ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240584

Probabilistically Plausible Counterfactual Explanations with Normalizing Flows

Patryk Wielopolski[a,*], **Oleksii Furman**[a], **Jerzy Stefanowski**[b] **and Maciej Zięba**[a,c]

[a]Wrocław University of Science and Technology
[b]Poznań University of Technology
[c]Tooploox Sp. z o.o.

Abstract. We present PPCEF, a novel method for generating probabilistically plausible counterfactual explanations (CFs). PPCEF advances beyond existing methods by combining a probabilistic formulation that leverages the data distribution with the optimization of plausibility within a unified framework. Compared to reference approaches, our method enforces plausibility by directly optimizing the explicit density function without assuming a particular family of parametrized distributions. This ensures CFs are not only valid (i.e., achieve class change) but also align with the underlying data's probability density. For that purpose, our approach leverages normalizing flows as powerful density estimators to capture the complex high-dimensional data distribution. Furthermore, we introduce a novel loss function that balances the trade-off between achieving class change and maintaining closeness to the original instance while also incorporating a probabilistic plausibility term. PPCEF's unconstrained formulation allows for an efficient gradient-based optimization with batch processing, leading to orders of magnitude faster computation compared to prior methods. Moreover, the unconstrained formulation of PPCEF allows for the seamless integration of future constraints tailored to specific counterfactual properties. Finally, extensive evaluations demonstrate PPCEF's superiority in generating high-quality, probabilistically plausible counterfactual explanations in high-dimensional tabular settings.

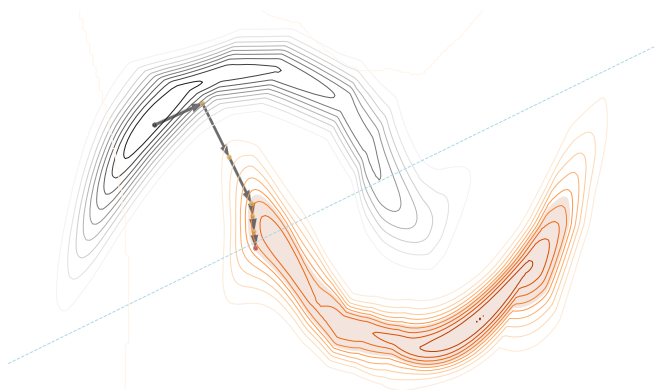

Figure 1: Probabilistically Plausibile Counterfactual Explanation Estimation Process on the Moons Dataset. We show an evolution of an instance from the initial instance (black dot) to the final counterfactual (red dot) against the linear classifier's decision boundary (blue line) and density threshold contours, highlighting the method's trajectory towards achieving target classification and probabilistic plausibility condition.

1 Introduction

Counterfactual explanations (briefly *counterfactuals*, and abbreviated as CF) are one particular type of such explanations of black box model predictions that provide information about how feature values of an example should be changed to obtain a more desired prediction of the model (i.e., to change its target decision) [30]. On the one hand, by interacting with the model using counterfactuals, the user can better understand how the system works by exploring "what would have happened if..." scenarios. On the other hand, a good counterfactual provides a practical recommendation to the user about what changes are needed in order to achieve the desired outcome.

There are many practical applications for counterfactual explanations, including loan or insurance decisions [31], recruitment processes [21], the discovery of chemical compounds [32], medical diagnosis [17], and many others, see, e.g., the recent survey [10].

More formally, a counterfactual explanation is an alternative input instance, denoted as $\mathbf{x}'$, which is minimally modified from the description of the original instance $\mathbf{x}_0$, such that the output of the classifier h changes from the original decision $y = h(\mathbf{x}_0)$ to a specific desired outcome $y' = h(\mathbf{x}')$.

