Interpretable and Efficient Counterfactual Generation with Disentangled Variational Autoencoders

Cesare Barbera[1,2] and Andrea Passerini[1]

[1] University of Trento, Povo TN 38123, IT
[2] Univeristy of Pisa, Pisa PI 56126, IT
{cesare.barbera,andrea.passerini}@unitn.it

Abstract. Among the various forms of post-hoc explanations for black box models, counterfactuals are among the most appealing for their intuitiveness and effectiveness. Long-standing issues in the field of counterfactual explanations regard the efficiency of the counterfactual search process, the likelihood of generated instances or their interpretability and in some cases even the validity of the explanations. In this work we present a generative framework capable of addressing all these issues. Our method leverages disentangled variational autoencoders to achieve two complementary objectives, namely generating high-quality instances and encouraging label disentanglement to gain full control over the decision boundary. This allows the model to sidestep expensive gradient based optimization to generate counterfactuals, which are instead directly generated according to the adversarial distribution. Preliminary results assess the effectiveness of the training procedure, the efficiency of the explanatory pipeline and the quality and interpretability of the explanations.

Keywords: Counterfactual Explanations · Generative XAI · Disentanglement

1 Introduction

Explainable AI is a field of research that arises from the need of transparency and to improve understanding of what are known as black-box models [13]. With the goal of explaining the inner workings of deep-learning models, researchers have provided users with many different techniques of post-hoc explanations. Among these, counterfactuals consist of instances describing the necessary changes in input features that alter the prediction to a predefined output [23], and are especially appealing for a human decision maker [9]. Counterfactual explanations should carry the following properties: i) *validity* – the model prediction on the counterfactual instance needs to follow a predetermined class; ii) *sparsity* – the perturbation applied to the original instance should be sparse; iii) *interpretability* – the explanatory instance should be interpretable, iv) *likeliness* – the explanation should be representative of the adversarial class distribution.

Despite the appeal of counterfactual explanations, existing approaches have struggled in satisfying the desired properties, especially likeliness [26, 4], actionability [12, 5] or sparsity [11] of the counterfactual being generated. Efficiency in generation is another major problem of existing solutions [7, 32, 17]. Simultaneously, there has been a noticeable growth in popularity of generative models in XAI with the aim to increase the quality of generated explanations [30]. Inspired by this, we propose a generative framework for counterfactual explanations that satisfies the desired properties while being computationally efficient, so as to allow real-time counterfactual generation. In a nutshell, our framework leverages a disentangled variational autoencoder to learn class-specific latent representations, which in turn allows the generation of counterfactuals by simply trading-off the likelihood of the explanation according to the adversarial distribution with its distance from the instance to explain. Likeliness of the output is assured by the underlying generative model, validity is guaranteed by the explicit modeling of the decision boundary between classes, while sparsity is encouraged by combining label-relevant latent dimensions with label-irrelevant ones which are shared among classes. Finally, efficiency is achieved by directly generating counterfactuals according to the adversarial distribution, thus sidestepping expensive gradient based optimization procedures. A preliminary experimental evaluation confirms the effectiveness of the proposed solution, which generates insightful, interpretable and valid counterfactuals in real-time for the popular fashionMNIST dataset.

2 Related Work

Contrastive explanations Contrastive explanations aim at justifying a choice by rejecting the other viable options. Throughout the years various techniques have been proposed to achieve this [27, 34, 15, 22] but the most popular option is counterfactuals. Since the use of Deep Generative Models, such as Generative Adversarial Networks (GANs) [10] and VAEs [19, 28], has been proposed to explain models choices, the most common procedure is to gradually twist the input in order to retrieve the most meaningful interpretable changes like in [8, 16, 21, 24, 29, 31]. Such operations can be computationally expensive and require complex gradient-based optimizations like for the case of [4], where concepts extracted from a disentangled VAE are central to the explanatory process. The proposal of [25] also leverages a disentangled VAE, but a classifier parameterized by a neural network is directly applied to its latent space. Their approach is effective but restricted to problems with a limited number of dimensions. In this work, we show how to overcome these limitations by directly exploiting the data distribution learned by the generative model.

Generative AI and latent disentanglement Disentanglement plays a central role in the framework we propose, in terms of both learning disentangled latent representations and label disentanglement in the latent space. Disentangled feature representations, or high level generative factors in disjoint subsets of the fea-

ture dimensions, carry many desirable properties such as intervention and interpretability [20, 1]. Due to the inherent trade-off between reconstruction quality and disentanglement [14], existing approaches [20, 2, 18] incorporate additional regularization components or derive alternative ELBO formulations. On a different note, [36] define Variation Predictability, a constraint encouraging disentanglement, which is directly optimized combining VAEs and GANs. Not surprisingly a body of works exploiting classification losses to encourage a disentangled latent representations already exists [3, 6, 35]. However, the first two [3, 6] are conceived for classification and cannot generate new instances, while the last [35] can perform generation but, in contrast with our approach, does not explicitly learn to classify the latent representations it reconstructs and that our method uses for the explanations.

3 Method Overview

In this section we present an overview of the methodology we propose. Our framework is centered around a disentangled VAE equipped with a label-relevant label-irrelevant approach to simultaneously learn a generative process and a classification task. This allows class distributions to guide both both the label predictions and their explanatory process. The novel technique for counterfactual generation we present operates under the assumption that data follows a mixture of Gaussian distributions, and it consists of a two step process: i) identification of a set of candidate counterfactuals according to a predefined set of rules; ii) extraction of the expected value of the set under the adversarial distribution as the generated counterfactual. This framework aims at capitalizing on the following advantages:

- *interpretability*: shaping the distribution of the data in the latent space increases the model transparency by controlling the complexity of the decision boundary;
- *Validity*: the assumptions of the predictive model are coherent with the ones of the chosen explanatory technique, allowing full control over the predictive mechanism;
- *Likeliness*: learning the latent-space data distribution allows for fast, efficient and high quality counterfactuals generation with the methodology we propose.

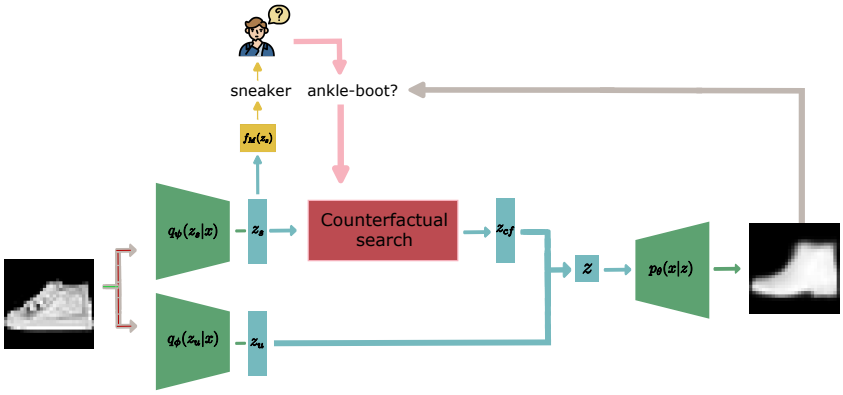


Fig. 1: Overall pipeline of the proposed counterfactual generation framework.

The full interactive explanatory pipeline, shown in Figure 1, can be divided in three main steps: an encoding step, a counterfactual search step and a decoding step.

The enconding step consists in extracting the label-relevant and label-irrelevant encodings of the input instance. The first set of latents is used for classification and the output of the model is presented to the user. The second set of latents is instead momentarily stored. This step finally concludes with the user (possibly) formulating a counterfactual query.

The counterfactual search receives the user-defined counterfactual class as input together with the label-relevant latents, and finds a transformation of the latent input such that the model prediction on this novel instance corresponds to the counterfactual class. This step concludes returning the latent vector that optimizes the likelihood of the explanation and the distance between the counterfactual and the original instance.

The final step takes as input the newly found vector that according to the model belongs to the counterfactual class and the label-irrelevant latent representation of the instance to explain. These are concatenated and fed to the decoder to generate a counterfactual in the input space which will be returned as final explanation.

4 The Generative Model

First we will introduce the variational generative model learning class-specific latent distributions. Our approach builds on the label-relevant/irrelevant VAE of [35] and, as mentioned in the related work, we propose a method to extend

with equivalent performance the learned classier to the latent representations used for reconstruction and exploited by our explanatory technique. We stick to Gaussian classification, instead of neural-network parametrization [25], to exploit the properties of such framework when generating counterfactuals. For these reasons, we derive an alternative ELBO formulation that, when minimized together with the classification loss we propose, allows to efficiently shape the latent space as a mixture of label-specific Gaussians, obtaining good classification performance while encouraging latent regularization. The architecture of the proposed model is shown in Figure 2. In the following we provide a detailed description of its elements.

