Diffeomorphic Counterfactuals with Generative Models

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Diffeomorphic Counterfactuals
With Generative Models

Ann-Kathrin Dombrowski ⬤, Jan E. Gerken ⬤, Klaus-Robert Müller ⬤, and Pan Kessel ⬤

Abstract—**Counterfactuals can explain classification decisions of neural networks in a human interpretable way. We propose a simple but effective method to generate such counterfactuals. More specifically, we perform a suitable diffeomorphic coordinate transformation and then perform gradient ascent in these coordinates to find counterfactuals which are classified with great confidence as a specified target class. We propose two methods to leverage generative models to construct such suitable coordinate systems that are either exactly or approximately diffeomorphic. We analyze the generation process theoretically using Riemannian differential geometry and validate the quality of the generated counterfactuals using various qualitative and quantitative measures.**

Index Terms—**Counterfactual explanations, explainable artificial intelligence, data manifold, generative models.**
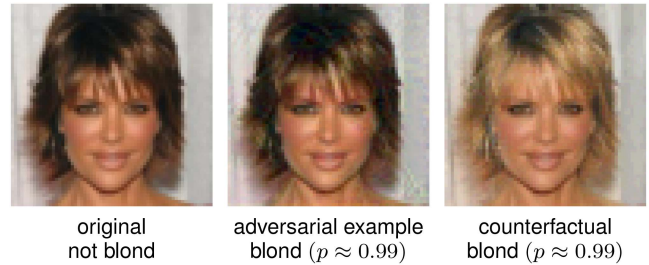

Fig. 1. Example of a counterfactual from the CelebA dataset. The original is classified as not blond. The adversarial is classified with high confidence as blond, but the difference to the original resembles unstructured noise. The counterfactual is also classified with high confidence as blond but in contrast to the adversarial example it shows semantic differences to the original.

I. INTRODUCTION

DEEP neural network models are widely used to solve complex problems from computer vision (e.g., [1], [2], [3], [4], [5]), strategic games and robotics (e.g., [6], [7], [8]), to

Ann-Kathrin Dombrowski was with the Berlin Institute of Technology (TU), 10587 Berlin, Germany. She is now with SERI ML Alignment Theory Scholars Program, Berkeley, CA 94704 USA, and also with the Principles of Intelligent Behavior in Biological and Social Systems (PIBBSS), 182 00 Prague, Czech Republic (e-mail: dombrowski@gmail.com).

Jan E. Gerken is with the Chalmers University of Technology and the University of Gothenburg, 41296 Gothenburg, Sweden (e-mail: gerken@chalmers.se).

Klaus-Robert Müller is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany, also with the Department of Artificial Intelligence, Korea University, Seoul 136-713, South Korea, also with Max Planck Institut für Informatik, 66123 Saarbrücken, Germany, and also with Google DeepMind, 10117 Berlin, Germany (e-mail: klaus-robert.mueller@tu-berlin.de).

Pan Kessel is with Prescient Design, Genentech, 4070 Basel, Switzerland (e-mail: pan.kessel@gmail.com).

medicine (e.g., [9], [10], [11], [12]) and the sciences (e.g., [13], [14], [15], [16], [17]). However, they are traditionally seen as black-box models, i.e., given the network model, it has been unclear to the user and even the engineer designing the algorithm, what has been most important to reach a particular output prediction. This can cause serious obstacles for applications since, say, networks using spurious image features that are only present in the training data might go unnoticed. Such undesired behaviour hampering the network's generalization ability has indeed been identified using explanation methods [18], [19] and is particularly problematic in safety-critical areas.

Supplying this desired transparency has been the subject of recent developments in the field of explainable AI (XAI) [20], [21], [22], [23], [24], ameliorating the aforementioned challenges. Prominent techniques in this area [20], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39] construct e.g., saliency maps for classifiers or regressors [40] which highlight areas of the input that were particularly important for the classification.

A different approach to explain a neural network is given by providing counterfactuals to the original inputs [41], [42], [43]. These are realistic-looking images which are semantically close to the original but differ in distinct features so that their classification matches the desired target class, cf. Fig. 1. Counterfactuals aim to answer questions like "Why was this input classified as A and not as B?" or "What would need to change in the input so that it is no longer classified as A but instead as B?" [41], [42], [43] and thereby provide an explanation for the classifier. Unlike attribution methods, counterfactuals do not provide a relevance map, but an image that is similar to the original input and serves as a kind of counter example or
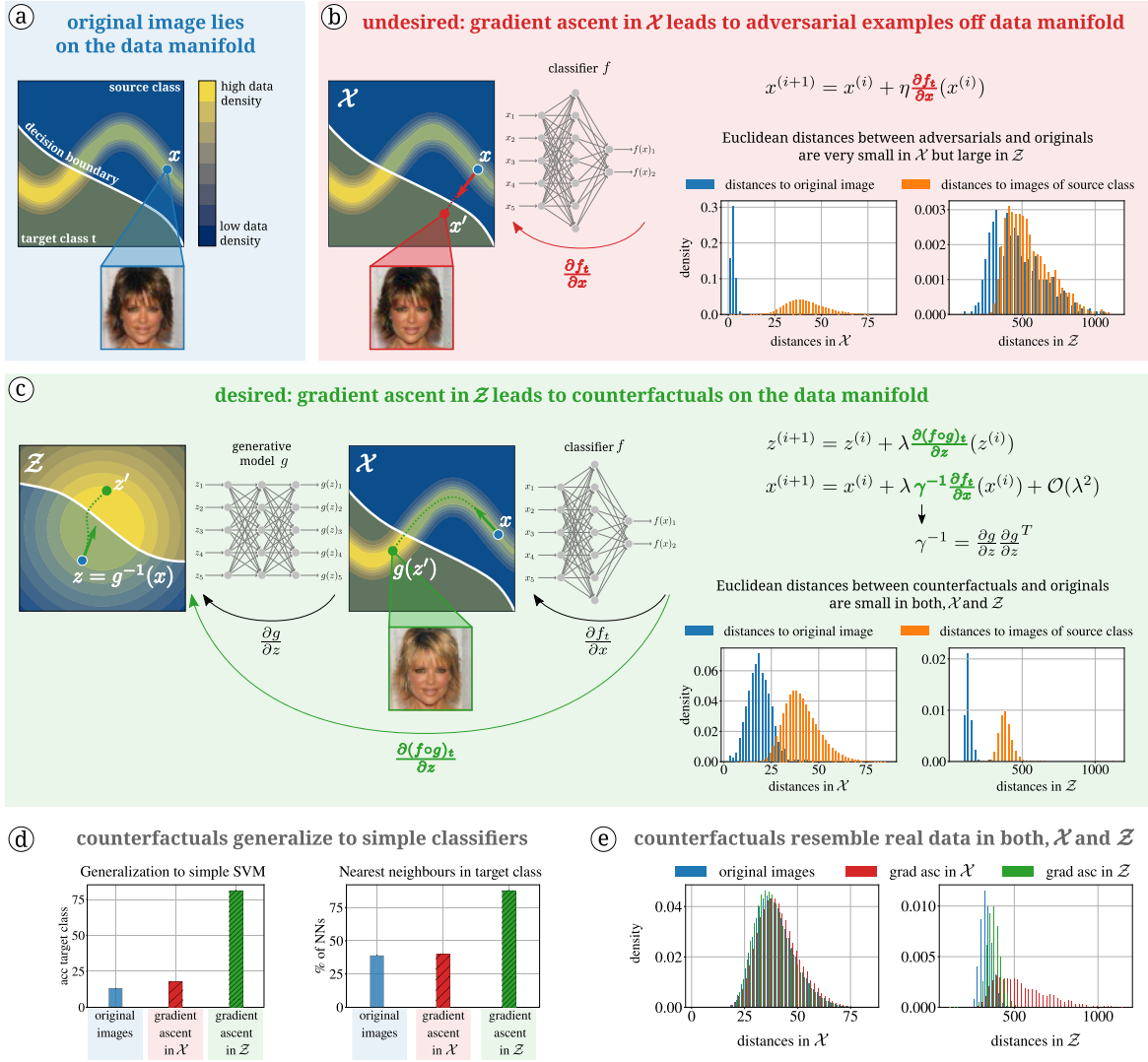
Fig. 2. (a) Image data usually lies on a lower dimensional data manifold, which is embedded in high dimensional space. We want to know what image features would have to change so that the classification flips. (b) If we follow the gradient of our target class with respect to the input $\frac{\partial f_t}{\partial x}$ the prediction flips but the resulting image is an adversarial example that looks indistinguishable from the original for a human observer. The changes to the original image are not semantic, but are limited to specific noise. The euclidean difference between adversarial and original is therefore very small when measured in $\mathcal{X}$ but large when measured in $\mathcal{Z}$, cf. Fig. 14. (c) We use to the normalizing flow g to obtain the latent space representation $z = g^{-1}(x)$ of our original image x. We then perform gradient ascent in the latent space $\mathcal{Z}$. The prediction flips, but this time the resulting image is a counterfactual. The changes to the original image are semantic. The euclidean difference between counterfactual and original is small when measured in $\mathcal{X}$ and $\mathcal{Z}$, cf. Fig. 14. (d) Left: Quantitative evaluations show that counterfactuals generalize to simple classifiers in contrast to adversarial examples. Right: Nearest neighbours of counterfactuals are mainly of the target class in contrast to NNs of adversarials. We show results for the CelebA data set. (e) Histograms for euclidean distances to training images for original images, adversarial examples and counterfactuals are indistinguishable when measuring the distances in the input space $\mathcal{X}$. When measuring the distances in the latent space $\mathcal{Z}$ we see that adversarial examples have larger distances. This confirms the hypothesis that adversarial examples lie off the data manifold. We show results for the CelebA data set.

hypothetical alternative for the original prediction. For certain applications, counterfactuals can yield insights that attribution methods cannot: for example, if we aim to classify circles and squares in a data set where all circles are twice as large as squares then the classifier could pick up on the size of the depicted object instead of its shape. In this situation, attribution maps would not be able to detect the size-correlation as they can only highlight existing features. In contrast, counterfactuals would show larger squares and smaller circles revealing that the classifier exploits a spurious correlation.

It is important to emphasize that the counterfactual is required to be a realistic sample from the data distribution in order to elucidate the behaviour of the network on the data. This requirement poses the greatest practical challenge to computing counterfactuals since naively optimizing the output of the network with respect to the input via gradient ascent yields adversarial examples [44] which essentially add a small amount of noise to the original input, as illustrated in the example given in Fig. 1. This behavior can be understood using the manifold hypothesis: the images are assumed to lie on a low dimensional manifold embedded in the high dimensional input space, cf. Fig. 2(a), that can be learned using a generative model. The gradient ascent algorithm then walks in a direction orthogonal to the decision boundary which is with high probability also orthogonal to

the data manifold, resulting in a small perturbation which is not semantic, as illustrated in Fig. 2(b). This is acceptable for adversarial examples as they aim to flip the classification by imperceptibly perturbing the input which may then also lie off the data manifold. In contrast, counterfactuals are supposed to facilitate the interpretation of the classifier's decision by a human operator. As such, unstructured, noisy, and therefore uninterpretable perturbations are unhelpful for this task, i.e., the change of the image must take a semantically meaningful form.

We propose to use insights from the mathematical discipline of differential geometry to mitigate this problem. Differential geometry can be understood as analysis on curved (hyper-) surfaces and thus provides the appropriate tools to study gradients on the data manifold. It has been valuable for the field of ML in general [45], [46], [47] and XAI specifically [48], [49], [50]. A cornerstone of differential geometry is the idea that geometric quantities can be described equivalently in different coordinate systems. However, not all coordinate systems are equally useful in practice. This phenomenon is ubiquitous in physics: for instance, the mathematical expressions governing planetary motions greatly simplify in a heliocentric (sun-centered) coordinate system as opposed to a geocentric (earth-centered) coordinate system. In heliocentric coordinates, the planetary motions can be described by simple Kepler ellipses and thus the relevant degrees of freedom are easily recognizable. In contrast, the orbits are very complicated in geocentric coordinates as can be observed in the night sky. As a result, physical intuition and interpretations are much easier to deduce in heliocentric coordinates. Similarly, we attribute the difficulty to construct counterfactuals by optimizing the output of a neural network classifier with respect to its input as in Fig. 2(b) to the poor choice of coordinates $\mathcal{X}$ in the input space given by the raw image data. In contrast, in a suitably chosen coordinate system $\mathcal{Z}$, the data manifold would extend more evenly in all directions, allowing for an optimization that stays on the data manifold and thereby producing a counterfactual that has been changed *semantically* when compared to the original. In order to find such a coordinate transformation (called a *diffeomorphism* in differential geometry) between $\mathcal{X}$ and $\mathcal{Z}$, we use a normalizing flow trained on the image data set under consideration. Since the flow is by construction bijective and differentiable with a differentiable inverse, it satisfies the technical conditions for a diffeomorphism in differential geometry. Furthermore, the base distribution of the flow is fixed to be a univariate Gaussian and hence free of pathological directions. Moreover, this change of coordinate system will lead to no information loss which is in stark contrast to existing methods for generating counterfactuals.

