Supplementary Material
Beyond Trivial Counterfactual Explanations with Diverse Valuable Explanations

Pau Rodríguez[1] Massimo Caccia[1,2,3] Alexandre Lacoste[1] Lee Zamparo[1] Issam Laradji[1,2,4]
Laurent Charlin[2,5,6] David Vazquez[1]
[1]Element AI [2]MILA [3]Université de Montréal [4]McGill University [5]HEC Montreal
[6]Canada CIFAR AI Chair
pau.rodriguez@servicenow.com

Section 1 contains the extended related work, Section 2 shows additional qualitative results, Section 3 contains additional results for identity preservation, Section 4 contains the implementation details, Section 5 contains additional information about the experimental setup, Section 6 provides the results of human evaluation of DiVE, Section 7 contains details about the model architecture, Section 8 contains the DiVE Algorithm, and Section 9 contains details about the OOD experiment.

1. Extended Related Work

Counterfactual explanation lies inside a more broadly-connected body of work for explaining classifier decisions. Different lines of work share this goal but vary in the assumptions they make about what elements of the model and data to emphasize as way of explanation.

Model-agnostic counterfactual explanation like [23, 30], these models make no assumptions about model structure, and interact solely with its label predictions. Karimi et al. [19] develop a model agnostic, as well as metric agnostic approach. They reduce the search for counterfactual explanations (along with user-provided constraints) into a series of satisfiability problems to be solved with off-the-shelf SAT solvers. Similar in spirit to [30], Guidotti et al. [14] first construct a local neighbourhood around test instances, finding both positive and negative exemplars within the neighbourhood. These are used to learn a shallow decision tree, and explanations are provided in terms of the inspection of its nodes and structure. Subsequent work builds on this local neighbourhood idea [15], but specializes to medical diagnostic images. They use a VAE to generate both positive and negative samples, then use random heuristic search to arrive at a balanced set. The generated explanatory samples are used to produce a saliency feature map for the test data point by considering the median absolute deviation of pixel-wise differences between the test point, and the positive and negative example sets.

Gradient based feature attribution. These methods identify input features responsible for the greatest change in the loss function, as measured by the magnitude of the gradient with respect to the inputs. Early work in this area focused on how methodological improvements for object detection in images could be re-purposed for feature attribution [38, 39], followed by work summarized gradient information in different ways [31, 33, 35]. Closer inspection identified pitfalls of gradient-based methods, including induced bias due to gradient saturation or network structure [1], as well as discontinuity due to activation functions [32]. These methods typically produce dense feature maps, which are difficult to interpret. In our work we address this by constraining the generative process of our counterfactual explanations.

Reference based feature attribution. These methods focus instead on measuring the differences observed by substituting observed input values with ones drawn from some reference distribution, and accumulating the effects of these changes as they are back-propagated to the input features. Shrikumar et al. [32] use a modified back-propagation approach to gracefully handle zero gradients and negative contributions, but leave the reference to be specified by the user. Fong and Vedaldi [10] propose three different heuristics for reference values: replacement with a constant, addition of noise, and blurring. Other recent efforts have focused on more complex proposals of the reference distribution. Chen et al. [5] construct a probabilistic model that acts as a lower bound on the mutual information between inputs and the predicted class, and choose zero values for regions deemed uninformative. Building on desiderata proposed by Dabkowski and Gal [6], Chang et al. [4] use a generative model to marginalize over latent values of relevant regions, drawing plausible values for each. These methods typically either do not identify changes that would *alter* a classifier decision, or they do not consider the plausibility of those changes.

Counterfactual explanations. Rather than identify a set of features, counterfactual explanation methods instead generate perturbed versions of observed data that result in a corresponding change in model prediction. These methods usually assume both more access to model output and parameters, as well as constructing a generative model of the data to find trajectories of variation that elucidate model behaviour for a given test instance.

Joshi et al. [18] propose a gradient guided search in latent space (via a learned encoder model), where they progressively take gradient steps with respect to a regularized loss that combines a term for plausibility of the generated data, and the loss of the ML model. Denton et al. [8] use a Generative Adversarial Network (GAN) [12] for detecting bias present in multi-label datasets. They modify the generator to obtain latent codes for different data points and learn a linear decision boundary in the latent space for each class attribute. By sampling generated data points along the vector orthogonal to the decision boundary, they observe how crossing the boundary for one attribute causes undesired changes in others. Some counterfactual estimation methods forego a generative model by instead solving a surrogate editing problem. Given an original image (with some predicted class), and an image with a desired class prediction value, Goyal et al. [13] produce a counterfactual explanation through a series of edits to the original image by value substitutions in the learned representations of both images. Similar in spirit are Dhurandhar et al. [9] and Van Looveren and Klaise [37]. The former propose a search over features to highlight subsets of those present in each test data point that are typically present in the assigned class, as well as features usually absent in examples from adjacent classes (instances of which are easily confused with the label for the test point predicted by the model). The latter generate counterfactual data that is proximal to x_test, with a sparse set of changes, and close to the training distribution. Their innovation is to use class prototypes to serve as an additional regularization term in the optimization problem whose solution produces a counterfactual.

Several methods go beyond providing counterfactually generated data for explaining model decisions, by additionally qualifying the effect of proposed changed between a test data point and each counterfactual. Mothilal et al. [24] focus on tabular data, and generate sets of counterfactual explanations through iterative gradient based improvement, measuring the cost of each counterfactual by either distance in feature space, or the sparsity of the set of changes (while also allowing domain expertise to be applied). Poyiadzi et al. [28] construct a weighted graph between each pair of data point, and identify counterfactuals (within the training data) by finding the shortest paths from a test data point to data points with opposing classes. Pawelczyk et al. [26] focus on modelling the density of the data to provide 'attainable' counterfactuals, defined to be proximal to test data points, yet not lying in low-density sub-spaces of the data. They further propose to weigh each counterfactual by the changes in percentiles of the cumulative distribution function for each feature, relative to the value of a test data point.

2. Qualitative results

Figure 1,2 present counterfactual explanations for additional persons and attributes. The results show that DiVE achieves higher quality reconstructions compared to other methods. Further, the reconstructions made by DiVE are more correlated with the desired target for the ML model output $f(x)$. We compare DiVE to PE and xGEM+. We found that gender changes with the "Smiling" attribute with f_{biased} while for f_{unbiased} it stayed the same. In addition, we also observed that for f_{biased} the correlation between "Smile" and "Gender" is higher than for PE. It can also be observed that xGEM+ fails to retain the identity of the person in $\mathbf{x}$ when compared to PE and our method. Finally, Figure 3 shows *successful counterfactuals* for different instantiations of DiVE.

