Discriminative Attribution from Paired Images

Anonymous ECCV submission

Paper ID 4

Abstract. We present a method for deep neural network interpretability by combining feature attribution with counterfactual explanations to generate attribution maps that highlight the most discriminative features between classes. Crucially, this method can be used to quantitatively evaluate the performance of feature attribution methods in an objective manner, thus preventing potential observer bias. We evaluate the proposed method on six diverse datasets, and use it to discover so far unknown morphological features of synapses in *Drosophila melanogaster*. We show quantitatively and qualitatively that the highlighted features are substantially more discriminative than those extracted using conventional attribution methods and improve upon similar approaches for counterfactual explainability. We argue that the extracted explanations are better suited for understanding fine grained class differences as learned by a deep neural network, in particular for image domains where humans have little to no visual priors, such as biomedical datasets.

1 Introduction

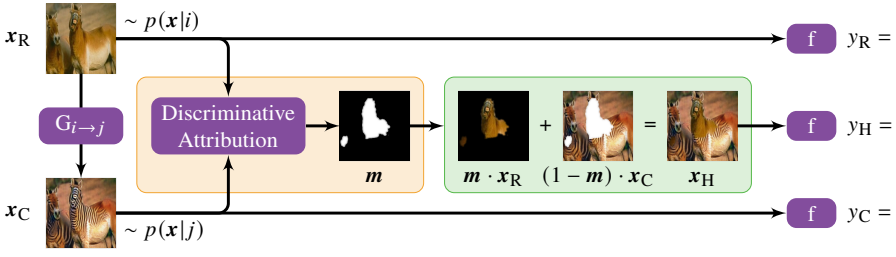

Fig. 1: Overview of the proposed method: An input image x_R of class i is converted through an independently trained cycle-GAN generator $G_{i\to j}$ into a counterfactual image x_C of class j, such that the classifier f we wish to interpret predicts $y_R = i$ and $y_C = j$. A discriminative attribution method then searches for the minimal mask m, such that copying the most discriminative parts of the real image x_R into the counterfactual x_C (resulting in the hybrid x_H) is again classified as $y_H = i$.

Machine Learning—and in particular Deep Learning—continues to see increased adoption in crucial aspects of society such as industry, science, and healthcare. As a result, there is an increasing need for tools that make Deep Neural Networks (DNNs) more interpretable for practitioners. Two popular approaches for explaining DNN classifiers

are so called *feature attribution* and *counterfactual* methods. Feature attribution methods highlight the input features that influence the output classification the most, whereas counterfactual methods present the user with a minimally modified input that would have led to a different output classification. In this work, we combine these two complementary approaches and devise a method for DNN interpretability that: (1) is able to highlight the most discriminative features of a given class pair, (2) can be objectively evaluated, and (3) is suitable for knowledge extraction from DNNs in cases where humans lack a clear understanding of class differences, a common situation for biomedical image datasets.

Feature attribution methods provide an explanation in terms of a heatmap over input pixels, highlighting and ranking areas of importance. A large number of approaches for feature attribution have been proposed in recent years (for a recent review see [27] and related work below) and they remain a popular choice for practitioners due to their ease of use, availability in popular Deep Learning frameworks, and intuitive outputs. However, the effectiveness, accuracy, and trustworthiness of these approaches is still debated [15,1,11,2], and an objective evaluation of those methods remains a difficult task [26,14]. Of particular concern are explanations that highlight the entire set of features that contributed to the output decision, including features that are shared between two different classes (so-called *distractors* [16]). This is problematic in the setting we are most concerned with, i.e., the extraction of information about class differences from a DNN in order to educate humans. This is particularly relevant in biomedical images, where humans have little to no visual priors (we show one such example in the experimental section of this work).

Counterfactual explanations are a complementary approach for explaining DNN decisions [20,36]. In contrast to feature attribution, counterfactual approaches attempt to explain a DNN output by presenting the user with another input that is close to the original input, but changes the classification decision of the DNN from class i to another class j. Comparing the two inputs then can be used to understand how class i differs from class j, enabling discriminative interpretability. For humans, this approach is arguably more natural and informative than presenting the full set of characteristics of class i, as done by feature attribution maps. However, generating counterfactual explanations typically involves an optimization procedure that needs to be carefully tuned in order to obtain a counterfactual with the desired properties. This process can be computationally expensive and does, in general, not allow for easy computation of attribution maps [35]. Due to these difficulties, counterfactual approaches are comparatively less popular for image data, where feature attribution methods arguably remain the dominant tool for practitioners.

To address these issues, we present a simple method that bridges the gap between counterfactual explainability and feature attribution (DAPI: Discriminative Attribution from Paired Images, see Fig. 1 for a visual summary). We use a cycle-GAN [40] to translate real images $\boldsymbol{x}_{\mathrm{R}}$ of class i to counterfactual images $\boldsymbol{x}_{\mathrm{C}}$ of class $j \neq i$. We then find an attribution map by processing the paired images with a new set of *discriminative attribution methods*, i.e., generalized versions of standard attribution methods. We show that this approach is able to generate sparse, high quality feature attribution maps that highlight the most discriminative features in the real and counterfactual image more precisely than standard attribution methods and prior approaches for discriminative

attribution. Crucially, the use of paired images allows quantification of the discriminatory power of attribution maps by performing an intervention, i.e., replacing the highlighted pixels in the counterfactual with the corresponding pixels in the real image. The difference in output classification score of this hybrid image, compared to the real image, then quantifies the importance of the swapped features. We validate DAPI on a set of six diverse datasets, including a challenging artificial dataset, a real world biological dataset (where a DNN solves a task human experts can not), MNIST, and three natural image datasets. For all six datasets, we show quantitatively and qualitatively that DAPI outperforms all other considered methods in identifying key discriminatory features between the classes.

Notably, we use DAPI to perform semi-automated knowledge extraction from a DNN that has been trained to classify electron microscopy images of synapses in *Drosophila melanogaster* by the neurotransmitter they release. Using DAPI, we are able to identify so far unknown morphological differences of synapses that release different neurotransmitters, discovering a crucial structure-function relationship for connectomics. Source code and datasets are publicly available at `anonymizedurl`.

2 Related Work

Interpretability methods can be broadly categorized into *local* or *global* methods. Local methods provide an explanation for every input, highlighting the reasons why a particular input is assigned to a certain class by the DNN. Global methods attempt to distill the DNN in a representation that is easier to understand for humans, such as decision trees. One can further distinguish between interpretability methods that are *post-hoc*, i.e., applicable to every DNN after it has been trained, and those methods that require modifications to existing architectures to perform interpretable classification as part of the model. In this work we focus on local post-hoc approaches to DNN interpretability for image classification.

Attribution Methods for Image Classification Even in this restricted class of approaches there is a large variety of methods [24,19,5,4,41,28,34,31,38,16,21,10,8,39,29,30,32]They have in common that they aim to highlight the most important features that contributed to the output classification score for a particular class, generating a heatmap indicating the influence of input pixels and features on the output classification. Among those, of particular interest to the work presented here are *baseline* feature attribution methods, which perform feature attribution estimation with reference to a second input. Those methods gained popularity as they assert sensitivity and implementation invariance [34,30]. The baseline is usually chosen to be the zero image as it is assumed to represent a neutral input.

Counterfactual Interpretability Since the introduction of counterfactual interpretability methods [20], the standard approach for generating counterfactuals broadly follows the procedure proposed by [36]. Concretely, a counterfactual is found as a result of an optimization aiming to maximize output differences while minimizing input differences between the real image $\boldsymbol{x}_\mathrm{R}$ and the counterfactual $\boldsymbol{x}_\mathrm{C}$: $\boldsymbol{x}_\mathrm{C} = \mathrm{argmin}_{\boldsymbol{x}} \quad L_i(\boldsymbol{x}_\mathrm{R}, \boldsymbol{x}) -$

$L_o(f(\boldsymbol{x}_\mathrm{R}), f(\boldsymbol{x}))$, with L_i and L_o some loss that measures the distance between inputs and outputs, respectively, and f the classifier in question. Optimizing this objective can be problematic because it contains competing losses and does not guarantee that the generated counterfactual $\boldsymbol{x}_\mathrm{C}$ is part of the data distribution $p(\boldsymbol{x})$. Current approaches try to remedy this by incorporating additional regularizers in the objective [18,35], such as adversarial losses that aim to ensure that the counterfactual $\boldsymbol{x}_\mathrm{C}$ is not distinguishable from a sample $\boldsymbol{x} \sim p(\boldsymbol{x})$ [6,18]. However, this does not address the core problem of competing objectives and will result in a compromise between obtaining in-distribution samples, maximizing class differences, and minimizing input differences. We circumvent this issue by omitting the input similarity loss in the generation of counterfactuals and instead enforce similarity post-hoc, similar to the strategy used by [22]. Similar to ours, the work by [23] uses a cycle-GAN to generate counterfactuals for DNN interpretability. However, this method differs in that the cycle-GAN is applied multiple times to a particular input in order to increase the visual differences in the real and counterfactual images for hypothesis generation. Subsequently, the found features are confirmed by contrasting the original classifiers performance with one that is trained on the discovered features, which does not lead to attribution maps or an objective evaluation of feature importance.

Attribution and Counterfactuals To the best of our knowledge, two other methods explore the use of counterfactuals for feature attribution: Wang et al. [37] introduces a novel family of so-called *discriminative explanations* that also leverage attribution on a real and counterfactual image in addition to confidence scores from the classifier to derive attributions for the real and counterfactual image that show highly discriminative features. This approach requires calculation of three different attribution maps, which are subsequently combined to produce a discriminative explanation. In addition, this method does not generate new counterfactuals using a generative model, but instead selects a real image from a different class. On one hand this is advantageous because it does not depend on the generator's performance, but on the other hand this does not allow creating hybrid images for the evaluation of attribution maps. Closest to our approach is the method presented by Goyal et al. [12], where counterfactual visual explanations are generated by searching for feature sets in two real images of different classes that, if swapped, influence the classification decision. To this end, an optimization problem is solved to find the best features to swap, utilizing the network's feature representations. The usage of real (instead of generated) counterfactuals can lead to artifacts during the replacement of features. Furthermore, the attributions are limited in resolution to the field of view of the feature layer considered for performing the swap. In addition, depending on the chosen architecture, swapping features between images is not equivalent to swapping the corresponding pixel regions as the field of view of multiple units are overlapping. This makes it difficult to cleanly associate input features with the observed change in classification score, as we show in Section 4.

Attribution Evaluation Being able to objectively evaluate attribution methods is important for method selection and trusting the generated explanations. Prior work evaluated the importance of highlighted features by removing them [26]. However, it has been noted that this strategy is problematic because it is unclear whether any observed performance degradation is due to the removal of relevant features or because the new sample comes

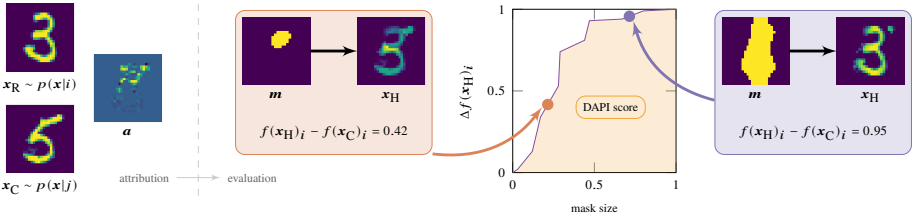

Fig. 2: Evaluation procedure for discriminative attribution methods: Given the real image $\boldsymbol{x}_\mathrm{R}$ of class i and its counterfactual $\boldsymbol{x}_\mathrm{C}$ of class j, we generate a sequence of binary masks $\boldsymbol{m}$ by applying different thresholds to the attribution map $\boldsymbol{a}$. Those masks are then used to generate a sequence of hybrid images $\boldsymbol{x}_\mathrm{H}$. The plot shows the change in classifier prediction $\Delta f(\boldsymbol{x}_\mathrm{H})_i = f(\boldsymbol{x}_\mathrm{H})_i - f(\boldsymbol{x}_\mathrm{C})_i$ over the size of the mask $\boldsymbol{m}$ (normalized between 0 and 1). The DAPI score is the area under the curve, i.e., a value between 0 and 1. Higher DAPI scores are better and indicate that a discriminative attribution method found small regions that lead to the starkest change in classification.

from a different distribution. As a result, strategies to remedy this issue have been proposed, for example by retraining classifiers on the modified samples [14]. Instead of removing entire features, in this work we replace them with their corresponding counterfactual features.

