SAFE: Saliency-Aware Counterfactual Explanations for DNN-based Automated Driving Systems

Amir Samadi[1], Amir Shirian[1], Konstantinos Koufos[1], Kurt Debattista[1] and Mehrdad Dianati[1,2]

***Abstract*— The explainability of Deep Neural Networks (DNNs) has recently gained significant importance especially in safety-critical applications such as automated/autonomous vehicles, a.k.a. automated driving systems. CounterFactual (CF) explanations have emerged as a promising approach for interpreting the behaviour of black-box DNNs. A CF explainer identifies the minimum modifications in the input that would alter the model's output to its complement. In other words, it computes the minimum modifications required to cross the model's decision boundary. Current deep generative CF models often work with user-selected features rather than focusing on the discriminative features of the black-box model. Consequently, such CF examples may not necessarily lie near the decision boundary, thereby contradicting the definition of CFs. To address this issue, we propose in this paper a novel approach that leverages saliency maps to generate more informative CF explanations. Our approach guides a Generative Adversarial Network based on the most influential features of the input of the black-box model to produce CFs near the decision boundary. We evaluate the performance using a real-world dataset of driving scenes, BDD100k, and demonstrate its superiority over several baseline methods in terms of well-known CF metrics, including proximity, sparsity and validity. Our work contributes to the ongoing efforts to improve the interpretability of DNNs and provides a promising direction for generating more accurate and informative CF explanations. The source codes are available at: `https://github.com/Amir-Samadi//Saliency_Aware_CF`.**

I. Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in solving complex tasks, ranging from image recognition [1] to natural language processing [2] and many others. However, their black-box nature has impeded their widespread adoption in safety-critical applications, such as healthcare [3] and automated driving systems (ADS) [4], where interpretability and transparency are paramount. The advent of Interpretable Artificial Intelligence (IAI) has offered a potential solution to these challenges, and CounterFactual (CF) explanations have gained prominence as a promising IAI approach for revealing in human-understandable terms the underlying rationale behind the decisions of black-box AI systems. A CF example identifies the minimum required changes to the model's input that would alter the model output to its complement.

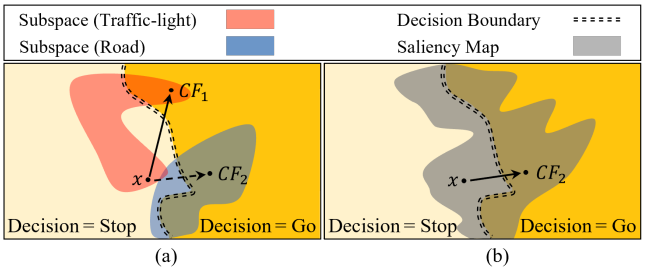

Fig. 1: SAFE explores a wider range of possible subspaces compared to previous methods. (a) Previous approaches are limited to predefined semantic subspaces (e.g., Traffic-light) resulting in sub-optimal solutions (e.g., CF_1). (b) While considering a wider subspace, SAFE finds the closest CF to the decision boundary based on the saliency maps coming from the black-box model, denoted as CF_2.

In recent years, a number of approaches have been proposed for generating CF explanations. Perturbation-based techniques modify the input data by adding or removing specific features [5], while gradient-based methods use the model gradients to identify and alter the most influential features of the input [6]. Despite their promising performance with low-dimensional tabular data, these methods suffer from various limitations. On the one hand, perturbation-based methods can be computationally expensive and often generate non-realistic or irrelevant CF instances [7], [8]. On the other hand, gradient-based techniques for CF generation may not be suitable for DNNs, particularly when processing high-dimensional image input data, resulting in adversarial examples [6], [9], [10].

To overcome the above shortcomings, deep generative models, such as Generative Adversarial Networks (GANs) [11], Variational Auto Encoders (VAEs) [12] and Diffusion Models [13] have emerged in the literature as promising solutions. Generative models can learn to map the distribution of (high-dimensional) input images between different domains. Transferring a query image from a source domain to another (target) domain is essentially the objective of the CF explanation. Therefore, deep generative models can produce diverse and realistic CF examples for high-dimensional input data [14]. However, for complex domains, e.g., ADS, generative models tend to generate implausible or semantically invalid examples [15]. One way to remedy this issue is to limit the potential solutions

[1]The authors are with the Warwick Manufacturing Group (WMG), The University of Warwick, Coventry CV4 7AL. (e-mail:amir.samadi, amir.shirian, konstantinos.koufos, k.debattista and m.dianati@warwick.ac.uk). This research is sponsored by Centre for Doctoral Training to Advance the Deployment of Future Mobility Technologies (CDT FMT) at the University of Warwick.

[2]The author is also with the School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University of Belfast (e-mail: m.dianati@qub.ac.uk).

by encoding the query images into a disentangled latent state and applying their sparse modification [15]–[17]. This approach involves the manual selection of the input features, which may lead to sub-optimal and biased explanations nonetheless. For example, the STEEX method developed in [15] generates CF examples by modifying groups of pixels specified in the segmented semantic layer. Although this can provide informative explanations, it actually deviates from the definition of CFs that seek minimal changes in the input. An alternative solution may exist within another semantic feature class that is closer to the decision boundary of the model, see Fig. 1(a) for an example illustration. This research gap in generating relevant and unbiased CF explanations near the decision boundary of the model serves as the key motivation for the present study.

