On the explainable properties of 1-Lipschitz Neural Networks: An Optimal Transport Perspective

Mathieu Serrurier[1] **Franck Mamalet**[2] **Thomas Fel**[3,4] **Louis Béthune**[1]

Thibaut Boissin[2]

[1]Université Paul-Sabatier, IRIT, Toulouse, France
[2]Institut de Recherche Technologique Saint-Exupéry, Toulouse, France
[3]Carney Institute for Brain Science,Brown University, USA
[4]Innovation & Research Division, SNCF, France

Abstract

Input gradients have a pivotal role in a variety of applications, including adversarial attack algorithms for evaluating model robustness, explainable AI techniques for generating Saliency Maps, and counterfactual explanations. However, Saliency Maps generated by traditional neural networks are often noisy and provide limited insights. In this paper, we demonstrate that, on the contrary, the Saliency Maps of 1-Lipschitz neural networks, learnt with the dual loss of an optimal transportation problem, exhibit desirable XAI properties: They are highly concentrated on the essential parts of the image with low noise, significantly outperforming state-of-the-art explanation approaches across various models and metrics. We also prove that these maps align unprecedentedly well with human explanations on ImageNet. To explain the particularly beneficial properties of the Saliency Map for such models, we prove this gradient encodes both the direction of the transportation plan and the direction towards the nearest adversarial attack. Following the gradient down to the decision boundary is no longer considered an adversarial attack, but rather a counterfactual explanation that explicitly transports the input from one class to another. Thus, Learning with such a loss jointly optimizes the classification objective and the alignment of the gradient , i.e. the Saliency Map, to the transportation plan direction. These networks were previously known to be certifiably robust by design, and we demonstrate that they scale well for large problems and models, and are tailored for explainability using a fast and straightforward method.

1 Introduction

The Lipschitz constant of a function expresses the extent to which the output may vary for a small shift in the input. As a composition of numerous functions, the Lipschitz constant of a neural network can be arbitrarily high, particularly when trained for a classification task [7]. Adversarial attacks [40] exploit this weakness by selecting minor modifications, such as imperceptible noise, for a given example to change the predicted class and deceive the network. Consequently, Saliency Maps [53] – gradient of output with respect to the input –, serve as the basis for most adversarial attacks and often highlight noisy patterns that fool the model instead of meaningful modifications, rendering them generally unsuitable for explaining model decisions. Therefore, several methods requiring more complex computations, such as SmoothGrad [55], Integrated Gradient [58], or Grad-CAM [50], have been proposed to provide smoother explanations. Recently, the XAI community has investigated the link between explainability and robustness and proposed methods and metrics

Accepted to the Optimal Transport and Machine Learning Workshop at NeurIPS 2023.

accordingly [30, 11, 37, 48]. However, the reliability of those automatic metrics can be compromised by artifacts introduced by the baselines [30, 57, 26, 33, 25], and there is no conclusive evidence demonstrating their correlation with the human understanding of explanations. To address this, a study by [18] suggests completing those metrics with the alignment between attribution methods and human feature importance using the ClickMe dataset [38].

In [51], authors propose to address the weakness with respect to adversarial attacks by training 1-Lipschitz constrained neural networks with a loss that is the dual of an optimal transport optimization problem, called hKR (Eq. 1). The resulting models have been shown to be robust with a certifiable margin. We refer to these networks as Optimal Transport Neural Networks (OTNN) hereafter.

In this paper, we demonstrate that OTNNs exhibit valuable explainability properties. Our experiments reveal that OTNN Saliency Maps significantly outperform various attribution methods for unconstrained networks across all tested state-of-the-art Explainable XAI metrics. This improvement is consistent across toy datasets, large image datasets, binary, and multiclass problems. Qualitatively, OTNN Saliency Maps concentrate on crucial image regions and generate less noise than maps of unconstrained networks, as illustrated in Figure 1. Figure 1.c presents the Saliency Maps of an OTNN based on ResNet50 alongside its unconstrained vanilla counterpart, both trained on ImageNet [15]. In the unconstrained case, the Saliency Maps appear sparse and uninformative, with the most critical pixels often located outside the subject. Conversely, the OTNN Saliency Map is less noisy and highlights significant image features. This distinction is emphasized in Figure 1.d), comparing the feature visualization of the two models. Feature visualization extracts the inverse prototypical image for a given class using gradient ascent [43, 42]. The results for vanilla ResNet are noticeably noisy, making class identification difficult. In contrast, feature visualization with OTNN yields clearer results, displaying easily identifiable features and classes (e.g., goldfish, butterfly, and medusa). Furthermore, modifying the image following the gradient direction provides an interpretable method for altering the image to change its class. Pictures 1.a) and 1.b) display the original image, the gradient direction, and the transformation following the gradient direction (refer to Section 4 for details). We observe explicit structural changes in the images, transforming them into an example of another class in both multiclass (MNIST) and high-resolution multi-labels cases (e.g., smile/not smile and blond hair/not blond hair classification). Lastly, a large-scale human experiment demonstrates that these maps are remarkably aligned with human attribution on ImageNet (Fig. 3).

We provide a theoretical justification for the well-behaved OTNN Saliency Maps. Building upon the fact that OTNNs encode the dual formulation of the optimal transport problem, we prove that the gradient of the optimal solution at a given point $\boldsymbol{x}$ is both (i) in the direction of the nearest adversarial example on the decision boundary, and (ii) in the direction of the image of x according to the underlying transport plan. This implies that adversarial attacks for an OTNN are equivalent to traversing the optimal transport path which can be achieved by following the gradient. Consequently, the resulting modification serves both as an adversarial attack and a counterfactual explanation, explaining why the decision was A and not B [35]. An optimal transport plan between two classes can be interpreted as a global approach for constructing counterfactuals, as suggested in [10, 14]. These counterfactuals may not correspond to the smallest transformation for a given input sample but rather the smallest transformation on average when pairing points from two classes. A consequence of this property is that the Saliency Map of an OTNN for an image indicates the importance of each pixel in the modifications needed to change the class. It is worth noting that several methods based on GAN [31] or causality penalty [32] produce highly realistic counterfactual images. However, the objective of our paper is not to compete with the quality of these results, but rather to demonstrate that OTNN Saliency Maps possess both theoretical and empirical foundations as counterfactual explanations.

We summarize our contributions as follows: first, after introducing the background on OTNN and XAI, we establish several properties of the gradient of an OTNN with respect to adversarial attacks, decision boundaries, and optimal transport. Second, we establish that the optimal transport properties of OTNN's gradient lead to a reinterpretation of adversarial attacks as counterfactual explanations, consequently endowing the Saliency Map with the favorable XAI properties inherent in these explanations. Third, our experiments support the theoretical results, showing that metric scores are higher for most of the XAI methods on OTNN compared to unconstrained neural networks. Additionally, we find that the Saliency Map for OTNN achieves top-ranked scores on XAI metrics compared to more sophisticated XAI methods, and is equivalent to Smoothgrad. Lastly, drawing from [18], we emphasize that OTNNs are naturally and remarkably aligned with human explanations, and we present several examples of gradient-based counterfactuals obtained with OTNNs.

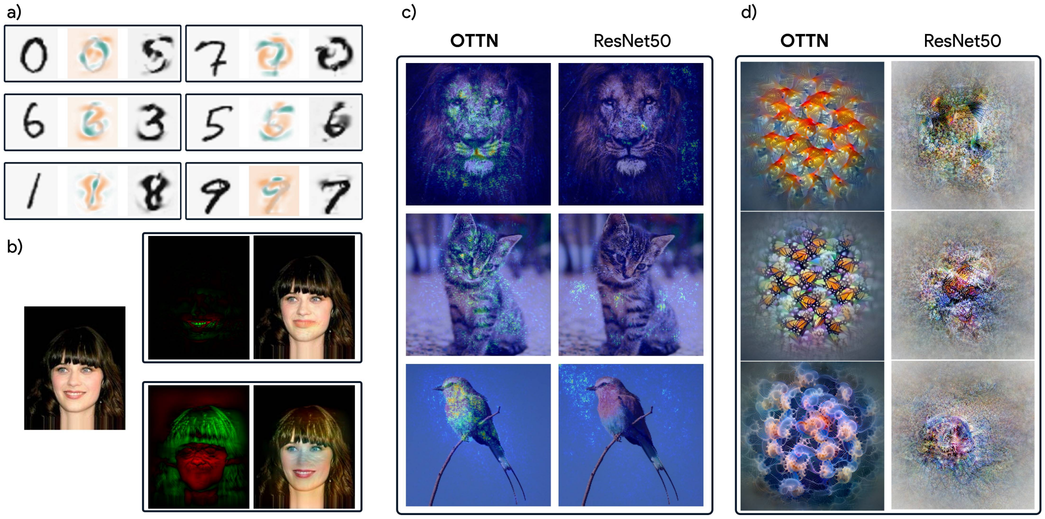


Figure 1: **Illustration of the beneficial properties of OTNN gradients.** Examples **a)** and **b)** show that the gradients naturally provide a direction that enables the generation of adversarial images - a theoretical justification based on optimal transport is provided in Section 3. By applying the gradient $\boldsymbol{x}' = \boldsymbol{x} - t\nabla_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})$ to the original image $\boldsymbol{x}$ (on the left), any digit from MNIST can be transformed into its counterfactual $\boldsymbol{x}'$ (e.g., turning a 0 into a 5). In **b)**, we illustrate that this approach can be applied to larger datasets, such as Celeb-A, by creating two counterfactual examples for the closed-mouth and blonde classes. In **c)**, we compare the Saliency Map of a classical model with those of OTNN gradients, which are more focused on relevant elements. Finally, in **d)**, we show that following the gradients of OTNN could generate convincing feature visualizations that ease the understanding of the model's features.

2 Related work

1-Lipschitz Neural network and optimal transport: Consider a classical supervised machine learning binary classification problem on $(\Omega, \mathcal{F}, \mathbb{P})$ – the underlying probability space – where Ω is the sample space, $\mathcal{F}$ is a σ-algebra on Ω, and $\mathbb{P}$ is a probability measure on $\mathcal{F}$. We denote the input space $\mathcal{X} \subseteq \mathbb{R}^d$ and the output space $\mathcal{Y} = \{\pm 1\}$. Let input data $\boldsymbol{x} : \Omega \to \mathcal{X}$ and target label $y : \Omega \to \mathcal{Y}$ are random variables with distributions $P_{\boldsymbol{x}}$, P_y, respectively. The joint random vector $(\boldsymbol{x}, y)$ on $(\Omega, \mathcal{F})$ has a joint distribution P defined over the product space $\mathcal{X} \times \mathcal{Y}$. Moreover, let $\mu = P(\boldsymbol{x}|y = 1)$ and $\nu = P(\boldsymbol{x}|y = -1)$ the conditional probability distributions of $\boldsymbol{x}$ given the true label. We assume that the supports of Ω, μ and ν are compact sets.

A function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}$ is a 1-Lipschitz functions over $\mathcal{X}$ (denoted $Lip_1(\mathcal{X})$) if and only if $\forall(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X}^2, ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{z})|| \leq ||\boldsymbol{x} - \boldsymbol{z}||$. 1-Lipschitz neural networks have received a lot of attention, especially due to the link with adversarial attacks. They provide certifiable robustness guarantees [27, 44], improve the generalizations [56] and the interpretability of the model [60]. The simplest way to constrain a network to be in $Lip_1(\mathcal{X})$ is to impose this 1-Lipschitz property to each layer. Frobenius normalization [49], or spectral normalization [41] can be used for linear layers, and can also be extended, in some situations, to orthogonalization [36, 1].

Optimal transport, 1-Lipschitz neural networks, and binary classification were first associated in the context of Wasserstein GAN (WGAN) [6]. The discriminator of a WGAN is the solution to the Kantorovich-Rubinstein dual formulation of the 1-Wasserstein distance [62], and it can be regarded as a binary classifier with a carefully chosen threshold. Nevertheless, it has been demonstrated in [51] that this type of classifier is suboptimal, even on a toy dataset. In the same paper, the authors address the suboptimality of the Wasserstein classifier by introducing the hKR loss $\mathcal{L}^{hKR}$, which adds a hinge regularization term to the Kantorovich-Rubinstein optimization objective :

$$\mathcal{L}^{hKR}_{\lambda,m}(\boldsymbol{f}) = \underset{\boldsymbol{x}\sim\nu}{\mathbb{E}}[\boldsymbol{f}(\boldsymbol{x})] - \underset{\boldsymbol{x}\sim\mu}{\mathbb{E}}[\boldsymbol{f}(\boldsymbol{x})] + \lambda \underset{(\boldsymbol{x},y)\sim P}{\mathbb{E}}(m - y\boldsymbol{f}(\boldsymbol{x}))_+ \quad (1)$$

where $m > 0$ is the margin, and $(z)_+ = max(0, z)$. We note $\boldsymbol{f}^*$ the optimal minimizer of $\mathcal{L}^{hKR}_{\lambda,m}$. The classification is given by the sign of $\boldsymbol{f}^*$. In the following, the 1-Lipschitz neural networks that minimize $\mathcal{L}^{hKR}_{\lambda,m}$ will be denoted as OTNN. Given a function $\boldsymbol{f}$, a classifier based on $\text{sign}(\boldsymbol{f})$ and an example $\boldsymbol{x}$, an adversarial example is defined as follows:

$$adv(\boldsymbol{f}, \boldsymbol{x}) = \underset{\boldsymbol{z}\in\mathcal{X}}{\arg\min} \parallel \boldsymbol{x} - \boldsymbol{z} \parallel \quad s.t. \;\; \text{sign}(\boldsymbol{f}(\boldsymbol{z})) \neq \text{sign}(\boldsymbol{f}(\boldsymbol{x})). \tag{2}$$

Since $\boldsymbol{f}^*$ is a 1-Lipschitz function, $|\boldsymbol{f}^*(\boldsymbol{x})|$ is a certifiable lower bound of the robustness of the classification of $\boldsymbol{x}$ (i.e. $\forall \boldsymbol{x} \in \mathcal{X}, |\boldsymbol{f}^*(\boldsymbol{x})| \leq ||\boldsymbol{x} - adv(\boldsymbol{f}^*, \boldsymbol{x})||$). The function $\boldsymbol{f}^*$ has the following properties [51] ***(i)*** if the supports of μ and ν are disjoints (separable classes) with a minimal distance of $\epsilon > 0$, then for $m < 2\epsilon$, f^* achieves 100% accuracy; ***(ii)*** minimizing $\mathcal{L}^{hKR}$ is still the dual formulation of an optimal transport problem (see appendix for more details).

Explainability and metrics: Attribution methods aim to explain the prediction of a deep neural network by pointing out input variables that support the prediction – typically pixels or image regions for images – which lead to importance maps. Saliency [53] was the first proposed white-box attribution method and consists of back-propagating the gradient from the output to the input. The resulting absolute gradient heatmap indicates which pixels affect the most the decision score. However, this family of methods suffers from problems inherent to the gradients of standard models. These methods were then followed by a plethora of other methods as Integrated Gradient [58], SmoothGrad [55], Grad-cam [50], Input Gradient [4].

However, it is becoming increasingly clear that current methods raise many issues [2, 28, 54] such as confirmation bias: it is not because the explanations make sense to humans that they reflect the evidence of the prediction. To address this challenge, a large number of metrics were proposed to provide objective evaluations of the quality of explanations. Deletion and Insertion methods [45] evaluate the drop in accuracy when important pixels are replaced by a baseline. μFidelity method [8] evaluates the correlation between the sum of importance scores of pixels and the drop of the score when removing these pixels. Other metrics consider properties such as generalizability, consistency [19], or stability [67, 8] of explanation methods. A recent approach [38] aims to evaluate the alignment between attribution methods and human feature importance across 200,000 unique ImageNet images (called ClickMe dataset). The alignment between DNN Saliency and human explanations is quantified using the mean Spearman correlation, normalized by the average inter-rater alignment of humans.

These works on explainability metrics have also initiated the emergence of links between the robustness of models and the quality of their explanations [13, 65]. In particular,[19] claimed that 1-Lipschitz networks explanations have better metrics scores. But this study was not on OTNNs and was limited to their proposed metrics.

To end with, recent literature is focusing on counterfactual explanations [63, 61] methods, providing information on "why the decision was A and not B". Several properties are desirable for these counterfactual explanations[61]: Validity (close sample and in another class), Actionability, Sparsity, Data Manifold closeness, and Causality. The three last properties are generally not covered by standard adversarial attacks and complex methods have been proposed [23, 47, 64]. Since often a causal model is hard to fully-define, recent papers [10, 14] have proposed a definition of counterfactual based on optimal transport easier to compute and that can sometimes coincide with causal model based ones. We will rely on this theoretical definition of counterfactuals.

3 Theoretical properties of OTNN gradient

In this section, we extend the properties of the OTNNs to the explainability framework, all the proofs are in the appendix A.1. We note π the optimal transport plan corresponding to the minimizer of $\mathcal{L}^{hKR}_{\lambda,m}$. In the most general setting, π is a joint distribution over μ, ν pairs. However when μ and ν admit a density function [46] with respect to Lebesgue measure, then the joint density describes a deterministic mapping, i.e. a Monge map. Given $\boldsymbol{x} \sim \mu$ (resp. ν) we note $\boldsymbol{z} = \boldsymbol{\gamma}_\pi(\boldsymbol{x}) \in \nu$ (resp. μ) the image of $\boldsymbol{x}$ with respect to π. When π is not deterministic (on real datasets that are defined as a discrete collection of Diracs), we take $\boldsymbol{\gamma}_\pi(\boldsymbol{x})$ as the point of maximal mass with respect to π.

