Counterfactual-based Saliency Map: Towards Visual Contrastive Explanations for Neural Networks

Xue Wang[‡], Zhibo Wang[†,‡,*],Haiqin Weng[♯],Hengchang Guo[‡], Zhifei Zhang[⋆], Lu Jin[♯],Tao Wei[♯],Kui Ren[†]
[†]School of Cyber Science and Technology, Zhejiang University, P. R. China,
[‡]School of Cyber Science and Engineering, Wuhan University, P. R. China,
[♯] Ant Group, [⋆]Adobe Research
{shannonwang, hc_guo}@whu.edu.cn, {zhibowang, kuiren}@zju.edu.cn,
{haiqin.wenghaiqin, lyla.jl, lenx.wei}@antgroup.com, zzhang@adobe.com

Abstract

*Explaining deep models in a human-understandable way has been explored by many works that mostly explain why an input causes a corresponding prediction (*i.e., *Why P?). However, seldom they could handle those more complex causal questions like "Why P rather than Q?" and "Why one is P, while another is Q?", which would better help humans understand the behavior of deep models. Considering the insufficient study on such complex causal questions, we make the first attempt to explain different causal questions by contrastive explanations in a unified framework, *i.e.*, Counterfactual Contrastive Explanation (CCE), which visually and intuitively explains the aforementioned questions via a novel positive-negative saliency-based explanation scheme. More specifically, we propose a content-aware counterfactual perturbing algorithm to stimulate contrastive examples, from which a pair of positive and negative saliency maps could be derived to contrastively explain why P (positive class) rather than Q (negative class). Beyond existing works, our counterfactual perturbation meets the principles of validity, sparsity, and data distribution closeness at the same time. In addition, by slightly adjusting the objective of perturbation, our framework can adapt to different causal questions. Extensive experimental evaluation demonstrates the effectiveness and superior performance of the proposed CCE on different benchmark metrics for interpretability, including Sanity Check, Class Deviation Score and Insertion-Deletion tests. A user study is conducted and the results show that user confidence is increasing significantly when presented with CCE compared to standard saliency map baselines.*

1. Introduction

Deep neural networks (DNNs) have achieved breakthrough performance in various computer vision tasks, *e.g.*, image classification [19, 34], object detection [28, 29], semantic segmentation [23], *etc*. However, the "black-box" nature of DNNs, *i.e.*, massive unexplainable parameters and lack of intuitive understanding in prediction, has been drawing chaos in the human-understandable analysis of their behavior. Since human-understandable explanations of DNNs would significantly facilitate the causal analysis, *e.g.*, what causes misclassification or bias, it is rising increasing demand for related research. One of the popular approaches to interpreting deep learning models is to display visual explanations in the form of saliency maps [6, 32, 33, 36]. Most saliency-based approaches highlight the most informative areas by assigning weights to image regions concerning their contributions to the final prediction, revealing their causal relationship.

However, in the field of social science, Miller [26] systematically surveyed explanation methods and pointed out that most explanations are contrastive. That is, people not only want to know "Why P?", where P refers to the given conclusion, but also have great interest in "Why P, rather than Q?", in which Q is often implicit from the context. Van Bouwel and Weber [40] defined the latter question as P-contrast referring to differences that occur on properties within an object, and formally state it as "Why does object a have property P, rather than property Q?" Saliency maps efficiently answer the question "Why P?" by highlighting the most informative regions, but it does not take the P-contrast question into account. For example, when an application for a loan is denied by the bank, the applicant would be left wondering not only why was the loan denied, but also how to pass the application. The saliency maps might only point out that the income is the main reason for denial but fail to state the difference between approval and denial.

For structured data such as tabular data, counterfactual explanation [42, 7] is a popular method to answer P-contrast question. A counterfactual explanation modifies the input

[*]Zhibo Wang is the corresponding author.

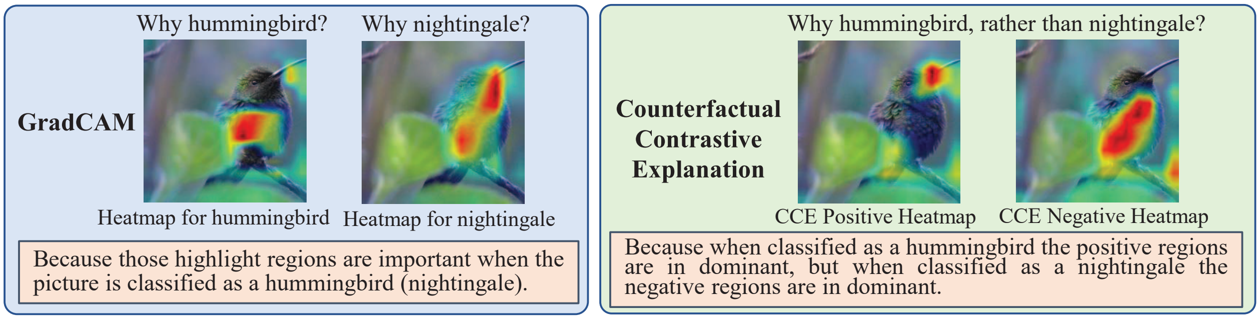

Why hummingbird?
Why nightingale?
GradCAM
Heatmap for hummingbird
Heatmap for nightingale
Because those highlight regions are important when the picture is classified as a hummingbird (nightingale).
Why hummingbird, rather than nightingale?
Counterfactual Contrastive Explanation
CCE Positive Heatmap
CCE Negative Heatmap
Because when classified as a hummingbird the positive regions are in dominant, but when classified as a nightingale the negative regions are in dominant.

