baselines and the green bars represent that of GANMEX baselines.

Table 2. Baseline distance analysis comparing the average intra-class distance D_{intra} and the average inter-class distance D_{inter}, with the average distance from the instance to the baseline input generated by GANMEX (GAN), MDTS, random selection (RAND), and zero inputs (Zero).

		Data Size		Avg. Distance		Baseline Distance			
	Dimension	Train	Test	D_{intra}	D_{inter}	GAN	MDTS	Rand	Zero
MNIST	784 (28 x 28 x 1)	60,000	10,000	8.96	10.32	**7.17**	7.18	10.32	9.28
SVHN	3072 (32 x 32 x 3)	73,257	26,032	14.42	14.48	**3.42**	5.94	15.44	26.52
CIFAR10	3072 (32 x 32 x 3)	50,000	10,000	18.28	19.06	**6.89**	10.54	19.01	29.04

clear that that by using a GANMEX baseline rather than the alternative zero baseline, the descent of the curve is much faster, meaning that we successfully capture the most important features using GANMEX. This holds true for all attribution methods. As a side note, notice that in SVHN when we flip all features the score delta goes back to where it was in the beginning as opposed to going down to zero. This is due to the fact that once all features are flipped, we are back to having the same digit as before.

Based on the perturbation curves, we evaluated AOPC_L for the different baseline choices. AOPC_L measures the area over the perturbation curve within the first L perturbation steps (Samek et al., 2017; Tomsett et al., 2019)). One potential downside of AOPC_L is that the metric is only sensitive to the top L features in the saliency map and not the rest. Therefore, in addition to $\text{AOPC}_{L=100}$, we calculated AOPC_{all}, the area over the perturbation curve across all feature. The gradient family of IG and EG generally outperformed DeepLIFT and occlusion on the AOPC metrics, with IG+GANMEX performed the best overall (Table 3).

C.2. Remove And Retrain (ROAR)

ROAR evaluates model explanations by removing features in a similar manner as in AOPC, but instead of directly measuring how the predictions of the same model deteriorates, ROAR retrains models with all images having the same number of pixels removed and measures the performance of the new models. The retraining process of ROAR ensures that the models are evaluated on datasets with the same distribution where they were trained. However, unlike AOPC which measures the particular model instance, ROAR measures on a series of retrained models which is more of an indicator of how the explanation method identifies the key features from the dataset.

For evaluating one-vs-one explanations with ROAR, we divide the classes in to pairs (c_i, c_i'), and for sample x and x' with the labels $y = c_i$ and $y' = c_i'$, we remove the features according to $A_{S,c_i \to c_i'}(x)$ and $A_{S,c_i' \to c_i}(x')$, respectively. A classifier is then trained for each pair of labels (c_i, c_i') to measure the effectiveness of the saliency maps. We reported the Area Over the ROAR Curve (AORoarC) in Table 3, showing that both GANMEX and MDTS outperformed the results from expected gradient and zero baselines, with Occlusion + GANMEX performed the best overall.

C.3. Sparsity

The sparsity is a desirable property for one-vs-one attribution. We expect a good one-vs-one explanation to highlight only the differentiating features for distinguishing between two classes. Compared with one-vs-all saliency maps, one-vs-one saliency maps should highlight a smaller subset of features, especially when the target classes are similar to the original classes. Therefore, one would expect a more sparse one-vs-one saliency map is more likely to be correct.

We calculated the Gini Index representing the sparsity of the saliency maps as proposed by Chalasani et al. (2018), where a larger score means sparser saliency map, which is a desired property. As shown in Table 3, saliency maps generated by the gradient family generally have higher Gini indices, and therefore are more sparse compared to the other two groups of saliency methods - DeepLIFT and Occlusion. IG+GANMEX and IG+MDTS were the best performers overall, whereas EG, on the other hand, consistently underperformed other gradient based methods on all datasets. IG with zero baseline did achieve sparsity comparable with other top methods. We suspected that the sparseness of zero baseline attribution was benefited from incorrectly hiding some key features, as shown in Figure 2.

