The CF Score: Evaluating Counterfactuals

Anonymous Author(s)
Affiliation
Address
email

Abstract

►Write a proper abstract◄

1 Introduction

With the rising popularity of machine learning applications within the image domain, an urgent need for understanding machine learning models arise. As a tool for understanding machine learning models, many techniques for explaining such models have been proposed. In some cases, when fully automated algorithms affects peoples lives, an explanation can even required by law [11]. To name but a few techniques, some methods are based on generating heatmaps that identify salient input features which drive model predictions [30, 5, 24, 6], others rely on simpler surrogate models that are easier to explain [14], and some produce contrastive examples [38, 8]. In this work, we focus on counterfactual explanations for the image domain.

Counterfactual explanations aims at identifying specific changes to specific inputs, such that the predicted outcome of a machine learning model changes to a desired outcome. Such explanations are of particular interest in situations where users of systems based on machine learning needs to analyze the inner workings of the system. For example, a surveillance system picking out candidates for screening could be assessed for biases.

Work on counterfactual explanations has become increasingly popular [26, 29, 39, 38, 13, 12, 22, 14, 37, 7, 28, 9, 8, 21, 18, 10, 17, 2, 40]. However, no one common interpretation of properties that counterfactual explanations should possess, neither one standard metric exists for evaluating counterfactual explanations. In this work, we consolidate the desiderata of image counterfactuals into two categories. In terms of the two categories, we discuss existing metrics for image counterfactuals and demonstrate experimentally how such metrics sometimes fail in capturing the intentions of the metrics. We further propose two additional metrics, which contributes to a more complete profile of counterfactual methods. While we demonstrate that existing metrics do have flaws, we argue that reporting all metrics to give a more complete profile of counterfactuals is essential for the field of counterfactual explanations to move forward properly.

► **Add that when data complexity increases, then the performance of the metrics decrease.** ◄

2 Counterfactual Examples

In their original setting, counterfactual examples are closely connected to causality and structural causal models. The counterfactual example relates to "what if?" questions.

Definition 1 ***Causal counterfactual:*** *A counterfactual question is one where we seek information about how the world would have been different given some specific change. For example; in a situation where X* did *happened, what if X had* not *happened?* ►***cite Causation book or somewhere else for a definition.***◄

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

With structural causal models, one can compute probabilities of events happening in a parallel universe with an identical causal model and identical random variable outcomes, had variables (X) been forced to take other values. However, in deep learning, we are typically lacking such causal model, and a theoretical analysis becomes impossible. As an alternative, we come up with examples, that resembles the original input to the model, but with specific changes that makes the model under consideration predict a desired outcome. For example, if we have a model that predicts digit labels from images, we might turn a four into a nine by connecting the two top-lines of the four.

Such an example has a close resemblance to Definition 1, but does not answer the question as such. Instead, it answers a different type of counterfactual question.

Definition 2 ***None-causal counterfactual:*** *A none-causal counterfactual question is one where we seek what should have been changed to obtain a desired outcome. For example; input X yielded outcome Y, how should we change X to obtain outcome Z.*

An answer to this question typically comes in the form "Had values $v_1, \ldots, v_m$ been $\hat{v}_1, \ldots, \hat{v}_m$ and all other values remained the same, then outcome Y would have been Z." Naturally, such answers come in various forms and might not convey the information that we would expect. As such, much related work has defined additional criteria that counterfactuals need to possess in order to be useful for humans. We devote the next section to summarize such criteria.

3 The Good Counterfactual

For image counterfactual explanations to be useful for humans, both changing too little and too much of the input may be undesirable. For example, much research has gone into adversarial attacks, where images are changed slightly to change the predicted outcome, while not being perceptually different from the input [34]. In relation to explainability, such attacks are undesirable as they convey no information that humans will understand. On the contrary, changing too much could be counterfactuals that maintains no information from the original input. Such example does not reveal any actionable information either. In turn, related work describes various criteria to limit the output space of valuable counterfactuals. In the following, we have categorized the most pronounced criteria.

Minimal changes. The first criterion is that changes to the input should be minimal [**?**]. If a counterfactual example does not resemble the original input, it is of little or no use. In [16], the minimality criterion is slightly adapted to state that only class-dependent features should change, *i.e.*, features like connecting the two top-lines when changing a four to a nine can change, but features like tilt and font weight should not.

Minimal changes are also important in the user context. A classic example is the bank loan application. Imagine a customer who is not able to get a loan. If the counterfactual example says is that the customer needs to get a pay raise, have larger savings, and ten other changes, then it may not be actionable for the customer.

Especially for simpler datasets like tabular data, the amount of change can simply be counted [36, 13, 15]. However, as we will demonstrate in our experiments, simple counting and similarly manhattan or euclidean distances may be useful but have drawbacks on high-dimensional data like images. Dhurandhar et al. [8] also computes proxies for the optimal counterfactual features to be changed on tabular data and quantifies how many of such features are identified by their method. Computing such proxies are computationally heavy to identify and arguably of little relevance as pixels are inherently dependent on each other.

Realistic changes. Another criterion is that the counterfactual examples should be likely to stem from the same data distribution as the training data [27]. That is, the counterfactual examples should be realistic or feasible as phrased by Poyiadzi et al. [27]. [25] describes how counterfactual explanations should be faithful by both being in the proximity of the input and being connected to samples of the counterfactual class. Similarly, [22] presents the causal edge score and [37] the $IM1$ score to compare densities. One can see such attempts of forcing changes to be realistic as an attempt to guards against adversarial attacks and other artificial changes to the input.

For some domains, realistic changes also means that the counterfactual examples should obay, *e.g.*, physical laws. For example, loan applicants cannot increase their educational level without increasing their age [3]. This particular aspect is sometimes denoted both actionable [27] and feasible [**?**].

As for minimal changes, realistic changes are hard to quantify in the image domain. Is an adversarial attack for example realistic? Although being important and useful for consolidating the field of counterfactual explanations, we find in experiments that the metrics we studies all have pitfalls and should be used with care.

Computation time. If counterfactual examples are to be used in a real world setting, it is also important that they can be computed efficiently [36, 37, 16]. Assume that it takes 20 seconds every time an explanation has to be generated. In such case, interaction with a system to find answers will be very poor. According to **?**], every process that takes more than 0.1 second is too slow to be used in an interactive setting.

Other quantities that affect computation time that are more method specific, such as the number of gradient steps needed [37] or number of patches swapped between the input and a sample from the target class [12]. As such quantities cannot be used in a broad evaluation, we do not include them in this work.