Up to now, several algorithms for generating counterfactual explanations have been introduced. They are based on different principles, and for comprehensive surveys, see, e.g., [10, 30]. Depending on the specific method, some properties of counterfactuals are expected to be met, such as *validity* of the decision change, *proximity* to the input instance, *sparsity* of recommended changes, their *actionability*, i.e., the counterfactual should not modify immutable features or violate monotonic constraints, and *plausibility* of locating the counterfactual within a high-density region of the data, ensuring that the proposed counterfactuals are realistic and feasible within the context of the observed data distribution.

Many of these methods are inspired by the formulation of Wachter et al. [31], which proposed framing counterfactual explanations as an unconstrained optimization problem. For a prediction function h and an input $\mathbf{x}_0 \in \mathbb{R}^d$, a counterfactual $\mathbf{x}' \in \mathbb{R}^d$ is computed by solving:

$$\arg\min_{\mathbf{x}' \in \mathbb{R}^d} \ell(h(\mathbf{x}'), y') + C \cdot d(\mathbf{x}_0, \mathbf{x}'). \tag{1}$$

* Corresponding Author. Email: patryk.wielopolski@pwr.edu.pl.

In this formulation, $\ell(\cdot,\cdot)$ represents a classification loss function, $d(\cdot,\cdot)$ is a penalty for deviation from the original input $\mathbf{x}_0$, and the term $C \geq 0$ serves as the regularization strength modifier.

An alternative approach [2] frames counterfactual explanations as a constrained optimization problem. This perspective focuses on directly finding the minimal perturbation required to achieve the target prediction under the constraint that the model's prediction for the counterfactual instance meets the specified criterion. Mathematically, this is represented as:

$$\arg\min_{\mathbf{x}' \in \mathbb{R}^d} d(\mathbf{x}_0, \mathbf{x}') \quad \text{s.t.} \quad h(\mathbf{x}') = y'. \tag{2}$$

In our study, we want to pay special attention to the *plausibility* of counterfactuals. Referring to arguments of [10], a counterfactual is plausible if the feature values describing the example are coherent (sufficiently similar) with those present in the original data X. This means it should be located in sufficiently dense regions of original instances in X from the target class. Plausibility helps in increasing users' trust in the explanation: it would be hard to trust a counterfactual if it is a combination of features that are unrealistic with respect to existing examples.

In previous works, plausibility has often been verified by simple k-neighbourhood analysis of the counterfactual with respect to the original data [26, 14, 27]. Few other approaches [3] model the conditional density in the target class and try to find the counterfactual example with the density value above the given threshold. Although the problem is quite well mathematically defined, the current methods apply simple approaches like kernel density estimators or a mixture of Gaussians to model conditional distributions that are difficult to apply for high-dimensional data. Moreover, the problem of estimating valid and plausible counterfactuals is defined as a complex constrained optimization problem with strict convexity assumptions [3]. Finally, the currently proposed methods, while providing valid counterfactuals, struggle to consistently produce observations that fulfill the plausibility criteria.

In this paper, we introduce PPCEF: Probabilistically Plausibile Counterfactual Explanations using Normalizing Flows - a novel approach to estimate counterfactual explanations for differentiable classifiers tailored for tabular problems. It includes a novel, unconstrained formulation of the problem that enables direct estimation of the plausibility property - to the best of our knowledge, a characteristic previously not achieved in the literature. For that purpose, we design loss functions to satisfy both validity and plausibility constraints and minimize the distance to the original example in a balanced way (see an example in Fig. 1). Our approach incorporates plausibility in the probabilistic sense by targeting observations with a probability density exceeding a predefined threshold [3]. Unlike existing methods limited to specific estimators of families of density functions, ours employs any differentiable conditional density model. Moreover, we postulate to utilize *conditional normalizing flows* for density estimation [24], ensuring independence from specific parameterized distribution families while enabling direct calculation of density values for complex, high-dimensional data. Finally, PPCEF leverages efficient batch processing utilizing gradient-based optimization techniques, leading to significant computational gains compared to previous methods.