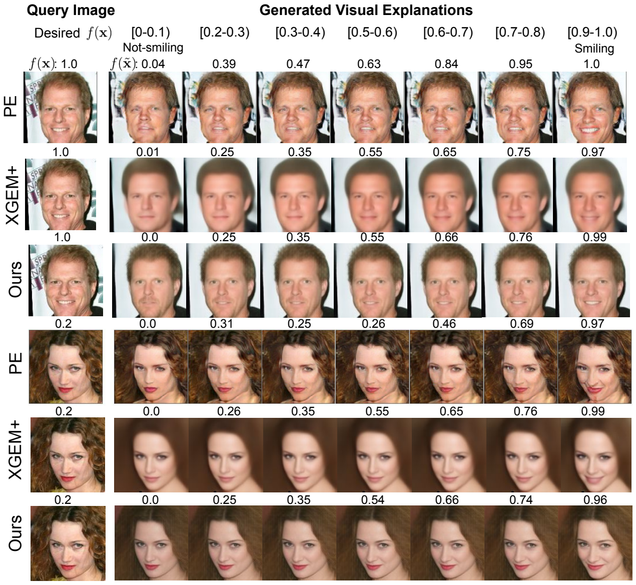


Figure 1: Qualitative results of DiVE, Progressive Exaggeration (PE) [34], and xGEM [18] for the "Smiling" attribute. Each column shows the explanations generated for a target probability output of the ML model. The numbers on top of each row show the actual output of the ML model.

Note that PE directly optimizes the generative model to take an input variable $\delta \in \mathbb{R}$ that defines the desired output

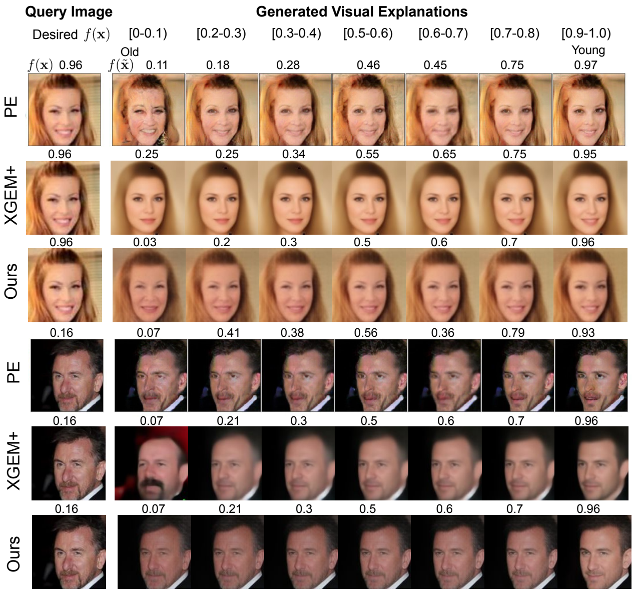


Figure 2: **Qualitative results** of DiVE, Progressive Exaggeration (PE) [34], and xGEM+ for the "Young" attribute. Each column shows the explanations generated for a target probability output of the ML model. The numbers on top of each row show the actual output of the ML model.

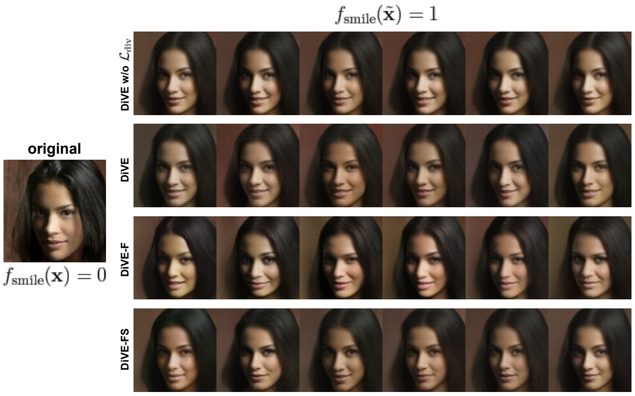

$f_{\text{smile}}(\tilde{\mathbf{x}}) = 1$
DiVE w/o $\mathcal{L}_{\text{div}}$
original
DiVE
DiVE-F
DiVE-FS
$f_{\text{smile}}(\mathbf{x}) = 0$

Figure 3: Successful counterfactual generations for different instantiations of DiVE. Here, the original image was misclassified as non-smiling. All methodologies were able to change the predicted class to "Smiling".

probability $\tilde{y} = f(\mathbf{x}) + \delta$. To obtain explanations at different probability targets, we train a second order spline on the trajectory of perturbations produced during the gradient descent steps of our method. Thus, given the set of perturbations $\{\varepsilon_t\}$, $\forall t \in 1..\tau$, obtained during τ gradient steps, and the corresponding black-box outputs $\{f(y|\varepsilon_t)\}$, the spline obtains the $\varepsilon_{\tilde{y}}$ for a target output $\tilde{y}$ by interpolation.

3. Identity preservation

As argued, valuable explanations should remain proximal to the original image. Accordingly, we performed the identity preservation experiment found in [34] to benchmark the methodologies against each other. Specifically, use the VGGFace2-based [3] oracle to extract latent codes for the original images as well as for the explanations and report **latent space closeness** as the fraction of time the explanations' latent codes are the closest to their respective original image latent codes' compared to the explanations on different original images. Further, we report **face verification** accuracy which consist of the fraction of time the cosine distance between the aforementioned latent codes is below 0.5.

Table 1 presents both metrics for DiVE and its baselines on the "Smilling" and "Young" classification tasks. We find that DiVE outperforms all other methods on the "Young" classification task and almost all on the "Smiling" task.

4. Implementation details

In this Section, we provide provide the details to ensure the that our method is reproducible.

Architecture details. DiVE's architecture is a variation BigGAN [2] as shown in Table 3. We chose this architecture because it achieved impressive FID results on the ImageNet [7]. The decoder (Table 3b) is a simplified version of the 128×128 BigGAN's residual generator, without non-local blocks nor feature concatenation. We use InstanceNorm [36] instead of BatchNorm [17] to obtain consistent outputs at inference time without the need of an additional mechanism such as recomputing statistics [2]. All the InstanceNorm operations of the decoder are conditioned on the input code $\mathbf{z}$ in the same way as FILM layers [27]. The encoder (Table 3a) follows the same structure as the BigGAN 128×128 discriminator with the same simplifications done to our generator. We use the Swish non-linearity [29] in all layers except for the output of the decoder, which uses a Tanh activation.

