The Gaussian Discriminant Variational Autoencoder (GdVAE): A Self-Explainable Model with Counterfactual Explanations

Anselm Haselhoff[1,2], Kevin Trelenberg[1],
Fabian Küppers[3], and Jonas Schneider[3]

[1] TrustIn.AI Lab, Ruhr West University of Applied Sciences, Germany
[2] TML Lab, The University of Sydney, Australia, [3] e:fs TechHub GmbH, Germany
{name.surname}@hs-ruhrwest.de, {name.surname}@efs-techhub.com
https://trustinai.github.io/gdvae

Fig. 1: FFHQ high-resolution (1024×1024) counterfactuals x^{δ} for smiling.

Abstract. Visual counterfactual explanation (CF) methods modify image concepts, *e.g.*, shape, to change a prediction to a predefined outcome while closely resembling the original query image. Unlike self-explainable models (SEMs) and heatmap techniques, they grant users the ability to examine hypothetical "what-if" scenarios. Previous CF methods either entail post-hoc training, limiting the balance between transparency and CF quality, or demand optimization during inference. To bridge the gap between transparent SEMs and CF methods, we introduce the GdVAE, a self-explainable model based on a conditional variational autoencoder (CVAE), featuring a Gaussian discriminant analysis (GDA) classifier and integrated CF explanations. Full transparency is achieved through a generative classifier that leverages class-specific prototypes for the downstream task and a closed-form solution for CFs in the latent space. The consistency of CFs is improved by regularizing the latent space with the explainer function. Extensive comparisons with existing approaches affirm the effectiveness of our method in producing high-quality CF explanations while preserving transparency. Code and models are public.

Keywords: Self-explainable generative model · counterfactual explanation · variational autoencoder · Riemannian metric · manifold traversal

1 Introduction

Deep neural networks (DNNs), such as generative adversarial networks (GANs) for image generation [25] and DNN classifiers [46], have achieved notable success. However, they suffer from limited interpretability, often being considered black boxes with decision processes not well understood by humans.

Generative explanation methods identify meaningful latent space directions related to independent factors of variation (*e.g.*, shape). Previous work finds these directions by enforcing disentanglement during training or analyzing the latent space [3,8,9,20,25,38,39,42]. Explanations are obtained by visualizing the effect of changes in the latent space. Generative models are also used in *counterfactual* (CF) reasoning, which answers questions like, "How can the example be changed to belong to category B instead of A?". This allows users to explore hypothetical "what-if" scenarios [14]. Recent advances combine generative models and classifiers to generate CF explanations, with enhanced techniques focusing on realism and consistency [14,15,21,26,30,40,43]. However, many methods lack transparency, as the CF generation often relies on a separate black-box model, and the classifier itself may not guarantee transparency either.

Self-explainable models (SEMs) provide explanations alongside their predictions without the need for post-hoc training [1,2,6,7,13]. Many SEMs are based on prototype learning, using these transparent and often visualizable prototypes as a bottleneck in a white-box classifier. This white-box classifier (*e.g.*, linear predictor) is optimized end-to-end. However, generating CFs for these models is only feasible through post-hoc methods, potentially reducing transparency.

To bridge the gap between transparent SEMs and CF methods, we introduce GdVAE, a conditional variational autoencoder (CVAE) designed for transparent classification and CF explanation tasks. Full transparency is achieved with a generative classifier using class-specific prototypes and a closed-form solution for CFs in the latent space, inspired by Euclidean and Riemannian manifold perspectives. The prototype explanations come from the distributions provided by the CVAE's prior network, meaning the classifier has no additional trainable parameters. We solve the inference problem of the CVAE, which involves unknown classes, using expectation maximization that iteratively uses the classifier. Finally, we generate local CF explanations in the latent space using a transparent linear function that supports user-defined classifier outputs, and then use the decoder to translate them back to the input space. Joint training of the classifier and generative model regularizes the latent space for class-specific attributes, enabling realistic image and CF generation. An additional regularizer ensures consistency between query confidence and true confidence of the classified CF.

In summary, our contributions are: (i) We introduce a SEM for vision applications, based on a CVAE, with an intrinsic ability to generate CFs; (ii) We offer global explanations in the form of prototypes directly utilized for the downstream task, visualizable in the input space; (iii) We provide transparent, realistic, and consistent local CF explanations, allowing users to specify a desired confidence value; (iv) We conduct a thorough comparative analysis of our method, analyzing performance, consistency, proximity, and realism on common vision datasets.

2 Related Work

Since our work is a SEM with integrated visual CF explanations, we begin by outlining the categorization criteria. Subsequently, we review generative and CF explanations, as well as prototype-based SEMs tailored for vision tasks. Generative models naturally serve as an integral component of an explainer function used for generating CF images. Typically, this function is learned through probing the classifier and optimizing it for specific properties. In CF research, while various properties are discussed, *realism*, *proximity*, and *consistency* stand out as widely accepted criteria. To simplify, CFs should resemble natural-looking images (*realism*), make minimal changes to the input (*proximity*), and maintain query confidence consistency with the classifier's predictions when used as input (*consistency*) [4, 14, 26, 43]. Similarly, in prototype-based SEMs, transparency is crucial, characterized by the visualization of prototypes (PT) in the input space and their utilization in a white-box classifier [13]. To align our work with CF methods and SEMs, we adopt the following predicates.

1. *Realism:* CFs should stem from the data manifold with a natural appearance.
2. *Consistency:* The explainer function $\mathcal{I}_F(x, \delta) : (\mathbb{R}^N, \mathbb{R}) \rightarrow \mathbb{R}^N$ should be conform with the desired classifier output $F(x^\delta) \approx F(x) - \bar{\delta} = \delta$, where $\bar{\delta}$ is the desired perturbation of the output function, δ the desired output, and $x^\delta = \mathcal{I}_F(x, \delta)$ the CF for the input x [43].
3. *Proximity*: The CF should minimally change the input.
4. *Transparency:* A model should use explanations (e.g., prototypes) as intrinsic parts of a white-box predictor, and they should be visualizable in input space.