Proposition 1 (Transportation plan direction) *Let $\boldsymbol{f}^*$ an optimal solution minimizing the $\mathcal{L}^{hKR}_{\lambda,m}$. Given $\boldsymbol{x} \sim \mu$ (resp. ν) and $\boldsymbol{z} = \boldsymbol{\gamma}_\pi(\boldsymbol{x})$, then $\exists t \geq 0$ (resp. $t \leq 0$) such that $\boldsymbol{\gamma}_\pi(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})$ almost surely.*

This proposition also holds for the Kantorovich-Rubinstein dual problem without hinge regularization, demonstrating that for $\boldsymbol{x} \sim P$, the gradient $\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})$ indicates the direction in the transportation plan almost surely.

Proposition 2 (Decision boundary) *Let μ and ν two distributions with disjoint supports with minimal distance ϵ and $\boldsymbol{f}^*$ an optimal solution minimizing the $\mathcal{L}^{hKR}_{\lambda,m}$ with $m < 2\epsilon$. Given $\boldsymbol{x} \sim P$, $\boldsymbol{x}_\delta = \boldsymbol{x} - \boldsymbol{f}^*(\boldsymbol{x})\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x}) \in \partial\mathcal{X}$ where $\partial\mathcal{X} = \{\boldsymbol{x}' \in \mathcal{X} | \boldsymbol{f}^*(\boldsymbol{x}') = 0\}$ is the decision boundary (i.e. the 0 level set of $\boldsymbol{f}^*$).*

Experiments suggest this probably remains true when the supports of μ and ν are not disjoint. Prop. 2 proves that for an OTNN $\boldsymbol{f}$ learnt by minimizing the $\mathcal{L}^{hKR}_{\lambda,m}$, $|\boldsymbol{f}(\boldsymbol{x})|$ provides a tight robustness certificate. A direct consequence of 2, is that t defined in 1 is such that $|t| \geq |\boldsymbol{f}^*(\boldsymbol{x})|$.

Corollary 1 *Let μ and ν two separable distributions with minimal distance ϵ and $\boldsymbol{f}^*$ an optimal solution minimizing the $\mathcal{L}^{hKR}_{\lambda,m}$ with $m < 2\epsilon$, given $\boldsymbol{x} \sim P$, $adv(\boldsymbol{f}^*, \boldsymbol{x}) = \boldsymbol{x}_\delta$ almost surely, where $\boldsymbol{x}_\delta = \boldsymbol{x} - \boldsymbol{f}^*(\boldsymbol{x})\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})$.*

This corollary shows that adversarial examples are precisely identified for the classifier based on $\mathcal{L}^{hKR}_{\lambda,m}$: the direction given by $\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})$ and distance by $|\boldsymbol{f}^*(\boldsymbol{x})| * ||\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})|| = |\boldsymbol{f}^*(\boldsymbol{x})|$. In this scenario, the optimal adversarial attacks align with the gradient direction (i.e., FGSM attack [22]). This supports the observations made in [51], where all attacks, such as PGD [40] or Carlini and Wagner [12], applied on an OTNN model, were equivalent to FGSM attacks.

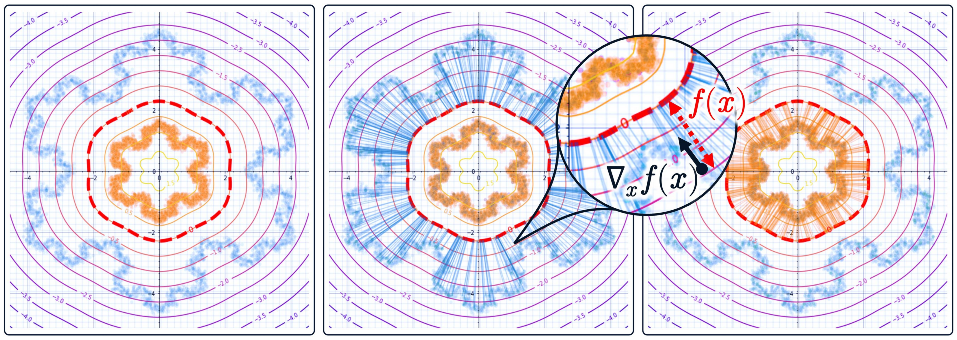
To illustrate these propositions, we learnt a dense binary classifier with $\mathcal{L}^{hKR}_{\lambda,m}$ to separate two complex distributions, following two concentric Koch snowflakes. Fig.2-a shows the two distributions (blue and orange snowflakes), the learnt boundary (0-levelset) (red dashed line). Fig.2-b,c show for random samples $\boldsymbol{x}$ from the two distributions, the segments $[\boldsymbol{x}, \boldsymbol{x}_\delta]$ where $\boldsymbol{x}_\delta$ is defined in Prop. 2 . As expected by Prop. 2, $\boldsymbol{x}_\delta$ points fall exactly on the decision boundary. Besides, as stated in Prop. 1 each segment provides the direction of the image with respect to the transport plan.

Finally, we showed that with OTNN, adversarial attacks are formally known and simple to compute. Furthermore, since we proved that these attacks are along the transportation map, they acquire a meaningful interpretation and are no longer mere adversarial examples exploiting local noise, but rather correspond to the global solution of a transportation problem.

4 Link between OTNN gradient and counterfactual explanations

The vulnerability of traditional networks to adversarial attacks indicates that their decision boundaries are locally flawed, deviating significantly from the Bayesian boundaries between classes. Since the gradient directs towards this anomalous boundary, Saliency Maps [53], given by $\boldsymbol{g}(\boldsymbol{x}) = |\nabla_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})|$ fails to represent a meaningful transition between classes and then often lead to noisy explanation (as stated in Section 2).

On the contrary, in the experiments, we will demonstrate that OTNN gradients induce meaningful explanations (Sec. 5). We justify these good properties by building a link with counterfactual explanations. Indeed, according to [10, 14], optimal transport plans are potential surrogates or even substitutes to causal counterfactuals. Optimal Transport plan provides for any sample $\boldsymbol{x} \in \mu$ a sample $\gamma_\pi(\boldsymbol{x}) \in \nu$, the closest in average on the pairing process. [14] prove that these optimal transport plans can even coincide with causal counterfactuals when available. Relying on this definition of OT counterfactual, Prop. 1 demonstrates that gradients of the optimal OTNN solution provide almost surely the direction to the counterfactual $\boldsymbol{\gamma}_\pi(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}}\boldsymbol{f}^*(\boldsymbol{x})$. Even if t is only partially known, using $t = \boldsymbol{f}^*(\boldsymbol{x})$, we know that $\boldsymbol{x}_\delta$ is on the decision boundary (Corr. 1) and is both an adversarial attack and a counterfactual explanation and $|t| \geq |\boldsymbol{f}^*(\boldsymbol{x})|$ is on the path to the other distribution.
Thus the learning process of OTNNs induces a strong constraint on the gradients of the neural network, aligning them to the optimal transport plan. We claim that is the reason why the simple Saliency Maps for OTNNs have very good properties: We will demonstrate in the Sec 5 that, for the



(a) (b) (c)

Figure 2: Level sets of an OTNN classifier $\boldsymbol{f}$ for two concentric Koch snowflakes (a). The decision boundary (denoted $\partial\mathcal{X}$, also called the 0-level set) is the red dashed line. Figure (b) (resp. (c)) represents the translation of the form $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})\nabla_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})$ of each point $\boldsymbol{x}$ of the first class (resp second class). $[\boldsymbol{x}, \boldsymbol{x}']$ pairs are represented by blue (resp. orange) segments.

Saliency Map explanations: (***i***) metrics scores are higher or comparable to other explanation methods (which is not the case for unconstrained networks), thus it has higher ranks; (***ii***) distance to other attribution methods such as Smoothgrad is imperceptible; (***iii***) scores obtained on metrics that can be compared between networks are higher than those obtained with unconstrained networks; (***iv***) alignment with human explanations is impressive.

5 Experiments

We conduct experiments with networks learnt on FashionMNIST [66], and 22 binary labels of CelebA [39] datasets, Cat vs Dog (binary classification, 224x224x3 uncentered images), and Imagenet [16]. Note that labels in CelebA are very unbalanced (see Table 2 in Appendix A.2, with for instance less than 5% samples for *Mustache* or *Wearing_Hat*).Architectures used for OTNNs and unconstrained networks are similar (same number of layers and neurons, a VGG for FashionMNIST and CelebA, a ResNet50 for Cat vs Dog and Imagenet). We also train an alternative of ResNet50 OTNN with twice the number of parameters (50 M). Unconstrained networks use batchnorm and ReLU layers for activation, whereas OTNNs only use GroupSort2 [5, 51] activation. OTNNs are built using the *DEEL.LIP*[1] library. The loss functions are cross-entropy for unconstrained networks (categorical for multiclass, and sigmoid for multilabel settings), and hKR $\mathcal{L}^{hKR}_{\lambda,m}$ (and the multiclass variants see appendix A.2.3) for OTNNs. We train all networks with Adam optimizer [34]. Details on architectures and parameters are given in Appendix A.2.

Classification performance: OTNN models achieve comparable results to unconstrained ones, confirming claims of [7]: they reach 88.5% average accuracy on FashionMNIST (Table 6), and 81% (resp. 82%) average Sensitivity (resp. Specificity) over labels on CelebA (Table 6 in Appendix A.3). We use Sensitivity and Specificity for CelebA to take into consideration the unbalanced labels. OTNNs achieve 96% accuracy (98% for the unconstrained version) on Cat vs Dog and 67% (75% for the unconstrained version) on Imagenet. The ResNet50 OTNN with 50M parameters achieves 70% accuracy on Imagenet.

We present the results of quantitative evaluations of XAI metrics to compare the Saliency Map method with other explanation methods on OTNN, and more generally compare XAI explanations methods on these networks and their unconstrained counterparts. On CelebA, we only present the results for the label *Mustache*, but results for the other labels are similar. Parameters for explanation methods and metrics are given in Appendix A.4. We have chosen to present in Table 1 two SoTA XAI metrics that enable comparison between OTNNs and unconstrained networks. μ**Fidelity metric** [8] is a well-known method that measures the correlation between important variables defined by the explanation method and the model score decreases when these variables are reset to a baseline state (or replaced by uniform noise). Another important property for explanations is their stability for

[1] `https://github.com/deel-ai/deel-lip` distributed under MIT License (MIT)

	Dataset							
	Fash. MNIST		CelebA		Cat vs Dog		Imagenet	
	OTNN	Uncst.	OTNN	Uncst.	OTNN	Uncst.	OTNN	Uncst.
Attribution	μ**Fidelity-Uniform (↑ is better)**							
Saliency	**0.156**	-0.001	**0.244**	0.052	**0.091**	0.080	**0.240**	0.004
GradCam	**0.215**	0.02	**0.028**	0.002	**0.101**	0.052	0.029	**0.046**
Attribution	μ**Fidelity-Zero (↑ is better)**							
Saliency	**0.246**	0.034	**0.325**	0.082	**0.121**	0.079	**0.147**	0.049
GradCam	**0.161**	0.046	**0.127**	0.061	**0.136**	0.049	0.048	**0.068**
Attribution	**Stability Spearman rank (↓ is better)**							
Saliency	**0.59**	0.91	**0.51**	0.77	**0.58**	0.69	**0.60**	0.74
Distance Saliency smoothgrad (↓ is better)								
Saliency	**7.5e-03**	7.0e-02	**3.1e-4**	1.4e-1	**3.7e-8**	4.1e-8	**3.7e-8**	4.3e-8

Table 1: Comparison of XAI metrics for different attributions methods and dataset for OTNN and unconstrained networks.

nearby samples. In [67], the authors proposed Stability metrics based on the L_2 distance. To better evaluate this stability and make it comparable for different models, we replace the L_2 distance by $1 - \rho$, ρ being the Spearman rank correlation. Other model-dependent metrics are described in the Appendix. Results from the 50M parameter ResNet OTNN are included in the human alignment study (Fig 3) to illustrate that enhancing the model's complexity can bolster both the accuracy and alignment. The following observations can be drawn from Table 1:

A similar observation holds for the Stability Spearman rank : OTNN scores are better whatever the attribution method.

OTNNs explanations are aligned with human ones : Adopting the method presented in [38], using the ClickMe dataset, we compute the human feature alignment of OTNN Saliency Maps and compare with the others models tested in [18]– more than 100 recent deep neural networks. In Figure 3, we demonstrate that OTNN models Saliency Maps, which also carries a theoretical interpretation as the direction of the transport plan, is more aligned with human attention than any other tested models and significantly surpasses the Pareto front discovered by [18]. The OTNN model is even more aligned than a ResNet50 model trained with a specific alignment objective, proposed by [18], and called *Harmonized ResNet50*. This finding is interesting as it indicates OTNNs are less prone to relying on spurious correlations [21] and better capture human visual strategies for object recognition. The implications of these results are crucial for both cognitive science and industrial applications. A model that more closely aligns with human attention and visual strategies can provide a more comprehensive understanding of how vision operates for humans, and also enhance the predictability, interpretability, and performance of object recognition models in industry settings. Furthermore, the drop in alignment observed in recent models highlights the necessity of considering the alignment of model visual strategies with human attention while developing object recognition models to reduce the reliance on spurious correlations and ensure that our models get things right for the right reasons.

Qualitative results: Using the learnt OTNN on CelebA (Mouth Slightly Open label), Cat vs Dog and Imagenet, Fig. 4 presents the original image, average gradients $\nabla_x f_j$ over the channels, and images in the direction of the transport plan (Prop. 1), other samples and datasets are given in Appendix A.5. We can see that most of the gradients are visually consistent, showing clearly what has to be changed in the input image to modify the class and act as a counterfactual explanation. This is less clear for the Imagenet examples. This could be due to the difficulty of defining a transport plan for each pair of the 1000 classes. However, feature visualizations in Figure 1 show that the internal representation of the classes is still more interpretable than the unconstraint one. More generally, we observe that the gradient gives clear information about how the classifier makes its decision. For instance, for the cat, it shows that the classifier does not need to encode perfectly the concept of a cat, but mainly to identify the color of the eyes and size of the nose.

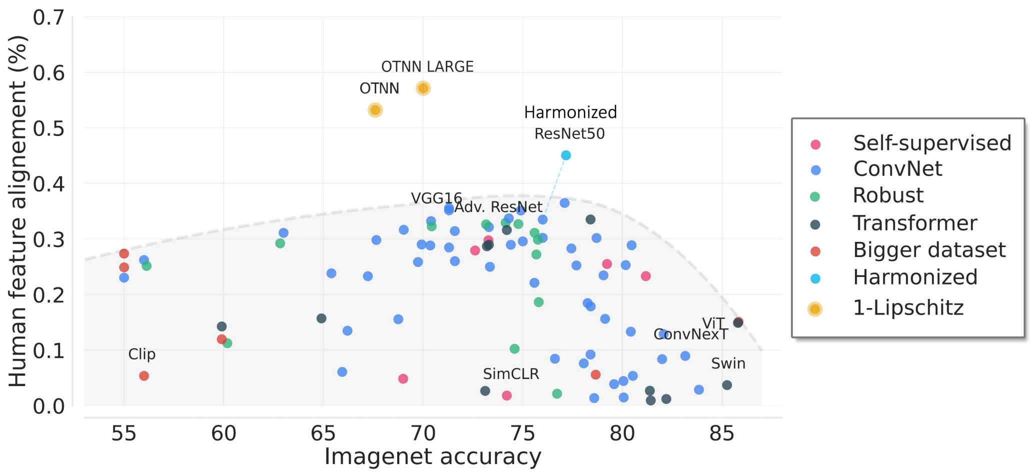
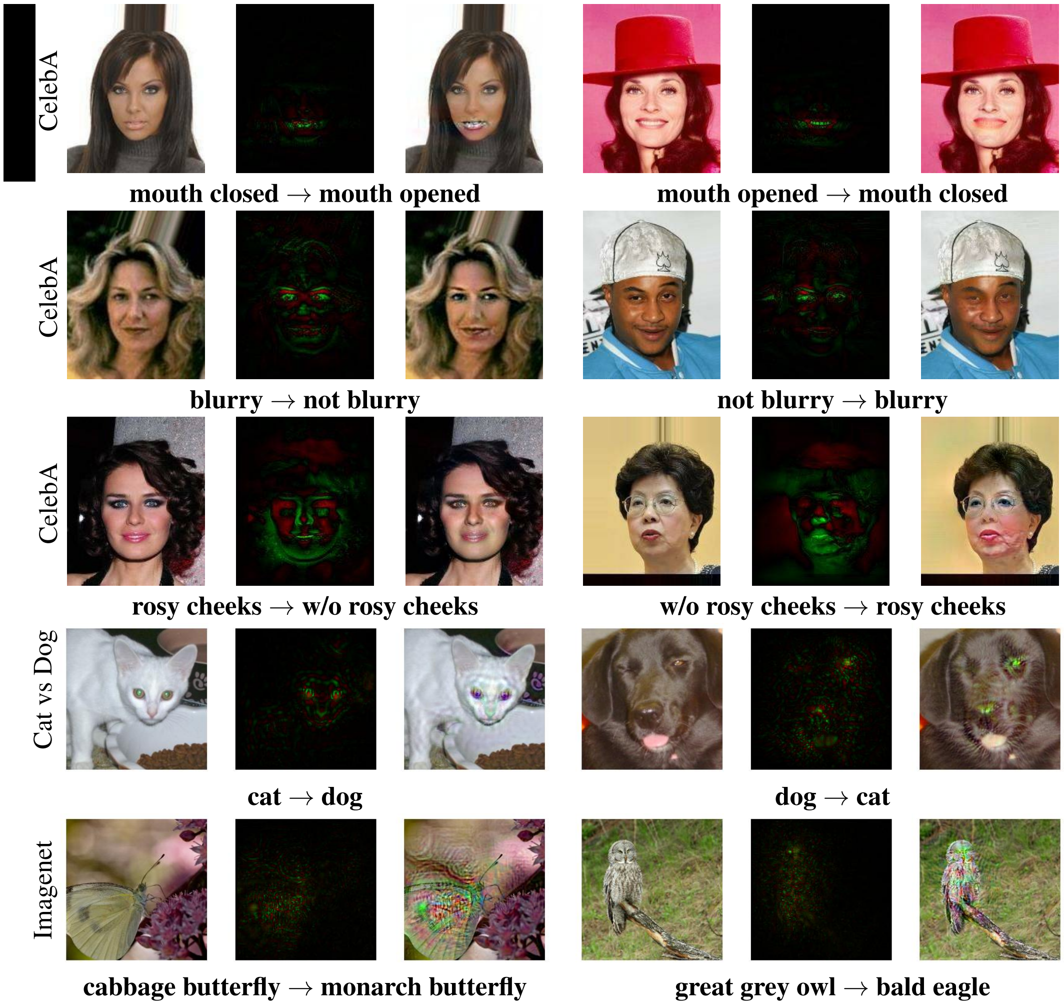