To this end, instead of altering user-defined input features, we propose a novel approach that leverages saliency maps to explicitly guide the generative model to focus on the most influential features of the input and generate more effective and unbiased CF explanations. Saliency maps constitute a significant development in advancing the realm of IAI, as they provide insights into the regions of the input that receive the most attention from the black-box model during its decision-making process [18], [19]. That can enhance the quality of the generated CF explanations and achieve a more accurate estimation of the decision boundary, which is crucial for improving the interpretability of DNN models. The connection between the saliency map and the decision boundary has been explored in previous studies [20], [21] and an intuitive illustration of that can be found in Fig. 1(b).

The main contribution of this article lies in integrating saliency maps, extracted from pre-trained off-the-shelf models, into the GAN training process to better understand the discriminative features in the input space. The proposed deep generative CF explainer is hereafter referred to as Saliency-Aware counterFactual Explainer (SAFE) and the associated deep generative model as SAcleGAN. The key contributions of this article to the state-of-art are:

- Devising a framework for improving the interpretability of DNNs in complex real-world applications by introducing the use of saliency maps encompassing decision boundary information.
- Introduction of a novel term in the loss function that ensures the proper fusion of saliency information in the SAcleGAN.
- A comprehensive performance analysis of the SAcleGAN using a complex driving scene dataset, i.e., the BDD. The evaluation demonstrates the superiority of SAFE over several baseline methods in terms of validity, sparsity, and proximity of generated CF examples.
- A qualitative analysis of the results to gain insights into the strengths and limitations of SAFE. The results provide strong empirical evidence of the effectiveness and reliability of our approach.

The rest of this paper is organised as follows. In Section II, we briefly review the related work on IAI and generating CF explanations. In Section III, we describe how to leverage saliency maps to guide the generation of CF explanations. Section IV presents the evaluation results and discusses their implications. Finally, we conclude this work in Section V.

II. Background and Related work

This section provides an overview of the related work in the area of interpretability with a focus on CF explanations. More specifically, we review generative CF methods as they have shown to be more effective on real-world data.

A. Interpretability in Machine Learning

Interpretability has been a topic of growing interest in the machine-learning community over the past few years. Various methods can increase the interpretability of models, ranging from simple rule-based models and decision trees to more complex techniques such as sensitivity analysis [22] and deep learning models [23]. These methods become particularly important in safety-critical situations such as ADS, where interpretability is essential for reliability, regulatory compliance, and end-user trust [6].

Interpretable methods include both post-hoc and intrinsic models. Intrinsic methods refer to the models that involve designing and training DNN models with built-in interpretability and post-hoc methods employ auxiliary models to extract explanations from already trained models. Post-hoc explanations can be global or local, depending on the level of detail they provide. The former provides a holistic view of the main factors driving the decision-making process of the model, and the latter targets to understand the model's behaviour on a specific input [24]. Various methods, such as saliency maps, LIME [25], SHAP [26], partial dependence plots [27], and feature importance rankings [28], facilitate the generation of both local and global explanations, allowing users to gain insights into model decisions, identify biases or errors, and understand the model's behaviour on individual instances as well as across the entire dataset.

Despite the recent progresses in IAI research, there is still a strong need for better models. The CF explanations are a subset approach to post-hoc explanations and are gaining momentum due to their ability to provide human-understandable insights into the model's behaviour [6]. Recent works have focused not only on explaining the model decisions but also on generating CF examples, which are minimally modified versions of the input data that would change the model's output [6]. In the context of ADS, CF explanations can improve our understanding of why an automated driving model chose a particular action and what features of the input were the most crucial during decision-making [29]. For example, CF explanations can indicate whether the colour of a traffic light or the orientation of a pedestrian's body influenced the model's decision to stop or proceed. Conventional CF explanation techniques employ optimisation methods, such as gradient-based [6], genetic [30]–[32] and game-theory [33], [34] approaches and perform well with simple tabular data [6]. However,

they struggle to generate visually appealing outputs when dealing with high-dimensional image data. To address this issue, deep generative CF models have been introduced.

B. Deep Generative Counterfactual Models

Deep generative CF explanations are a promising approach to increasing the interpretability of complex decision-making systems, such as DNNs [15], [35], [36]. For instance, in an initial work [37], a trained classifier named as AttGAN provides labels for an image-to-image translation model [38]. This deep generative model then transfers the input image to the target domain serving as a CF explanation example. In a follow-up work [39], a starGAN [40] is employed to generate CF examples for a trained reinforcement learning-based model. The authors in [41] conducted a comparison of various image-to-image translation models for generating CF examples focusing on image quality metrics. The majority of works can explain classifiers working on simple images, such as face portraits or featuring a single and centered object. Nevertheless, generating CF explanations in complex environments, such as those encountered in ADS, can be challenging due to the high diversity and scattered distribution of images in the available driving datasets [15]. To this end, several works tend to limit the solution space for guiding the learning of generative models.