Generative Explanations (a). The first group of approaches aims to explain pre-trained generative models (*e.g.*, GANs). Directions for interpretable control can be derived through unsupervised [11, 22, 37, 49] or supervised [15, 42, 52] analysis of generative models. GANalyze [15] employs a pre-trained classifier to learn linear transformations in the latent space, whereas [42] directly use a linear classifier in the latent space to define the direction. Except for UDID [49], all mentioned methods use linear explainer functions for manifold traversal. Most of these methods, due to their linear explainer function, provide transparency in latent space manipulation. Transparent classification and CF generation aren't their primary focus, though they can generate CFs without optimizing for factors like *realism*. Our method aligns with these post-hoc methods by using a transparent linear explainer function for CF generation. In contrast, our approach excels by more effectively regularizing the latent space through end-to-end training.

Visual Counterfactual Explanations (b). The second category of methods focuses on CF generation, optimizing *realism*, *proximity*, and *consistency*. EBPE [43] and its extension [14] explain pre-trained classifiers by using a GAN to generate CF images with user-defined confidence values. Similarly, works like [21, 23, 24, 26, 30, 40], train generators with a simpler consistency task, where the user pre-defines the class label only, without specifying the confidence. DiME [24] optimizes CFs iteratively, incurring significant computational costs. Unlike other methods, C3LT [26] only manipulates the latent space with neural networks,

Table 1: Comparison of explanation methods. "Design" column groups approaches according to the headings: (a), (b), and (c). The symbol ∼ indicates that most methods use a transparent linear function for latent space traversal and may not be explicitly designed for generating CFs. Explanations are categorized into Counterfactuals ("CF") and Prototype-based ("PT"). †: some works [7, 50] use alternating optimization.

Design	Approach	Transparency	Explanation		Optimization
			CF	PT	
(a)	[11, 15, 22, 37, 42, 49, 52]	∼	∼		post-hoc
(b)	[14,21,23,24,26,30,40,43]		✓		post-hoc
(c)	[7, 13, 50]	✓		✓	end-to-end†
	GdVAE (ours)	✓	✓	✓	end-to-end

similar to methods in the first category, requiring access to a pre-trained generative model. A different line of research [16, 47] seeks to replace image regions based on distractor images of the CF class. In [30] and [21], the classifier and generator are closely coupled during training to enforce a latent space that encodes class-specific information. StylEx [30], like [24], requires time-consuming inference-time optimization and classifier probing to identify influential coordinates for each input image. In contrast, ECINN [21] is unique in its use of a transparent linear explainer function and an invertible model. Our method is closely related to ECINN, with the distinction that they require a post-hoc analysis of the training data to determine the parameters of the explainer function. Consequently, unlike our model, they approximate the true decision function of their classifier for CF generation, resulting in a loss of transparency. In contrast, all the other methods described employ complex DNNs for CF generation and the classifier, limiting their transparency. Our approach mirrors these CF generation processes but stands out with a transparent, linear explainer function analytically linked to our white-box classifier's decision function.

Self-explainable Models (c). The classifier and CF generation of our GdVAE are closely tied to the same prototypical space. A line of works that comprises this prototype-based learning can be found in SEM research [2,7,13,17,36,50,51]. In [13], a categorization of SEMs was introduced, and our specific focus is on methods prioritizing the *transparency* property [7, 13, 50]. To maintain interpretability, these SEMs employ similarity scores that measure the likeness between features and prototypes within the latent space. Afterwards, these scores are employed within a linear classifier, which encodes the attribution of each prototype to the decision. Unlike ProtoPNet [7] and TesNET [50], ProtoVAE [13] uses end-to-end training, utilizing a model capable of decoding learned prototypes, resulting in a smooth and regularized prototypical space.

Our GdVAE employs one prototype per class with a linear Bayes' classifier, implicitly utilizing Mahalanobis distance instead of a 2-norm-based similarity. Unlike ProtoVAE, our SEM enhances transparency and CF generation, unifying these research areas effectively. Refer to Tab. 1 for an overview.

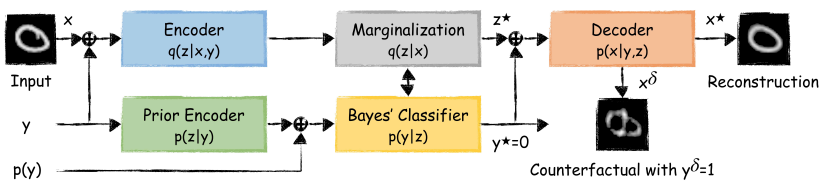


Fig. 2: The GdVAE has three branches: *1.) Feature Detection & Reconstruction:* The encoder, akin to a recognition network in a CVAE, generates latent code z. During inference, with an unknown class y, the marginal $q(z|x)$ acts as a feature detection module. The decoder reconstructs the input image x using samples $z^\star$ from the marginal and $y^\star$ from the classifier. *2.) Prior Encoder & Classifier:* The prior encoder learns the latent feature distribution independently of the input image, providing necessary distributions for the generative classifier. *3.) Explanation:* During inference, the model generates a class prediction $y^\star$ and a latent variable $z^\star$. The user requests a CF by defining a desired confidence value and uses a linear function $z^\delta = \mathcal{I}_f(z^\star, \delta)$ to modify $z^\star$ to z^δ. The CF x^δ is obtained by transforming z^δ to image space using the decoder. The CF illustrates crossing the decision boundary, showing features of digits 0 and 1.

3 Method

Notation. We address a supervised learning problem with input samples $x \in \mathbb{R}^N$ (*e.g.*, images) and class labels $y \in \{1, \ldots, K\}$. The latent variable $z \in \mathbb{R}^M$ is used for both autoencoding and classification. Model parameters θ and ϕ define the neural networks (NNs) for probabilistic models. For example, we use a Gaussian posterior $q_\phi(z|x, y) = \mathcal{N}(\mu_z(x, y; \phi), \Sigma_z(x, y; \phi))$, with $\mu_z(x, y; \phi)$ and $\Sigma_z(x, y; \phi)$ as NNs. In discussions involving encoders and decoders, we omit the class input y for simplicity and employ shorthand notations for encoders and decoders, such as $h(x) = \mu_z(x; \phi)$ and $g(z) = \mu_x(z; \theta)$. We express a probabilistic classifier for discrete variables as $p_\theta(y|z)$, which can be transformed into discriminant functions, denoted as $f^{(i)}(z) = \log p_\theta(y = i|z)$. For the two-class problem we can use a single discriminant $f(z) = f^{(c)}(z) - f^{(k)}(z)$, where positive values correspond to class c and negative values to class k. The following explanation methods are discussed solely for the two-class problem. The composition of the encoder $h(x)$ and the discriminant $f(z)$ can be used as an input-dependent discriminant function $F(x) = (f \circ h)(x)$. Similarly, we can obtain CF images by generating CFs in the latent space with respect to $f(z)$ and using the decoder to transform them into the image space $\mathcal{I}_F(x, \delta) = (g \circ \mathcal{I}_f)(z, \delta)$.