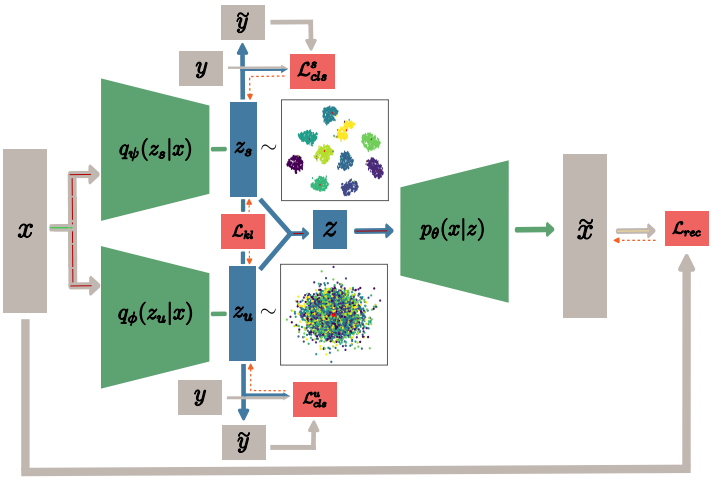


Fig. 2: The model architecture. First inputs are encoded with two separate modules to extract label-relevant (z_s) and label-irrelevant (z_u) latent dimensions. z_s are then used to compute parameters of a mixture of Gaussians. Such parameters should allow label disentanglement which is encouraged by $\mathcal{L}^s_{cls}$. On the other hand for z_u this should not be possible and $\mathcal{L}^u_{cls}$ discourages it penalizing predicted class probabilities deviating from a uniform distribution. Finally, through reparametrization-trick, the latent z to reconstruct is sampled and fed to the decoder. Its output is compared with the original instance and $\mathcal{L}_{rec}$ encourages fidelity in the reconstructions.

4.1 Background

A VAE is a type of parametric model following an encoding $q_\phi(z|x)$ and decoding $p_\theta(x|z)$ mechanism trained with the goal of maximizing likelihood of evidence.

Such quantity is maximized through its lower bound (ELBO):

$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{kl}}(q_\phi(z|x) \parallel p(z)) \tag{1}$$

The encoder is parameterized by ϕ and the decoder by θ. According to such formulation, $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ is the reconstruction loss, which encourages encoded inputs to be decoded with fidelity, and $D_{\mathrm{kl}}(q_\phi(z|x) \parallel p(z))$ is the Kullback-Leibler divergence between the output of the recognition model $q_\phi(z|x)$ and the prior latent distribution $p(z)$. The former is extracted from the encoder which returns mean $\mu_\phi(x)$ and variance $\Sigma_\phi(x)$ parameters for every input x, while the latter is typically modelled as a standard Gaussian.

The ELBO objective can be extended to incorporating classification terms as in [35], with the idea of disentangling the latent space via label supervision. With respect to the standard VAE, an extra recognition model is employed such that the latent representation is split in two different sub-spaces z_s and z_u. In practice the encoders $q_\psi(z_s|x)$ and $q_\phi(z_u|x)$ map the input x to label-relevant (z_s) and label-irrelevant (z_u) latent codes respectively, so to ensure label disentanglement with z_s and to discourage it with z_u.

In conclusion we briefly introduce the Gaussian mixture framework of [33]. They propose to apply to the latent representation z_i of instance x_i with label y_i a loss made of two components: a Gaussian classification term and a likelihood regularization term:

$$\mathcal{L}_{GM} = -\frac{1}{N}\sum_c \mathbb{I}(y_i = c) \sum_i \underbrace{\log \frac{\mathcal{N}(z_i; \mu_{y_i}, \Sigma_{y_i})p(y_i)}{\sum_c \mathcal{N}(z_i; \mu_c, \Sigma_c)p(c)}}_{\mathcal{L}_{cls}} + \underbrace{N \log \mathcal{N}(z_i; \mu_{y_i}, \Sigma_{y_i})}_{\mathcal{L}_{lkd}} \tag{2}$$

where mean μ_c and variance Σ_c parameters are encoding statistics gradually computed during training. This loss is commonly used to encourage label disentanglement in the label-relevant latent space.

4.2 An ELBO for Label Disentanglement

In this section we present the formulation of the ELBO we maximize, and highlight a problem with the current approaches to its computation that inhibits the simultaneous learning of a classification model that effectively disentangles the label-relevant latent representations used for reconstruction.

Proposition 1 (ELBO). *Given the joint distribution over input and latent space:*

$$p(x, z_s, z_u) = \sum_y p_\theta(x|z_s, z_u)p(z_s, y)p(z_u)$$

and given that $p(z_s, y) = p(z_s|y)p(y)$ *and, assuming conditional independence,* $q_\psi(z_s, y|x) = q_\psi(z_s|x)p(y|x)$ *where* $p(y|x)$ *is the one-hot encoding of the label,*

the ELBO of label-relevant/irrelevant VAEs [35] can be written as:

$$\begin{aligned} ELBO \; = & \underbrace{\mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\Big[log\; p_\theta(x|z_s,z_u)\Big]}_{\mathcal{L}_{rec}} - \underbrace{D_{kl}\left(q_\phi(z_u|x) \parallel p(z_u)\right)}_{\mathcal{L}_{kl_u}} \\ & - \underbrace{D_{kl}\left(q_\psi(z_s|x)p(y|x) \parallel p(z_s|y)\right)}_{\mathcal{L}_{kl_s}} \end{aligned} \tag{3}$$

Proof can be found in appendix A. The first term $\mathcal{L}_{rec}$ is optimized by minimizing the squared error between the input and a single-sample Monte-Carlo estimation of the reconstruction. The second term $\mathcal{L}_{kl_u}$ has a closed form solution by setting the label irrelevant prior to an isotropic Gaussian $p(z_u) \sim N(0, I)$. The last term $\mathcal{L}_{kl_s}$ has a closed form solution as long as $q_\psi(z_s|x) \sim \mathcal{N}(\mu_\psi(x), \Sigma_\psi(x))$ and $p(z_s|y) \sim \mathcal{N}(\mu_{z_s|y}, \Sigma_{z_s|y})$, under diagonal covariance matrices assumption.

The original label-relevant/irrelevant VAE approach [35] reduces the $\mathcal{L}_{kl_s}$ term to a simpler log-likelihood term under the assumption that $\Sigma_\psi(x) \to 0$:

$$\mathcal{L}_{kl_s} = -\log \mathcal{N}(\mu_\psi(x); \mu_y, \Sigma_y)$$

where μ_y and Σ_y are the learned mean and variance parameters of the latent distribution for the label of x. Such formulation disregards variance regularisation for the label relevant latents z_s and, if this is coupled with a classification loss applied directly on $\mu_\psi(x)$, the framework becomes sub-optimal for our explanatory purposes. Indeed, we require a Gaussian classifier with equivalent performance on the deterministic and sampled representations, as these are actually used for reconstruction and leveraged to generate explanations, but the latter now have arbitrary variance. For this reason we disregard this kl-divergence formulation and take variance regularisation into account.

4.3 A Training Loss Combining Classification and Regularization

According to our learning objective, a Gaussian classifier implemented on the latent encodings should achieve very accurate results on both deterministic and stochastic latent representations and we also exploit the properties of the Gaussian-mixture loss in Eq.2 to coordinate the efforts of classification and regularization. We motivate still applying this framework to the deterministic latent representations $\mu_\psi(x)$ by proving that in such condition the likelihood regularization component of $\mathcal{L}_{GM}$ is inherently computed in $\mathcal{L}_{kl_s}$ and by showing that the learned decision boundary extends with equivalent performance to the sampled latents as their variance tends to 0.

Proposition 2 (Regularized deterministic latent classification). *Adding a Gaussian classification loss computed on the deterministic output of the encoder $\mu_\psi(x)$ to the kl-divergence regularization term is equivalent to implementing the Gaussian mixture loss framework coupled with an additional variance*

regularization term:

$$\mathcal{L}_{kl_s} + \mathcal{L}^s_{cls} = \mathcal{L}_{GM} + \mathcal{L}_{var} \tag{4}$$

where $\mathcal{L}^s_{cls}$ is the Gaussian classification loss and $\mathcal{L}_{var}$ is the additional variance regularization term.

We invite readers to refer to appendix for proof. Such result allows to operate efficiently in the GM loss framework which is a corroborate approach to classify with latent features regularization. The second great advantage is not dealing with the additional noise due to latent sampling while learning the task. On a final note, variance regularization does not hinder classification performance as it is neural-network parameterized separately from the mean.

We further support our proposal by showing that the learned decision boundary on the deterministic encodings applies with equivalent performance to the stochastic encodings by proving that the expected label assigned to a stochastic latent is the same of its corresponding deterministic representation.