To summarize, our contributions are as follows:

- The formulation of counterfactual explanations within an unconstrained optimization framework employing direct optimization of plausibility and novel loss functions.
- The utilization of normalizing flows as density estimators to capture the complex high-dimensional data distribution effectively.
- The experimental evaluations demonstrating PPCEF's ability to efficiently generate high-quality, probabilistically plausible counterfactuals in high-dimensional tabular datasets for both binary and multiclass classification problems, outperforming existing reference methods.

2 Related Works

2.1 Plausible Counterfactual Explanations

The approaches for obtaining plausible counterfactual explanations are primarily categorized into *endogenous* and *exogenous* ones [10]. Endogenous counterfactuals are crafted using feature values from existing data instances, ensuring their naturally occurring status and grounding them in real-world contexts, thereby enhancing their plausibility. In contrast, exogenous counterfactuals are generated through methods such as interpolations or random data generation, which do not strictly rely on existing dataset features. While this offers greater flexibility, it does not inherently assure the plausibility of these counterfactuals, as they might represent feature combinations not found in actual data.

2.1.1 Endogenous Counterfactual Explanations

Endogenous approaches to counterfactual explanations revolve around leveraging existing instances within the dataset to generate plausible counterfactuals. These methods, which include instance-based or case-based approaches, primarily utilize nearest neighbors' techniques to identify instances that closely resemble the input but yield different outcomes.

Examples of endogenous methods include the Nearest-Neighbor Counterfactual Explainer (NNCE) [26], selecting similar yet outcome-divergent instances from the dataset as counterfactuals. The Case-Based Counterfactual Explainer (CBCE) [14] forms 'explanation cases' by pairing similar instances with contrasting outcomes, creating counterfactuals by merging features from these pairs. Extending this concept, the approach by Smyth and Keane [27] adapts to k-nearest neighbors, utilizing multiple nearest candidates for generating counterfactuals. Feasible and Actionable Counterfactual Explanations (FACE) [23] constructs a graph over data points, applying user-defined parameters to find actionable paths to desired outcomes. Lastly, PROPLACE [12] employs bi-level optimization and Mixed-Integer Linear Programming, generating robust counterfactuals from Δ-robust nearest neighbors that closely align with data distribution and model robustness.

2.1.2 Exogenous Counterfactual Explanations

In the landscape of exogenous counterfactual explanations, methods generally involve introducing external modifications to original instances, diverging from reliance on existing instances and their features. These approaches utilize a range of computational techniques, such as autoencoders, linear programming, gradient-based methods, and generative models, to ensure that the resulting counterfactuals are plausible.

Firstly, the Contrastive Explanation Method (CEM) [5] innovates by adding perturbations to an instance and utilizing an autoencoder to verify the closeness of the modified instance to known data, ensuring plausibility. Meanwhile, the Diverse Coherent Explanations (DCE) [25] method leverages linear programming to create varied counterfactuals, with additional linear constraints to maintain both

diversity and plausibility. Further, the Distribution-Aware Counterfactual Explanation (DACE) [13] method incorporates the Mahalanobis distance and Local Outlier Factor (LOF) in its loss function, focusing on minimizing this distance while keeping a low LOF score to signify higher plausibility. The Diverse Counterfactual Explanations (DICE) [18] approach involves solving an optimization problem to generate multiple counterfactuals, with a specific emphasis on the diversity and actionability of these counterfactuals to determine their plausibility. Additionally, Counterfactual Explanations Guided by Prototypes (CEGP) [15] adopts a similar loss function to CEM but introduces a prototype-based loss term. This guides perturbations towards a counterfactual that aligns with the data distribution of a specific class, using the encoder of an autoencoder based on the average encoding of the nearest instances in the latent space with the same class label.