Additional criteria. Some methods are based on simpler surrogate models, which mimic the more complex model under consideration. The surrogate models are used to explain the more complex models. Although we do not consider such approaches in this work, we briefly include some additional criteria here for completeness. It is obvious that such surrogate models should behave similarly to how the complex models behaves. To quantify this behavior, scores like percent wise agreement, accuracy, and F1-scores between surrogate and complex models are often used [14, 36, 26].

It is of course also important that humans can actually use the generated counterfactuals. In turn, multiple works have been testing their methods for, *e.g.*, human teaching [12] or expert verification [8]. The tests were both domain and method specific and thus prohotit a generalized test. Therefor, we do not include them for further evaluation here.

4 Evaluating Counterfactuals

As a basis for comparing methods in terms of minimal and realistic changes, multiple metrics have been proposed. In this section, we present those metrics which have been applied to images and discuss their applicability in terms of how they measure changes. We argue that the metrics are useful, but results should be evaluated as a whole and not stand-alone.

4.1 Existing Metrics

Simple distance metrics. A natural first approach to measuring the amount of change between inputs and counterfactuals is metrics like $L_1$and L_2-norms. However, as is broadly agreed in literature, such norms work poorly on high-dimensional data like images [18]. We include it here because both metrics (or variants there of) are used in objective functions for gradient based counterfactual methods [38, 7, 37] and are thus also a natural first choice for metrics.

In the experiment section, we include a hybrid metric denoted the elastic net distance (EN), which is defined as $EN(x, c) = \|x - c\|_1 + \beta\|x - c\|_2$, where x is the input and c is the counterfactual. We use this metric because it has both the properties of the $L_1$and the L_2-norm.

►I am not sure how much to write here. It is very simple norms that everyone knows, so I don't want to state the obvious too much. The main thing thay may be missing is what the pros and cons are in terms of minimal and realistic changes.◄

Target-class validity. A central property that is sometimes reported is the target-class validity [22], which tells the percentage of the generated counterfactuals that are predicted to be of the target class by the classifier under consideration. The score tells how effective a given method is in generating counterfactual examples.

FakeMNIST. Despite not being a metric as such, Hvilshøj et al. [16] introduced an image dataset which allows evaluating whether only class dependent features are changed, when producing counterfactual examples. MNIST images [20] were shuffled and assigned new random labels in order to generate the dataset. The top-left 10×1 pixels were then colored according to the new labels (see first row of Figure 1a in the experiments for examples). In turn, the digits present in the images are independent of the labels while only the top-left pixels are label-dependent. If a counterfactual method operates properly, only the ten top left pixels should change, as the rest of the image contains no class relevant information.

In a similar spirit as how methods for producing saliency maps should be able to pass the sanity checks described in [1], we find is reasonable to argue the counterfactual methods should be able to produce counterfactual examples that only change the ten top left pixels on the FakeMNIST dataset.

$IM1$ score. One way to measure realistic changes is proposed by [37], which employs auto-encoders to quantify how well counterfactual examples follow the distribution of the training data.

The $IM1$score is based on variational auto-encoders trained on the same training data as the classifier to be explained. The score shows the ration 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.

$$\text{IM1}\,(x, c) := \frac{\|c - \text{AE}_q(c)\|_2^2}{\|c - \text{AE}_p(c)\|_2^2 + \epsilon}. \tag{1}$$

A value below one means that c has less error when reconstructed with AE_q than with AE_p. According to Van Looveren and Klaise [37], a lower value means that $\hat{x}^{(q)}$ follows the distribution of the target class better than the distribution of the input class.

As argued in the previous section, it is important to measure how well counterfactual examples follows the distribution of the training data. The $IM1$score is a valuable tool for assessing such properties. As the score is quantitative, it also allows comparing different methods across publications. Finally, the score is somewhat established as a metric, as multiple published (and unpublished) papers report the score [37, 22, 32].

There are, however, some concerns about the score as well. First, we note that the L_2-normcan be deceiving when used on images. For example, we find in our experiments, that methods which make small changes to the input can get good scores, presumably because tiny changes that are not recovered yields only tiny additional errors in the score. Second, we have a concern that some classes may be easier to reconstruct than others, resulting in skewed scores for different inputs. For example, images of the digits one is arguably easier to reconstruct than, *e.g.*, images of the eight digit. As such, converting a one to an eight may yield a higher score than the oposite way. Finally, we have not identified any publicly available pre-trained auto-encoders for computing the scores. When new auto-encoders need to be trained for every publication, results may vary and may not be comparable across publications. In our experiments, we support this final concern by demonstrating how, *e.g.*, data-normalization will yield incomparable scores.

$IM2$ [37]. Van Looveren and Klaise [37] also propose the $IM2$score which utilize the discrepancy between reconstructions made by a class specific auto-encoder AE_q and an auto-encoder trained on the entire training set AE. The metric is defined as

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

In Equation (2), the squared discrepancy norm is normalized by the L_1-norm of the counterfactual. 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 quality of the score is however debatable. Mahajan et al. [22] argues that both $IM1$ and $IM2$ is better reported by displaying both the denominator and numerator of each score. Recently, Schut et al. [32] also demonstrated that the $IM2$ score fails to identify out-of-sample images. In a similar

vein, we find in our experiments that when comparing sufficiently complex datasets, the $IM2$ score becomes statistically insignificant amongst three different methods.

►**There may be more papers but I didn't find they yet.**◄

Fréchet Inception Distance. The Fréchet Inception Distance (FID) is a metric widely used for evaluating generative models [23]. The metric compares how much two datasets align by comparing statistics of embeddings of the Inception V3 network [35]. In terms of image counterfacatuals, the score has been used to evaluate how well counterfactuals align with the original dataset [31, 33]. The FID is defined from mean Inception embeddings μ_1 and μ_2, and covariance matrices Σ_1 and Σ_2 of the embeddings:

$$\text{FID} = \|\mu_1 - \mu_2\|_2^2 + tr[\Sigma_1 + \Sigma_2] - 2tr\left[\sqrt{\Sigma_1\Sigma_2}\right] \tag{3}$$

In this work, we consider images that are smaller (64×64) compared to inputs of the Inception V3 model (299×299). As a consequence, we compute the same score, but for a different network. In particular, we use the embeddings from the AE used in the $IM2$ score and denote it the Fréchet auto-encoder distance (FAED). Although the score is highly dependent on the embedding network, we believe that our experimental results will also extend to the inception network.

The score is well suited for evaluating whether generated counterfactuals follow the distribution of the training data. However, we find in experiments, that methods generating tiny changes to the input may be deemed of high quality because tiny changes are either filtered out by the auto-encoder or does not affect summarizing statistics.