Proposition 3 (Noise invariant label assignment). *Let $x^i \in \mathcal{X}$ be an instance, $\mu_\psi(x^i)$ and $z_s \sim N(\mu_\psi(x^i), \Sigma_\psi(x^i))$ its deterministic and stochastic encodings respectively and $y^*_d = f_M(\mu_\psi(x^i))$ and $y^*_{st} = f_M(z_s)$ its label as predicted by the latent Gaussian classification model $f_M(\cdot)$ applied to its deterministic and stochastic encoding respectively. The expectation of y^*_{st} over the stochastic encodings is the label of the deterministic encoding:*

$$\mathbb{E}_{z_s}[f_M(z_s)] = f_M(\mu_\psi(x^i)) \tag{5}$$

Proof can be found in appendix. This result shows that a model trained to classify the deterministic encodings can be seamlessly used to classify the stochastic encodings used for reconstruction. On a final note, such results relies on the central limit theorem and therefore justifies the additional variance regularisation term introduced in our loss.

4.4 The Training Algorithm

The final loss function to be optimized by the model is a combination of the elements regarding the label relevant ($\mathcal{L}_s$) and the label-irrelevant ($\mathcal{L}_u$) branches and the reconstruction loss ($\mathcal{L}_{rec}$).

The loss of the label-relevant branch composed of the Gaussian-mixture loss framework and the additional variance regularizer:

$$\mathcal{L}_s = \mathcal{L}_{GM} + \lambda_s \mathcal{L}_{var}$$

Concerning the label-irrelevant loss, common practice to ensure label independence for the label-irrelevant encodings is to use an adversarial classifier and feed it encodings such that the label is not predictable. We instead propose to

stick to the mixture of Gaussian classification framework, and directly apply Gaussian classification to the output of the label-irrelevant encoder, with the only difference that the posterior class probabilities now should follow a uniform distribution:

$$\mathcal{L}^u_{cls} = -\frac{1}{n}\sum_i \sum_y \frac{1}{|\mathcal{Y}|} \log \frac{\mathcal{N}(\mu_\phi(x_i); \mu_{z_s|y}, I)p(y)}{\sum_y \mathcal{N}(\mu_\phi(x_i); \mu_{z_s|y}, I)p(y)}$$

The loss of the label-irrelevant branch is a combination of the adversarial Gaussian classification loss and the kl-divergence term (Eq. 3):

$$\mathcal{L}_u = \lambda_u \mathcal{L}_{kl_u} + \mathcal{L}^u_{cls}$$

The final model loss is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_u + \mathcal{L}_s \tag{6}$$

Algorithm 1 shows the pseudo-code of the training algorithm, which consists of the following steps: 1) an instance and the corresponding output are sampled from the dataset; 2) the instance is encoded separately with the label-relevant and label-irrelevant branches; 3) the two kl-divergence terms are computed and, with the supervision of the label, the adversarial classification loss and the classification loss are computed; 4) stochastic representations are sampled and concatenated; 5) the latent representation is decoded and the reconstruction loss is computed; 6) the parameters of the encoders and the decoder are updated with a gradient step. The process iterates until convergence.

5 Counterfactual Generation

In the previous section we showed how to train a deep generative model with a Gaussian classifier that labels instances according to their label-relevant latent representation. In this section, we present our proposal to generate counterfactuals explaining the predictions to a human users.

At a high level, the procedure works as follows. Let x' be an instance to be predicted. The predicted label y^* is computed by feeding x' to the encoders, sampling $z'_s \sim \mathcal{N}(\mu_\psi(x'), \Sigma_\psi(x'))$ and $z'_u \sim \mathcal{N}(\mu_\phi(x'), \Sigma_\phi(x'))$ and computing $y^* = f_M(z'_s)$. The user is provided with the predicted label. In case of disagreement, the user can provide an alternative label y_{cf}. In this case, a counterfactual explanation x'_{cf} is generated to show how the instance should change in order to be predicted as having label y_{cf}. The counterfactual generation consists in computing a latent instance z'_{cf} such that $f_M(z'_{cf}) = y_{cf}$ that optimizes the trade-off between the likelihood of z'_{cf} according to the adversarial distribution and the latent distance between z'_{cf} and z'_s. The counterfactual instance x'_{cf} is then obtained by decoding the concatenation of z'_u with z'_{cf}.

Algorithm 1 Training Algorithm

Require: ψ, ϕ, and π the initial parameters of ENC_s, ENC_u and DEC; n the number of iterations; λ_u and λ_s the weights of regularization terms.

while not convergence **do**
  for $i = 0$ **to** n **do**
    Sample $\{x, y\} \sim \mathcal{D}$
    $\mu_{z_s|x}, \sigma^2_{z_s|x} \leftarrow ENC_s(x)$
    $\mu_{z_u|x}, \sigma^2_{z_u|x} \leftarrow ENC_u(x)$
    $\mathcal{L}_s \leftarrow \mathcal{L}_{GM} + \lambda_s \mathcal{L}_{var}$
    $\mathcal{L}_u \leftarrow \lambda_u \mathcal{L}_{kl_u} + \mathcal{L}^u_{cls}$
    Sample $\{z_s\} \sim \mathcal{N}(\mu_\psi(x), \Sigma_\psi(x))$
    Sample $\{z_u\} \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$
    $z \leftarrow \textsc{Concat}(z_s, z_u)$
    $\tilde{x} \leftarrow DEC(z)$
    $\mathcal{L} \leftarrow \mathcal{L}_{rec} + \mathcal{L}_u + \mathcal{L}_s$
    $\psi, \phi, \pi \overset{+}{\leftarrow} -\nabla_{\psi,\phi,\pi}\mathcal{L}$
  end for
end while

With regard to the counterfactual search process, in order to optimize latent distances, we define a set, called counterfactual candidates, containing the points that for every possible value of distance between z'_{cf} and z'_s are minimally distant to the adversarial mean and classified as the requested label. Finally, to pick an instance from this set without explicitly stating a distance value, we compute the expected value of these candidates according to the adversarial distribution. These steps are further detailed in the following.

5.1 Counterfactual Candidates

In the following section we describe the formal properties of a candidate counterfactual.

Definition 1 (properties of counterfactual candidates). *Given an instance to explain x' with latent encoding z'_s predicted as class y^* with distribution centroid μ_{y^*}, an instance z'_{cf} belongs to the set of counterfactual candidates $\mathcal{C}$ for the label y_i with centroid μ_{y_i}, if for any point z in the latent space $\mathbb{R}^k$, given the following properties:*

$$\mathcal{P}_1 : \underset{y}{\operatorname{argmin}} \parallel z - \mu_y \parallel = y_i$$
$$\mathcal{P}_2 : \parallel z - z'_s \parallel < \parallel z'_{cf} - z'_s \parallel \wedge \parallel z - \mu_{y_i} \parallel \leq \parallel z'_{cf} - \mu_{y_i} \parallel$$
$$\mathcal{P}_3 : \parallel z - z'_s \parallel \leq \parallel z'_{cf} - z'_s \parallel \wedge \parallel z - \mu_{y_i} \parallel < \parallel z'_{cf} - \mu_{y_i} \parallel$$

Property 1 and 2 or property 1 and 3 are never simultaneously satisfied, or:

$$\forall z'_{cf} \in C, \nexists\, z \in \mathbb{R}^k : (\mathcal{P}_1 \wedge \mathcal{P}_2) \vee (\mathcal{P}_1 \wedge \mathcal{P}_3) \tag{7}$$

The first condition ensures that the candidate counterfactual is always predicted as the query class removing any form of validity issue, while the second and third conditions ensure the non existence of a strictly better counterfactual.

It is straighforward to see that all the points that satisfy the first condition and lie on the segment S_1 from z'_s to μ_{y_i} are counterfactual candidates. In addition, we have that the decision boundary for class y^* and y_i is the plane P defined as:

$$P : (\mu_{y^*} - \mu_{y_i}) \cdot (z - \frac{\mu_{y^*} + \mu_{y_i}}{2}) = 0.$$

Now, if $S_1 = \{(1-t)z'_s + t\mu_{y_i} \mid t \in [0,1]\}$, the intersection I between S_1 and P is:

$$I = (1-t^*)z'_s + t^*\mu_{y_i} : t^* = \frac{(\mu_{y^*} - \mu_{y_i}) \cdot (\frac{\mu_{y^*}+\mu_{y_i}}{2} - z'_s)}{(\mu_{y^*} - \mu_{y_i}) \cdot (\mu_{y_i} - z'_s)}$$

Finally we define the orthogonal projection of z'_s on P as $\textsc{Proj}_P(z'_s)$.

Proposition 4 (Set of counterfactual candidates). *Given an instance to explain x' with latent encoding z'_s predicted as class y^*, the set of counterfactual candidates $\mathcal{C}$ for label y_i consists of:*

1. *the points on the segment from z'_s to μ_{y_i}*

$$S_1 = \{(1-t)z'_s + t\mu_{y_i} \mid t \in [0,1]\} \tag{8}$$

2. *the points on the segment connecting the intersection between S_1 and the decision boundary with the closest point to z'_s predicted as y_i*

$$S_2 = \{(1-t)I + t\textit{Proj}_P(z'_s) \mid t \in [0,1]\} \tag{9}$$

Please refer to the appendix for the proof. A graphical representation of the set of counterfactual candidates for an instance can be found in Figure 3.