Within the field of exogenous counterfactual explanations, a subcategory particularly relevant to our work utilizes deep generative models. Variational Autoencoders (VAEs) are exploited in methods like Example-Based Counterfactual (EBCF) [16] and the approach by Vercheval and Pizurica [29]. EBCF incorporates known causal relationships into the VAE, promoting realistic counterfactuals. The method by Vercheval and Pizurica [29] enables visual counterfactual generation through VAE-based latent space exploration. Generative Adversarial Networks (GANs) play a crucial role in the PCATTGAN approach [1]. It utilizes adversarial examples within a multi-objective optimization framework to create plausible counterfactuals, considering validity, minimality, and a notion of plausibility defined as human-understandable, non-automated changes. Diffusion models underpin methods proposed in [11, 4]. While these approaches specialize in visual counterfactual generation, their focus lies primarily on counterfactual sampling, not controlling plausibility via density-based optimization. Lastly, Normalizing Flow-based methods [8, 9] center on pinpointing counterfactuals within their latent spaces. These methods leverage the invertible nature of normalizing flows to explore counterfactual regions in the latent representation of the data.

All of the reference methods, except Artelt and Hammer [3], do not provide an explicit probabilistic formulations of plausibility. Compared to Artelt and Hammer [3], we propose an alternative problem formulation in unconstrained form with no prior constraints on the density model.

3 Background

In this work, we consider the problem formulation of probabilistically plausible counterfactual explanations introduced by Artelt and Hammer [3]. This approach extends the problem formulation given by eq. (2) by adding a target-specific density constraint to enforce the plausibility of counterfactuals using a probabilistic framework. The constrained optimization problem is formulated as follows:

$$\arg\min_{\mathbf{x}'\in\mathbb{R}^d} d(\mathbf{x}_0, \mathbf{x}') \tag{3a}$$

$$\text{s.t.} \quad h(\mathbf{x}') = y' \tag{3b}$$

$$\delta \leq p(\mathbf{x}'|y'), \tag{3c}$$

where $p(\mathbf{x}'|y')$ denotes conditional probability of the counterfactual explanation $\mathbf{x}'$ under desired target class value y' and δ represents the density threshold.

This approach's crucial aspect is finding the proper model to represent the conditional density function $p(\mathbf{x}|y)$. Typically, kernel density estimators (KDEs) are used to model conditional densities, but the use of non-linear kernels results in the highly non-convex optimization problem formulation. Gaussian Mixture Model (GMM) can be applied alternatively, but convexity constraints are still not satisfied. To facilitate the desired optimization process, the authors of Artelt and Hammer [3] propose to approximate the density value $p(\mathbf{x}'|y')$ using a component-wise maximum of GMM components:

$$\hat{p}_G(\mathbf{x}'|y') = \max_j \Big(\pi_{j,y'}\mathcal{N}(\mathbf{x}'|\boldsymbol{\mu}_{j,y'}, \boldsymbol{\Sigma}_{j,y'})\Big), \tag{4}$$

where $\boldsymbol{\mu}_{j,y'}$, $\boldsymbol{\Sigma}_{j,y'}$ and $\pi_{j,y'}$ are means, covariances and prior values for component j considering class y.

This approximation is transformed into a convex quadratic constraint for each GMM component j, resulting in the following formula:

$$(\mathbf{x}' - \boldsymbol{\mu}_{j,y'})^T \boldsymbol{\Sigma}_{j,y'} (\mathbf{x}' - \boldsymbol{\mu}_{j,y'}) + c_j \leq \delta', \tag{5}$$

where c_j is constant from the Gaussian normalization factor and $\delta' = -2\log\delta$.

For each component j, the optimization problem is solved, resulting in a set of convex programs - one for each component. This step is crucial because knowing beforehand which component will produce a feasible and plausible counterfactual is impossible. Finally, the counterfactual $\mathbf{x}'$ that yields the smallest value for the objective function is selected.