4.2 New Proposed Metrics.

Oracle score. Through our experiments, we find that tiny changes similar to adversarial attacks often yield undesirable but good scores when evaluated with the presented metrics. To mitigate this issue, we introduce the oracle score, which is similar to the target-class validity but evaluated on an external classifier. The objective is to make a score which is less vulnerable to adversarial attacks.

The score is based on the intuition that adversarial attacks are highly specific to a particular classifier ►**make sure this is the case**◄. On the other hand, proper counterfactuals that change only class relevant features should generalize to other classifiers. The oracle score is based on training an additional classifier – the oracle – which is used to classify the counterfactuals. The score is then the percentage of counterfactuals that are predicted to be of the target class. ►**Should we discuss pros and cons further here?**◄

Independent feature comparison. For multi-label datasets, we can compute counterfactual examples for one label and use classifiers trained on the other labels to evaluate how much other classes change for the counterfactual examples. Our proposed method is similar to how Singla et al. [33] evaluate how a multi-class classifier changes the predicted outcome on the counterfactuals. We propose to use the Jensen-Shannon (JS) divergence to monitor predicted outcome of different classes. Over an entire test set, we compute counterfactuals for a particular label, and then we report average JS divergences between predictions on inputs and counterfactuals.

Intuitively, non-related labels should remain unchanged, while labels that correlate may co-vary. For example, if we change an image of a face without makeup to a face with makeup, we would expect the face to be smiling as much as the input, but the outcome of predicting whether the face is wearing lipstick should roughly follow the makeup label as lipstick is arguably a subset of wearing makeup. Experimentally, we find that more realistically looking counterfactuals behave more realistically than, *e.g.*, examples with tiny adversarial-like changes.

One particular example of a dataset, where our metric is applicable is the FakeMNIST dataset. On that dataset, we can train a classifier to predict the new "fake" labels and another to predict the original labels. This way, we can evaluate whether countefactual exmaples affect only the "fake" labels. In our experiments, our proposed metric yields scores that align well with human interpretation on two different datasets.

5 Experiments

In this section, we demonstrate how existing metrics need to accompany each other, when counterfactual methods are proposed. In particular, we demonstrate through multiple experiments that present metrics have vulnerabilities and should not be used in isolation.

Mehtods compared. Throughout the experimental section, we compare three different algorithms for producing counterfactual explanations. The three methods were chosen to represent a spectrum of methods ranging from simple gradient based methods in one extreme to methods based purely on generative models in the other extreme.

In one end of the spectrum, Wachter et al. [38] presents a gradient based method (denoted GB below). The method generates counterfactuals by minimizing a loss-function consisting of two terms, a prediction loss and a distance loss:

$$c = \arg\min_{c' \in \mathcal{X}} \lambda(f(c') - q)^2 + d(x, c'). \tag{4}$$

In Equation (4), f is the predictive model, d is a distance metric. For d, we use the L_1-norm, as suggested by the authors. However, we do not normalize by the median absolute deviation of each input feature, as this makes little sense for images which have identical value ranges. The first term in the minimization makes the predicted class change, while the second term encourages small and sparse changes to the input. Another method that lies in this end of the spectrum is, *e.g.*, [8] which follows a similar structure as Equation (4), but with a more complex distance function. It should be mentioned that both methods were originally introduced for tabular data. As we seek to investigate properties of metrics and not compare performances of methods, we see no issue in applying such a method to the image domain.

At the other end of the spectrum, we include ECINN [16] as a representative for the methods based on generative models (denoted GEN below). The method is based on invertible neural networks (INNs) which is a particular type of deep generative models that can be altered to also do classification [4]. ECINN makes corrections to embeddings of inputs. Counterfactual examples are successively generated by inverting the embeddings with the INN. We find ECINN to be the most extreme case of this end of the spectrum, compared to, *e.g.*, [31, 33], because it is the same neural network which is used for both predictions and for generating counterfactuals. In contrast, [31] and [33] trains a surrogate auto-encoder and generative adversarial network, respectively, which is used for sampling counterfactuals.

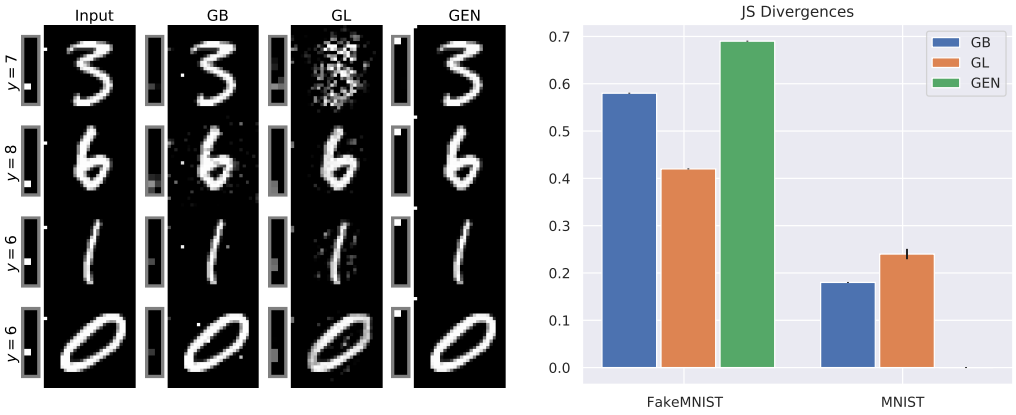
At the middle of the spectrum, methods exist which use gradients to compute counterfactuals similar to that of Equation (4), but where the gradient optimization is guided by derivatives of generative models or other more sophisticated loss terms [7, 37]. In our experiments, we use the method proposed by [37] as representative, which uses embeddings from an auto-encoder to define a class-prototype loss. We denote this group guided loss (GL) below.

The algorithms from [38] and [37] were implemented using the `alibi` framework,[1] while ECINN was adopted from the official code repository[2]. The former two algorithms are used to identify counterfactuals for the same "vanilla" convolutional neural network (CNN), identical to the one described in [37]. ECINN is based on inherent properties of INNs and is therefor used with an INN as predictive model. In turn, the presented results may be contributed to differences in architectures and not algorithms as such. However, the goal of the experiments is not to identify a superior algorithm, but to demonstrate properties of existing metrics for evaluating counterfactual explanations on images.

Experimental setup.

- Scores reported on test set
- Compute only on successful counterfactuals.
- Report 95% CI where appropriate.
- Independent feature comparison all with "Independent models", even the counterfactual label.

[1] `https://docs.seldon.io/projects/alibi` (v.0.5.9)
[2] MAKEURLHERE



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

(b) Mean JS-divergences.

Figure 1: FakeMNIST results. In (a), rectangular images left of larger images are the top left 10×2 pixels enlarged for improved readability.