5.2 Counterfactual as Expectation over Candidates

In the following section we define a technique to compute the expected value of the counterfactual candidates and suggest to return it as a counterfactual explanation. We argue that such counterfactual intrinsically optimizes the trade-off between the likelihood of the explanation and the distance from the instance to explain. Intuitively this is obtained by pushing the explanation away from the adversarial mean according to the probabilities of the other candidates, which are the weights in the expected value computation. We show that computing such expectation has no closed form solution and a large number of samples from a multivariate normal distribution is necessary. We then derive specific conditions under which such estimate can be reduced to a fast and efficient sampling from a univariate distribution.

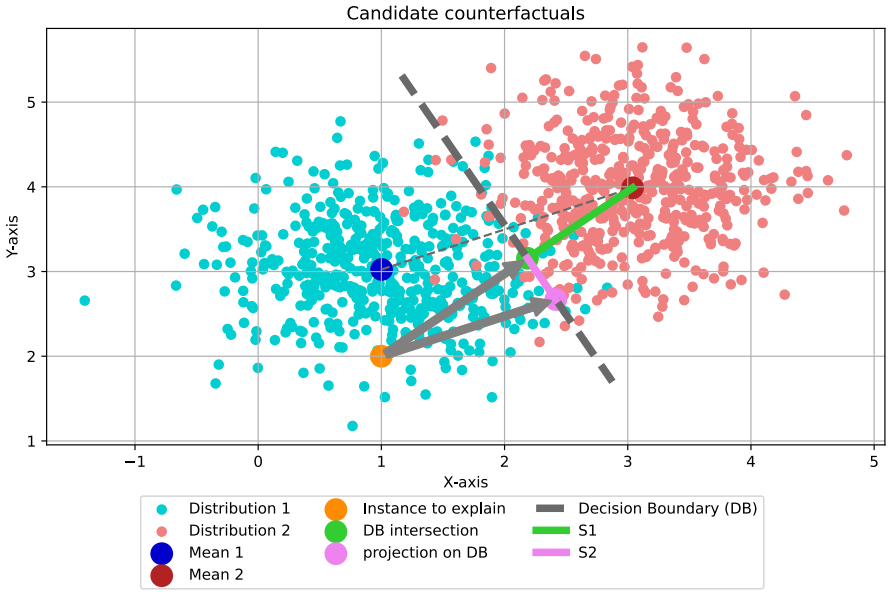


Fig. 3: The set of candidate counterfactuals for a random instance to explain lay on segments S_1 and S_2 as per Proposition 4.

In our derivations we treat expected value computations separately for S_1 and S_2, and return a density-based weighted sum of the two as the final counterfactual:

$$
\begin{aligned}
&z'_{cf_1} = \mathbb{E}_{S_1}[z] \;;\; z'_{cf_2} = \mathbb{E}_{S_2}[z] \\
&z'_{cf} = w_1 z'_{cf_1} + w_2 z'_{cf_2} \\
&\text{with } w_1 = \frac{\mathcal{N}(z'_{cf_1}; \mu_{y_i}, I)}{\mathcal{N}(z'_{cf_1}; \mu_{y_i}, I) + \mathcal{N}(z'_{cf_2}; \mu_{y_i}, I)} \text{ and } w_2 = 1 - w_1
\end{aligned}
$$

Given a generic segment S from a to b as: $S = \{Z(t) = (1-t)a + tb \mid t \in [0,1]\}$ and a density function f_Z, the expected value of the elements of the segment can be expressed as:

$$
\mathbb{E}_S[z] = \frac{\int_0^1 Z(t) f_Z(Z(t)) dt}{\int_0^1 f_Z(Z(t)) dt} = \frac{\int_0^1 \Big((1-t)a + tb\Big) f_Z\Big((1-t)a + tb\Big) dt}{\int_0^1 f_Z\Big((1-t)a + tb\Big) dt} \tag{10}
$$

where the denominator assures that the density integrates to one over domain of the segment.

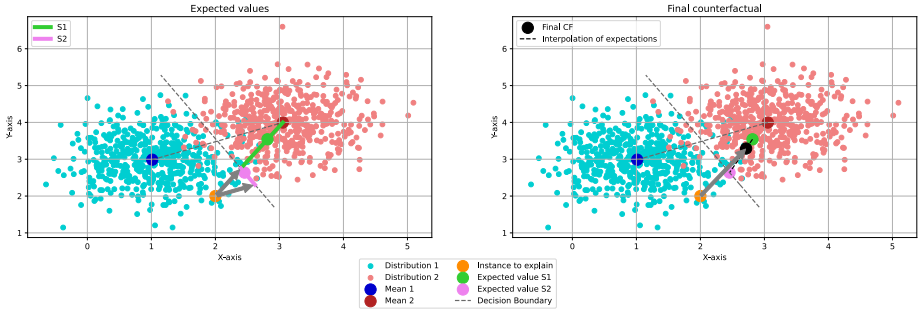


Fig. 4: Expected counterfactual visualized. (l) The expectations along segments S_1 and S_2 are found; (r) The interpolation with the relative densities as weights is the final output.

5.3 Efficient Computation of the Counterfactual

The integral for computing the expectation (Eq.10) has no closed-form solution and requires numerical methods to estimate. Sampling based methods like Monte Carlo Integration require a considerable number of samples to produce accurate estimates, as the density of points vanishes as the dimensions of the distributions increase. In order to speed-up the estimation we propose an alternative sampling based technique that achieves accurate results while being computationally efficient.

Proposition 5 (Expectation along a segment parallel to an axis). *Let* $a = (c, c, ..., c, a_k)$ *and* $b = (c, c, ..., c, b_k) \in \mathbb{R}^k$*, be two points aligned along the last axis. Let* $S = \{(1-t)a + tb \mid t \in [0,1]\}$ *be the segment connecting them, and* $Z(t) = (1-t)a_k + t(b_k)$ *the function of the last component of the segment. In addition let* $f_Z(z) = f_{Z_1,Z_2,...,Z_k}(z)$ *the density function of the underlying distribution of the expectation. The expected value of the elements in* S *according to an isotropic Gaussian is a vector with unchanged components except for the last one. This is the expected value of a univariate distribution in the interval* $[\min(a_k, b_k), \max(a_k, b_k)]$*:*

$$\mathbb{E}_S[z] = \left(c, c, ..., c, \frac{\int_0^1 Z(t) f_{Z_k}(Z(t)) dt}{\int_0^1 f_{Z_k}(Z(t)) dt} \right) \tag{11}$$

Please refer to the appendix for the proof. Intuitively, the expected value of the elements in a segment parallel to the last axis keeps all components intact expect the last one. [3] This expectation still has no closed form solution, but it is much cheaper to estimate as it requires univariate samples only.

[3] The choice of the last axis is arbitrary, and the result clearly holds for any axis.

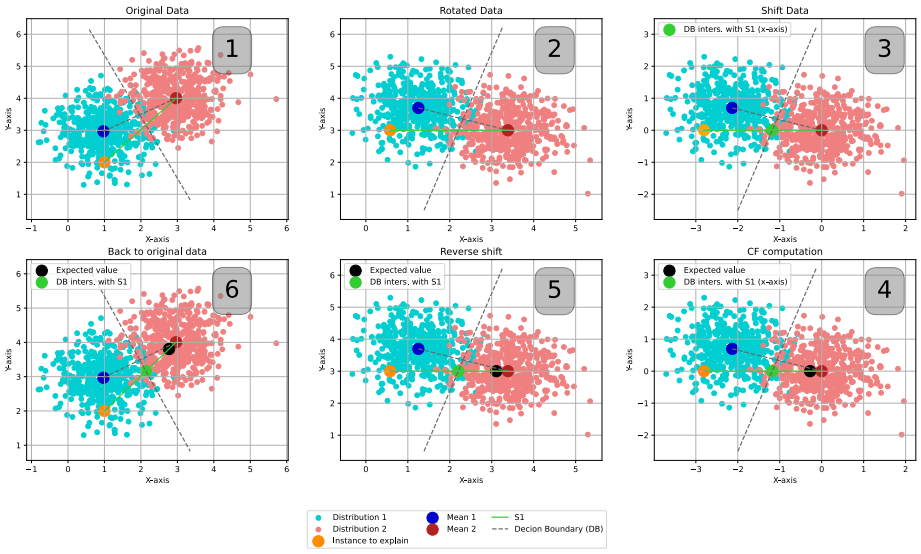


Fig. 5: Rotating the space to compute expectations 2-d visualisation step-by-step. (u-l) The original input space; (u-c) Data is rotated such that S_1 is parallel to the x-axis; (u-r) Data is shifted to zero the adversarial mean and the intercept between the decision boundary and S_1 is computed; (b-r) Expected value along the segment is estimated via one-dimensional sampling; (b-c) data shifted back; (b-l) Data is mapped back to its original values inverting the rotation matrix.

Unfortunately, segments S_1 and S_2 are never simultaneously parallel to the last axis, except for extremely rare cases. In the rest of the section, we will show how to manipulate the space to satisfy such requirement.