However, this approach has few limitations. First, the number of components should be predefined for each class. Second, the family of parametrized distributions limits the ability to adjust to a data distribution. Third, the approach is difficult to be applied to high-dimensional data due to the Gaussian components.

In order to cope with the listed limitations, we postulate to model conditional density function $p(\mathbf{x}|y)$ using the normalizing flows [24]. This group of models can adjust to very complex, high-dimensional data distributions, which allows for calculating the density value from the change-of-variable formula. Moreover, we propose an alternative unconstrained problem formulation that allows solving using a gradient-based approach for any differentiable representation of conditional distribution $p(\mathbf{x}|y)$.

4 Method

This section introduces a novel approach to the problem of plausible counterfactual explanation formulated by eq. (3). First, we reformulate the problem of calculating counterfactuals as unconstrained optimization suitable for direct, gradient-based optimization. Next, we show how to train the flow model to estimate the class-conditional distributions. Finally, we show how the counterfactuals can be efficiently estimated using a gradient-based approach.

4.1 *Unconstrained Probabilistically Plausible Counterfactual Explanations*

We consider a binary classification problem, $y \in \{0, 1\}$. However, our considerations can be easily extended to the multiclass case. Further, we consider a discriminative differentiable model (e.g., Logistic Regression or MLP) $p_d(y|\mathbf{x})$ and reformulate the validity constraint $h(\mathbf{x}') = y'$ as $p_d(y'|\mathbf{x}') \geq 0.5 + \epsilon$, where $\epsilon \to 0$, practically represented as small enough value close to 0.

We postulate the following unconstrained optimization problem:

$$\arg\min_{\mathbf{x}'\in\mathbb{R}^d} d(\mathbf{x}_0, \mathbf{x}') + \lambda \cdot \Big(\ell_v(\mathbf{x}', y') + \ell_p(\mathbf{x}', y')\Big), \tag{6}$$

where $\lambda = \infty$, practically, is large enough.

The loss $\ell_v(\mathbf{x}', y')$ component controls the validity constraint and is defined as follows:

$$\ell_v(\mathbf{x}', y') = \max\Big(0.5 + \epsilon - p_d(y'|\mathbf{x}'), 0\Big). \tag{7}$$

The Binary Cross Entropy (BCE) criterion can be used alternatively. However, using such criteria enforces 100% confidence of the discriminative model, while our approach aims at achieving the current classification accuracy with the margin controlled with the ϵ parameter. While using our criterion, the model can focus more on producing closer and more plausible counterfactuals, which we show in ablation studies.

Additionally, we extend the validity loss component to the multi-class scenario in the following way:

$$\ell_v(\mathbf{x}', y') = \max\Big(\max_{y \neq y'} p_d(y|\mathbf{x}') + \epsilon - p_d(y'|\mathbf{x}'), 0\Big), \tag{8}$$

where we replace the 0.5 threshold value with the highest probability value returned by the discriminative model, excluding the value for target class y'. This guarantees that $p_d(y|\mathbf{x}')$ will be higher than the most probable class among the remaining classes by the ϵ margin.

The loss component $\ell_p(\mathbf{x}', y')$ controls probabilistic plausibility constraint ($\delta \leq p(\mathbf{x}'|y')$) and is defined as:

$$\ell_p(\mathbf{x}', y') = \max\Big(\delta - p(\mathbf{x}'|y'), 0\Big), \tag{9}$$

where δ is the density threshold calculated in the same way as in [3], i.e., by utilizing the median of the training dataset. The conditional distribution $p(\mathbf{x}|y)$ can be represented by any differentiable model (e.g., Mixture of Gaussians, KDE). In this work, we postulate to model the distribution using conditional normalizing flow due to the flexibility and ability to adjust to multidimensional complex distributions. Thanks to the unconstrained problem formulation given by eq. (6) and differentiation assumption for the models, the counterfactuals can be easily calculated using a gradient-based approach.