In our method, the counterfactual is computed by taking the gradient in the gradient ascent update with respect to the representation in the base space of the normalizing flow as opposed to the input of the classifier (Fig. 2(b)). This method comes with rigorous theoretical guarantees and we refer to it as *diffeomorphic counterfactuals*. In particular, we show that this introduces a metric into the update step which shrinks the gradient in directions orthogonal to the data manifold. Furthermore, we propose two separate methods which only *approximately* lead to a diffeomorphism. While these approximate methods come

with a lower level of theoretical guarantees, and can, in practice, lead to some information loss, they can be scaled easily to very high-dimensional datasets, as we demonstrate experimentally. We refer to these methods as *approximate diffeomorphic counterfactuals*. We theoretically prove that these methods also stay on the data manifold under suitable assumptions. Our theoretical analysis therefore provides a unified mathematical framework for the application of generative models in the context of counterfactuals. Importantly, we can not only optimize the output of a classifier network on the data manifold in this manner, but also that of a regressor.

This analysis is supported by our experimental results for various application domains, such as computer vision and medical radiology and a number of architectures for classifiers, regressors and generative models. Note that we lay emphasis on using quantitative metrics — as opposed to only qualitative analysis — to evaluate the proposed methods; some quantitative results are exemplified in Fig. 2(d) and (e).

The main contributions of our work are as follows:
1) We outline a simple and theoretically principled framework for the generation of counterfactual explanations using generative models.
2) We prove rigorously using differential geometry that the resulting counterfactuals lie on the data manifold for well-trained generative models.
3) We show that the proposed framework can be straightforwardly extended to a broad class of tasks and generative model architectures and demonstrate this in detailed numerical experiments.

The paper is structured as follows: in Section II, we introduce the proposed methods. Specifically, we will introduce diffeomorphic explanations in Section II-C and the approximate versions thereof in Section II-D. We then analyse the proposed methods theoretically using Riemannian differential geometry in Section III. This is followed by Section IV which provides a detailed experimental analysis of our proposed methods. In Section VI, we give an extensive discussion of related work. The code for a toy example and our main experiments is publicly accessible.[1]

II. METHODS

In this section, we will introduce in detail our novel diffeomorphic and approximately diffeomorphic counterfactuals. For this, we will start by reviewing the basics of counterfactual explanations and then present our two proposed methods.

A. Counterfactual Explanations

Consider a classifier[2] $f : \mathcal{X} \to \mathbb{R}^C$ which assigns to an input $x \in \mathcal{X}$ the probability $f(x)_c$ to be part of class $c \in \{1, \ldots, C\}$. Counterfactual explanations of the classifier f provide minimal deformations $x' = x + \delta x$ such that the prediction of the classifier is changed.

---

[1]https://github.com/annahdo/counterfactuals
[2]In Appendix A, available online, we provide a list of the mathematical symbols used as a convenience for the reader.

In many cases of practical relevance, the data lies approximately on a submanifold $\mathcal{D} \subset \mathcal{X}$ which is of significantly lower dimensionality $N_{\mathcal{D}}$ than the dimensionality $N_{\mathcal{X}}$ of the input space $\mathcal{X}$. This is known as the manifold hypothesis in the literature (see e.g., [51]). For counterfactual explanations, as opposed to adversarial examples, we are interested in deformations x' which lie on the data manifold. Additionally, we require the deformations to the original data to be minimal, i.e., the perturbation δx should be as small as possible. cf. (3) below. The relevant norm is however measured along the data manifold and not calculated in the input space. For example, a slightly rotated number in an MNIST image may have large pixel-wise distance but should be considered an infinitesimal perturbation of the original image.

We mathematically formalize the manifold hypothesis by assuming that the data is concentrated in a small region of extension δ around $\mathcal{D}$. As we will show in Section III, this implies that the support S of the data density p is a product manifold

$$S = \mathcal{D} \times \mathcal{I}_{\delta_1} \times \cdots \times \mathcal{I}_{\delta_{N_{\mathcal{X}}-N_{\mathcal{D}}}}, \tag{1}$$

where $\mathcal{I}_\delta = (-\frac{\delta}{2}, \frac{\delta}{2})$ is an open interval of length δ (with respect to the euclidean distance on the input space $\mathcal{X}$). We assume that δ is small, i.e., the data lies approximately on the low-dimensional manifold $\mathcal{D}$ and thus fulfills the manifold hypothesis. We can think of the $\mathcal{I}_\delta$ as arising from the inherent noise in the data.

Furthermore, we define the set of points in S classified with confidence $\Lambda \in (0, 1)$ as class $t \in \{1, \ldots, C\}$ by

$$S_{t,\Lambda} = \{x \in S \mid t = \operatorname*{argmax}_j f_j(x) \text{ and } f_t(x) > \Lambda\}. \tag{2}$$

A *counterfactual* $x' \in \mathcal{X}$ for class t of the original sample $x \in \mathcal{X}$ then is the closest point to x in $S_{t,\Lambda}$,

$$x' \in S_{t,\Lambda} \quad \text{and} \quad \operatorname*{argmin}_y d_\gamma(x, y) = x', \tag{3}$$

where $d_\gamma(x', x)$ is the distance computed by the Riemannian metric γ on S (which is induced from the flat metric by the diffeomorphism given by the generative model). We will review the necessary concepts of Riemannian geometry in Section III-A.

B. Generation of Counterfactuals

Often, counterfactuals are generated by performing gradient ascent in the input space $\mathcal{X}$, see [42] for a recent review on counterfactuals. More precisely, for step size η and target class t, one performs the gradient ascent step

$$x^{(i+1)} = x^{(i)} + \eta \frac{\partial f_t}{\partial x}(x^{(i)}) \tag{4}$$

until the classifier has reached a threshold confidence Λ, i.e., $f(x^{(i+1)})_t > \Lambda$. The resulting samples will however often not lie on the data manifold and differ from the original image x only in added unstructured noise rather than in an interpretable and semantically meaningful manner. Especially when applied to high dimensional image data such samples are usually referred to as adversarial examples and not counterfactuals. The reason for the noisy gradients is that the classifier is trained only on
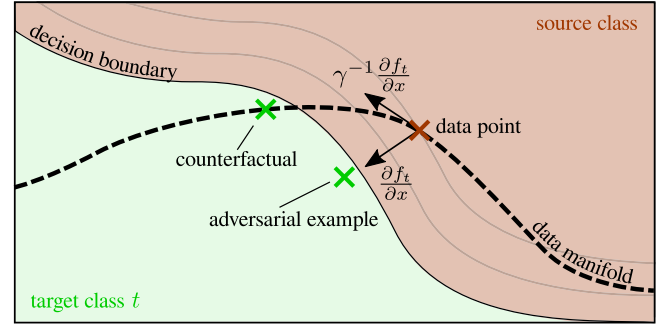

Fig. 3. When the gradient ascent optimization of the target class is performed in the input space of the classifier, one leaves the data manifold and obtains an adversarial example. If instead the gradient ascent is performed in the latent space of a generative model, one stays on the data manifold, resulting in a counterfactual example.

Algorithm 1: Generating Counterfactuals.

Require: $x, f, g, g^{-1}, t, \Lambda, \lambda, N$
1: $z \leftarrow g^{-1}(x)$
2: **for** i in range(N) **do**
3: $\quad \nabla_z \leftarrow \frac{\partial (f \circ g)_t}{\partial z}$
4: $\quad z \leftarrow$ optimizer.step(λ, ∇_z)
5: $\quad$ **if** $f(g(z))_t > \Lambda$ **then**
6: $\quad\quad$ **return** $g(z)$
7: $\quad$ **end if**
8: **end for**
9: **return** None

Note: x is the input for which we desire to find a counterfactual explanation, f the predictive model, g the generative model, g^{-1} the (approximate) inverse of g, t the target class, Λ the target confidence, λ the learning rate and N the maximum number of update steps. If the target confidence could not be reached after N steps, the algorithm returns None.

---

the data manifold, so gradient directions orthogonal to the data manifold are ill-specified (e.g., [52]).

We therefore propose to estimate the counterfactual x' of the original data point x by using a diffeomorphism $g : \mathcal{Z} \to S$. We then perform gradient ascent in the latent space $\mathcal{Z}$, i.e.,

$$z^{(i+1)} = z^{(i)} + \lambda \frac{\partial (f \circ g)_t}{\partial z}(z^{(i)}) \tag{5}$$

with step size $\lambda \in \mathbb{R}_+$. This has the important advantage that the resulting counterfactual will lie on the data manifold. Furthermore, since we consider a diffeomorphism g, and thus in particular a bijective map, no information will be lost by considering the classifier $f \circ g$ on $\mathcal{Z}$ instead of the original classifier f on the data manifold S, i.e., there exists a unique $z = g^{-1}(x) \in \mathcal{Z}$ for any $x \in S$. We show pseudo code for our approach in Algorithm 1.

As illustrated in Fig. 3, gradient ascent in $\mathcal{X}$ and $\mathcal{Z}$ are well-suited to generate adversarial examples and counterfactuals, respectively.

For regression tasks there is no explicit decision boundary, but we can still follow the the same algorithm by directly

maximizing (or minimizing) the output r of regressor $f(x)$ until we reach the desired target regression value.

C. Novel Method 1: Diffeomorphic Counterfactuals

We propose to model the map g by a normalizing flow and will refer to the corresponding modified data x' as diffeomorphic counterfactuals in the following.

Specifically, a flow g is an invertible neural network which equips, by the change-of-variable theorem, the input space $\mathcal{X}$ with a probability density

$$q(x) = q_Z(g^{-1}(x)) \left| \det \frac{\partial z}{\partial x} \right|, \qquad (6)$$

where q_Z is a simple base density, such as a univariate normal density, on the latent space $\mathcal{Z}$. The flow can be trained by maximum likelihood, i.e., by minimizing

$$\mathrm{KL}(p|q) = -\mathbb{E}_{x \sim p} \log q(x) + \mathrm{const.}$$

$$\approx -\frac{1}{N} \sum_{i=1}^{N} \log(q(x_i)) + \mathrm{const.}, \qquad (7)$$

where $x_i \sim p$ are samples from the data density p. Since the flow is bijective on the entire input space $\mathcal{X}$, it will, in particular, be bijective on the data manifold $S \subset \mathcal{X}$. Furthermore, we will also rigorously show in Section III-C that a well-trained flow maps (to very good approximation) only to the data manifold, i.e., $g(\mathcal{Z}) \approx S$.

Therefore, flows guarantee that no information is lost when performing gradient ascent in the latent space $\mathcal{Z}$ and also ensure that the resulting counterfactuals lie on the data manifold S. Indeed, the flow can be understood as inducing a certain coordinate change of the input space $\mathcal{X}$ which is particularly suited for the generation of counterfactuals.

D. Novel Method 2: Approximate Diffeomorphic Counterfactuals

While the method of the last section is very appealing as it comes with strong guarantees, it may be challenging to scale to very high-dimensional data sets. This is because flows have a very large memory footprint on such datasets as each layer has the same dimensionality as the data space $\mathcal{X}$ to ensure bijectivity. We therefore posit an alternative method, called approximate diffeomorphic counterfactuals, which comes with less rigorous theoretical guarantees, but can scale better to very high-dimensional data. Specifically, we propose two varieties of approximate diffeomorphic counterfactuals:

Autoencoder-Based: The reconstruction loss of an autoencoder (AE), i.e.,

$$\mathcal{L} = \mathbb{E}_{x \sim p} ||g(e(x)) - x||^2, \qquad (8)$$

with encoder $e : \mathcal{X} \to \mathcal{Z}$ and generator $g : \mathcal{Z} \to \mathcal{X}$ is minimized if the encoder is the inverse of the generator on the data manifold S, i.e.,

$$e|_S = g^{-1}|_S. \qquad (9)$$

This implies, in particular, that $g(\mathcal{Z}) = S$ if $\dim(\mathcal{Z}) = \dim(S)$. As for normalizing flows, the image of the autoencoder is the data manifold if the model has been perfectly trained. However, an autoencoder will only be invertible on the data manifold in this perfect training limit and if the latent space $\mathcal{Z}$ has the same dimension as the data space S. This is in contrast to normalizing flows which are invertible on all of $\mathcal{X}$ by construction. As a result, the autoencoder will necessarily lead to loss of information unless the model is perfectly trained and the latent space dimensionality perfectly matches the dimension of the data.