For a fair comparison, the remaining metrics are only computed on the successful counterfactuals. Where appropriate, scores are reported with 95% confidence intervals in parentheses.

5.1 FakeMNIST

[16] proposed FakeMNIST; an artificial dataset which dictates the relationship between pixels and labels. The dataset can be used as a sanity check for whether counterfactual methods change only class relevant features. Originally, the dataset was introduced for a qualitative assessment of methods. As a result, no quantitative metric has been associated with the dataset. We apply our independent feature comparison to further the evaluation protocol for the dataset. In this experiment, we test how well quantitative and qualitative evaluations of counterfactuals align by quantifying whether only label dependent features of the input are changed.

Figure 1a displays four samples from the FakeMNIST testset. Smaller rectangles represent the first 10×2 pixels enlarged for readability. The first column displays inputs with labels 7, 8, 6, and 6, respectively (cf. labels or dot location in smaller rectangles). The following three columns are counterfactuals with target class $q = 0$, generated by the three paradigms we compare – gradient based (GB), guided loss (GL), and generative models (GEN).

As described, we would expect only the top-left 10×1 pixels to change. By inspecting Figure 1a, we observe three things. First, GB and GL both produce counterfactuals that do not seem to follow the distribution of the dataset. Specifically, the magnified parts either have multiple pixels colored or most color removed. Second, the gradient based approach seems to produce adversarial attacks, in which few lonely pixels are colored. Third, counterfactuals from GEN correspond well to what we would expect of a counterfactual.

We apply the independent feature comparison for counterfactuals generated for the entire testset. In Figure 1, we show how JS-divergences behave on the FakeMNIST labels (left) and on the original MNIST labels (right). Well aligned with what we see quantitatively, GEN is doing the best job in changing the predicted class on the FakeMNIST labels (high JS-divergence) and has almost no effect on the predicted MNIST labels. GB also has a high JS-divergence on the FakeMNIST labels but does also affect the predicted MNIST labels with a JS-divergence of just below 0.2. GL performs the worst by having the lowest FakeMNIST JS-divergence and the highest JS-divergence on MNIST. These observations align well with the intuition we get from Figure 1a, where GEN behaves as expected, GB behaves less sensible but with only small modifications to the input, and GL produces more blurry and artificially looking counterfactuals. In Table 3 in the supplementary material, we also include scores of all other metrics described in Section 4. On all the scores the results are similar to what we observe in this experiment, *i.e.*, GEN has the best scores, GL the worst, and GB is in between. In conclusion, we find that for this simple dataset, qualitative observations and quantitative evaluations are well aligned.

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

Method	$IM1$	$100 \cdot IM2$	EN	Oracle	FAED
$[-0.5; 0.5]$ normalization					
GB	1.04 (0.00)	0.96 (0.01)	**16.04 (0.18)**	73.15%	**0.19**
GL	1.04 (0.01)	0.89 (0.01)	42.72 (0.31)	37.64%	1.12
GEN	**0.77 (0.00)**	**0.21 (0.00)**	108.25 (0.59)	**97.78%**	7.12
$[0; 1]$ normalization					
GB	1.06 (0.00)	2.45 (0.00)	**16.04 (0.18)**	81.18%	**0.41**
GL	1.04 (0.00)	1.94 (0.00)	42.72 (0.31)	45.13%	2.12
GEN	**0.67 (0.00)**	**0.95 (0.00)**	108.25 (0.59)	**96.16%**	14.34
$[-1; 1]$ normalization					
GB	1.05 (0.00)	1.87 (0.02)	**32.09 (0.36)**	67.34%	**1.84**
GL	0.99 (0.00)	1.73 (0.01)	85.44 (0.62)	38.43%	9.97
GEN	**0.86 (0.00)**	**0.47 (0.01)**	216.49 (1.17)	**96.61%**	68.49

5.2 Normalization and Models Matters

Most metrics presented in this paper depend on data normalization or particular predictive models. The dependence makes reporting both data normalization and model specifications crucial for reproducibility. In this section, we demonstrate the mentioned issue with a practical example.

For different ranges of normalization, we train models used for the various evaluation metrics, *i.e.*, auto-encoders for IM1, IM2, and FAED; and a new classifier for the oracle score. For each counterfactual method, we use the same counterfactuals, but normalized differently, and evaluate each set with the metrics specified for the corresponding normalization.

In Table 1, we report mean scores for three different normalizations. Parenthesis indicate 95% confidence intervals. By pattern matching the bold numbers between the tables, we see that the best performing method would be concluded to be the same, independent of the normalization. In Figure 6 in the supplementary material, we even find the orders 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 scaling of the data range, which is what we also see in the table.

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 have been looking better than GEN. Although this issue may seem obvious, it occures in literature. If one compares reported $IM2$ scores between [37] and [22], the difference is about an order of magnitude. [37] normalizes values to the range $[-0.5, 0.5]$, while [22] normalizes values to the range $[0, 1]$. We believe that the normalization differences contribute to explaining the difference between the reported scores.

5.3 Inspecting Scores

To get a deeper insight into how different metrics behave, we have identified cases of input and counterfactual pairs, where the differences in scores are extreme. The extreme cases demonstrate situations, where the metrics may need to be used with caution.

L1-norm. For the image domain, the simple norms including the EN distance are known to work poorly in terms of interpretability [18]. For completeness, we give an example in figure Figure 2a. The figure shows a seven to the left and two successful counterfactuals with target class $q = 9$ (center and right). The EN distance is displayed above the two counterfactuals. Arguably, the center image still looks most like a seven and the right image convincingly 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 adversarial examples, simply because they change the input much less.

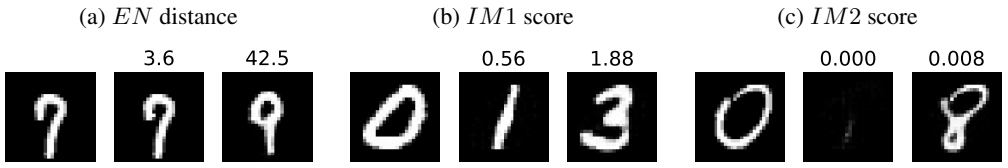


Figure 2: Each figure contains three samples. Left: input image, center: a counterfactual with a good score, right: a counterfactual with bad score.

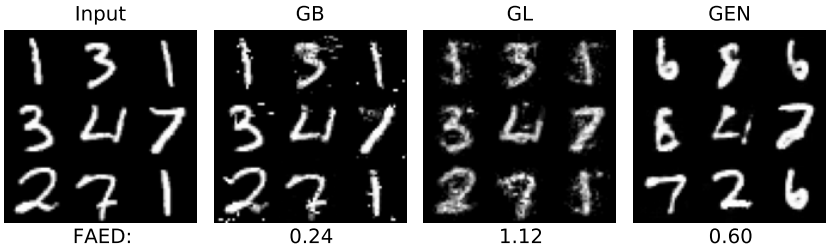


Figure 3: Counterfactual samples and the test-set-wide FAED.