For all experiments we use a latent feature space of 128 dimensions. The ELBO has a natural principled way of selecting the dimensionality of the latent representation. If d is larger than necessary, it will not enhance the reconstruction error and the optimization of the ELBO will make the posterior equal to the prior for these extra dimensions. More can be found on the topic in [22]. In practice, we experimented with $d = \{64, 128, 256\}$ and found that with $d = 128$ we achieved a slightly lower ELBO.

To project the 2d features produced by the encoder to a flat vector (μ, $\log(\sigma^2)$), and to project the sampled codes $\mathbf{z}$ to a 2d space for the decoder, we use 3-layer MLPs. For

	CelebA:Smiling				CelebA:Young			
	xGEM	PE	xGEM+	DiVE (ours)	xGEM	PE	xGEM+	DiVE (ours)
Latent Space Closeness	88.2	88.0	**99.8**	98.7	89.5	81.6	97.5	**99.1**
Face Verification Accuracy	0.0	85.3	91.2	**97.3**	0.0	72.2	97.4	**98.2**

Table 1: Identity preserving performance on two prediction tasks.

the face attribute classifiers, we use the same DenseNet [16] architecture as described in Progressive Exaggeration [34].

Optimization details. All the models are optimized with Adam [20] with a batch size of 256. During the training step, the auto-encoders are optimized for 400 epochs with a learning rate of $4 \cdot 10^{-4}$. The classifiers are optimized for 100 epochs with a learning rate of 10^{-4}. To prevent the auto-encoders from suffering KL vanishing, we adopt the cyclical annealing schedule proposed by Fu et al. [11].

Counterfactual inference details. At inference time, the perturbations are optimized with Adam until the ML model output for the generated explanation $f(\tilde{\mathbf{x}})$ only differs from the target output $\tilde{y}$ by a margin δ or when a maximum number of iterations τ is reached. We set $\tau = 20$ for all the experiments since more than 90% of the counterfactuals are found after that many iterations. The different $\boldsymbol{\epsilon_i}$ are initialized by sampling from a normal distribution $\mathcal{N} \sim (0, 0.01)$. For the DiVE$_{Fisher}$ baseline, to identify the most valuable explanations, we sort $\boldsymbol{\epsilon}$ by the magnitude of the diagonal of the Fisher Information Matrix, i.e. $\mathbf{f} = \mathrm{diag}(\boldsymbol{F})$. Then, we divide the dimensions of the sorted $\boldsymbol{\epsilon}$ into N contiguous partitions of size $k = \frac{D}{N}$, where D is the dimensionality of $\mathcal{Z}$. Formally, let $\boldsymbol{\epsilon}^{(\mathbf{f})}$ be $\boldsymbol{\epsilon}$ sorted by $\mathbf{f}$, then $\boldsymbol{\epsilon}^{(\mathbf{f})}$ is constrained as follows,

$$\varepsilon^{(\mathbf{f})}_{i,j} = \begin{cases} 0, & \text{if } j \in [(i-1)\cdot k, i \cdot k] \\ \varepsilon^{(\mathbf{f})}_{i,j}, & \text{otherwise} \end{cases}, \qquad (1)$$

where $i \in 1..N$ indexes each of the multiple $\boldsymbol{\varepsilon}$, and $j \in 1..D$ indexes the dimensions of $\boldsymbol{\varepsilon}$. As a result we obtain partitions with different order of complexity. Masking the first partition results in explanations that are most implicit within the model and the data. On the other hand, masking the last partition results in explanations that are more explicit.

To compare with Singla et al. [34] in Figures 1-2 we produced counterfactuals at arbitrary target values $\tilde{y}$ of the output of the ML model classifier. One way to achieve this would be to optimize $\mathcal{L}_{\text{CF}}$ for each of the target probabilities. However, these successive optimizations would slow down the process of counterfactual generation. Instead, we propose to directly maximize the target class probability and then interpolate between the points obtained in the gradient descent trajectory to obtain the latent factors of the different target probabilities. Thus, given the set of perturbations $\{\boldsymbol{\varepsilon}_t\}$, $\forall t \in 1..\tau$, obtained during τ gradient steps, and the corresponding ML model outputs $\{f(y|\boldsymbol{\varepsilon}_t)\}$, we obtain the $\boldsymbol{\varepsilon}_{\tilde{y}}$ for a target output $\tilde{y}$ by interpolation. We do such interpolation by fitting a piecewise quadratic polynomial on the latent trajectory, commonly known as Spline in the computer graphics literature.

5. Beyond Trivial Explanations Experimental Setup

The experimental benchmark proposed in Section 4.1 is performed on a subset of the validation set of CelebA. This subset is composed of 4 images for each CelebA attribute. From these 4 images, 2 were correctly classified by the ML model, while the other 2 were misclassified. The two correctly classified images are chosen so that one was classified with a high confidence of 0.9 and the other one with low confidence of 0.1. The 2 misclassifications were chosen with the same criterion. The total size of the dataset is of 320 images. For each of these images we generate k counterfactual explanations. From these counterfactuals, we report the ratio of successful explanations.

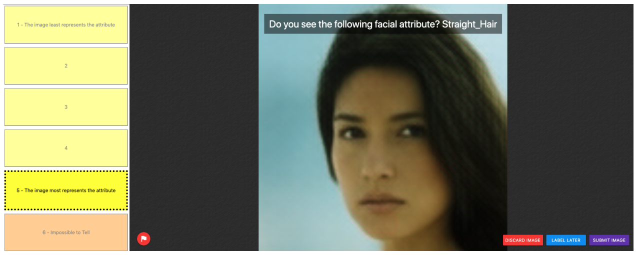
Here are the specific values we tried in our hyperparameter search: $\gamma \in [0.0, 0.001, 0.1, 1.0]$, $\alpha \in [0.0, 0.001, 0.1, 1.0]$, $\lambda \in [0.0001, 0.0005, 0.001]$, number of explanations 2 to 15 and learning rate $\in [0.05, 0.1]$. Because xGEM+ does not have a γ nor α parameter, we increased its learning rate span to $[0.01, 0.05, 0.1]$ to reduce the gap in its search space compared with DiVE. We also changed the random seeds and ran a total of 256 trials.