3 Method

The method we propose combines counterfactual interpretability with discriminative attribution methods to find and highlight the most important features between images of two distinct classes i and j, given a pretrained classifier f. For that, we first generate for a given input image $\boldsymbol{x}_\mathrm{R}$ of class i a counterfactual image $\boldsymbol{x}_\mathrm{C}$ of class j. We then use a *discriminative* attribution method to find the attribution map of the classifier for this pair of images. As we will show qualitatively and quantitatively in Section 4, using paired images results in attribution maps of higher quality. Furthermore, the use of a counterfactual image gives rise to an objective evaluation procedure for attribution maps. In the next sections we describe (1) our choice for generating counterfactual images, (2) the derivation of discriminative attribution methods from existing baseline attribution methods, and (3) how to use counterfactual images to evaluate attribution maps. We denote with f a pretrained classifier with N output classes, input images $\boldsymbol{x} \in \mathbb{R}^{h \times w}$, and output vector $f(\boldsymbol{x}) = \boldsymbol{y} \in [0,1]^N$ with $\sum_i y_i = 1$.

3.1 Creation of Counterfactuals

We train a cycle-GAN [40] for each pair of image classes $i \neq j \in \{1, ..., N\}$, which enables translation of images of class i into images of class j and vice versa. We perform this translation for each image of class i and each target class $j \neq i$ to obtain datasets of paired images $D_{i \to j} = \left\{(\boldsymbol{x}_\mathrm{R}^k, \boldsymbol{x}_\mathrm{C}^k) | k = 1, \ldots, n(i)\right\}$, where $\boldsymbol{x}_\mathrm{R}^k$ denotes the kth real image of class i and $\boldsymbol{x}_\mathrm{C}^k$ its counterfactual of class j. We then test for each image in the dataset

whether the translation was successful by classifying the counterfactual image $\boldsymbol{x}_{\mathrm{C}}$ and reject a sample pair whenever $f(\boldsymbol{x}_{\mathrm{C}})_j < \theta$, with θ a threshold parameter (in the rest of this work we set $\theta = 0.8$). This procedure results in a dataset of paired images, where the majority of the differences between an image pair is expected to be relevant for the classifiers decision, i.e., we retain formerly present non-discriminatory distractors such as orientation, lighting, or background. We encourage that the translation makes as little changes as necessary by choosing a ResNet [13] architecture for the cycle-GAN generator, which is able to trivially learn the identity function.

3.2 Discriminative Attribution

The datasets $D_{i \to j}$ are already useful to visualize data-intrinsic class differences (see Fig. 3 for examples). However, we wish to understand which input features the classifier f makes use of. Specifically, we are interested in finding the smallest binary mask $\boldsymbol{m}$, such that swapping the contents of $\boldsymbol{x}_{\mathrm{C}}$ with $\boldsymbol{x}_{\mathrm{R}}$ within this mask changes the classification under f. To find $\boldsymbol{m}$, we repurpose existing attribution methods that are amendable to be used with a reference image. The goal of those methods is to produce attribution maps $\boldsymbol{a}$, which we convert into a binary mask via thresholding. A natural choice for our purposes are so-called *baseline attribution methods*, which derive attribution maps by contrasting an input image with a baseline sample (e.g., a zero image). In the following, we review suitable attribution methods and derive discriminative versions that use the counterfactual image as their baseline. We will denote the discriminative versions with the prefix D.

Input * Gradients One of the first and simplest attribution methods is *Input * Gradients* (INGRADS) [30,31], which is motivated by the first order Taylor expansion of the output class with respect to the input around the zero point:

$$\mathrm{INGRADS}(\boldsymbol{x}) = |\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_i \cdot \boldsymbol{x}|, \tag{1}$$

where i is the class for which an attribution map is to be generated. We derive an explicit baseline version for the discriminatory attribution of the real $\boldsymbol{x}_{\mathrm{R}}$ and its counterfactual $\boldsymbol{x}_{\mathrm{C}}$ by choosing $\boldsymbol{x}_{\mathrm{C}}$ as the Taylor expansion point:

$$D\text{-}\mathrm{INGRADS}(\boldsymbol{x}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{C}}) = \left|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_j \Big|_{\boldsymbol{x}=\boldsymbol{x}_{\mathrm{C}}} \cdot (\boldsymbol{x}_{\mathrm{C}} - \boldsymbol{x}_{\mathrm{R}})\right|, \tag{2}$$

where j is the class of the counterfactual image.

Integrated Gradients *Integrated Gradients* (IG) is an explicit baseline attribution method, where gradients are accumulated along the straight path from a baseline input $\boldsymbol{x}_0$ to the input image $\boldsymbol{x}$ to generate the attribution map [34]. Integrated gradients along the kth dimension are given by:

$$\mathrm{IG}_k(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0)_k \cdot \int_{\alpha=0}^{1} \frac{\partial f(\boldsymbol{x}_0 + \alpha(\boldsymbol{x} - \boldsymbol{x}_0))_i}{\partial \boldsymbol{x}_k} d\alpha. \tag{3}$$

We derive a discriminatory version of IG by replacing the baseline as follows:

$$D\text{-IG}_k(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) = (\boldsymbol{x}_\text{C} - \boldsymbol{x}_\text{R})_k \cdot \int_{\alpha=0}^{1} \frac{\partial f(\boldsymbol{x}_\text{R} + \alpha(\boldsymbol{x}_\text{C} - \boldsymbol{x}_\text{R}))_j}{\partial \boldsymbol{x}_k} d\alpha. \tag{4}$$

Deep Lift *Deep Lift* (DL) is also an explicit baseline attribution method which aims to compare individual neuron's activations of an input w.r.t. a reference baseline input [30]. It can be expressed in terms of the gradient in a similar functional form to IG:

$$\text{DL}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0) \cdot F_{DL}, \tag{5}$$

where F_{DL} is some function of the gradient of the output (see [3] for the full expression). The discriminative attribution we consider is simply:

$$D\text{-DL}(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) = (\boldsymbol{x}_\text{C} - \boldsymbol{x}_\text{R}) \cdot F_{DL}. \tag{6}$$

GradCAM *GradCAM* (GC) is an attribution method that considers the gradient weighted activations of a particular layer, usually the last convolutional layer, and propagates this value back to the input image [28]. We denote the activation of a pixel (u, v) in layer l with size (h, w) and channel k by $C^l_{k,u,v}$ and write the gradient w.r.t. the output $\boldsymbol{y}$ as:

$$\nabla_{C^l_k} \boldsymbol{y} = \left(\frac{d\boldsymbol{y}}{dC^l_{k,0,0}}, \frac{d\boldsymbol{y}}{dC^l_{k,1,0}}, \frac{d\boldsymbol{y}}{dC^l_{k,2,0}}, ..., \frac{d\boldsymbol{y}}{dC^l_{k,h,w}}\right) \tag{7}$$

The original GC is then defined as:

$$\begin{aligned} \text{GC}(\boldsymbol{x}) &= \text{ReLU}\left(\sum_k \nabla_{C_k} \boldsymbol{y} \cdot \vec{C}_k\right) \\ &= \text{ReLU}\left(\sum_k \sum_{u,v} \frac{d\boldsymbol{y}}{dC_{k,u,v}} C_{k,u,v}\right) \\ &= \text{ReLU}\left(\sum_k \alpha_k C_k\right), \end{aligned} \tag{8}$$

where we ommitted the layer index l for brevity. Each term $\frac{d\boldsymbol{y}}{dC_{k,u,v}} C_{k,u,v}$ is the contribution of pixel u, v in channel k to the output classification score $\boldsymbol{y}$ under a linear model. GC utilizes this fact and projects the layer attribution from layer l back to the input image, generating the final attribution map. In contrast to the setting considered by GC, we have access to a matching pair of real and counterfactual images $\boldsymbol{x}_\text{R}$ and $\boldsymbol{x}_\text{C}$. We extend GC to consider both feature maps $C^{\boldsymbol{x}_\text{R}}_k$ and $C^{\boldsymbol{x}_\text{C}}_k$ by treating GC as an implicit zero baseline method similar to INGRADS:

$$D\text{-GC}_k(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) = \left.\frac{d\boldsymbol{y}_j}{dC_k}\right|_{C=C^{\boldsymbol{x}_\text{C}}_k} \left(C_k(\boldsymbol{x}_\text{C}) - C_k(\boldsymbol{x}_\text{R})\right). \tag{9}$$

Averaging those gradients over feature maps k, and projecting the activations back to image space then highlights pixels that are most discriminative for a particular pair:

$$D\text{-GC}_\text{P}(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) = \left| \mathbb{P} \sum_k D\text{-GC}_k(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) \right|, \tag{10}$$

where $\mathbb{P}$ is the projection matrix (in this work we simply rescale to the input size) from feature space C to input space X. Note that in contrast to GC, we use the absolute value of the output attribution, as we do not apply ReLU activations to layer attributions. Because feature maps can be of lower resolution than the input space, GC tends to produce coarse attribution maps [28]. To address this issue it is often combined with *Guided Backpropagation* (GBP), a method that uses the (positive) gradients of the output class w.r.t. the input image as the attribution map [33]. *Guided GradCAM* (GGC) uses this strategy to sharpen the attribution of GC via element-wise multiplication of the attribution maps [28]. For the baseline versions we thus consider multiplication of D-GC with the GBP attribution maps:

$$\text{GBP}(\boldsymbol{x}) = \nabla_x f(\boldsymbol{x})_i \quad \text{, with} \nabla\text{ReLU} > 0 \tag{11}$$

$$\text{GGC}(\boldsymbol{x}) = \text{GC}(\boldsymbol{x}) \cdot \text{GBP}(\boldsymbol{x}) \tag{12}$$

$$D\text{-GGC}(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) = D\text{-GC}(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C}) \cdot \text{GBP}(\boldsymbol{x}_\text{R}). \tag{13}$$

3.3 Evaluation of Attribution Maps

The discriminative attribution map $\boldsymbol{a}$ obtained for pair of images $(\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{C})$ can be used to quantify the causal effect of the attribution. Specifically, we can copy the area highlighted by $\boldsymbol{a}$ from the real image $\boldsymbol{x}_\text{R}$ of class i to the counterfactual image $\boldsymbol{x}_\text{C}$ of class j, resulting in a hybrid image $\boldsymbol{x}_\text{H}$. If the attribution accurately captures class-relevant features, we would expect that the classifier f assigns a high probability to $\boldsymbol{x}_\text{H}$ being of class i. The ability to create those hybrid images is akin to an intervention, and has two important practical implications: First, it allows us to find a minimal binary mask that captures the most class-relevant areas for a given input image. Second, we can compare the change in classification score for hybrids derived from different attribution maps. This allows us to compare different methods in an objective manner, following the intuition that an attribution map is better, if it changes the classification with less pixels changed. To find a minimal binary mask $\boldsymbol{m}_\text{min}$, we search for a threshold of the attribution map $\boldsymbol{a}$, such that the mask score $\Delta f(\boldsymbol{x}_\text{H}) = f(\boldsymbol{x}_\text{H})_i - f(\boldsymbol{x}_\text{C})_i$ (i.e., the change in classification score) is maximized while the size of the mask is minimized, i.e., $\boldsymbol{m}_\text{min} = \arg\min_{\boldsymbol{m}} \ |\boldsymbol{m}| - \Delta f(\boldsymbol{x}_\text{H})$ (where we omitted the dependency of $\boldsymbol{x}_\text{H}$ on $\boldsymbol{m}$ for brevity). In order to minimize artifacts in the copying process we also apply a morphological closing operation with a window size of 10 pixels followed by a Gaussian Blur with $\sigma = 11px$. The final masks highlight the relevant class discriminators by showing the user the counterfactual features, the original features they are replaced with, and the corresponding mask score $\Delta f(\boldsymbol{x}_\text{H})$, indicating the quantitative effect of the replacement on the classifier. See Fig. 3 for example pairs and corresponding areas $\boldsymbol{m}_\text{min}$. Furthermore, by applying a sequence of thresholds for the attribution map $\boldsymbol{a}$, we derive an objective evaluation procedure for