One way to restrict the solution space utilises an encoder-decoder pair in deep generative models, which effectively maps the input data distribution into a disentangled latent space [15], [16]. This learned feature space ensures that each dimension captures a specific attribute of the data independently of other dimensions, enabling control and understanding of the underlying features of the data that contribute to the generation of the CF example. Deep generative methods can provide minimal and meaningful CF examples by applying optimised manipulations into the latent space. However, such CF explanations are obtained from a user-selected feature space rather than being optimised within the entire input feature space. For instance, the STEEX model in [15] generates CF examples within the driving environment from a semantic layer selected by the end-user, such as traffic light segmentation, which might end up being suboptimal as the feature space is restricted. Unlike previous methods, in this study, we use saliency map features which lead the generative model to search near the black-box model's decision boundary, resulting in a global minimum solution.

In this study, avoiding intermediate transferring (to the disentangled latent state) enables our approach to extract more comprehensive features (unknown latent state) and decode them to the queried domain. Meanwhile, for limiting the potential solutions, we use a novel loss function that forces the deep generative model to apply changes merely in salient pixels. Considering that saliency maps encompass decision boundaries [20], [21], we have in practice limited the SAcleGAN to apply modifications near the decision boundary regions.

III. Saliency-Aware counterFctual Explainer

This section provides insights into the SAFE approach that leverages saliency maps to generate CFs using an unpaired dataset in the SAcleGAN. We formulate the problem of generating CFs for a given grey-box model in Section III-A, present an overview of the proposed approach in Section III-B and finally, describe the SAFE training phase in Section III-C.

A. Problem Definition

Let $x \in \mathcal{R}^{H \times W \times C}$ be an input instance with H, W and C being the height, width and channel. Also, let $M(\cdot)$ be our grey-box model (having access to some internal hyperparameters), classifying x as $y \in \mathcal{Z}^d$, where d is the number of classes in the grey-box model. Given a target class $y' \neq y$, we shall generate a CF $x' \in \mathcal{R}^{H \times W \times C}$ such that $M(x') = y'$ and x' is as similar as possible to x. To this end, the problem in this paper is formulated as leveraging saliency maps to guide the generation of x', i.e., the changes made to x in the search of x' are concentrated in the most salient regions yielding minimal and effective substitutions.

B. Overview of SAFE

In this section, we first discuss the construction of the saliency map, and then detail the procedure of CF generation using the SAcleGAN model, see Fig. 2. Previous studies utilised different GAN-based models to create CFs [15], [16], [37], [39], [41], but they proved inadequate in uncovering those associated with the minimal and effective changes. To address this, we use AttentionGAN [42] as backbone model and introduce a novel loss function term to ensure that the applied changes are minimal within an effective area. Following the conventional GAN architecture [43], AttentionGAN is composed of two competing modules, i.e., the generator $G(\cdot)$ and the discriminator $D(\cdot)$, iteratively trained to compete against each other in the manner of two-player minimax games.

1) Saliency Map Generation: Saliency maps are visualisations used to understand the behaviour of deep learning models in image classification tasks by highlighting the parts of an image used to determine the output class. Recently, gradient-based approaches have been used to compute saliency maps by measuring the impact of each pixel on the output. For an input image $x \in \mathcal{R}^{H \times W \times C}$ classified as y, the saliency map generator returns a 2D map $s = S(x, y) \in \mathcal{R}^{H \times W}$ representing the salient parts of the input image with respect to the model's output. In order to generate a saliency map, we employ the Gradient-weighted Class Activation Mapping (Grad-CAM) [44] method.

2) Generating CF Explanation: The saliency map along with the original image and the target label enables the generator to extract important features about the grey-box model and alter them to obtain the desired CF. Fig. 2 illustrates the SAcleGAN training phase, where for a query image x, the grey-box model $M(\cdot)$ provides a label y. Accessing the gradient signals of the model (hyperparameters), the Grad-CAM provides the saliency