Overview. The GdVAE enhances an autoencoder with an integrated generative classifier. We consider a generative model $p_\theta(x, y, z) = p_\theta(x|y, z)p_\theta(y, z)$ with two distinct factorizations for $p_\theta(y, z) = p_\theta(z|y)p_\theta(y) = p_\theta(y|z)p_\theta(z)$, defining coupled processes. The first factorization establishes a class conditional prior $p_\theta(z|y)$ for the latent variable z and delineates an autoencoder (M1), while the second integrates a discriminative classifier $p_\theta(y|z)$ (M2) using the latent variable. Later, we'll employ a generative classifier using the prior encoder's mean values as decision prototypes that will benefit from the discriminative learning signal. See an overview and description in Fig. 2.

3.1 Autoencoding and Generative Classification

Model Distributions. *CVAE including a class prior (M1):* For the first factorization of $p_\theta(x, y, z)$ we assume the observed variable x to be generated from the set of latent variables z and y through the following process

$$y \sim p_\theta(y) = \mathrm{Cat}_y(\pi(\theta)) \,, \tag{1}$$

$$z|y \sim p_\theta(z|y) = \mathcal{N}\left(\mu_z(y;\theta), \Sigma_z(y;\theta)\right) , \tag{2}$$

$$x|y,z \sim p_\theta(x|y,z) = \mathcal{N}\left(\mu_x(y,z;\theta), \Sigma_x(y,z;\theta)\right) , \tag{3}$$

with categorical distribution $\mathrm{Cat}_y(\pi(\theta)) = \prod_{k=1}^{K} \pi(\theta)_k^{\mathbb{1}\{y=k\}}$, where π is a probability vector and $\mathbb{1}\{\cdot\}$ is the indicator function. This process defines a CVAE [45] with an added class prior $p_\theta(y)$, capturing class frequency. Thus, we capture both a prior encoder $p_\theta(z|y)$ and a class prior, which are used by our classifier.

GDA model with latent prior (M2): The second factorization of $p_\theta(x, y, z)$ describes our classification model, where the target class y (observable during training) is generated by the latent code z according to our second process

$$z \sim p(z) = \mathcal{N}\left(0, I\right) , \tag{4}$$

$$y|z \sim p_\theta(y|z) = \mathrm{Cat}_y(\tau(z;\theta)) \,, \tag{5}$$

$$x|y,z \sim p_\theta(x|y,z) = \mathcal{N}\left(\mu_x(y,z;\theta), \Sigma_x(y,z;\theta)\right) . \tag{6}$$

Instead of using a separate NN to estimate τ, we reuse M1's distributions to obtain the categorical distribution $p_\theta(y|z) = \eta p_\theta(z|y) p_\theta(y)$, where η is a normalization constant in the context of Bayes' theorem. In addition to this coupling, both models are jointly trained using a unified learning objective.

Learning Objective. Our generative models feature non-conjugate dependencies, making it intractable to maximize the conditional log-likelihood. Thus, we employ a surrogate posterior $q_\phi(z|x, y)$ to approximate the true posterior $p_\theta(z|y)$ [27]. The surrogate, also called the recognition model, adapts the latent code distribution based on x. Instead of maximizing the log-likelihood $\log p_\theta(x, y)$ of our model, we use the evidence lower bound (ELBO) to define our loss. The resulting per sample loss for the GdVAE is $\mathcal{L}^{gd} = \alpha\mathcal{L}^{M1} + \beta\mathcal{L}^{M2}$, with

$$\mathcal{L}^{M1} = -\mathbb{E}_{z,y\sim q_\phi}[\log p_\theta(x|y,z)] + KL(q_\phi(z|x,y)||p_\theta(z|y)) - \log p_\theta(y), \tag{7}$$

$$\mathcal{L}^{M2} = -\mathbb{E}_{z,y\sim q_\phi}[\log p_\theta(x|y,z)] + KL(q_\phi(z|x,y)||p(z)) - \mathbb{E}_{z\sim q_\phi}[\log p_\theta(y|z)] \,. \tag{8}$$

α and β control the balance between M1 and M2, and KL denotes the Kullback-Leibler divergence. The derivation of the loss and ELBO can be found in the Supplement. Note that during inference, we cannot directly sample from the encoder $q_\phi(z|x, y)$ since the class y is unknown. Instead, we conduct ancestral sampling by first sampling from $q_\phi(y|x)$ and afterwards from $q_\phi(z|x, y)$ to approximate $q_\phi(z|x)$. To ensure coherence between the training and inference processes, we compute the expectations relative to $q_\phi(z|x)$ and $q_\phi(z, y|x)$ during training, respectively. This alignment enhances the accuracy of predictions.

Marginalization. The training process is straightforward when labels are observable, and we can directly sample from the conditional encoder $q_\phi(z|x,y)$. Likewise, during inference with the model, we require an estimate of z given x and y. The challenge here is that y is unknown during inference.

Therefore, we draw inspiration from semi-supervised learning [28], employ a factorized probabilistic model $q_\phi(z,y|x) = q_\phi(z|x,y)q_\phi(y|x)$ and perform a marginalization $q_\phi(z|x) = \sum_{y=1}^{K} q_\phi(z|x,y)q_\phi(y|x)$. In practice, besides the conditional encoder $q_\phi(z|x,y)$, a classifier $q_\phi(y|x)$ is needed. To avoid the need for sampling in the image space [45], we initialize the classifier with the class prior $p_\theta(y)$ and iteratively refine both the classifier and the latent feature model. This expectation-maximization (EM) approach is detailed in Algorithm 1, with a proof in the Supplement. In contrast to a standard EM for a Gaussian mixture model (GMM), where we usually estimate mean and covariance values, we employ the GMM to generate S data samples $z^{(s)}$. Subsequently, we perform a soft assignment using the fixed classifier $p_\theta(y|z)$ and, akin to [12], reestimate $q_\phi(y|x)$. The closer our estimate aligns with the true class of x, the more samples $z^{(s)}$ we obtain from the correct class, as $q_\phi(z|x,y)$ is weighted by $q_\phi(y|x)$.