C.4. Faithfulness and Monotonicity

We measured the faithfulness reported by (Alvarez-Melis & Jaakkola, 2018; Tomsett et al., 2019) and monotonicity suggested by (phi Nguyen & Martínez, 2020). Instead of measuring the cumulative effect of alternating a set of features, both faithfulness and monotonicity measure the impacts on alternating single features. We found that EG and Occlusion+zero baseline are the best performers on those two metrics (Figure 2).

Table 3. Additional metrics for attribution methods using the zero baseline (Zero), MDTS, and GANMEX (GAN).

		Integrated Gradient			EG	DeepLIFT			Occlusion		
Metrics	Dataset	Zero	MDTS	GAN		Zero	MDTS	GAN	Zero	MDTS	GAN
$AOPC_{100}$	MIST	0.614	1.249	**1.421**	1.260	0.505	0.639	0.724	0.705	1.050	1.221
	SVHN	0.346	0.861	**0.921**	0.878	0.377	0.634	0.621	0.317	0.547	0.549
	CIFAR10	0.298	0.485	0.494	**0.516**	0.323	0.464	0.451	0.441	0.492	0.440
$AOPC_{all}$	MIST	0.889	1.098	**1.263**	1.13	0.859	0.933	0.992	0.877	1.019	1.114
	SVHN	0.564	**0.844**	0.822	0.626	0.649	0.750	0.751	0.528	0.586	0.585
	CIFAR10	0.696	0.788	**0.808**	0.789	0.726	0.780	0.794	0.628	0.694	0.706
$AORoarC_{all}$	MIST	0.176	0.295	0.249	0.211	0.209	0.294	0.293	0.196	0.303	**0.327**
sparsity	MIST	0.909	0.911	**0.919**	0.827	0.047	0.047	0.046	0.062	0.058	0.058
	SVHN	0.606	0.713	**0.783**	0.615	0.131	0.133	0.139	0.168	0.139	0.144
	CIFAR10	0.565	**0.639**	0.626	0.522	0.164	0.171	0.169	0.325	0.260	0.223
faithfulness	MIST	0.182	0.224	0.280	**0.407**	0.075	0.003	0.031	0.257	0.254	0.291
	SVHN	0.017	0.265	0.270	**0.548**	-0.041	0.075	0.017	0.007	0.306	0.243
	CIFAR10	0.005	0.028	0.027	0.054	0.003	0.017	0.017	**0.288**	0.285	0.225
monotonicity	MIST	0.118	0.196	0.264	**0.357**	0.087	0.150	0.206	0.244	0.239	0.280
	SVHN	0.129	0.212	0.248	**0.340**	0.095	0.150	0.182	0.057	0.210	0.211
	CIFAR10	0.008	0.050	0.042	0.058	0.004	0.044	0.033	**0.175**	0.140	0.087
inv. localization	SVHN	0.268	0.128	0.113	0.217	0.268	0.156	0.123	0.268	0.144	**0.100**

C.5. Inverse Localization Metrics for SVHN Dataset

Lastly, we applied the inverse localization metric described in Section 4.1 to the SVHN dataset. We observed that the digits mostly have < 1 aspect ratios, meaning that their widths are smaller then their heights. As a results, the areas at the two vertical edges are generally not covered by the primary numbers, and instead, they usually show the background or the neighboring numbers. Therefore, we can reasonably expect the saliency map sensitivity to be location in the center area (area excluding the vertical edges), and not the vertical edge area (Figure 6.A).

Based on this observation, we define the inverse localization metric for SVHN to be $L(A(x)) = (\frac{1}{card(S_{\text{edge}})} \sum_{i \in S_{\text{edge}}} |A_i(x)|) / (\frac{1}{card(S_{\text{center}})} \sum_{i \in S_{\text{center}}} |A_i(x)|)$, which calculates the ratio of the average absolute sensitivity between the vertical edge area and the center area. S_{center} and S_{edge} represent the feature set in the center area and the edge area, respectively. While S_{center} is just an outer bound of the distinguishing feature set, we still expect such metric provide meaningful evaluation for the attribution methods.