Figure 3: **OTNN are aligned with Human attention.** Our study shows that the Saliency Map of OTNN model is highly aligned with human attention.



mouth closed → mouth opened **mouth opened → mouth closed**

blurry → not blurry **not blurry → blurry**

rosy cheeks → w/o rosy cheeks **w/o rosy cheeks → rosy cheeks**

cat → dog **dog → cat**

cabbage butterfly → monarch butterfly **great grey owl → bald eagle**

Figure 4: Samples of counterfactuals from different datasets: (left) source image , (center) gradient image, (right) counterfactual of the form $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_{\boldsymbol{x}}\hat{f}(\boldsymbol{x})$, for $t > 1$

6 Conclusions

In this paper, we study the explainability properties of OTNN (Optimal Transport Neural Networks) that are 1-Lipschitz constrained neural networks trained with an optimal transport dual loss. We establish that the gradient of the optimal solution aligns with the transportation plan direction, and the closest decision boundary point (adversarial example) also lies in this gradient direction at a distance of the absolute value of the network output. Relying on a formal definition of Optimal Transport counterfactuals, we build a link between the OTNN gradients and counterfactual explanations . We thus show that OTNNs loss jointly targets the classification task and induces a gradient alignment to the transportation plan. The experiments show that the simple Saliency Map has top-rank scores on state-of-the-art XAI metrics, and largely outperforms any method applied to unconstrained networks. And even if counterfactual explanations are less compelling for Imagenet, probably due to the complexity of transport for large number of classes, we prove that OTNNs Saliency Maps are impressively aligned to human explanations.

This paper demonstrates how leveraging an Optimal Transport (OT) objective for classification tasks provides additional interesting properties for critical problems. Besides the well known certifiable robustness of 1-Lipschitz Neural Networks, Optimal Transport Neural Networks (OTNNs) offer inherent explainability by design through the utilization of the simple Saliency Map method. Furthermore, our study reveals that OTNNs can achieve accuracy comparable to those of unconstrained networks. While the training duration for OTNNs is three to six times longer than that for unconstrained networks, they share comparable computational costs for inference. We hope that this contribution will further add to the growing interest in the application of Optimal Transport for Neural Networks.

Acknowledgments and Disclosure of Funding

This work has benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French "Investing for the Future – PIA3" program under the Grant agreement ANR-19-P3IA-0004. The authors gratefully acknowledge the support of the DEEL project.[2]

References

[1] E. M. Achour, F. Malgouyres, and F. Mamalet. Existence, stability and scalability of orthogonal convolutional neural networks, 2021.

[2] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc.

[3] L. Ambrosio and A. Pratelli. *Existence and stability results in the L1 theory of optimal transportation*, pages 123–160. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[4] M. Ancona, E. Ceolini, A. C. Öztireli, and M. H. Gross. A unified view of gradient-based attribution methods for deep neural networks. *CoRR*, 2017.

[5] C. Anil, J. Lucas, and R. Grosse. Sorting out Lipschitz function approximation. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 291–301, Long Beach, California, USA, June 2019. PMLR.

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, Aug. 2017. PMLR.

[7] L. Béthune, A. González-Sanz, F. Mamalet, and M. Serrurier. The many faces of 1-lipschitz neural networks. *CoRR*, abs/2104.05097, 2021.

[2] `https://www.deel.ai/`

[8] U. Bhatt, A. Weller, and J. M. F. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2020.

[9] Å. Björck and C. Bowie. An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, June 1971.

[10] E. Black, S. Yeom, and M. Fredrikson. Fliptest: Fairness auditing via optimal transport. *CoRR*, abs/1906.09218, 2019.

[11] A. Boopathy, S. Liu, G. Zhang, C. Liu, P.-Y. Chen, S. Chang, and L. Daniel. Proper network interpretability helps adversarial robustness in classification. In *International Conference on Machine Learning*, 2020.

[12] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[13] P. Chalasani, J. Chen, A. R. Chowdhury, S. Jha, and X. Wu. Concise explanations of neural networks using adversarial training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[14] L. de Lara, A. González-Sanz, N. Asher, and J.-M. Loubes. Transport-based counterfactual models. *arxiv:2108.13025*, 2021.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] T. Fel, M. Ducoffe, D. Vigouroux, R. Cadene, M. Capelle, C. Nicodeme, and T. Serre. Don't lie to me! robust and efficient explainability with verified perturbation analysis. *arXiv preprint arXiv:2202.07728*, 2022.

[18] T. Fel, I. Felipe, D. Linsley, and T. Serre. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[19] T. Fel, D. Vigouroux, R. Cadène, and T. Serre. How Good is your Explanation? Algorithmic Stability Measures to Assess the Quality of Explanations for Deep Neural Networks. In *2022 CVF Winter Conference on Applications of Computer Vision (WACV)*, Hawaii, United States, Jan. 2022.

[20] Fel, Thomas and Hervier, Lucas. Xplique: an neural networks explainability toolbox. 2021.

[21] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572 [stat.ML]*, Dec. 2014.

[23] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[25] P. Hase, H. Xie, and M. Bansal. The out-of-distribution problem in explainability and search methods for feature importance explanations. *NeurIPS*, 2021.

[26] J. Haug, S. Zürn, P. El-Jiz, and G. Kasneci. On baselines for local feature attributions. *arXiv preprint arXiv:2101.00905*, 2021.

[27] M. Hein and M. Andriushchenko. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. *arXiv:1705.08475 [cs, stat]*, May 2017.

[28] J. Heo, S. Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[29] C. Hsieh, C. Yeh, X. Liu, P. K. Ravikumar, S. Kim, S. Kumar, and C. Hsieh. Evaluations and methods for explanation through robustness analysis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[30] C.-Y. Hsieh, C.-K. Yeh, X. Liu, P. Ravikumar, S. Kim, S. Kumar, and C.-J. Hsieh. Evaluations and methods for explanation through robustness analysis. In *International Conference on Learning Representations*, 2021.

[31] P. Jacob, É. Zablocki, H. Ben-Younes, M. Chen, P. Pérez, and M. Cord. STEEX: steering counterfactual explanations with semantics. *CoRR*, abs/2111.09094, 2021.

[32] A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *ACM Conference on Fairness, Accountability, and Transparency*, pages 353–362, 2021.

[33] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un) reliability of saliency methods. 2019.

[34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] D. Lewis. Causation. *Journal of Philosophy*, 70(17):556–567, 1973.

[36] Q. Li, S. Haque, C. Anil, J. Lucas, R. B. Grosse, and J. Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. *arXiv:1911.00937*, Apr. 2019.

[37] Z. Q. Lin, M. J. Shafiee, S. Bochkarev, M. S. Jules, X. Y. Wang, and A. Wong. Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. In *NIPS*, 2019.

[38] D. Linsley, S. Eberhardt, T. Sharma, P. Gupta, and T. Serre. What are the visual features underlying human versus machine vision? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2706–2714, 2017.

[39] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[40] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, 2017.

[41] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018.

[42] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Visualization for Deep Learning workshop, International Conference in Machine Learning*, 2016. arXiv preprint arXiv:1602.03616.

[43] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[44] H. Ono, T. Takahashi, and K. Kakizaki. Lightweight Lipschitz Margin Training for Certified Defense against Adversarial Examples. *arXiv:1811.08080 [cs, stat]*, Nov. 2018.

[45] V. Petsiuk, A. Das, and K. Saenko. RISE: randomized input sampling for explanation of black-box models. *CoRR*, abs/1806.07421, 2018.

[46] G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–206, 2018.

[47] P. Rodriguez, M. Caccia, A. Lacoste, L. Zamparo, I. Laradji, L. Charlin, and D. Vazquez. Beyond trivial counterfactual explanations with diverse valuable explanations. In *Int. Conf in Computer Vision (ICCV)*, 2021.

[48] A. Ross, H. Lakkaraju, and O. Bastani. Learning models for actionable recourse. *NeurIPS*, 2021.

[49] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *CoRR*, abs/1602.07868, 2016.

[50] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.

[51] M. Serrurier, F. Mamalet, A. González-Sanz, T. Boissin, J.-M. Loubes, and E. Del Barrio. Achieving robustness in classification using optimal transport with hinge regularization. In *Conference on Computer Vision and Pattern Recognition (CVPR'21)*, 2021.

[52] H. Shah, P. Jain, and P. Netrapalli. Do input gradients highlight discriminative features? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2046–2059, 2021.

[53] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014.

[54] L. Sixt, M. Granz, and T. Landgraf. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, 2020.

[55] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

[56] J. Sokolic, R. Giryes, G. Sapiro, and M. R. D. Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, Aug. 2017.

[57] P. Sturmfels, S. Lundberg, and S.-I. Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020.

[58] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks, 2017.

[59] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece. Sanity checks for saliency metrics. In *Association for the Advancement of Artificial Intelligence*, 2019.

[60] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy, 2018.

[61] S. Verma, J. P. Dickerson, and K. Hines. Counterfactual explanations for machine learning: A review. *CoRR*, abs/2010.10596, 2020.

[62] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[63] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[64] P. Wang and N. Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[65] Z. Wang, M. Fredrikson, and A. Datta. Robust models are more interpretable because attributions look normal, 2022.

[66] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.

[67] C.-K. Yeh, C.-Y. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, 2019.

A Appendix

A.1 Additional definition and proofs

Let us first recall the optimal transport problem associated with the minimization of $\mathcal{L}^{hKR}_{\lambda,m}$:

$$\inf_{f \in Lip_1(\Omega)} \mathcal{L}^{hKR}_{\lambda(f),m} = \inf_{\pi \in \Pi^p_\lambda(\mu,\nu)} \int_{\Omega\times\Omega} |\boldsymbol{x} - \boldsymbol{z}| d\pi + \pi_{\boldsymbol{x}}(\Omega) + \pi_{\boldsymbol{z}}(\Omega) - 1 \tag{3}$$

Where $\Pi^p_\lambda(\mu, \nu)$ is the set consisting of positive measures $\pi \in \mathcal{M}_+(\Omega \times \Omega)$ which are absolutely continuous with respect to the joint measure $d\mu \times d\nu$ and $\frac{d\pi_{\mathbf{x}}}{d\mu} \in [p, p(m + \lambda)]$, $\frac{d\pi_{\mathbf{z}}}{d\nu} \in [1 - p, (1 - p)(m + \lambda)]$. We name π^* the optimal transport plan according to Eq.3 and and f^* the associated potential function.

Proof of proposition 1: According to [51], we have

$$||\nabla_x f^*(\boldsymbol{x})|| = 1$$

almost surely and

$$\mathbb{P}_{(\boldsymbol{x},y)\sim\pi^*} \left(|f^*(\boldsymbol{x}) - f^*(y)| = ||\boldsymbol{x} - y||\right) = 1$$

Following the proof of proposition 1 in [24] and [3] we have :
Given $\boldsymbol{x}_\alpha = \alpha * x + (1 - \alpha)y, 0 \leq \alpha \leq 1$

$$\mathbb{P}_{(\boldsymbol{x},y)\sim\pi^*} \left(\nabla_x f^*(\boldsymbol{x}_\alpha) = \frac{\boldsymbol{x}_\alpha - y}{||\boldsymbol{x}_\alpha - y||}\right) = 1.$$

So for for $\alpha = 1$ whe have

$$\mathbb{P}_{(\boldsymbol{x},y)\sim\pi^*} \left(\nabla_x f^*(\boldsymbol{x}) = \frac{\boldsymbol{x} - y}{||\boldsymbol{x} - y||}\right) = 1$$

and then

$$\mathbb{P}_{(\boldsymbol{x},y)\sim\pi^*} \left(y = \boldsymbol{x} - \nabla_x f^*(\boldsymbol{x}).||\boldsymbol{x} - y||\right) = 1$$

This prove the proposition 1 by choosing $t = ||\boldsymbol{x} - y||$.■

Proof of proposition 2: Let μ and ν two distributions with disjoint support with minimal distance ϵ and f^* an optimal solution minimizing the $\mathcal{L}^{hKR}_{\lambda,m}$ with $m < 2\epsilon$. According to [51], f^* is 100% accurate. Since the classification is based on the sign of f we have : $\forall \boldsymbol{x} \in \mu, f^*(\boldsymbol{x}) \geq 0$ and $\forall y \in \nu, f^*(y) \leq 0$. Given $\boldsymbol{x} \in \mu$ and $y = tr_\pi(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_x f^*(\boldsymbol{x})$ and $y \in \nu$. According to the previous proposition we have :

$$\begin{aligned}
|f^*(\boldsymbol{x}) - f^*(y)| &= ||\boldsymbol{x} - y|| \\
|f^*(\boldsymbol{x}) - f^*(y)| &= ||\boldsymbol{x} - (\boldsymbol{x} - t\nabla_x f^*(\boldsymbol{x}))|| \\
|f^*(\boldsymbol{x}) - f^*(y)| &= ||t\nabla_x f^*(\boldsymbol{x}))|| \\
|f^*(\boldsymbol{x}) - f^*(y)| &= t.||\nabla_x f^*(\boldsymbol{x}))|| && (t \geq 0) \\
|f^*(\boldsymbol{x}) - f^*(y)| &= t && (\nabla_x f^*(\boldsymbol{x}) = 1) \\
f^*(\boldsymbol{x}) - f^*(y) &= t && (f^*(\boldsymbol{x}) \geq 0, f^*(y) \leq 0) \\
f^*(y) &= f^*(\boldsymbol{x}) - t
\end{aligned}$$

since $f^*(y) \leq 0$ we obtain :

$$f^*(\boldsymbol{x}) \leq t$$

Since f^* is continuous, $\exists t' > 0$ such that $\boldsymbol{x}_\delta = \boldsymbol{x} - t'\nabla_x f^*(\boldsymbol{x})$ and $f^*(\boldsymbol{x}_\delta) = 0$. We have :

$$\begin{aligned}
|f^*(\boldsymbol{x}) - f^*(\boldsymbol{x}_\delta|) &\leq ||\boldsymbol{x} - \boldsymbol{x}_\delta|| \\
f^*(\boldsymbol{x}) &\leq ||\boldsymbol{x} - (\boldsymbol{x} - t'\nabla_x f^*(\boldsymbol{x}))|| \\
f^*(\boldsymbol{x}) &\leq t'
\end{aligned}$$

and

$$\begin{aligned} |f^*(\boldsymbol{x}_\delta) - f^*(y)| &\leq ||\boldsymbol{x}_\delta - y|| \\ -f^*(y) &\leq ||(\boldsymbol{x} - t'\nabla_x f^*(\boldsymbol{x})) - (\boldsymbol{x} - t\nabla_x f^*(\boldsymbol{x}))|| \\ -f^*(y) &\leq t - t' \\ -f^*(y) &\leq ||\boldsymbol{x} - y|| - t' \end{aligned} \qquad)$$

Then, if $f^*(\boldsymbol{x}) < t'$ we have

$$\begin{aligned} f^*(\boldsymbol{x}) - f^*(y) <& t' + ||\boldsymbol{x} - y|| - t' \\ f^*(\boldsymbol{x}) - f^*(y) <& ||\boldsymbol{x} - y|| \end{aligned}$$

which is a contradiction so $f^*(\boldsymbol{x}) = t'$ and

$$\boldsymbol{x}_\delta = \boldsymbol{x} - f^*(\boldsymbol{x})\nabla_x f^*(\boldsymbol{x})$$

■

A.2 Parameters and architectures

A.2.1 Datasets

FashionMNIST has 50,000 images for training and 10,000 for test of size $28 \times 28 \times 1$, with 10 classes.

