On Quantitative Evaluations of Counterfactuals

Frederik Hvilshøj
Computer Science
Aarhus University
Aarhus, Denmark
fhvilshoj@cs.au.dk

Alexandros Iosifidis
Electrical and Computer Engineering
Aarhus University
Aarhus, Denmark
ai@ece.au.dk

Ira Assent
Computer Science
Aarhus University
Aarhus, Denmark
ira@cs.au.dk

Abstract

As counterfactual examples become increasingly popular for explaining decisions of deep learning models, it is essential to understand what properties quantitative evaluation metrics *do* capture and equally important what they *do not* capture. Currently, such understanding is lacking, potentially slowing down scientific progress. In this paper, we consolidate the work on evaluating visual counterfactual examples through an analysis and experiments. We find that while most metrics behave as intended for sufficiently simple datasets, some fail to tell the difference between good and bad counterfactuals when the complexity increases. We observe experimentally that metrics give good scores to tiny adversarial-like changes, wrongly identifying such changes as superior counterfactual examples. To mitigate this issue, we propose two new metrics, the Label Variation Score and the oracle score, which are both less vulnerable to such tiny changes. We conclude that a proper quantitative evaluation of visual counterfactual examples should combine metrics to ensure that all aspects of good counterfactuals are quantified.

1 Introduction

With the increased popularity of machine learning applications, a need for understanding machine learning models arises. Many methods have been proposed to explain predictions of machine learning models. To name but a few, some are based on heatmaps that identify salient input features [22, 3, 18, 4], others rely on transparent surrogate models [8, 1], and some produce counterfactual examples [30, 6, 25]. In this work, we consider the latter group, focusing on the image domain.

Counterfactual examples identify specific changes to inputs, such that the predicted outcome of a machine learning model changes. Such examples allow interactions with the model to gain insights into its behavior. For example, surveillance images of candidates picked out for screening can be assessed for biases by identifying features to change for the system to ignore the candidates [7].

Work on counterfactual explanations has become increasingly popular [26]. For images, counterfactuals should convey realistic changes to the input that are minimal and necessary while being valid, sparse, and proximal [19, 5, 16, 23]. Various metrics have been proposed to quantify aspects of the quality of counterfactual examples. However, most metrics are used in isolation to evaluate a method proposed in the corresponding paper and to compare the method to others that are evaluated on different metrics. Furthermore, there exists little or no research on what properties the different metrics actually capture. Lacking standard metrics and knowledge about what metrics capture makes it difficult to compare methods, which potentially slows down scientific progress within the field.

In this work, we analyze and evaluate existing metrics to understand what each metric expresses in terms of realistic and minimal changes. Through experiments, we find that most metrics have the intended behavior for image datasets of lower complexity. For a more complex dataset, our experiments show how multiple existing metrics fail to distinguish between good and bad counterfactuals. We also expose vulnerabilities of different metrics and propose to account for such vulnerabilities

by reporting multiple scores in combination. Counterfactual examples comprising tiny unrealistic changes are found to often yield unintended good scores. To mitigate this issue, we propose two new metrics that are less susceptible to tiny changes and align well with qualitative evaluations. We argue that when presenting a proper evaluation of a counterfactual method, the evaluation would need a metric for quantifying how realistic counterfactuals are, *e.g.*, the Fréchet Inception Distance, and a metric like our Latent Variable Score to assert that the validity of the counterfactuals generalize. Upon publication, we will also publish an evaluation framework for easy comparisons of methods.

2 Counterfactual Examples

The counterfactual question seeks to find *necessary* and *minimal* changes to an input to obtain an alternative outcome from a classifier [30]. An answer often comes in the form "Had values $v_1, ..., v_m$ been $\hat{v}_1, ..., \hat{v}_m$ and all other values remained the same, then outcome Y would have been Z." For images, an answer would be a new image similar to the input but with specific features changed.

Naturally, counterfactual examples come in various forms and might not convey the information that we would expect. For example, adversarial attacks, which adds imperceptible noise to inputs in order to change predictions [27], are of little or no relevance in terms of interpretability. As such, multiple additional criteria have been proposed that counterfactuals need to possess to be intuitive for humans.

Realistic changes. To be useful for humans, counterfactual examples should look realistic [5, 24]. The criterion has also been described as counterfactuals being likely to stem from the same data distribution as the training data [21, 6]. In the image domain, realistic changes can be hard to quantify. How do you, for example, distinguish an adversarial attack from a proper counterfactual, when the attack may be closer to the input in terms of, *e.g.*, Euclidean distance? Methods for quantifying how realistic counterfactuals are typically rely on a form of connectedness [17, 20] or on embedding spaces of deep learning models [16, 9]. Although being important for consolidating the field of counterfactual explanations, we find experimentally that metrics have unintended behaviors when quantifying how realistic tiny adversarial-like changes are.

Minimal changes. For counterfactual examples to be more useful to humans, input features need to change minimally for the prediction to change [30]. Associated properties are sparsity and proximity, which relates to changing only few features and changing features such that the counterfactual stays in the proximity of the input [19]. When only few features are changed, the counterfactuals are said to be more interpretable [30]. In high-dimensional domains like the image domain, quantifying minimal changes with, *e.g.*, Euclidean distance may yield undesired results. For example, we demonstrate with an experiment that tiny adversarial-like changes can be deemed better (smaller) compared to realistic changes that naturally need to change more pixels. In turn, a method that performs well only on minimal changes may be producing unrealistic adversarial-like counterfactuals that are of little or no value. For a good performance on minimal changes to be meaningful, a method must thus also perform well on metrics quantifying how realistic the counterfactuals look.