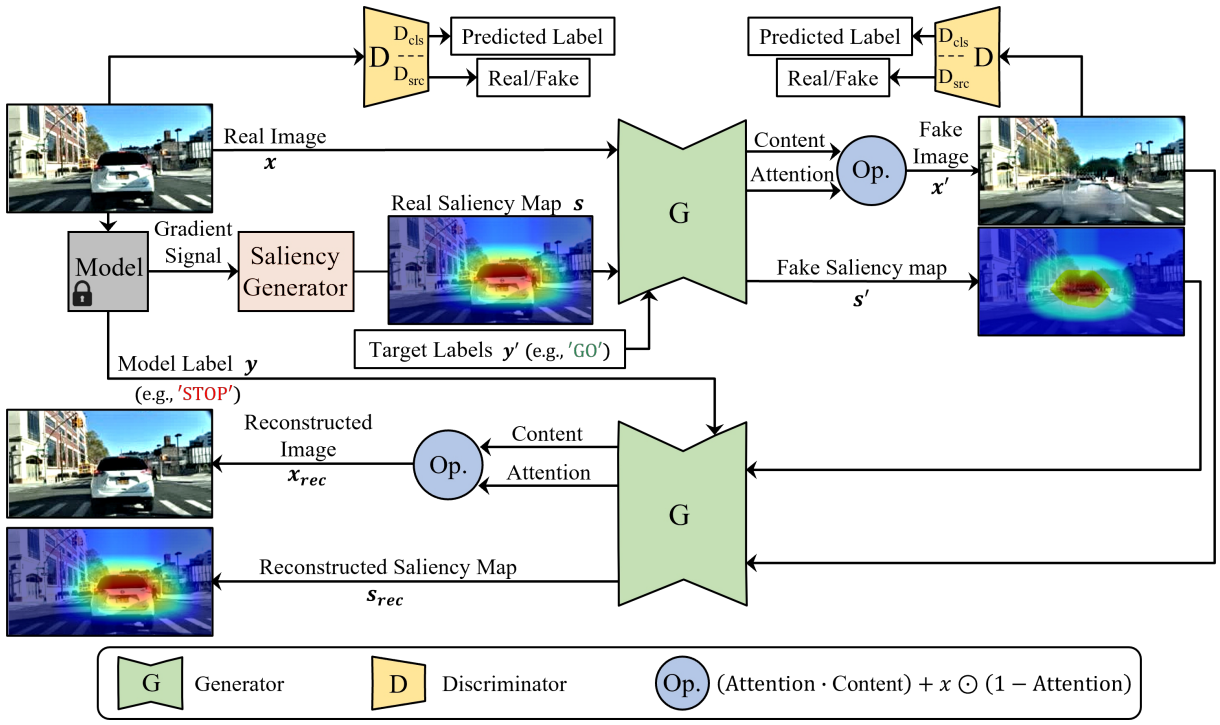

Fig. 2: Block diagram of the SAcleGAN model.

map s of the real image. The Generator network $G(\cdot)$ then transfers these images into the target domain/label y', i.e., $G(x, s, y') = (x', s')$, where x' and s' denote the CF example (fake image) and its predicted saliency map, respectively. Since paired ground-truth images are not available, the SAcleGAN transfers the provided images back to the initial domain to evaluate the generator's performance, i.e., $G(x', s', y) = (x_{rec}, s_{rec})$ where x_{rec}, s_{rec} are the reconstructed image and saliency map, respectively that are used to train the model. Meanwhile, the discriminator network $D(\cdot)$ learns to predict image labels, denoted by $D_{cls}(\cdot)$, and distinguish real images from fake ones, showed by $D_{src}(\cdot)$. In Fig. 2 one can also find attention and content layers, following the architecture of AttentionGAN in [45]. The attention matrix indicates the pixels that need to be varied in one channel, while the content matrices provide the (target) RGB values of the indicated pixels. The 'Op.' component in the figure implements the operation specified in the legend, resulting in the generation of the desired image.

C. Training phase

Let us assume that the real image x is labeled as $y \in Y$ during training, and the generator provides a fake image (CF example) x' that can be classified with the target label $y' \neq y$. The salience maps of the input x and the CF x' are s and s' respectively. The generator's objective is to deceive the discriminator D by generating images that appear to be real. Conversely, the discriminator D aims to enhance its ability to distinguish between generated samples and real data samples. This dynamic interplay encourages both the generator and discriminator to improve their respective performances. Thus, the adversarial loss function is designed in a way to encourage the aforementioned behaviours [39] :

$$\mathcal{L}_{adv}^{D} = \mathbb{E}_x\Big[D_{src}(x)\Big] - \mathbb{E}_{x'}\Big[D_{src}(x')\Big] + \lambda_{gp}\,\mathbb{E}_{\hat{x}}\Big[(\|\nabla D_{src}(\hat{x})\|_2 - 1)^2\Big]. \tag{1}$$

$$\hat{x} = \alpha x + (1-\alpha)x', \tag{2}$$

where $\lambda_{gp} > 0$ is a regularisation term, α is a uniform random variable in $(0, 1)$, $\|\cdot\|_2$ the Euclidean norm of a vector, $\nabla D_{src}(\hat{x})$ is the gradient of D_{src} evaluated at $\hat{x}$ and $\hat{x}$ is a linear combination of x and x'.

While the first two terms in Eq. (1) guide the discriminator to distinguish real images from fake ones, the gradient penalty loss term

$$\mathcal{L}_{gp} = \mathbb{E}_{\hat{x}}\Big[(\|\nabla D_{src}(\hat{x})\|_2 - 1)^2\Big] \tag{3}$$

prevents the discriminator from being trapped near trivial solutions [46]. It ensures that the discriminator's gradients are bounded for better convergence calculated by choosing a random point α along the direction connecting a real image sample and a generated one, see Eq. (2), and then taking the gradient at that point. Intuitively, minimising the loss $\mathcal{L}_{gp}$ for a random point between real and fake images, ensures that the discriminator has smooth gradients along the data manifold, preventing it from ignoring or focusing too much on individual samples and improving the overall stability of the GAN training process.