Algorithm 1 An EM-based classifier

$q_\phi(y|x) \leftarrow p_\theta(y)$
for iterations $t \in \{1,\dots,T\}$ **do**
  E-Step: Ancestral sampling for GMM
  $z^{(s)} \sim q_\phi(z|x) = \sum_{y=1}^{K} q_\phi(z|x,y)q_\phi(y|x)$
  E-Step: GDA classifier
  $p_\theta(y|z^{(s)}) \leftarrow \eta p_\theta(z^{(s)}|y)p_\theta(y)$
  M-Step: Assign mean confidence to q
  $q_\phi(y|x) \leftarrow p_\theta(y|z) = \frac{1}{S}\sum_{s=1}^{S} p_\theta(y|z^{(s)})$
end for
return $q_\phi(y|x)$

The algorithm yields the classifier $q_\phi(y|x)$, used in the learning objective to estimate $q_\phi(z|x)$. We perform ancestral sampling, initially drawing samples from $q_\phi(y|x)$, then from $q_\phi(z|x,y)$ to approximate $q_\phi(z|x)$ (see Algorithm 1).

Generative Classifier. The generative classifier is built upon a Gaussian discriminant analysis model (GDA) [18] and does not have any additional parameters. Its purpose is to transform the features z from the recognition network and marginalization process into an interpretable class prediction.

During the training of the entire GdVAE, the prior network learns the class-conditional mean $\mu_z(y;\theta) = \mu_{z|y}$ and covariance $\Sigma_z(y;\theta) = \Sigma_{z|y}$ as the parameters of our distribution $p_\theta(z|y) = \mathcal{N}(\mu_z(y;\theta), \Sigma_z(y;\theta))$. We assume conditional independence and decompose the likelihood as $p_\theta(z|y) = \prod_{j=1}^{M} p_\theta(z_j|y)$. In practice, this results in a diagonal covariance matrix $\Sigma_{z|y} = \operatorname{diag}\left(\sigma^2_{z_1|y},\dots,\sigma^2_{z_M|y}\right)$. We use this distribution to determine the likelihood values for the GDA classifier. The class prior $p_\theta(y)$ can be learned either jointly or separately as the final component of the GDA model. Thus, we use the mean values as class prototypes and the covariance to measure the distance to these prototypes.

To infer the class, we apply Bayes' theorem using the detected feature z from the recognition model $p_\theta(y=i|z) = \eta p_\theta(z|y=i)p_\theta(y=i)$, with the normalizer η. For the explanation method, we further assume equal covariance matrices Σ_z

(independent of y), yielding linear discriminants $f^{(i)}(z) = {w^{(i)}}^T z + b^{(i)}$, where the weight and bias are given by $w^{(i)} = \Sigma_z^{-1}\mu_{z|i}$ and $b^{(i)} = -\frac{1}{2}\mu_{z|i}^T\Sigma_z^{-1}\mu_{z|i} + \log p_\theta(y = i)$. For two classes we get $f(z) = f^{(c)}(z) - f^{(k)}(z) = w^T z + b$.

3.2 Counterfactual Explanations (CF)

Instead of directly employing a DNN to define an explainer function $x^\delta = \mathcal{I}_F(x, \delta)$, we generate CFs in the latent space and visualize the outcome using the decoder $\mathcal{I}_F(x, \delta) = g(\mathcal{I}_f(z, \delta))$. Since the discriminant $f(z) = w^T z + b$ of our classifier is linear by construction, we will see that the optimal explainer function is also linear $\mathcal{I}_f(z, \kappa) = z + \kappa\overline{w}$, where the latent vector is adjusted in the direction of $\overline{w} \in \mathbb{R}^M$. Here, $\kappa \in \mathbb{R}$—a tuning knob for data traversal—represents the strength of the manipulation. Our proposed CF methods are shown in Fig. 4a.

1.) Local counterfactuals: A local explanation should meet both consistency and proximity properties. Therefore, the optimal CF z^δ minimizes the distance to the current instance z while ensuring the decision function matches the requested value δ. This involves solving the following constrained optimization problem

$$\mathcal{I}_f(z, \delta) = \arg\min_{z^\delta} \text{dist}(z^\delta, z), \quad \text{subject to } f(z^\delta) = \delta, \tag{9}$$

where dist(.,.) is a distance metric that guarantees *proximity* and the constraint ensures *consistency*. Regardless of whether we choose the common L2-norm [24, 43] or a Riemannian-based metric (Mahalanobis distance) induced by VAEs [5], the solution to Eq. (9) is a linear explainer function

$$\mathcal{I}_f(z, \delta) = z^\delta = z + \kappa\overline{w}, \text{ with } \kappa = \frac{\delta - w^T z - b}{w^T\overline{w}}, \tag{10}$$

where $w = \Sigma_z^{-1}(\mu_{z|c} - \mu_{z|k})$ is the gradient direction of our discriminant. In this approach, any negative value of δ would lead to a change in the class prediction, and $\delta = 0$ corresponds to both classes having equal probability. To simplify user interaction, one can specify the value in terms of a probability using the logit function, such that $\delta = \log\frac{p_c}{1-p_c}$ with $p_c = p(y = c|z^\delta)$.

Using the L2-norm, we obtain the intuitive solution where $\overline{w} = w$ (local-L2). The CF is generated by using the shortest path (perpendicular to the decision surface) to cross the decision boundary (see Fig. 4a). The theoretical analysis on Riemannian manifolds [5] shows that samples close in the latent space with respect to a Riemannian metric lead to close images in terms of the L2-norm, thus optimizing proximity. A Riemannian-based solution using the Mahalanobis distance is $\overline{w} = \Sigma_z w$ (local-M). Training with a spherical covariance $\Sigma_z = \sigma^2 I$ instead of $\Sigma_z = \text{diag}\left(\sigma_{z_1}^2, \ldots, \sigma_{z_M}^2\right)$ yields equivalent functions and therefore equal empirical results for both Riemannian and L2-based CFs. Proofs, assumptions, and implications for non-linear methods are provided in the Supplement.

2.) Global counterfactuals: The second CF approach is to move directly in the direction of the prototype of the opposing class, termed the *counterfactual prototype*. In this scenario, we take a direct path from our current input z to the

CF prototype $\mu_{z|k}$, defining the direction as $\overline{w} = (\mu_{z|k} - z)$, and reuse the local explainer function from Eq. (10).