Figure 1: Comparison between GradCAM and our proposed Counterfactual Contrastive Explanation (CCE), *i.e.*, non-contrastive vs. contrastive explanations.

appropriately to change the output of the model from P to Q. For the previous example, a potential counterfactual explanation could be: "The loan was denied as your income was $20,000, you would have been offered a loan if your income had increased to $25,000." These approaches emphasize the counterfactual example itself as the explanation and ignore the importance of considering the model's decision changes in a contrasting perspective. For users, "increasing income by $500" is a more accurate explanation, which can be easily converted in tabular data. Several previous studies [4, 5, 12, 44, 16] have proposed methods to answer P-contrast question, they use GAN to generate counterfactuals as explanations directly or find additional images as counterfactuals to show contrast. However, for unstructured data such as images, the difference between counterfactual and original sample is not intuitive, and the correlation between counterfactual and model decision is lower than that of tabular data.

Besides P-contrast question, Van Bouwel and Weber [40] also defined O-contrast question, that is "Why does object a have property P, while another object b has property Q?", which note differences occur on properties between objects themselves. In deep learning, this question is highly relevant to the adversarial example [39], that carefully constructed perturbations are added to input and aimed to maximize the change of the model's decision. Typically, saliency maps explain an adversarial example and its benign counterpart by highlighting different regions. However, Zhang et al. [47] proposed an attack method that changes the model decision while keeping the saliency maps unchanged. This study reveals that the existing explanation approaches are not fully aligned with the model decisions and cannot provide convincing explanations for the phenomenon of adversarial examples. We argue that to interpret the adversarial example phenomenon, it is essential to answer the O-contrast question "Why is the original input assigned to class P, while the adversarial example is considered to be class Q?", which is not considered in existing explanation approaches.

Therefore, we propose a novel visual explanation method, *i.e.*, Counterfactual Contrastive Explanation (CCE), which gives the first attempt to answer "Why P?", P-contrast, and O-contrast questions in a unified framework for image classification task. As illustrated in Fig. 1, our method contrastively explains why the image is classified as a hummingbird rather than a nightingale, while the traditional way naïvely explains the two categories in separate and overlapped saliency maps that are difficult to tackle contrastive questions. The core of contrastive explanation is to identify suitable contrastive objects, such that we could contrastively reason what results in different predictions. Correspondingly, we propose to counterfactually synthesize such contrastive objects. Ideally, a counterfactual example should minimally change the features but maximally differ the predictions [42]. Towards this end, we generate sparse and content-aware counterfactual perturbations that are overlayed onto the original image to change the model confidence in the target label while preserving the confidence in the rest classes. Intuitively, comparing such counterfactual examples with their original inputs is supposed to generate visual contrastive explanations. To better visualize dominant features in a contrastive manner, we propose gradient-aware feature maps that utilize class-specific gradients to achieve decoupled saliency maps based on original and counterfactual examples. More specifically, we disentangle feature representations of DNNs into positive and negative saliency maps, which contrastively visualize what features dominate corresponding predictions. We summarize our contributions as below:

- To the best of our knowledge, we give the first attempt to explain three types of causal questions "Why P?", "Why P rather than Q?", and "Why a is P while b is Q?" in a unified framework, which drastically boosts generalization capacity as compared to existing works.
- We propose a positive-negative saliency scheme to provide visual contrastive explanations, which can intuitively answer the aforementioned causal questions by comparing original and counterfactual examples.
- We propose a counterfactual perturbing algorithm for

images to generate counterfactual examples that simultaneously satisfy the principles of validity, sparsity, and data distribution closeness by constraining the changes of logits.

- Extensive experimental results demonstrate that the proposed counterfactual contrastive explanations (CCE), outperform existing works in multiple interpretability metrics including Sanity Check [3], Class Deviation Score and Insertion-Deletion tests [30]. Further, we conducted a user study demonstrating the effectiveness of CCE in helping users gain a deeper understanding of model's prediction.

2. Related Works

Numerous methods have been proposed to explain model decisions by highlighting the important regions that are responsible for the predictions, which can be categorized as feature attribution-based explanation methods and visual counterfactual-based explanation methods.

2.1. Feature Attribution-based Explanations

Feature attribution-based explanation methods can be further categorized as perturbation-based, propagation-based and activation-based methods. (1) Perturbation-based methods [41, 15] generally occlude the input image using different types of perturbations and then record the corresponding change of the class score, thus determining the regions that are important to a certain class. These methods can generate heatmaps straightforwardly but are inefficient. (2) Propagation-based methods [33, 36, 38] use gradients (back-propagated from the model output to the input) to visualize relevant regions (*i.e.*, saliency maps) for a given class. However, such gradient-based saliency maps are usually visually noisy. (3) Activation-based methods [32, 6, 43, 18] highlight the key regions by resorting to the activation of feature maps, in which the GradCAM [32] is the most widely used method which visualizes CNNs by weighting the feature maps with gradients.

2.2. Visual Counterfactual-based Explanations

Visual counterfactual-based explanation methods construct counterfactual examples to change the model output and then locate the important regions for the model by comparing the difference between counterfactual and original examples. Chang et al. [5] employed a generative model to find the smallest region in the image that can change the classifier's prediction and then obtain the resultant saliency map. Similarly, Dhurandhar et al. [8] proposed an optimization objective to search for minimal pixels that are sufficiently present or necessarily absent to produce a contrastive explanation. Hendricks et al. [14] utilized a recurrent neural network to generate several candidate counterfactual explanations, and select the most class-specific one to be the counterfactual image while being the most relevant to the original input image. Elliott et al. [9] proposed the Perceptual Perturbations(PPE) as Explanations to generate explainable adversarial examples to work as counterfactual explanations by adding extra visual similarity constraints.However, we argue that the counterfactual examples that only consider visual constraints would involve extra unexpected semantic priors, violating the minimization principle of counterfactual explanations. CE[24] and SCOUT [44] are newest counterfactual-based explanation methods but none of them is aimed at O-contrast question.