To evaluate whether the CF examples can be classified as y' by the grey-box model M, the discriminator should learn the mapping of images to corresponding labels. This can be achieved through the following loss function:

$$\mathcal{L}_{cls}^{y} = \mathbb{E}_{x,y}\Big[(-\log D_{cls}(y|x)\Big]. \tag{4}$$

A linear combination of the two loss functions is used to train the discriminator network:

$$\mathcal{L}_D = -\mathcal{L}_{adv}^D + \lambda_{cls}\mathcal{L}_{cls}^y. \tag{5}$$

To further encourage the generator, the following loss functions are utilised to generate real-like images. To achieve this objective, the generator is penalised if it fails to deceive the discriminator in differentiating real images from generated ones and mis-classifies generated images into the target labels.

$$\begin{aligned} \mathcal{L}_{adv}^G &= \mathbb{E}_{x'}\Big[D_{src}(x')\Big]. \\ \mathcal{L}_{cls}^{y'} &= \mathbb{E}_{x',y'}\Big[-\log D_{cls}(y'|x')\Big]. \end{aligned} \tag{6}$$

Moreover, since the ground truth of CF examples is unavailable, following the SAcleGAN approach, we transfer back the generated CF example (fake image) into the initial domain and compare it with the query image. The following loss penalises the generator model for adding excessive artifacts to enforce forward and backward consistency:

$$\mathcal{L}_{rec} = \mathbb{E}_{x,y'}\Big[\|(x,s) - G(G(x,s,y'),y)\|_1\Big], \tag{7}$$

where the vector norm can be calculated after the pairs of images and saliency maps are concatenated into a column vector.

To incorporate the saliency maps into the learning phase, we propose a novel term, $\mathcal{L}_{fuse}$, into the loss function. This term ensures that the saliency map information is fused into the generator model properly. Penalising the generator for modifying the non-salient features, $(1-s)$, leads the model to perceive the salient region and alter only that region's pixels towards the generation of the CF example:

$$\mathcal{L}_{fuse} = \mathbb{E}_{x,y,y'}\Big[||(x-x') \odot (1-s)||_1\Big], \tag{8}$$

where $\odot$ denotes element-wise multiplication of matrices.

Finally, a linear combination of the three introduced loss functions shapes the generator model's objective $\mathcal{L}_{\text{G}}$:

$$\mathcal{L}_G = -\mathcal{L}_{adv}^G + \lambda_{cls}\mathcal{L}_{cls}^{y'} + \lambda_{fuse}\mathcal{L}_{fuse}. \tag{9}$$

The pseudocode under Algorithm 1 summarises the SAFE approach generating the CF examples.

Algorithm 1: Pseudo-code of SAFE

```
/*          Training Procedure          /*
Data: Input image x labeled as y, target label y'
Result: CF instance x'
N ← 0
for x in data do
    l_cls ← L_cls(D_cls(x), M(x));
    l_real ← L_adv^D(D_src(x));
    s ← Grad-cam(x, y, y');
    (x', s') ← G(x, s, y');
    l_fake ← L_adv^D(D_src(x'));
    α ← rand(0, 1);
    x̂ ← αx + (1 − α)x';
    l_gp ← L_gp(D_src(x̂));
    l_D ← −l_real + l_fake + λ_cls l_cls − λ_gp l_gp;
    D ← ADAM(D, l_D, learning-rate);
    N ← N + 1;
    if N is 5 then
        (x', s') ← G(x, s, y');
        l_fake ← D_src(x');
        l_cls ← L_cls(D_cls(x'), M(x'));
        x_rec, s_rec ← G(x', s', M(x));
        l_rec ← L_rec(x, s, x_rec, s_rec);
        l_fuse ← L_fuse(x, x', s);
        l_G ← −l_fake + λ_cls l_cls + λ_rec l_rec + λ_fuse l_fuse;
        G ← ADAM(G, l_G, learning-rate);
        N ← 0
    end
end
```

IV. Evaluations

In this section, we carry out an analysis of the effectiveness of the SAFE model on the BDD dataset, and compare its performance against state-of-the-art methods [15], [16], [37], [39], [41]. The implementation details of the SAcleGAN are presented in Section IV-A. We employ various metrics to measure the quality of the generated CF explanations, which are presented in Section IV-B. After that we proceed with the description of the selected dataset in Section IV-C and the discussion of the performance evaluation results in Section IV-D. Finally, we conduct in Section IV-E a qualitative study that illustrates how well the generated CFs convey explanations comprehensible to humans.

A. Implementation Details

All input images are resized to 128×256 pixel resolution. To ensure the robustness of our findings, we repeat our experiments five times with different seeds and report the mean values of the performance metrics. For all network models we use the Adam optimiser with a learning rate of 10^{-4}. The network hyper-parameters and loss coefficients are explored heuristically and set to $\lambda_{cls} = 1, \lambda_{gp} = 10, \lambda_{rec} = 10, \lambda_{fuse} = 1$ and $\lambda_{fuse} = 5$ (see Eq. 5 and 9) for all experiments. The SAcleGAN employs the limited modified generator network and the pure discriminator network of attentionGAN [42] models, and for the subject grey-box model we use ResNet50, see Fig. 2. Furthermore, The 'Saliency Generator' component of our approach utilises the Grad-CAM method [18]. To execute our implementation, we employ Pytorch framework on an NVIDIA RTX-3090 GPU.