The local approach minimizes input attribute changes (*proximity*), while global explanations gradually converge to common CF prototypes to reveal the overall model behavior for a category of examples. Both methods maintain the *consistency* property in the latent space. For *realism*, we argue that transitioning directly to the CF prototype or minimizing a distance function is the most effective way to stay within the data distribution, resulting in a natural appearance.

Consistency Loss. Our explainer function implicitly assumes that the encoder and decoder act as inverses of each other. Consequently, it is imperative to ensure that a reconstruction $x^\delta = g(z^\delta)$, based on the latent representation $z^\delta = \mathcal{I}_f(h(x), \delta)$, results in a similar latent representation when encoded once more, i.e., $h(x^\delta) \approx z^\delta$. This alignment is crucial to ensure that the classifier provides the desired confidence when a CF is used as input. To enforce this property, similar to [30, 43, 44], we introduce a tailored consistency loss

$$\mathcal{L}^{con} = \mathbb{E}_{p(\delta)}\left[KL\big(q_\phi(z|x^\delta)||q_\phi(z^\delta|x)\big)\right], \tag{11}$$

where the term addresses classification consistency for generated CF inputs. Essentially, we are probing latent values between the distributions $p_\theta(z|y = c)$ and $p_\theta(z|y = k)$ to optimize for the consistency property. $q_\phi(z^\delta|x)$ is obtained by applying the linear transformation of the explainer function $\mathcal{I}_f(z, \delta)$ to $q_\phi(z|x)$. In other words, we simply shift the mean value and keep the variance. We use both global and local explainer functions to generate training samples. $p(\delta)$ defines the desired perturbation of the latent variable and we use $p(\delta) = \mathcal{U}(-\varepsilon, \varepsilon)$, where ε can be specified in terms of a probability. The final loss is then given by $\mathcal{L} = \mathcal{L}^{gd} + \gamma\mathcal{L}^{con}$, where γ controls the impact of the consistency regularizer.

4 Experiments

The empirical evaluation aims to validate the performance of our model, focusing on two components: the predictive performance of the GdVAE and the quality of the CFs. We present quantitative results of the predictive performance and CFs in Secs. 4.1 and 4.2, along with qualitative results in Sec. 4.3. In the Supplement, we conduct a hyperparameter investigation covering all method parameterizations. This includes exploring the model balance between M1 and M2, consistency loss, and presenting additional quantitative and qualitative results.

Datasets and Implementation. We employ four image datasets: MNIST [31], CelebA [32], CIFAR-10 [29], and the high-resolution dataset FFHQ [25]. Our neural networks are intentionally designed to be compact. For CelebA, the encoder has five convolutional layers and one linear layer for $\mu_z(x, y; \phi)$ and $\Sigma_z(x, y; \phi)$, which define the distribution $q_\phi(z|x, y)$. The decoder's architecture is symmetrical to that of the encoder. Prior encoders use fully connected networks with four layers to compute $\mu_z(y; \theta)$ and $\Sigma_z(y; \theta)$, defining our distribution $p_\theta(z|y)$. All baseline methods employ identical backbones as the GdVAE, and when feasible, publicly available code was adjusted to ensure a fair comparison. See the Supplement for details on datasets, models, and metrics.

Table 2: Predictive performance: Importance sampling (IS), ProtoVAE, and a black-box baseline. Classifier accuracy (ACC) and mean squared error (MSE) of reconstructions (scaled by 10^2) are reported. Mean values and standard deviations are from four training runs with different seeds. †: incl. ProtoVAE's augmentation and preprocessing.

Method	MNIST		CIFAR-10		CelebA - Gender	
	$ACC\% \uparrow$	$MSE \downarrow$	$ACC\% \uparrow$	$MSE \downarrow$	$ACC\% \uparrow$	$MSE \downarrow$
IS [45, 48, 54]	**99.0±0.08**	**1.04±0.01**	55.0±0.59	2.45±0.03	94.7±0.44	1.77±0.08
Ours	**99.0±0.11**	1.10±0.04	**65.1±0.78**	**1.71±0.02**	**96.7±0.13**	**0.91±0.01**
Baseline	99.3±0.04	1.12±0.02	69.0±0.54	1.45±0.01	96.7±0.26	0.82±0.00
ProtoVAE [13]	**99.1±0.17**	1.51±0.23	76.6±0.35	2.69±0.02	96.6±0.24	1.32±0.10
Ours†	98.7±0.05	**0.93±0.01**	**76.8±0.91**	**1.18±0.02**	**96.8±0.04**	**0.71±0.01**

4.1 Evaluation of Predictive Performance

Methodology. For a trustworthy SEM, performance should align with the closest black-box model [13]. Thus, the goal of this evaluation is not to outperform state-of-the-art results on specific datasets but to offer a relative comparison for the GdVAE architecture and various training methods. In all approaches, both the classifier and autoencoder are jointly trained, sharing the same backbone.
Baselines. First, optimal performance for the selected architecture is established using a black-box model, comprising a jointly trained CVAE and classifier as the *baseline*. Next, GdVAE's inference method is evaluated against the leading CVAE technique, *importance sampling (IS)* [45, 48, 54]. Lastly, *ProtoVAE* [13] is referenced as a prototype-per-class VAE benchmark.
Results. The results in Tab. 2 indicate good generalization in classification and reconstruction across MNIST and CelebA. The GdVAE's EM-based inference achieves performance close to the optimal baseline with a separate classifier, except for CIFAR where there is a four-percentage-point gap in accuracy. Comparing our EM and the IS approach suggests that our method is more efficient for higher-dimensional images, benefiting from sampling in the lower-dimensional latent instead of image space. With data augmentation and normalization from ProtoVAE, GdVAE achieves comparable results to ProtoVAE.

Takeaway: The inference procedure of our SEM closely matches the performance of a discriminative black-box model. Furthermore, our method consistently delivers competitive results to state-of-the-art approaches, particularly when applied to higher-dimensional images. The class-conditional GdVAE offers better reconstructions compared to ProtoVAE, the only unconditional model.

4.2 Quantitative Evaluation of CF Explanations

Methodology. The experiments aim to evaluate the quality of CFs regarding *realism*, *consistency*, and *proximity*. *Realism*, as defined in [14, 26] or data consistency [43], refers to the CF images being realistic and capturing identifiable concepts. To measure realism, we employ the Fréchet Inception Distance (FID) [14, 26, 43] as a common metric. Akin to [26], *proximity* is assessed using the mean squared error (MSE) between the CF and the query image.