4.2 *Probabistically Plausible Counterfactual Explanations via Normalizing Flow-based Density Estimation*

KDE or GMMs can be used to model the conditional distributions. However, those models have limited modeling capabilities due to the parametrized (usually Gaussian) form of $p(\mathbf{x}|y)$ or the inability to model high-dimensional data (KDE). Therefore, in this work, we postulate the use of a conditional normalizing flow model [24] to estimate the density for the joint distribution of the attributes for each class.

Normalizing Flows have surged in popularity within generative models due to their adaptability and the simplicity of training via direct negative log-likelihood (NLL) optimization. Their adaptability stems from the change-of-variable technique, which transforms a latent variable $\mathbf{z}$ with a known prior distribution $p(\mathbf{z})$ into an observed space variable $\mathbf{x}$ with an unknown distribution. This transformation occurs through a sequence of invertible (parametric) functions: $\mathbf{x} = \mathbf{f}_K \circ \cdots \circ \mathbf{f}_1(\mathbf{z}, y)$. Assuming a known prior $p(\mathbf{z})$ for $\mathbf{z}$, the conditional log-likelihood for $\mathbf{x}$ is expressed as:

$$\log \hat{p}_F(\mathbf{x}|y) = \log p(\mathbf{z}) - \sum_{k=1}^{K} \log \left| \det \frac{\partial \mathbf{f}_k}{\partial \mathbf{z}_{k-1}} \right|, \tag{10}$$

where $\mathbf{z} = \mathbf{f}_1^{-1} \circ \cdots \circ \mathbf{f}_K^{-1}(\mathbf{x}, y)$ is a result of the invertible mapping. The biggest challenge in normalizing flows is the choice of the invertible functions $\mathbf{f}_K, \ldots, \mathbf{f}_1$. Several solutions have been proposed in the literature to address this issue with notable approaches, including NICE [6], RealNVP [7], and MAF [19].

For a given training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ we simply train the conditional normalizing flow by minimizing negative log-likelihood:

$$Q = -\sum_{n=1}^{N} \log \hat{p}_F(\mathbf{x}_n|y_n), \tag{11}$$

where $\log \hat{p}_F(\mathbf{x}_n|y_n)$ is defined by eq. (10). The model is trained using a gradient-based approach applied to the flow parameters stored in $\mathbf{f}_k$ functions.

4.3 *Estimating Counterfactuals*

For a trained conditional normalizing flow, the counterfactual explanation can be easily calculated simply by optimizing the criterion given by eq. (6). The parameters of the flow model are frozen, and $\mathbf{x}'$ is optimized using the gradient-based procedure, starting from the point $\mathbf{x}_0$. To enhance the efficiency of our method, we have incorporated batch processing capabilities, allowing for the simultaneous calculation of multiple counterfactual explanations. This is achieved by aggregating instances and employing an average aggregation for loss calculation. Such a feature is notably absent in the other approaches compared to this study, providing our method with a distinct computational advantage.

5 Experiments

In this section, we aim to demonstrate and validate our counterfactual explanation method through a series of experiments. Initially, we illustrate our method's intuition with the Moons dataset and Logistic Regression model. Next, we compare our approach against the only reference method in a probabilistically plausible CFs area - Artelt and Hammer [3], as well as other established CF methods. This comparison focuses on the impact of plausibility on proximity metrics and time efficiency. Lastly, we conduct broader comparisons using other classifier models: Logistic Regression (LR), Multilayer Perceptron (MLP), and Neural Oblivious Decision Ensembles (NODE) [22]. The code for these experiments is publicly released on GitHub[1].