3. Counterfactual Contrastive Explanations

In this paper, we propose the Counterfactual Contrastive Explanation (CCE) to provide a visual explanation for three types of causal questions which can be redefined as follows: 1) "Why P", *i.e.*,"Why is image a labeled as class P, rather than not-P?". 2) P-contrast question, *i.e.*, "Why is image a labeled as class P, rather than Q?". 3) O-contrast question, *i.e.*,"Why is image a labeled as class P, while image b is labeled as class Q?". This section will first overview the proposed CCE, and then detail the generation of counterfactual examples and the designed flow of explanation. Finally, we will further describe how to use our framework to explain different causal questions.

The flow of CCE is shown in Fig. 2, which mainly consists of three steps: 1) Sparse Counterfactual Example Generation, 2) Weighted Class Activation Feature Maps, and 3) Contrastive Saliency Map Generation. This framework takes the original input as the side to be interpreted and the corresponding counterfactual example as the other for comparison, and finally outputs contrast saliency maps to contrastively explain the question. First, we generate different types of counterfactual example for contrast to the original example according to the question. Then, we calculate the model's feature representations on the original and counterfactual examples based on gradient-weighted feature maps to determine those dominant features in the model decision. Finally, we compare the feature representation differences and derive positive and negative saliency maps to answer the questions.

To accomplish different types of causal questions, we propose to design corresponding types of counterfactual examples. For the "Why P" question, the optimization objective of the counterfactual example is solely to decrease the probability of class P, which is expected to locate those P-specific features. Then, for the P-contrast question, the counterfactual example should be generated by increasing the probability of class Q while maintaining the probability distribution of the other classes, which can help determine those Q-specific features. When it comes to the O-contrast question, we can decompose it into a combination of two

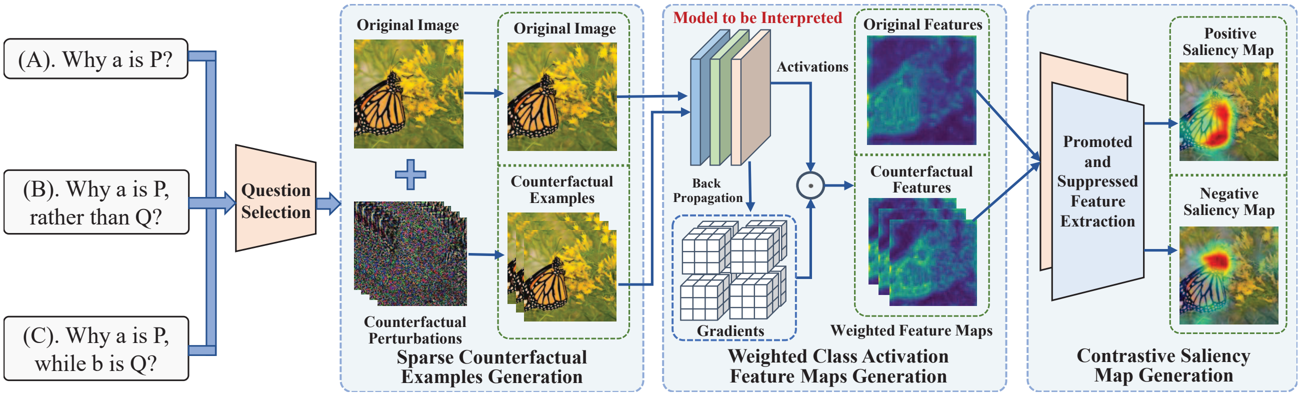

(A). Why a is P?
(B). Why a is P, rather than Q?
(C). Why a is P, while b is Q?
Question Selection
Original Image
Original Image
Counterfactual Examples
Counterfactual Perturbations
Sparse Counterfactual Examples Generation
Model to be Interpreted
Activations
Back Propagation
Gradients
Original Features
Counterfactual Features
Weighted Feature Maps
Weighted Class Activation Feature Maps Generation
Promoted and Suppressed Feature Extraction
Positive Saliency Map
Negative Saliency Map
Contrastive Saliency Map Generation

Figure 2: Overview of the proposed Counterfactual Contrastive Explanation (CCE).

P-contrast questions, that is"Why is a labeled as class P, rather than Q" and "Why is b labeled as class Q, rather than P?". The counterfactual example can be regarded as a bridge between original inputs and adversarial examples, and the explanation of the O-contrast question also can provide a more comprehensive explanation about the existence phenomenon of adversarial examples.

3.1. Sparse Counterfactual Example Generation

To achieve a contrastive explanation, we initially need to generate reasonable counterfactual examples that serve as contrasts to the original sample. Existing work focused only on the probability change of the target class when generating counterfactuals, which may lead to drastic changes in the probability of non-target classes. To address this issue, we explicitly regulate both the probability change of the target and non-target classes, and thus make the generated counterfactual examples cross the correct classification boundaries.