GAN-Based: Generative Adversarial Networks (GANs) consist of a generator $g : \mathcal{Z} \to \mathcal{X}$ and a discriminator $d : \mathcal{X} \to \{0, 1\}$. Training then proceeds by minimizing a certain minimax loss, see [53] for details. It can be shown that the global minimizer of this loss function ensures that samples of the optimal generator g are distributed according to the data distribution, i.e.,

$$g(z) \sim p \quad \text{for} \quad z \sim q_Z. \qquad (10)$$

We refer to Section IV-A of [53] for a proof. However, the optimal generator g is not necessarily bijective on the data manifold. This implies that even for a perfectly trained GAN, there may not exist a unique $z \in \mathcal{Z}$ for a given data sample $x \in \mathcal{X}$ such that $x = g(z)$. Furthermore, there is no manifest mechanism to obtain the corresponding latent sample $z \in \mathcal{Z}$ for a given input $x \in \mathcal{X}$. This is in contrast to normalizing flows and autoencoders, since, for these generative models, the inverse map $g^{-1} : \mathcal{X} \to \mathcal{Z}$ is either explicitly or approximately known, respectively.

However, there is an extensive literature for GAN inversion, see [54] for a recent review. For a given generator g and data sample $x \in \mathcal{X}$, these methods aim to find a latent vector $z \in \mathcal{Z}$ such that $x \approx g(z)$. This is often done by minimizing the difference between the activations of an intermediate layer of some auxiliary network, i.e.,

$$z = \underset{\hat{z} \in \mathcal{Z}}{\mathrm{argmin}} \, ||h(g(\hat{z})) - h(x)||. \qquad (11)$$

For example, h can be chosen to be an intermediate layer of an Inception network [55] trained on samples from the data density p. Note that these inversion methods do not come with rigorous guarantees as the optimization objective is non-convex and it is unclear whether the values of the intermediate layer activations are sufficient to distinguish different inputs.

III. THEORETICAL ANALYSIS

In this section, we employ tools from differential geometry to show that for well-trained generative models, the gradient ascent update (5) in the latent space $\mathcal{Z}$ does indeed stay on the data manifold, as confirmed by the experimental results presented in Section IV. Intuitively, since in (5) we take small steps in $\mathcal{Z}$, where the probability distribution is, for example, a normal with unit variance, we do not leave the region of high probability in the latent space and hence stay in a region of high probability also in $\mathcal{X}$.

We prove this statement for the case of diffeomorphic counterfactuals, i.e., for normalizing flows, and – under stronger

assumptions – also for approximate diffeomorphic counterfactuals, i.e., for autoencoders and generative adversarial networks.

A. Differential Geometry

In this section, we briefly introduce the most fundamental notions of differential geometry used in our discussion further down. For a comprehensive textbook, see e.g., [56].

Differential geometry is the study of smooth (hyper-)surfaces. The central notion of this branch of mathematics is that of an n-dimensional (differentiable) manifold $\mathcal{M}$ which is equipped with coordinate functions $x^\mu : \mathcal{M} \to \mathbb{R}^n$ (so-called *charts* which are assembled into an *atlas*). These coordinates allow for explicit calculations in $\mathbb{R}^n$, but geometric objects (tensors) are independent of the chosen coordinates and transform in well-defined ways under changes of coordinates. Such a change of coordinates can be interpreted as a differentiable bijection $\phi : \mathcal{M} \to \mathcal{M}$ whose inverse is also differentiable, a so-called *diffeomorphism*.

At each point $p \in \mathcal{M}$, we attach an n-dimensional vector space $T_p\mathcal{M}$, the *tangent space* at p. Coordinates x^μ induce a basis in $T_p\mathcal{M}$ and we will denote the components of $v \in T_p\mathcal{M}$ in this basis by v^μ. Under a diffeomorphism ϕ, the components of v transform as

$$v^\alpha = \sum_{\mu=1}^{n} \frac{\partial \phi^\alpha}{\partial x^\mu} v^\mu = \frac{\partial \phi^\alpha}{\partial x^\mu} v^\mu \,, \qquad (12)$$

where in the second equality, we introduced the Einstein sum convention which implies sums over repeated upper and lower indices (the index in the denominator of the differential counts as a lower index). We will use this sum convention in the following to streamline the notation.

To capture the notion of distance (and curvature) on $\mathcal{M}$, a *metric tensor* $\gamma(p) : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ is used. The metric defines a canonical isomorphism between $T_p\mathcal{M}$ and its dual space $T_p^*\mathcal{M}$ by $v^\star = \gamma(p)(v, \cdot)$ and $v = \gamma^{-1}(p)(v^*, \cdot)$. Following the usual convention in the general relativity literature, we use lower indices v_μ to denote the components of the dual vector v^*. This implies that in components the isomorphism reads

$$v_\mu = \gamma_{\mu\nu} v^\nu \quad \text{and} \quad v^\mu = \gamma^{\mu\nu} v_\nu \,, \qquad (13)$$

where the sums over ν are implied and we introduced $\gamma^{\mu\nu}$ for the inverse of $\gamma_{\mu\nu}$, the components of γ in the basis induced by the coordinates x^μ. In short, contraction with the metric is used to raise and lower indices.

Given a metric, it is natural to consider shortest paths between points on $\mathcal{M}$. The corresponding curves are called *geodesics*. If the length of the tangent vector of a geodesic σ is constant (as measured by the metric) along σ, the geodesic is *affinely-parametrized*. Importantly, the notion of an affinely parametrized geodesic is coordinate independent and can therefore itself be used to construct coordinates on $\mathcal{M}$, as we will see below.

B. Mathematical Setup

In order to analyze the gradient ascent (5) in the latent space $\mathcal{Z}$, we define in this section the necessary manifolds and coordinates.

As above, let $\mathcal{X}$ be an $N_\mathcal{X}$-dimensional manifold which is the input space of the classifier $f : \mathcal{X} \to \mathbb{R}^C$ with C classes. An implementation of the classifier corresponds to a function on $\mathbb{R}^{N_\mathcal{X}}$ and we denote the coordinates on $\mathcal{X}$ in which our classifier is given by x^α. These coordinates could e.g., be suitably normalized pixel values. We furthermore use an $N_\mathcal{Z}$-dimensional manifold $\mathcal{Z}$ as the latent space for our generative model $g : \mathcal{Z} \to \mathcal{X}$. For GANs and AEs, we typically have $N_\mathcal{Z} < N_\mathcal{X}$ and for normalizing flows $N_\mathcal{Z} = N_\mathcal{X}$. In the latter case we have moreover $\mathcal{X} = \mathcal{Z}$ and g bijective with differentiable inverse implying that g is a diffeomorphism. Similarly to the classifier, also the generative model is implemented in specific coordinates on $\mathcal{Z}$ which we denote by z^a.

We equip $\mathcal{Z}$ with a flat euclidean metric δ_{ab}. Then, the generative model g induces an inverse metric $\gamma^{\alpha\beta}$ on $g(\mathcal{Z})$ by

$$\gamma^{\alpha\beta} = \delta^{ab} \frac{\partial g^\alpha}{\partial z^a} \frac{\partial g^\beta}{\partial z^b} \,. \qquad (14)$$

In the case of $N_\mathcal{Z} < N_\mathcal{X}$, γ is singular. This metric is the crucial new ingredient when performing the gradient ascent update in the latent space (5) as opposed to in the input space (4), as the following calculation shows.

One step of gradient ascent in the latent space $\mathcal{Z}$ is given by the image under g of the update step (5). In x^α coordinates and to linear order in the learning rate λ, it is given by

$$g^\alpha(z^{(i+1)}) = g^\alpha(z^{(i)}) + \lambda \frac{\partial g^\alpha}{\partial z^a} \frac{\partial (f \circ g)_t}{\partial z^a} + \mathcal{O}(\lambda^2)$$

$$= g^\alpha(z^{(i)}) + \lambda \frac{\partial g^\alpha}{\partial z^a} \frac{\partial g^\beta}{\partial z^a} \frac{\partial f_t}{\partial x^\beta} + \mathcal{O}(\lambda^2)$$

$$= g^\alpha(z^{(i)}) + \lambda \gamma^{\alpha\beta} \frac{\partial f_t}{\partial x^\beta} + \mathcal{O}(\lambda^2) \,. \qquad (15)$$

If we start from the same points, $x^{(i)} = g(z^{(i)})$, the difference between gradient ascent in latent space (5) and input space (4) is just given by the contraction of the gradient of f with respect to x with the inverse induced metric $\gamma^{\alpha\beta} = \frac{\partial g^\alpha}{\partial z^a} \frac{\partial g^\beta}{\partial z^a}$. Hence, in order to understand why the prescription (5) stays on the data manifold, we will in the following investigate the properties of γ for the case of well-trained generative models.

Before returning to γ, we will first discuss the structure of the data. The probability density of the data on $\mathcal{X}$ is denoted by $p : \mathcal{X} \to \mathbb{R}$ and the probability density induced by g is denoted by $q : \mathcal{X} \to \mathbb{R}$. For q in x^α coordinates, we use the notation $q_x : \mathbb{R}^{N_\mathcal{X}} \to \mathbb{R}$. The data is characterized by $S = \text{supp}(p) \subset \mathcal{X}$ which becomes $S_x \subset \mathbb{R}^{N_\mathcal{X}}$ in x^α coordinates. We will assume that the data lives approximately on a submanifold $\mathcal{D} \subset S$ of $\mathcal{X}$ with dimension $N_\mathcal{D} \ll N_\mathcal{X}$. In relation to the dimension of our generative model, we assume that $N_\mathcal{D} \leq N_\mathcal{Z} \leq N_\mathcal{X}$. As a subset of $\mathcal{X}$ and in x^α coordinates, $\mathcal{D}$ will be denoted by $\mathcal{D}_x \subset \mathbb{R}^{N_\mathcal{X}}$. To capture that the data does not extend far beyond $\mathcal{D}$, we assume that S has euclidean extension $\delta \ll 1$, normal to $\mathcal{D}$ in
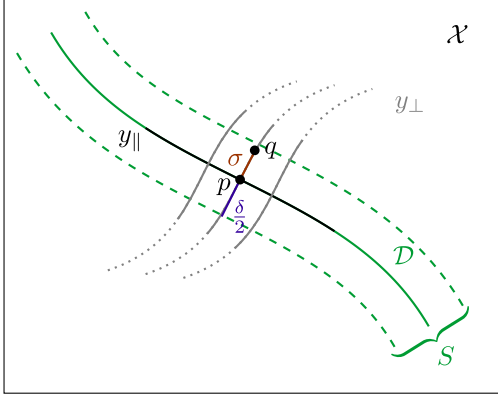
Fig. 4. Construction of the y^μ coordinates which are aligned with the data manifold $\mathcal{D}$.

x^α coordinates, i.e.,[3]

$$S_x = \left\{ x_\mathcal{D} + x_\delta \mid x_\mathcal{D} \in \mathcal{D}_x,\ x_\delta^\alpha \in \left(-\frac{\delta}{2}, \frac{\delta}{2}\right) \right\}. \quad (16)$$

Next, we will define coordinates in a neighborhood of $\mathcal{D}$ which separate the directions tangential and normal to $\mathcal{D}$ as illustrated in Fig. 4. Our construction is similar to the constructions of Riemannian and Gaussian normal coordinates, adapted for a submanifold of codimension larger than one. First, we choose coordinates $y_\parallel$ on $\mathcal{D}$ and, for each $p \in \mathcal{D}$, a basis $\{n_i\}$ for the tangent space $T_p\mathcal{D}_\perp$ of the normal to $\mathcal{D}$ at p. Following the usual construction of Riemannian normal coordinates, we assign coordinates to a point q in some neighborhood of $\mathcal{D}$ by constructing an affinely parametrized geodesic $\sigma : [0,1] \to \mathcal{X}$ which satisfies $\sigma(0) = p$ and $\sigma(1) = q$ and which has tangent vector $\sigma'(0) \in T_p\mathcal{D}_\perp$. The coordinates of q are then $y(q) = (y_\parallel(p), y_\perp) \in \mathbb{R}^{N_\mathcal{D}} \oplus \mathbb{R}^{N_\mathcal{X} - N_\mathcal{D}}$, where the ith component of $y_\perp$ is given by the ith component of $\sigma'(0)$ in the basis $\{n_i\}$. In a sufficiently small neighborhood around $\mathcal{D}$, we can find a unique basepoint $p \in \mathcal{D}$ and geodesic σ for every q.