Additional properties. In an interactive setting, computational efficiency is important [29, 16, 10]. If computations are too slow, interactions with a system will be poor. Although computation time is important, we do not study it here, as it does not quantify the quality of the counterfactuals. It is also important that humans can use the generated counterfactuals. Consequently, multiple works have done human studies of their methods [7, 5, 25]. Such tests are typically domain specific and thus prohibit a generalized test. Therefor, we do not include them for further evaluation here.

3 Evaluating Counterfactuals Quantitatively

In this section, we present those quantitative metrics which have been applied to images in at least two publications and analyze their applicability in terms of how they measure changes. To mitigate an observed issue with tiny adversarial-like changes, we additionally propose two new metrics. We find that each metric reflects specific aspects of counterfactual quality and need to be reported in combination with other metrics to avoid isolated drawbacks of the metrics.

3.1 Existing Metrics

Simple distance metrics. A natural first approach to measuring changes between inputs and counterfactuals is to use metrics like L_1 and L_2-norms even though such norms are known to work poorly on high-dimensional data like images [11]. We include those metrics because they are present in objective functions for gradient based counterfactual methods [30, 5, 16] and are in turn a natural first choice for measuring minimal changes. In our experiments, we include a hybrid metric denoted the elastic net distance (EN), which is defined as $EN(x, c) = \|x - c\|_1 + \|x - c\|_2$, where x is the input and c is the counterfactual. We use this metric because it combines the L_1 and the L_2-norm.

Target-class validity. The Target Class Validity (TCV) [17] quantifies the percentage of the generated counterfactuals that are predicted to be of the target class by the classifier under consideration:

$$\text{TCV} = \frac{1}{|X|} \sum_{x \in X} 1_{[f(x) \neq f(cf(x))]}. \tag{1}$$

In Equation (1), $1_{[.]}$ is the indicator function, X is the test set, f is the predictive function, and $cf(\cdot)$ is the function that generates the counterfactual examples.

The score quantifies how effective a method is in creating counterfactual examples that successfully change the class. It does not quantify the quality of the counterfactuals in terms of neither minimal nor realistic changes. In turn, it should be reported along with other metrics quantifying those properties.

IM1. [16] introduces the IM1 score, which employs auto-encoders to approximate how well counterfactual examples follow the training data distribution. The score shows the ratio between how well the counterfactual example c of target class q can be reconstructed by an auto-encoder trained on data from the target class AE_q and an auto-encoder AE_p trained on the data of the input class p:

$$\text{IM1}\,(c) = \frac{\|c - \text{AE}_q(c)\|_2^2}{\|c - \text{AE}_p(c)\|_2^2 + \epsilon}. \tag{2}$$

A lower value means that c follows the distribution of the class q better than that of class p [16].

As argued in the previous section, it is important to measure how well counterfactual examples follow the distribution of the training data. The IM1 score is a valuable tool for assessing such property. As the score is quantitative, it also allows comparing different methods across publications. Furthermore, the score is somewhat established as a metric, as multiple papers report the score [16, 17, 24].

The IM1 score can, however, be deceiving. We find experimentally that methods which make tiny changes to the input can get an undesired good score, presumably because tiny changes yields almost no error even if the changes are not preserved by the auto-encoders. Some classes may also be easier to reconstruct than others, resulting in skewed scores for different target classes. One target class may simply yield a lower numerator in Equation (2) than another target class, just because one is easier to reconstruct than the other. Finally, we know of no publicly available pre-trained auto-encoders for computing the score. When new auto-encoders need to be trained for each publication, results may not be comparable across publications. Through experiments, we demonstrate the issue by showing how, *e.g.*, differences in normalization yield incomparable scores.

IM2. The IM2 score is also introduced in [16]. It utilizes the discrepancy between reconstructions made by a class specific auto-encoder AE_q and an auto-encoder trained on the entire training set AE:

$$\text{IM2}\,(c) = \frac{\|\text{AE}_q\,(c) - \text{AE}\,(c)\|_2^2}{\|c\|_1 + \epsilon}. \tag{3}$$

According to the authors, a low value of IM2 indicates an interpretable counterfactual because the counterfactual follows the distribution of the target class as well as the distribution of the whole data set. The applicability of the score is however debatable. Schut et al. [24] demonstrate that the IM2 score fails to identify out-of-sample images. Mahajan et al. [17] also argue that both IM1 and IM2 are better reported by displaying both the denominator and numerator of each score. For both IM1 and IM2, we further find experimentally that when the complexity of the dataset increases, the computed scores get close to statistically insignificant amongst three different counterfactual methods. In turn, the two metrics may be best suited for datasets of lower complexity.

Fréchet Inception Distance. The Fréchet Inception Distance (FID) is a metric used for evaluating generative models [9]. The metric compares how similar two datasets are by comparing statistics of embeddings from the Inception V3 network [28]. For counterfactuals, the score has been used to evaluate how well counterfactuals align with the original dataset [23, 25]. FID is defined from mean Inception embeddings μ_1 and μ_2, and covariance matrices Σ_1 and Σ_2 of the test set and associated counterfactuals, respectively:

$$\text{FID} = \|\mu_1 - \mu_2\|_2^2 + tr[\Sigma_1 + \Sigma_2] - 2tr\left[\sqrt{\Sigma_1\Sigma_2}\right]. \tag{4}$$

In this work, we consider images that are smaller (64×64 pixels) compared to inputs of the Inception V3 model (299×299 pixels). Consequently, we compute the score for a different network. We use embeddings from the last hidden layer of a convolutional neural network, which is identical to the model being explained by the counterfactual methods. The last hidden layer has 256 output neurons, so we denote the score FID_{256}, to avoid any misconceptions. Although the score depends on the embedding network, we believe that our results will extend to the Inception V3 network.