6. Human Evaluation

We built a web-based human evaluation task to assess if DiVE is more successful at finding *non-trivial* counterfactuals than previous state of the art and the effectiveness of the VGG-based oracle, see Figure 4. For that, we present to a diverse set of 20 humans from different countries and backgrounds with valid counterfactuals and ask them whether the main attribute being classified by the ML model is present in the image or not. We use a subset of CelebA containing a random sample of 4 images per attribute, each one classified by the VGG$_{\text{Face}}$ oracle as containing the attribute with the following levels of confidence: $[0.1, 0.4, 0.6, 0.9]$. From each of these 160 images, we generated counterfactu-

Method	Human $\neq$ ML Classifier (real non-trivial)	Correlation	p-value
xGEM+ [18]	38.37%	0.37	0.000
DiVE	38.65%	0.25	0.002
DiVE$_{\text{Random}}$	38.89%	0.24	0.001
DiVE$_{\text{Fisher}}$	40.56%	0.17	0.023
DiVE$_{\text{FisherSpectral}}$	**41.90%**	0.23	0.001

Table 2: Human evaluation. The first column contains the percentage of non-trivial counterfactuals from the perspective of the human oracle. These counterfactuals confuse the ML classifier without changing the main attribute being classified from the perspective of a human. The second column contains the Pearson correlation between the human and the oracle's predictions. The third column contains the p-value for a t-test with the null hypothesis of the human and oracle predictions being uncorrelated.


Do you see the following facial attribute? Straight_Hair

1 - The image least represents the attribute
2
3
4
5 - The image most represents the attribute
6 - Impossible to Tell
DISCARD IMAGE
LABEL LATER
SUBMIT IMAGE

Figure 4: Labelling interface. The user is presented with a counterfactual image and has to choose if the target attribute is present or not in the image.

als with xGEM+ [18], DiVE, DiVE$_{\text{Random}}$, DiVE$_{\text{Fisher}}$, and DiVE$_{\text{FisherSpectral}}$ and show the valid counterfactuals to the human annotators. Results are reported in Table 2. In the left column we observe that leveraging the Fisher information results in finding more non-trivial counterfactuals, which confuse the ML model without changing the main attribute being classified. In the second column we report the Pearson correlation between the oracle and the classifier predictions. A statistical inference test reveals a significant correlation (p-value$\leq$0.02).

7. Model Architecture

Table 3 presents the architecture of the encoder and decoder used in DiVE.

8. Model Algorithm

Algorithm 1 presents the steps needed for DiVE to generate explanations for a given ML model using a sample input image.

Table 3: DiCe architecture for 128 × 128 images. *ch* represents the channel width multiplier in each network.

RGB image $x \in \mathbb{R}^{128\times128\times3}$
ResBlock down $3ch \rightarrow 16ch$
ResBlock $16ch \rightarrow 32ch$
ResBlock down $32ch \rightarrow 32ch$
ResBlock $32ch \rightarrow 64ch$
ResBlock down $64ch \rightarrow 64ch$
ResBlock $64ch \rightarrow 128ch$
ResBlock down $128ch \rightarrow 128ch$
ResBlock $128ch \rightarrow 128ch$
ResBlock down $128ch \rightarrow 128ch$
IN, Swish, Linear $128ch \times 4 \times 4 \rightarrow 128ch$
IN, Swish, Linear $128ch \rightarrow 128ch$
IN, Swish, Linear $128ch \rightarrow 128ch \times 2$
$z \sim \mathcal{N}(\mu \in \mathbb{R}^{128}, \sigma \in \mathbb{R}^{128})$

(a) Encoder

$z \in \mathbb{R}^{128}$
Linear $128ch \rightarrow 128ch$
Linear $128ch \rightarrow 128ch$
Linear $128ch \rightarrow 128ch \times 4 \times 4$
ResBlock up $128ch \rightarrow 64ch$
ResBlock up $64ch \rightarrow 32ch$
ResBlock $32ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock $16ch \rightarrow 16ch$
IN, Swish, Conv $16ch \rightarrow 3$ tanh

(b) Decoder

9. Out-of-distribution experiment

We test the out-of-distribution (OOD) performance of DiVE with the Synbols dataset [21]. Synbols is an image generator with characters from the Unicode standard and the wide range of artistic fonts provided by the open font community. This provides us to better control on the features present in each set when compared to CelebA. We generate 100K black and white of 32×32 images from 48 characters in the latin alphabet and more than 1K

Algorithm 1: Generating Explanations

```
Input  : Sample image $x$, ML model $f(\cdot)$
Output : Generated Conterfactuals $\tilde{\mathbf{x}}$
Initialize the perturbations matrix parameter of size $n \times d$
$\boldsymbol{\Sigma} \leftarrow randn(\mu = 0, \sigma = 0.01)$

Get the original output from the ML model
$y \leftarrow f(x)$

Extract the latent features of the original input
$z \leftarrow q_\phi(x)$

Obtain fisher information on $z$
$f_z \leftarrow \boldsymbol{F}(z)$

Obtain $k$ partitions using spectral clustering
$\boldsymbol{P} \leftarrow SpectralClustering(f_z)$

Initialize counter
$i \leftarrow 0$

while $i < \tau$ do
    for each $\varepsilon, \boldsymbol{p} \in (\boldsymbol{\Sigma}, \boldsymbol{P})$ do
        Perturb the latent features
        $\tilde{\mathbf{x}} \leftarrow p_\theta(\mathbf{z} + \varepsilon)$

        Pass the perturbed image through the ML model
        $\hat{y} \leftarrow f(\tilde{\mathbf{x}})$

        Learn to reconstruct $\hat{Y}$ from $Y$
        $\mathcal{L} \leftarrow$ compute Eq. 4

        Update $\varepsilon$ while masking a subset of the gradients
        $\varepsilon \leftarrow \varepsilon + \frac{\partial \mathcal{L}}{\partial \varepsilon} \cdot p$
    end
    Update counter
    $i \leftarrow i + 1$
end
```

fonts (Figure 5). In order to train the VAE on a different disjoint character set, we randomly select the following 32 training characters: {a, b, d, e, f, g, i, j, l, m, n, p, q, r, t, y, z, à, á, ã, å, è, é, ê, ë, î, ñ, ò, ö, ù, ú, û}. Counterfactuals are then generated for the remaining 16 characters: {c, h, k, o, s, u, v, w, x, â, ì, í, ï, ó, ô, ü}.