One important aspect of this construction is that by rescaling the basis vectors $\{n_i\}$, we can rescale the components of $\sigma'(0)$.[4] This means we can rescale the $y_\perp$ coordinates arbitrarily and hence we can use this freedom to bound S in y coordinates by the same δ that appeared in (16),

$$S_y = \left\{ (y_\parallel, y_\perp) \in \mathbb{R}^{N_\mathcal{X}} \mid y_\parallel \in \mathcal{D}_y,\ y_\perp^i \in \left(-\frac{\delta}{2}, \frac{\delta}{2}\right) \right\}. \quad (17)$$

Furthermore, in $g(\mathcal{Z})$, we can choose the basis $\{n_i\}$ orthogonal with respect to the (singular) metric γ and obtain in some

---

[3]The form (16) restricts the slices $S_\perp(x_\mathcal{D})$ through S normal to $\mathcal{D}$ to be L_1 balls whose size is independent of $x_\mathcal{D}$. We make this restriction to simplify notation but the argument can straightforwardly be extended to arbitrary shapes of $S_\perp(x_\mathcal{D})$ by bounding it by an L_2 ball of radius $\delta/2$.

[4]Note that this does not change the parametrization of the geodesic, hence we still have $\sigma(0) = p$ and $\sigma(1) = q$.

neighborhood of $\mathcal{D} \cap g(\mathcal{Z})$

$$\gamma^{\mu\nu}(y) = \begin{pmatrix} \gamma_\mathcal{D}^{-1}(y) & & & \\ & \gamma_{\perp_1}^{-1} & & \\ & & \ddots & \\ & & & \gamma_{\perp_{N_\mathcal{X} - N_\mathcal{D}}}^{-1} \end{pmatrix}^{\mu\nu}. \quad (18)$$

Note that this form of the metric together with (17) implies in particular that S takes the product form mentioned in (1). In the following, we will show that for well-trained generative networks and thin data distributions (i.e., for small δ), $\gamma_{\perp_i}^{-1} \to 0$. To understand the consequences for the gradient ascent update step, consider (15) in y^μ coordinates

$$\gamma^{\alpha\beta} \frac{\partial f_t}{\partial x^\beta} = \frac{\partial x^\alpha}{\partial y^\mu} \gamma^{\mu\nu} \frac{\partial f_t}{\partial y^\nu}$$

$$= \frac{\partial x^\alpha}{\partial y_\parallel^\mu} \gamma_\mathcal{D}^{\mu\nu} \frac{\partial f_t}{\partial y_\parallel^\nu} + \frac{\partial x^\alpha}{\partial y_\perp^i} \gamma_{\perp_i}^{-1} \frac{\partial f_t}{\partial y_\perp^i}. \quad (19)$$

For $\gamma_{\perp_i}^{-1} \to 0$ and $\frac{\partial x}{\partial y_\perp}$ bounded, the second term vanishes and we arrive at

$$\gamma^{\alpha\beta} \frac{\partial f_t}{\partial x^\beta} \to \frac{\partial x^\alpha}{\partial y_\parallel^\mu} \gamma_\mathcal{D}^{\mu\nu} \frac{\partial f_t}{\partial y_\parallel^\nu} \quad (20)$$

and hence the orthogonal directions in the update step (15), leading away from the data manifold $\mathcal{D}$, are suppressed. Therefore, (15) produces counterfactuals instead of adversarial examples.

C. Diffeomorphic Counterfactuals

In this section, we show that for well-trained normalizing flows, the orthogonal components of the inverse metric $\gamma_{\perp_i}^{-1}$ vanish for thin data manifolds, as formalized in the following theorem.

Theorem 1: For $\epsilon \in (0, 1)$ and g a normalizing flow with Kullback–Leibler divergence $\mathrm{KL}(p, q) < \epsilon$,

$$\gamma_{\perp_i}^{-1} \to 0 \quad \text{as} \quad \delta \to 0$$

for all $i \in \{1, \ldots, N_\mathcal{X} - N_\mathcal{D}\}$.

The main argument of the formal proof given in Appendix B.1, available online, proceeds as follows: First, we show that a small Kullback–Leibler divergence implies that most of the induced probability mass lies in the support of the data distribution,

$$\int_{S_x} q_x(x) \mathrm{d}x > 1 - \epsilon. \quad (21)$$

Next, we write q_x as the pull-back of the latent distribution q_z under the flow g using the familiar change-of-variables formula for normalizing flows. In the y^μ coordinates introduced above, the resulting integral then factorizes according to the block-diagonal structure (18) of the metric with integration domain $[-\delta/2, \delta/2]$ for the $y_\perp^i$ directions. As $\delta \to 0$, the bound (21) can only remain satisfied if the associated metric component $\gamma_{\perp_i}$ diverges, implying that $\gamma_{\perp_i}^{-1} \to 0$.

Following the steps at the end of Section III-B, we see that this necessarily implies that the gradient ascent update (5) stays on the data manifold, since $\frac{\partial x}{\partial y_\perp}$ is constant (and therefore bounded) as $\delta \to 0$.
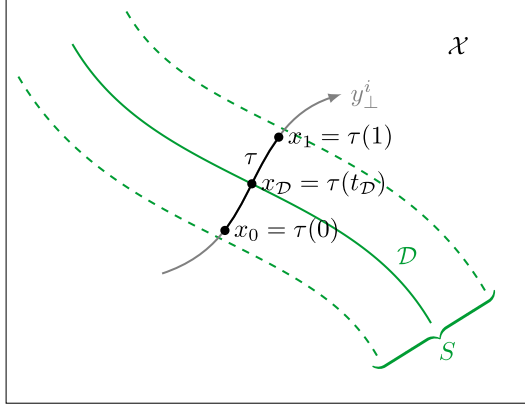
Fig. 5.   Construction of the curve τ used in Section III-D.

Fig. 6.   Gradient ascent in $\mathcal{X}$ leads to points that lie significantly off-manifold while gradient ascent in $\mathcal{Z}$ moves along the data manifold. The ground truth for different classes is depicted in orange (source class) and gray (target class).

D. Approximate Diffeomorphic Counterfactuals

We now present a theorem similar to Theorem 1 for the case of approximately diffeomorphic counterfactuals, i.e., for AEs and GANs, showing that these models can also be used to construct counterfactuals. This will however necessitate stronger assumptions since the generative model is in this case not bijective. In particular, we will assume that the generative model captures all of the data, i.e., that $\mathcal{D} \subset g(\mathcal{Z})$, implying that in y coordinates, although γ is singular for $N_{\mathcal{Z}} < N_{\mathcal{X}}$, the component $\gamma_{\mathcal{D}}$ is non-singular. Therefore, we split the $y_{\perp,i}$ directions into $N_{\mathcal{X}} - N_{\mathcal{Z}}$ singular directions and $N_{\mathcal{Z}} - N_{\mathcal{D}}$ non-singular directions. Since the inverse metric vanishes by definition in the singular directions, the theorem focuses on the non-singular directions and can then be stated as follows,

Theorem 2: If $g : \mathcal{Z} \to \mathcal{X}$ is a generative model with $\mathcal{D} \subset g(\mathcal{Z})$ and image $g(\mathcal{Z})$ which extends in any non-singular orthogonal direction $y_{\perp}^i$ outside of $\mathcal{D}$,

$$\gamma_{\perp_i}^{-1} \to 0$$

for $\delta \to 0$ for all non-singular orthogonal directions $y_{\perp}^i$.

The proof can be found in Appendix B.2, available online, and proceeds as follows: First, we construct a curve $\tau : [0, 1] \to \mathcal{Z}$ which cuts through S along the $y_{\perp}^i$-coordinate line and lies completely in $g(\mathcal{Z})$, as illustrated in Fig. 5. Then, the length $\mathcal{L}(\tau)$ of this curve (with respect to γ) computed in y^{μ}-coordinates is, for small δ, approximately given by

$$\mathcal{L}(\tau) \approx \sqrt{\gamma_{\perp_i}(x_{\mathcal{D}})} \left(x_{1,\perp}{}^i - x_{0,\perp}{}^i \right). \tag{22}$$

Bounding the difference by δ and using that $\mathcal{L}(\tau)$ is constant, yields the desired result. As in the case of Theorem 1 above, this implies again that the gradient ascent update (5) does not leave the data manifold as shown in (20).

IV. EXPERIMENTS

Equipped with our theoretical results, we are now ready to present our experimental findings.

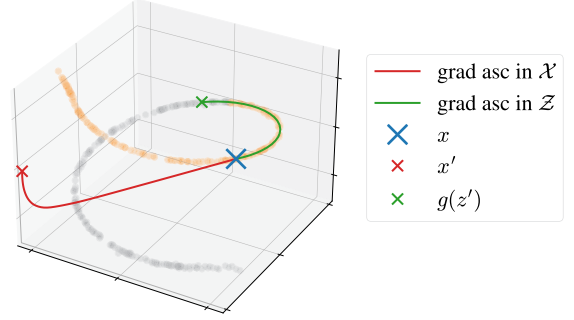We start by illustrating diffeomorphic explanations using a toy example in three-dimensional space. This allows us to directly visualize the data manifold and the trajectories of gradient ascent in $\mathcal{X}$ and $\mathcal{Z}$.

We then apply our diffeomorphic counterfactual method, using normalizing flows, to four different image data sets. We use MNIST, CelebA and CheXpert for classification tasks and the Mall data set for a regression task. We evaluate the results qualitatively and quantitatively. Furthermore we discuss approximate diffeomorphic counterfactuals, using VAEs and GANs, which allow us to consider high resolution data.

For all experiments, we use the same setup: We require a pretrained generator g and a pretrained classifier f. We start with a data point x from the test set that is predicted by the classifier f as belonging to the source class. We define target class t and target confidence Λ. To produce an adversarial example, we then update the original data point following the gradient in $\mathcal{X}$, $\frac{\partial f_t(x)}{\partial x}$, until we reach the desired target confidence. To produce a counterfactual we first project the original data point into the latent space of the generative model g by applying the inverse generative model $g^{-1}(x) = z$, or an appropriate approximation (for GANs). We then update the original latent representation z following the gradient in $\mathcal{Z}$, $\frac{\partial (f_t \circ g)(z)}{\partial z}$, until we reach the desired target confidence.

For more details on model configuration, training and hyperparameters we refer to Appendix C, available online.

A. Toy Example

We consider data uniformly distributed on a one-dimensional manifold, a helix, that is embedded in three-dimensional space and train a simple normalizing flow that approximates the data distribution. As illustrated in Fig. 6, we divide the data into two classes corresponding to the upper and the lower half of the helix and train a classifier.

We then generate counterfactuals by the gradient ascent optimization in input space $\mathcal{X}$ and in the latent space of the flow $\mathcal{Z}$, i.e., by using (4) and (5) respectively.

Starting from the original data point x, we observe that gradient ascent in $\mathcal{X}$ leads to points that lie significantly off data manifold S. In contrast to that, the updates of gradient ascent in the latent space $\mathcal{Z}$ follow a trajectory along the data manifold
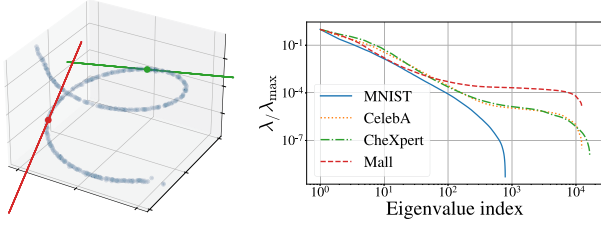
Fig. 7. *Left:* As expected from the theoretical analysis, the parallelepiped spanned by all three eigenvectors of the inverse induced metric scaled by the corresponding eigenvalues is to good approximation one-dimensional, i.e., of the same dimension as the data manifold, and tangential to it. *Right:* The Jacobians of the trained flows have a low number of large and a large number of small eigenvalues, suggesting that the images lie approximately on a low-dimensional manifold. Both axes are scaled logarithmically.

resulting in counterfactuals with the desired target classification which lie on the data manifold. We illustrate this in Fig. 6.

As we have an analytic description of the data manifold, we can reliably calculate the distances to the data manifold for all points found via gradient ascent in $\mathcal{X}$ or $\mathcal{Z}$. We compare 1000 successful optimizations (all optimizations reach the desired target confidence) in the input space $\mathcal{X}$ and latent space $\mathcal{Z}$. The median value for the distances to the data manifold when performing gradient ascent in $\mathcal{X}$ or in $\mathcal{Z}$ are 2.34 and 0.01 respectively (see also Fig. 17 in the appendix, available online). This intuitively and clearly illustrates the benefit of performing gradient ascent in the latent space $\mathcal{Z}$.