CelebA contains 162,770 training samples, 19,962 samples for test of size $218 \times 178 \times 3$. We have used a subset of 22 labels: *Attractive*, *Bald*, *Big_Nose*, *Black_Hair*, *Blond_Hair*, *Blurry*, *Brown_Hair*, *Eyeglasses*, *Gray_Hair*, *Heavy_Makeup*, *Male*, *Mouth_Slightly_Open*, *Mustache*, *Receding_Hairline*, *Rosy_Cheeks*, *Sideburns*, *Smiling*, *Wearing_Earrings*, *Wearing_Hat*, *Wearing_Lipstick*, *Wearing_Necktie*, *Young*.

Note that labels in CelebA are very unbalanced (see Table 2, with less than 5% samples for *Mustache* or *Wearing_Hat* for instance). Thus we will use Sensibility and Specificity as metrics.

Table 2: CelebA label distribution: proportion of positive samples in training set (testing set) [bold: very unbalanced labels]

Attractive	*Bald*	*Big_Nose*	*Black_Hair*	*Blond_Hair*
0.51 (0.50)	**0.02 (0.02)**	**0.24 (0.21)**	**0.24 (0.27)**	**0.15 (0.13)**
Blurry	*Brown_Hair*	*Eyeglasses*	*Gray_Hair*	*Heavy_Makeup*
0.05 (0.05)	**0.20 (0.18)**	**0.06 (0.06)**	**0.04 (0.03)**	0.38 (0.40)
Male	*Mouth_Slightly_Open*	*Mustache*	*Receding_Hairline*	*Rosy_Cheeks*
0.42 (0.39)	0.48 (0.50)	**0.04 (0.04)**	**0.08 (0.08)**	**0.06 (0.07)**
Sideburns	*Smiling*	*Wearing_Earrings*	*Wearing_Hat*	*Wearing_Lipstick*
0.06 (0.05)	0.48 (0.50)	**0.19 (0.21)**	**0.05 (0.04)**	0.47 (0.52)
Wearing_Necktie	*Young*			
0.12 (0.14)	0.78 (0.76)			

Cat vs Dog contains 17400 training samples, 5800 test samples of various size.

Imagenet contains 1M training samples, 100 000 samples for test of various size.

preprocessing: For FashionMNIST Images are normalized between $[0, 1]$ with no augmentation. For CelebA dataset, data augmentation is used with random crop, horizontal flip, random brightness, and random contrast. For imagenet and cat vs dog we use the standart preprocessing of resnet (with no normalization in $[0, 1]$)

A.2.2 Architectures

As indicated in the paper, linear layers for OTNN and unconstrained networks are equivalent (same number of layers and neurons), but unconstrained networks use batchnorm and ReLU layer for activation, whereas OTNN only use GroupSort2 [5, 51] activation. OTNN are built using *DEEL.LIP*[3] library.

1-Lipschitz networks parametrization. Several soltutions have been proposed to set the Lipschitz constant of affine layers: Weight clipping [6] (WGAN), Frobenius normalization [49] and spectral normalization [41]. In order to avoid vanishing gradients, orthogonalization can be done using Björck algorithm [9]. DEEL.LIP implements most of these solutions, but we focus on layers called *SpectralDense* and *SpectralConv2D*, with spectral normalization [41] and Björck algorithm [9]. Most activation functions are Lipschitz, including ReLU, sigmoid, but we use GroupSort2 proposed by [5], and defined by the following equation:

$$\text{GroupSort2}(x)_{2i,2i+1} = [\min(x_{2i}, x_{2i+1}), \max(x_{2i}, x_{2i+1})]$$

Network architectures used for CelebA dataset are described in Table 3.

Network architectures used for FashionMNIST dataset are described in Table 4. The same OTNN architecture is used for MNIST expermentation presented in Fig. 1.

[3] `https://github.com/deel-ai/deel-lip` distributed under MIT License (MIT)

Table 3: CelebA Neural network architectures: Sconv2D is SpectralConv2D, GS2 is GroupSort2, L2Pool is L2NormPooling, SDense is SpectralDense, BN is BatchNorm, AvgPool is AveragePooling

Dataset	OTNN	Unconstrained NN	
	Layer	Layer	Output size
CelebA	Input	Input	$218 \times 178 \times 3$
	SConv2D, GS2	Conv2D, BN, ReLU	$218 \times 178 \times 16$
	SConv2D, GS2	Conv2D, BN, ReLU	$218 \times 178 \times 16$
	L2Pool	AvgPool	$109 \times 89 \times 16$
	SConv2D, GS2	Conv2D, BN, ReLU	$109 \times 89 \times 32$
	SConv2D, GS2	Conv2D, BN, ReLU	$109 \times 89 \times 32$
	L2Pool	AvgPool	$54 \times 44 \times 32$
	SConv2D, GS2	Conv2D, BN, ReLU	$54 \times 44 \times 64$
	SConv2D, GS2	Conv2D, BN, ReLU	$54 \times 44 \times 64$
	SConv2D, GS2	Conv2D, BN, ReLU	$54 \times 44 \times 64$
	L2Pool	AvgPool	$27 \times 22 \times 64$
	SConv2D, GS2	Conv2D, BN, ReLU	$27 \times 22 \times 128$
	SConv2D, GS2	Conv2D, BN, ReLU	$27 \times 22 \times 128$
	SConv2D, GS2	Conv2D, BN, ReLU	$27 \times 22 \times 128$
	L2Pool	AvgPool	$13 \times 11 \times 128$
	SConv2D, GS2	Conv2D, BN, ReLU	$13 \times 11 \times 128$
	SConv2D, GS2	Conv2D, BN, ReLU	$13 \times 11 \times 128$
	SConv2D, GS2	Conv2D, BN, ReLU	$13 \times 11 \times 128$
	L2Pool	AvgPool	$6 \times 5 \times 128$
	Flatten, SDense, GS2	Flatten, Dense, BN, ReLU	256
	SDense, GS2	Dense,BN, ReLU	256
	SDense	Dense	22

The 1-Lipschitz version of resnet50 is described in Table 5. As the unconstrained version, It has around 25M parameters. For the large version, we simply multiply the number channels in hidden layers by 1.5. The unconstrained version is the standart resnet50 architecture. In the case of imagenet we use the pretrained version provided by tensorflow.

Table 4: FashionMNIST Neural network architectures: Sconv2D is SpectralConv2D, GS2 is GroupSort2, SDense is SpectralDense, BN is BatchNorm, AvgPool is AveragePooling, SGAvgPool is ScaledGlobalAveragePooling (DEEL.LIP), GAvgPool is GlobalAveragePooling

Dataset	OTNN	Unconstrained NN	
	Layer	Layer	Output size
FashionMNIST	Input	Input	$28 \times 28 \times 1$
	SConv2D, GS2	Conv2D, BN, ReLU	$28 \times 28 \times 96$
	SConv2D, GS2	Conv2D, BN, ReLU	$28 \times 28 \times 96$
	SConv2D, GS2	Conv2D, BN, ReLU	$28 \times 28 \times 96$
	SConv2D (stride=2), GS2	Conv2D (stride=2), BN, ReLU	$14 \times 14 \times 96$
	SConv2D, GS2	Conv2D, BN, ReLU	$14 \times 14 \times 192$
	SConv2D, GS2	Conv2D, BN, ReLU	$14 \times 14 \times 192$
	SConv2D, GS2	Conv2D, BN, ReLU	$14 \times 14 \times 192$
	SConv2D (stride=2), GS2	Conv2D (stride=2), BN, ReLU	$7 \times 7 \times 192$
	SConv2D, GS2	Conv2D, BN, ReLU	$7 \times 7 \times 384$
	SConv2D, GS2	Conv2D, BN, ReLU	$7 \times 7 \times 384$
	SConv2D, GS2	Conv2D, BN, ReLU	$7 \times 7 \times 384$
	SGAvgPool	GAvgPool	384
	SDense	Dense	10

Table 5: 1-lip resnet architecture for Imagenet and cat vs dog: Sconv2D is SpectralConv2D, GS2 is GroupSort2, SDense is SpectralDense, BC is Batchcentering (centeing without normalization), SL2npool is ScaledL2NormPooling2D, SGAvgl2Pool is ScaledGlobalL2NormPooling2D, GAvgPool is GlobalAveragePooling

Layer				output
Input				$224 \times 224 \times 3$
SConv2D 7-64 (stride=2), BC, GS2				$112 \times 112 \times 64$
InvertibleDownSampling				$56 \times 56 \times 256$
SConv2D 1×1	64	BC, GS2	$\times 3$	$56 \times 56 \times 256$
SConv2D 3×3	64	BC, GS2		
SConv2D 1×1	256	BC		
add-lip		BC, GS2		
SL2npool				$28 \times 28 \times 256$
SConv2D 1×1	128	BC, GS2	$\times 4$	$28 \times 28 \times 512$
SConv2D 3×3	128	BC, GS2		
SConv2D 1×1	512	BC		
add-lip		BC, GS2		
SL2npool				$14 \times 14 \times 512$
SConv2D 1×1	256	BC, GS2	$\times 6$	$14 \times 14 \times 1024$
SConv2D 3×3	256	BC, GS2		
SConv2D 1×1	1024	BC		
add-lip		BC, GS2		
SL2npool				$7 \times 7 \times 1024$
SConv2D 1×1	256	BC, GS2	$\times 3$	$7 \times 7 \times 2048$
SConv2D 3×3	256	BC, GS2		
SConv2D 1×1	1024	BC		
add-lip		BC, GS2		
SGAvgl2Pool				2048
SDense				1 cat vs dog 1000 imagenet

A.2.3 Losses and optimizer

An extension of $\mathcal{L}^{hKR}$ to the multiclass case with q classes. has also been proposed in [51] The idea is to learn q 1-Lipschitz functions $f_1, \dots, f_q$, each component f_i being a *one-versus-all* binary classifier. The loss proposed was the following

$$\mathcal{L}^{hKR}_{\lambda,m}(f_{1,\dots,q}) = \sum_{k=1}^{q} \left[\underset{\mathbf{x} \sim \neg P_k}{\mathbb{E}} [f_k(\mathbf{x})] - \underset{\mathbf{x} \sim P_k}{\mathbb{E}} [f_k(\mathbf{x})] \right] + \lambda \underset{\mathbf{x},y \sim \bigcup_{k=1}^{q} P_k}{\mathbb{E}} (H(f_1(\boldsymbol{x}), \dots, f_q(\boldsymbol{x}), y, m) \tag{4}$$

with :

$$H(f_1(\boldsymbol{x}), \dots, f_q(\boldsymbol{x}), y, m) = (m - f_y(\boldsymbol{x}))_+ + \sum_{k \neq y} (m + f_k(\boldsymbol{x}))_+$$

This formulation has three main drawbacks: (i) for large number of classes several outputs may have few or no positive sample within a batch leading to slow convergence, (ii) weight of $f_y(\boldsymbol{x})$ (the function of the true class) with respect to the other decreases when the number of classes increases, (iii) the expectancy has to be evaluated through the batch, making the loss dependant of the size of the batch.

To overcome these drawbacks, we propose first to replace the Hinge term H with a softmax weighted version. The softmax on all but true class is defined by:

$$\sigma(f_k(\boldsymbol{x}), y, \alpha) = \frac{e^{\alpha * f_k(\boldsymbol{x})}}{\sum_{j \neq y} e^{\alpha * f_j(\boldsymbol{x})}}$$

We can define a weighted version of H function:

$$H_\sigma\left(f_1(\boldsymbol{x}),\ldots,f_q(\boldsymbol{x}),y,m,\alpha\right) = (m - f_y(\boldsymbol{x}))_+ + \sum_{k\neq y} \sigma(f_k(\boldsymbol{x}),y,\alpha) * (m + f_k(\boldsymbol{x}))_+$$

In this function, the value of $f_y(\boldsymbol{x})$ for the true class maintains consistent weight relative to the values of other functions, regardless of the number of classes. α is a temperature parameter. Initially, the softmax behaves like an average as all the values of f_k are close. However, during the learning process, as the values of $|f_k|$ increase, the softmax transitions to function like a maximum. Similarly, if a low value is chosen for α , the softmax behaves as an average, resulting in a one vs all hKR loss. By choosing a higher value for α, the softmax unbalances the weights. Thus the loss persists as a one vs all hKR but incorporates a re-weighting of the opposing classes for each targeted class.

We also propose a sample-wise and weighted version of the KR part (left term in Eq 4). to get the proposed loss:

$$\begin{aligned}\mathcal{L}^{hKR}_{\lambda,m,\alpha}(f_{1,\ldots,q},x,y) = &\left[\sum_{k\neq y}\left[f_k(\boldsymbol{x}) * \sigma(f_k(\boldsymbol{x}),y,\alpha)\right] - f_y(\mathbf{x})\right] \\ &+ \lambda * H_\sigma\left(f_1(\boldsymbol{x}),\ldots,f_q(\boldsymbol{x}),y,m,\alpha\right)\end{aligned} \tag{5}$$

It's important to note that this definition only applys to the balanced multiclass case (as in FashionMnist and ImageNet). In the unbalanced scenario, the weight must be rescaled according to the a priori distribution of the classes.

For CelebA, with hyperparameters λ is set to 20, and $m = 1$. For FashionMNIST, we use Eq. 5, λ is set to 5, $\alpha = 10$ and $m = 0.5$. For cat vs dog λ is set to 10 and $m = 18$. For imagenet λ is set to 500, $\alpha = 200$ and $m = 0.05$.

We train all networks with ADAM optimizer [34]. We use a batch size of 128, 200 epochs , and a fixed learning rate $1\text{e}{-2}$ for CelebA. For FashionMNIST we perform 200 epochs with a batch size of 128. We fix the learing rate to $5\text{e}{-4}$ for the 50 first epochs, $5\text{e}{-5}$ for the epochs 50-75, $1\text{e}{-6}$ for the last epochs. For cat vs dog we perform 200 epochs with a batch size of 256. We fix the learing rate to $1\text{e}{-2}$ for the 100 first epochs, $1\text{e}{-3}$ for the epochs 100-150, $1\text{e}{-4}$ for the epochs 150-180 and $1\text{e}{-9}$ for the last epochs. For imagenet we perform 40 epochs with a batch size of 512. We fix the learing rate to $5\text{e}{-4}$ for the 30 first epochs, $5\text{e}{-5}$ for the epochs 30-35, $1\text{e}{-5}$ for the epochs 35-38 and $1\text{e}{-9}$ for the last epochs.

A.3 Complementary results

A.3.1 FashionMNIST performances and ablation study

Table 6 presents different performance resuts on FashionMNIST. First line is the reference unconstrained network. Second line shows the performances of the new version of $\mathcal{L}^{hKR}_{\lambda,\alpha}$. Table 6 also shows that the new version of the $\mathcal{L}^{hKR}_{\lambda,m,\alpha}$ in the multiclass case (Eq. 5) outperforms the $\mathcal{L}^{hKR}_{\lambda,m}$defined in [51] (Eq. 4). Obviously, the accuracy enhancement is obtained at the expense of the robustness. The main interest of this new loss is to provide a wider range in the accuracy/robustness trade-off.

Table 6: FashionMNIST accuracy comparison with the different version of multiclass $\mathcal{L}^{hKR}_{\lambda,m}$. For the fixed margin, we use the one that performs best by parameter tuning (i.e. $m = 0.5$)

Model	Accuracy
Unconstrained	88.5
OTNN $\mathcal{L}^{hKR}_{\lambda,m}$multiclass version [51] ($\lambda = 10$, $m = 0.5$)	72.2
(Ours) OTNN $\mathcal{L}^{hKR}_{\lambda,m,\alpha}$ ($\lambda = 10$, $m = 0.5$, $\alpha = 10$) (Eq. 5)	**88.6**

A.3.2 CelebA performances

Table 7 presents the Sensibility and Specificity for each label reached by Unconstrained network and OTNN.

As a reminder, given True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) samples, Sensitivity (true positive rate or Recall) is defined by:

$$Sens = \frac{TP}{TP + FN}$$

Specificity (true negative rate) is defined by:

$$Spec = \frac{TN}{TN + FP}$$

Table 7: CelebA performance results for unconstrained and OTNN networks

Model	Metrics: Sensibility/Specificity			
	Attractive	*Bald*	*Big_Nose*	*Black_Hair*
Unconstrained	**0.83 / 0.81**	0.64 / 1.00	**0.65 / 0.87**	**0.74 / 0.95**
OTNN	0.80 / 0.75	**0.87 / 0.83**	0.73 / 0.70	0.78 / 0.84
	Blond_Hair	*Blurry*	*Brown_Hair*	*Eyeglasses*
Unconstrained	**0.86 / 0.97**	**0.49 / 0.99**	**0.80 / 0.88**	**0.96 / 1.00**
OTNN	0.86 / 0.89	0.66 / 0.72	0.81 / 0.73	0.80 / 0.89
	Gray_Hair	*Heavy_Makeup*	*Male*	*Mouth_Slightly_Open*
Unconstrained	0.62 / 0.99	**0.84 / 0.95**	**0.98 / 0.98**	**0.93 / 0.94**
OTNN	**0.84 / 0.83**	0.89 / 0.83	0.92 / 0.89	0.80 / 0.89
	Mustache	*Receding_Hairline*	*Rosy_Cheeks*	*Sideburns*
Unconstrained	0.47 / 0.99	0.47 / 0.98	0.46 / 0.99	**0.79 / 0.98**
OTNN	**0.86 / 0.76**	**0.81 / 0.79**	**0.82 / 0.80**	0.79 / 0.82
	Smiling	*Wearing_Earrings*	*Wearing_Hat*	*Wearing_Lipstick*
Unconstrained	**0.90 / 0.95**	**0.84 / 0.90**	**0.89 / 0.99**	**0.90 / 0.96**
OTNN	0.84 / 0.88	0.78 / 0.72	0.86 / 0.90	0.90 / 0.89
	Wearing_Necktie	*Young*		
Unconstrained	0.75 / 0.98	**0.95 / 0.65**		
OTNN	**0.87 / 0.86**	0.79 / 0.69		

A.4 Complementary explanations metrics

A.4.1 Explanation attribution methods

An attribution method provides an importance score for each input variables x_i in the output $f(x)$. The library used to generate the attribution maps is Xplique [20].