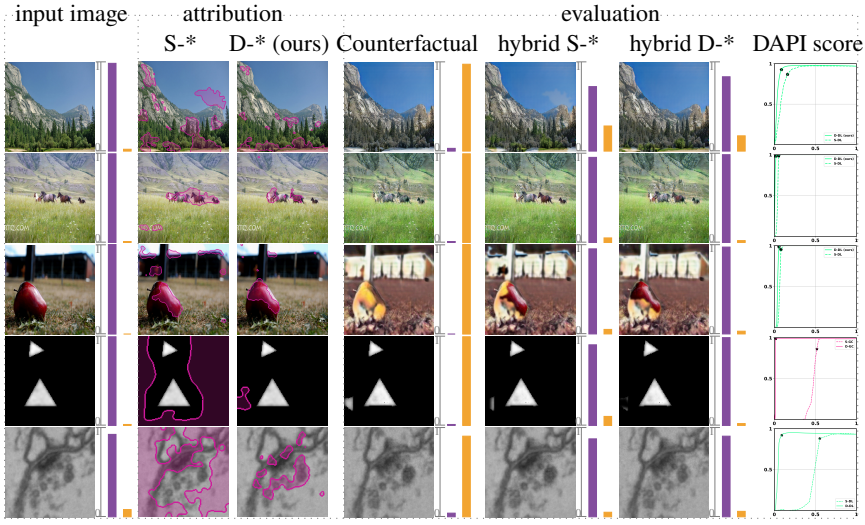

Fig. 3: Samples from the best performing method pairs (S: "single input", D: discriminative) on SUMMER, HORSES, APPLES, DISC-A and SYNAPSES (different rows). Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. Bars indicate the classifier's prediction for class i (purple) and j (orange). The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score). Additional qualitative results can be found in Supplementary Section 2.

a given attribution map: For each hybrid image $\boldsymbol{x}_{\mathrm{H}}$ in the sequence of thresholds, we consider the change in classifier prediction relative to the size of the mask that has been used to create the hybrid. We accumulate the change in classifier prediction over all mask sizes to derive our proposed DAPI score. This procedure is explained in detail in Fig. 2 for a single pair of images. When reporting the DAPI score for a particular attribution method, we average the single DAPI scores over all images, and all distinct pairs of classes.

4 Experiments

We evaluate the presented method on six datasets: MNIST [17], SYNAPSES [9][1], two versions of a synthetic dataset that we call DISC-A and DISC-B, and three natural image datasets HORSES, APPLES and SUMMER from [40]. For a more detailed overview of all considered datasets, see Supplementary Section 3.

SYNAPSES A real world biological dataset, consisting of $1 \times 128 \times 128$px electron microscopy images of synaptic sites in the brain of *Drosophila melanogaster*. Each image

[1] Dataset kindly provided by the authors of [9].

is labelled with a functional property of the synapse (six different classes), namely the neurotransmitter it releases (the label was acquired using immunohistochemistry labelling, see [9] for details). This dataset is of particular interest for interpretability, since a DNN can recover the neurotransmitter label from the images with high accuracy, but human experts are not able to do so. Interpretability methods like the one presented here can thus be used to gain insights into the relation between structure (from the electron microscopy image) and function (the neurotransmitter released). See Fig. 6 for an example of a discriminatory feature between synapses that release the two different neurotransmitters GABA and Acetylcholine, discovered by applying DAPI on the SYNAPSES dataset. A full description of all discovered feature differences can be found in the supplementary material.

HORSES, APPLES & SUMMER Three natural image datasets corresponding to binary classification tasks. HORSES consists of two sets of images showing horses and zebras respectively, APPLES is a dataset of images depicting apples and oranges, and SUMMER shows landscape pictures in summer and winter. We scale each image to $3 \times 256 \times 256$ for classification and image translation.

DISC-A & DISC-B Two synthetic datasets with different discriminatory features. Each image is $1 \times 128 \times 128$px in size and contains spheres, triangles or squares. For DISC-A, the goal is to correctly classify images containing an even or odd number of triangles. DISC-B contains images that show exactly two of the three available shapes and the goal is to predict which shape is missing (e.g., an image with only triangles and squares is to be classified as "does not contain spheres"). This dataset was deliberately designed to investigate attribution methods in a setting where the discrimination depends on the absence of a feature.

Training For MNIST, DISC, HORSES, APPLES and SUMMER we train a VGG and ResNet for 100 epochs and select the epoch with highest accuracy on a held out validation dataset. For SYNAPSES we adapt the 3D-VGG architecture from [9] to 2D and train for 500,000 iterations. We select the iteration with the highest validation accuracy for testing. For MNIST, DISC & SYNAPSES we train one cycle-GAN for 200 epochs, on each class pair and on the same training set the respective classifier was trained on. For HORSES, APPLES and SUMMER we use the pretrained cycle-GAN checkpoints provided by [40] (the full network specifications are given in the supplement).

Results Quantitative results (in terms of the DAPI score, see Section 3.3) for each investigated attribution method are shown in Fig. 4 and Table 1. In summary, we find that attribution maps generated from the proposed discriminative attribution methods consistently outperform their original versions in terms of the DAPI score. This observation also holds visually: the generated masks from discriminative attribution methods are smaller and more often highlight the main discriminatory parts of a considered image pair (see Fig. 3). In particular, the proposed method substantially outperforms the considered random baseline, whereas standard attribution methods sometimes fail to do so (e.g., GC on dataset SYNAPSES). Furthermore, on MNIST and DISC-A, the mask derived from the residual of real and counterfactual image is already competitive with

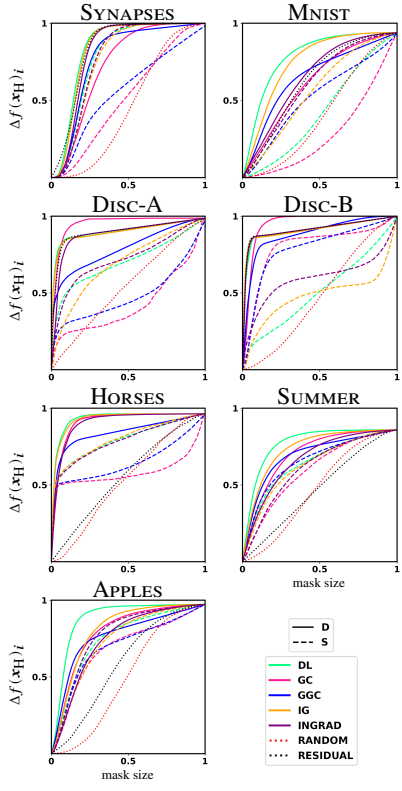

Fig. 4: Quantitative evaluation of discriminative (D - solid) and corresponding original (S for "single input" - dashed) attribution methods over six datasets (SYNAPSES, MNIST, two versions of DISC, HORSES, SUMMER, and APPLES). Attributions for D and S methods are calculated as described in the methods section, with a zero baseline for all S methods that are explicit baseline methods, following standard practice. Corresponding D and S versions of the same method are shown in the same color. For each, we plot the average change of classifier prediction $\Delta f(\boldsymbol{x}_H)_i^k = f(\boldsymbol{x}_H)_i - f(\boldsymbol{x}_C)_i$ as a function of mask size $\boldsymbol{m} \in [0, 1]$. In addition we show performance of the two considered baselines: masks derived from random attribution maps (random - red, dotted) and mask derived from the residual of the real and counterfactual image (residual - black, dotted). On all considered datasets all versions of D attribution outperform their S counterparts with the single exception of INGRAD on the APPLES dataset. All experiments shown here are performed with VGG architectures (we observe similar results with ResNet, see Supplementary Section 2).

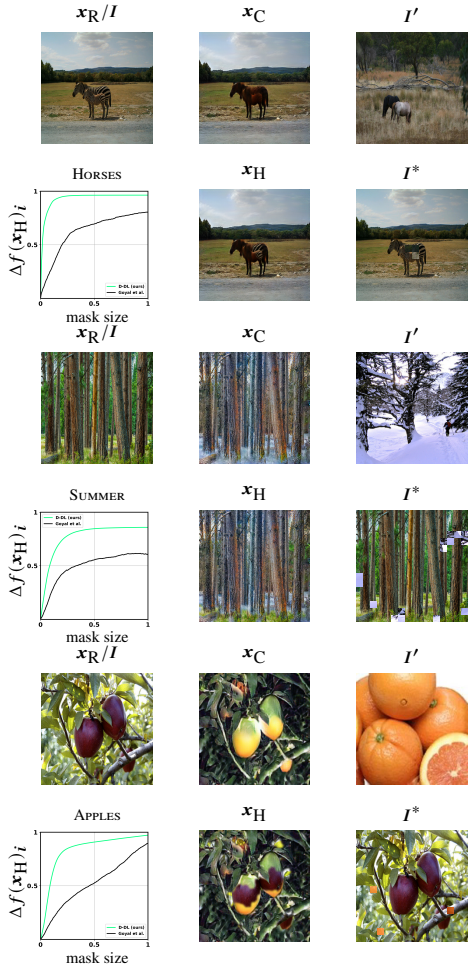

Fig. 5: Comparison of our method to the method presented by Goyal et al. [12] for the three natural image datasets Horses, Summer and Apples (See Supplementary Section 4 for the other datasets and Table 1 for associated DAPI scores). For each dataset we show the real/query image(x_R/I) as well as the counterfactual x_C and hybrid x_H for DAPI. For [12], we show the distractor image I' as well as the hybrid I^* after the minimal amount of swaps needed to change the classifier decision. In addition, we show the DAPI scores for [12] and our best performing method for each dataset averaged over all image pairs. Note that [12] considers replacement of patches from the distractor image I' to the query image I, while we consider the opposite direction $x_R \rightarrow x_C$.

Dataset	D-IG	D-DL	D-INGR.	D-GC	D-GGC	RES.	IG	DL	INGR.	GC	GGC	RND.	GOYAL
MNIST	0.83	**0.84**	0.82	0.73	0.78	0.84	0.77	0.79	0.77	0.52	0.56	0.46	0.27
SYNAPSES	0.75	**0.79**	0.65	0.62	0.65	0.63	0.56	0.43	0.61	0.28	0.52	0.41	0.21
DISC-A	0.9	0.9	0.88	**0.95**	0.79	0.9	0.69	0.7	0.72	0.43	0.48	0.54	0.41
DISC-B	0.91	0.91	0.91	**0.95**	0.88	0.91	0.48	0.51	0.6	0.8	0.79	0.48	0.46
HORSES	0.93	**0.93**	0.9	0.91	0.84	0.56	0.8	0.79	0.78	0.58	0.65	0.53	0.62
SUMMER	0.73	**0.77**	0.67	0.7	0.69	0.47	0.63	0.64	0.61	0.6	0.65	0.48	0.49
APPLES	0.8	**0.88**	0.74	0.79	0.77	0.59	0.73	0.75	0.77	0.68	0.69	0.5	0.52

Table 1: Summary of DAPI scores using VGG architecures for each investigated method on the six datasets MNIST, SYNAPSES, HORSES, SUMMER, APPLES and DISC (two versions) corresponding to 4. Best results are highlighted.

the best considered methods and outperforms standard attribution substantially. However, for more complex datasets such as SYNAPSES, HORSES, APPLES and SUMMER, the residual becomes less accurate in highlighting discriminative features. Here, the discriminatory attributions outperform all other considered methods.