A non-trivial consequence of our theoretical insights is that we can infer the tangent space of each point on the data manifold from our flow g. Specifically, we perform a singular value decomposition of the Jacobian $\frac{\partial g}{\partial z} = U \Sigma V$ and rewrite the inverse induced metric as

$$\gamma^{-1} = \frac{\partial g}{\partial z} \frac{\partial g}{\partial z}^T = U \Sigma^2 U^T . \qquad (23)$$

As we saw in Section III, for data concentrated on an $N_{\mathcal{D}}$-dimensional data manifold $\mathcal{D}$ in an $N_{\mathcal{X}}$-dimensional embedding space $\mathcal{X}$, the inverse induced metric γ^{-1} has $N_{\mathcal{X}} - N_{\mathcal{D}}$ small eigenvalues. Furthermore, the eigenvectors corresponding to the large eigenvalues will approximately span the tangent space of the data manifold. For our toy example from Section IV-A, we can directly show the parallelepiped spanned by the three eigenvectors in three-dimensional space. Fig. 7 (left) indeed shows that the parallelepipeds are significantly contracted in two of the three dimensions making them appear as one dimensional lines. For the high dimensional image data sets, which are discussed in Section IV-B, we show the sorted eigenvalues, averaged over 100 random data points per data set, cf. Fig. 7 (right). Our experiments confirm the theoretical expectation that the large eigenvectors indeed span the tangent space of the manifold.

B. Image Classification and Regression

We now demonstrate applications of diffeomorphic counterfactuals to image classification in several domains.

1) Setup: We use MNIST [57], CelebA [58], CheXpert [59] (a data set of labeled chest X-rays) and a the Mall data set [60], [61], [62], [63] (a crowd-counting data set with video frames from a shopping mall with head annotations of pedestrians).

Classifiers/Regressors: We train a ten-class CNN on MNIST (test accuracy of 99%). For CelebA, we train a binary CNN on the blond attribute (test accuracy of 94%). For CheXpert, we train a binary CNN on the cardiomegaly attribute (test accuracy of 86%).

For the Mall data set, we train a U-Net [64] that outputs a probability map of the size of the image and a scalar regression value, which corresponds to the approximated number of pedestrians in the picture. Following the definitions by Ribera et al. [65], our trained U-Net reaches a RMSE for the head count of 0.63. When we run our gradient ascent algorithm, we aim to maximize/minimize merely the scalar regression value, i.e., the number of pedestrians.

Generative Models: For diffeomorphic counterfactuals, we choose a flow with RealNVP-type couplings [66] for MNIST and the Glow architecture [67] for CelebA, CheXpert and the Mall data set. For approximately diffeomorphic counterfactuals, we use a simple convolutional GAN (dcGAN) and convolutional VAE (cVAE) for MNIST and a progressive GAN (pGAN) [68] for CelebA.

Generation of Counterfactuals: We start from original data points x of the classes 'four' for MNIST, 'not blond' for CelebA and 'healthy' for CheXpert. We select the classes 'nine', 'blond', and 'cardiomegaly' as targets t for MNIST, CelebA, and CheXpert, respectively, and take the confidence threshold to be $\Lambda = 0.99$. For the Mall data set, we maximize the regression value r (threshold at $r = 10$) if few pedestrians were detected in the original image x and minimize the regression value (threshold at $r = 0.01$) if many pedestrians were detected in the original image x. We use Adam to optimize in $\mathcal{X}$ or $\mathcal{Z}$ until the confidence threshold Λ for the target class t is reached.

As discussed in Section II-D, GANs, which we use for approximately diffeomorphic counterfactuals, do generically not require an encoder during the training process. We apply GAN inversion methods to find an encoding of the source image. Specifically, for the low-dimensional MNIST data set, we find the latent representation z by minimising the euclidean norm between the decoded latent representation $g(z)$ and the original image x. To find the initial latent representation for more high-dimensional datasets, we use HyperStyle [69] GAN-inversion techniques.

2) Qualitative Analysis: Our diffeomorphic counterfactuals produced by the normalizing flows indeed show semantically meaningful deformations in particular when compared to adversarial examples produced by gradient ascent in the data space $\mathcal{X}$.

We show examples in Fig. 8. The counterfactuals resemble images from the data set that have the target class as the ground truth label. At the same time the counterfactuals are similar to their respective source images with respect to features that are irrelevant for the differentiation between source and target class.

For MNIST, the stroke width and the writing angle remain unchanged in the counterfactuals while the gap in the upper
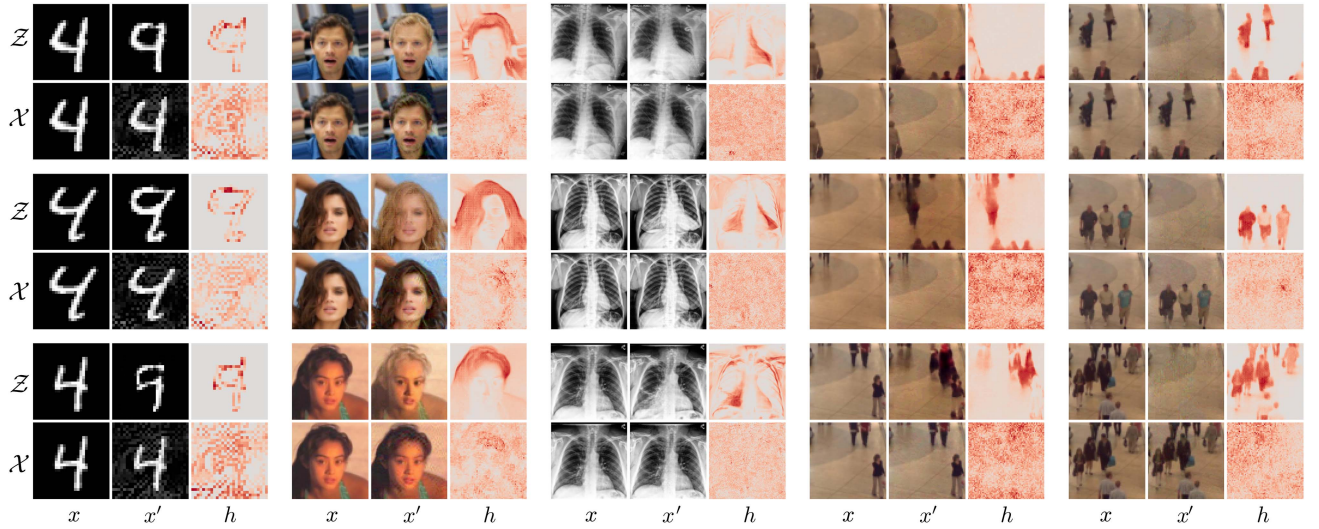
Fig. 8. Counterfactuals for MNIST ('four' to 'nine'), CelebA ('not-blond' to 'blond'), CheXpert ('healthy' to 'cardiomegaly'), Mall ('few' to 'many') and Mall ('many' to 'few'). Columns of each block show original image x, counterfactual x', and difference h for three selected datapoints. First row of each block is our diffeomorphic counterfactuals, i.e., obtained by gradient ascent in $\mathcal{Z}$ space. Second row of each block is standard gradient ascent in $\mathcal{X}$ space. Heatmaps h show the difference $|x - x'|$ summed over color channels.
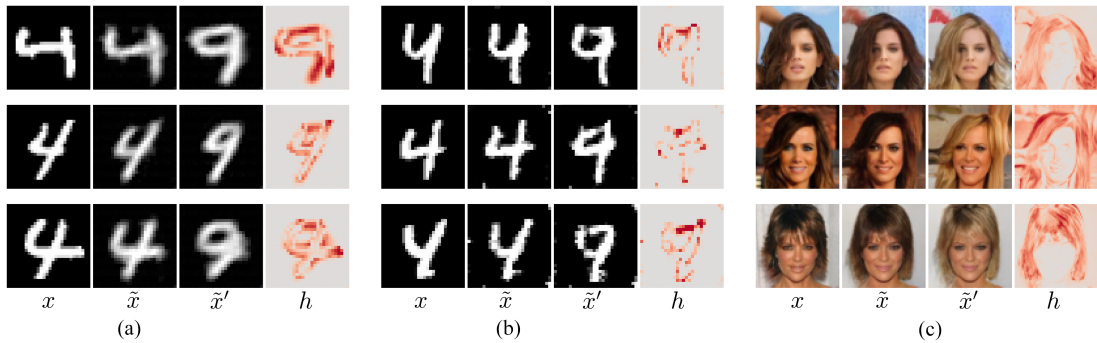

Fig. 9. Counterfactuals for cVAE on MNIST (a), dcGAN on MNIST (b) and pGAN on CelebA (c). Columns of each block show original image, decoded latent representation of original, counterfactual and absolute difference $|\tilde{x} - \tilde{x}'|$ summed over color channels.

part of the 'four' changes to the characteristic upper loop of the 'nine'.

For CelebA, the changes in the counterfactuals are focused on the hair area as evident from the heatmaps. Facial features and background stay (approximately) constant.

The counterfactuals for the CheXpert data set mostly brighten the pixels in the central region of the picture leading to the appearance of an enlarged heart. The other structures in the image remain mostly constant.

Also for pictures taken from the Mall data set, we observe that the counterfactuals remain close to the original images. When maximizing the regression value, pedestrians are generated at the picture's edge or appear around darker areas in the original image. When minimizing pedestrians, we observe that the counterfactuals reproduce the darker parts of the floor and lines between the tiles.

Results for approximately diffeomorphic counterfactuals are shown in Fig. 9 in the middle and right block. The dcGAN on MNIST produces some random pixel artifacts, but the generated images are sharper than those produced by the cVAE. For the

CelebA images generated with pGAN, we see that the decoded optimized latent representation of the original image deviates slightly from the original. This is especially visible if the composition is not typical (arm is not properly reproduced in the first row) or the background is highly structured (second row). For the approximate diffeomorphic counterfactuals, we observe even larger changes in the background. This may be attributed to the imperfect inversion process and the quality of the pGAN, i.e., the fact that the diffomorphism is only approximate and not exact. To demonstrate that approximate diffeomorphic explanations can scale to very high-dimensional data, we use a pretrained StyleGAN [70], [71] for images of resolution 1024×1024 from the CelebA-HQ data set [68]. Note that the StyleGAN can also be used at the same resolution even for the more diverse ImageNet dataset if required, see [72]. In order to use the same classifier as before, we downscale the images to 64×64 resolution before using them as input to the classifier. As demonstrated by Fig. 10, approximate diffeomorphic counterfactuals lead to semantically meaningful and interpretable results even on these very high-dimensional data sets. More examples for diffeomorphic-
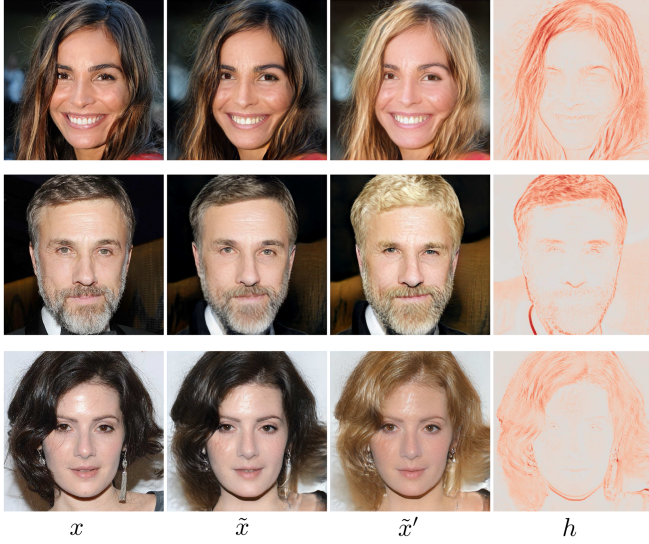
Fig. 10. Counterfactuals generated with HyperStyle and Celeba-HQ. Columns show original, decoded latent representation, counterfactual and absolute difference $|\tilde{x} - \tilde{x}'|$ summed over color channels.

Fig. 11. *Left:* accuracy with respect to the target class k generalizes better to SVM for diffeomorphic counterfactuals. *Right:* regression values for oracle are closer to target values for $\mathcal{Z}$-based counterfactuals (bars show means and errors denote one standard deviation).