The score is a good fit for evaluating whether generated counterfactuals follow the distribution of the training data, as it is currently the standard metric for evaluating generative models. It does, however, not take into account the relation between each specific input and its associated output. As such, the metric could, *e.g.*, be "fooled" by a high performing generative model producing realistic samples independent of the inputs. Consequently, the metric should be reported in combination with another metric which evaluates the validity of each counterfactual. We also find in experiments that methods generating tiny changes to the input may be deemed of high quality; maybe because the tiny changes are either filtered out by the embedding network or do not affect summarizing statistics of the score.

3.2 New Proposed Metrics.

Through experiments, we find that tiny changes similar to adversarial attacks often yield undesirable good scores. To mitigate this issue, we introduce two new metrics. Both metrics rely on the assumption that tiny adversarial-like counterfactuals are very model specific [15]. Under this assumption, evaluating counterfactuals on other classifiers should be less susceptible to tiny changes and more effective if the changes are semantically correct.

Label Variation Score. For datasets where each data point is associated with multiple class labels, individual classifiers for each class label can give insights into how each class is affected by a counterfactual change. Naturally, the class targeted by the counterfactual should be affected, while unrelated classes should not. At a high level, we use individual classifiers for each class label as a proxy for how much the concept related to the given class has changed in the counterfactual image.

We propose the Label Variation Score (LVS) to monitor predicted outcomes over different class labels. LVS computes average Jensen Shannon (JS) divergences, denoted d_{js}, between predictions on inputs and counterfactuals. Let o_l be an "oracle" trained on the class label l, which outputs a discrete probability distribution over the labels. then, LVS is defined as

$$\text{lvs}_l = \frac{1}{|x|}\sum_{x \in x} d_{js}\left[o_l(x)||o_l(cf(x))\right]. \tag{5}$$

As the score is based on individual classifiers for each class label, the score should be affected less by adversarial attacks. Intuitively, non-related labels should not be affected by counterfactuals and thus have a low LVS, while labels that correlate with the counterfactual label may co-vary and get a higher LVS. For example, if an image of a face without makeup is changed to one with makeup, the face in the counterfactual should be predicted to smile as much as before, but the prediction of "wearing lipstick" may follow the prediction of "wearing makeup" as lipstick is a subset of makeup.

LVS yields a rich picture of which features are changed by the counterfactuals, and it allows human judgement of which features are allowed to be changed, as with the makeup and lipstick example. Using the score, it becomes easier for humans to detect biases in the predictive model by identifying features that are changed unintendedly. On the contrary, LVS has the drawback that it needs multiple class labels to be applicable. Also, attributes that are not labeled will not be possible to monitor. In our experiments, LVS yields scores that align well with human interpretation on two different datasets. We further verify the underlying hypothesis described above by finding that more realistically looking counterfactuals get better scores than, *e.g.*, examples with tiny adversarial-like changes.

The oracle score. For datasets where LVS is not applicable, we propose to use a simpler metric which is based on training an additional classifier – the oracle – that is used to classify the counterfactuals examples. The score is the percentage of counterfactuals C that are classified to the target class by both the classifier being explained f and the oracle o:

$$\text{Oracle} = \frac{1}{|C|} \sum_{c \in C} 1_{[f(c)=o(c)]}. \tag{6}$$

The score is similar to TCV, but it is intended to avoid giving good scores to tiny adversarial-like changes. The oracle score depends on the additional oracle, which could tell more about the oracle than the predictive model itself. For example, it may be that adversarial attacks working on f also work on o, which would wrongly yield a good score for such attacks. However, we find experimentally that the score gives better scores for realistic counterfactual examples than tiny adversarial-like changes.

4 Experiments

In this section, we study the above described metrics for different types of counterfactual methods to characterize what properties different metrics capture. We demonstrate through multiple experiments that no metric can express all desirable properties, and thus they should be used in combination.

Methods. Throughout the experimental section, we compare three different methods for producing counterfactual explanations. The methods were chosen to represent a spectrum of methods ranging from gradient based methods producing sparse but less realistic changes in one extreme to methods based on generative models generating more realistic but larger changes in the other extreme.

In one end of the spectrum, Wachter et al. [30] present a gradient based method (denoted GB). Counterfactuals are generated through gradient descent on the input to minimize a loss-function comprising an L_1-norm "distance" term which encourages minimal and sparse changes and a squared "prediction" loss on the predicted label, which encourages valid counterfactual examples.[1] Another method that lies in this end of the spectrum is [6] which follows a similar loss-function as [30], but with a more complex distance function. It should be mentioned that both methods were originally introduced for tabular data. We here study it in the image domain as a simple method that produce counterfactuals with minimal changes that are looking less realistic.

At the other end of the spectrum, we include the method proposed by Hvilshøj et al. [10] as a representative for methods based on generative models (denoted GEN). The method is based on conditional invertible neural networks (INNs) which are generative models that can also do classification [2]. Counterfactual embeddings are found by correcting embeddings of inputs such that the predicted class provably change. Counterfactual examples are successively generated by inverting the embeddings with the INN. We find the method from [10] to be the most extreme case in this end of the spectrum, compared to, *e.g.*, [23, 25], because it uses the same neural network for both predictions and for generating counterfactuals. In contrast, [23] and [25] train surrogate generative models, which are used for sampling counterfactuals. We study the method as a more complex method which produces more realistic counterfactuals but with larger amounts of change.