B. Performance Metrics

In order to assess the quality of the generated CF examples, we measure their adherence to well-known CF criteria namely **Proximity**, **Sparsity**, and **Validity**. Proximity refers to the similarity or closeness between the query image and its corresponding CF instance, which can be calculated by measuring the mean value of pixels that are modified. Sparsity refers to the extent to which the changes to the

generated CFs are minimal and focus only on a small subset of features. Validity measures the success rate of the method generating CFs as being equal to the percentage of generated CF examples that altered the model's output to the target label.

Apart from the traditional performance metrics, one can find in the literature *generative* type of metrics that assess the visual realism of the generated CF explanations, such as the **FID**, **LPIPS**, **KID**, and **IS**. FID measures the similarity between generated and real images based on the statistical properties of their feature representations. LPIPS quantifies perceptual differences between images using high-level visual features. KID measures the dissimilarity between feature distributions of generated and real images using the Maximum Mean Discrepancy method, and, IS evaluates the quality and diversity of generated images by comparing them to the real-world distribution of images.

C. Berkeley DeepDrive Dataset (BDD)

The BDD100k dataset consists of 100,000 detailed images depicting diverse driving scenes [47]. The grey-box DNN model used in our experiments is trained to predict actions such as "Move Forward" and "Stop/Slow down" on the BDD-OIA dataset [48], which consists of 20,000 scenes specifically selected and annotated with high-level actions from the BDD100k dataset. To generate CF examples, both the SAFE and baseline explainer models are fed with the BDD100k dataset.

D. Performance Evaluation

Table I contains the comprehensive performance comparison results between the proposed model, SAFE, and two well-known image-to-image translation models, namely starGAN [40] and attentionGAN [45]. For the SAFE model, we investigate the effectiveness of the newly introduced loss function, $\mathcal{L}_{\text{fuse}}$, in Eq. (8) for two values of the associated coefficient $\lambda_{fuse} = 1$ and $\lambda_{fuse} = 5$, indicated by SAFE_1 and SAFE_5 respectively. To ensure a fair comparison with previous studies [37], [39], [41], the baseline models are fed with the decision model's output instead of the ground-truth labels. This approach allows for improvements in the proximity, sparsity, and validity metrics of the baseline models. By examining the Table I, we observe that SAFE outperforms the baselines in terms of validity substantially, while exhibiting negligible differences in the generative metrics that assess the realism of the generated images (FID, LPIPS, KID, IS). Notably, the effect of the term $\mathcal{L}_{\text{fuse}}$ in the loss function is evident, where for the lower value of λ_{fuse}, SAFE applies more modifications to the query image, resulting in higher validity and greater distance from the initial image. Conversely, a higher value of λ_{fuse} leads to a different trade-off where modifications are reduced, resulting in lower validity and a closer resemblance to the initial image.

Table II presents the performance comparison between SAFE and the latest state-of-the-art CF explainers, namely STEEX [15] and OCTET [16] based on the results reported in their respective papers. Upon examining this table, we

TABLE I: Performance comparison of SAFE with image-to-image translation models. The direction of arrows preceding the metric indicates which values are desirable (low ↓, or high ↑). The best model is highlighted in bold, while the second-best model is underlined. Proximity, sparsity, and validity are denoted by prx, sprs, and vld, respectively. Note that *CF-* represents the counterfactual variant of the model.

Method	↓FID	↓LPIPS	↓KID	↑IS	↓Prx	↓Sprs	↑Vld
CF-StarGAN	**19.31**	**0.117**	0.067	3.151	0.039	0.999	0.564
CF-AttentionGAN	<u>28.90</u>	<u>0.196</u>	0.072	<u>3.120</u>	**0.009**	0.574	0.551
SAFE_1	28.45	0.204	<u>0.064</u>	3.081	<u>0.021</u>	<u>0.413</u>	**0.938**
SAFE_5	29.21	0.225	**0.059**	**3.242**	0.026	**0.409**	<u>0.914</u>

TABLE II: Performance comparison of SAFE with CF explainers. The direction of the arrow preceding the metric indicates which values are desirable (low ↓, or high ↑). The best performing model is indicated in bold. The validity metric is denoted by vld.