IM1. In Figure 2b, we depict a zero with two different counterfactuals of target class $q = 1$ and $q = 3$, respectively. Visually, the three in the right image resembles the input better than the one in the center image, as it preserves most of the shape, tilt and font weight of the zero. The one seems more distant as the shape of the digit is much different from the zero. The center image does, however, get a three times better $IM1$ score than the three, *i.e.*, the one was better reconstructed by the auto-encoders than the three. Holding all else equal, this is presumably because more white pixels yields a larger loss.

[F: The mean reconstruction loss for AE_0 on class zero digits is 1.28, while it is 0.61 for AE_1 on class one inputs, and 1.45 for AE_3 on threes.]

IM2. For the IM2 metric, we show an extreme case in Figure 2c. Figure 2c shows how an all black image (center) yields a better (lower) score than a regular digit of an eight. Again, the argument is presumably that there is less to reconstruct for the auto-encoders in the center image than the right image. This observation is confirmed in the appendix, where we compute $IM2$ scores over samples where we gradually turn white pixel into black. In turn, the particular metric might wrongly give good scores for methods that simply removes information from the inputs.

FID. For Frechét distances like FID, it is not possible to identify single extreme cases because it is a population wide metric. Instead we compare scores between counterfactuals generated by GB which makes small local changes, GL which makes more blurry changes, and GEN which generally makes more global changes (cf. Figure 3 and the appendix for more samples).

In Figure 3, we show counterfactual examples and the test-set-wide FAED scores of ten thousand counterfactuals generated by the three methods, compared to the MNIST test set. The figure shows how GB has the smallest (best) FAED score, GL the highest (worst), and GEN is in between. From a human perspective, when the goal is to change the class of the input, both GB and GL does a poor job in generating realistic changes, as it is harder for humans to identify what the target class is in most cases. The two extremes are deemed good (bad) according to the FAED metric, because they make small (large) changes to the input. GEN also makes large changes to the input, but the changes better resembles the true data distribution and thus gets a lower score than GL. In turn, smaller values of FAED may be better, but very small changes similar to adversarial attacks may wrongly yield a low score.

We see that averages work well but in isolated cases, which may be important for evaluating methods on specific datasets or cases with specific properties....

► I find it hard to conclude this subsection in a positive way. We basically find that metrics have different flaws in extreme cases (vulnerable to adversarial attacks, simpler classes, and

Figure 4: Counterfactuals for the CelebA dataset.

Table 2: CelebA-HQ scores

Method	TCV	$IM1$	$100 \cdot IM2$	EN	Oracle	$FAED$
GB	96.07% (0.72)	**0.98 (0.00)**	**0.48 (0.01)**	**142.21 (2.08)**	0.83 (0.01)	0.70
GL	81.09% (1.44)	0.99 (0.00)	0.53 (0.01)	330.14 (18.24)	0.33 (0.02)	4.61
GEN	**99.26% (0.32)**	1.03 (0.00)	0.53 (0.01)	684.26 (11.86)	**0.90 (0.01)**	**0.63**

removing information). We could do the discussion on hardness of small but not too small changes here. But what is the positive message? ◀

5.4 Complex data

In this section, we scale our experiments to the larger and more complex dataset, CelebA-HQ [19]. The goal is to compare qualitative and quantitative assessments of the counterfactual examples generated in a more complex setting.

CelebA-HQ is a dataset of faces, where each sample is associated with 40 different labels. Figure 4 presents ten different inputs in the first row. The first five have a positive makeup label and the last five have a negative label. The following three rows represent counterfactual examples. Similar to what was observed in former experiments, GB generates counterfactual examples which represents tiny local visible changes in the form of few pixels being changed to extreme color values. GL generates more noisy and artificial changes. In most cases, GEN produces counterfactual examples that look more natures. However, we also observe artificially looking changes like the one in the lower right corner. Arguably, from a human perspective, the last row of counterfactuals seems to contin counterfactuals of the highest quality.

Table 2 presents scores computed on the 2824 test samples. Slightly in contradiction with the qualitative evaluation above, the table indicates that as an overall picture, GB and GEN are the best performing, as they each performs best on three metrics. GL does not perform best on any metric. Except from the $IM1$ and $IM2$ metric, the scores are well separated and Taking a closer look at the scores, the $IM1$ and $IM2$ scores have similar values for all three methods. As such, it seems the only little can be concluded from the particular value.

Comparing Figure 4 and Table 2 jointly reveals how on this dataset, the $IM1$ and $IM2$ support our observation that the qualitative and quantitative evaluations support each other poorly. As such, there is an apparent need for a suit of metrics that follows human intuition better.

6 Future Work

- Alignment of desiderata and metrics for images.

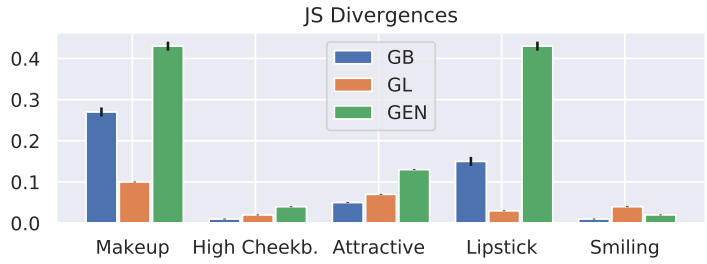


Figure 5: Average JS-divergences between predictions of originals and counterfactual examples for varying labels. Black vertical bars indicate 95% confidence intervals.

- Examples:
  - Leave class-independent features alone.
  - Better distance metrics that reflects our intuition.
- Interventional data.
- A stronger tie to causal models.

In [16], they describe how it is important that only class-dependent features are changed. The CelebA-HQ data lends itself useful to investigate this property. We have trained five independent classifiers to predict five different labels. We then use the Jensen-Shannon (JS) divergence to compare average predicted outcomes of the five classifiers, when comparing the original input to counterfactuals generated. For each method, we use each classifier to predict the class of the original inputs and the generated counterfactuals and compute the average JS divergences across the test set. Figure 5 displays the results. The counterfactuals were generated for the makeup label which is why the divergence should be high on that label, *i.e.*, the predicted class should have changed. On the remaining classes like, *e.g.*, smiling, the divergence should stay low, because the smiling property should not be changed when adding or removing makeup. According to the table, the divergences does in deed mostly remain low for the additional classes. For ECINN and partly Wachter, the divergence of the libstick label does increase with the counterfactuals. This is presumably because libstick is highly correlated with wearing makeup in general. A similar argument can be made for being attractive. Figure 5 reveils how ECINN and to some extend Wachter generates counterfactuals that maintain such relationship, while this is generally not the case for Van Looveren.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018.