In addition to the considered baseline attribution methods we also compare DAPI with the method proposed by Goyal et al. [12]. To this end, we iterated the BESTEDIT procedure until the features of the query image I have been entirely replaced with features from the distractor image I' (see Algorithm 1 in [12]). We then swapped input patches that underlie those features between I and I' accordingly to obtain a sequence of I^* that gradually transforms I into a shuffled version of I'. We measure the impact on the classifier score in the same way as with $\boldsymbol{x}_\mathrm{H}$ in our method (see Fig. 5 for examples on HORSES, SUMMER, and APPLES; Table 1 for aggregate results on all datasets; and Supplement Section 4 for an extended analysis). We find that DAPI consistently finds smaller regions with larger explanatory power.

5 Discussion

This work demonstrates that the combination of counterfactual interpretability with suitable attribution methods is more accurate in extracting key discriminative features between class pairs than standard methods. While the method succeeds in the presented experiments, it comes with a number of limitations. It requires the training of cycle-GANs, one for each pair of output classes. Thus training time and compute cost scale quadratically in the number of output classes and it is therefore not feasible for classification problems with a large number of classes. Furthermore, the translation from the real to the counterfactual image could fail for a large fraction of input images, i.e., $f(\boldsymbol{x}_\mathrm{C}) \neq j$. In this work, we only consider those image pairs where translation was successful, as we focus on extracting knowledge about class differences from the classifier. For applications that require an attribution for each input image this approach is not suitable. An additional concern is that focusing only on images that have a successful translation may bias the dataset we consider and with it the results. GANs are known to exhibit so-called mode collapse [7,25], meaning they focus on only a small set of modes of the full distribution. As a consequence, the method described here may miss discriminatory features present

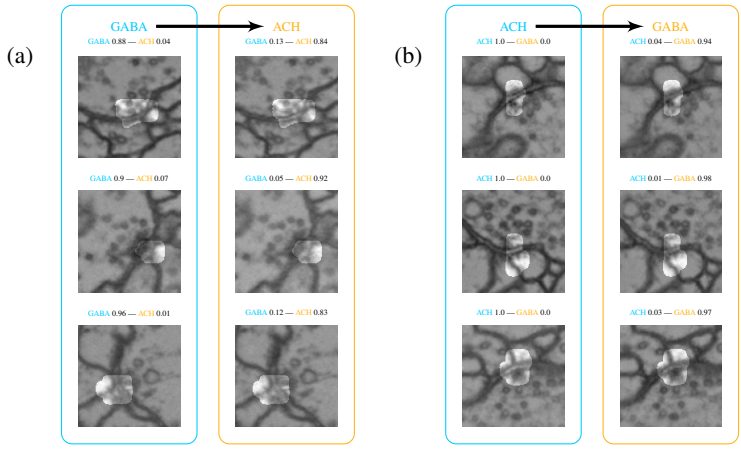

Fig. 6: DAPI reveals so far unnoticed morphological features of synapses. Shown are electron microscopy images and associated masks generated by DAPI for real synapses (blue boxes) that express the neurotransmitters GABA (a) and Acetylcholine (b), next to their respective translations to the other class (orange boxes). The classification score is shown above each image, validating a successful translation. As before, masks highlight regions that change the classification decision back to the real class if swapped from real to fake sample. Masks consistently highlight the synaptic cleft, and we observe a brightening of the inside of the cleft when translating from GABA to ACH and a darkening the other way around.

in other modes. Furthermore, image classes need to be sufficiently similar in appearance for the cycle-GAN to work, and translating, e.g., an image of a mouse into an image of a tree is unlikely to work and produce meaningful attributions. However, we believe that the generation of masks in combination with the corresponding mask score is superior to classical attribution maps for interpreting DNN decision boundaries, especially for the analysis of fine-grained class differences; a common situation for biomedical image datasets and exemplified here by the SYNAPSES dataset. Although we present this work in the context of understanding DNNs and the features they make use of, an uncritical adaptation of this and other similar interpretability methods can potentially lead to ethical concerns. As an example, results should be critically evaluated when using this method to interpret classifiers that have been trained to predict human behaviour, or demographic and socioeconomic features. As with any data-heavy method, it is important to realize that results will be reflective of data- and model-intrinsic biases. As such, an interpretability method like the one we present here can at most identify a correlation between input features and labels, but not true causal links. The method presented here should therefore not be used to “prove” that a particular feature leads to a particular outcome.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. arXiv preprint arXiv:1810.03292 (2018) 2
2. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049 (2018) 2
3. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Sy21R9JAW 7
4. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015) 3
5. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. The Journal of Machine Learning Research **11**, 1803–1831 (2010) 3
6. Barredo-Arrieta, A., Del Ser, J.: Plausible counterfactuals: Auditing deep learning classifiers with realistic adversarial examples. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2020) 4
7. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136 (2016) 13
8. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. arXiv preprint arXiv:1705.07857 (2017) 3
9. Eckstein, N., Bates, A.S., Du, M., Hartenstein, V., Jefferis, G.S., Funke, J.: Neurotransmitter classification from electron microscopy images at synaptic sites in drosophila. BioRxiv (2020) 9, 10
10. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3429–3437 (2017) 3
11. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3681–3688 (2019) 2
12. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: International Conference on Machine Learning. pp. 2376–2384. PMLR (2019) 4, 12, 13
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 6
14. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. arXiv preprint arXiv:1806.10758 (2018) 2, 5
15. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 267–280. Springer (2019) 2
16. Kindermans, P.J., Schütt, K.T., Alber, M., Müller, K.R., Erhan, D., Kim, B., Dähne, S.: Learning how to explain neural networks: Patternnet and patternattribution. arXiv preprint arXiv:1705.05598 (2017) 2, 3
17. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/ 9
18. Liu, S., Kailkhura, B., Loveland, D., Han, Y.: Generative counterfactual introspection for explainable deep learning. arXiv preprint arXiv:1907.03077 (2019) 4
19. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874 (2017) 3

20. Martens, D., Provost, F.: Explaining data-driven document classifications. Mis Quarterly **38**(1), 73–100 (2014) 2, 3
21. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition **65**, 211–222 (2017) 3
22. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 607–617 (2020) 4
23. Narayanaswamy, A., Venugopalan, S., Webster, D.R., Peng, L., Corrado, G.S., Ruamviboonsuk, P., Bavishi, P., Brenner, M., Nelson, P.C., Varadarajan, A.V.: Scientific discovery by generating counterfactuals using image translation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 273–283. Springer (2020) 4
24. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. arXiv preprint arXiv:1606.05386 (2016) 3
25. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. arXiv preprint arXiv:1606.03498 (2016) 13
26. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems **28**(11), 2660–2673 (2016) 2, 4
27. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE **109**(3), 247–278 (2021) 2
28. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017) 3, 7, 8
29. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International Conference on Machine Learning. pp. 3145–3153. PMLR (2017) 3
30. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016) 3, 6, 7
31. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps (2014) 3, 6
32. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017) 3
33. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014) 8
34. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3319–3328. JMLR. org (2017) 3, 6
35. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: A review. arXiv preprint arXiv:2010.10596 (2020) 2, 4
36. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017) 2, 3
37. Wang, P., Vasconcelos, N.: Scout: Self-aware discriminant counterfactual explanations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8981–8990 (2020) 4
38. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014) 3
39. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 (2016) 3

40. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017) 2, 5, 9, 10
41. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595 (2017) 3

Discriminative Attribution from Paired Images

Supplementary Material

Anonymous ECCV submission

Paper ID 4

A Training Details

A.1 Network Architectures

Cycle-GAN We use the cylce-GAN implementation from https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix. For MNIST, SYNAPSES and DISC, we use a 9-block RESNET generator and a 70×70 PatchGAN [3] discriminator. For training we use a least squares loss (LSGAN [5]), a batch size of one, instance normalization and normal initialization. We use the Adam optimizer [4] with momentum $\beta_1 = 0.5$ and a learning rate of 0.0002 with a linear decay to zero after the first 100 epochs. For HORSES, SUMMER and APPLES we use the pretrained networks from [7].

Classifiers The classifiers used for attribution are either VGG (for datasets SYNAPSES, MNIST, DISC, HORSES, SUMMER and APPLES) or RESNET (for datasets MNIST, DISC, HORSES, SUMMER and APPLES) architectures, trained using a cross-entropy loss. Individual layers are shown in Table 1 and Table 2. For the training of the VGG network on the SYNAPSES dataset, we use the same strategy (including augmentations) as described in [1], with the only difference being that we consider 2D images instead of 3D volumes. We did not attempt to train a RESNET on the SYNAPSES dataset. For the training of the VGG and RESNET architectures on the MNIST and DISC datasets we did not make use of augmentations and trained each network for 100 epochs with a batch size of 32 using the Adam optimizer (learning rate 10^{-4}).

A.2 Compute

The most significant part of the compute costs come from training the cycle-GANs. For each experiment, cycle-GAN training for 200 epochs took around 5 days on a single RTX 2080Ti GPU. For MNIST experiments we trained a total of 45 cycle GANs, 15 for SYNAPSES, and 4 for DISC. In total this results in roughly 320 GPU-days for cycle-GAN training. In contrast, attribution and mask generation is comparatively cheap and takes between 1-3 hours on 20 RTX 2080Ti GPUs for each dataset, resulting in 60 GPU hours for each experiment and 30 GPU days in total.

To alleviate the comparatively large compute costs incurred by the training of cycle-GANs for each class pair, one could consider a one-vs-all CycleGAN and perform subsequent attribution on n (instead of n^2) cases. However, for practical applications, the computational complexity of the pairwise analysis is likely acceptable as it is inherited from the concrete question one seeks to answer and ultimately limited by the number of class pairs a human is able or willing to look at.

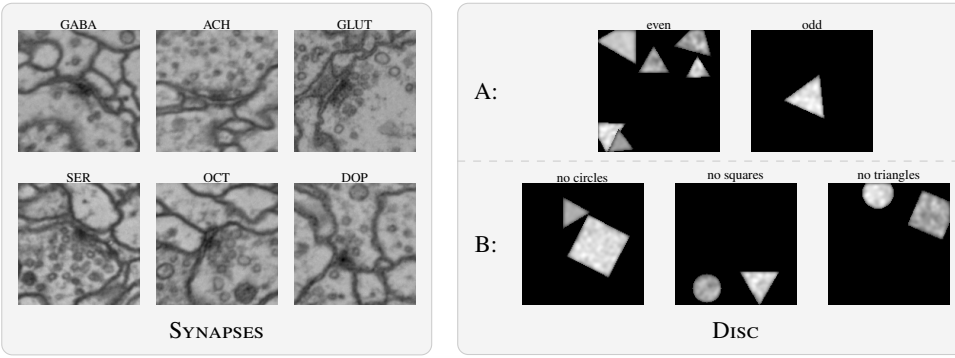

Fig. 1: Example images of datasets SYNAPSES and DISC. SYNAPSES consists of electron microscopy images of synapses. Each class is defined by the neurotransmitter the synapse releases. DISC is a synthetic dataset we designed in order to highlight failure cases of popular attribution methods. We consider two subsets: DISC-A shows triangles in each image and classes are defined by the parity of the number of triangles. DISC-B consists of images showing triangles, squares, and disks. Each class is one combination of two shapes.