Given an input image $x_0 \in \mathbb{R}^{H\times W\times 3}$ and a deep neural network $\mathcal{F}$ whose logit output is denoted as $\mathcal{Z}\left(x_0\right)$, and $\max \mathcal{Z}\left(x_0\right) = \mathcal{Z}_m\left(x_0\right)$ when image x_0 is correctly classified to class M. We aim to find a perturbation δ to construct counterfactual example x_c, where $x_c = x_0 + \delta$. We will generate two different types of counterfactual examples according to the questions that we need to answer.

For the "Why P?" question where P is an arbitrary class, the counterfactual example minimize the logit for class P and maintain logits for other classes $\overset{\neg}{p}$ unchanged. Then, the proposed objective function can be written as follows:

$$\begin{aligned}&\arg\min_{x_c}\ \mathcal{Z}_p(x_c)^2 + \gamma D_{\text{mse}}\left(\mathcal{Z}_{\overset{\neg}{p}}\left(x_0\right), \mathcal{Z}_{\overset{\neg}{p}}\left(x_c\right)\right),\\ &\quad s.t.\ \left\|x_0 - x_c\right\|_2 \le \epsilon\end{aligned} \tag{1}$$

where D_{mse} is the mean square error metric to measure the distance between two distribution, the hyperparameter γ is used to balance the two losses, and the ϵ is the perturbation threshold. The distance constraint between the counterfactual example and the original sample ensures the counterfactual example is close to the data manifold.

For the "Why P, rather than Q?" question, the optimization objective is to minimize the logit for class P and maximize the logit for target class Q while maintaining rest logits of the other classes r and can be formulated as follows:

$$\begin{aligned}&\arg\min_{x_c}\ \mathcal{Z}_p(x_c)^2 - \mathcal{Z}_q(x_c)^2 + \gamma D_{\text{mse}}\left(\mathcal{Z}_r\left(x_0\right), \mathcal{Z}_r\left(x_c\right)\right),\\ &\quad s.t.\ \left\|x_0 - x_c\right\|_2 \le \epsilon.\end{aligned} \tag{2}$$

The optimization objectives (*i.e.*, Eq. 1 and Eq. 2) can be solved by existing optimization methods, and we adopt the same optimization algorithm like PGD [25] in this work.

3.2. Weighted Class Activation Feature Maps

Existing counterfactual explanation methods usually treat counterfactual perturbations directly as explanations. In contrast, we propose that the difference between the counterfactual example and the original example in the model's feature representation is a more consistent explanation for the model decision. Inspired by GradCam [32], we also utilize weighted class activation feature maps to construct the **feature representation**.

We denote $\mathcal{F}_t\left(x_0\right)$ as the probability score on any targeted class t, and l represents the index of the internal layer to be explained, which contains K feature maps. The activation of the layer is $\mathcal{A}_l$ and the k-th feature map can be denoted as $\mathcal{A}_l^k$. For any target class t, the feature representation $R_t\left(x_0\right)$ is defined as:

$$R_t\left(x_0\right) = \sum_{k=1}^{K}\left(w_t^k \cdot \mathcal{A}_l^k\right), \tag{3}$$

where w_t^k is the weight of the corresponding feature map $\mathcal{A}_l^k$. Here, we use the gradients with respect to k-th feature map $\mathcal{A}_l^k$ as the weight. Then, the gradients are globally average pooled over the height and width dimensions (indexed by i, j respectively) to obtain the channel-wise weights:

$$w_t^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial \mathcal{F}_t\left(x_0\right)}{\partial A_{ij}^k\left(x_0\right)}, \tag{4}$$

where A_{ij}^k is the pixel in position (i,j) of $\mathcal{A}_l^k$, and Z is the number of pixels in $\mathcal{A}_l^k$. In order to make the feature representation reflect the detailed discrimination behavior of the model, we do not use ReLU rectified function to filter negative values in contrast to GradCam [32].

3.3. Contrastive Saliency Map Generation

Finally, following the Eq. 3, we obtain the feature representations for both original and counterfactual examples and generate the corresponding positive and negative saliency maps by comparing the representation differences. According to the aforementioned formulation (*i.e.*, Eq. 3), the feature representation for the original example with respect to class t can be expressed as $R_t\left(x_0\right)$. However, [46] pointed out that the implied noise in the original images would introduce class-independent effects to the gradients, which may lead to the deviation of the feature representation. Inspired by this work, we propose an aggregation version to reduce such effects by adding subtle Gaussian noises to the original examples, which can be expressed as follows:

$$\bar{R}_t\left(x_0\right) = \frac{1}{M} \sum_{m=1}^{M} R_t\left(x_0 + \varepsilon_m\right), \varepsilon_m \sim \mathcal{N}\left(0, \sigma^2\right), \tag{5}$$

where t indicates the target class, M indicates the number of random Gaussian noise ε_m applied to the input x_0 and σ is the deviation of the normal distribution.

Since we define the contrastive features as the differences between the original and counterfactual feature representations, we then decompose the differences relative to class t into promoted features (*i.e.*, M_{pro}) and suppressed features (*i.e.*, M_{sup}). M_{pro} denotes the features that are promoted to be dominant in the counterfactual example, while M_{sup} indicates the features that are suppressed or unchanged in the counterfactual examples, which are formulated as follows:

$$M_{\text{pro}} = \frac{1}{N} \sum_{n=1}^{N} \operatorname{ReLU}\left(R_t\left(x_c^n\right) - \bar{R}_t\left(x_0\right)\right), \tag{6}$$

$$M_{\text{sup}} = \frac{1}{N} \sum_{n=1}^{N} \operatorname{ReLU}\left(\bar{R}_t\left(x_0\right) - R_t\left(x_c^n\right)\right) \tag{7}$$

where x_c^n is the counterfactual example generated from x_0 with different initialization and N indicates the number of counterfactuals. After decomposing such features, we then scale the values of M_{pro} and M_{sup} into [0,1] by utilizing Min-Max normalization, and upsample them with bilinear interpolating to the same resolution of x_0 to generate contrastive saliency maps.