Given that rotating an isotropic Gaussian leaves densities of the points intact as distances are not affected by rotations, we can define a rotation matrix R to map a generic segment S into a segment which is parallel to the last axis.

Algorithm 2 Rotation Algorithm

ROTATE$(\cdot; m, v)$

Require: m, v, vector to map to rotated space z

$z^r \leftarrow z$
for $i = 0$ **to** $k - 1$ **do**
  $\theta \leftarrow \text{atan2}(v_i, v_{i+1})$
  $R \leftarrow I$
  $R_{i,i} \leftarrow \cos\theta$
  $R_{i,i+1} \leftarrow -\sin\theta$
  $R_{i+1,i} \leftarrow \sin\theta$
  $R_{i+1,i+1} \leftarrow \cos\theta$
  $z^r \leftarrow (z^r - m) \cdot R + m$
end for
return z^r

More precisely, given a and b reference points in the space connected by a segment S, we define a sequence of invertible rotations with respect to $m = \frac{a+b}{2}$, given the segment direction vector $v = b - a$. Each rotation will vanish the angle between the current component and the base vector of the next one, so to achieve our goal in $k-1$ steps. The procedure is shown in Algorithm 2. To invert the rotations and map back to the original space, we simply store the rotation matrices and gradually update z as: $z \leftarrow R_i^T(z - m) + m$, where R_i^T is the transpose of the rotations matrices presented in inverse order of computation. We name this inverse procedure ROTATE^{-1}.

Wrapping up, this procedure allows us to rotate the original label-relevant latent space, compute expectations with sampling on the rotated space, and map the expected value back in the original space without loss of information.

5.4 The Counterfactual Generation Algorithm

In the following section we assemble the various components presented so far. The full counterfactual generation process is presented in Algorithm 3. Given an instance x predicted as having label y^* and a user-provided counterfactual label $y_i \neq y^*$, the explanatory pipeline consists of: 1) encoding the instance to explain x in z_s and z_u; 2) rotating the S_1 and S_2 segments to align them on the last axis and sampling their expectations; 3) computing the expected counterfactual z_{cf} in latent space by averaging the expectations from the segments; 4) concatenating

the label-relevant and label-irrelevant latent representations and decoding the resulting latent representation into the final counterfactual explanation x_{cf}.

Algorithm 3 Explanation Algorithm

Require: x, y^*, y_i, instance to explain, predicted class and counterfactual class
Encode instances and extract label relevant and label irrelevant encodings
$\mu_\psi(x), \Sigma_\psi(x) \leftarrow ENC_s(x)$
$\mu_\phi(x), \Sigma_\phi(x) \leftarrow ENC_u(x)$
Sample $z_s \sim \mathcal{N}(\mu_\psi(x), \Sigma_\psi(x))$
Sample $z_u \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$
Rotate space to compute expectations along S_1 and S_2 sets of candidate counterfactuals with one dimensional sampling
$m, v \leftarrow 0.5(z_s + \mu_{y_i}), (\mu_{y_i} - z_s)$
$S_1 \leftarrow \{(1-t)\textsc{Rotate}(\mu_{y_i}; m, v) + t\textsc{Rotate}(z_s; m, v)\} \mid t \in [0, 1]$
$z_{cf_1} \leftarrow \textsc{Rotate}^{-1}(\mathbb{E}_{S_1}[z]; m)$
$M, v \leftarrow 0.5(\mu_{y^*} + \mu_{y_i}), (\mu_{y_i} - \mu_{y^*})$
$S_2 \leftarrow \{(1-t)\textsc{Rotate}(z_s; m, v) + t\textsc{Rotate}(\text{proj}_P(z_s); m, v)\} \mid t \in [0, 1]$
$z_{cf_2} \leftarrow \textsc{Rotate}^{-1}(\mathbb{E}_{S_2}[z]; m)$
Compute expected counterfactual as density based weighted sum
$w_1 \leftarrow \mathcal{N}(cf_1; \mu_{y_i}, I) \frac{1}{\mathcal{N}(cf_1; \mu_{y_i}, I) + \mathcal{N}(cf_2; \mu_{y_i}, I)}$
$z_{cf} \leftarrow w_1 z_{cf_1} + (1 - w_1) z_{cf_2}$
Concatenate label-irrelevant encoding of original instance with newly found label-relevant encoding and decode to generate the explanation
$x_{cf} \leftarrow DEC(\textsc{Concat}(z_u, z_{cf}))$
return x_{cf}

The benefit of this procedure is that explanations have a natural and interpretable connection with the instance to explain, as the label irrelevant generative factors are shared. Moreover, the estimate of the expected value via sampling ensures in-distribution outputs as they are directly generated from the adversarial distribution.

6 Preliminary Results

In the following section we illustrate results obtained with the implementation of our proposal. We demonstrate the effectiveness of our model in both satisfying assumptions of our counterfactual generative technique and learning the classification task. We also test our explanatory pipeline to generate counterfactual instances and show our explanations for qualitative evaluation.

6.1 Data

We focus on a very popular datasets in the machine learning community to assess the properties of our framework. FashionMNIST is an image datasets

containing 70k samples of shape 28x28x1. Images depicts clothing articles and we believe such domain is particularly suited to compare generated explanations with original instances in terms of shared and perturbed aspects due to its simple and intuitive nature.

6.2 Quantitative Evaluation

Model performance We train a Gaussian classifying deep generative model on FashionMNIST following the procedure presented in section 2. We obtain 90.51% accuracy on the test-set using the deterministic encodings and 86.33% accuracy using a single sample for the stochastic encodings. As anticipated repeating sampling leads to almost identical performances as using 10 samples with label assigned to majority votes the accuracy obtained is 90.03%. In figure 6 latent space configurations for the label-relevant and label-irrelevant branches are shown. T-SNE of the label-relevant encodings shows good label disentanglement, as classification performance anticipated, whereas through PCA on the label-irrelevant encoding we can see that such latents tend to conform to the prior distribution and are incapable of separating classes proving the effectiveness of our methodology.

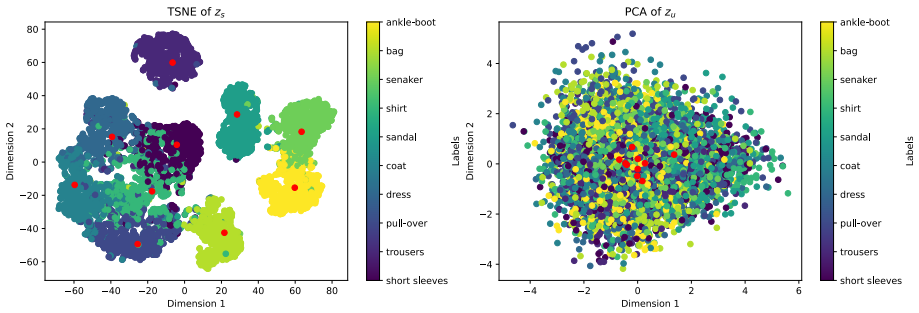


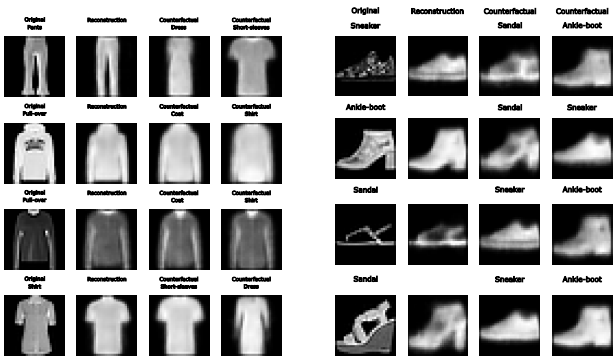
Fig. 6: Latent space for visualized via dimensionality reduction. For the label-relevant encodings (right) T-SNE was implemented to check class label disentanglement in the latent space. For the label-irrelevant encodings instead PCA was performed to check if latent codes conformed to the prior distribution.

Counterfactual generation One of the greatest strengths of our counterfactual generation algorithm is the efficiency of the counterfactual search process. Indeed as it simply consists on encoding instances, one-dimensional expected value computations via sampling and decoding, we can generate a counterfactual instance in just few seconds. More precisely average generation time is 2.156 seconds. In conclusion, another great advantage of our framework is removing any form of validity issue. We are indeed able to generate counterfactuals that are always predicted as the input class thanks to both our definition of candidate

counterfactuals coupled with direct access to the model decision boundary. The correctness of our implementation is validated by 100% validity obtained on the generated samples.