B Extended Results

In addition to the results using VGG architectures in the main text, below we show additional results for RESNET architectures on MNIST, DISC-B, HORSES, SUMMER and APPLES (see Fig. 2 and Table 3). We do not show results for DISC-A (even vs. odd number of shapes), because the considered RESNET architecture failed to achieve more than chance level accuracy on the validation dataset (note that we did not attempt any architecture optimization as this is not the focus of this study). Since our goal is to understand what the classifier learned about class differences, using a network that did not successfully learn to classify will not produce meaningful results. Similarly we do not show RESNET results for the SYNAPSES dataset, because the original study uses a VGG classifier, and reproducing those results with RESNET architectures is out of the scope of this work. In summary, the results for RESNET architectures follow a similar pattern as observed in the main VGG results: Almost all discriminative attribution methods outperform their counterparts in terms of DAPI-score. However, DAPI scores of all considered methods are reduced compared to the results obtained with a VGG, indicating that attribution methods at large are less effective for RESNET architectures. As a consequence, for MNIST and SUMMER, the overall best performing method is the residual, which is architecture independent and already performed well for VGG experiments. However, in general, the residual is not a good choice for an attribution map as intensity differences between classes do not generally correlate with feature importance.

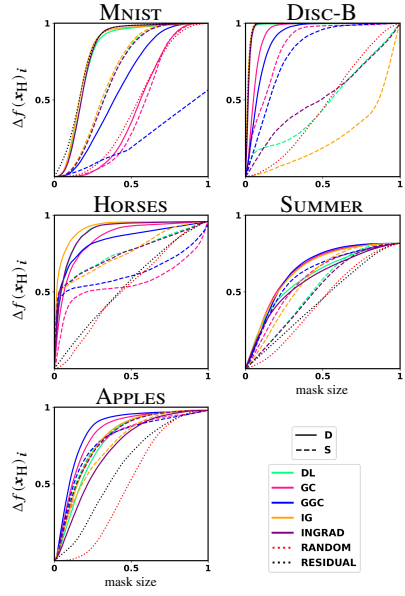

Fig. 2: Quantitative evaluation of discriminative (D - solid) and corresponding original (S for "single input" - dashed) attribution methods for MNIST, DISC-B, HORSES, SUMMER and APPLES using a **RESNET** architecture. Attributions for D and S methods are calculated as described in the methods section, with a zero baseline for all S methods that are explicit baseline methods, following standard practice. Corresponding D and S versions of the same method are shown in the same color. For each, we plot the average change of classifier prediction $\Delta f(\boldsymbol{x}_H)_i^k = f(\boldsymbol{x}_H)_i - f(\boldsymbol{x}_C)_i$ as a function of mask size $\boldsymbol{m} \in [0, 1]$. In addition we show performance of the two considered baselines: masks derived from random attribution maps (random - red, dotted) and mask derived from the residual of the real and counterfactual image (residual - black, dotted). On all considered datasets all versions of D attribution outperform their S counterparts, except for DL and INGRADS on the APPLES dataset.

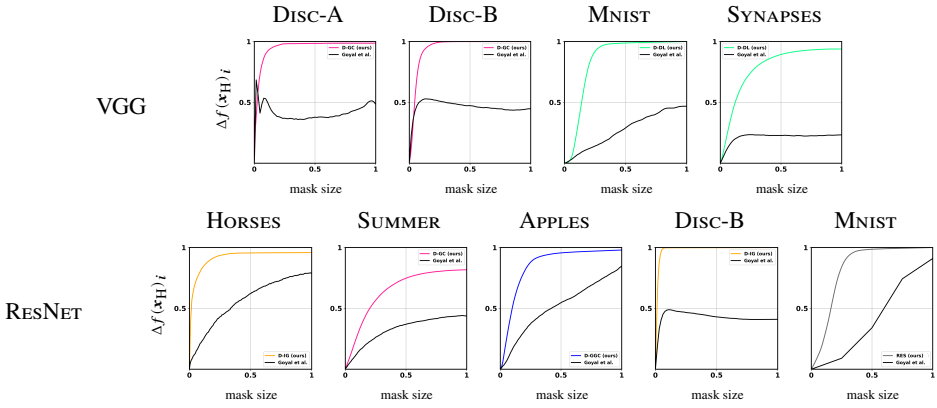

Fig. 3: Comparison of our method to the method presented by Goyal et al. [2] for the remaining four datasets using VGG architectures (top row, see main text Fig. 5 for the three natural image datasets) and all considered RESNET datasets (bottom row). For each dataset we show the DAPI scores for [2] and our best performing method averaged over all image pairs. For all datasets, we outperform [2] in terms of DAPI score. A curious finding is that the residual for RESNET- SUMMER has an almost perfect linear relationship between mask size and classification score, potentially caused by a focus on average color in the input image by the classifier.

B.1 Hypothetical Discriminators in the SYNAPSES Dataset

In order to understand the decision boundaries learned by the SYNAPSES classifier [1], we use DAPI on synapse samples and look for features in the set of the top 40 image pairs with smallest mask m for each pair of neurotransmitters nt_A, nt_B. We only accept sample pairs if the real image $\boldsymbol{x}_\mathrm{R}$ and the fake image $\boldsymbol{x}_\mathrm{C}$ are classified correctly, i.e. $f(\boldsymbol{x}_\mathrm{R}) > 0.9$ and $f(\boldsymbol{x}_\mathrm{C}) > 0.9$. For each pair we consider both directions $\mathrm{nt}_A^R \rightarrow \mathrm{nt}_B^C$ and $nt_B^R \rightarrow nt_A^C$ and note an observed change in features as a hypothetical discriminator, if the feature consistently changes in one direction, and is symmetrically reversed when going in the other. For example, if we observe a darkening of the cleft from $\mathrm{nt}_A^R \rightarrow \mathrm{nt}_B^C$, we require the cleft to become brighter going from $\mathrm{nt}_B^R \rightarrow \mathrm{nt}_A^C$. This ensures that we do not pick up on features that are not present in real synapses.

All symmetric, hypothetical discriminators can be seen in Fig. 11. *Brighter Cleft* refers to less electron density inside the synaptic cleft. *Add DCVs* is the addition of dense core vesicles, *Lower Density* means an overall reduction of content in the pre-synaptic site. *Fill Vesicles* means the darkening of the inside of vesicles. *Darker Cleft* refers to more electron density inside the synaptic cleft. *Remove DCVs* is the removal of dense core vesicles. *Remove small Vs* means the removal of vesicles with a small diameter. *Brighter DVs* refers to the brightening of the inside of medium size vesicles that we call *Dense Vesicles*. *Thinner Cleft* means that the distance between pre and post-synaptic partners is reduced. *!Circ. Vesicles* refers to a change of shape from circular to non-

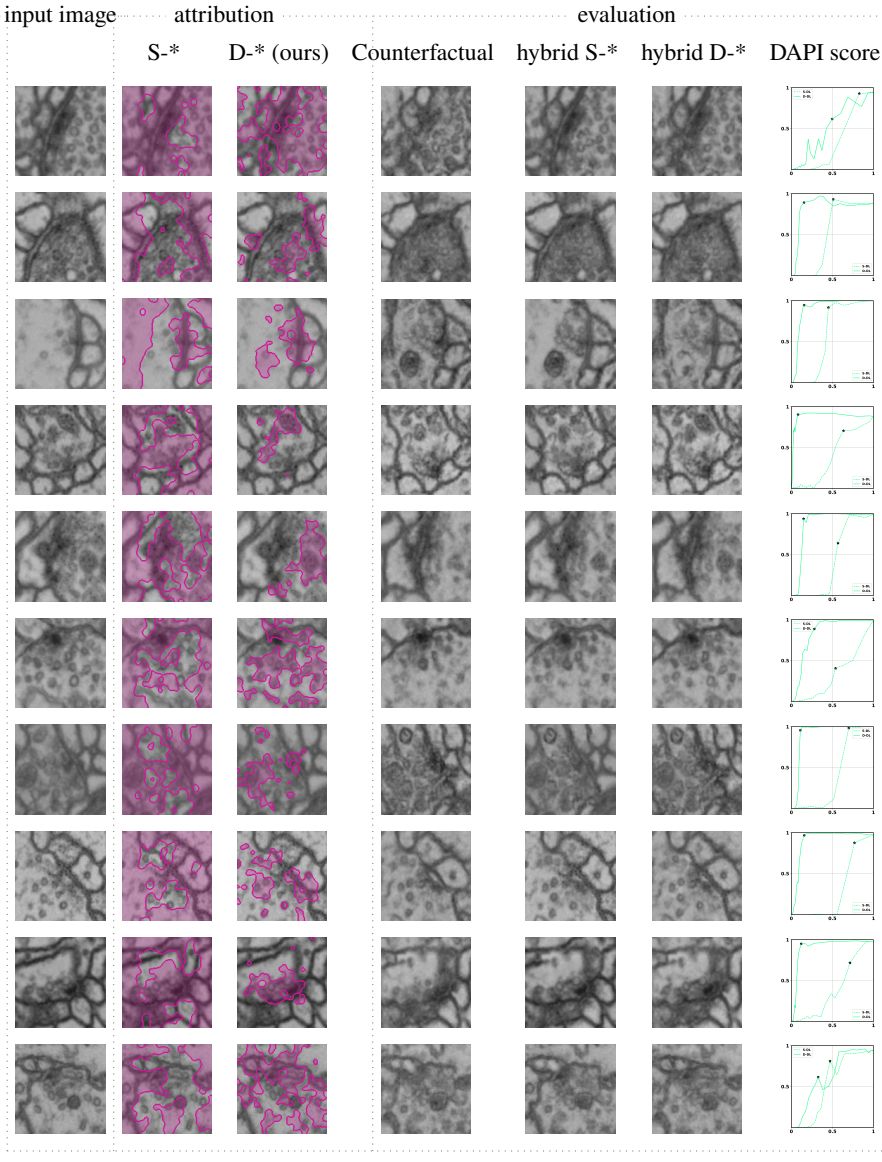

Fig. 4: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on SYNAPSES**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

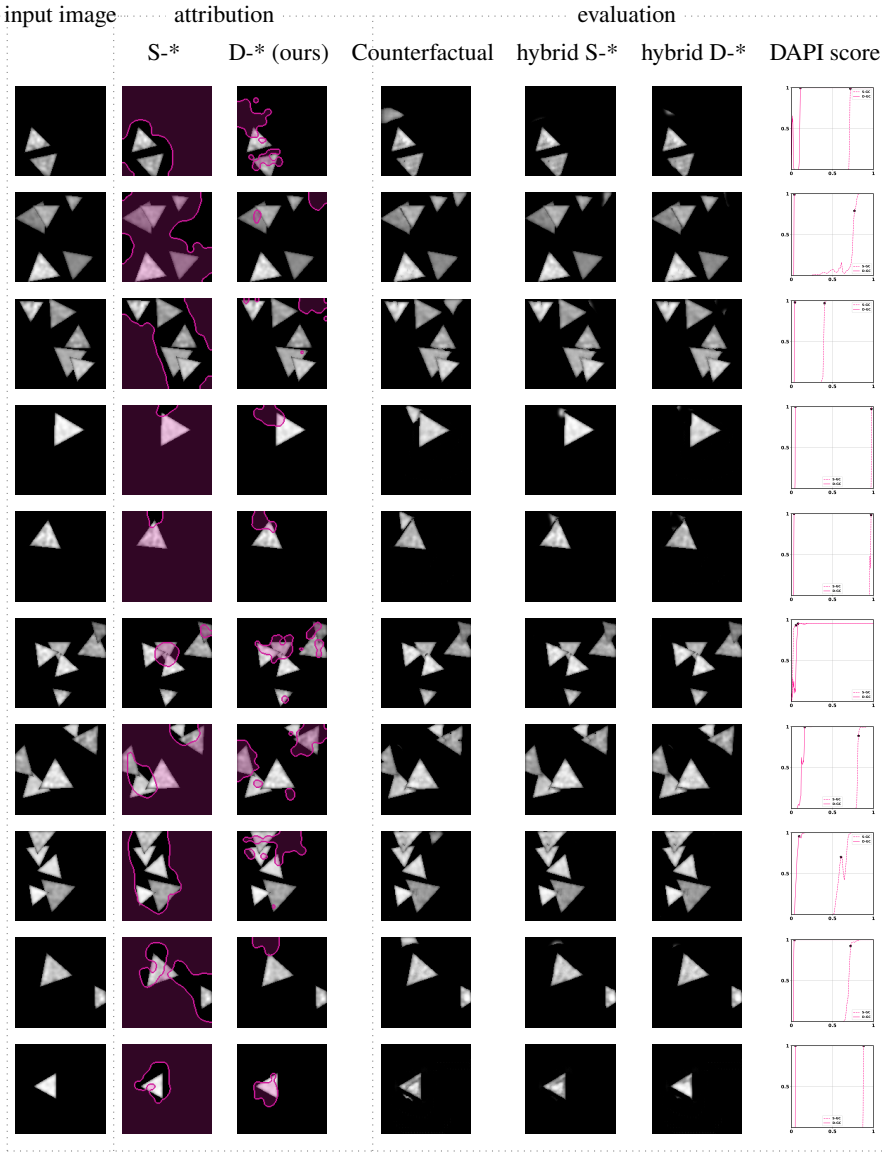

Fig. 5: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on Disc-A**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