For a full description of attribution methods, we advise to read [17], Appendix B. We will only remind here the equations of

- Saliency: $g(\boldsymbol{x}) = |\nabla_{\boldsymbol{x}} f(\boldsymbol{x})|$
- SmoothGrad: $g(\boldsymbol{x}) = \underset{\delta \sim \mathcal{N}(0, \mathbf{I}\sigma)}{\mathbb{E}} (\nabla f(\boldsymbol{x} + \delta))$

SmoothGrad is evaluated on $N = 50$ samples on a normal distribution of standard deviation $\sigma = 0.2$ around x. Integrated Gradient [58], noted IG, is also evaluated on $N = 50$ samples at regular intervals. Grad-CAM [50], noted GC, is classically applied on the last convolutional layer.

A.4.2 XAI metrics

For the experiments we use four fidelity metrics, evaluated on 1000 samples of test datasets:

- Deletion [45]: it consists in measuring the drop of the score when the important variables are set to a baseline state. Formally, at step k, with u the k most important variables according to an attribution method, the Deletion$^{(k)}$ score is given by:

  $$\text{Deletion}^{(k)} = f(\boldsymbol{x}_{[\boldsymbol{x}_u = \boldsymbol{x}_0]})$$

  The AUC of the Deletion scores is then measured to compare the attribution methods (↓ is better). The baseline x_0 can either be a zero value (*Deletion-zero*), or a uniform random value (*Deletion-uniform*).

- Insertion [45]: this metric is the inverse of Deletion, starting with an image in a baseline state and then progressively adding the most important variables. Formally, at step k, with u the most important variables according to an attribution method, the Insertion$^{(k)}$ score is given by:

  $$\text{Insertion}^{(k)} = f(\boldsymbol{x}_{[\boldsymbol{x}_{\overline{u}} = \boldsymbol{x}_0]})$$

  The AUC is also measured to compare attribution methods (↑ is better). The baseline is the same as for Deletion.

- μFidelity [8]: this metric measures the correlation between the fall of the score when variables are put at a baseline state and the importance of these variables. Formally:

  $$\mu\text{Fidelity} = \underset{\substack{u \subseteq \{1, \dots, d\} \\ |u| = k}}{\text{Corr}} \left(\sum_{i \in u} g(\boldsymbol{x})_i, f(\boldsymbol{x}) - f(\boldsymbol{x}_{[\boldsymbol{x}_u = \boldsymbol{x}_0]}) \right)$$

  For all experiments, k is equal to 20% of the total number of variables, and cutting the image in a grid of 20×20 (9×9 for cat vs dog and imagenet). The baseline is the same as the one used by Deletion. Being a correlation score, we can either compare attribution methods, or different neural networks on the same attribution method (↑ is better).

- Robustness-Sr [29]: this metric evaluate the average adversarial distance when the attack is done only on the most relevant features. Formally, given the u most important variables:

  $$\text{Robustness-Sr} = \left\{ \underset{\delta}{min} ||\delta|| s.t. argmax(f(\boldsymbol{x} + \delta)) \neq argmax(f(\boldsymbol{x})), \delta_{\overline{u}} = 0 \right\}$$

  where $\delta_{\overline{u}} = 0$ indicates that adversarial attack is authorized only on the set u. The AUC is measured to compare attribution methods (↓ is better). Note this metric cannot be used to compare different networks, since it depends on the robustness of the network.

We use also several other metrics:

- Distances between explanations: to compare two explanation $f(x)$, we use either L_2 distance, or $1 - \rho$ where ρ is the Spearman rank correlation [2, 19, 59] (↓ is better).
- Explanation complexity: we use the JPEG compression size as a proxy of the Kolmogorov complexity (↓ is better).
- Stability: As proposed in [67], the Stability is evaluated by the average distance of explanations provided for random samples drawn in a ball of radius 0.3 (0.15 for cat vs dog and imagenet) around x. As before, the distance can be either L_2 or $1 - \rho$ (↓ is better).
- Accuracy: To assess the relevance of explanation [52] use a semi-real dataset, called BlockMNIST, having a random *null block and evaluate the proportion of top-k feature attribution values within the null block.*

A.4.3 Supplementary metric results

In this section we present several experiments and metrics that we were not able to insert in the core of the paper.

Deletion-zero and Insertion-zero are evaluated on CelebA and FashionMNIST dataset. It is known that the baseline value can be a bias for these metrics, and we are convinced that it has a higher influence with 1-Lipschitz networks. Even if results for Deletion-zero and Insertion-zero are less obvious than for Deletion and Insertion Uniform, we can see in Table 8, that for these metrics, the rank of Saliency is most of the time higher for OTNN.

Table 8: Insertion and Deletion metrics evaluation; GC: GradCam, GI: Gradient.Input, IG: Integrated Gradient, Saliency Rk : Rank (comparison by line only : in bold best score)

Dataset	Network	Deletion-Zero (↓ is better)					
		GC	GI	IG	Rise	Saliency	SmoothGrad
		Deletion-Zero					
CelebA	OTNN	8.01	7.04	7.05	7.09	6.98 (Rk2)	**6.96**
	Unconstrained	5.77	4.56	4.38	5.07	**4.13** (Rk1)	4.51
Fashion-	OTNN	0.24	0.16	**0.15**	0.26	0.20 (Rk4)	0.19
MNIST	Unconstrained	0.33	0.28	0.23	**0.16**	0.38 (Rk5)	0.39
		Insertion-zero (↑ is better)					
CelebA	OTNN	10.26	11.63	11.58	**15.50**	10.06 (Rk6)	10.10
	Unconstrained	14.24	11.71	12.37	**15.70**	6.67 (Rk6)	7.65
Fashion-	OTNN	0.31	0.46	**0.47**	0.36	0.36 (Rk4)	0.39
MNIST	Unconstrained	0.53	0.59	0.68	**0.73**	0.45 (Rk6)	0.46

To leverage the bias of the baseline value, as proposed in [29] we evaluated the Robustness-SR metric, Saliency map on OTNN achieves top-ranking scores. One might argue that scores for unconstrained networks are lower, but this is directly linked to the higher intrinsic robustness of OTNNand thus cannot be compared.

Table 9: Robustness-SR metrics evaluation; GC: GradCam, GI: Gradient.Input, IG: Integrated Gradient, Saliency Rk : Rank (comparison by line only : in bold best score)

Dataset	Network	Robustness-SR (↓ is better)					
		GC	GI	IG	Rise	Saliency	SmoothGrad
CelebA	OTNN	28.54	14.01	13.28	30.54	**11.64** (Rk1)	12.65
	Unconstrained	11.11	9.19	10.00	15.15	7.38 (Rk2)	**7.20**
Fashion-	OTNN	**1.69**	3.31	3.36	3.27	2.29 (Rk3)	2.01
MNIST	Unconstrained	1.17	1.36	1.17	**1.15**	1.21 (Rk4)	1.25

The full results for the explanation complexity is given on Table 10. The complexity is still lower for OTNN on FashionMNIST, even if the gap with Unconstrained networks is narrower than for CelebA.

Accuracy was also assessed by learning an OTNN (MLP architecture) on BlockMNIST dataset [52], and evaluating the proportion of top-k saliency map values that fall in the *null* block. Fig 5 presents

Table 10: Complexity of Saliency map by JPEG compression (kB): lower is better

	CelebA	FashionMNIST
OTNN	9.48	0.92
Unconstrained	16.84	0.94

several samples of input images and saliency maps of BlockMNIST dataset, to be compared to Fig.1 in [52]. It shows that OTNN gradients are almost only on the signal block (digit). And Table 11 evaluates the proxy metric proposed in [52], and confirms that OTNNs saliency maps top-k values point even more on discriminative features than those of adversarial trained networks.

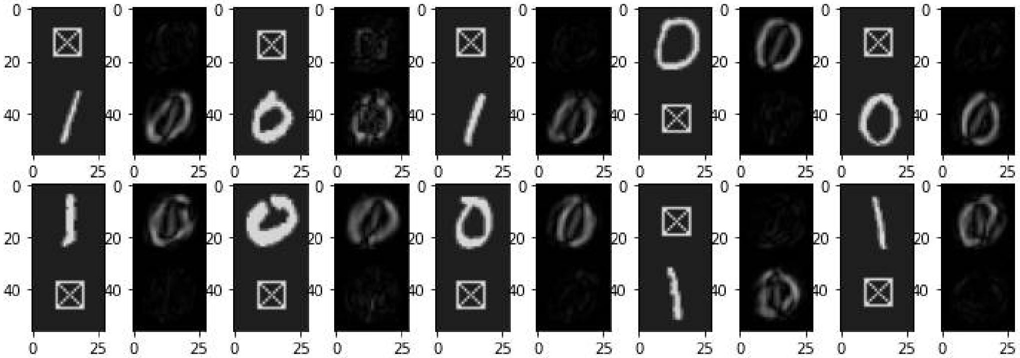


Figure 5: blockMNIST experiments using the proposed OTNN framework (10 blockMNIST images and their respective OTNN gradients). *null* marks are almost invisible in the gradients

Table 11: Comparison of Proxy metric on BlockMNIST (0 vs. 1) data for OTNNs, standard and robust models (values for the two last are directly extracted from [52])

Unmasking fraction k	2.5	5	10	15	20	25	30
Type of NN	Fraction of top-k pixels in null block						
Unconstrained [52]	43.8	42.5	44.8	46.5	47.5	48.1	48.4
Adversarial [52]	< 1	< 1	2.3	7.9	**16.7**	**24.2**	**29.9**
OTNN	< 1	< 1	**1.8**	**6.6**	16.8	26.0	32.9

A.5 Complementary qualitative results

In this section, we provide more samples of qualitative results and couterfactual exlanations for OTNN, based on the gradient, i.e. $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$ for $t > 1$.

A.5.1 FashionMNIST

Fig. 6 gives more results on FashionMNIST.

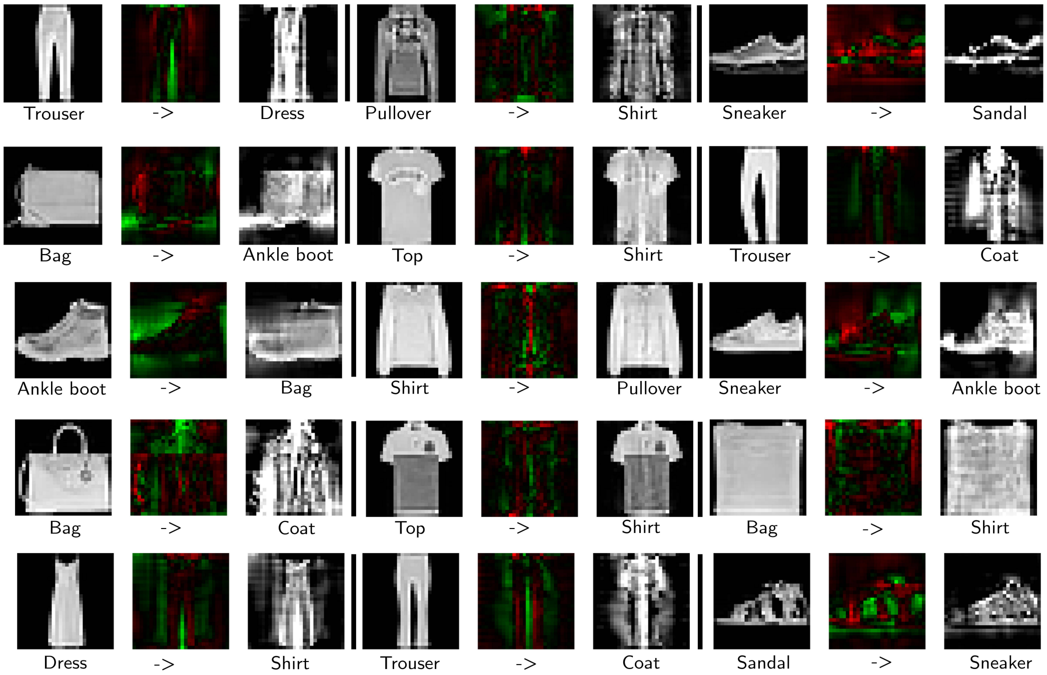


Figure 6: FashionMNIST samples

A.5.2 CelebA

We presents results for other labels of CelebA. For ethic concerns we have hidden labels that can be subject to misinterpretation, such as *Attractive*, *Male*, *Big_Nose*. Fig. 7 to 25 present more results on the labels presented in the core of the paper, *Mouth_Slightly_Open*, *Mustache*,*Wearing_Hat*.

A.5.3 Cat vs Dog

We present some supplementary comparison of Saliency Maps and counterfactual examples for cat vs dog(Fig. 26 and 27).

A.5.4 Imagenet

We present some supplementary comparison of Saliency Maps Imagenet (Fig. 28). As pointed out previously, our model doesn't produce significant counterfacutal explanations on Imagenet.

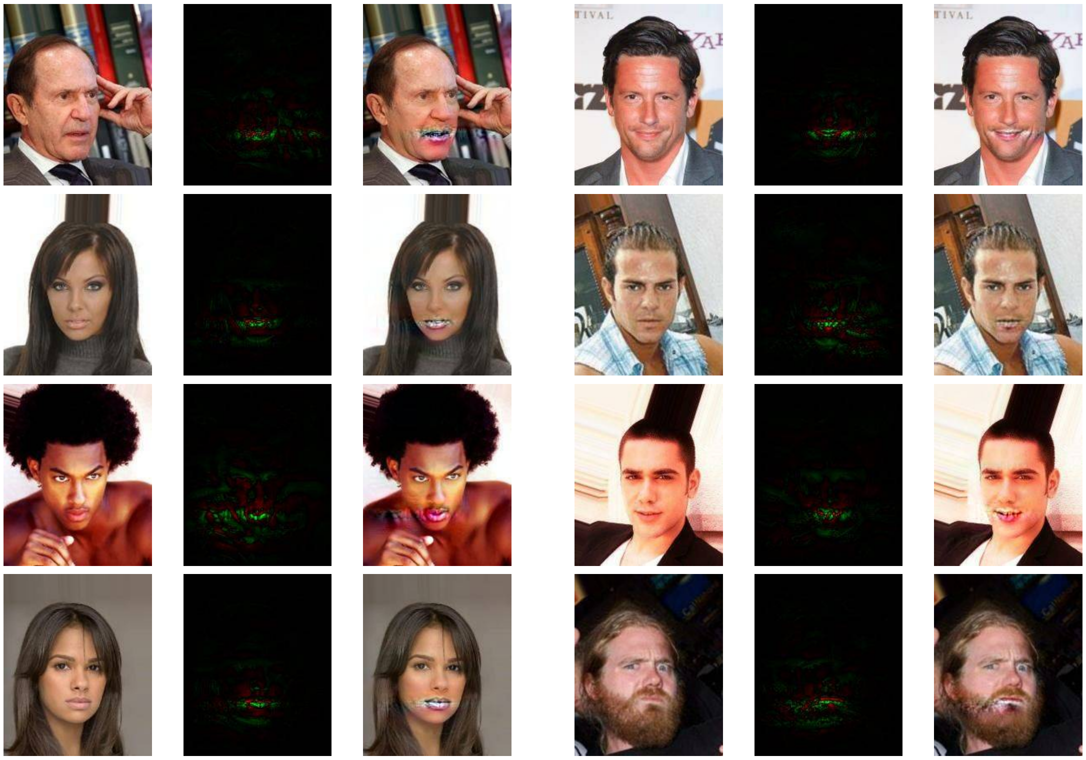


Figure 7: Samples from label Mouth_slightly_open: left source image (closed) , center difference image, right counterfactual (open) of form $\boldsymbol{x} - 10 * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$