Fig. 12. Head locations for pedestrians in counterfactuals and adversarial examples when maximizing pedestrians (upper two rows) and minimizing pedestrians (lower two rows). The original U-Net is fooled by the adversarial examples, leading to false positives (second row) and false negatives (forth row) when detecting pedestrians. The oracle U-Net generalizes to the diffeomorphic counterfactuals found by gradient ascent in $\mathcal{Z}$ (odd rows) but not to the adversarial examples found by gradient ascent in $\mathcal{X}$ (even rows).

and approximately diffeomorphic counterfactuals are printed in Appendix D, available online.

3) Quantitative Analysis: To quantitatively assess the quality of our counterfactuals, we use several measures, as detailed in the following.

Oracle: We train a 10-class SVM on MNIST (test accuracy of 92%) and binary SVMs on CelebA (test accuracy of 85%) and CheXpert (test accuracy of 70%). The counterfactuals found by performing gradient ascent in the base space of the flow generalize significantly better to these simple models suggesting that they indeed use semantically more relevant deformations than conventional adversarial examples produced by gradient ascent in X space.

For the Mall data set, we train a slightly larger U-Net (RMSE for head count 0.72) and calculate regression values for the original images, the images modified with gradient ascent in $\mathcal{X}$-space and the images modified with gradient ascent in $\mathcal{Z}$-space. As expected, the regression values for the counterfactuals are significantly closer to the target values (0.01 for minimizing pedestrians and 10 for maximizing pedestrians) than those of original images and adversarial examples. Fig. 11 summarizes these findings.

In Fig. 12, we show the localization of heads for the counterfactuals and the adversarial examples for the Mall data set from Fig. 8 using the original and the oracle U-Net. In order to find the head locations, the regression value is rounded to the closest integer representing the number of pedestrians in the image. A Gaussian mixture model with the number of pedestrians as components is then fitted to the probability map. Finally the head positions are defined as the means of the fitted Gaussians. The original U-Net is deceived by the adversarial examples: When maximizing pedestrians (second row) the original U-Net produces false positives, leading to markers at head locations where there are no pedestrians. When minimizing pedestrians,
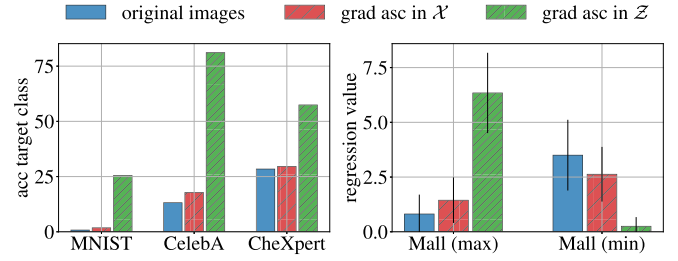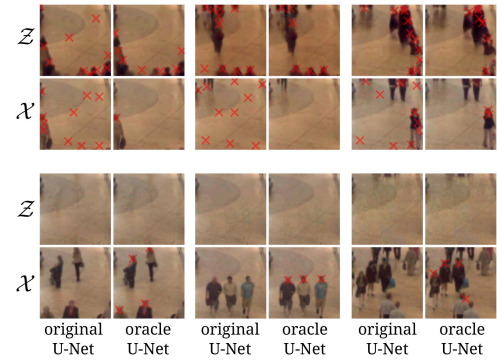
the adversarial examples (forth row) fool the original U-Net into making false negative errors, that is failing to detect pedestrians, although they are clearly present. The oracle U-Net on the other hand produces regression values and probability maps that enable correct identification of pedestrian's head positions (or lack thereof) for the adversarial examples when maximizing (second row) and minimizing (forth row) pedestrians. For the diffeomorphic counterfactuals (first and third row in Fig. 12), the predictions of the two U-Nets are similar, showing that these counterfactuals generalize to the independently trained oracle U-Net.

Nearest neighbours: We compare the original images and the images modified in $\mathcal{X}$ and $\mathcal{Z}$ with data from the data set. We find the k-nearest neighbours (with respect to the euclidean norm) and their respective ground truth classification/regression value. For MNIST, CelebA and CheXpert, we then check what percentage of the nearest neighbours was classified as the target class. For Mall, we check the average number of pedestrians present. Fig. 13 shows that the ten nearest neighbours of the diffeomorphic counterfactuals for MNIST, CelebA and CheXpert share the target classification more often than original images or adversarial examples. For the Mall data set the three nearest neighbours of each counterfactual on average have regression
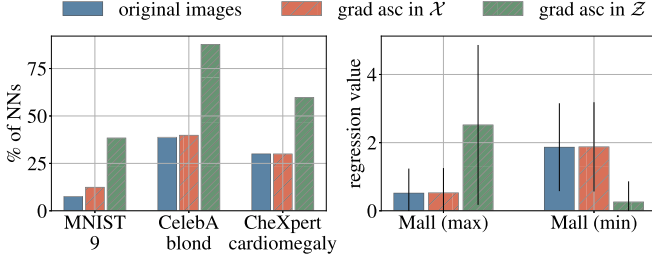
Fig. 13. *Left:* ground truth class for the ten nearest neighbours (NNs) matches the target value ('9', 'blond' and 'cardiomegaly') more often for the counterfactuals found in $\mathcal{Z}$. *Right:* ground truth pedestrian counts averaged over the three nearest neighbours are closer to target values for diffeomorphic counterfactuals. Bars show means and errors denote one standard deviation.

TABLE I
INTERPRETABILITY METRIC IM1 VALUES FOR MNIST AND CELEBA CALCULATED FOR ORIGINAL IMAGES, ADVERSARIAL EXAMPLES AND COUNTERFACTUALS

data set	images	IM1
MNIST	original	2.250 ± 0.711
	gradient ascent in $\mathcal{X}$	1.603 ± 0.317
	gradient ascent in $\mathcal{Z}$	1.056 ± 0.233
CelebA	original	1.160 ± 0.303
	gradient ascent in $\mathcal{X}$	1.144 ± 0.287
	gradient ascent in $\mathcal{Z}$	0.807 ± 0.222

Low values mean better interpretability. We show mean and standard deviation.

values that more closely match the target regression value ($r = 10$ when maximizing and $r = 0$ when minimizing).

IM1 and IM2: Van Looveren and Klaise [73] propose two metrics to test interpretability: IM1 is defined by

$$IM1 = \frac{||x' - AE_t(x')||}{||x' - AE_{c_0}(x')|| + \epsilon}, \qquad (24)$$

where AE_{c_0} and AE_t are two autoencoders which were each trained on data from only one class (original class c_0 and target class t, respectively) and ϵ is a small positive value. The second metric IM2 is defined by

$$IM2 = \frac{||AE_t(x') - AE(x')||}{||x'||_1 + \epsilon}, \qquad (25)$$

where AE is an autoencoder trained on all classes.

IM1 and especially IM2 have been repeatedly criticized [74], [75], [76]. For IM2, we devide by the one-norm $||x'||_1$ of the modified image. This value is large if the image has more bright pixels. Consequently, images with brighter pixels will tend to have a smaller IM2, even though they might not be more interpretable. We therefore limit our evaluation to IM1.

In Table I, we show mean and standard deviation for the interpretability metric IM1 for two data sets; MNIST and CelebA. We calculate the values for the original images, the adversarial examples, produced by gradient ascent in $\mathcal{X}$ space, and the diffeomorphic counterfactuals, produced by gradient ascent in $\mathcal{Z}$ space. A low value for IM1 means the image is better represented by an autoencoder trained on only the target class. Diffeomorphic counterfactuals achieve a lower IM1 value than the adversarial examples, suggesting they are more interpretable.
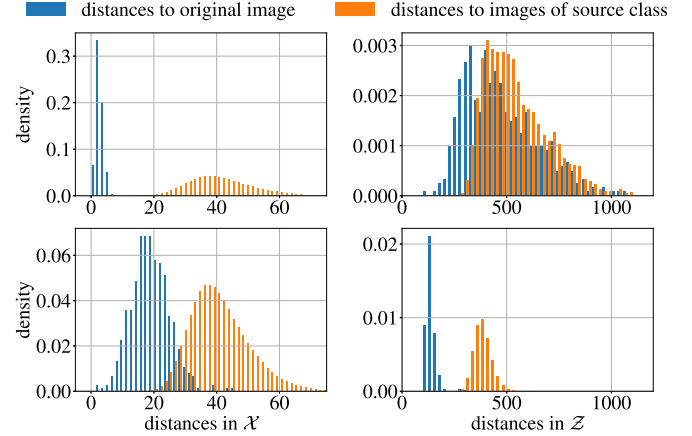

Fig. 14. Euclidean distances in $\mathcal{X}$ and $\mathcal{Z}$ for adversarial examples (first row) and counterfactuals (second row) for the CelebA dataset. Counterfactuals lie closer to their respective source image than adversarial examples when measured in $\mathcal{Z}$, i.e., along the data manifold.

Similarity to original images: Counterfactuals are usually required to be minimal, that is they should be the closest point to the original data point, that lies on the data manifold and reaches the desired confidence Λ with respect to the target class t. Unlike other approaches, we do not encourage similarity by explicitly minimizing the euclidean norm between counterfactual and original image in $\mathcal{X}$ space since the relevant distance is to be computed by the induced metric on the data manifold S or, equivalently, the flat metric in the latent space $\mathcal{Z}$. However, our counterfactuals still preserve high similarity to the respective source image. We confirm this by calculating the euclidean distances in $\mathcal{X}$ and $\mathcal{Z}$ between counterfactuals and all images of the source class (this effect is illustrated for the CelebA dataset in Fig. 14).

The average euclidean norm between counterfactuals and the respective source images is significantly lower than the average euclidean norm between counterfactuals and all images of the source class. For adversarial examples, we expect the euclidean distances in $\mathcal{X}$ to the respective source image to be very small while the euclidean distances in $\mathcal{Z}$ should be larger. Fig. 14 shows the distribution of distances in $\mathcal{X}$ and $\mathcal{Z}$ between counterfactuals/adversarials and their respective source images as well as distances between counterfactuals/adversarials and all images of the source class for the CelebA data set.

We refer to the Appendix C.7, available online, for graphs for the other data sets.

In Tables II and III we show the averaged euclidean norms of the distances in $\mathcal{X}$ and $\mathcal{Z}$ for counterfactuals and adversarials respectively, confirming our expectations.

V. COMPARISON TO OTHER APPROACHES

In Section II, we introduced our method to generate diffeomorphic and approximately diffeomorphic counterfactuals. These methods are independent of the specific classifier and generative model used. While many approaches to generate counterfactuals exist (see Section VI), few are independent of classifier

TABLE II
EUCLIDEAN NORMS L^2 IN $\mathcal{X}$ FOR ADVERSARIAL EXAMPLES FOUND VIA
GRADIENT ASCENT IN $\mathcal{X}$ AND COUNTERFACTUALS FOUND
VIA GRADIENT ASCENT IN $\mathcal{Z}$

data set	img	L^2 source image			L^2 source class		
MNIST	in $\mathcal{X}$	2.54	$\pm$	0.61	8.77	$\pm$	1.25
	in $\mathcal{Z}$	4.82	$\pm$	1.20	9.27	$\pm$	1.32
CelebA	in $\mathcal{X}$	2.84	$\pm$	1.15	41.13	$\pm$	10.03
	in $\mathcal{Z}$	19.10	$\pm$	6.08	40.79	$\pm$	9.19
CheXpert	in $\mathcal{X}$	1.59	$\pm$	0.46	33.68	$\pm$	6.49
	in $\mathcal{Z}$	13.69	$\pm$	4.54	34.71	$\pm$	6.51
Mall (min)	in $\mathcal{X}$	1.34	$\pm$	0.39	19.11	$\pm$	2.67
	in $\mathcal{Z}$	10.98	$\pm$	3.66	17.39	$\pm$	3.10
Mall (max)	in $\mathcal{X}$	1.33	$\pm$	0.17	9.31	$\pm$	3.40
	in $\mathcal{Z}$	15.90	$\pm$	5.28	19.35	$\pm$	6.36

We show mean and standard deviation.