Method	↓ FID	↓LPIPS	↑Vld
STEEX	61.35	0.451	**0.973**
OCTET	54.95	0.422	0.961
SAFE_1	**28.45**	**0.204**	0.938

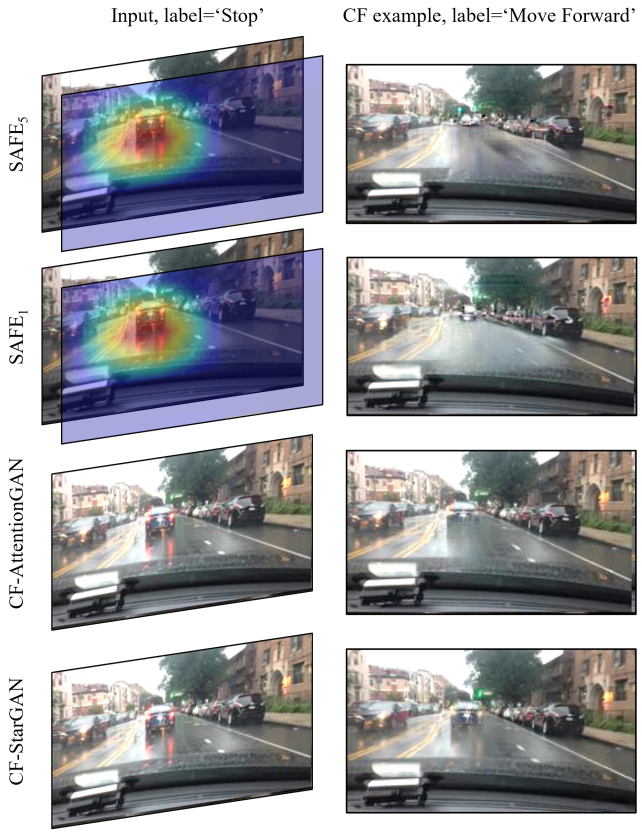

Fig. 3: Performance comparison by visual inspection between the SAFE and two image-to-image translation baseline models.

Fig. 4: Performance comparison by visual inspection between the SAFE and the AttentionGAN baseline model. The target label is 'GO' for the top row and 'STOP' for the bottom row.

do not observe any significant difference in terms of the explainer model's validity performance, however, when it comes to the realism metrics, our approach demonstrates notable improvements. Specifically, it achieves a 53 % improvement in LPIPS and a 55 % improvement in FID metrics as compared to the state-of-the-art CF explainers STEEX and OCTET. These results highlight the superior quality and realism of the CF examples generated by the SAFE model. It is worth noting that both baseline methods, STEEX and OCTET, modify disentangled latent states using user-selected features. This approach may result in a counterfactual example that is far from the decision boundary of the grey-box DNN model, as discussed in sections I and II.

E. Quality of Counterfactual Explanations

Fig. 3 illustrates a visual comparison for the quality of the generated CF examples between SAFE and the baseline image-to-image translation models. Given an input labelled as 'Stop' by the decision model (Fig. 3, first column), the SAFE model leverages a saliency map (depicted as an image overlay) and fades the front vehicle completely to change the decision to 'Go' (Fig. 3, second column). On the contrary, the baseline models encounter difficulties in achieving such clear modifications in the original image and fail to generate CF examples with similar effectiveness. This visual comparison highlights the superior performance of SAFE in generating meaningful, clear and effective CF explanations as compared to the baseline models.

Fig. 4 (top row) illustrates another driving scene showcasing the generation of a CF example that alters the original decision from 'STOP' to 'GO'. The visualisation reveals that SAFE has faded both the pedestrian and the yellow participant vehicle, and moreover, it alters the street sign to a green traffic light. Based on these modifications, we can infer that, under these circumstances, the automated vehicle should proceed forward, taking into account factors such as an obstacle-free urban intersection and the presence of a green traffic light signal. The bottom row of Fig. 4 presents a different scene in a motorway, wherein the grey-box decision model would only stop if there is a leading vehicle nearby. To summarise, these CF examples effectively highlight the necessary modifications required in the input image to change the automated vehicle's decision, thereby indicating that the DNN model has learned to focus on relevant features during training. This is also evident from the distribution of changed pixels in Fig. 4, where one can see that the SAFE model has successfully concentrated on the important features within the input image, which can be attributed to the utilisation of the saliency maps. For both driving scenarios we have also generated the CF examples and the changed input pixels by the AttentionGAN [45] model. The latter did not succeed in generating plausible CFs as can be also reflected by the scattered distribution of the changed pixels across the image.

V. Conclusions

This paper designed a novel method named after SAFE for generating counterfactual (CF) explanations leveraging a cycle-GAN guided by saliency maps. The latter was adopted to highlight the most salient regions of the input image that play a decisive role in determining the output of the machine learning model and subsequently help alter only those regions given the target (complement) label. The proposed method significantly advanced the state-of-art models in terms of validity (the higher, the better) and sparsity (the lower, the better) of the generated CF examples using the Berkeley DeepDrive dataset. At the same time, the generated CFs were more interpretable and clear to understand by visual inspection. We believe that this method has the potential to strengthen our trust and facilitate the adoption of machine learning models in real-world applications, such as autonomous vehicles and automated driving systems in general, by providing more transparent and interpretable explanations for their predictions and decision-making. In the future, it is worth validating the performance of SAFE with other datasets too.

References

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.
[2] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.
[3] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *TNNLS*, vol. 29, no. 6, pp. 2063–2079, 2018.

[4] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods."

[5] M. Ivanovs, R. Kadikis, and K. Ozols, "Perturbation-based methods for explaining deep neural networks: A survey," *Pattern Recognition Letters*, vol. 150, pp. 228–234, 2021.