[2] Arjun Akula, Shuai Wang, and Song-Chun Zhu. CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[3] Holger Andreas and Lorenzo Casini. Hypothetical interventions and belief changes. *Foundations of Science*, 24(4):681–704, 2019.

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

[5] 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.

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

[7] 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.

[8] Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model agnostic contrastive explanations for structured data. *arXiv*, 2019.

[9] Nadine Elzein. The demand for contrastive explanations. *Philosophical Studies*, 176(5):1325–1339, 2019. URL `https://doi.org/10.1007/s11098-018-1065-z`.

[10] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. ViCE: Visual Counterfactual Explanations for Machine Learning Models. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pages 531–535, 2020.

[11] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

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

[13] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. Interpretable Credit Application Predictions With Counterfactual Explanations. *arXiv preprint arXiv:1811.05245*, 2018.

[14] Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and Counterfactual Explanations for Black Box Decision Making. *IEEE Intelligent Systems*, 34:14–23, 2019.

[15] Masoud Hashemi and Ali Fathi. PermuteAttack: Counterfactual Explanation of Machine Learning Credit Scorecards. (August), 2020. URL `http://arxiv.org/abs/2008.10138`.

[16] Frederik Hvilshøj, Alexandros Iosifidis, and Ira Assent. ECINN: efficient counterfactuals from invertible neural networks. *arXiv preprint arXiv:2103.13701*, 2021.

[17] Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. Contrastive Explanations for Model Interpretability. *arXiv preprint arXiv:2103.01378*, 2021.

[18] Sin Han Kang, Hong Gyu Jung, Dong Ok Won, and Seong Whan Lee. Counterfactual explanation based on gradual construction for deep networks. *arXiv*, pages 1–26, 2020.

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

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

[21] Weixin Liang, James Zou, and Zhou Yu. ALICE: Active Learning with Contrastive Natural Language Explanations. pages 4380–4391, 2020. ISSN 23318422.

[22] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv*, 2019.

[23] Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fr\'echet inception distance. *arXiv preprint arXiv:2009.14075*, 2020.

[24] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, 2017.

[25] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, pages 3126–3132, 2020.

[26] Tejaswini Pedapati, Avinash Balakrishnan, Karthikeyan Shanmugam, and Amit Dhurandhar. Learning global transparent models from local contrastive explanations. *arXiv*, (NeurIPS), 2020.

[27] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020. doi: 10.1145/3375627.3375850.

[28] Mohit Prabhushankar, Gukyeong Kwon, Dogancan Temel, and Ghassan AlRegib. Contrastive Explanations in Neural Networks. In *ICIP*, 2020.

[29] Shubham Rathi. Generating Counterfactual and Contrastive Explanations using SHAP. *arXiv*, 2019.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *KDD*, 2016.

[31] Pau Rodriguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam Laradji, Laurent Charlin, and David Vazquez. Beyond Trivial Counterfactual Explanations with Diverse Valuable Explanations. 2021. URL `http://arxiv.org/abs/2103.10226`.

[32] 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. 130, 2021. URL `http://arxiv.org/abs/2103.08951`.

[33] Sumedha Singla, Brian Pollack, Junxiang Chen, and Kayhan Batmanghelich. Explanation by Progressive Exaggeration. pages 1–20, 2019.

[34] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

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

[36] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive Explanations with Local Foil Trees. 2018. URL `http://arxiv.org/abs/1806.07470`.

[37] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019.

[38] 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.

[39] Pei Wang and Nuno Vasconcelos. SCOUT: Self-aware Discriminant Counterfactual Explanations. In *CVPR*, 2020.

[40] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. Polyjuice: Automated, General-purpose Counterfactual Generation. *arXiv preprint arXiv:2101.00288*, 2021.

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

Method	$IM1$	$100 \cdot IM2$	TCV	EN	Oracle	$FAED$
GB	1.22 (0.01)	0.50 (0.01)	0.68 (0.01)	11.56 (0.49)	0.88 (0.01)	0.52
GL	1.02 (0.00)	1.23 (0.03)	0.84 (0.01)	47.34 (0.91)	0.56 (0.01)	2.44
GEN	**0.70 (0.00)**	**0.22 (0.00)**	**1.00 (0.00)**	**6.72 (0.04)**	**1.00 (0.00)**	**0.15**

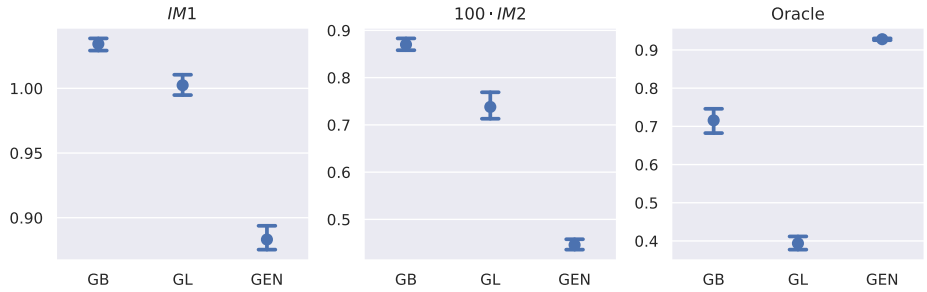


Figure 6: Mean scores on MNIST. Confidence intervals for ten trials with ten randomly initialized evaluation models.