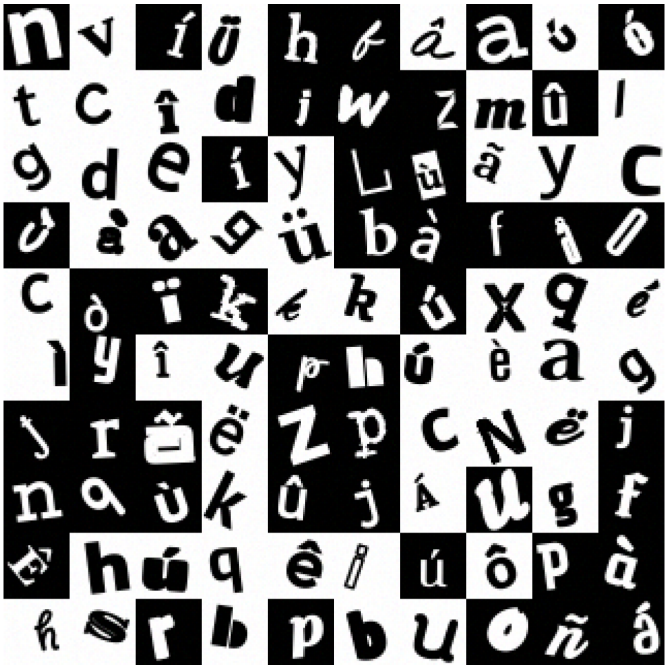


Figure 5: Sample of the synbols dataset.

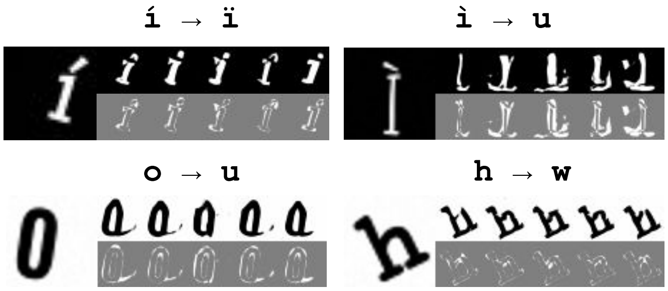


Figure 6: Successful counterfactuals for four different synbols in the OOD regime. Each sample consists of the original image in bigger size, five different counterfactuals generated by DiVE$_{\text{FisherSpectral}}$, and the difference in pixel space with respect to the original image (gray background). The header in each sample indicates the target class, *e.g.*, ï, u, w. All the counterfactuals are predicted by the ML model as belonging to the target class and differ from the oracle (*non-trivial*).

DiVE's objective is to discover biases on the ML model and the data. Thus, we use the *font* attribute in order to bias each of the characters on small disjoint subsets of fonts. Font subsets are chosen so that they are visually similar. In order to assess their similarity, we train a ResNet12 [25] to classify the fonts of the 100K images and calculate similarity in embedding space. Concretely, we use K-Means to obtain 16 clusters which are associated with each of the 16 characters used for counterfactual generation. The font assignments are reported in Table 4. Results for four different random counterfactuals are displayed in Figure 6.