TABLE III
EUCLIDEAN NORMS L^2 IN $\mathcal{Z}$ FOR ADVERSARIAL EXAMPLES FOUND VIA
GRADIENT ASCENT IN $\mathcal{X}$ AND COUNTERFACTUALS FOUND
VIA GRADIENT ASCENT IN $\mathcal{Z}$

data set	img	L^2 source image			L^2 source class		
MNIST	in $\mathcal{X}$	41.86	$\pm$	2.00	42.12	$\pm$	0.74
	in $\mathcal{Z}$	35.26	$\pm$	4.68	39.91	$\pm$	1.61
CelebA	in $\mathcal{X}$	473.72	$\pm$	171.49	542.21	$\pm$	149.63
	in $\mathcal{Z}$	138.35	$\pm$	23.43	380.24	$\pm$	41.23
CheXpert	in $\mathcal{X}$	355.50	$\pm$	114.23	539.18	$\pm$	88.40
	in $\mathcal{Z}$	64.90	$\pm$	25.33	400.31	$\pm$	48.49
Mall (min)	in $\mathcal{X}$	160.53	$\pm$	33.67	199.45	$\pm$	28.56
	in $\mathcal{Z}$	78.56	$\pm$	11.84	180.39	$\pm$	15.78
Mall (max)	in $\mathcal{X}$	142.50	$\pm$	7.61	153.96	$\pm$	8.80
	in $\mathcal{Z}$	116.85	$\pm$	27.50	161.64	$\pm$	23.99

We show mean and standard deviation.

and generative model. As we are interested in comparing methods to generate counterfactuals rather than comparing generative models or classifiers, we thus restrict the comparison of out method to methods that can be applied to independently trained generative models and classifiers. We identify three recent approaches that meet these criteria: Joshi et al. [77], Zhao et al. [78] and Looveren et al. [73]. We generate counterfactuals for the CelebA dataset, using the setup detailed in Section IV-B. As the algorithm by Looveren is not guaranteed to find counterfactuals with high target confidence, we set the target confidence for all algorithms to $\Lambda = 0.5$, and repeat our experiments for this value. We show examples of the generated counterfactuals in Fig. 15 and summarize the quantitative evaluation results in Table IV.

Joshi et al. [77] introduces an algorithm that performs gradient ascent in the latent space of a generative model (specifically VAEs and GANs), while minimizing the difference between original and modified data points. This method is closely related to ours, as we retrieve our proposed method for $\lambda \rightarrow 0$ in (2) in [77]. Setting λ to a larger value leads to larger similarities between the original image and the counterfactual, especially when measured in the image space. However, the resulting counterfactuals generalize worse to the independently trained SVM and Joshi's algorithm requires a longer runtime compared to ours until a counterfactual with the desired target confidence is found (see Table IV).

Zhao et al. [78] propose to perturb the latent representation of a Wasserstein GAN using exhaustive search or continuous relaxation and test the generated images using the classifier. The algorithm then returns the image for which the desired target
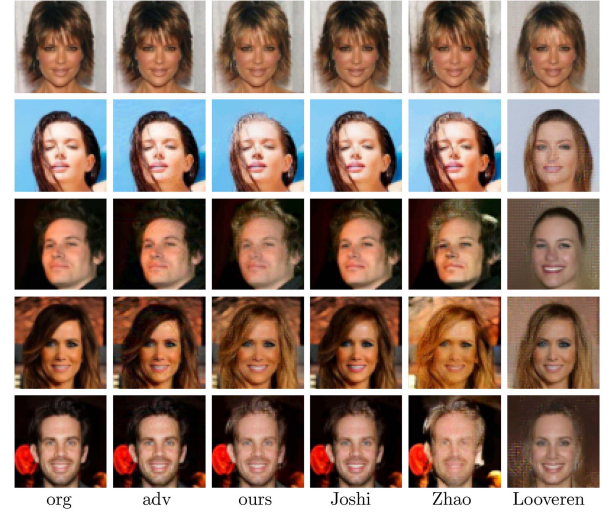

Fig. 15. Selected examples for CelebA ('not blond' to 'blond'). From left to right: original images, adversarial examples, counterfactuals retrieved with our method, Joshi et al. [77], Zhao et al. [78] and Looveren et al. [73]. We use the same generative model (Glow) and classifier for all methods.

confidence is reached within the maximum number of iterations. As we test Zhao's method on a normalizing flow, the base space dimension and thus the search space is much larger than for a comparable GAN. This leads to long computation times (median is 15 min). Furthermore, minimal counterfactuals are less likely to be found resulting in reduced similarity between original and counterfactual (see Table IV). We also note that some searches did not result in counterfactuals as only 438 of 500 runs were successful. This number can be improved by increasing the search radius, or the number of sampled images per iteration at the cost of even longer computation time.

Looveren et al. [73] propose to use a prototype loss that guides the counterfactuals towards the closest prototype that does not represent the source class while simultaneously minimizing the L_1 and L_2 norm of the perturbation in image space. The prototypes are defined in the latent space. Looveren et al. apply their algorithm to a VAE trained on MNIST. As we apply their algorithm to CelebA, a much more diverse data set, the search for prototypes takes very long (6 min per image as we search in the training set to capture more diversity) and the resulting prototypes (the average over the 5 nearest neighbors that are classified as the target class) are of mediocre quality. As a consequence, high-level information of the original image is lost when generating the counterfactuals, as can be seen in Fig. 15. As a consequence, the SSIM between the generated counterfactual and the original is substantially lower than for the other methods. However, the counterfactuals do generalize well to the independently trained SVM.

In summary, our method produces convincing counterfactuals (see Fig. 15) and shows good performance on all quantitative evaluation metrics, while being significantly faster than the algorithms proposed by Joshi et al. [77], Zhao et al. [78] or Looveren et al. [73].

TABLE IV
QUANTITATIVE COMPARISON BETWEEN OUR METHOD AND OTHER APPROACHES TO GENERATE COUNTERFACTUALS

criterion	originals	adversarials	counterfactuals			
			ours	Joshi et al. [76]	Zhao et al. [77]	Looveren et al. [72]
median $SSIM(x, x')$	-	0.99	0.91	0.97	0.86	0.70
median $L_2(x, x')$	-	1.16	9.11	3.07	11.97	8.42
median $L_2(z, z')$	-	279.53	132.20	122.75	160.46	148.02
% predicted as t by SVM	13.17	15.70	29.17	20.03	28.75	37.53
successful runs (total=500)	-	500	500	499	438	498
average computation time	-	00:00:01	00:00:05	00:00:10	00:15:36	00:06:56

VI. RELATED WORK

In this section, we compare our approach with existing methods.

This work builds upon and substantially extends our workshop contribution [79]. Specifically, we introduce both diffeomorphic and approximate diffeomorphic counterfactuals and discuss their relationship. To this end, we theoretically analyze this broader class of generative models in a unified manner which allows us to compare the relative strengths and weaknesses of these approaches. In addition, we provide an expanded discussion of the rigorous construction of the different coordinate systems. In our experimental studies, we consider additional generative models, specifically GANs and AEs, to find approximately diffeomorphic explanations for different data sets demonstrating that our method scales to a high-resolution data sets. Furthermore, we include experiments for additional datasets, such as CelebA-HQ and Mall, as well as further tasks, i.e., regression in addition to classification, and neural network architectures. We also consider a variety of quantitative evaluations of counterfactuals to evaluate the performance of the proposed methods.

A. Counterfactuals With Generative Models

A comparatively small number of publications consider normalizing flows, which started to gain attention relatively recently, in the context of generating counterfactuals.

Sixt et al. [80] train a linear binary classifier directly in the base space of the flow. Adding the weight vector corresponding to the target class to the base space representation and projecting back to image space then produces a counterfactual with semantically changed features. Unlike our method, this approach requires training of a classifier and does not work for a general classifier of arbitrary architecture. In addition to that, their approach relies on the classifier being linear and binary so that the direction in which the image is modified can be determined analytically. Our method is more modular in the sense that the classifier can be pretrained and is independent of the generative model. Furthermore, we allow for the classifier to be non-binary as well as non-linear and we apply our framework for regression.

Our approach is closest in spirit to the one taken by Joshi et al. [77], as discussed above in Section V.

Other works also use autoencoders to generate counterfactuals.

Dhurandhar et al. [81] use elastic net regularization to keep the perturbation δ to the original data small and sparse. Furthermore, they use an autoencoder to minimize the reconstruction loss of the modified image and thus make sure the counterfactual lies on the data manifold. This approach was expanded by adding a prototype loss [73], as discussed in Section V above. Both approaches apply their algorithm on tabular data and MNIST. Our approach differs from these works as we are not using generative models as a regularizer but directly modify the latent space representation. Our method also does not require access to labelled training data in order to compute prototypes, is applicable to high dimensional image data sets, and has no hyperparameters weighting different loss components.

A number of references such as Zhao et al. [78] discussed in Section V above, use GANs to generate counterfactuals.

Samangouei et al. [82] propose to use classifier prediction specific encoders together with a GAN which are trained to generate a reconstruction, a counterfactual, and a mask indicating which pixels should change between the counterfactual and the original. A similar approach is proposed by Singla et al. [83] as they also train a GAN, that is conditioned on the classifier predictions, jointly with an encoder to produce realistic looking counterfactuals. In both works, the classifier is incorporated in the training process of the GAN. After the training, the GAN generates counterfactuals without querying the classifier. As a consequence information about the classifier is integrated into the GAN purely during training, while our approach can be applied to independently trained models, which allows us to find counterfactuals for different classifiers using the same generative model.

Goetschalckx et al. [84] learn directions in the latent space by differentiating through a classifier and the generator so that cognitive properties of generated images, such as memorability, can be modified by moving in those directions. They do not specifically aim to produce counterfactuals but their approach touches on related concepts. A difference to our work is that the latent representation is restricted to be modified along a single direction, while for our method the direction of change is dictated by the gradient over several update steps.

Lius et al. [85] use a GAN specifically trained for editing that they condition on the original query image and the desired attributes. They apply gradient descent to find attributes that cause the GAN to generate an image that the classifier predicts as the target class, while at the same time enforcing the image to be close to the original. In contrast to this approach, we directly modify the latent representation rendering our approach independent of the exact structure of the generative model. We also observe that our counterfactuals stay close to the original image without explicitly enforcing similarity.

Finally, generative models have been used before in the context of adversarial examples. Song et al. [86] optimize in the latent space of a (conditioned) GAN to generate adversarial

examples which are unrelated to any data point (*unrestricted adversarial examples*) and validated against human observers. Lin et al. [87] use adversarial training in the latent space of a generative model (they use StyleGAN) to improve robustness against adversarial attacks. A further line of research studies learned latent manifolds in the context of enhanced sampling methods for GAN training, see [88], [89].

We present the results from a detailed qualitative and quantitative comparison between our method and other approaches that generate counterfactual explanations in Section V.

B. Quantitative Metrics for Counterfactuals

Many of the above works are limited to qualitative assessment of counterfactuals. The quantitative assessment of counterfactuals is still an active research area, a summary of quantitative measures can be found in [74].

The most reliable evaluation of counterfactuals can be obtained by a large study that lets human agents evaluate the generated counterfactuals. As those are relatively costly to conduct and can introduce unexpected biases if not designed carefully, their application is often infeasible. Nevertheless, a few works [78], [80], [90], [91] undertake small user studies ($9 \leq N \leq 60$) on a relatively limited set of generated counterfactuals. We aim to approximate an independent human evaluation by testing our counterfactuals on newly trained models that serve as oracles.

Some works [83], [92], [93] apply a metric commonly used for generative model evaluation, the Fréchet Inception Distance (FID) score [94], measuring the quality of the generated explanation compared to samples from the data set. As we find counterfactuals by moving directly in latent space the FID for our counterfactuals would be very similar to the FID of the generative model itself. We therefore do not consider the FID to be a meaningful metric for (approximate) diffeomorphic counterfactuals.

Van Looveren and Klaise [73] propose two metrics to test interpretability: IM1 (defined in (24) above) uses two autoencoders which were each trained on data from only one class and computes the relative reconstruction error of the counterfactual. The second metric IM2 (defined in (25) above) calculates the normalized difference between a reconstruction of an autoencoder trained on the target class and an autoencoder trained on all classes. Van Looveren and Klaise use these two metrics to compare how different loss functions effect the relative interpretability measured by IM1 and IM2 for the MNIST data set. We limit the quantitative evaluation of our counterfactuals to IM1, since IM2 has been subject to controversies.

Other works [82], [93] check substitutability. They train classifiers on a training set consisting of generated counterfactuals and compare their performance on the original test data set to a classifier trained on the original training set. As we can generate counterfactuals that are classified with very different confidence, this method may not be useful as results may be highly dependent on the choice of confidence.

Other methods aim to evaluate explanations by replacing pixel values or entire regions based on the importance of features in the explanation [28], [82], [95], [96], [97] and testing the performance of a classifier on the modified images. Those methods may suffer from creating images that lie off the data manifold, so that a thorough comparison may require extensive retraining [98].

VII. CONCLUSION

In this work, we proposed theoretically rigorous yet practical methods to generate counterfactuals for both classification as well as regression tasks, namely exact and approximate diffeomorphic counterfactuals. The exact diffeomorphic counterfactuals are obtained by following gradient ascent in the base space of a normalizing flow. While approximate diffeomorphism are obtained with the help of either generative adversarial networks or variational autoencoders. Our thorough theoretical analysis, using Riemannian differential geometry, shows that for well-trained models, our counterfactuals necessarily stay on the data manifold during the search process and consequently exhibit semantic features corresponding to the target class. Approximate diffeomorphic counterfactuals come with the risk of information loss but allow excellent scalability to higher dimensional data. Our theoretical findings are backed by experiments which both quantitatively and qualitatively demonstrate the performance of our method on different classification as well as regression tasks and for numerous data sets.