At the middle of the spectrum, methods use gradients to compute counterfactuals similar [30], but where gradient optimizations are guided by derivatives of generative models or other more sophisticated loss terms to enhance the quality of the counterfactuals [5, 16]. In our experiments, we use the method proposed by Looveren and Klaise [16] as representative (denoted GL). The method uses embeddings from an auto-encoder to optimize a class-prototype loss. Such method should produce counterfactuals where both the visual quality and the amount of changes is in between GB and GEN. However, we find that in most cases, the visual quality is on par with GB in practice.

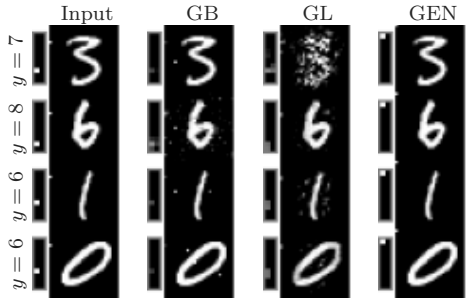
Experimental details. The methods from [30] and [16] were implemented using the `alibi` framework,[2] and [10] was adopted from the official code.[3] The former two methods are used to identify counterfactuals for the same "vanilla" convolutional neural network, identical to the one described in

[1] We do not normalize each feature by the median absolute deviation, as images have identical value ranges.

[2] `https://docs.seldon.io/projects/alibi` (v.0.5.9), default parameters. Apache License 2.0.

[3] `https://github.com/fhvilshoj/ECINN`, default parameters. MIT license.

(a) Counterfactual examples with target class $q = 0$.



(b) LVS scores.

Figure 1: Experimental results for the FakeMNIST dataset [10].

[16]. The latter method is based on a conditional INN as predictive model, identical to that of [10]. In turn, the presented results may be contributed to differences in architectures and not methods as such. However, the goal of the experiments is not to identify a superior method, but to demonstrate properties of metrics for evaluating counterfactual explanations on images. We note that the method from [10] generates counterfactuals for all classes different from the input class, so throughout experiments, we choose one target class uniformly at random for each input. Additional experimental details on, *e.g.*, hyperparameters for training are provided in the supplementary material.

Throughout the experiments, we report mean scores over the entire test set and 95% confidence intervals in parentheses. Except from TCV, we report scores only on valid counterfactual examples from the test sets, *i.e.*, we do not include counterfactuals that did not change the predicted class.

4.1 FakeMNIST

[10] propose FakeMNIST; an artificial dataset which dictates the relationship between pixels and labels. To generate the dataset, MNIST images [14] are shuffled and assigned new random labels. The top-left 10×1 pixels are colored according to the new labels, see first row of Figure 1a. The digits present in the images are independent of the labels while only the top-left pixels are label-dependent. The dataset can be used to test whether counterfactual methods change only class-related features. There are, however, no metrics associated with the dataset. We apply the LVS to further the evaluation protocol for the dataset and test if LVS detects methods that change label-independent features.

Column 1 of Figure 1a displays four samples from the FakeMNIST test set. Smaller rectangles magnify the top-left 10×2 pixels for increased readability. The first column displays inputs with labels 7, 8, 6, and 6, respectively (cf. labels or top-left dot locations). The following three columns are counterfactuals with target class $q = 0$, generated by the three representative methods. In Figure 1b, we show LVS for both the FakeMNIST labels and on the original MNIST labels. As intended, the LVS finds that GEN most successfully produces counterfactuals that change the predicted class (high LVS on FakeMNIST) and leaves the digit related pixels untouched (zero LVS on MNIST). Furthermore, the LVS reveals that both GB and GL produces less effective counterfactuals, as their LVS on FakeMNIST are lower. Through the high LVS on MNIST, the metric also finds that the two methods wrongly alters digit related pixels when generating counterfactuals.

In Table 3 in the supplementary material, we include scores of all other metrics described in Section 3. All the scores behave as expected and quantify differences between the methods properly. In conclusion, we find that for this simple dataset, qualitative observations and quantitative evaluations are well aligned in general. In turn, we argue that to provide a complete picture of performance, new methods can provide all the presented scores for the FakeMNIST dataset.

4.2 Normalization

Most metrics presented in this paper depend on data normalization or pretrained models. The dependence makes reporting both data normalization and model specifications crucial for reproducibility.

Table 1: Scores on MNIST for counterfactuals with different normalizations.

Method	EN	IM1	$100 \cdot$ IM2	FID_{256}	Oracle
		$[-0.5; 0.5]$ normalization			
GB	**16.07 (0.18)**	0.99 (0.00)	0.55 (0.01)	**50.23**	73.38% (0.87)
GL	42.76 (0.31)	0.99 (0.00)	0.53 (0.00)	308.43	37.71% (0.95)
GEN	99.17 (0.58)	**0.88 (0.00)**	**0.17 (0.00)**	90.73	**93.13% (0.50)**
		$[0; 1]$ normalization			
GB	**16.07 (0.18)**	1.06 (0.00)	2.46 (0.02)	**24.92**	48.25% (0.98)
GL	42.76 (0.31)	1.04 (0.00)	1.94 (0.01)	173.82	38.53% (0.95)
GEN	99.17 (0.58)	**0.89 (0.00)**	**1.47 (0.01)**	37.89	**91.92% (0.53)**

We demonstrate the normalization issue with a practical example where we apply the metrics to the same counterfactuals but with different normalization. The metrics have been adjusted to each normalization, *i.e.*, new models were trained to operate on the particular normalization.