c	Anonymous Pro, Architects Daughter, BioRhyme Expanded, Bruno Ace, Bruno Ace SC, Cantarell, Eligible, Eligible Sans, Futurespore, Happy Monkey, Lexend Exa, Lexend Mega, Lexend Tera, Lexend Zetta, Michroma, Orbitron, Questrial, Revalia, Stint Ultra Expanded, Syncopate, Topeka, Turret Road
h	Acme, Alatsi, Alfa Slab One, Amaranth, Archivo Black, Atma, Baloo Bhai, Bevan, Black Ops One, Bowlby One SC, Bubblegum Sans, Bungee, Bungee Inline, Calistoga, Candal, CantoraOne, Carter One, Ceviche One, Changa One, Chau Philomene One, Chela One, Chivo, Concert One, Cookbook Title, Days One, Erica One, Francois One, Fredoka One, Fugaz One, Galindo, Gidugu, Gorditas, Gurajada, Halt, Haunting Spirits, Holtwood One SC, Imperial One, Jaldi, Jockey One, Jomhuria, Joti One, Kanit, Kavoon, Keania One, Kenia, Lalezar, Lemon, Lilita One, Londrina Solid, Luckiest Guy, Mitr, Monoton, Palanquin Dark, Passero One, Passion, Passion One, Patua One, Paytone One, Peace Sans, PoetsenOne, Power, Pridi, Printed Circuit Board, Rakkas, Rammetto One, Ranchers, Rowdies, Rubik Mono One, Rubik One, Russo One, Saira Stencil One, Secular One, Seymour One, Sigmar One, Skranji, Squada One, Suez One, Teko, USSR STENCIL, Ultra, Unity Titling, Unlock, Viafont, Wattauchimma, simplicity
k	Abel, Anaheim, Antic, Antic Slab, Armata, Barriecito, Bhavuka, Buda, Chilanka, Comfortaa, Comic Neue, Cutive, Cutive Mono, Delius, Delius Unicase, Didact Gothic, Duru Sans, Dustismo, Fauna One, Feronia, Flamenco, Gafata, Glass Antiqua, Gothic A1, Gotu, Handlee, IBM Plex Mono Light, IBM Plex Sans Condensed Light, IBM Plex Sans Light, Indie Flower, Josefin Sans Std, Kite One, Manjari, Metrophobic, Miriam Libre, News Cycle, Offside, Overlock, Overlock SC, Panefresco 250wt, Pavanam, Pontano Sans, Post No Bills Colombo Medium, Puritan, Quattrocento Sans, Quicksand, Ruda, Scope One, Snippet, Sulphur Point, Tajawal Light, Text Me One, Thabit, Unkempt, Varta, WeblySleek UI, Yaldevi Colombo Light
o	Abhaya Libre Medium, Abyssinica SIL, Adamina, Alice, Alike, Almendra SC, Amethysta, Andada, Aquifer, Arapey, Asar, Average, Baskervville, Brawler, Cambo, Della Respira, Donegal One, Esteban, Fanwood Text, Fenix, Fjord, GFS Didot, Gabriela, Goudy Bookletter 1911, Habibi, HeadlandOne, IM FELL French Canon, IM FELL French Canon SC, Inika, Jomolhari, Karma, Kreon, Kurale, Lancelot, Ledger, Linden Hill, Linux Libertine Display, Lustria, Mate, Mate SC, Metamorphous, Milonga, Montaga, New Athena Unicode, OFL Sorts Mill Goudy TT, Ovo, PT Serif Caption, Petrona, Poly, Prociono, Quando, Rosarivo, Sawarabi Mincho, Sedan, Sedan SC, Theano Didot, Theano Modern, Theano Old Style, Trocchi, Trykker, Uchen
s	Abril Fatface, Almendra, Arbutus, Asset, Audiowide, BPdotsCondensed, BPdotsCondensedDiamond, Bigshot One, Blazium, Bowlby One, Brazier Flame, Broken Glass, Bungee Outline, Bungee Shade, Butcherman, CAT Kurier, Cabin Sketch, Catenary Stamp, Chango, Charmonman, Cherry Bomb, Cherry Cream Soda, Chonburi, Codystar, Coiny, Corben, Coustard, Creepster, CriminalHand, DIN Schablonierschrift Cracked, Devonshire, Diplomata, Diplomata SC, Dr Sugiyama, DrawveticaMini, Eater, Elsie, Emblema One, Faster One, Flavors, Fontdiner Swanky, Fredericka the Great, Frijole, Fruktur, Geostar, Geostar Fill, Goblin One, Gravitas One, Hanalei Fill, Irish Grover, Kabinett Fraktur, Katibeh, Knewave, Leckerli One, Lily Script One, Limelight, Linux Biolinum Outline, Linux Biolinum Shadow, Lucien Schoenschriftv CAT, Membra, Metal Mania, Miltonian Tattoo, Modak, Molle, Mortified Drip, Mrs Sheppards, Multivac, New Rocker, Niconne, Nosifer, Notable, Oleo Script, Open Flame, Overhaul, Paper Cuts 2, Peralta, Piedra, Plaster, Poller One, Porter Sans Block, Press Start 2P, Purple Purse, Racing Sans One, Remix, Ringling, Risaltyp, Ruslan Display, Rye, Sail, Sanidana, Sarina, Sarpanch, Schulze Werbekraft, Severely, Shojumaru, Shrikhand, Slackey, Smokum, Sniglet, Sonsie One, Sortefax, Spicy Rice, Stalin One, SudegnakNo2, Sun Dried, Tangerine, Thickhead, Tippa, Titan One, Tomorrow, Trade Winds, VT323, Vampiro One, Vast Shadow, Video, Vipond Angular, Wallpoet, Yesteryear, Zilla Slab Highlight
u	Alef, Alegreya Sans, Almarai, Archivo, Arimo, Arsenal, Arya, Assistant, Asul, Averia Libre, Averia Sans Libre, Be Vietnam, Belleza, Cambay, Comme, Cousine, DM Sans, Darker Grotesque, DejaVu Sans Mono, Dhyana, Expletus Sans, Hack, Istok Web, KoHo, Krub, Lato, Liberation Mono, Liberation Sans, Libre Franklin, Luna Sans, Marcellus, Martel Sans, Merriweather Sans, Monda, Mplus 1p, Mukta, Muli, Niramit, Noto Sans, Open Sans, Open Sans Hebrew, PT Sans, PT Sans Caption, Raleway, Rambla, Roboto Mono, Rosario, Rounded Mplus 1c, Sansation, Sarabun, Sarala, Scada, Sintony, Tajawal, Tenor Sans, Voces, Yantramanav
v	Alex Toth, Antar, Averia Gruesa Libre, Baloo Bhai 2, Bromine, Butterfly Kids, Caesar Dressing, Cagliostro, Capriola, Caveat, Caveat Brush, Chelsea Market, Chewy, Class Coder, Convincing Pirate, Copse, Counterproductive, Courgette, Courier Prime, Covered By Your Grace, Damion, Dear Old Dad, Dekko, Domestic Manners, Dosis, Erica Type, Erika Ormig, Espresso Dolce, Farsan, Finger Paint, Freckle Face, Fuckin Gwenhwyfar, Gloria Hallelujah, Gochi Hand, Grand Hotel, Halogen, IM FELL DW Pica, IM FELL DW Pica SC, IM FELL English SC, Itim, Janitor, Junior CAT, Just Another Hand, Just Me Again Down Here, Kalam, Kodchasan, Lacquer, Lorem Ipsum, Love Ya Like A Sister, Macondo, Mali, Mansalva, Margarine, Marmelad, Matias, Mogra, Mortified, Nunito, Objective, Oldenburg, Oregano, Pacifico, Pangolin, Patrick Hand, Patrick Hand SC, Pecita, Pianaforma, Pompiere, Rancho, Reenie Beanie, Rock Salt, Ruge Boogie, Sacramento, Salsa, Schoolbell, Sedgwick Ave, Sedgwick Ave Display, Shadows Into Light, Short Stack, SirinStencil, Sofadi One, Solway, Special Elite, Sriracha, Stalemate, Sue Ellen Francisco, Sunshiney, Supercomputer, Supermercado, Swanky and Moo Moo, Sweet Spots, Varela Round, Vibur, Walter Turncoat, Warnes, Wellfleet, Yellowtail, Zeyada
w	Alata, Alte DIN 1451 Mittelschrift, Alte DIN 1451 Mittelschrift gepraegt, Amble, Arvo, Asap, Asap VF Beta, Athiti, Atomic Age, B612, B612 Mono, Barlow, Barlow Semi Condensed, Basic, Blinker, Bree Serif, Cairo, Cello Sans, Chakra Petch, Changa Medium, Cherry Swash, Crete Round, DejaVu Sans, Doppio One, Droid Sans, Elaine Sans, Encode Sans, Encode Sans Condensed, Encode Sans Expanded, Encode Sans Semi Condensed, Encode Sans Wide, Exo, Exo 2, Federo, Fira Code, Fira Sans, Fira Sans Condensed, Gontserrat, HammersmithOne, Hand Drawn Shapes, Harmattan, Heebo, Hepta Slab, Hind, Hind Kochi, IBM Plex Mono, IBM Plex Mono Medium, IBM Plex Sans, IBM Plex Sans Condensed, Iceland, Josefin Sans, Krona One, Livvic, Mada, Magra, Maven Pro, Maven Pro VF Beta, Mirza, Montserrat, Montserrat Alternates, Myanmar Khyay, NATS, Nobile, Oxanium, Play, Poppins, Prompt, Prosto One, Proza Libre, Quantico, Red Hat Text, Reem Kufi, Renner*, Righteous, Saira, Saira SemiCondensed, Semi, Share, Signika, Soniano Sans Unicode, Source Code Pro, Source Sans Pro, Spartan, Viga, Yatra One, Zilla Slab