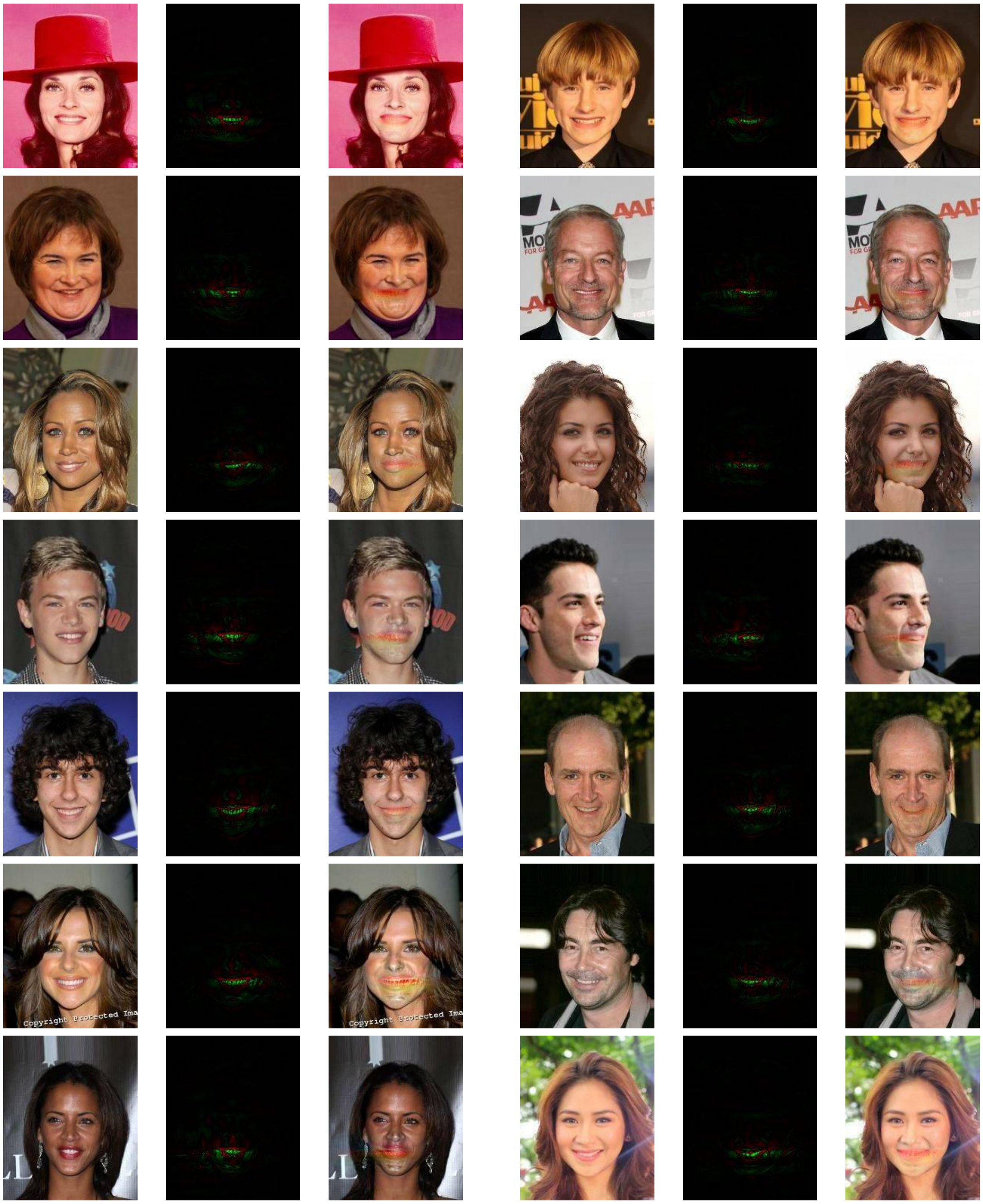


Figure 8: Samples from label Mouth_slightly_open: left source image (open) , center difference image, right counterfactual (close) of form $\boldsymbol{x} - 10 * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$

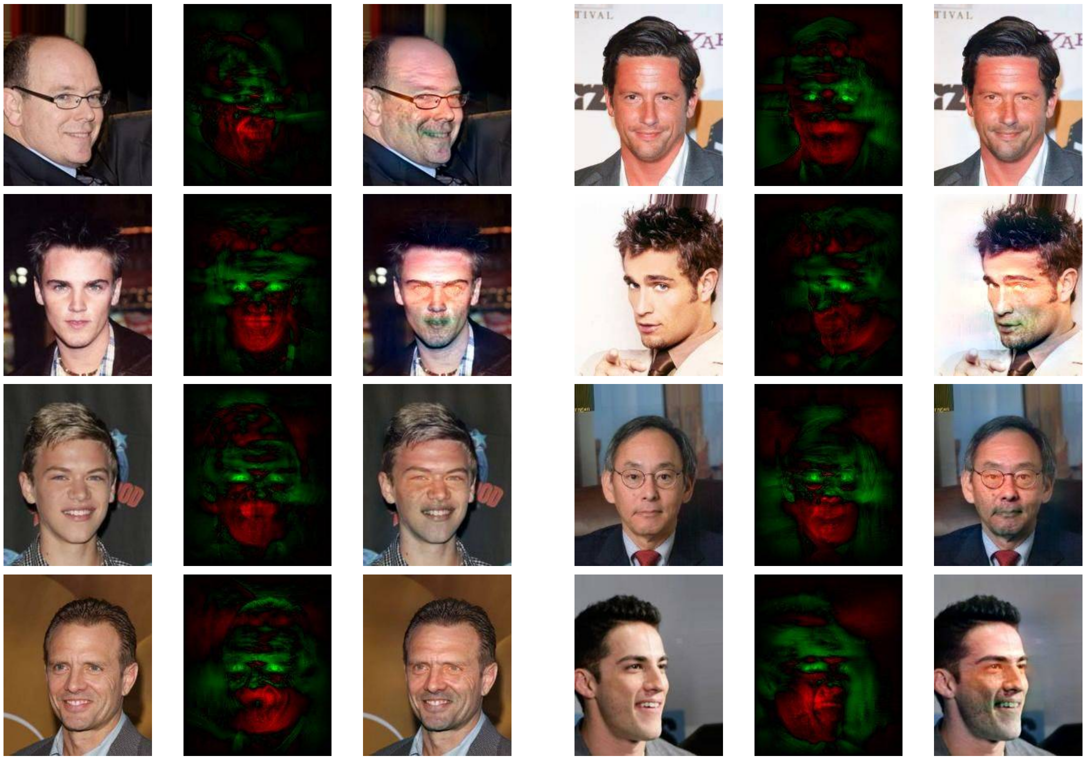


Figure 9: Samples from label Mustache: left source image (no mustache) , center difference image, right counterfactual (mustache) of form $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$ with $t \in \{5, 10, 20\}$

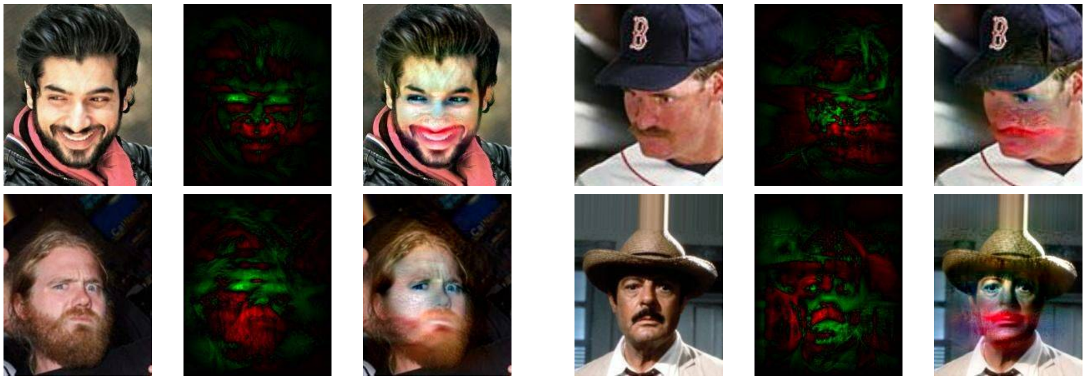


Figure 10: Samples from label Mustache: left source image (Mustache) , center difference image, right counterfactual (Non Mustache) of form $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$, $t \in 5, 10$

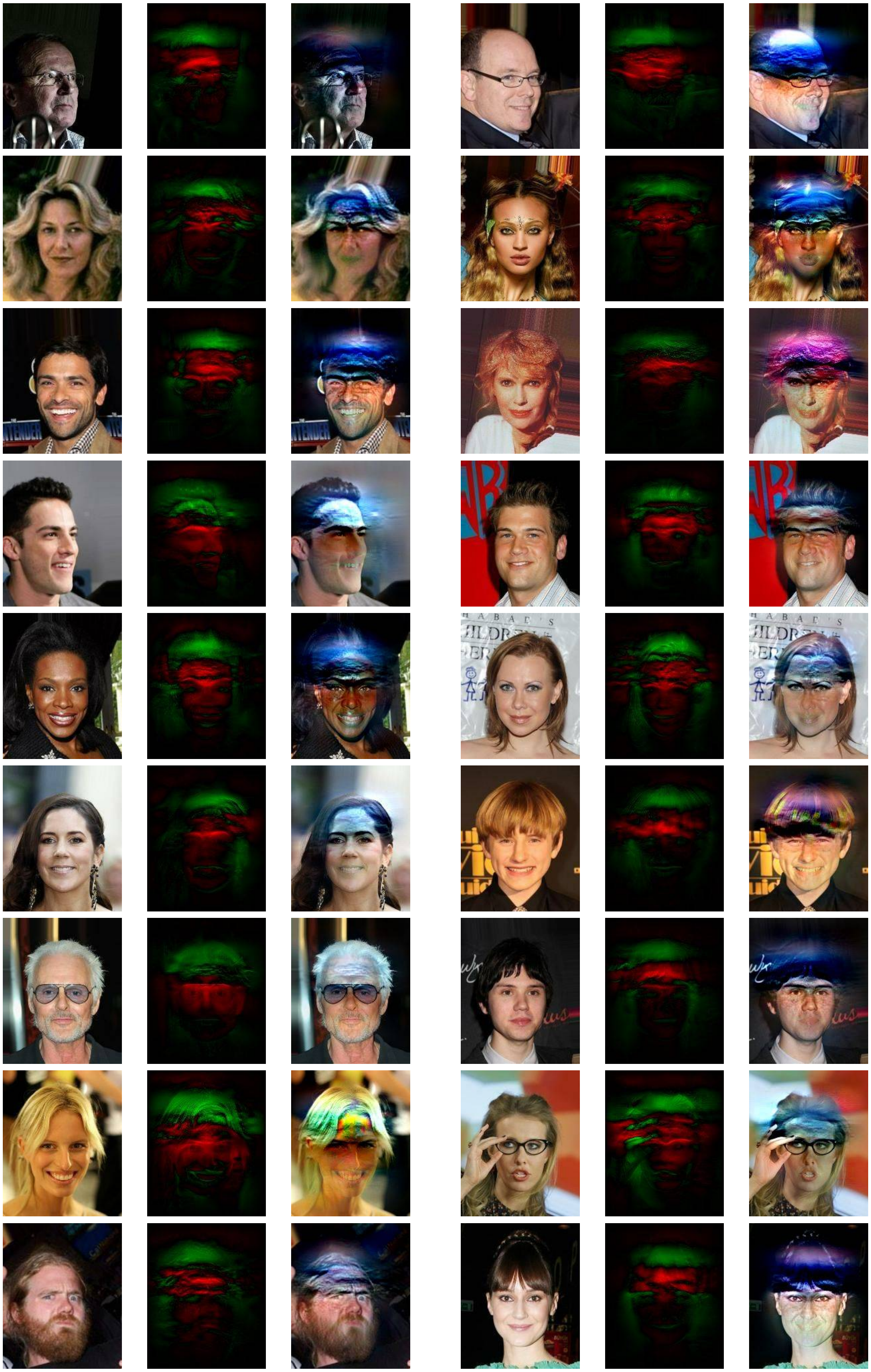


Figure 11: Samples from label Wearing Hat: left source image (No Hat) , center difference image, right counterfactual (Hat) of form $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x}), t \in 5, 10$

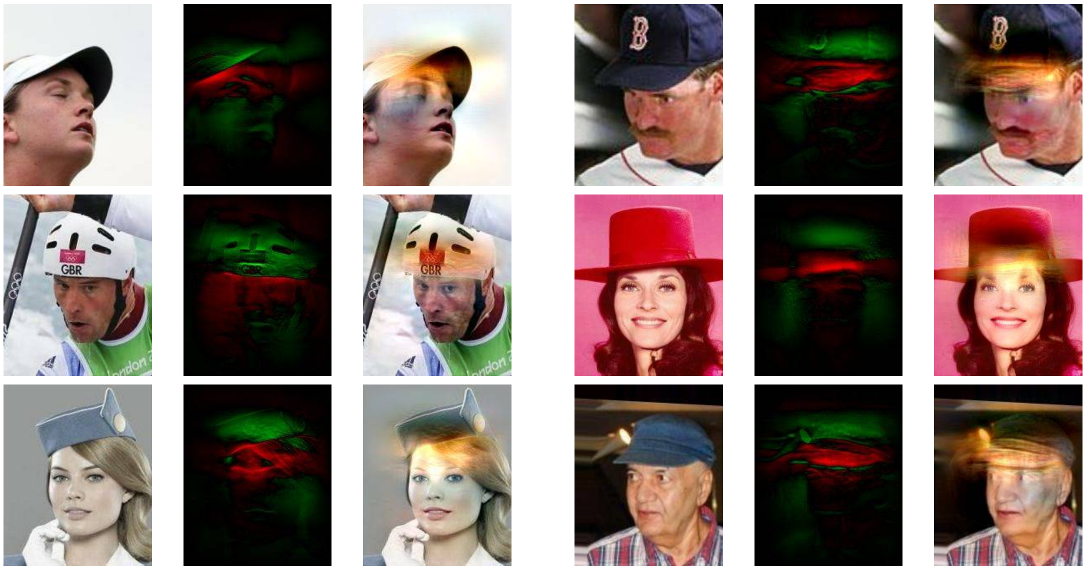


Figure 12: Samples from label Wearing Hat: left source image (Hat) , center difference image, right counterfactual (No Hat) of form $\boldsymbol{x} - t * \hat{f}(\boldsymbol{x})\nabla_x \hat{f}(\boldsymbol{x})$, $t \in 5, 10$