In Table 1, we report mean scores for both a $[-0.5, 0.5]$ and a $[0, 1]$ normalization. By comparing the numbers between normalizations, we see that the best performing method for each metric is the same, independent of the normalization. In Figure 6 in the supplementary material, we even find this result to be statistically significant across 10 independently initialized models. It should be noted that the EN score is invariant to data shifts and scales linearly with the normalization range (cf. Table 1).

As the table indicates, there is, however, an issue. Had the IM2 metric been used to compare GL with a $[-0.5, 0.5]$ normalization against GEN with a $[0, 1]$ normalization, the conclusion would have been wrong, as GL would be deemed better than GEN. Although this issue may seem obvious, it occurs in literature. If one compares reported IM2 scores between [16] and [17], the difference is about an order of magnitude. [16] use normalization range $[-0.5, 0.5]$, while [17] use $[0, 1]$. We believe that the normalization differences contribute to explaining the difference between the reported scores. In turn, we propose to establish a common set of models with a fixed normalization range to be used for every evaluation, such that comparison across publications becomes possible. Upon publication, we will release our code to allow other researchers to easily evaluate their counterfactual methods.

4.3 Inspecting Scores

To get a deeper insight into how different metrics behave, we have identified pairs of inputs and counterfactuals for which there are unintended differences in scores. We find that in some cases, which may be important for evaluating counterfactuals on specific datasets with specific properties, existing metrics can be a source of wrong conclusions when applied in isolation. Except for FID_{256}, similar findings as those presented here were found for the CelebA-HQ dataset (see appendix).

EN. For the image domain, the EN distance is known to work poorly in terms of quantifying small interpretable changes [11]. For completeness, we demonstrate the issue in Figure 2a which shows a seven to the left and two counterfactuals with target class $q = 9$ (center and right). The EN distance is displayed above the two counterfactuals. Arguably, the center image looks most like a seven and the right image looks like a nine. However, according to the EN distance, the center image is an order of magnitude better than the right. The example illustrates how tiny adversarial attacks may be deemed better than proper counterfactual examples, just because they change the input less.

Figure 2: Examples of input (left) and counterfactual pairs (good: center, bad: right).

IM1. Figure 2b depicts a one with two counterfactuals with target class $q = 7$ and $q = 0$, respectively. The IM1 score is meant to quantify how realistic counterfactual examples are. Visually, the two counterfactual examples look similarly realistic. The center image does, however, get twice as good a score compared to the right, *i.e.*, the seven was deemed to be more realistic by the score. Holding all else equal, this might be because more white pixels yield a larger loss. For isolated cases like this, the IM1 score may produce undesirable results. As observed in Table 1, the metric does, however, seem to work well on average when comparing methods on MNIST. As such, the IM1 score is best used as a summarizing statistic to compare averages of many samples across methods.

IM2. For the IM2 score, which should yield lower values for more interpretable counterfactuals, Figure 2c shows how the score gives an almost completely black image a better score than an image of an eight digit. On the contrary, the right image with the worst score seems more interpretable from a human perspective. The center image is presumably scoring best because it contains close to no information, which is easier to reconstruct than the right image which contains more information. We demonstrate in the appendix that it holds more generally that the IM2 score decreases, when we decrease pixel values toward their minimal value. In turn, the IM2 score might wrongly give good scores to methods that produce less interpretable counterfactuals by removing information from the inputs. To account for such drawback, also reporting, *e.g.*, the oracle score, will make it harder to get good scores on both at once. A high score on both metrics at once is thus preferred.

FID_{256}. For FID_{256}, which quantifies the population wide similarity of sets of embedded images, it is not possible to identify single extreme samples. Instead, we observe how well FID_{256} distinguishes realistic and unrealistic samples. Figure 3 shows counterfactual examples and the test-set-wide FID_{256} scores. For a human observer, both GB and GL does a poor job in generating realistic changes. It is, *e.g.*, harder for humans to identify the target class for the two methods. FID_{256} does not identify realistically looking samples in this particular case, as it yields better scores for both GB and GL. Interestingly, we find in the next section that FID_{256} successfully identifies the more realistically looking counterfactuals for the more complex CelebA-HQ dataset. To deal with the identified issue, we argue that the FID_{256} score should be reported together with the LVS or the Oracle score, which are less vulnerable to tiny adversarial-like changes. If both the FID_{256} and the LVS scores are good, then the quality of the counterfactuals is more likely to be high.

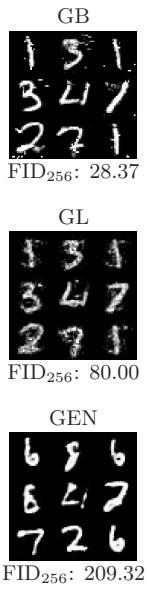


Figure 3: Counterfactuals on MNIST.

In conclusion, we find that in isolated cases, the metrics may be deceiving. We argue that the metrics should be reported jointly to account for each other's drawbacks. For example, if a method gets a low IM2 score indicating interpretable counterfactuals, a low FID_{256} score indicating realistic counterfactuals, and a high oracle score indicating that the counterfactuals generalize, it is a strong indicator that it is a good method.

4.4 Complex data

In this section, we scale our experiments to the more complex dataset, CelebA-HQ [12]. The goal is to evaluate how the studied metrics work in a more complex setting.