6.3 Qualitative Evaluation

In the following section we show generated counterfactuals for similar classes to the ones of randomly sampled instances to explain. We split our analysis between tow categories: clothing images and foot-wear images. Indeed images are drastically different going from one category to the other and generating contrastive explanations for elements of different categories is somewhat pointless as the recognition task becomes trivial. Finally we analyze few model mistakes to further test the effectiveness of our approach on ambiguous instances. Images



(a) (b)

Fig. 7: Counterfactuals for footwear (a) and clothing (b). Each row presents the original image next to the model reconstruction. Two counterfactuals for the most probable classes are also associated to them.

of clothing and the corresponding counterfactuals can be visualised in Figure 7a which provides for each row the instance to explain, the model reconstruction and two counterfactuals for the second and third most probable classes according to the model. Interestingly, even though counterfactual generation does not not directly optimize distance in the input space, a connection between the original instance and the explanations is evident. In our framework, sharing the label-irrelevant encoding with the original instance is responsible for such effect and the presented images are empirical evidence of the effectiveness of such approach. Indeed this is particularly evident in the first row where going from pant to dress leads to a very narrow dress generation or in the third row where a dark pull-over is associated with dark coats and shirts. On a final note it is worth observing that changes are highly interpretable as in row 4 going from a shirt instance

to a short-sleeves one consists in a simple change in the collar or in the second row where going from a pull-over to a shirt is obtained exclusively removing the hoodie.

With regard to footwear, we visualize explanations in Figure 7b to which many of the observations of above can be extended. It is particularly evident in the first two rows that high shoes are associated to counterfactuals with that property and darker images lead to darker counterfactuals. Changes here are also rather interpretable although they are somewhat simple and consist in addition or removal of texture and modifying the heel.

In conclusion, we present few instances consisting of model mistakes. We ask the model why the true label was discarded and also for a counterfactual of a very likely class and results are depicted in figure 8. Interestingly, a large portion of the informative feedback of the explanatory process seems to arrive from the model reconstruction. Indeed it can be seen that model reconstructions for such

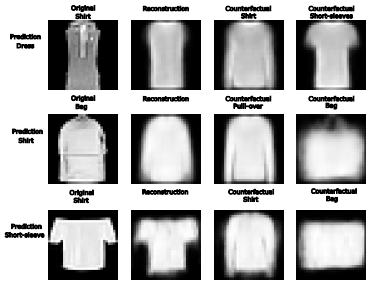


Fig. 8

instances justify the model prediction and this is particularly evident for the second row of the figure where the bag representation of the model has sleeves. When asked the changes to correctly predict the label the answer is also very informative as a circular handle is added to the top of the image. In addition, both in first and third row where the model wrongly does not assign the shirt label it appears evident that it strongly relies on the presence of long sleeves for its prediction. On a final note, it is worth noticing that exploiting the model reconstruction for explanatory purposes is extremely convenient to understand the model inner representation of the input which is somewhat of a depiction of how the model 'perceives' the instance it is fed.

7 Conclusion

In this work we presented an efficient explanatory pipeline based on generative models and counterfactual explanations. We showed how to train a deep model that jointly addresses generation and classification relying entirely on the data

distribution it models, and we presented a technique to provide counterfactuals based on the the assumption that data follows a mixture of Gaussians. We then validated the effectiveness of the method running preliminary experiments on FashionMNIST dataset to gain insight on the benefits and potential pitfalls of our approach. Our results confirm the advantage of unifying the predictive with the explanatory mechanism as they both rely on the same data-distribution assumptions. Given the strengths of our approach, efficient generation and interpretable outputs, we believe it could be especially suited for an interactive classification setting and we plan to perform future experiments in such hybrid setting.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
2. Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. Advances in neural information processing systems **31** (2018)
3. Dhuliawala, S., Sachan, M., Allen, C.: Variational classification. arXiv preprint arXiv:2305.10406 (2023)
4. Dhurandhar, A., Chen, P.Y., Luss, R., Tu, C.C., Ting, P., Shanmugam, K., Das, P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. Advances in neural information processing systems **31** (2018)
5. Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P.Y., Shanmugam, K., Puri, R.: Model agnostic contrastive explanations for structured data. arXiv preprint arXiv:1906.00117 (2019)
6. Ding, Z., Xu, Y., Xu, W., Parmar, G., Yang, Y., Welling, M., Tu, Z.: Guided variational autoencoder for disentanglement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7920–7929 (2020)
7. Farid, K., Schrodi, S., Argus, M., Brox, T.: Latent diffusion counterfactual explanations. arXiv preprint arXiv:2310.06668 (2023)
8. Feghahati, A., Shelton, C.R., Pazzani, M.J., Tang, K.: Cdeepex: Contrastive deep explanations. In: ECAI 2020, pp. 1143–1151. IOS Press (2020)
9. Fernández-Loría, C., Provost, F., Han, X.: Explaining data-driven decisions made by ai systems: The counterfactual approach (2021)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
11. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022)
12. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems **34**(6), 14–23 (2019)
13. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: Xai—explainable artificial intelligence. Science robotics **4**(37) (2019)
14. Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. ICLR (Poster) **3** (2017)

15. Jacovi, A., Swayamdipta, S., Ravfogel, S., Elazar, Y., Choi, Y., Goldberg, Y.: Contrastive explanations for model interpretability. arXiv preprint arXiv:2103.01378 (2021)
16. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint arXiv:1907.09615 (2019)
17. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In: IJCAI. pp. 2855–2862 (2020)
18. Kim, H., Mnih, A.: Disentangling by factorising. In: International conference on machine learning. pp. 2649–2658. PMLR (2018)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848 (2017)
21. Liu, S., Kailkhura, B., Loveland, D., Han, Y.: Generative counterfactual introspection for explainable deep learning. In: 2019 IEEE global conference on signal and information processing (GlobalSIP). pp. 1–5. IEEE (2019)
22. Miller, T.: Contrastive explanation: A structural-model approach. The Knowledge Engineering Review **36**, e14 (2021)
23. Molnar, C.: Interpretable Machine Learning. 2 edn. (2022), https://christophm.github.io/interpretable-ml-book
24. O'Shaughnessy, M., Canal, G., Connor, M., Rozell, C., Davenport, M.: Generative causal explanations of black-box classifiers. Advances in neural information processing systems **33**, 5453–5467 (2020)
25. Poels, Y., Menkovski, V.: Vae-ce: Visual contrastive explanation using disentangled vaes. In: International Symposium on Intelligent Data Analysis. pp. 237–250. Springer (2022)
26. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 344–350 (2020)
27. Prabhushankar, M., Kwon, G., Temel, D., AlRegib, G.: Contrastive explanations in neural networks. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3289–3293. IEEE (2020)
28. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International conference on machine learning. pp. 1278–1286. PMLR (2014)
29. Samangouei, P., Saeedi, A., Nakagawa, L., Silberman, N.: Explaingan: Model explanation via decision boundary crossing transformations. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 666–681 (2018)
30. Schneider, J.: Explainable generative ai (genxai): A survey, conceptualization, and research agenda. arXiv preprint arXiv:2404.09554 (2024)
31. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
32. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017)
33. Wan, W., Zhong, Y., Li, T., Chen, J.: Rethinking feature distribution for loss functions in image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9117–9126 (2018)

34. Wang, Y., Wang, X.: "why not other classes?": Towards class-contrastive back-propagation explanations. Advances in Neural Information Processing Systems **35**, 9085–9097 (2022)
35. Zheng, Z., Sun, L.: Disentangling latent space for vae by label relevant/irrelevant dimensions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12192–12201 (2019)
36. Zhu, X., Xu, C., Tao, D.: Learning disentangled representations with latent variation predictability. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16. pp. 684–700. Springer (2020)

Appendix

A: ELBO derivation

We start by defining the following joint distribution:

$$p(x, z_s, z_u) = \sum_c p_\theta(x|z_s, z_u)p(z_s, y)p(z_u)$$

According to Jensens inequality we have:

$$\begin{aligned}\log\, p(x) &= \log\, \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\frac{p_\theta(x|z_s, z_u)p(z_s|y)p(y)p(z_u)}{q_\psi(z_s, y|x)q_\phi(z_u|x)}\right] \\ &\geq \mathbb{E}_{q_\psi(z_s,y|x),q_\phi(z_u|x)}\log\left[\frac{p_\theta(x|z_s, z_u)p(z_s, y)p(z_u)}{q_\psi(z_s, y|x)q_\phi(z_u|x)}\right]\end{aligned}$$

and:

$$\begin{aligned}&\mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\log\, \frac{p_\theta(x|z_s, z_u)p(z_s|y)p(y)p(z_u)}{q_\psi(z_s|x)p(y|x)q_\phi(z_u|x)}\right] \\ &= \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\log\, p_\theta(x|z_s, z_u)\right] \\ &+ \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\log\, \frac{p(z_u)}{q_\phi(z_u|x)}\right] \\ &+ \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\log\, \frac{p(z_s|y)}{q_\psi(z_s|x)p(y|x)}\right] \\ &+ \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\left[\log\, p(y)\right]\end{aligned}$$

The last term is a prior probability and is constant during training so it can be ignored. The final result is:

$$\begin{aligned}\text{ELBO } &= \mathbb{E}_{q_\psi(z_s|x)p(y|x),q_\phi(z_u|x)}\Big[\log\, p_\theta(x|z_s,z_u)\Big] \\ &- D_{\text{kl}}\left(q_\phi(z_u|x) \parallel p(z_u)\right) \\ &- D_{\text{kl}}\left(q_\psi(z_s|x)p(y|x) \parallel p(z_s|y)\right)\end{aligned}$$

which is the formulation presented in 3.