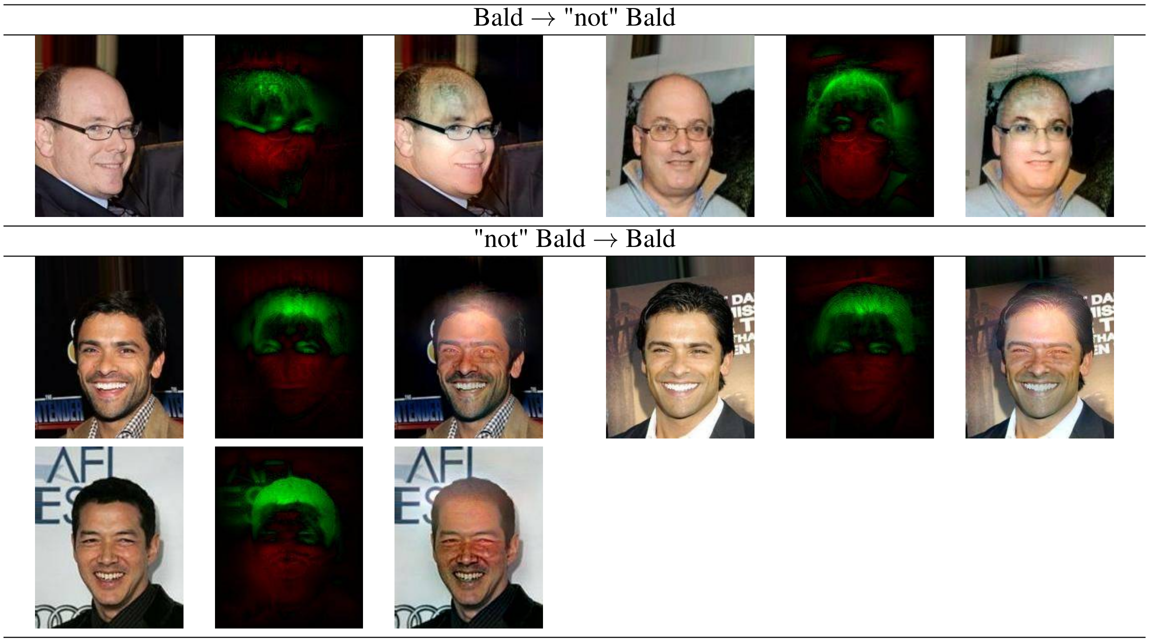


Figure 13: Samples from label Bald

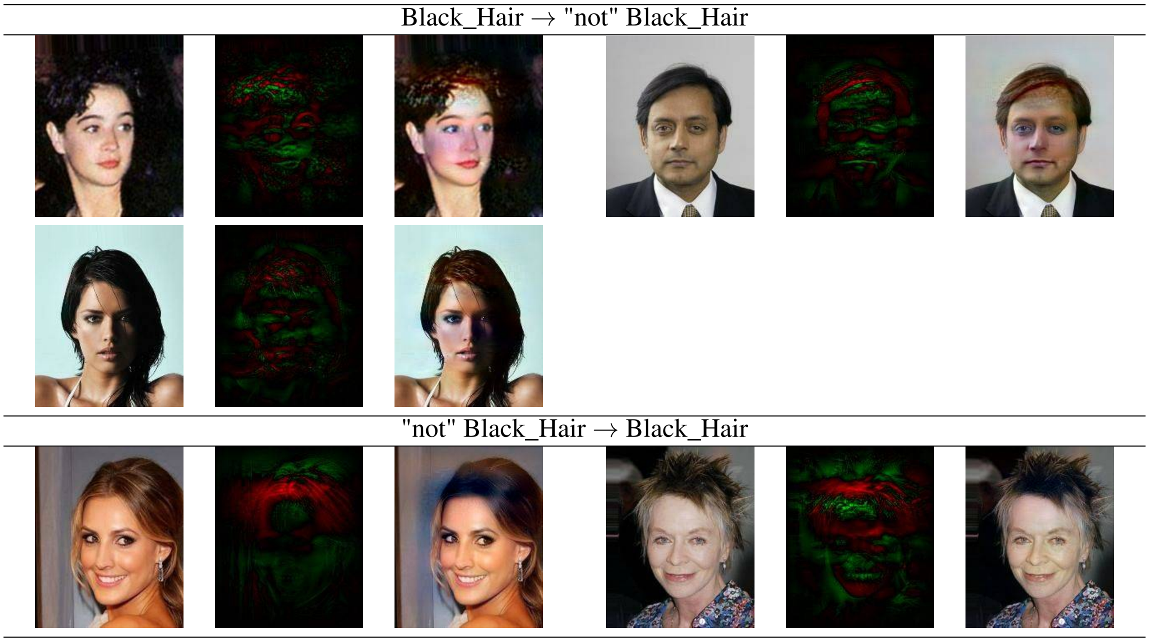


Figure 14: Samples from label Black_Hair

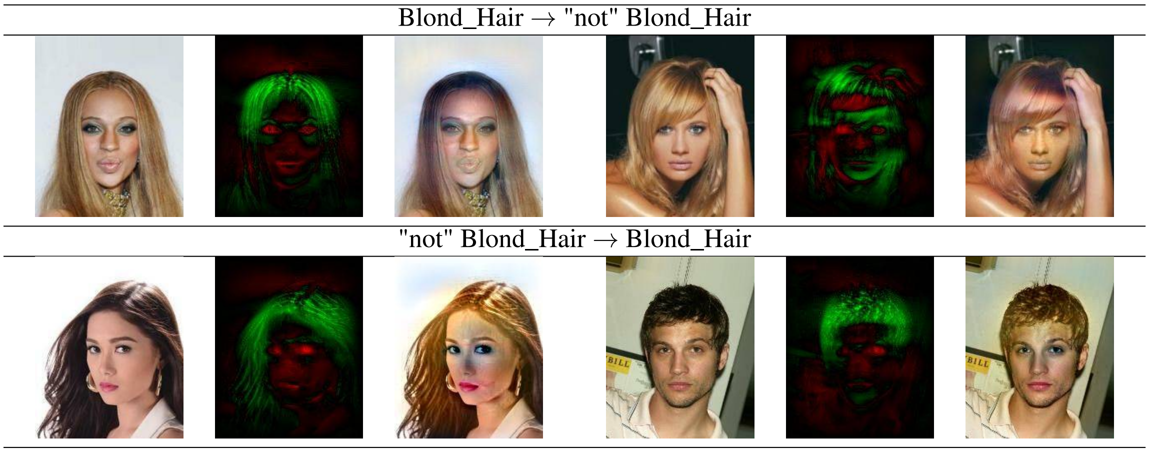


Figure 15: Samples from label Blond_Hair

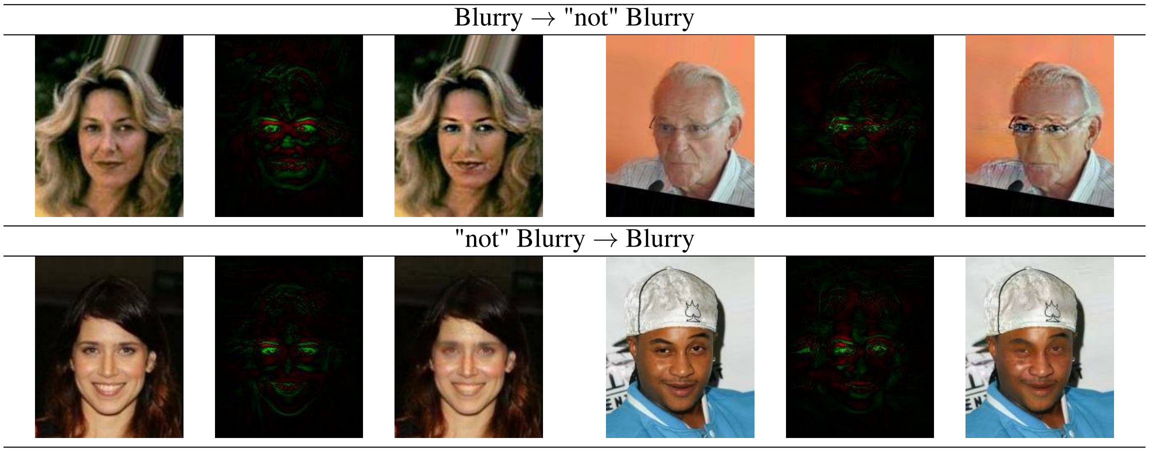
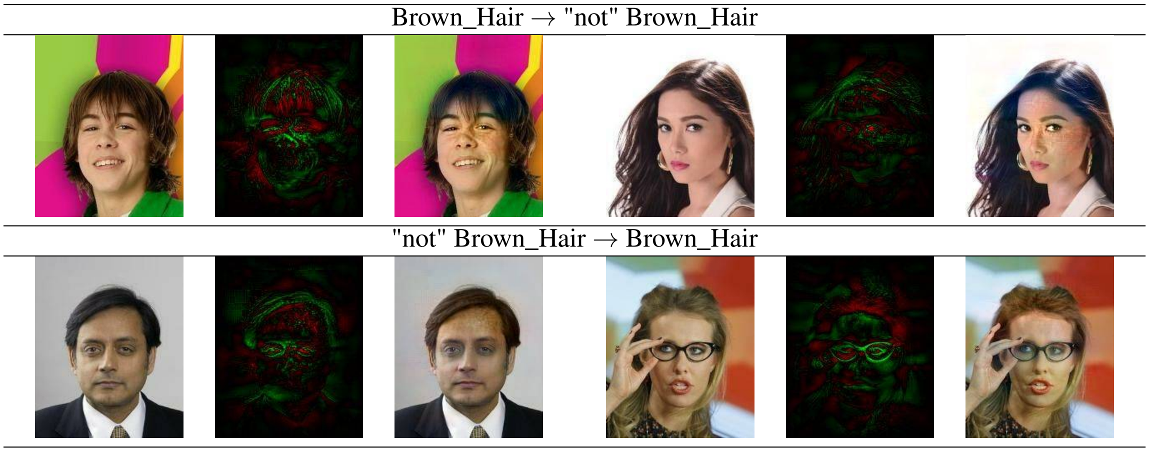


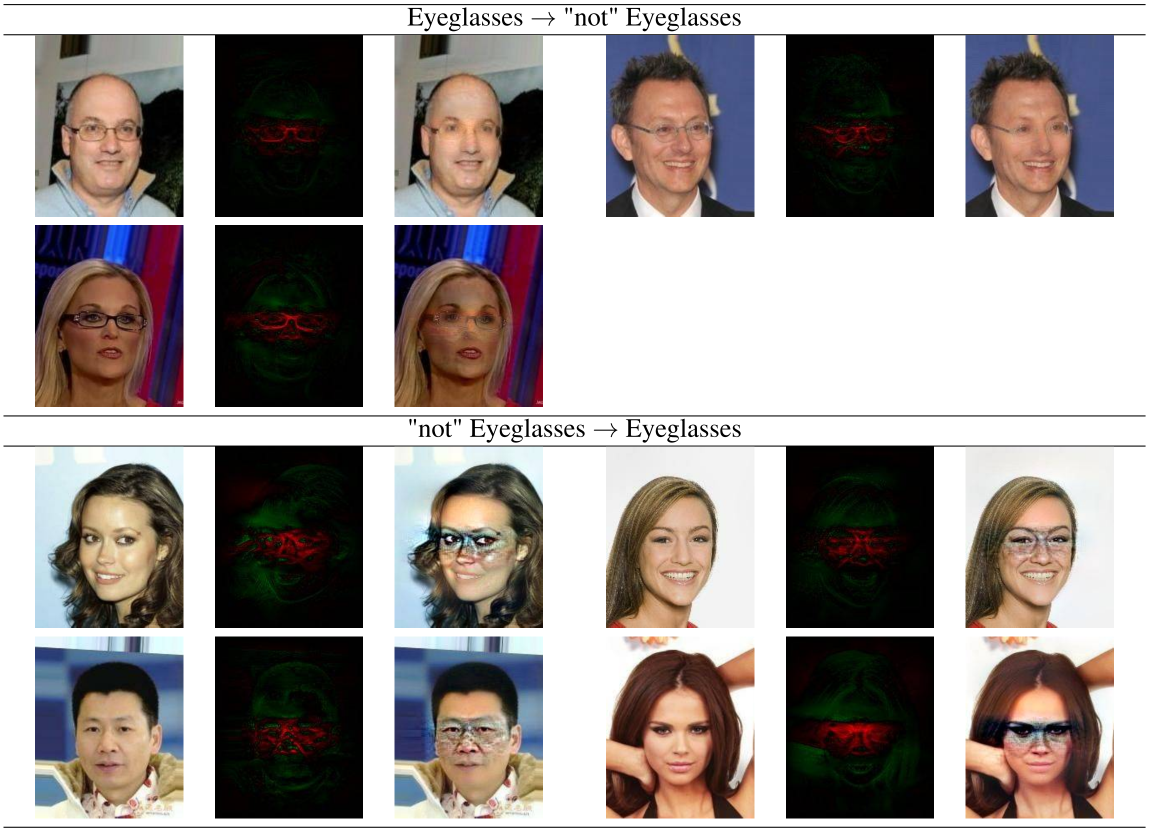
Figure 16: Samples from label Blurry



Brown_Hair → "not" Brown_Hair

"not" Brown_Hair → Brown_Hair

Figure 17: Samples from label Brown_Hair



Eyeglasses → "not" Eyeglasses

"not" Eyeglasses → Eyeglasses

Figure 18: Samples from label Eyeglasses

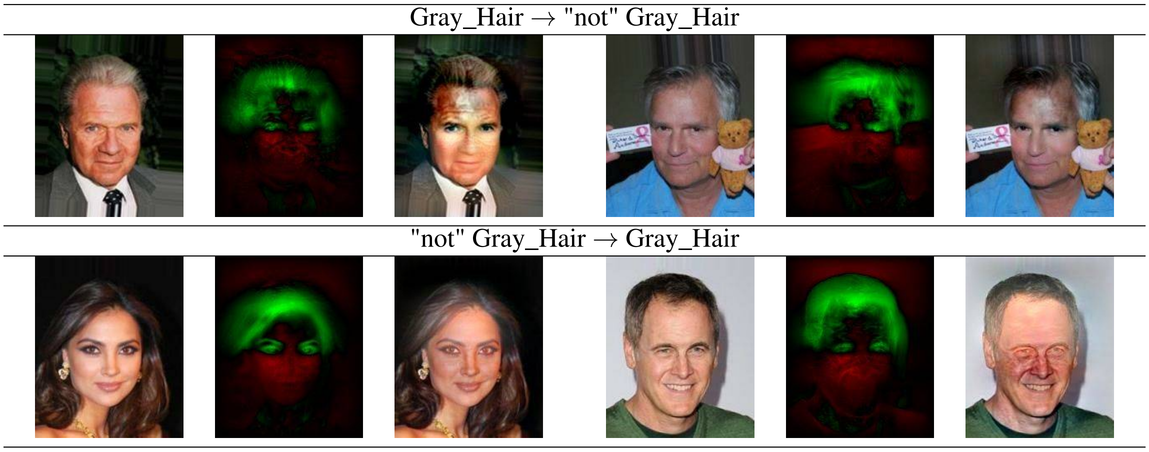


Figure 19: Samples from label Gray_Hair

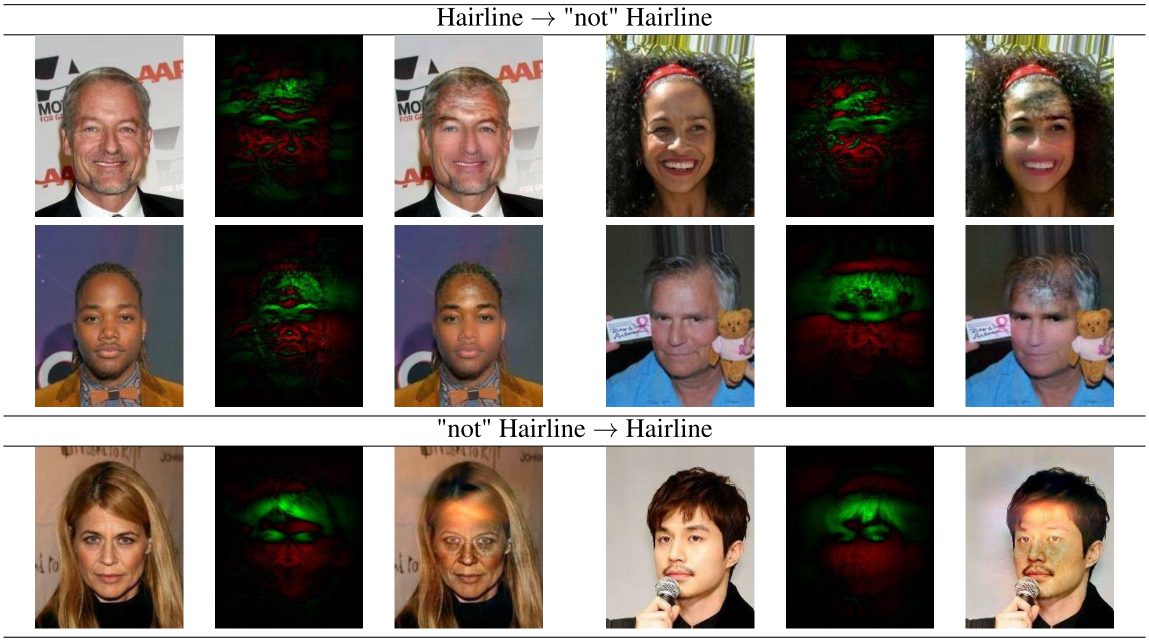
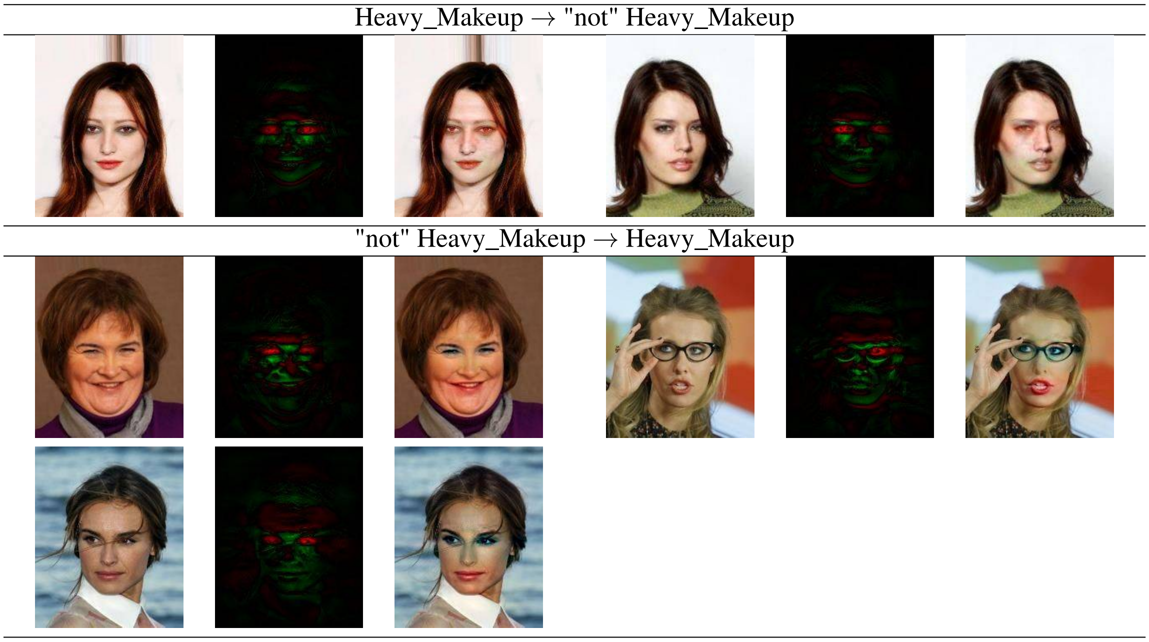