CelebA-HQ is a dataset of faces, where each sample is associated with 40 binary class labels. Figure 4 presents four different inputs in the first column. The first two have a positive makeup label and the last two have a negative label. The following three columns represent counterfactual examples of the opposite label value. Qualitatively, we find the three compared methods to produce counterfactual examples with similar properties as for FakeMNIST and MNIST. On the contrary, when we consider Table 2, we find that for some metrics the quantitative results vary from the previous experiments. Specifically, we see that the IM1 and IM2 scores fail to distinguish good from bad counterfactuals, as the scores yield almost the same value for all three methods. We also observe that the FID_{256}, in contrast to the previous experiment, successfully distinguishes the realistic from the unrealistic counterfactuals by giving GEN the lowest (best) score and GL the highest. In turn, for this more complex dataset, FID_{256} is not as vulnerable to tiny adversarial like attacks. For completeness, we also mention that the Oracle score and the EN metric behave as expected. That is, the oracle score

Table 2: CelebA-HQ scores

Method	TCV	EN	IM1	100·IM2	FID_{256}	Oracle
GB	96.07% (0.72)	**147.04 (2.04)**	**0.98 (0.00)**	**0.47 (0.01)**	205.59	82.82% (1.42)
GL	81.09% (1.44)	344.02 (18.13)	0.99 (0.00)	0.52 (0.01)	484.08	32.84% (1.92)
GEN	**99.26% (0.32)**	684.26 (11.86)	1.03 (0.00)	0.53 (0.01)	**98.35**	**89.91% (1.11)**

successfully identify the generative based method to most properly change the predicted class by the oracle, while the two other methods are found to be less successful. The EN metric correctly identifies the smallest changes, but the score is of little interest in the present comparison, as adversarial-like changes is still favored by the metric. In a comparison of two methods which do not produce such tiny changes, the metric might, however, be valuable to quantify how much each method changes.

To also evaluate LVS on the more complex dataset, we have computed the score for the counterfactual label (smile versus no smile) and four other labels. We chose the labels "lipstick" and "attractive" which should correlate more with the makeup label than the other two labels, "high cheekbones" and "smiling." Also on this dataset, LVS successfully avoids giving the best stores for the tiny adversarial-like changes and favors more realistic changes. Specifically, LVS identifies that GEN has a larger effect (high LVS) for the related makeup, attractiveness, and lipstick labels, while having similar low effect (low LVS) on the less related labels high cheekbones and smiling. LVS also successfully identifies how the changes made by, *e.g.*, GL has less effect on all the labels, which indicates that the counterfactuals are highly model specific and behave more like adversarial examples.

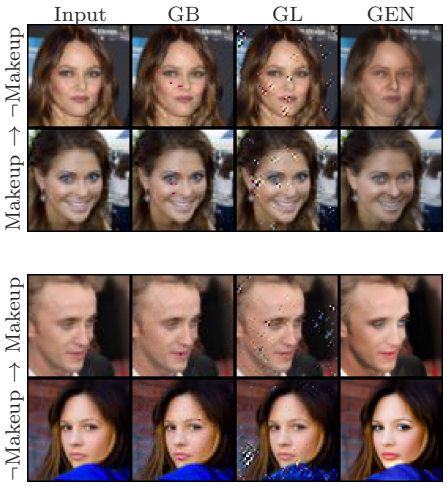


Figure 4: Counterfactuals for CelebA-HQ.

In summary, we find that for the more complex CelebA-HQ dataset, both the IM1 and the IM2 scores are less useful, while combining FID_{256} with the LVS yields a trustworthy quantitative evaluation of how realistic and valid the counterfactuals are, respectively. Minimal changes are still hard to quantify, but with two methods that perform on par on FID_{256} and LVS, the EN distance may be applicable as to judge how much each method changes.

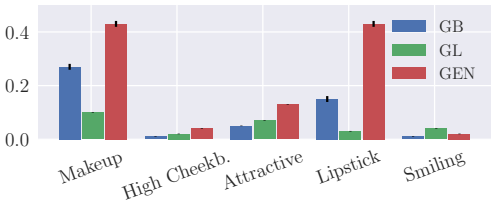


Figure 5: LVS for CelebA-HQ. Black vertical bars indicate 95% confidence intervals.

5 Conclusion

Through an analysis and experimental evaluations, we find that each quantitative metrics for evaluating visual counterfactual examples captures only some desired properties of good counterfactual examples. On the sufficiently simple dataset FakeMNIST, we found that all metrics considered behaves as expected. However, on the more complex datasets like MNIST and CelebA-HQ, behaviors deviate more from the intended. One particular issue is that visually unrealistic and tiny adversarial-like counterfactuals are very model specific and are often unintendedly deemed to be good by the metrics. To overcome this issue, we present the Label Variation Score and the oracle score, which are both based on surrogate predictive models that are less vulnerable to such tiny changes. To make a proper quantitative evaluation of visual counterfactual examples, we conclude that capturing all the desired properties is best done by reporting metrics concerning both realistic changes and validity together.

6 Limitations and Broader Impact

By analyses and experimental evaluations of quantitative metrics for evaluating counterfactual examples, this work contributes to improve scientific progress within counterfactual examples. As such, the work contributes to better understanding what can and can not be expected of different quantitative metrics. Such understanding will arguably yield better evaluations of counterfactuals and consequently improve the performance of methods for generating counterfactual examples. As such, we do not see any direct social impacts of this work. Indirectly, improving counterfactual examples can potentially enable attackers to fool automated machine learning systems by creating realistically looking adversarial examples, which yield desired outcomes.

We also recognize the limitations of our work. First, by limiting our evaluation to metrics that have been published at least twice, we have not done a complete evaluation of all existing metrics for evaluating counterfactual examples. In turn, there may be other metrics which better capture desired properties of counterfactual examples. Second, to limit the scope, we have chosen three representative counterfactual methods which represents specific properties in counterfactuals. As such, there may be other properties of counterfactual examples, that we have not evaluated and consequently do not know whether they effect metrics. Finally, due to a large spread in datasets used across publications, we have restricted our evaluation to three datasets of increasing complexity. From our work, it is not clear how our results extend to other datasets.