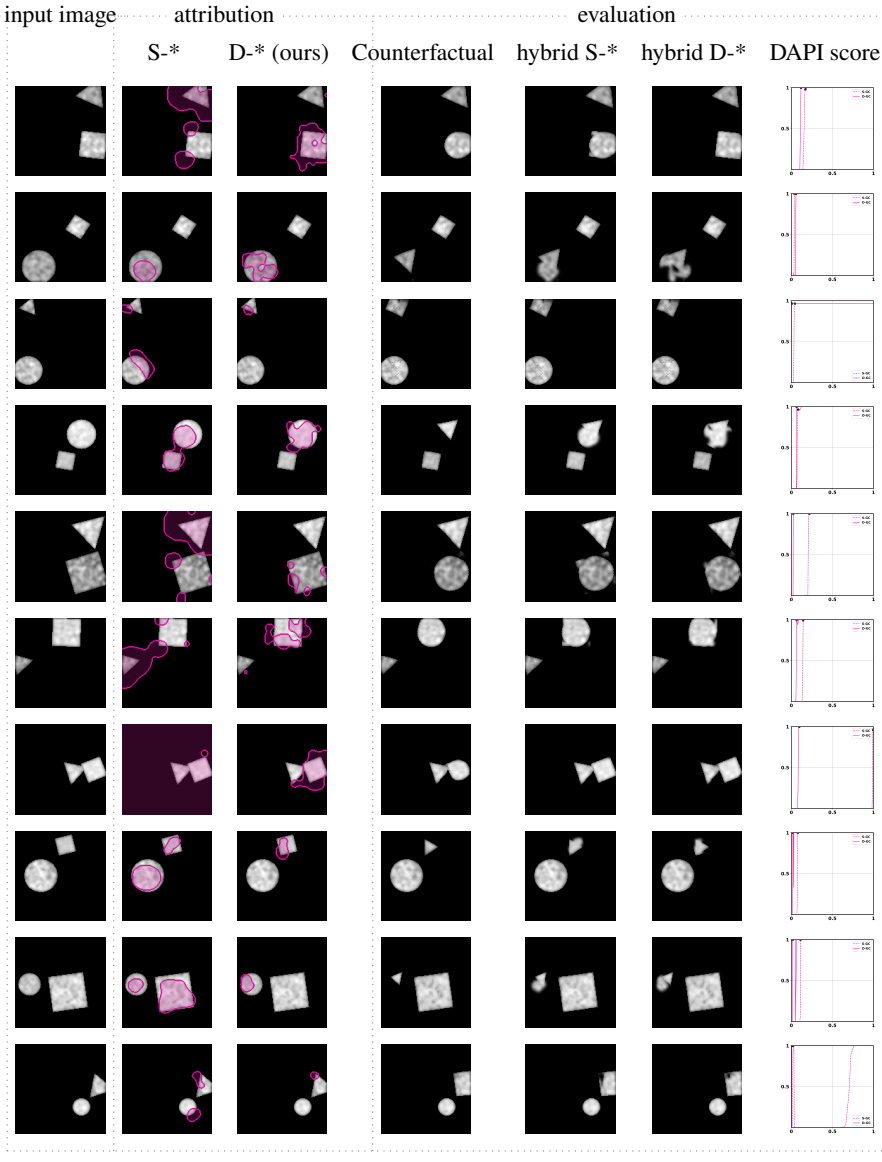

Fig. 6: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on Disc-B**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

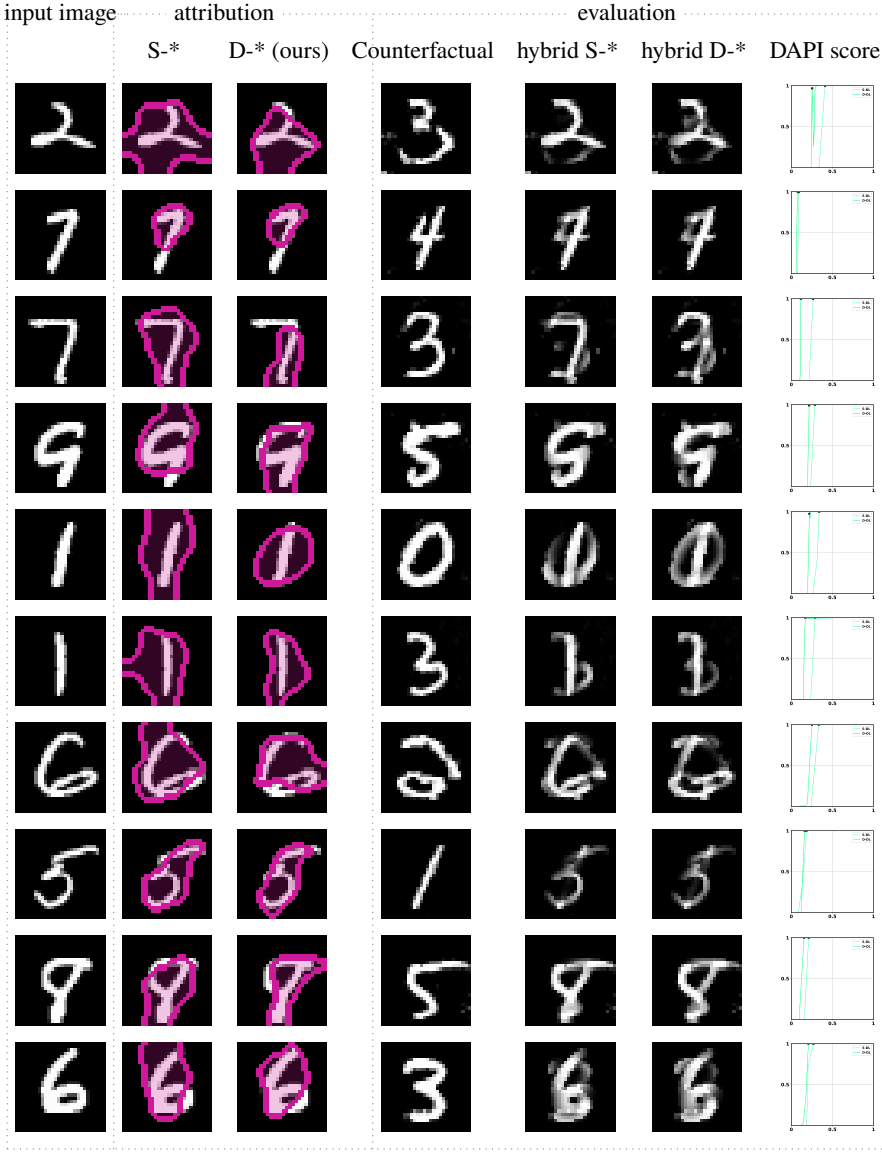

Fig. 7: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on MNIST**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

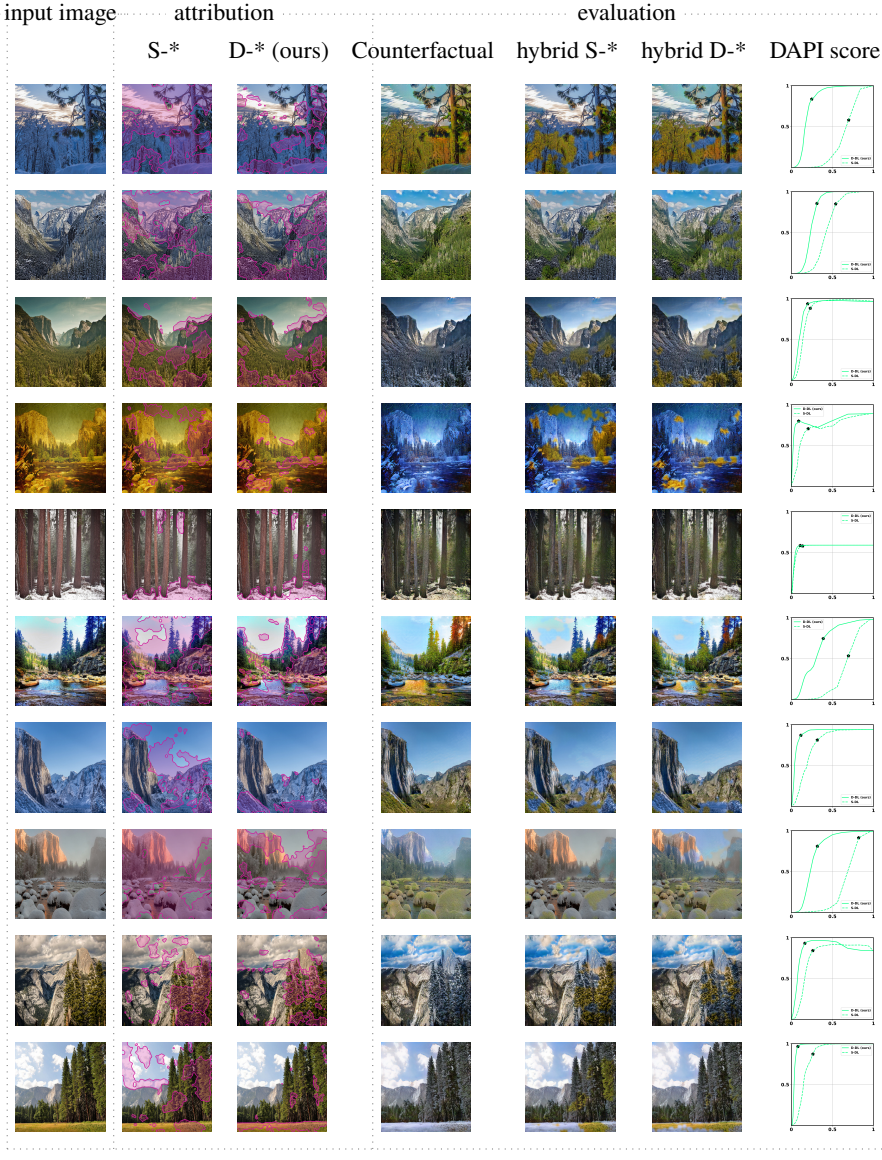

Fig. 8: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on SUMMER**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