When answering the "Why p" question, the target class t should be replaced by p, and M_{sup} will denote the features that are the most relevant to class p, which will be used to construct the positive saliency map and M_{pro} for the negative saliency map, which will be used to construct the negative saliency map. For the "Why p, rather q" question, we calculate M_{pro} and M_{sup} with respect to class q. In this case, M_{pro} contains features that are introduced for class q, hence will be used to generate the negative saliency map and M_{sup} for the positive saliency map. In a word, the positive saliency map highlights important features with respect to class p and the negative saliency map represents features that benefit other classes (*i.e.*, not-P or Q).

4. Experiments

4.1. Experimental Setup

Datasets and Models: We conduct experiments on the commonly-used computer vision datasets, including the ImageNet-compatible dataset (containing 1k images used for NIPS 2017 adversarial competition) [1], the validation set of PASCAL VOC 2007 [10] (containing 5k images) and the split of validation set of MS COCO2017 [22] (containing 5k images). All images will be resized to 3×224×224 and normalized to [0,1] without performing other preprocessing methods. The proposed method will be evaluated on pre-trained VGG-19 [35] and ResNet-50 [13] which are from torchvision library [1].

Baseline Methods: To demonstrate the effectiveness of the proposed CCE, we compare it to diverse state-of-the-art saliency map interpretation methods, including propagation-based methods (Guided Backpropagation [36], Integrate-Grad [38]), perturbation-based methods (RISE [41], EP[11]), activation-based methods (GradCAM [32], GradCAM++ [6], ScoreCAM [43],I-GOS [27],FullG [37],CALM [18]) and counterfactual-based methods(PPE [2],CE [24],SCOUT [44]) .

Parameter Settings: For the counterfactual examples, we generate counterfactual examples with the perturbation threshold $\epsilon = 2.0$ and the number of iteration = 20 and set $\sigma = 0.05$ for the Gaussian noise for Eq. 5. Numbers for Gaussian noise(N) and counterfactuals(M) are both 5, and optimization iterations is set 30 for counterfactuals in the following experiments. When CCE interprets the "Why P" problem, the positive saliency map highlights the features most relevant and unique to the original class P, while the negative saliency map can still provide some additional information. For comparison with existing methods, we use M_{pro} and mix of two types of saliency maps M_p for class p, which is defined as $M_p = \alpha \cdot M_{\text{sup}} + (1 - \alpha) \cdot M_{\text{pro}}$ and α is

[1] https://pypi.org/project/torchvision/

Method	VOC 2007		ImageNet 2012		Coco 2017	
	Res.	VGG	Res.	VGG	Res.	VGG
GN	1.098	1.597	1.138	1.574	1.178	1.665
PGD [25]	0.613	0.630	0.673	0.740	0.649	0.727
PPE [2]	0.747	0.795	0.735	0.811	0.792	0.844
EP [11]	3.495	3.875	4.426	3.975	3.158	3.647
CCE	**5.559**	**5.178**	**5.959**	**5.394**	**5.671**	**5.334**

Table 1: Comparative evaluation in terms of Class Deviation Score which higher is better. The result shows our method performs significantly better than other methods.

set to 0.7 for following experiments. All parameters are set according to results of ablation experiments in Appendix.

4.2. Class Deviation Score

According to research [17], logits of non-target classes are changed unexpectedly when using perturbations or counterfactuals for interpretation. In this paper, we introduce a new evaluation metric Class Deviation Score (CDS) to quantify the degree of output predictions changes as well as the occurrence of class deviation. We divide the classes into target class p and non-target classes $\overset{\neg}{p}$ and Class Deviation Score is define as:

$$CDS = \frac{\mathcal{Z}_p\left(x_0\right) - \mathcal{Z}_p\left(x_c\right)}{1 + \mathbb{E}\left[D_{\mathrm{KL}}\left(\mathcal{Z}_{\overset{\neg}{p}}\left(x_0\right), \mathcal{Z}_{\overset{\neg}{p}}\left(x_c\right)\right)\right]} \tag{8}$$

The numerator term in Eq. 8 measures the degree of change for the probability of the target class. If a method succeeds in decreasing the network prediction for the target class, consequently, the distance between the original and perturbed predictions will be maximized. Concomitantly, the denominator of Eq. 8 adopt the Kullback-Leibler divergence metric[20] to assess the average distance of all non-target class distributions and penalizes the CDS when class deviation occurs. We randomly select 1000 images from three datasets to evaluate the mean CDS for Guassian Noise (σ=0.25), PGD attack and counterfactual-based methods. As shown in Tab. 1, we can observe that our CCE outperform other approaches. This is because our framework explicitly perturbs the target class while upholding output probabilities for non-target classes.

4.3. Sanity Check by Parameters Randomization

The intention of sanity check experiment [3] is to identify whether the explanation of the model decisions would change if the model's parameters are randomized. If the explanation remains unchanged, then the explanation and the model are not aligned. The cascade randomization experiment randomizes the model starting from the last layer and generates an explanation at each randomization step.