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018.

[2] Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. In *NeurIPS*, 2020.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 2015.

[4] Chun Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *ICLR*, 2019.

[5] 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. In *NeurIPS*, 2018.

[6] Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model agnostic contrastive explanations for structured data. *CoRR*, abs/1906.00117, 2019. URL `http://arxiv.org/abs/1906.00117`.

[7] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual Visual Explanations. In *ICML*, 2019.

[8] Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.*, 34(6):14–23, 2019. doi: 10.1109/MIS.2019.2957223. URL `https://doi.org/10.1109/MIS.2019.2957223`.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

[10] Frederik Hvilshøj, Alexandros Iosifidis, and Ira Assent. ECINN: efficient counterfactuals from invertible neural networks. *CoRR*, abs/2103.13701, 2021. URL `https://arxiv.org/abs/2103.13701`.

[11] Sin-Han Kang, Honggyu Jung, Dong-Ok Won, and Seong-Whan Lee. Counterfactual explanation based on gradual construction for deep networks. *CoRR*, abs/2008.01897, 2020. URL `https://arxiv.org/abs/2008.01897`.

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR*, 2018.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[14] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[15] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *ICLR*, 2017.

[16] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *CoRR*, abs/1907.02584, 2019. URL http://arxiv.org/abs/1907.02584.

[17] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *CoRR*, abs/1912.03277, 2019. URL http://arxiv.org/abs/1912.03277.

[18] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognit.*, 65:211–222, 2017. doi: 10.1016/j.patcog.2016.11.008. URL https://doi.org/10.1016/j.patcog.2016.11.008.

[19] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT**, 2020.

[20] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW: The Web Conference*, pages 3126–3132. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380087. URL https://doi.org/10.1145/3366423.3380087.

[21] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, Tijl De Bie, and Peter A. Flach. FACE: feasible and actionable counterfactual explanations. In Annette N. Markham, Julia Powles, Toby Walsh, and Anne L. Washington, editors, *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350. ACM, 2020. doi: 10.1145/3375627.3375850. URL https://doi.org/10.1145/3375627.3375850.

[22] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *SIGKDD*, 2016.

[23] Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam H. Laradji, Laurent Charlin, and David Vázquez. Beyond trivial counterfactual explanations with diverse valuable explanations. *CoRR*, abs/2103.10226, 2021. URL https://arxiv.org/abs/2103.10226.

[24] Lisa Schut, Oscar Key, Rory McGrath, Luca Costabello, Bogdan Sacaleanu, Medb Corcoran, and Yarin Gal. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*. PMLR, 2021.

[25] Sumedha Singla, Brian Pollack, Junxiang Chen, and Kayhan Batmanghelich. Explanation by Progressive Exaggeration. In *ICLR*, 2020.

[26] Ilia Stepin, José Maria Alonso, Alejandro Catalá, and Martin Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021. doi: 10.1109/ACCESS.2021.3051315. URL https://doi.org/10.1109/ACCESS.2021.3051315.

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[29] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark A. Neerincx. Contrastive explanations with local foil trees. *CoRR*, abs/1806.07470, 2018. URL http://arxiv.org/abs/1806.07470.

[30] 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.

Table 3: Test set wide mean (95% confidence intervals) on the FakeMNIST dataset. Best scores are reported in bold.

Method	TCV	EN	IM1	100·IM2	FID_{256}	Oracle
GB	68.11% (0.91)	11.60 (0.49)	1.22 (0.01)	0.49 (0.01)	252.5	88.31% (0.76)
GL	84.07% (0.72)	47.38 (0.90)	1.03 (0.00)	1.23 (0.03)	309.95	55.51% (1.06)
GEN	**100.00% (0.00)**	**6.81 (0.04)**	**0.68 (0.00)**	**0.21 (0.00)**	**0.12**	**99.98% (0.03)**

A Additional Experimental Results

A.1 FakeMNIST

In addition to the LVS, we also ran all other metrics on the FakeMNIST dataset. Table 3 presents all the scores. We find that all included scores behave as expected. Specifically, we expect the IM1 and IM2scores to identify that counterfactuals generated by GEN are the most realistic, as they change only the top left pixels, which should be easier to capture by the auto-encoders compared to the more scattered changes by both GB and GL. As only changing the top left pixels should produce a little difference in terms of the EN distance, we would also expect GEN to get the lowest score, which is also the case in Table 3. A similar argument also works for the FID_{256}. The FID_{256} should capture that the most realistic samples are those where only the top-left pixels are changed, which is also the case.

The TCVis not based on the perceptual quality of the counterfactuals, but on how effective each method is in changing the predicted class on the given classifier. We see from Table 3 that GEN is the most effective, which aligns well with the rest of our experiments. Finally, we see that the oracle score, which indicates whether the counterfactual examples also generalize to another classifier, also identifies how counterfactuals from GEN generalize better than those of GB and GL.

A.2 MNIST

To evaluate how sensitive the model based scores IM1, IM2, and the oracle score are to initialization of models, we trained ten individual classifiers with different random initializations for the MNIST dataset and computed the mean scores along with 95% confidence intervals. In Figure 6, we display the results, where bars represent mean values and horizontal black lined indicate confidence intervals. From the figure, we see that all three scores have statistically significant differences on the 95% level. It should be mentioned, that we test ten identical model architectures. In turn, the experiment does not reveal any information on whether results are also robust across different model architectures.