As shown in Table 3, we see a consistent trend of the saliency maps with GANMEX baselines being more localized (lower inverse localization) compared to MDTS baselines across all attribution methods, and EG and the zero baselines generally lead to the worst results. The saliency maps produced by occlusion+GANMEX was the most localized among all the methods tested.

To summarize, we evaluated multiple attribution methods and baseline combinations with metrics that assess different properties of the saliency maps. While there was inconsistency between different metrics as observed by Tomsett et al. (2019), we see a strong trend of class-targeted baselines, especially GANMEX, leading to more desirable attributions. Most importantly, the only ground-truth driven metric - inverse localization has showed that GANMEX significantly improved the attributions.

D. Hyper-parameter Analysis

We tested how the generated baselines change with respect to the hyperparemeters in the GANMEX loss function. The hyper-parameters, λ^f_{cls}, λ_{rec}, and λ_{sim}, presented in Equation 9 control the degrees of the classification loss, reconstruction loss, and similarity loss, respectively. We performed the hyper-parameter scan on the SVHN dataset as it has enough complex and yet simple enough for visually assessing the attribution.

Classification Loss (λ^f_{cls}) Low classification loss tended to make some transformation unsuccessful, and high classification loss introduced additional noise that make the images unrealistic.

Similarity Loss (λ_{rec}) Similarity loss is the key component for minimum distance optimization. As we have shown in Section 5 and Figure 5.B, at zero similarity loss, the generator is only constraint by the reconstruction loss and can lead to incorrect font colors and background. High similarity loss, on the other hand, makes the baselines to be

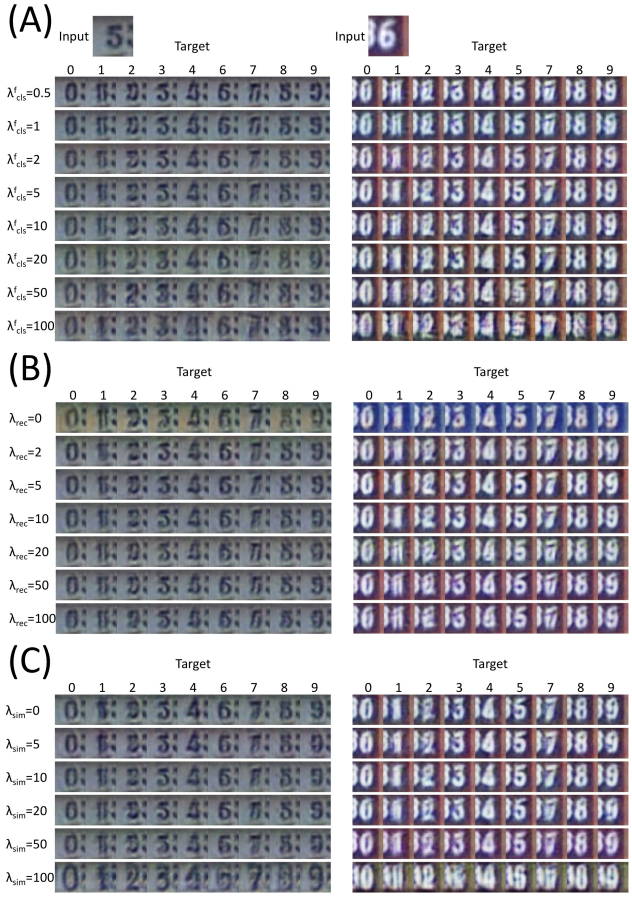

Figure 8. GANMEX baselines generated with various weights for the (A) classification loss, (B) similarity loss, and (C) reconstruction loss.

too similar to the original images.

Reconstruction Loss (λ_{sim}) As we have mentioned in Section 5 and Figure 5.B, reconstruction loss is not required for GANMEX, but it slightly helps GAN to converged. In contrast, high reconstruction loss can lead to incorrect outputs.