x	Abhaya Libre, Amita, Antic Didone, Aref Ruqaa, Arima Madurai, Bellefair, CAT Childs, CAT Linz, Cardo, Caudex, Cinzel, Cormorant, Cormorant Garamond, Cormorant SC, Cormorant Unicase, Cormorant Upright, Cuprum, Domine, Dustismo Roman, El Messiri, Fahkwang, FogtwoNo5, Forum, Galatia SIL, Gayathri, Gilda Display, Glegoo, GlukMixer, Gputeks, Griffy, Gupter, IBM Plex Serif Light, Inria Serif, Italiana, Judson, Junge, Libre Caslon Display, Linux Biolinum Capitals, Linux Biolinum Slanted, Linux Libertine Capitals, Lobster Two, Marcellus SC, Martel, Merienda, Modern Antiqua, Montserrat Subrayada, Mountains of Christmas, Mystery Quest, Old Standard TT, Playfair Display, Playfair Display SC, Portmanteau, Prata, Pretzel, Prida36, Quattrocento, Resagnicto, Risque, Rufina, Spectral SC, Trirong, Viaoda Libre, Wes, kawoszeh, okolaks
â	Accuratist, Advent Pro, Archivo Narrow, Aubrey, Cabin Condensed, Convergence, Encode Sans Compressed, Farro, Fira Sans Extra Condensed, Galdeano, Gemunu Libre, Gemunu Libre Light, Geo, Gudea, Homenaje, Iceberg, Liberty Sans, Mohave, Nova Cut, Nova Flat, Nova Oval, Nova Round, Nova Slim, NovaMono, Open Sans Condensed, Open Sans Hebrew Condensed, PT Sans Narrow, Port Lligat Sans, Port Lligat Slab, Pragati Narrow, Rajdhani, Rationale, Roboto Condensed, Ropa Sans, Saira Condensed, Saira ExtraCondensed, Share Tech, Share Tech Mono, Strait, Strong, Tauri, Ubuntu Condensed, Voltaire, Yaldevi Colombo, Yaldevi Colombo Medium
ì	Aladin, Alegre Sans, Allan, Amarante, Anton, Antonio, Asap Condensed, Astloch, At Sign, Bad Script, Bahiana, Bahianita, Bangers, Barloesius Schrift, Barlow Condensed, Barrio, Bebas Neue, BenchNine, Berlin Email Serif, Berlin Email Serif Shadow, Berolina, Bertholdr Mainzer Fraktur, Biedermeier Kursiv, Big Shoulders Display, Big Shoulders Text, Bigelow Rules, Bimbo JVE, Bonbon, Boogaloo, CAT FrankenDeutsch, CAT Liebing Gotisch, Calligraserif, Casa Sans, Chicle, Combo, Contrail One, Crushed, DN Titling, Dagerotypos, Denk One, Digital Numbers, Dorsa, Economica, Eleventh Square, Engagement, Euphoria Script, Ewert, Fette Mikado, Fjalla One, Flubby, Friedolin, Galada, Germania One, Gianna, Graduate, Hanalei, Jacques Francois Shadow, Jena Gotisch, Jolly Lodger, Julee, Kanzler, Kavivanar, Kazmann Sans, Kelly Slab, Khand, Kotta One, Kranky, Lemonada, Loved by the King, Maiden Orange, Marck Script, MedievalSharp, Medula One, Merienda One, Miltonian, Mouse Memoirs, Nova Script, Odibee Sans, Oswald, Paprika, Pathway Gothic One, Penguin Attack, Pirata One, Pommern Gotisch, Post No Bills Colombo, Princess Sofia, Redressed, Ribeye Marrow, Rum Raisin, Sancreek, Saniretro, Sanitechtro, Sevillana, Six Caps, Slim Jim, Smythe, Sofia, Sportrop, Staatliches, Stint Ultra Condensed, Tillana, Tulpen One, Underdog, Unica One, UnifrakturMaguntia, Yanone Kaffeesatz
í	Amatic SC, Bellota, Bellota Text, Bernardo Moda, Blokletters Balpen, Blokletters Potlood, Bubbler One, Bungee Hairline, Coming Soon, Crafty Girls, Gemunu Libre ExtraLight, Give You Glory, Gold Plated, Gruppo, IBM Plex Mono ExtraLight, IBM Plex Mono Thin, IBM Plex Sans Condensed ExtraLight, IBM Plex Sans Condensed Thin, IBM Plex Serif ExtraLight, IBM Plex Serif Thin, Julius Sans One, Jura, Lazenby Computer, Life Savers, Londrina Outline, Londrina Shadow, Mada ExtraLight, Mada Light, Major Mono Display, Megrim, Nixie One, Northampton, Over the Rainbow, Panefresco 1wt, Poiret One, Post No Bills Colombo Light, Raleway Dots, RawengulkSans, Sansation Light, Shadows Into Light Two, Sierra Nevada Road, Slimamif, Snowburst One, Tajawal ExtraLight, Terminal Dosis, Thasadith, The Girl Next Door, Thin Pencil Handwriting, Vibes, Waiting for the Sunrise, Wire One, Yaldevi Colombo ExtraLight
ï	Amiri, Buenard, Caladea, Charis SIL, Crimson Text, DejaVu Serif, Dita Sweet, Droid Serif, EB Garamond, Eagle Lake, Fondamento, Frank Ruhl Libre, Gelasio, Gentium Basic, Gentium Book Basic, Ibarra Real Nova, Junicode, Liberation Serif, Libre Baskerville, Libre Caslon Text, Linux Biolinum, Linux Libertine, Linux Libertine Slanted, Lusitana, Manuale, Merriweather, Noticia Text, PT Serif, Scheherazade, Spectral, Taviraj, Unna, Vesper Devanagari Libre
ó	ABeeZee, Actor, Aldrich, Alegreya Sans SC, Aleo, Amiko, Andika, Annie Use Your Telescope, Average Sans, Bai Jamjuree, Baumans, Belgrano, BioRhyme, Biryani, Cabin, Cabin VF Beta, Calling Code, Carme, Carrois Gothic, Carrois Gothic SC, Catamaran, Changa Light, Convincing, Droid Sans Mono, Electrolize, Englebert, Fresca, GFS Neohellenic, Imprima, Inconsolata, Inder, Inria Sans, Josefin Slab, K2D, Karla, Kulim Park, Lekton, Lexend Deca, Mako, McLaren, Meera Inimai, Molengo, Numans, Orienta, Overpass, Overpass Mono, Oxygen Mono, PT Mono, Panefresco 400wt, Podkova, Podkova VF Beta, Red Hat Display, Rhodium Libre, Ruluko, Sanchez, Sani Trixie, Sawarabi Gothic, Sen, Shanti, Slabo 13px, Slabo 27px, Sometype Mono, Space Mono, Spinnaker, Tuffy, TuffyInfant, TuffyScript, Ubuntu Mono, Varela
ô	Aguafina Script, Akronim, Alex Brush, Arizonia, Beth Ellen, Bilbo, Brausepulver, Calligraffitti, Cedarville Cursive, Charm, Clicker Script, Condiment, Cookie, Dancing Script, Dawning of a New Day, Dynalight, Felipa, Great Vibes, Herr Von Muellerhoff, Homemade Apple, Italianno, Jim Nightshade, Kaushan Script, Kristi, La Belle Aurore, League Script, Meddon, Meie Script, Mervale Script, Miama, Miniver, Miss Fajardose, Monsieur La Doulaise, Montez, Mr Bedfort, Mr Dafoe, Mr De Haviland, Mrs Saint Delafield, Norican, Nothing You Could Do, Parisienne, Petit Formal Script, Pinyon Script, Playball, Promocyja, Quintessential, Qwigley, Rochester, Romanesco, Rouge Script, Ruthie, Satisfy, Seaweed Script, Srisakdi, Vengeance
ü	Aclonica, Alegreya, Alegreya SC, Andada SC, Artifika, Averia Serif Libre, Balthazar, Bentham, Bitter, Cantata One, Crimson Pro, Croissant One, DM Serif Display, Eczar, Emilys Candy, Enriqueta, Faustina, Federant, Girassol, Grenze, Halant, Henny Penny, Hermeneus One, IBM Plex Serif, IBM Plex Serif Medium, IM FELL Double Pica, IM FELL Double Pica SC, IM FELL English, IM FELL Great Primer SC, Inknut Antiqua, Jacques Francois, Judges, Kameron, Laila, Lateef, Literata, Lora, Maitree, Markazi Text, Marko One, Monteiro Lobato, Original Surfer, Psicopatologia de la Vida Cotidiana, Radley, Rasa, Ribeye, Roboto Slab, Rokkitt, Rozha One, Sahitya, Sansita, Simonetta, Source Serif Pro, Spirax, Stardos Stencil, Stoke, Sumana, Uncial Antiqua, Vidaloka, Volkhov, Vollkorn, Vollkorn SC