A.3 Inspecting Scores on CelebA-HQ

Similar to how we inspected scores on the MNIST dataset in Section 4.3, we have also considered similar input and counterfactual pairs for the more compled CelebA-HQ dataset. Results are shown

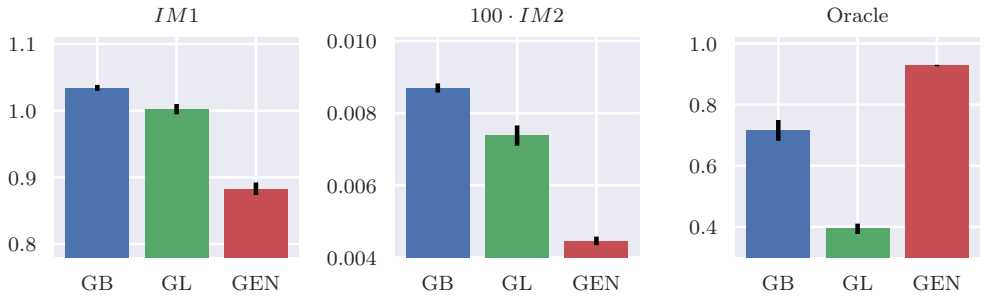


Figure 6: Mean scores on MNIST with 95% confidence intervals for ten trials with ten randomly initialized evaluation models.

(a) Counterfactuals examples for random CelebA-HQ sample.

(b) Scores for counterfactuals in Figure 7.

Method	EN	IM1	100·IM2
GB	**100.70**	**1.00**	0.25
GL	787.69	**1.00**	0.47
GEN	569.36	1.12	**0.22**

Figure 7: An example of the behavior of the three scores EN, IM1, and IM2on counterfactuals adding makeup to a face.

Table 4: Training configurations for convolutional classifiers used throughout the paper. FakeMNIST and MNIST classifiers were trained in an identical manner, thus (Fake)MNIST means both FakeMNIST and MNIST. † ADAM [13] was used with Keras default parameters: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$.

Configuration	(Fake)MNIST classifier	(Fake)MNIST oracle	CelebA
Learning rate	10^{-3}	10^{-3}	10^{-3}
Optimizer	ADAM†	ADAM†	ADAM†
Batch size	64	128	64
epochs	10	10	100

in Figure 7a, which shows a random sample from the dataset along with the counterfactuals generated by the three counterfactual methods used in this work. Table 7b shows the related scores. Figure 7 confirms the observations mentioned in Section 4.3, but also observations from our other experiment on CelebA-HQ (Section 4.4). Specifically, we see that EN finds the tiny adversarial-like changes from GB to be the best, which does not align with what a human observer would deem a good counterfactual example. As found in Section 4.4, IM1fail to distinguish counterfactuals from GB and GL. Finally, the IM2 score yields similar scores for GB and GEN, is also in contradiction to the human observations, as the sample form GENseems more interpretable.

B Experimental Details

In this section, we list all the relevant training details for the models used in this paper. We note that we also supply code at `https://github.com/fhvilshoj/EvaluatingCounterfactuals`, which also contains all counterfactual examples used throughout the experiments, the code used for evaluation, and all the models used.

Convolutional Neural Networks. GB and GL both generate counterfactual examples for convolutaional neural networks with the model architecture described in Looveren and Klaise [16]. For simplicity, we used the same model architecture for classifiers used with the oracle score, but with a different random initialization. Unless explicitly stated differently in the main paper, all data was normalized to a $[-0.5; 0.5]$ range. All convolutional neural networks were trained with categorical cross-entropy. Remaining configurations for the convolutaional neural networks are stated in Table 4.

Auto-encoders. The auto-encoders used for generating counterfactuals for GL ([16]) and for computing IM1 and IM2 scores had the same architecture as described by Looveren and Klaise [16]. We use independently initialized auto-encoders for computing and evaluating counterfactuals, respectively. The models were trained with mean squared error loss and remaining configurations presented in Table 5.

Conditional INNs. The conditional INNs used in this paper used exactly the architectures and the loss function described in [2]. We use $\beta = 1.4265$ for FakeMNIST and MNIST and $\beta = 1.0$ for CelebA, which was found to work well in [10]. We only present the "convincing" counterfactuals from [10] with the α_1-value suggested in the paper. For both the FakeMNIST and MNIST datasets, we use the smaller architecture in [2] and for the CelebA dataset, we use the deeper architecture

Table 5: Training configurations for auto-encoders used throughout the paper. FakeMNIST and MNIST models were trained in an identical manner, thus (Fake)MNIST means both FakeMNIST and MNIST. † Adam was used with Keras default parameters: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$.

Configuration	(Fake)MNIST oracle	CelebA
Learning rate	10^{-3}	10^{-3}
Optimizer	Adam†	Adam†
Batch size	128	64
epochs	50	50

presented for the CIFAR10 dataset in [2]. Additional configurations are presented in Table 6. We note that a full model specification and parameter configuration is also available in the public code repository.

Table 6: Training configurations for auto-encoders used throughout the paper. All configurations are identical to those of [10]. Stochastic Gradient Descent is abbreviated SGD below.

Configuration	(Fake)MNIST oracle	CelebA
β	1.4265	1.0
Learning rate	0.07	$5 \cdot 10^{-5}$
Optimizer	SGD	ADAM
Optimizer parameters	Momentum 0.9	$\beta_1 = 0.95, \beta_2 = 0.99$
Batch size	128	32
epochs	60	800
Scheduler	10^{-1} milestone	10^{-1} milestone
Milestones	50	200, 400, 600
Dequantization	Uniform	Uniform
Noise amplitude	10^{-2}	10^{-2}
Label smoothing	10^{-2}	0
Gradient norm clipping	8.0	2.0
Weight decay	10^{-4}	10^{-4}