A Additional Experimental Results

A.1 FakeMNIST

► Describe ◄

A.2 MNIST

► Describe ◄

A.3 Additional Samples

FakeMNIST

MNIST

CELEBA

A.4 Extremes

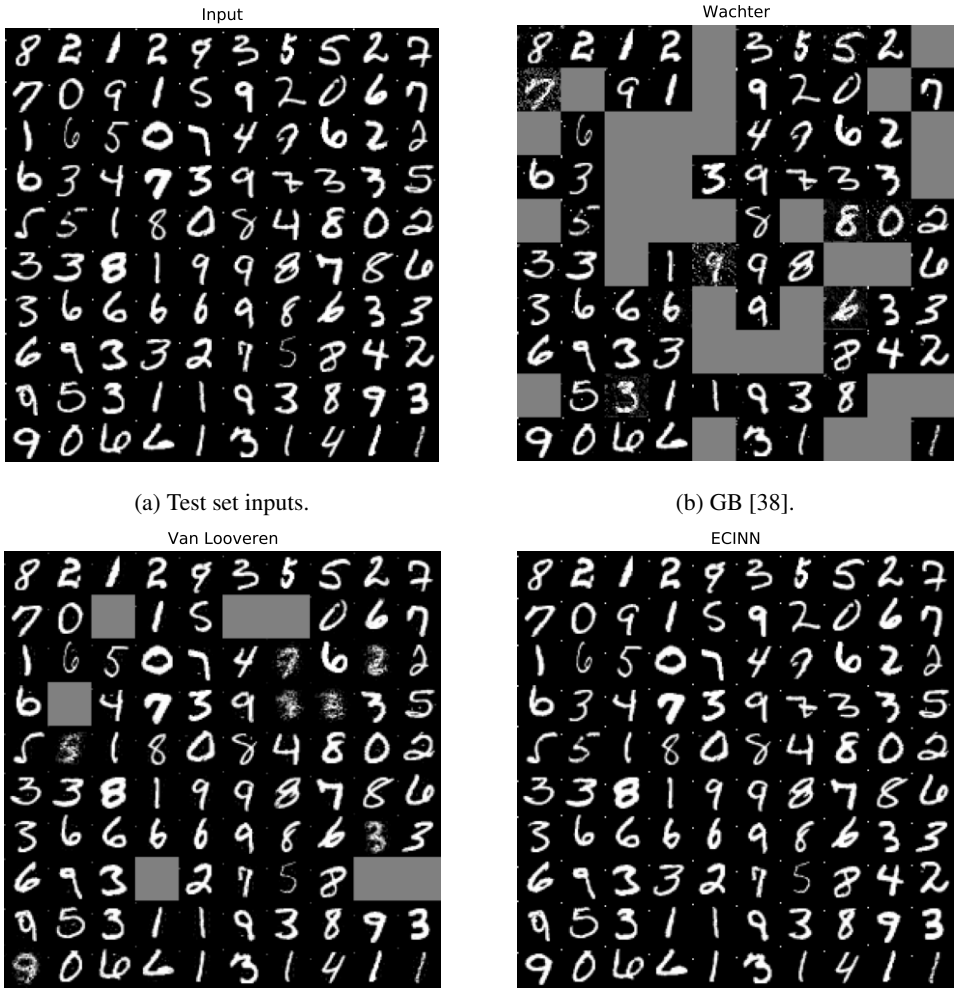
L1

L2

EN

IM1 AE distances might be misleading:

To test this interpretation, for each MNIST class, we computed average reconstruction errors for each of the ten autoencoders AE_i, $\ell_i = \frac{1}{n}\sum_x \|x - AE_i(x)\|^2$. Figure 16 displays the average per-row relationship between input classes (rows) and autoencoders (columns). If the interpretation was true, we would expect the samples from class i to have the lowest reconstruction loss of the corresponding autoencoder, ℓ_i, thus making the diagonal in Figure 16 dark. However, inspecting Figure 16 reveals



(a) Test set inputs.

(b) GB [38].

(c) Prototype [37].

(d) INN [16].

Figure 7: FakeMNIST [16] samples. Gray samples are those where no solution were found.

that on average, threes are reconstructed better with, e.g., AE_2, AE_5, or AE_8 than with AE_3. In turn, the interpretation above does not seem to hold in practice.

Additionally, Figure 16 shows that AE_1 is bad at reconstructing anything but ones, judging by the mostly yellow second column. As such, when $||x - AE_1(x)||_2^2$ becomes the denominator in Equation (1), then the score will look better than with the other autoencoders. Simply because AE_1 cannot reconstruct, e.g., an eight, as well as either of the other autoencoders.

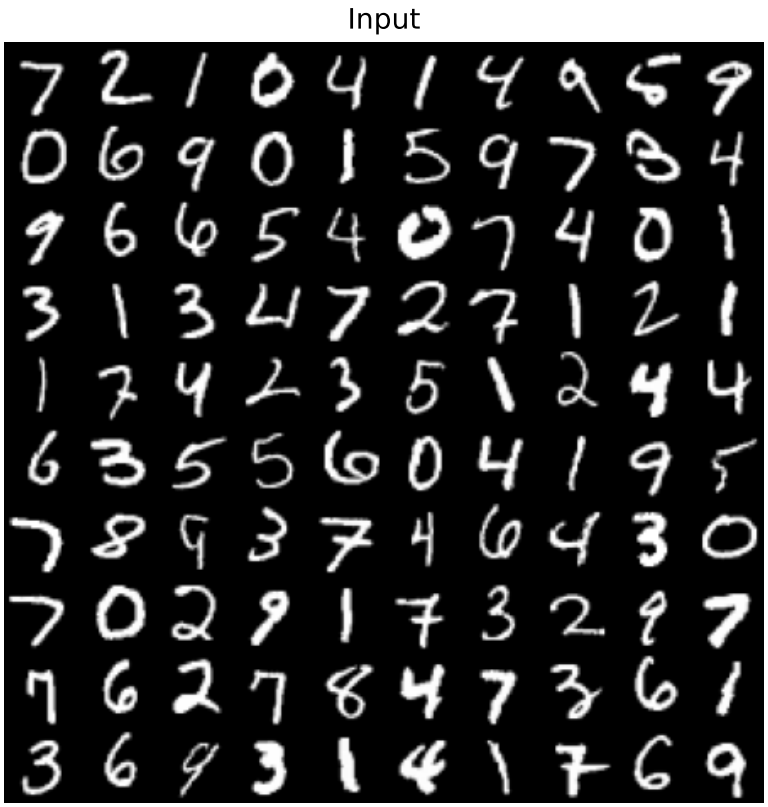
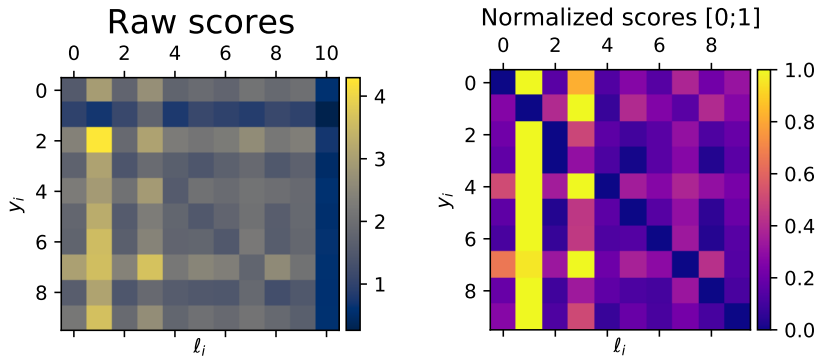


Figure 8: MNIST test set samples.