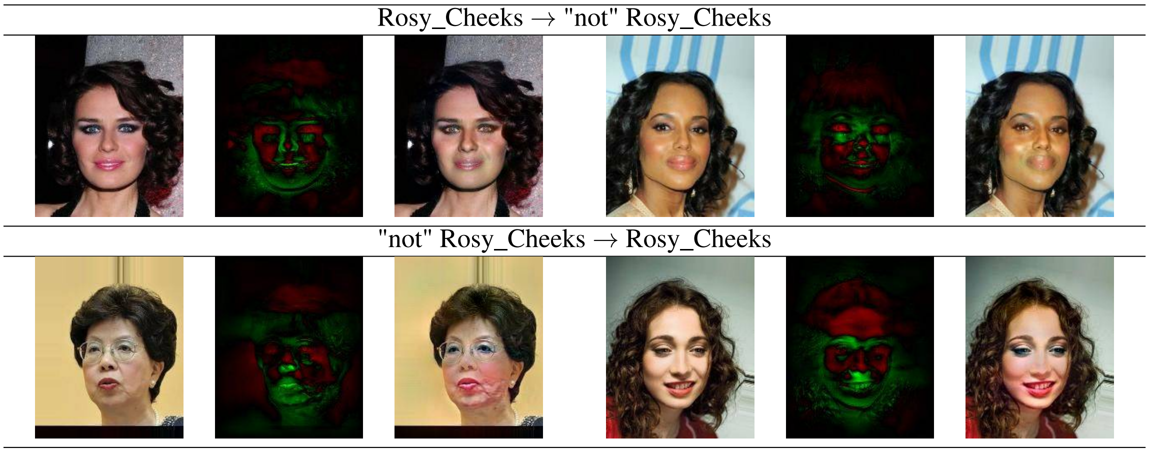
Figure 20: Samples from label Hairline



Heavy_Makeup → "not" Heavy_Makeup

"not" Heavy_Makeup → Heavy_Makeup

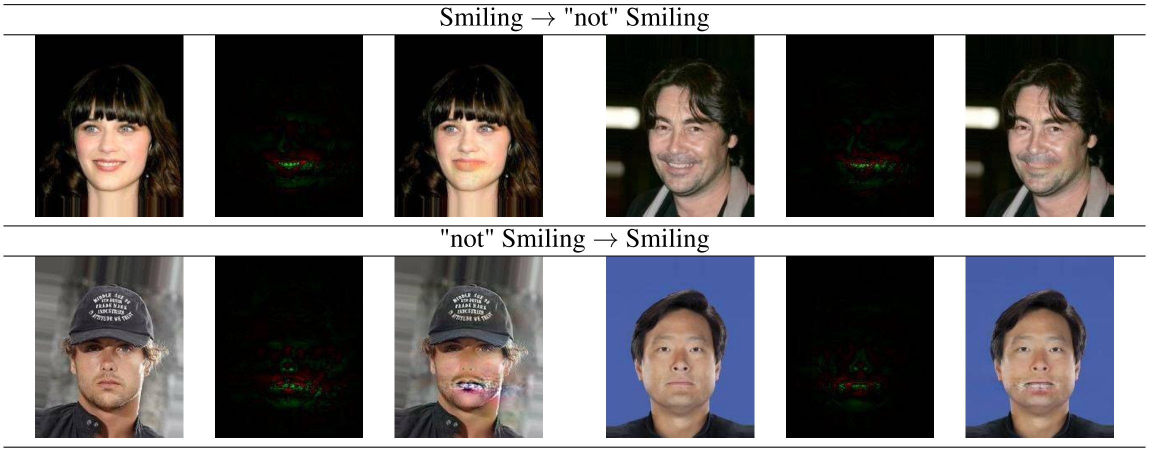
Figure 21: Samples from label Heavy_Makeup



Rosy_Cheeks → "not" Rosy_Cheeks

"not" Rosy_Cheeks → Rosy_Cheeks

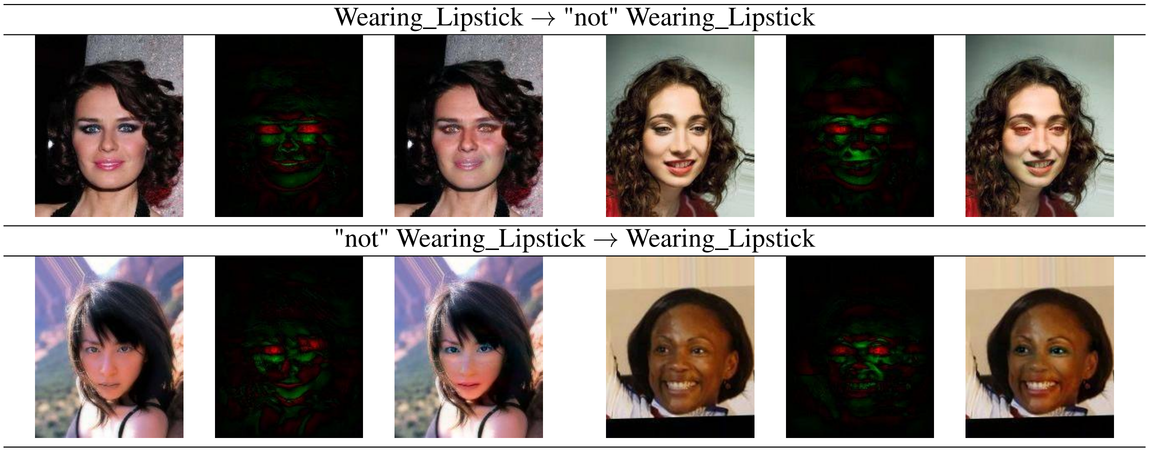
Figure 22: Samples from label Rosy_Cheeks



Smiling → "not" Smiling

"not" Smiling → Smiling

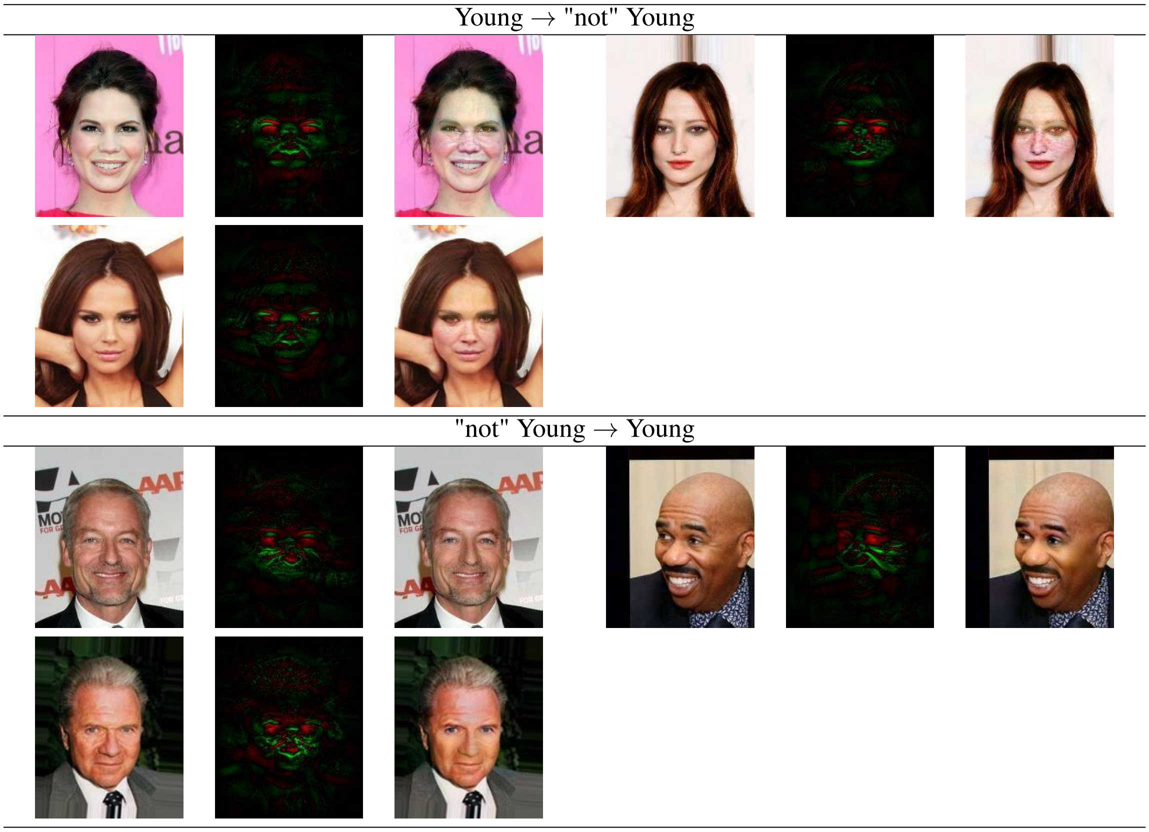
Figure 23: Samples from label Smiling



Wearing_Lipstick → "not" Wearing_Lipstick

"not" Wearing_Lipstick → Wearing_Lipstick

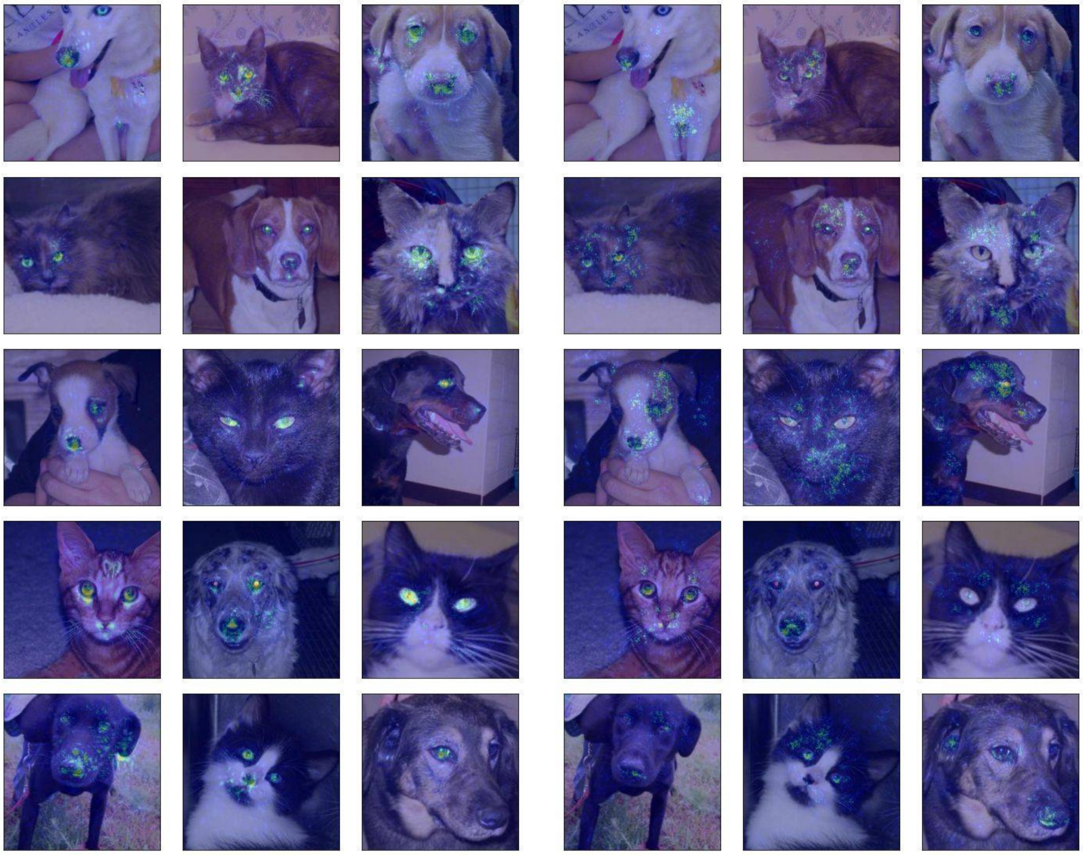
Figure 24: Samples from label Wearing_Lipstick



Young → "not" Young

"not" Young → Young

Figure 25: Samples from label Young



(a) OTNN

(b) Unconstrained

Figure 26: Cat vs Dog Saliency Map samples

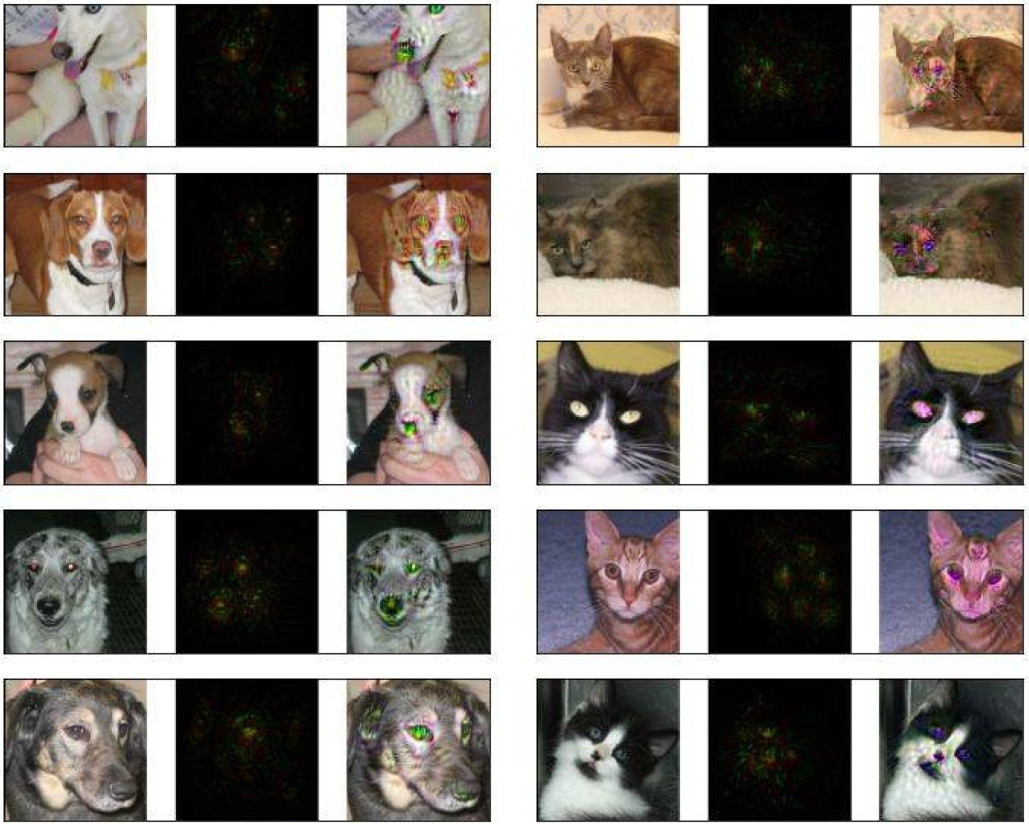
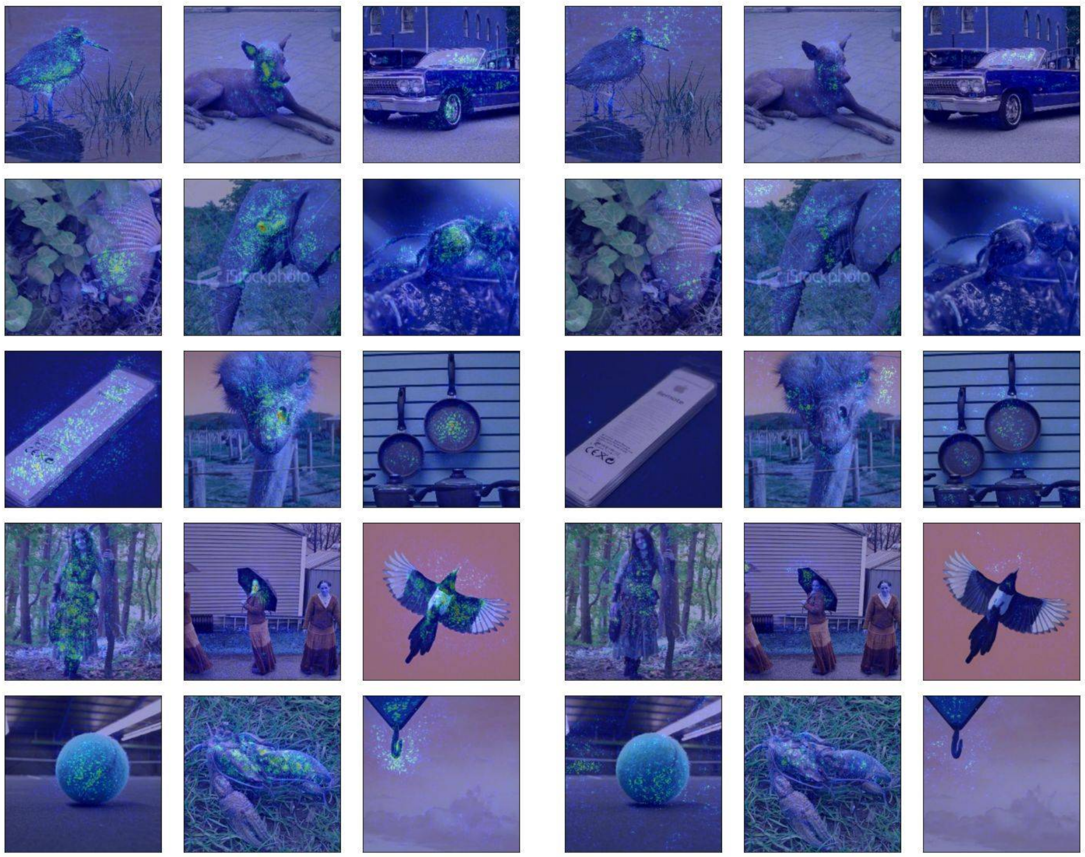


Figure 27: Cat vs Dog Saliency counterfactual samples. Left dog to cat, right cat to dog



(a) OTNN

(b) Unconstrained

Figure 28: Imagenet Saliency Map samples