Method	ResNet-50			VGG-19		
	Ins. ↑	Del. ↓	Diff. ↑	Ins. ↑	Del. ↓	Diff. ↑
GBP [36]	49.71	**10.72**	38.99	40.12	10.71	29.41
IG [38]	37.38	11.56	25.83	36.58	9.46	27.12
RISE [41]	55.49	17.27	38.21	49.55	11.46	38.09
EP [11]	52.16	14.87	37.29	47.05	11.65	35.40
GradCAM [32]	56.43	18.87	37.56	44.87	14.65	30.21
GradCAM++ [6]	56.58	18.51	38.07	47.26	13.86	33.40
ScoreCAM [43]	57.09	18.56	38.53	49.45	14.85	34.61
I-GOS [27]	56.46	16.75	39.71	50.16	**9.45**	40.71
FullG [37]	50.30	12.89	37.42	52.01	14.19	37.82
CALM [18]	54.18	14.23	39.95	51.82	11.91	39.91
PPE [2]	52.84	13.27	39.57	49.38	10.12	39.26
CE [24]	57.18	17.92	39.26	48.39	12.57	35.82
SCOUT [44]	53.78	14.87	38.91	50.64	11.02	39.62
CCE(mixed)	**58.02**	17.96	**40.06**	**52.25**	11.27	**40.99**
CCE(positive)	57.47	18.14	39.33	51.86	11.79	40.07

Table 2: Insertion-Deletion Tests results. Higher insertion score (Ins.) are better and lower deletion score (Del.). The difference score (Diff. which higher is better) shows that CCE outperforms other related methods. The best records are marked in bold.

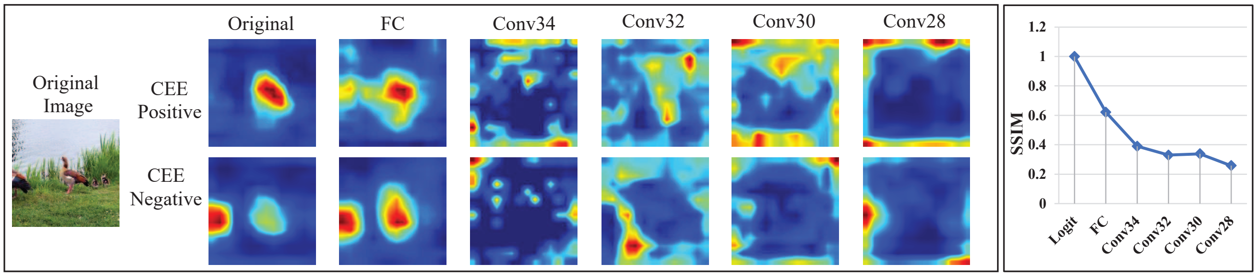
We use the ImageNet-compatible dataset to measure the mean similarities of visual explanations using the Structural Similarity Index Metric (SSIM) [45]. Specifically, we employ cascade randomization to compare the output of CCE on a pre-trained VGG19 model. The Fig. 3 shows the variation of SSIM changes when the model is randomized and the experimental results for one image, respectively.

We observe that the CCE is sensitive to the parameter randomization of the model, and the saliency maps change dramatically while the SSIM indicates the same results.

4.4. Insertion and Deletion Tests

We conduct the insertion and deletion tests following [31] to evaluate different saliency approaches using 1000 images from the ImageNet-compatible dataset [1]. The intuitive assumption behind the deletion metric is that removing the pixels/regions most relevant to a class will result in a significant drop in classification scores [31]. We gradually replace 3.6%($i.e.$ 224×8) pixels in the original image with a highly blurred version each time according to the values of the saliency map until no pixels are left. On the other hand, the insertion metric gradually reinserts the content of the original image starting with a blurred image, which produces images closer to the data manifold and has the advantage of mitigating the adversarial effects [41]. The insertion test replaces 3.6% pixels of the blurred image with the original one until the image is well recovered. We calculate the area under the curve (AUC) of the classification score after SoftMax as a quantitative indicator. Besides, we provide the overall score to comprehensively evaluate the deletion and insertion results, which can be calculated by $AUC(insertion) - AUC(deletion)$.

As the average results illustrated in Tab. 2, our method outperforms other methods for the insertion metric and



(a)

(b)

Figure 3: Sanity check results by cascade randomization. Both visual results (a) and SSIM (b) suggest that our explanations change significantly when the model parameters are randomised.

overall score, while is comparable to other methods for the deletion metric (with blur). The propagation-based approach produces more dispersed saliency maps so that important isolated pixels can be located more accurately in deletion experiments.

4.5. Computation Costs Analysis

We calculate the average running time on 1000 images in ImageNet-compatible dataset[1] for ResNet50 and VGG19 to quantify the time complexity in table 3. The best records are marked in bold. CCE takes five times as long as CE does, because CCE calculates 10 samples for each input, while the contrastive-based method CE[24] calculates two saliency maps for each input. The counterfactual-based PPE[2] takes twice the time of CCE, which has much lower computational efficiency than CCE. Please refer to the appendix for the visualization results of multiple objects.

Method	GBP	RISE	GCAM+	SCAM	FullG	CE	PPE	SCOUT	CCE
Time(s)	0.81	39.45	**0.09**	5.57	4.31	0.93	7.21	16.17	4.17

Table 3: Average result on ResNet50 and VGG19.

4.6. Qualitative Comparison

We qualitatively compare the saliency maps produced by recent SOTA methods. As shown in Fig. 4, the results generated by CCE are more precise than that of perturbation-based and activation-based methods. In addition, the saliency map from CCE is smoother and contains less noise compared with propagation-based methods.