B: proof of proposition 2 on regularized deterministic latent classification

We prove that a Gaussian classification loss computed on the deterministic output of the encoder $\mu_\psi(x)$ to the kl-divergence regularization term is equivalent to implementing the Gaussian mixture loss framework coupled with an additional variance regularization term:

$$\mathcal{L}_{kl_s} + \mathcal{L}^s_{cls} = \mathcal{L}_{GM} + \mathcal{L}_{var}$$

Proof: Given a batch of size N set:

$$\begin{aligned}&\Sigma_{z_s|y} = I \\ &\mathcal{L}_{cls} = -\frac{1}{N}\sum_c \mathbb{I}(y_i = c)\sum_i\ \log\ \frac{\mathcal{N}(\mu_\psi(x_i);\mu_{z_s|y_i},I)p(y_i)}{\sum_c \mathcal{N}(\mu_\psi(x_i);\mu_{z_s|c},I)p(c)} \\ &\mathcal{L}_{lkd_s} = -\sum_c\sum_i \mathbb{I}(y_i = c)\ \log \mathcal{N}(\mu_\psi(x_i);\mu_{z_s|y_i},I) \\ &\mathcal{L}_{GM} = \mathcal{L}_{lkd_s} + \mathcal{L}_{cls}\end{aligned}$$

.
The last term of the 3 is instead given by:

$$\begin{aligned}&- D_{\text{kl}}\left(q_\psi(z_s|x)p(y|x) \parallel p(z_s|y)\right) \\ &= -D_{\text{kl}}\left(\mathcal{N}(\mu_\psi(x),\Sigma_\psi(x))p(y|x) \parallel \mathcal{N}(\mu_{z_s|y},I)\right) \\ &= -\sum_c\sum_{i=1}^N \int q_\psi(z_s|x_i)\mathbb{I}(y_i = c)\ \log\ \frac{p(z_s|y_i)}{q_\psi(z_s|x_i)\mathbb{I}(y_i = c)}dz_s\end{aligned}$$

When the identity function is satisfied, this has closed form solution for two isotropic Gaussians:

$$\mathcal{L}_{kl_s} = -\sum_c \mathbb{I}(y_i = c)\sum_{i=1}^N -\log\frac{\Sigma_\psi(x_i)}{2} + \frac{\Sigma_\psi(x_i)}{2} + \frac{(\mu_\psi(x_i) - \mu_{z_s|y_i})^2}{2} - \frac{1}{2}$$

It is rather straightforward to see that this contains $\mathcal{L}_{lkd_s}$ when the identity function of its formulation is satisfied as:

$$\mathcal{L}_{lkd_s} = -\sum_{c} \mathbb{I}(y_i = c) \sum_{i=1}^{N} \frac{(\mu_\psi(x_i) - \mu_{z_s|y_i})^2}{2} + const$$

and the remaining terms of the closed form KL divergence are interpreted as additional variance regularization terms: $\mathcal{L}_{var}$.
Finally we can write as in 4:

$$\mathcal{L}_{kl_s} = \mathcal{L}_{lkd_s} + \mathcal{L}_{var}$$
$$\mathcal{L}_{kl_s} + \mathcal{L}_{cls} = \mathcal{L}_{lkd_s} + \mathcal{L}_{cls} + \mathcal{L}_{var} = \mathcal{L}_{GM} + \mathcal{L}_{var}$$

C: proof of proposition 3 on noise invariant label assignment

We prove that the expected label assigned to an instance using a sampled stochastic encoding is the same as the one assigned using its corresponding deterministic representation.
Proof: Given an instance $x_i \in \mathcal{X}$, $\mu_\psi(x_i)$ and $z_s \sim N(\mu_\psi(x_i), \Sigma_\psi(x_i))$ its deterministic and stochastic label-relevant encodings, $f_M(\cdot)$ is the latent Gaussian classification model such that $f_M(z_s) = y^*_{st}$ and $f_M(\mu_\psi(x_i)) = y^*_d$ are the two labels predicted respectively with the sampled and deterministic latents and the label assigned is obtained as follows:

- Class-wise distribution parameters are obtained computing statistics on the latent encodings;
- Elements are assigned to the maximum posterior probability class according to the Gaussian density functions.

With regard to the first element, deterministic encodings are simply encoded and mean and variance parameters are trivially computed. For the prediction process instead, we note that each stochastic element is predicted as the corresponding deterministic element if their respective closest class-distribution centroid is the same, due to diagonal covariance matrices assumption. Therefore:

$$y^*_{st} = \underset{y}{\operatorname{argmin}} \parallel z_s - \hat{\mu}_y \parallel_2^2 \ ; y^*_d = \underset{y}{\operatorname{argmin}} \parallel \mu_\psi(x_i) - \hat{\mu}_y \parallel_2^2$$

Follows that $y^*_{st} = y^*_d$ if and only if:

$$\parallel z_s - \hat{\mu}_{y^*_d} \parallel_2^2 - \parallel z_s - \hat{\mu}_{y_i} \parallel_2^2 < 0 \ \forall \ y_i \in \mathcal{Y} = \{y_1, ..., y_c\}/\{y^*_d\}$$

By taking expectation we have:

$$\mathbb{E}_{z_s}\left[\parallel z_s - \hat{\mu}_{y_d}^* \parallel_2^2 - \parallel z_s - \hat{\mu}_{y_i} \parallel_2^2 \right] = \mathbb{E}_{z_s}\left[z_s^2 + {\mu_{y_d}^*}^2 - 2z_s\mu_{y_d}^* - z_s^2 - {\mu_{y_i}}^2 + 2z_s\mu_{y_i} \right]$$
$$= \mathbb{E}_{z_s}[z_s^2] + {\mu_{y_d}^*}^2 - 2z_s\mu_{y_d}^* - \mathbb{E}_{z_s}[z_s^2] - {\mu_{y_i}}^2 + 2z_s\mu_{y_i}]$$
$$= \mathbb{E}_{z_s}[z_s^2] + {\mu_{y_{det}}^*}^2 - 2\mathbb{E}_{z_s}[z_s]\mu_{y_d}^* - \mathbb{E}_{z_s}[z_s^2] - {\mu_{y_i}}^2 + 2\mathbb{E}_{z_s}[z_s]\mu_{y_i}$$
$$= \mathrm{Var}[z_s] + \mathbb{E}_{z_s}[z_s]^2 + {\mu_{y_d}^*}^2 - 2\mathbb{E}_{z_s}[z_s]\mu_{y_d}^* - \mathrm{Var}[z_s] - \mathbb{E}_{z_s}[z_s]^2 - {\mu_{y_i}}^2 + 2\mathbb{E}_{z_s}[z_s]\mu_{y_i}$$
$$= \mu_\psi(x_i)^2 + {\mu_{y_d}^*}^2 - 2\mu_\psi(x_i)\mu_{y_d}^* - \mu_\psi(x_i)^2 - {\mu_{y_i}}^2 + 2\mu_\psi(x_i)\mu_{y_i}$$
$$= \parallel \mu_\psi(x_i) - \hat{\mu}_{y_d}^* \parallel_2^2 - \parallel \mu_\psi(x_i) - \hat{\mu}_{y_i} \parallel_2^2$$

But by assumption:

$$\parallel \mu_\psi(x_i) - \hat{\mu}_{y_d}^* \parallel_2^2 - \parallel \mu_\psi(x_i) - \hat{\mu}_{y_i} \parallel_2^2 < 0 \; \forall \; y_i \in \mathcal{Y} = \{y_1, ..., y_c\}/\{y_d^*\}$$

Concluding our proof.