Table 4: Font clusters assigned to each character.

$\text{DiVE}_{\text{FisherSpectral}}$ successfuly confuses the ML model without changing the oracle prediction, revealing biases of the ML model.

10. Details on the Bias Detection Metric

In Table 1 in the main text, we follow the procedure in first developped in [18] and adapted in [34] and report a confounding metric for bias detection. Namely, the “Male” and “Female” is the accuracy of the oracle on those class conditioned on the target label of the original image. For example, we can see that the generated explanations for the the biased classifier, most methods generated an higher amount of Non-smiling females and smiling males, which was expected. The confounding metric, denoted as overall, is the fraction of generated explanations for which the gender was changed with respect to the original image. It thus reflect the magnitude of the the bias as approximated by the explainers. Singla et al. [34] consider that a model is better than another if the confounding metric is the highest on f_{biased} and the lowest on f_{unbiased}.

This is however not entirely true. There is no guarantee that f_{biased} will perfectly latch on the spurious correlation. In that case, an explainer's ratio could potentially be too high which would reflect an overestimation of the bias. We thus need to a way to quantify the gender bias in each model. To do so, we look at the difference between the classifiers accuracy on "Smiling" when the image is of a "Male" versus a "Female". Intuitively, the magnitude of this difference approximates how much the classifier latched on the "Male" attribute to make its smiling predictions. We compute the same metric for in the non-smiling case. We average both of them, which we refer as ground truth in Table 1 (main text). As expected, this value is high for the f_{biased} and low for f_{unbiased}. Formally, the ground truth is computed as

$$\mathbb{E}_{a\sim p(a)}\Big[\mathbb{E}_{x,y\sim p(x,y|a)}\big[\big|\mathbb{1}[y=f(x)|a=a_1] \\ -\mathbb{1}[y=f(x)|a=a_2]\big|\big]\Big], \tag{2}$$

where a represents the attribute, in this case the gender.

References

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018. 1

[2] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[3] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018. 3

[4] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019. 1

[5] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018. 1

[6] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. *arXiv preprint arXiv:1705.07857*, 2017. 1

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009. 3

[8] E. Denton, B. Hutchinson, M. Mitchell, and T. Gebru. Detecting bias with generative counterfactual face attribute augmentation. *arXiv preprint arXiv:1906.06439*, 2019. 2

[9] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018. 2

[10] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *International Conference on Computer Vision*, 2017. 1

[11] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019. 4

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 2

[13] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. *arXiv preprint arXiv:1904.07451*, 2019. 2

[14] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34 (6):14–23, 2019. 1

[15] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi. Black box explanation by learning image exemplars in the latent feature space. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 189–205. Springer, 2019. 1

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Computer Vision and Pattern recognition*, 2017. 4

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015. 3

[18] S. Joshi, O. Koyejo, B. Kim, and J. Ghosh. xgems: Generating examplars to explain black-box models. *arXiv preprint arXiv:1806.08867*, 2018. 2, 5, 7

[19] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, pages 895–905, 2020. 1

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[21] A. Lacoste, P. Rodríguez López, F. Branchaud-Charron, P. Atighehchian, M. Caccia, I. H. Laradji, A. Drouin, M. Craddock, L. Charlin, and D. Vázquez. Synbols: Probing learning algorithms with synthetic datasets. *Advances in Neural Information Processing Systems*, 33, 2020. 5

[22] J. Lucas, G. Tucker, R. Grosse, and M. Norouzi. Understanding posterior collapse in generative latent variable models. 2019. 3

[23] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 2017. 1

[24] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Conference on Fairness, Accountability, and Transparency*, 2020. 2

[25] B. Oreshkin, P. Rodríguez López, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31:721–731, 2018. 6

[26] M. Pawelczyk, K. Broelemann, and G. Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, pages 3126–3132, 2020. 2

[27] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general con-

ditioning layer. *arXiv preprint arXiv:1709.07871*, 2017. 3

[28] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020. 2

[29] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *International Conference on Learning Representations*, 2018. 3

[30] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should i trust you?” explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining*, 2016. 1

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, 2017. 1

[32] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017. 1

[33] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[34] S. Singla, B. Pollack, J. Chen, and K. Batmanghelich. Explanation by progressive exaggeration. In *International Conference on Learning Representations*, 2020. 2, 3, 4, 7

[35] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1

[36] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[37] A. Van Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019. 2

[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 1

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition*, 2016. 1