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

[7] F. Yang, N. Liu, M. Du, and X. Hu, "Generative counterfactuals for neural networks via attribute-informed perturbation," *ACM SIGKDD Explorations Newsletter*, vol. 23, no. 1, pp. 59–68, 2021.

[8] A. Samadi, K. Koufos, and M. Dianati, "Counterfactual explainer framework for deep reinforcement learning models using policy distillation," *arXiv preprint arXiv:2305.16532*, 2023.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[12] H. Kim, S. Shin, J. Jang, K. Song, W. Joo, W. Kang, and I.-C. Moon, "Counterfactual fairness with disentangled causal effect variational autoencoder," in *AAAI*, vol. 35, no. 9, 2021, pp. 8128–8136.

[13] G. Jeanneret, L. Simon, and F. Jurie, "Diffusion models for counterfactual explanations," in *ACCV*, 2022, pp. 858–876.

[14] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, and J. Jorge, "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications," *Information Fusion*, vol. 81, pp. 59–83, 2022.

[15] P. Jacob, É. Zablocki, H. Ben-Younes, M. Chen, P. Pérez, and M. Cord, "Steex: steering counterfactual explanations with semantics," in *ECCV*. Springer, 2022, pp. 387–403.

[16] M. Zemni, M. Chen, É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Octet: Object-aware counterfactual explanations," *arXiv preprint arXiv:2211.12380*, 2022.

[17] P. Rodríguez, M. Caccia, A. Lacoste, L. Zamparo, I. Laradji, L. Charlin, and D. Vazquez, "Beyond trivial counterfactual explanations with diverse valuable explanations," in *ICCV*, 2021, pp. 1056–1065.

[18] 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," in *ICCV*, 2017, pp. 618–626.

[19] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *CVPRW*, 2020, pp. 24–25.

[20] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb, "On the connection between adversarial robustness and saliency map interpretability," *arXiv preprint arXiv:1905.04172*, 2019.

[21] P. Mangla, V. Singh, and V. N. Balasubramanian, "On saliency maps and adversarial robustness," in *ECML PKDD*. Springer, 2021, pp. 272–288.

[22] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *NeurIPS*, vol. 28, no. 11, pp. 2660–2673, 2016.

[23] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding human-to-vehicle advice for self-driving vehicles," in *CVPR*, 2019, pp. 10 591–10 599.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *SIGKDD*, 2016, pp. 1135–1144.

[25] M. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *AAAI*, vol. 32, no. 1, 2018.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *NeurIPS*, vol. 30, 2017.

[27] E. R. Ziegel, "The elements of statistical learning," 2003.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[29] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *ICCV*, 2017, pp. 2942–2950.

[30] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," *arXiv preprint arXiv:1805.10820*, 2018.

[31] S. Sharma, J. Henderson, and J. Ghosh, "Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," *arXiv preprint arXiv:1905.07857*, 2019.

[32] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2020, pp. 448–469.

[33] Y. Ramon, D. Martens, F. Provost, and T. Evgeniou, "A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, lime-c and shap-c," *Advances in Data Analysis and Classification*, vol. 14, no. 4, pp. 801–819, 2020.

[34] S. Rathi, "Generating counterfactual and contrastive explanations using shap," *arXiv preprint arXiv:1906.09293*, 2019.

[35] M. Augustin, V. Boreiko, F. Croce, and M. Hein, "Diffusion visual counterfactual explanations," in *NeurIPS*, 2022.

[36] S. Khorram and L. Fuxin, "Cycle-consistent counterfactuals by latent transformations," in *CVPR*, 2022, pp. 10 203–10 212.

[37] S. Liu, B. Kailkhura, D. Loveland, and Y. Han, "Generative counterfactual introspection for explainable deep learning," in *GlobalSIP*. IEEE, 2019, pp. 1–5.

[38] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE transactions on image processing*, vol. 28, no. 11, pp. 5464–5478, 2019.

[39] T. Huber, M. Demmler, S. Mertes, M. L. Olson, and E. André, "Ganterfactual-rl: Understanding reinforcement learning agents' strategies through visual counterfactual explanations," *arXiv preprint arXiv:2302.12689*, 2023.

[40] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018, pp. 8789–8797.

[41] I. Kothiyal, A. Patil, V. Horta, and A. Mileo, "Utilization of gan for automatic evaluation of counterfactuals: Challenges and opportunities," *Digital Book of Abstracts*.

[42] H. Tang, D. Xu, N. Sebe, and Y. Yan, "Attention-guided generative adversarial networks for unsupervised image-to-image translation," in *IJCNN*. IEEE, 2019, pp. 1–8.

[43] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *CoRR*, vol. abs/1406.2661, 2014.

[44] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *CVPRW*. Computer Vision Foundation / IEEE, 2020, pp. 111–119.

[45] H. Tang, H. Liu, D. Xu, P. H. Torr, and N. Sebe, "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks," *TNNLS*, 2021.

[46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[47] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020, pp. 2636–2645.

[48] Y. Xu, X. Yang, L. Gong, H.-C. Lin, T.-Y. Wu, Y. Li, and N. Vasconcelos, "Explainable object-induced action decision for autonomous vehicles," in *CVPR*, 2020, pp. 9523–9532.