E. Compute Time Analysis

In Table 4, we measure the GANMEX compute time compared with various attribution methods. While the GAN component takes 5-23 hours to train depending on the datasets, the inference step only requires one single forward operation, and the compute time (MNIST: 9.3 ms, SVHN: 15.4 ms, CIFAR10: 96.0 ms) is at the same order with IG and Occlusion. More experiment details are provided in the caption of Table 4.

F. Intuitions Behind the Minimum Distance Requirements

Here we present the intuitions behind the baseline selection criteria from Section 3.1 using a simplified formulation. Assuming a transformation ρ projecting from a set of high-level concept variables V to a sample x, with $x = \rho(V)$, and we can separate V into three groups $V = \{V^{\text{dis}}, V^{\text{con}}, V^{\text{irr}}\}$. Here, V^{dis} are the discriminating variables that leads to the model decision, the color of the fruits in our apply/orange dataset for example; V^{con} are the contingent variables that are independent to the model decision but correlate with how the discriminating variables are presented (eg. sizes and locations of the fruits); V^{irr} are the irrelevant variables that are both independent to the model decision and uncorrelated to the presentations of the discriminating variables (eg. background colors of the images). The expected one-vs-all explanation under the formulation would be

$$\begin{aligned} A_{c_o}(x) &= A_{c_o}(\rho(V)) \\ &= A(\rho(V^{\text{dis}}_{c_o}, V^{\text{con}}, V^{\text{irr}})) \\ &= \alpha_{c_o}(V^{\text{dis}}_{c_o}, V^{\text{con}}) \end{aligned} \tag{10}$$

with α being a transformation from the underlying concept variables to the explanation. Here we assumed a correct mapping α should be independent of V^{irr} because the variable set has no impact on the discriminating variables themselves or how discriminating variables are presented.

Now if we apply the concept of one-vs-one, we expect a one-vs-one explanation to produce

$$A_{c_0 \to c_t}(x, B_{c_t}(x)) = \alpha_{c_0 \to c_t}(V^{\text{dis}}_{c_o}, V^{\text{dis}}_{c_t}, V^{\text{con}}) \tag{11}$$

where $A_{c_0 \to c_t}$ and B_{c_t} were defined in Section 3, and $V^{\text{dis}}_{c_o}$ and $V^{\text{dis}}_{c_t}$ are the discriminating variables for class c_o and c_t. In the apple to orange example, $V^{\text{dis}}_{c_o}$ would represent the red color, and $V^{\text{dis}}_{c_t}$ would represent the orange color.

The baseline generation function B_{c_t} maps the original sample $x = \rho(V^{\text{dis}}_{c_o}, V^{\text{con}}, V^{\text{irr}})$ to the baseline sample $B_{c_t}(x) = \rho(\tilde{V}^{\text{dis}}, \tilde{V}^{\text{con}}, \tilde{V}^{\text{irr}})$, where $\tilde{V}^{\text{dis}}$, $\tilde{V}^{\text{con}}$ and $\tilde{V}^{\text{irr}}$ are the concept variables for the generated baseline input. We can explicitly write out

$$\begin{aligned} A_{c_0 \to c_t}(x, B_{c_t}(x)) = A_{c_0 \to c_t}(&\rho(V^{\text{dis}}, V^{\text{con}}, V^{\text{irr}}), \\ &\rho(\tilde{V}^{\text{dis}}, \tilde{V}^{\text{con}}, \tilde{V}^{\text{irr}})) \end{aligned} \tag{12}$$

Although $A_{c_0 \to c_t}$ could be designed to be independent of $\tilde{V}^{\text{con}}$, V^{irr}, and $\tilde{V}^{\text{irr}}$ to make Equation 12 satisfy the

Table 4. Run Time Analysis comparing the attribution computation time for IG, EG, DeepLIFT (DL) and Occlusion (Occ), as well as the baseline generation time for EG, GANMEX (GAN), and MDTS. The computation was performed on a single Tesla V100 GPU, and the compute time was measure in seconds on calculation over all samples for the dataset, and the baseline generation time is the additional compute time on top of the attribution methods. (§) We selected the same sampling number for IG and EG, and the baseline selection time of EG was estimated by the complexity difference of EG and IG. (†) The attribution inference time for CIFAR10 was measured in 10 separate batches due to the memory constraint. (‡) MDTS search was performed on CPU instead of GPU.