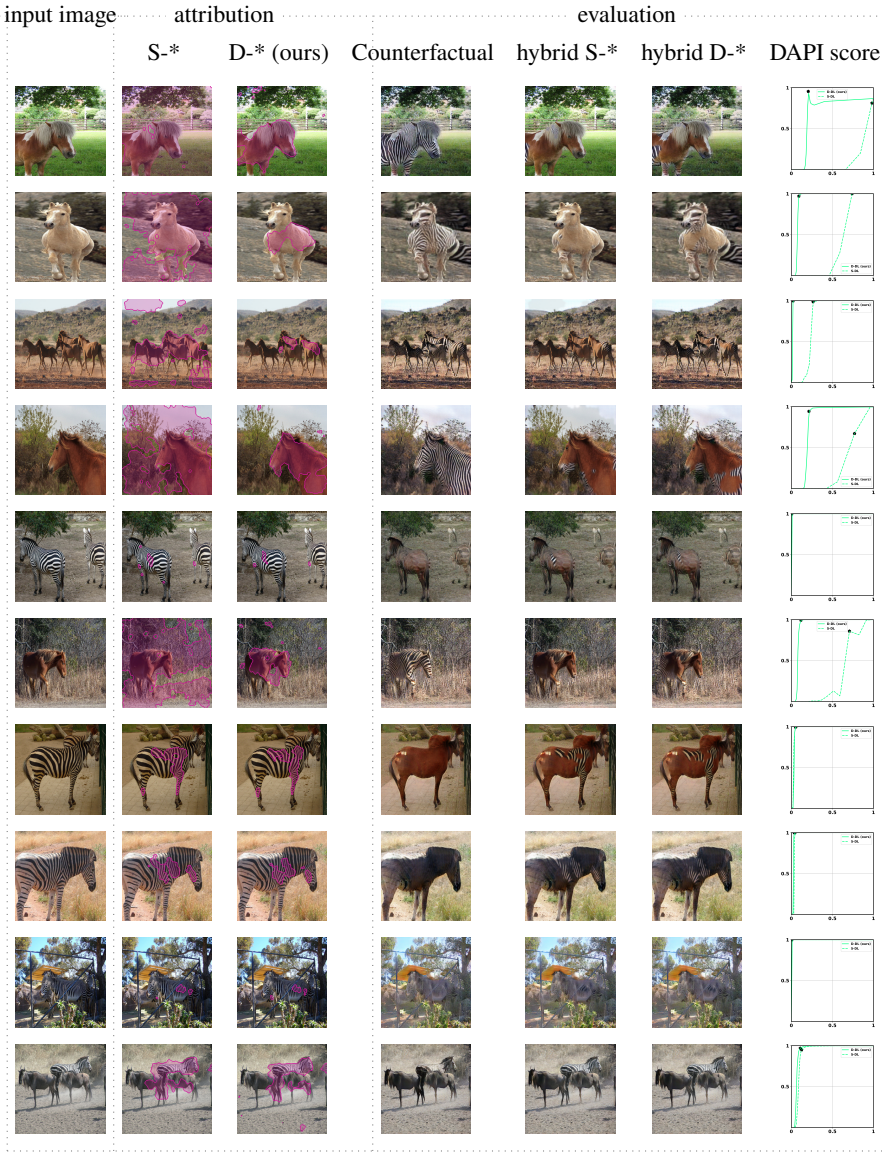

Fig. 9: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on Horses**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

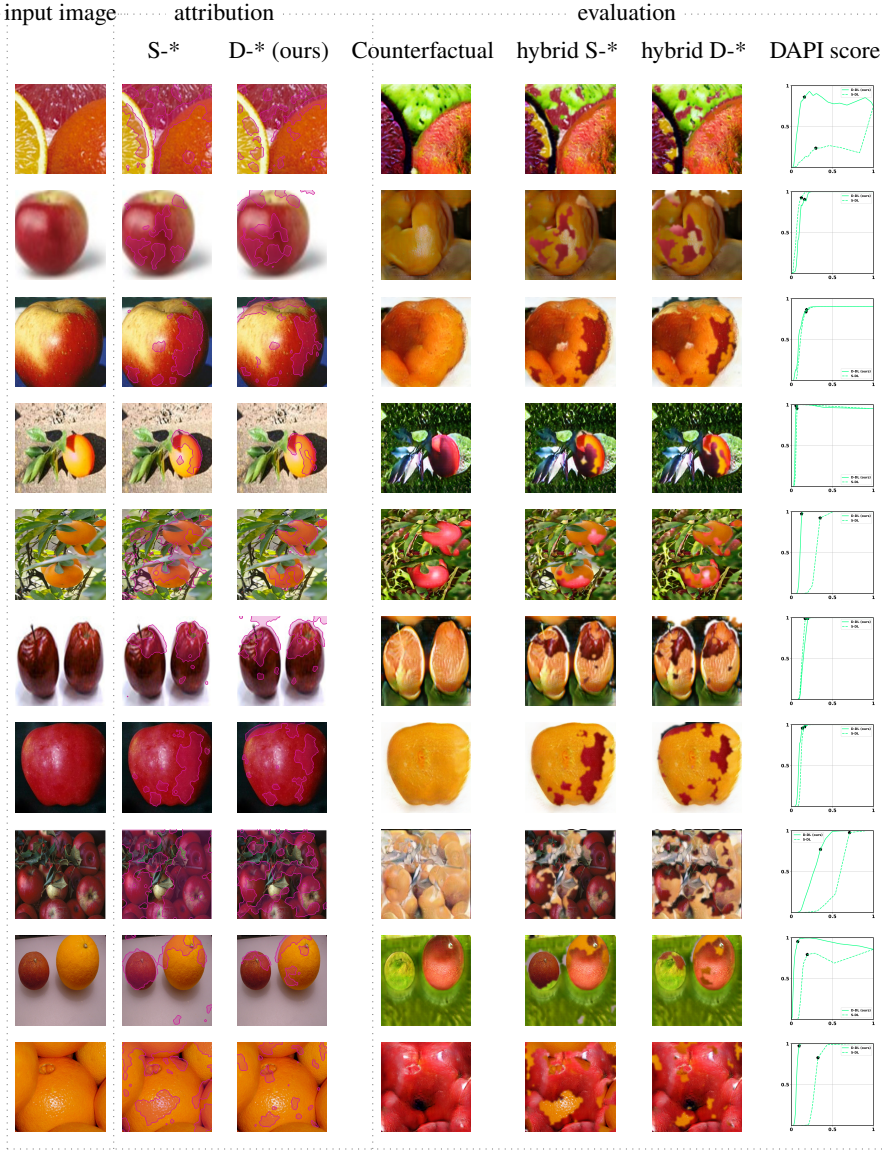

Fig. 10: **Randomly drawn qualitative samples** from the best performing method pairs (S: "single input", D: discriminative) **on Apples**. Shown in each row is the input image of class i, attribution maps according to S- and D- methods (best performing on the respective dataset), the counterfactual image of class j, and the hybrid images. The last column shows the single image DAPI score. Stars indicate the shown threshold (corresponding to the optimal DAPI score).

circular vesicles. *Remove PSDs* is the removal of post-synaptic densities. *Larger Vesicles* means the increase of vesicle diameter. *Add DVs* means there is an addition of *Dense Vesicles*. *Less Post-Synapses* describes a reduction in the number of post-synaptic partners.

The most notable findings are that the classical transmitters GABA, glutamate and acetylcholine look different in very subtle ways. For example we observe a consistent brightening of the inside of the synaptic cleft going from GABA to acetylcholine and slightly enlarged vesicles going from GABA to glutamate. Changes from acetylcholine to glutamate are a darker T-bar and a darker cleft. Other notable features are the apparent removal of post synaptic densities going from acetylcholine and glutamate to dopamine, in line with findings in mammalian cells [6]. A known discriminator we were able to rediscover is the addition of dense core vesicles when going from the classical transmitters to serotonin and octopamine. We leave confirmation of these hypotheses for future work.

C Datasets

The Disc dataset was specifically designed to highlight the advantage of discriminative attribution over vanilla attribution. In particular, the discriminatory feature of Disc-A is the parity of the number of triangles in the image. This feature is non-local and it is unclear what vanilla attribution is supposed to highlight. In Disc-B the classes are defined by the absence of a feature, another situation where vanilla attribution is not designed to give a sensible answer and will often highlight all objects in the image, providing little information to the user.

Disc-A For each image we randomly draw an even (class 0) or odd (class 1) number between one and six, indicating the number of triangles to generate. Each triangle has a random size between 20 and 40% of the image size of 128 pixels and a random position. In addition we draw a random intensity value between 120 and 200, a random rotation angle, and additive noise strength before applying Gaussian smoothing to generate different textures. We reject a sample if the fraction of foreground pixels and the total expected area of all shapes (assuming no overlap) is below 90%, thus avoiding strongly overlapping configurations.

Disc-B Similar to Disc-A, we draw a random position, intensity value, rotation and additive noise strength to generate images showing pairs of a triangle and a square, a disk and a square or a disk and a triangle. We reject a sample if the fraction of foreground pixels and the total expected area of all shapes (assuming no overlap) is below 90%.

D Comparison to Goyal et al.

The method presented in [2] is relevant in the context of our work, as it relies on a similar idea: finding swaps between images in order to teach humans discriminative features of two image classes. In addition, it does so without the need for generating counterfactuals with a cycle GAN, instead a real, random distractor image I' is picked for any given query

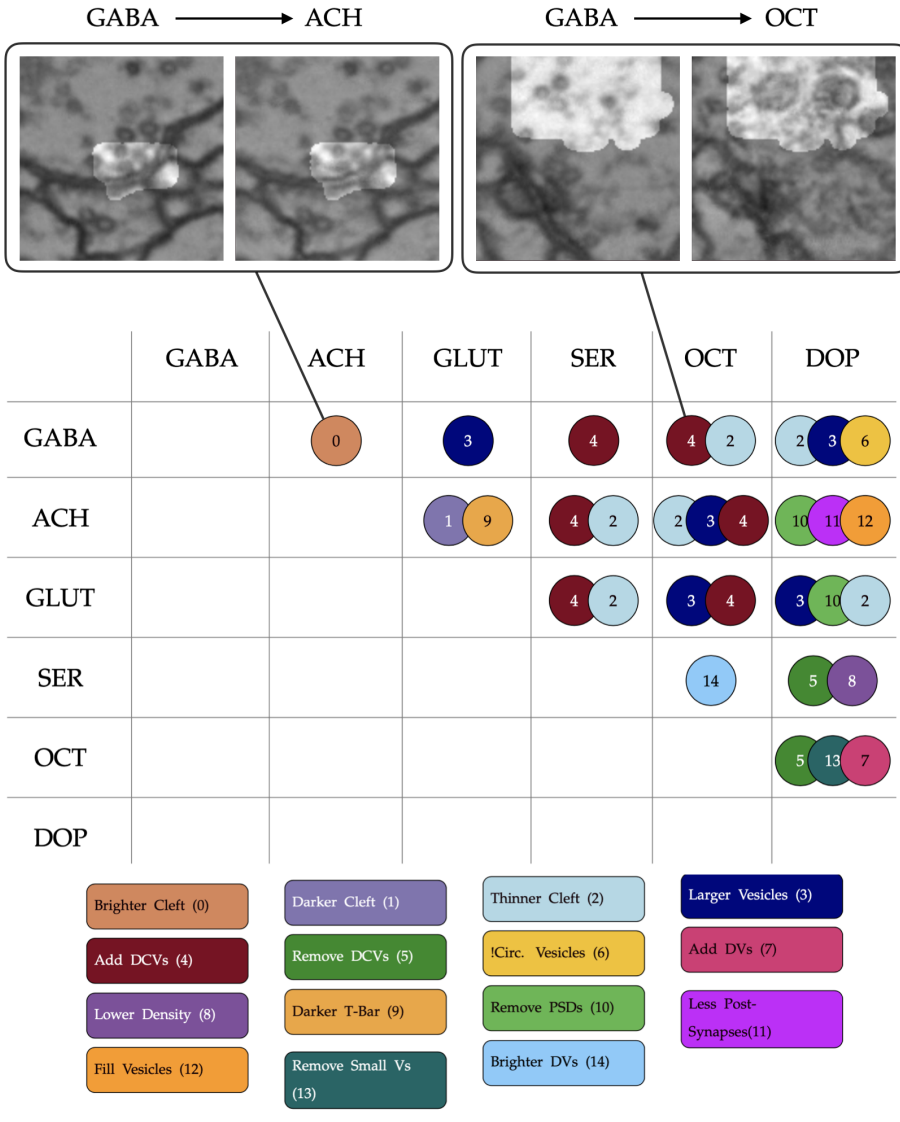

Fig. 11: Hypothetical Discriminator Matrix of features that change between images of two different neurotransmitter classes as discoverd by DAPI. We only show the direction from rows to columns in the upper triangle of the matrix, as we only consider symmetrically reversed features as valid hypotheses. Each box shows the feature change from the respective row title to the column title. See section B.1 for an explanation of each discovered feature. Inlets above the hypothesis matrix show example image pairs and highlighted regions lead to a change of classification decision in the indicated direction if swapped.