Table 3: Evaluation of CF explanations using Pearson correlation (ρ_p), ACC, and MSE (scaled by 10^2) for consistency, Fréchet Inception Distance (FID) for realism, and MSE (scaled by 10^2) for proximity. Mean values and standard deviations are from four runs with different seeds. The first and second best results are **bolded** and underlined.

	Method	Consistency			Realism	Proximity
		$\rho_p \uparrow$	$ACC\% \uparrow$	$MSE \downarrow$	$FID \downarrow$	$MSE \downarrow$
MNIST - Binary 0/1	GANalyze [15]	0.84±0.04	5.5±1.3	6.75±1.27	54.89±4.19	6.33±1.73
	UDID [49]	0.85±0.01	1.2±0.3	8.82±0.18	**38.89±2.01**	7.44±0.81
	ECINN [21]	0.93±0.02	33.0±7.5	1.76±0.81	87.25±12.63	**3.47±0.75**
	EBPE [43]	**0.97±0.01**	**44.6±4.3**	**0.50±0.13**	108.94±13.61	25.73±20.69
	C3LT [26]	0.89±0.03	3.6±0.8	6.32±1.39	57.09±10.78	5.83±1.47
	Ours (local-L2)	0.95±0.00	42.9±2.7	0.95±0.11	91.22±11.04	4.58±1.00
	Ours (local-M)	0.95±0.01	**44.6±2.5**	0.87±0.13	89.91±5.58	4.10±0.37
	Ours (global)	0.97±0.01	54.2±4.0	0.55±0.13	125.45±11.32	6.23±0.53
CelebA - Smiling	GANalyze [15]	0.78±0.03	15.2±3.3	5.42±0.97	147.43±19.49	13.47±9.36
	UDID [49]	0.86±0.06	15.8±9.2	4.22±2.17	178.23±75.84	13.73±9.41
	ECINN [21]	0.72±0.21	21.3±9.6	5.68±4.32	95.35±14.48	1.16±0.22
	EBPE [43]	**0.94±0.01**	**41.9±3.1**	**1.22±0.16**	191.67±20.51	1.54±0.06
	C3LT [26]	0.90±0.01	11.8±5.5	3.94±0.66	101.46±11.56	3.97±0.86
	Ours (local-L2)	0.81±0.04	25.0±4.9	3.65±1.06	**85.52±2.37**	0.99±0.02
	Ours (local-M)	0.82±0.05	25.7±5.1	3.51±1.05	85.56±2.39	**0.92±0.03**
	Ours (global)	0.89±0.01	45.9±12.3	2.08±0.54	128.93±4.94	5.81±0.53

The *consistency* property, also known as compatibility [43] or importance [14], is evaluated using mean squared error (MSE), accuracy (ACC), as well as the Pearson correlation coefficient. We create CFs for every image by requesting confidences within the range $p_c \in [0.05, 0.95]$, with a step size of 0.05. The metrics compare the expected outcome of the classifier p_c (desired probability score of CFs) with the actual probability $\widehat{p}_c$ obtained from the classifier for the CF.

Baselines. We employ methods from different designs (see Sec. 2) as baselines with shared backbones. To ensure a fair comparison, we slightly modify methods that originally tackle the simpler consistency task [21, 26] or those intended for unsupervised scenarios [49], aligning them with the consistency defined in Sec. 2. First, we apply generative explanation methods, including GANalyze [15] and UDID [49], while utilizing our pre-trained GdVAE as an autoencoder and classifier. Second, we adapt post-hoc CF methods to be compatible with our GdVAE architecture. We adapt the method from ECINN [21] to approximate our classifier. C3LT [26] is trained to generate CFs for our GdVAE model using a non-linear explainer function instead of our linear one. Finally, EBPE [43] is adjusted to train an encoder and decoder based on the GdVAE architecture and the pre-trained classifier. These approaches are compared to our CF methods.

Results. The results in Tab. 3 reveal performance across diverse datasets in binary classification challenges. Considering that the GdVAE is the sole transparent model, it is essential to bear in mind that most models operate on the GdVAE's pre-regularized latent space (Fig. 3) when interpreting the results. Consequently, with the exception of EBPE, these methods face a less complex

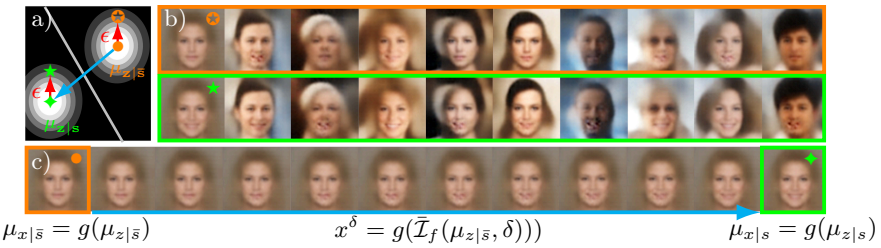
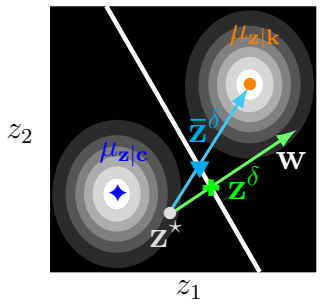


Fig. 3: Regularized latent space. a) Distribution $p_\theta(z|y)$ with class-conditional mean values for not-smiling (orange, ●) and smiling (green, ✦), where $y = \bar{s} = not - smiling$ and $y = s = smiling$. b) Reconstructed random samples for not-smiling (top, orange ✪) and smiling (bottom, green ★), arranged in ascending order of their Mahalanobis distance from left to right. In each column, the Mahalanobis distance is made consistent by adding the same random vector ϵ (red vector in a) to the mean of both classes, aligning samples along isocontours. c) The global explainer function interpolates between class-conditional means along the straight-line path (cyan arrow in a).