			Attribution Inference				Baseline Generation			GAN Training	
Dataset	Size	Dim.	IG	EG	DL	Occ	EG§	GAN	MDTS‡	Steps	t (hour)
MNIST	10k	784	43.5	64.1	0.6	75.7	20.6	92.5	850.8	100k	5.2
SVHN	26k	3072	557.3	658.9	1.9	4917.0	101.6	399.8	3674.4	300k	18.2
CIFAR10†	10k	3072	239.4	294.8	30.0	1532.1	55.3	959.7	1085.8	400k	23.8

form of Equation 11, anti-symmetric attribution methods with $A_{c_0 \to c_t}(x, B_{c_t}(x)) = -A_{c_t \to c_0}(B_{c_t}(x), x)$ such as IG would not satisfy such requirements. Alternatively, we can require B_{c_t} to satisfy the following.

$$\tilde{V}^{\text{dis}} = V^{\text{dis}}_{c_t} \quad (13)$$

$$\tilde{V}^{\text{con}} = V^{\text{con}} \quad (14)$$

$$\tilde{V}^{\text{irr}} = V^{\text{irr}} \quad (15)$$

Equation 13 requires the baseline to belong to the target class c_t, and this implies that a class-targeted one-vs-one baseline is required for correct one-vs-one explanations.

Equation 14 and Equation 15 combined have led to the closest input requirements described in Section 3.1. Assuming a smooth transformation (ρ), minimizing the distance of $\|x - B_{c_t}(x)\|$ provides an effective way of ensuring Equation 14 and Equation 15. Going back to the apple/orange example, a baseline satisfying Equation 13-15 for an apple image input would be an image with an orange fruit of the same size, at the same location, with the same background to the original input, and all of the above can be achieved by the minimum distance sample described in Section 3.1.

Non-class-targeted baselines, such as zero baselines, max value baselines, or blurred images clearly violate Equation 13-15. Specifically, all three non-class-targeted baselines mentioned here violates all of Equation 13-15, and therefore, they do not lead to correct one-vs-one attributions. This can be easily spotted in the examples in Figure 2, 4 and tested in the BAM dataset evaluations (Table 1) and by the sanity checks in Figure 3.

G. Additional Figures

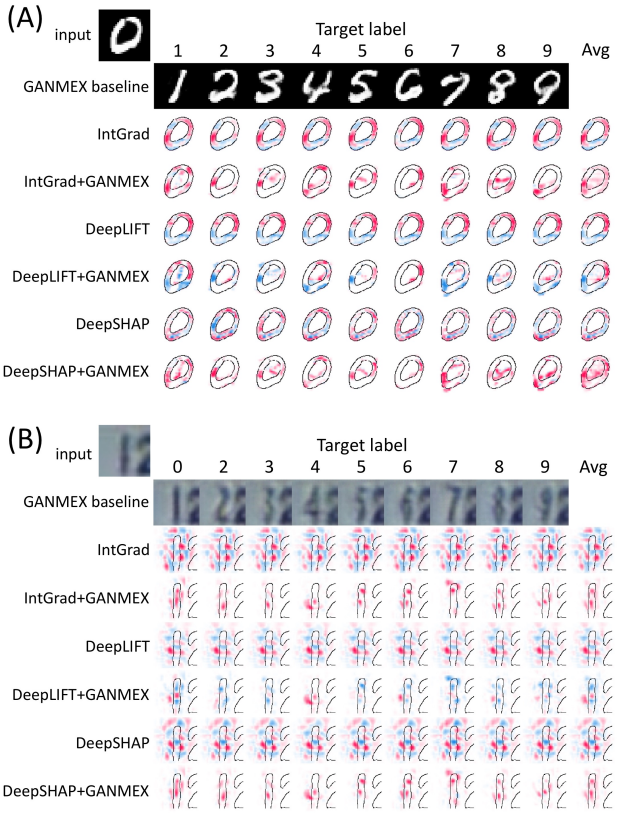

Figure 9. One-vs-one saliency maps using class-targeted baselines (GANMEX) vs non-class-targeted baselines (zero baselines). One-vs-one saliency maps generated using zero baselines show almost the same attributions regardless of the target class, making the one-vs-one saliency maps (columns with target labels) similar to the one-vs-all saliency maps (the "Avg" columns that show the averaged saliency maps over all target classes). GANMEX baselines corrected the behavior for both IG, DeepLIFT and DeepSHAP, and produced different attributions depending on the target classes.