image I (see main text Fig. 5 for an example.). This is advantageous in terms of compute cost and can also be applied to datasets where training a cycle GAN is impossible. This is achieved by searching for (a minimal set of) spatial features in a query image I that, if replaced with spatial features from a distractor image I', leads to a hybrid I^* that is classified as I', i.e.:

$$f(I^*) = (\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I') \tag{1}$$

with $\circ$ the Hadamard product and $\mathbf{a}$, P a binary vector and a permutation matrix, which together define the feature replacements and are found via relaxation of a discrete optimization problem.

However, this approach comes with a number of downsides. In particular, optimizing for replacements in feature space is problematic, because for most DNN architectures, there is not a one to one correspondence between input pixels and spatial features due to overlapping field of views. Thus, swapping spatial features can produce different results to swapping the corresponding field of view in the input image. In order to show this, we compute the classification scores of the optimal $f(I^*)$ and calculate how often we do not reach the desired target class (defined by the distractor). In addition, we compute how often we fail to reach the desired target, when we use the same edits as found during optimization in feature space, but this time replace input patches corresponding to the spatial features (via upscaling of the feature map). The results are shown in Table 4 and reveal that there is a significant discrepancy between these two approaches. This is problematic if the goal is to educate humans on class relevant differences in the input images. Note that this issue is shared by any method that produces counterfactuals via optimization in feature space. In addition, we observed that the relaxation of the optimization in [2] can lead to cases where the target class is not reached, even if we only consider scores based on feature replacements, as it can get stuck in local minima (see Table 4 - SYNAPSES dataset). In our experiments, we also observed that a common failure case leading to significantly decreased DAPI scores is caused by the greedy relaxation. Because each edit is considered individually, there are cases where each single replacement has an identical score and no ordering of swaps can be derived. Thus a linear scan or a random replacement is the only option, unnecessarily increasing mask size and reducing DAPI score. Devising improved relaxations is thus a promising avenue for future work considering discriminative attribution from unpaired images.

E Code and Data Availability

All code, datasets, checkpoints, and instructions needed to reproduce the presented results are available at `anonymizedurl`.

Operation	Tensor Size
input image	(c, w, h)
`Conv2d`, size $(3,3)$	$(12, w, h)$
`BatchNorm2d`	$(12, w, h)$
`ReLU`	$(12, w, h)$
`Conv2d`, size $(3,3)$	$(12, w, h)$
`BatchNorm2d`	$(12, w, h)$
`ReLU`	$(12, w, h)$
`MaxPool2d`, size $(2,2)$	$(12, w/2, h/2)$
`Conv2d`, size $(3,3)$	$(24, w/2, h/2)$
`BatchNorm2d`	$(24, w/2, h/2)$
`ReLU`	$(24, w/2, h/2)$
`Conv2d`, size $(3,3)$	$(24, w/2, h/2)$
`BatchNorm2d`	$(24, w/2, h/2)$
`ReLU`	$(24, w/2, h/2)$
`MaxPool2d`, size $(2,2)$	$(24, w/4, h/4)$
`Conv2d`, size $(3,3)$	$(48, w/4, h/4)$
`BatchNorm2d`	$(48, w/4, h/4)$
`ReLU`	$(48, w/4, h/4)$
`Conv2d`, size $(3,3)$	$(48, w/4, h/4)$
`BatchNorm2d`	$(48, w/4, h/4)$
`ReLU`	$(48, w/4, h/4)$
`MaxPool2d`, size $(2,2)$	$(48, w/8, h/8)$
`Conv2d`, size $(3,3)$	$(96, w/8, h/8)$
`BatchNorm2d`	$(96, w/8, h/8)$
`ReLU`	$(96, w/8, h/8)$
`Conv2d`, size $(3,3)$	$(96, w/8, h/8)$
`BatchNorm2d`	$(96, w/8, h/8)$
`ReLU`	$(96, w/8, h/8)$
`MaxPool2d`, size $(2,2)$	$(96, w/16, h/16)$
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(k)

(a) VGG architecture used for the SYNAPSES ($c = 1, h = w = 128, k = 6$), DISC-A ($c = 1, h = w = 128, k = 2$), DISC-B ($c = 1, h = w = 128, k = 3$), HORSES ($c = 3, h = w = 256, k = 2$), SUMMER ($c = 3, h = w = 256, k = 2$) and APPLES ($c = 3, h = w = 256, k = 2$) dataset.

Operation	Tensor Size
input image	$(28, 28)$
`Conv2d`, size $(3,3)$	$(12, 28, 28)$
`BatchNorm2d`	$(12, 28, 28)$
`ReLU`	$(12, 28, 28)$
`Conv2d`, size $(3,3)$	$(12, 28, 28)$
`BatchNorm2d`	$(12, 28, 28)$
`ReLU`	$(12, 28, 28)$
`MaxPool2d`, size $(2,2)$	$(12, 14, 14)$
`Conv2d`, size $(3,3)$	$(24, 14, 14)$
`BatchNorm2d`	$(24, 14, 14)$
`ReLU`	$(24, 14, 14)$
`Conv2d`, size $(3,3)$	$(24, 14, 14)$
`BatchNorm2d`	$(24, 14, 14)$
`ReLU`	$(24, 14, 14)$
`MaxPool2d`, size $(2,2)$	$(24, 7, 7)$
`Conv2d`, size $(3,3)$	$(48, 7, 7)$
`BatchNorm2d`	$(48, 7, 7)$
`ReLU`	$(48, 7, 7)$
`Conv2d`, size $(3,3)$	$(48, 7, 7)$
`BatchNorm2d`	$(48, 7, 7)$
`ReLU`	$(48, 7, 7)$
`Conv2d`, size $(3,3)$	$(96, 7, 7)$
`BatchNorm2d`	$(96, 7, 7)$
`ReLU`	$(96, 7, 7)$
`Conv2d`, size $(3,3)$	$(96, 7, 7)$
`BatchNorm2d`	$(96, 7, 7)$
`ReLU`	$(96, 7, 7)$
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(10)

(b) VGG architecture used for the MNIST dataset.

Table 1: VGG classifier network architectures.

Operation	Tensor Size
input image	(c, w, h)
`Conv2d`, size $(3, 3)$	$(12, w, h)$
`BatchNorm2d`	$(12, w, h)$
`ReLU`	$(12, w, h)$
`ResBlock`, stride $(2, 2)$	$(12, w/2, h/2)$
`ResBlock`	$(12, w/2, h/2)$
`ResBlock`, stride $(2, 2)$	$(24, w/4, h/4)$
`ResBlock`	$(24, w/4, h/4)$
`ResBlock`, stride $(2, 2)$	$(48, w/8, h/8)$
`ResBlock`	$(48, w/8, h/8)$
`ResBlock`, stride $(2, 2)$	$(96, w/16, h/16)$
`ResBlock`	$(96, w/16, h/16)$
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(k)

(a) ResNet architecture used for the Disc-A ($c = 1, h = w = 128, k = 2$), Disc-B ($c = 1, h = w = 128, k = 3$), Horses ($c = 3, h = w = 256, k = 2$), Summer ($c = 3, h = w = 256, k = 2$) and Apples ($c = 3, h = w = 256, k = 2$) dataset.

Operation	Tensor Size
input image	$(28, 28)$
`Conv2d`, size $(3, 3)$	$(12, 28, 28)$
`BatchNorm2d`	$(12, 28, 28)$
`ReLU`	$(12, 28, 28)$
`ResBlock`, stride $(2, 2)$	$(12, 14, 14)$
`ResBlock`	$(12, 14, 14)$
`ResBlock`, stride $(2, 2)$	$(24, 7, 7)$
`ResBlock`	$(24, 7, 7)$
`ResBlock`, stride $(2, 2)$	$(48, 3, 3)$
`ResBlock`	$(48, 3, 3)$
`ResBlock`, stride $(2, 2)$	$(96, 1, 1)$
`ResBlock`	$(96, 1, 1)$
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(4096)
`ReLU`	(4096)
`Dropout`	(4096)
`Linear`	(10)

(b) ResNet architecture used for the Mnist dataset.

Table 2: ResNet classifier network architectures.

Dataset	D-IG	D-DL	D-INGR.	D-GC	D-GGC	RES.	IG	DL	INGR.	GC	GGC	RND.	GOYAL
Mnist	0.82	0.8	0.81	0.44	0.6	**0.83**	0.68	0.66	0.66	0.42	0.23	0.46	0.41
Disc-B	**0.98**	0.98	0.98	0.94	0.91	0.98	0.26	0.48	0.52	0.86	0.81	0.47	0.43
Horses	**0.92**	0.89	0.89	0.85	0.84	0.55	0.76	0.76	0.75	0.56	0.63	0.53	0.56
Summer	0.64	0.6	0.58	**0.65**	0.65	0.45	0.58	0.52	0.51	0.58	0.61	0.43	0.33
Apples	0.79	0.77	0.72	0.83	**0.85**	0.61	0.75	0.79	0.79	0.77	0.78	0.5	0.51

Table 3: Summary of DAPI scores for ResNet architectures on Disc, Mnist, Horses, Summer and Apples corresponding to Fig. 2. Best results are highlighted.

Dataset	Total	Feature	Input
MNIST	9004	1 (0.01%)	2706 (30%)
DISC-B	4876	1 (0.02%)	1937 (39%)
SYNAPSES	1875	214 (11%)	888 (47%)

Table 4: Number of times [2] reaches the target class during optimization for the three datasets MNIST, DISC-B and SYNAPSES (non-binary classification tasks). *Total* shows the total number of considered samples. *Features* shows how many of those did **not** reach the target label at any point during the optimization when replacing spatial features. *Input* shows how many of *Total* did **not** reach the target label at any point during the optimization when replacing input pixels corresponding to the spatial feature edits found during optimization. Failure cases increase significantly when replacing input pixels.

References

1. Eckstein, N., Bates, A.S., Du, M., Hartenstein, V., Jefferis, G.S., Funke, J.: Neurotransmitter classification from electron microscopy images at synaptic sites in drosophila. BioRxiv (2020) 1, 4
2. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: International Conference on Machine Learning. pp. 2376–2384. PMLR (2019) 4, 12, 14, 17
3. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017) 1
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 1
5. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017) 1
6. Uchigashima, M., Ohtsuka, T., Kobayashi, K., Watanabe, M.: Dopamine synapse is a neuroligin-2–mediated contact between dopaminergic presynaptic and gabaergic postsynaptic structures. Proceedings of the National Academy of Sciences **113**(15), 4206–4211 (2016) 12
7. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017) 1