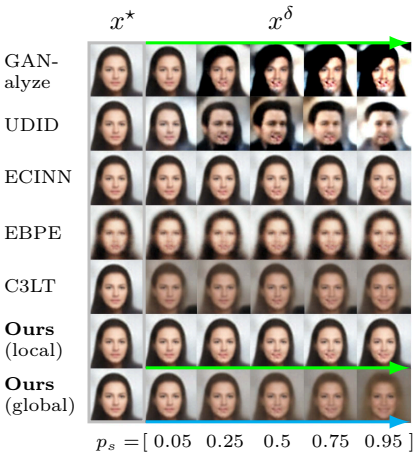
task and should approximate the "true" linear direction of our local CFs post-GdVAE training. It becomes evident that our global CF method exhibits a higher degree of consistency with the classifier, albeit falling short in terms of realism when compared to the local approaches. This divergence is expected as the global method converges toward the mean representation of the CF prototype, producing relatively blurred representations with a notable distance from the query image (poor proximity). However, global CFs effectively uncover the model's overarching decision logic through its prototypes (see Fig. 3c).

Specifically, on the MNIST dataset, our local methods achieve the best or second-best results in all consistency metrics, producing CFs with well-calibrated confidence values. The notably high accuracy values indicate that our methods generate CFs covering the entire confidence range, effectively capturing samples near the decision boundary. However, the realism metric is affected due to the absence of MNIST images near the decision boundary, notably those representing shared concepts of digits 0 and 1 (see Fig. 2). In summary, a favorable trade-off between consistency, realism, and proximity is achieved by EBPE, ECINN, and our local methods. A distinct perspective arises when considering the CelebA dataset, where our local methods excel in achieving optimal results for both proximity and realism, maintaining a low FID score. In terms of consistency metrics like ACC and MSE, our local method (local-M) ranks among the top two performers. Global CFs are distinguished with a separating line in Tab. 3, indicating their deviation from query images by consistently approaching the same prototypes, which effectively reveals biases (e.g., toward female prototypes in CelebA, see Sec. 4.3).

Takeaway: Our SEM, featuring both linear local and global explanations, yields results that stand on par with leading post-hoc explanation techniques such as C3LT and EBPE. Moreover, our model slightly outperforms ECINN across various metrics, with ECINN serving as an optimized post-hoc variant



(a) CF generation. Consider input x with latent vector $z^\star$ and class prediction c. Our first CF method, denoted as z^δ (green ✚), walks along the gradient direction w or the slightly rotated gradient direction $\Sigma_z w$ (omitted for clarity). The second method creates a CF, $\bar{z}^\delta$ (cyan ▼), by moving from $z^\star$ to the prototype $\mu_{z|k}$ (orange ●) of the contrasting class k. Here, we show a solution for $\delta = 0$ where the decision boundary is crossed.



(b) We generate CelebA CFs (x^δ) linearly for the input, with increasing confidence for smiling ($y = s$) from left to right. On the leftmost side, $x^\star$ denotes the reconstruction of the input x. Local denotes both local-M and local-L2 methods, as their images are indistinguishable.

Fig. 4: Left: Counterfactual generation. Right: Counterfactual examples.

of our local CFs. The findings suggest a trade-off between consistency and realism, as no single method excels in all metrics. Notably, as realism decreases (higher FID), consistency (correlation) increases, resulting in different working points for each CF method. For further insights regarding this trade-off between consistency and realism, please refer to the Supplement.

4.3 Qualitative Evaluation

Prototypical Space and Bias Detection. The prototypical space of the GdVAE is shown in Fig. 3. This section's results reinforce GdVAE's transparency through easily comprehensible global explanations and latent space visualization. We achieve this by displaying the decoded prototypes and interpolating between them through our global explainer function (see Fig. 3c). The transparent classifier's prototypes directly uncover biases without the need for quantitative analysis of counterfactuals on simulated datasets, as shown in prior work (e.g., [43]).

Illustrated in Fig. 3c, the classifier's decision on smiling is shaped by female prototypes, revealing a potential bias or data imbalance not observed in our local CFs and other CF methods. The gender bias is exposed by evaluating the smile classifier across hidden attributes (Tab. 4), indicating reduced performance with increased uncertainty in males. Finally, we leverage the generative capabilities of the CVAE

Table 4: CelebA bias.

Attr.	$ACC\% \uparrow$	$MSE \downarrow$
male	89.9±1.37	0.92±0.03
female	91.1±0.33	0.88±0.02

structure, generating samples for various classes. The results are presented in Fig. 3b and organized based on their distance to the corresponding prior. Due to regularization, the latent space preserves the identity of individuals when generating samples for different classes using the same random vector.

CF Explanations. Regarding visual quality, Fig. 4b directly compares our approach with other tested methods, using a random CelebA image. Our local method achieves visual quality comparable to state-of-the-art approaches, yielding results akin to ECINN. C3LT demonstrates smooth outcomes reminiscent of our global method, while EBPE preserves image concepts like our local method but with slight reconstruction variations. The alignment of C3LT with our global CFs, despite its expected approximation of the direction of our local CFs, highlights the advantage of our analytic link between the classifier and CFs.

High-resolution CFs. We showcase our classifier's scalability in more complex scenarios, such as higher resolutions, by embedding it within a pre-trained StyleGAN architecture on the FFHQ dataset. Local CF explanations for smiling with $p_s = 0.99$ are depicted in Fig. 1. Similar to findings with CelebA data, our method retains the background while altering only pertinent attributes.

5 Conclusion

In this paper, we present a novel self-explainable model capable of delivering counterfactual explanations alongside transparent class predictions. Our approach uses a linear classifier in the latent space that utilizes visualizable prototypes for the downstream task. With the known linear structure, we can provide an analytical solution to generate counterfactual images. Our extensive experiments substantiate our method's ability to yield results that are on par with state-of-the-art approaches in terms of consistency, proximity, and realism while maintaining transparency. Furthermore, we illustrate how prototypes offer insight into decision logic and aid in identifying classifier bias. We see our method as a significant step toward the integration of self-explainable models and counterfactual explanation techniques. In contrast to previous work that requires post-hoc analysis for generating counterfactuals, our transparent model constrains the shared latent space to support consistency, proximity, and realism. Finally, resembling C3LT, our approach scales and seamlessly integrates with larger network architectures, as demonstrated on the FFHQ dataset.

Acknowledgments

The authors thank e:fs TechHub GmbH, Germany, and Tongliang Liu's Trustworthy Machine Learning Lab at the University of Sydney for their support. Special appreciation goes to Muyang Li for insightful discussions and Niclas Hüwe for assistance with implementation.