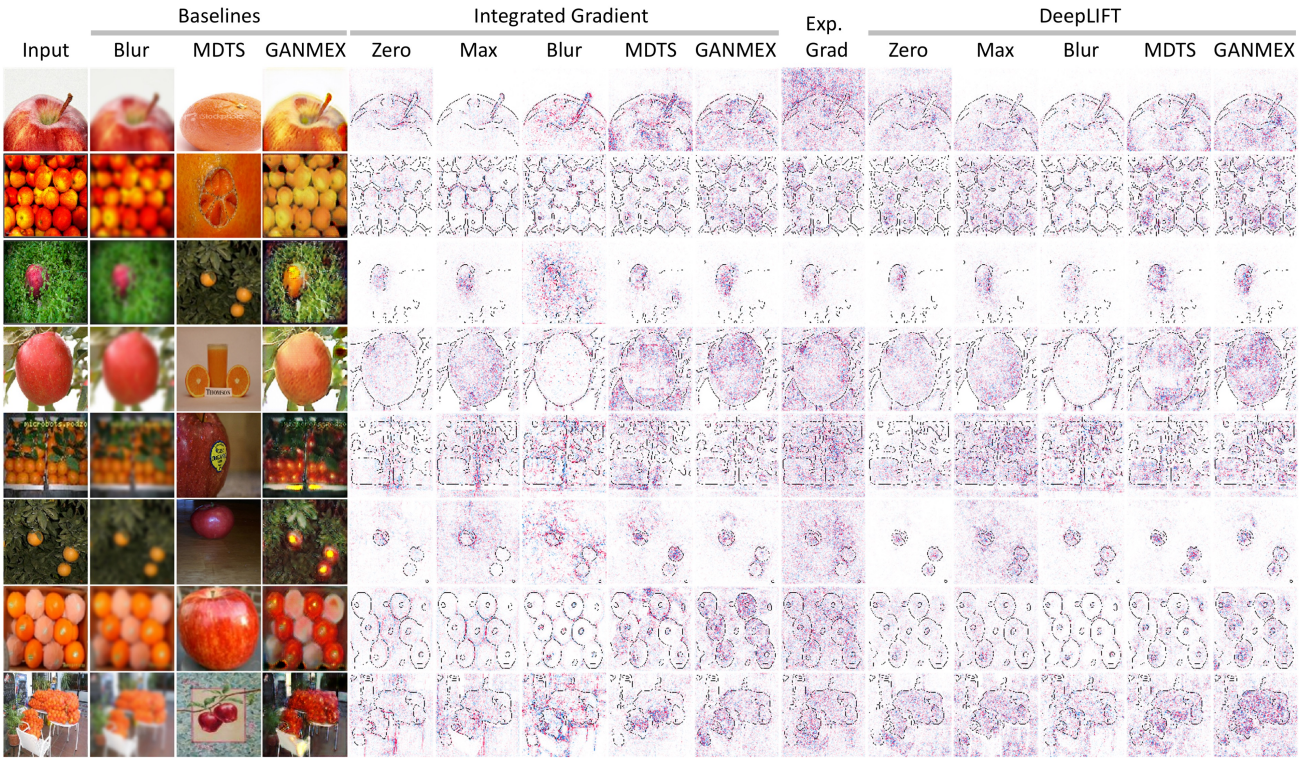

Figure 10. Additional examples of saliency maps for the classifier on the apple2orange dataset with four baseline choices: zero baseline (Zero), maximum value baselines (Max), blurred baselines (Blur), MDTS, Expected Gradient, and GANMEX baselines.