IM2 Graying out gives lower score:

In a similar vein as how we compared scores of $im1$ in figure 2b, we have displayed an input and two counterfactuals with low (good) respectively high (bad) $im2$ score in figure 2c. The figure shows how removing almost all content from the input yields good $IM2$ score while a more reasonable counterfactual where the input zero with an "open top-line" is translated into an eight with a similar "open top-line" gets a 15 times worse score.

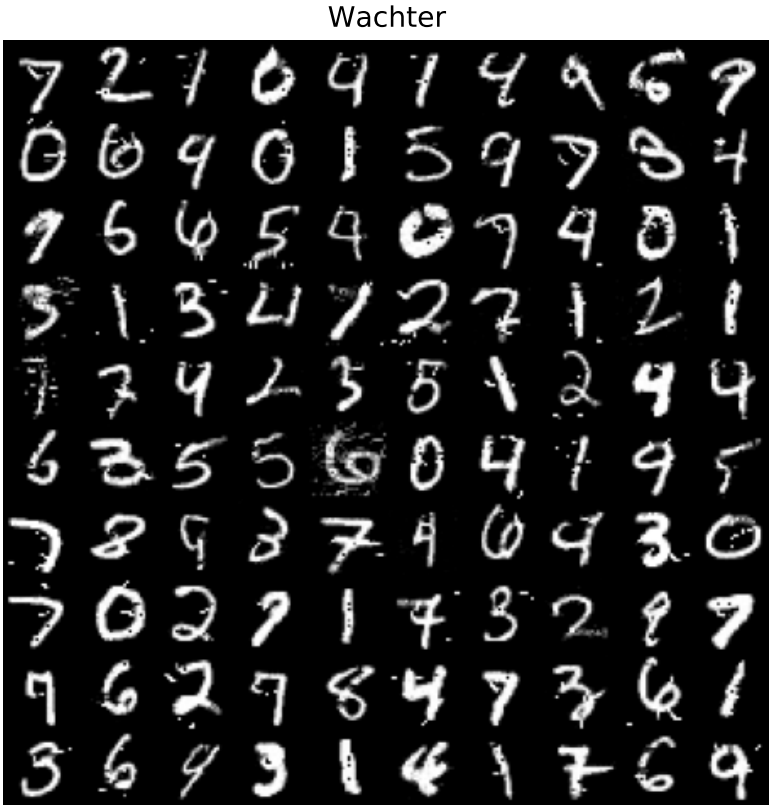
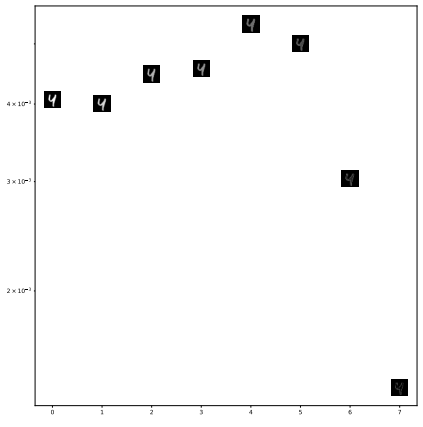


Figure 9: MNIST counterfactuals generated by algorithm from [38].



Van Looveren

Figure 10: MNIST counterfactuals generated by algorithm from [37]

ECINN

Figure 11: MNIST counterfactuals generated by algorithm from [16]

Figure 12: CelebA-HQ test set samples.

Figure 13: CelebA-HQ counterfactuals generated by algorithm from [38].

Figure 14: CelebA-HQ counterfactuals generated by algorithm from [37]

ECINN

Figure 15: CelebA-HQ counterfactuals generated by algorithm from [16]

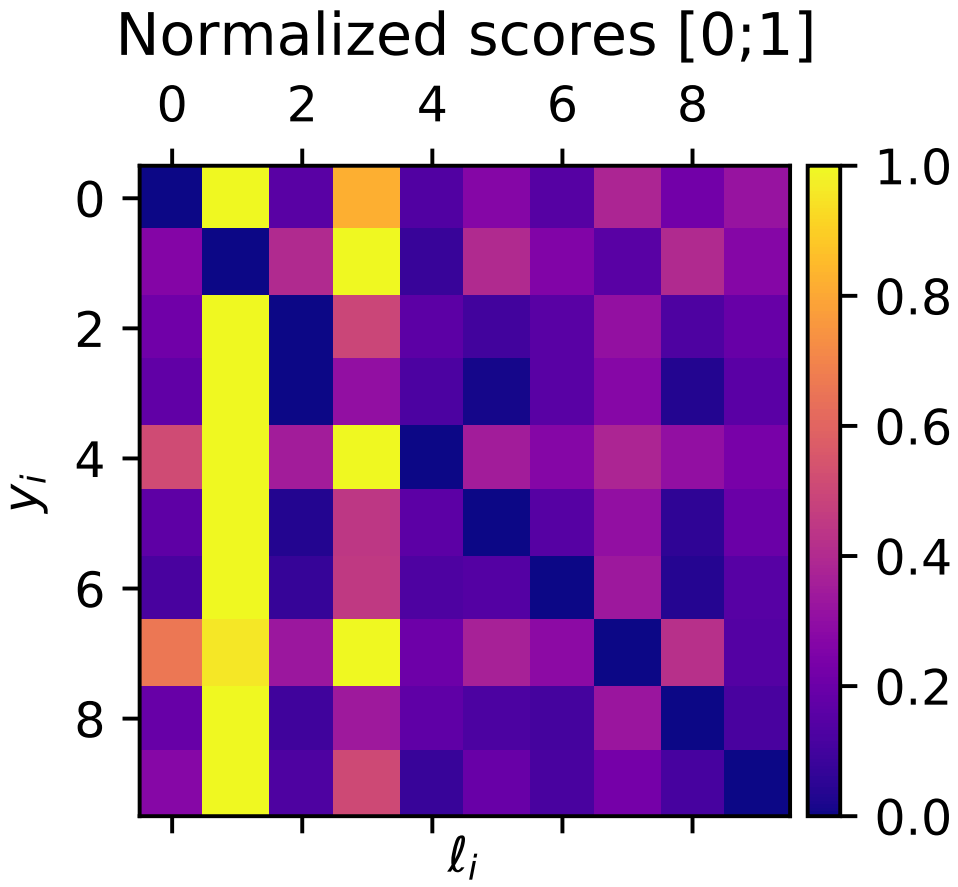


Figure 16: Mean autoencoder reconstruction errors. Rows correspond to input class, columns correspond to autoencoder idx. The plot only represents row-wise relationships, as each row is normalized such that the largest value is 1 and the lowest is 0.