4.6.1 Explanations for P-contrast Question

The visual answers to "Why P, rather than Q?" from different classes are provided in Fig. 5. Each row represents the explanations for two different Q-classes for one image. First, we show the original image and its respective Grad-CAM [32] explanation. For target class Q, we first visualize a representative image of it, and afterward show the positive and negative saliency maps corresponding to class Q. Note that the network is not trained or generates explanations based on these representative images, and the explanations are based on the network's conception of class P and class Q.

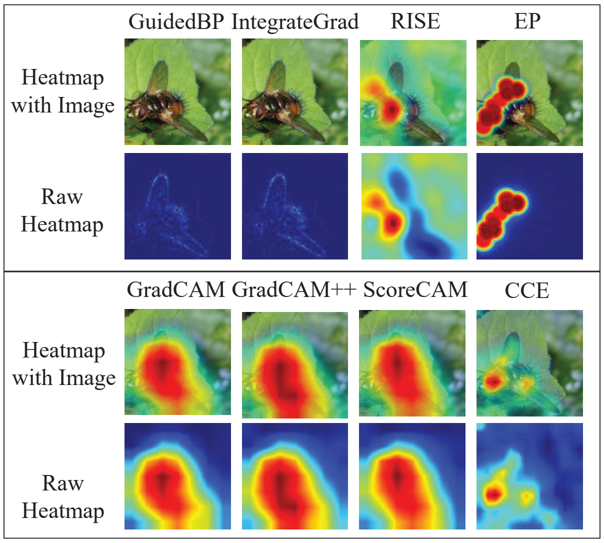


Figure 4: Qualitative Comparison. The heatmap from CCE contains less noise than propagation-based methods and is more compelling than perturbation-based and activation-based methods.

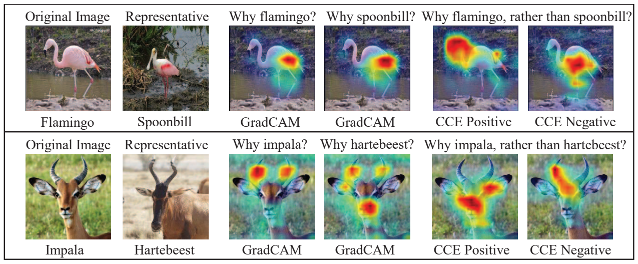


Figure 5: Counterfactual Contrastive explanations (CCE) for P-contrast Questions.

The counterfactual contrastive explanations provide a human interpretable insight into the decisions of neural networks. For instance, the first row of Fig. 5 are explanations for "Why flamingo, rather than spoonbill? ". The Grad-CAM explanations for flamingo and spoonbill are quite

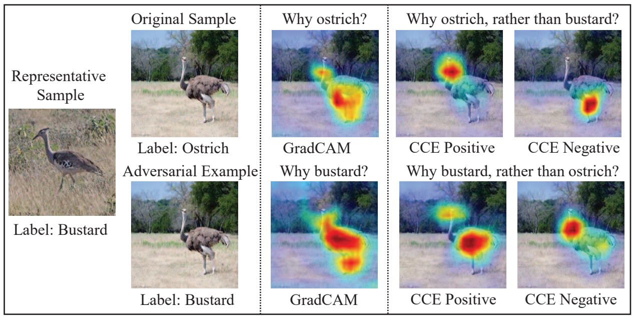
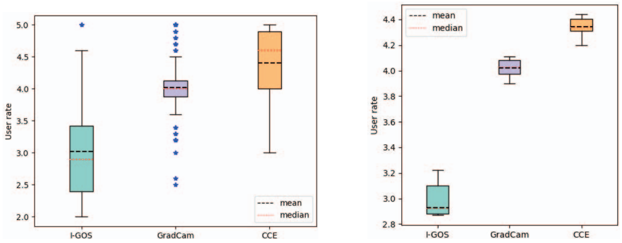


Figure 6: Counterfactual Contrastive explanations (CCE) for O-contrast Questions.



(a) Per-user Result.

(b) Per-picture Result.

Figure 7: User study results comparing CCE to I-GOS and GradCam. User rated from 1 to 5, the larger the better. Dashed red lines in the box plots denote mean values.

similar, offering little information. However, for CCE, the negative saliency map clearly points out the head regions are mostly different, while the positive saliency map emphasizes the similarity of bodies. The second row of Fig. 5 shows explanations for "Why impala, rather than hartebeest?". According to the hartebeest's representative image, the two species have similar horns but distinctly different faces. GradCAM explanations show few differences which are ambiguous to humans, while we provide distinguishable explanations. The negative saliency map indicates the suppressed features are concentrated on the face, while the positive saliency map which emphasizes the promoted features is concentrated on the area of the horns.

4.6.2 Explanations for O-contrast Question

In this section, we conduct the PGD attack on image x to generate adversarial example x' and form the O-contrast question "Why is image x labeled as class P, while image x' is labeled as class Q?". The visual answers to this question from GradCAM and CCE are provided in Fig 6.

The first column presents a representative image of a bustard, the same class as the adversarial example, and the second column shows the pristine image of an ostrich along with the adversarial example. The following columns present the GradCAM and contrastive explanations respectively. It is hard to capture the differences between two GradCAM explanations, for they both highlight the body regions. However, CCE is totally different for the two images. In the contrastive explanations for why the image x is not a bustard, the negative map highlights the neck while the positive map emphasizes the thighs. In contrast, the explanations for why the image x' is not an ostrich point out that features corresponding to the head and body are crucial for a bustard, while the neck is of less importance. Hence, contrastive explanations provide additional context and information that is not available from sole explanations.