The application of our counterfactual explanation method is straightforward and requires no retraining, so that it can be readily applied to investigate common problems in deep learning like identifying biases for classifiers or training data or scrutinizing falsely classified examples — all common tasks for applications in computer vision.

For future work, we intend to investigate the benefit of our counterfactuals in the sciences, in particular for medical applications such as digital pathology [11] or brain computer-interfaces [99].

Furthermore, the method presented in this work is not restricted to the generation of counterfactuals for image data or in computer vision. In particular, one could imagine applications e.g., in chemistry and physics where the technique proposed here may be used to optimize desired properties of stable molecules which are restricted to minima of the associated potential energy surface [15], [100], [101], [102].

In practical applications, it is often beneficial to incorporate symmetries as inductive bias following a well-established paradigm in machine learning [103], [104]. It is straightforward to incorporate symmetries into our method by employing equivariant normalizing flows as constructed e.g., in [105].

In conclusion, our method is applicable to a broad range of computer vision problems and beyond as it provides a way to optimize the output of a predictive model on the data manifold given only indirectly by a trained generative model.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[6] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, 2016, Art. no. 484.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[8] D.-O. Won, K.-R. Müller, and S.-W. Lee, "An adaptive deep reinforcement learning framework enables curling robots with human-like performance in real-world conditions," *Sci. Robot.*, vol. 5, no. 46, 2020, Art. no. eabb9764.

[9] T. Lengauer, O. Sander, S. Sierra, A. Thielen, and R. Kaiser, "Bioinformatics prediction of HIV coreceptor usage," *Nature Biotechnol.*, vol. 25, no. 12, 2007, Art. no. 1407.

[10] D. Capper et al., "DNA methylation-based classification of central nervous system tumours," *Nature*, vol. 555, no. 7697, 2018, Art. no. 469.

[11] A. Binder et al., "Morphological and molecular breast cancer profiling through explainable machine learning," *Nature Mach. Intell.*, vol. 3, no. 4, pp. 355–366, 2021.

[12] F. Klauschen, "Toward explainable artificial intelligence for precision pathology," *Annu. Rev. Pathol.: Mechanisms Dis.*, vol. 19, 2024, pp. 541–570.

[13] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Commun.*, vol. 5, 2014, Art. no. 4308.

[14] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," *Science*, vol. 365, no. 6457, 2019, Art. no. eaaw1147.

[15] O. T. Unke et al., "Machine learning force fields," *Chem. Rev.*, vol. 121, no. 16, pp. 10 142–10 186, 2021.

[16] O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, "SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects," *Nature Commun.*, vol. 12, 2021, Art. no. 7273.

[17] J. Degrave et al., "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[18] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature Commun.*, vol. 10, 2019, Art. no. 1096.

[19] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin, "Finding and removing Clever Hans: Using explanation methods to debug and improve deep models," *Inf. Fusion*, vol. 77, pp. 261–295, 2022.

[20] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, no. 61, pp. 1803–1831, 2010.

[21] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[22] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham, Switzerland: Springer, 2019.

[23] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," in *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, Mar. 2021.

[24] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 9, no. 4, 2019, Art. no. e1312.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.

[26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[27] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[28] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, no. 7, 2015, Art. no. e0130140.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2016, pp. 1135–1144.

[30] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.

[31] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.

[32] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.

[33] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6967–6976.

[34] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," in *Proc. Int. Conf. Mach. Learn. Workshop Visualization Deep Learn.*, 2017.

[35] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[36] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3449–3457.

[37] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognit.*, vol. 65, pp. 211–222, 2017.

[38] P.-J. Kindermans et al., "Learning how to explain neural networks: PatternNet and PatternAttribution," in *Proc. Int. Conf. Learn. Representations*, 2018.

[39] B. Kim et al., "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2673–2682.

[40] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller, and G. Montavon, "Toward explainable artificial intelligence for regression models: A methodological perspective," *IEEE Signal Process. Mag.*, vol. 39, no. 4, pp. 40–58, Jul. 2022.

[41] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019.

[42] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," in *Proc. Int. Conf. Neural Inf. Process. Syst. ML-RSA Workshop*, 2020.

[43] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Fariña, "A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence," *IEEE Access*, vol. 9, pp. 11 974–12 001, 2021.

[44] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[45] S.-I. Amari, *Information Geometry and its Applications*. Berlin, Germany: Springer, 2016.

[46] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[47] H. Shao, A. Kumar, and P. Thomas Fletcher, "The Riemannian geometry of deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 315–323.

[48] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13 567–13 578.

[49] C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel, "Fairwashing explanations with off-manifold detergent," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 314–323.

[50] A.-K. Dombrowski, C. J. Anders, K.-R. Müller, and P. Kessel, "Towards robust explanations for deep neural networks," *Pattern Recognit.*, vol. 121, 2022, Art. no. 108194.

[51] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[52] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding density functionals with machine learning," *Phys. Rev. Lett.*, vol. 108, no. 25, 2012, Art. no. 253002.

[53] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[54] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "GAN inversion: A survey," 2021, *arXiv:2101.05278*.

[55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[56] J. Lee, *Introduction to Smooth Manifolds*, 2nd ed. New York, NY, USA: Springer-Verlag, 2012.

[57] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[58] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.

[59] J. Irvin et al., "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 590–597.

[60] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. Brit. Mach. Vis. Conf.*, 2012, Art. no. 3.

[61] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2467–2474.

[62] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2256–2263.

[63] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*. Berlin, Germany: Springer, 2013, pp. 347–382.

[64] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer, 2015, pp. 234–241.

[65] J. Ribera, D. Guera, Y. Chen, and E. J. Delp, "Locating objects without bounding boxes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6479–6489.

[66] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using RealNVP," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.

[67] D. P. Kingma and P. Dhariwal, "GLOW: Generative flow with invertible 1x1 convolutions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 10236–10245.

[68] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018.

[69] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. H. Bermano, "HyperStyle: StyleGAN inversion with hypernetworks for real image editing," 2021, *arXiv:2111.15666*.

[70] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[71] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.

[72] A. Sauer, K. Schwarz, and A. Geiger, "StyleGAN-XL: Scaling StyleGAN to large diverse datasets," in *Proc. ACM SIGGRAPH Conf.*, 2022, Art. no. 49.

[73] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, and J. A. Lozano, Eds., Cham: Springer, 2021, pp. 650–665.

[74] F. Hvilshøj, A. Iosifidis, and I. Assent, "On quantitative evaluations of counterfactuals," 2021, *arXiv:2111.00177*.

[75] D. Mahajan, C. Tan, and A. Sharma, "Preserving causal constraints in counterfactual explanations for machine learning classifiers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019.

[76] L. Schut et al., "Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1756–1764.

[77] S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh, "Towards realistic individual recourse and actionable explanations in black-box decision making systems," in *Proc. Safe Mach. Learn. Workshop ICLR*, 2019.

[78] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2018.

[79] A.-K. Dombrowski, J. E. Gerken, and P. Kessel, "Diffeomorphic explanations with normalizing flows," in *Proc. ICML Workshop Invertible Neural Netw. Normalizing Flows Explicit Likelihood Models*, 2021.

[80] L. Sixt, M. Schuessler, P. Weiß, and T. Landgraf, "Interpretability through invertibility: A deep convolutional network with ideal counterfactuals and isosurfaces," 2021. [Online]. Available: https://openreview.net/forum?id=8YFhXYe1Ps

[81] A. Dhurandhar et al., "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 590–601.

[82] P. Samangouei, A. Saeedi, L. Nakagawa, and N. Silberman, "ExplainGAN: Model explanation via decision boundary crossing transformations," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 666–681.

[83] S. Singla, B. Pollack, J. Chen, and K. Batmanghelich, "Explanation by progressive exaggeration," in *Proc. Int. Conf. Learn. Representations*, 2020.

[84] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "GANalyze: Toward visual definitions of cognitive image properties," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5744–5753.

[85] S. Liu, B. Kailkhura, D. Loveland, and Y. Han, "Generative counterfactual introspection for explainable deep learning," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2019, pp. 1–5.

[86] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 8322–8333.

[87] W.-A. Lin, C. P. Lau, A. Levine, R. Chellappa, and S. Feizi, "Dual manifold adversarial robustness: Defense against Lp and non-Lp adversarial attacks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, pp. 3487–3498.

[88] J. Cao, Y. Guo, Q. Wu, C. Shen, J. Huang, and M. Tan, "Adversarial learning with local coordinate coding," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 707–715.

[89] J. Cao, Y. Guo, Q. Wu, C. Shen, J. Huang, and M. Tan, "Improving generative adversarial networks with local coordinate coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 211–227, Jan. 2022.

[90] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2376–2384.

[91] O. Lang et al., "Explaining in style: Training a GAN to explain a classifier in StyleSpace," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 693–702.

[92] R. Rombach, P. Esser, and B. Ommer, "Making sense of CNNs: Interpreting deep representations & their invariances with INNs," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 647–664.

[93] H. Kim et al., "Counterfactual fairness with disentangled causal effect variational autoencoder," in *Proc. 35th AAAI Conf. Artif. Intell. 33rd Conf. Innov. Appl. Artif. Intell. 11th Symp. Educ. Adv. Artif. Intell.*, 2021, pp. 8128–8136.

[94] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

[95] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

[96] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," in *Proc. 8th Workshop Comput. Approaches Subjectivity Sentiment Social Media Anal.*, Copenhagen, Denmark, 2017, pp. 159–168.

[97] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity checks for saliency metrics," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6021–6029.

[98] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 9734–9745.

[99] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller, "Introduction to machine learning for brain imaging," *Neuroimage*, vol. 56, no. 2, pp. 387–399, 2011.

[100] N. Gebauer, M. Gastegger, and K. Schütt, "Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 680.

[101] N. W. Gebauer, M. Gastegger, S. S. Hessmann, K.-R. Müller, and K. T. Schütt, "Inverse design of 3D molecular structures with conditional generative neural networks," *Nature Commun.*, vol. 13, 2022, Art. no. 973.

[102] V. G. Satorras, E. Hoogeboom, F. Fuchs, I. Posner, and M. Welling, "E(n) equivariant normalizing flows," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 4181–4192.

[103] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nature Commun.*, vol. 9, 2018, Art. no. 3887.

[104] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Sci. Adv.*, vol. 3, no. 5, 2017, Art. no. e1603015.

[105] J. Köhler, L. Klein, and F. Noé, "Equivariant flows: Exact likelihood generative learning for symmetric densities," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5361–5370.
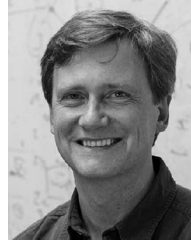
Klaus-Robert Müller is a professor of computer science with TU Berlin and is director of the Berlin Institute for the Foundations of Learning and Data. In 2017, he became external scientific member of the Max Planck Society. In 2012, he was elected member of the German National Academy of Sciences–Leopoldina, in 2017 of the Berlin Brandenburg Academy of Sciences and in 2021 of the German Academy of Sciences and Enigneering. He is also an ISI Highly Cited researcher in the cross-disciplinary area.
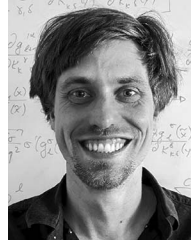
Pan Kessel received the PhD degree in string theory from the Max Planck Institute for Gravitational Physics, in 2017 after which he joined the Machine Learning Group, TU Berlin and with the Berlin Institute for the Foundations of Learning and Data as a postdoc. He is now a senior machine learning scientist with Prescient Design, Genentech.

Ann-Kathrin Dombrowski received the bachelor's and master's degrees in computational engineering science from RWTH Aachen, in 2014 and 2017, respectively, and the PhD degree from the Machine Learning Group, TU Berlin, in 2022 under supervision of Prof. Müller.

Jan E. Gerken received the PhD degree in string theory from the Max Planck Institute for Gravitational Physics, in 2020. After a postdoc position with the Department for Mathematical Sciences, Chalmers in Gothenburg, he worked as a postdoctoral researcher with the ML Group, TU Berlin and with the Berlin Institute for the Foundations of Learning and Data. Currently, he works as an assistant professor in mathematics for AI foundations at Chalmers in Gothenburg.