D: proof of proposition 4 on definition of the candidates

In this section we provide proof for the statement of proposition 5.
Given set of counterfactual candidates $\mathcal{C}$, centroid μ_{y^*} instance $z_s' : f_M(z_s') = y^*$, and segment from z_s' to μ_{y_i} (other class centroid) $S_1 = \{(1-t)z_s' + (t)\mu_{y_i} \mid t \in [0,1]\}$, we have that the decision boundary for class y^* and y_i is the plane P defined as:

$$P : (\mu_{\mathbf{y}^*} - \mu_{\mathbf{y_i}}) \cdot (\mathbf{z} - \frac{\mu_{\mathbf{y}^*} + \mu_{\mathbf{y_i}}}{\mathbf{2}}) = 0.$$

And the the intersection I between S_1 and P is:

$$I = (1-t^*)z_s' + (t^*)\mu_{y_i} \text{ with } t^* = \frac{(\mu_{\mathbf{y}^*} - \mu_{\mathbf{y_i}}) \cdot (\frac{\mu_{y^*} + \mu_{y_i}}{2} - z_s')}{(\mu_{\mathbf{y}^*} - \mu_{\mathbf{y_i}}) \cdot (\mu_{y_i} - z_s')}$$

Finally we define the orthogonal projection of z_s' on P as $\mathrm{proj}_P(z_s')$.
We want to prove that points on the segment S_2 connecting the intersection between S_1 and the decision boundary with the closest point to z_s' predicted as y_i belong to the set of candidates:

$$S_2 = \{(1-t)I + (t)\mathrm{proj}_P(z_s') \mid t \in [0,1]\} \subseteq \mathcal{C}$$

Given the counterfactual candidates definition provided with 7, in order to support the statement of proposition 4 we need to show that for all points $z \in \mathcal{R}^k$ in the latent space such that $f_M(z) = y_i$ at any given distance d from z_s' such that $\parallel \mathrm{proj}_P(z_s') - z_s' \parallel_2^2 < d < \parallel I - z_s' \parallel_2^2$ the closest point z^* to μ_{y_i} lies on S_2.
This is because for all elements at a distance $d > \parallel I - z_s' \parallel_2^2$ to z_s' the trivial optimum z^* lies on S_1 and also:

$$\nexists \; z \in \mathcal{R}^k : \parallel z - z_s' \parallel_2^2 < \parallel \mathrm{proj}_P(z_s') - z_s' \parallel_2^2 \wedge f_M(z) = y_i$$

We provide readers with a proof sketch with the help of the following image:

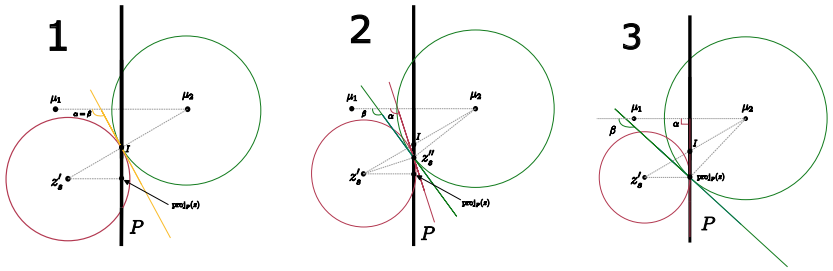


Fig. 9

Starting from the left-most picture, we see a 2-d depiction of the mean vectors μ_{y^*} and μ_{y_2}, the instance to explain z'_s, the decision boundary P, the intersection I and the orthogonal projection of z' on P. One can also notice two circles tangent on I. The green circle represents the collection of points at distance $d = \| I - \mu_{y_2} \|_2^2$ to μ_{y_2} and the red circle the points a distance $d = \| I - z'_s \|_2^2$ to z'_s. Finally, the line tangent to the red circle in I forms an angle with the vector $\mu_{y^*} - \mu_{y_2}$ named α and the line tangent to the green circles in I forms an angle named β. Since the two circles are tangent in I these two lines correspond (yellow in the image) and the angles α and β are the same.

To prove tour proposition it suffices to show that for a point $z''_S \in S_2$ the two circles centered in μ_{y_2} and z'_s and intersecting in z''_s do not intersect again for any point such that $f_M(z''_s) = y_2$. Intuitively, if this holds then any point z'''_s such that $\| z'''_s - z'_s \|_2^2 = \| z''_s - z'_s \|_2^2$ we have: $\| z'''_s - \mu'_{y_2} \|_2^2 > \| z''_s - \mu_{y_2} \|_2^2$.

We prove this with the help of picture 2 and 3. First we notice that all the points in a circle are bounded by a line tangent to the circle in a given point. We exploit such tangents for our proof. If for any point $z''_s \in S_2$, given the circles centered in μ_{y_2} and z'_s and intersecting in z''_s, the lines tangent to the two circles, respectively l_μ and l_z, intersect in z''_s as can be seen in picture 2. To conclude our proof we need to show that all the points in l_z such that they are predicted μ_{y_2} are closer to z'_s than all the points of l_μ such that they are predicted μ_{y_2}.

Such relation can be inferred by taking into consideration the angles α and β between the tangents and the vector $\mu_{y^*} - \mu_{y_2}$. Considering all 3 pictures together we can see indeed that the angle β is always greater than α for all points in S_2. The implication of this is that before the intersection between the two tangents z''_s the points on l_μ will be closer to z'_s but after the intersection,

after which the points in l_z and l_μ are predicted μ_{y_2}, the points in l_z will be closer to z'_s.

E: proof of proposition 5 on expectation along a segment parallel to an axis

We show that the expected value, according to an isotropic Gaussian, of the elements in a segment S parallel to the last axis can be computed with single-dimensional sampling as depicted by equation 11.

$$\mathbb{E}_S[z] = \left(c, c, ..., c, \frac{\int_{a_k}^{b_k} z_k f_{Z_k}(z_k)dt}{\int_{a_k}^{b_k} f_{Z_k}(z_k)dt}\right)$$

proof: Take two points aligned along the last axis $A = (c, c, ..., c, a_k)$ and $B = (c, c, ..., c, b_k) \in \mathbb{R}^k$, with $c, a_k, b_k \in \mathbb{R}$ and $a_k < b_k$ and the segment S connecting them $S = \{(1-t)A+(t)(B) \mid t \in [0, 1]\}$. Indeed any point $z \in S$ can be expressed as a function of t: $Z(t) = (1-t)A + (t)(B)$. More precisely any component of any point $z \in S$ can be expressed as a function of the corresponding components of A and B and t: $Z_i(t) = (1-t)a_i + t(b_i)$. Clearly such formulation leaves all components intact ($a_i = c = b_i \ \forall \ i \neq k$) expect the last one which is: $Z_k(t) = (1-t)a_k + t(b_k)$.
In addition, if the underlying distribution of the points in S is an isotropic Gaussian we can factorize the density as follows:

$$f_{Z_1,...,Z_k}(z_1, ..., z_k) = \prod_i^k f_{Z_i}(z_i)$$

And the expected value becomes:

$$\mathbb{E}_S[z] = \frac{\int_0^1 Z(t) f_Z(Z(t))dt}{\int_0^1 f_Z(Z(t))dt} = \frac{\int_0^1 Z(t) \prod_{i=1}^k f_{Z_i}(Z_i(t))dt}{\int_0^1 \prod_{i=1}^k f_{Z_i}(Z_i(t))dt}$$

But:

$$\prod_{i=1}^k f_{Z_i}(Z_i(t)) = f_{Z_k}(Z_k(t)) \prod_{i=1}^{k-1} f_{Z_i}(c)$$

and:

$$\frac{\int_0^1 Z(t) \prod_{i=1}^k f_{Z_i}(Z_i(t))dt}{\int_0^1 \prod_{i=1}^k f_{Z_i}(Z_i(t))dt} = \frac{\prod_{i=1}^{k-1} f_{Z_i}(c) \int_0^1 Z(t) f_{Z_k}(Z_k(t))dt}{\prod_{i=1}^{k-1} f_{Z_i}(c) \int_0^1 f_{Z_k}(Z_k(t))dt} = \frac{\int_0^1 Z(t) f_{Z_k}(Z_k(t))dt}{\int_0^1 f_{Z_k}(Z_k(t))dt}$$

To conclude our proof we have that for a given t value $Z(t)$ is a vector of the form $(c, c, ..., c, Z_k(t))$ and we can write:

$$\begin{aligned}
\mathbb{E}_S[z] &= \left(\frac{\int_0^1 c f_{Z_k}(Z_k(t))}{\int_0^1 f_{Z_k}(Z_k(t))dt}, ..., \frac{\int_0^1 c f_{Z_k}(Z_k(t))}{\int_0^1 f_{Z_k}(Z_k(t))dt}, \frac{\int_0^1 Z_k(t) f_{Z_k}(Z_k(t))dt}{\int_0^1 f_{Z_k}(Z_k(t))dt} \right) \\
&= \left(\frac{c\int_0^1 f_{Z_k}(Z_k(t))}{\int_0^1 f_{Z_k}(Z_k(t))dt}, ..., \frac{c\int_0^1 f_{Z_k}(Z_k(t))}{\int_0^1 f_{Z_k}(Z_k(t))dt}, \frac{\int_0^1 Z_k(t) f_{Z_k}(Z_k(t))dt}{\int_0^1 f_{Z_k}(Z_k(t))dt} \right) \\
&= \left(c, ..., c, \frac{\int_0^1 Z_k(t) f_{Z_k}(Z_k(t))dt}{\int_0^1 f_{Z_k}(Z_k(t))dt} \right)
\end{aligned}$$

with:

$$\frac{\int_0^1 Z_k(t) f_{Z_k}(Z_k(t))dt}{\int_0^1 f_{Z_k}(Z_k(t))dt} = \frac{\int_{a_k}^{b_k} z_k f_{Z_k}(z_k)dz_k}{\int_{a_k}^{b_k} f_{Z_k}(z_k)dz_k}$$

Proving that to estimate the last component, which is the only one whose value is modified, we can resort to one-dimensional sampling.