4.7. User Study

User study has been used to evaluate explanations in many researches. For instance, GradCAM[32] conducted a user study to evaluate faithfulness and user trust on their saliency maps, and by showing explanations to the participants. We conducted a user study to evaluate the effectiveness of our proposed CCE visualization. For comparison, we choose two typical methods from gradient-based and CAM-based explanation. All images were randomly chosen.

We recruited 100 participants comprising of graduate and undergraduate students in engineering students at our university (65 males, 35 females, age: 18-30 years) and they were compensated based on the local-level minimum wage. They were first shown a tutorial informing them about the basic of image classification and saliency map explanations. Then they were directed to the task that involved answering 10 sets of questions of images from ImageNet. Each question set composed of two sections. In the first section, participants were shown the image with its possible classes, from which participants selected classes in which they guessed the image might be classified by the model. In the second section, the participants were shown the results of the model and then evaluated and ranked three types of saliency maps explaining the result from I-GOS, GradCAM and CCE, based on the degree to which the saliency map was able to convince them of the model's result. They were also asked to provide a confidence rating about how sure they were about their response. Participants rated the confidence in their answer or their understanding of the model on a 5-point Likert scale [21] (1: very poor, 2: poor, 3: fair, 4: good, 5: very good) allowing for degrees of opinion to be measured. At the end of all tests, participants directly evaluated the ability of the contrasting explanation compared to the single explanation, regarding the ability to increase the confidence of the model. Please refer to the appendix for the picture of user experiment interface.

For statistical analysis, we report the mean accuracy and standard deviation. Fig 7 shows the results comparing the metrics across the three methods. Fig 7(a) indicates that participants had the highest confidence in the model when they were provided with CCE explanations (Mean=4.343, SD=0.6837) than when they were provided with I-GOS

(Mean=2.927, SD=0.8359) or GradCAM (Mean=4.022, SD=0.4901) explanations. The differences between CCE and each of the two other methods are statistically significant ($p < 0.0001$ in Mann-Whitney U tests for both). In the final evaluation question, 92% of participants reported that the contrastive explanation improved the confidence of the model, with an average rating of 4.29.

5. Conclusion and Future Work

In this work, we proposed a unified framework *i.e.*, Counterfactual Contrastive Explanations (CCE), to contrastively explain three types of causal questions for neural networks: "Why P", P-contrast, and O-contrast questions. We introduce a content-aware counterfactual perturbing algorithm to generate contrastive examples, which is a reuse of original image without need for extra data in other methods. Furthermore, we present a positive-negative saliency scheme to provide visual contrastive explanations by comparing original and counterfactual examples. Our contrasting solution achieves a more detailed explanation for adversarial example. Both qualitative and quantitative experiments, including user study, show the outstanding performance of the proposed CCE explanation methods.

Although our work is a step towards solving causal problems, there are still some limitations in our work. Our framework can be easily applied to other saliency-based schemes, however, currently our implementation is an activation based method. A detailed study of extensibility of our framework to other explanation schemes is one of our future research direction.

Acknowledgments

This work was supported by Key R&D Program of Zhejiang (Grant No. 2022C01018), National Natural Science Foundation of China (Grants No. 62122066, U20A20182) and National Key R&D Program of China (Grant No. 2021ZD0112803).

References

[1] Nips17 adversarial attacks and defenses competition. Website, 2017. `https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition`.

[2] Elliott A. et al. Explaining classifiers using adversarial perturbations on the perceptual ball. In *CVPR*, 2021.

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[4] Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020.

[5] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2018.

[6] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision*, pages 839–847. IEEE, 2018.

[7] Ruoyu Chen, Jingzhi Li, Hua Zhang, Changchong Sheng, Li Liu, and Xiaochun Cao. Sim2word: Explaining similarity with representative attribute words via counterfactual explanations. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022.

[8] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

[9] Andrew Elliott, Stephen Law, and Chris Russell. Explaining classifiers using adversarial perturbations on the perceptual ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10693–10702, 2021.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019.

[12] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision*, pages 264–279, 2018.

[15] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4948–4957, 2019.

[16] Saeed Khorram and Li Fuxin. Cycle-consistent counterfactuals by latent transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10203–10212, 2022.

[17] Junho Kim, Seongyeop Kim, Seong Tae Kim, and Yong Man Ro. Robust perturbation for visual explanation: Cross-checking mask optimization to avoid class distortion. *IEEE Transactions on Image Processing*, 31:301–313, 2021.

[18] Jae Myung Kim et al. Keep calm and improve visual feature attribution. In *ICCV*, pages 8350–8360, 2021.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[20] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[21] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Prabhushankar M. et al. Contrastive explanations in neural networks. In *ICIP*. IEEE, 2020.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[26] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[27] Zhongang Qi, Saeed Khorram, and Fuxin Li. Visualizing deep networks by optimizing with integrated gradients. In *CVPR Workshops*, volume 2, 2019.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[30] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[31] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[33] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[37] Suraj Srinivas et al. Full-gradient representation for neural network visualization. *NeurIPS*, 32, 2019.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*, pages 1,2, 2014.

[40] Jeroen Van Bouwel and Erik Weber. Remote causes, bad explanations? *Journal for the Theory of Social Behaviour*, 32(4):437–449, 2002.

[41] Petsiuk Vitali, Das Abir, and Saenko Kate. Rise: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018*, page 151, 2018.

[42] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[43] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

[44] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8990, 2020.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[46] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7619–7628. IEEE, 2021.

[47] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1659–1676, 2020.