References

1. Agarwal, R., Frosst, N., Zhang, X., Caruana, R., Hinton, G.E.: Neural additive models: Interpretable machine learning with neural nets. In: NeurIPS (2021)
2. Alvarez-Melis, D., Jaakkola, T.S.: Towards robust interpretability with self-explaining neural networks. In: NeurIPS (2018)
3. Bau, D., Zhu, J., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: GAN dissection: Visualizing and understanding generative adversarial networks. In: ICLR (2019)
4. Black, E., Wang, Z., Fredrikson, M.: Consistent counterfactuals for deep models. In: ICLR (2022)
5. Chadebec, C., Allassonniere, S.: A geometric perspective on variational autoencoders. In: NeurIPS (2022)
6. Chang, C.H., Caruana, R., Goldenberg, A.: NODE-GAM: Neural generalized additive model for interpretable deep learning. In: ICLR (2022)
7. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: Deep learning for interpretable image recognition. In: NeurIPS (2019)
8. Chen, R.T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. In: NeurIPS (2018)
9. Ding, Z., Xu, Y., Xu, W., Parmar, G., Yang, Y., Welling, M., Tu, Z.: Guided variational autoencoder for disentanglement learning. In: CVPR (2020)
10. Do, M.: Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. IEEE Signal Processing Letters **10**(4), 115–118 (2003)
11. Esser, P., Rombach, R., Ommer, B.: A disentangling invertible interpretation network for explaining latent representations. In: CVPR (2020)
12. Falck, F., Zhang, H., Willetts, M., Nicholson, G., Yau, C., Holmes, C.C.: Multi-facet clustering variational autoencoders. In: NeurIPS (2021)
13. Gautam, S., Boubekki, A., Hansen, S., Salahuddin, S.A., Jenssen, R., Höhne, M.M., Kampffmeyer, M.: Protovae: A trustworthy self-explainable prototypical variational model. In: NeurIPS (2022)
14. Ghandeharioun, A., Kim, B., Li, C.L., Jou, B., Eoff, B., Picard, R.: DISSECT: Disentangled simultaneous explanations via concept traversals. In: ICLR (2022)
15. Goetschalckx, L., Andonian, A., Oliva, A., Isola, P.: Ganalyze: Toward visual definitions of cognitive image properties. In: ICCV (2019)
16. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: ICML (2019)
17. Guyomard, V., Fessant, F., Guyet, T., Bouadi, T., Termier, A.: Vcnet: A self-explaining model for realistic counterfactual generation. In: Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 437–453. Springer International Publishing, Cham (2023)
18. Haselhoff, A., Kronenberger, J., Küppers, F., Schneider, J.: Towards black-box explainability with gaussian discriminant knowledge distillation. In: CVPRW (2021)
19. Hauberg, S.r., Freifeld, O., Black, M.: A geometric take on metric learning. In: NeurIPS (2012)
20. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: ICLR (2017)
21. Hvilshøj, F., Iosifidis, A., Assent, I.: Ecinn: Efficient counterfactuals from invertible neural networks. In: BMVC (2021)

22. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: NeurIPS (2020)
23. Jacob, P., Zablocki, É., Ben-Younes, H., Chen, M., Pérez, P., Cord, M.: STEEX: steering counterfactual explanations with semantics. In: ECCV (2022)
24. Jeanneret, G., Simon, L., Jurie, F.: Diffusion models for counterfactual explanations. In: ACCV (2022)
25. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)
26. Khorram, S., Fuxin, L.: Cycle-consistent counterfactuals by latent transformations. In: CVPR (2022)
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
28. Kingma, D.P., Mohamed, S., Jimenez Rezende, D., Welling, M.: Semi-supervised learning with deep generative models. In: NeurIPS (2014)
29. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research) `http://www.cs.toronto.edu/~kriz/cifar.html`
30. Lang, O., Gandelsman, Y., Yarom, M., Wald, Y., Elidan, G., Hassidim, A., Freeman, W.T., Isola, P., Globerson, A., Irani, M., Mosseri, I.: Explaining in style: Training a gan to explain a classifier in stylespace. In: ICCV (2021)
31. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). `https://doi.org/10.1109/5.726791`
32. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
33. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.S.: The variational fair autoencoder. In: ICLR (2016)
34. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: NeurIPS (2017)
35. Lundstrom, D.D., Huang, T., Razaviyayn, M.: A rigorous study of integrated gradients method and extensions to internal neuron attributions. In: ICML (2022)
36. Parekh, J., Mozharovskyi, P., d'Alché Buc, F.: A framework to learn with interpretation. In: NeurIPS (2021)
37. Plumerault, A., Borgne, H.L., Hudelot, C.: Controlling generative models with continuous factors of variations. In: ICLR (2020)
38. Ren, X., Yang, T., Wang, Y., Zeng, W.J.: Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In: ICLR (2021)
39. Rhodes, T., Lee, D.: Local disentanglement in variational auto-encoders using jacobian l_1 regularization. In: NeurIPS (2021)
40. Samangouei, P., Saeedi, A., Nakagawa, L., Silberman, N.: Explaingan: Model explanation via decision boundary crossing transformations. In: ECCV (2018)
41. Seitzer, M.: pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid` (August 2020), version 0.3.0
42. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: CVPR (2020)
43. Singla, S., Pollack, B., Chen, J., Batmanghelich, K.: Explanation by progressive exaggeration. In: ICLR (2020)
44. Sinha, S., Dieng, A.B.: Consistency regularization for variational auto-encoders. In: NeurIPS (2021)
45. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: NeurIPS (2015)

46. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
47. Vandenhende, S., Mahajan, D., Radenovic, F., Ghadiyaram, D.: Making heads or tails: Towards semantically consistent visual counterfactuals. In: ECCV (2022)
48. van de Ven, G.M., Li, Z., Tolias, A.S.: Class-incremental learning with generative classifiers. In: CVPRW (2021)
49. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the gan latent space. In: ICML (2020)
50. Wang, J., Liu, H., Wang, X., Jing, L.: Interpretable image recognition by constructing transparent embedding space. In: ICCV (2021)
51. Wang, Y., Wang, X.: Self-interpretable model with transformation equivariant interpretation. In: NeurIPS (2021)
52. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. IJCV **129**(5), 1451–1466 (2021). https://doi.org/10.1007/s11263-020-01429-5
53. Yu, C., Wang, W.: Diverse similarity encoder for deep gan inversion. arXiv preprint arXiv:2108.10201 (2022), https://arxiv.org/abs/2108.10201
54. Zhang, C., Zhang, K., Li, Y.: A causal view on robustness of neural networks. In: NeurIPS (2020)