Datasets To evaluate PPCEF's effectiveness, we conducted experiments on seven numerical-only tabular datasets. Four datasets (Law, Heloc, Moons, and Audit) represent binary classification problems, whereas the first two datasets (Law and Heloc) are commonly used benchmarks for counterfactual explanation tasks. The remaining three datasets (Blobs, Digits, and Wine) address multiclass classification problems. Detailed descriptions of these datasets are available in the Appendix B [33]. Overall, they represent broad diversity in sample sizes (up to approximately 10.000), number of variables (up to 64), and number of classes (up to 10). For preprocessing purposes, we implemented two key steps to prepare the datasets. First, we addressed class imbalance by downsampling the majority class to match the size of the minority class. Second, we normalized all features across the datasets to a $[0, 1]$ range, enabling consistent scale and comparability among features. Thirdly, to ensure robust method evaluation, we employed stratified 5-fold cross-validation on each dataset. Finally, for clarity, the main manuscript reports average values, while the appendix [33] includes standard deviation for detailed analysis.

[1] https://github.com/ofurman/counterfactuals

Table 1: Comparative Results of Probabilistically Plausible Counterfactual Explanation Methods. We contrast the performance of PPCEF method with Artelt & Hammer [3] and other methods across Logistic Regression (LR) classifier. The results demonstrate our method's consistently valid and probabilistically plausible results and its ability to produce counterfactuals even in complex scenarios like high-dimensional data.

Dataset	Method	Coverage ↑	Validity ↑	Prob. Plaus. ↑	LOF	IsoForest	Log Dens. ↑	L1 ↓	L2 ↓	Time ↓
Moons	CBCE	**1.00**	**1.00**	0.10	1.06	0.03	-5.81	0.62	0.48	**0.07** s
	CEGP	**1.00**	**1.00**	0.09	1.36	0.00	-6.66	0.36	**0.28**	904.11 s
	CEM	**1.00**	**1.00**	0.14	2.03	-0.07	-10.09	0.55	0.50	211.56 s
	WACH	0.98	**1.00**	0.11	1.55	-0.01	-6.34	0.49	0.36	198.29 s
	ARTELT	**1.00**	**1.00**	0.08	1.53	-0.03	-8.74	**0.32**	0.32	4.15 s
	PPCEF	**1.00**	**1.00**	**1.00**	1.01	0.04	**1.69**	0.45	0.36	1.85 s
Law	CBCE	**1.00**	**1.00**	0.49	1.05	0.04	1.28	0.61	0.40	**0.23** s
	CEGP	**1.00**	**1.00**	0.49	1.07	0.04	1.08	0.23	0.18	1973.76 s
	CEM	**1.00**	**1.00**	0.26	1.26	-0.02	-0.56	0.33	0.31	368.10 s
	WACH	**1.00**	**1.00**	0.39	1.30	-0.01	-0.29	0.45	0.35	359.00 s
	ARTELT	**1.00**	**1.00**	0.40	1.12	0.02	0.54	**0.20**	**0.20**	4.02 s
	PPCEF	**1.00**	**1.00**	**1.00**	1.03	0.07	**2.05**	0.37	0.23	2.42 s
Audit	CBCE	**1.00**	**1.00**	0.79	11.70	0.14	**54.97**	2.55	1.24	**0.04** s
	CEGP	0.97	**1.00**	0.02	$6.08{\cdot}10^{7}$	0.02	8.09	1.56	0.57	561.04 s
	CEM	0.52	**1.00**	0.00	$8.28{\cdot}10^{6}$	-0.04	20.84	1.20	**0.37**	105.92 s
	WACH	**0.99**	**1.00**	0.02	$1.42{\cdot}10^{8}$	0.06	-40.34	1.78	0.80	101.27 s
	ARTELT	0.60	0.97	0.00	$4.09{\cdot}10^{8}$	0.10	-3585.76	**0.90**	0.88	43.84 s
	PPCEF	**1.00**	**0.99**	**0.99**	$4.25{\cdot}10^{7}$	0.08	51.64	2.04	0.79	7.01 s
Heloc	CBCE	**1.00**	**1.00**	0.54	1.10	0.07	28.01	2.84	0.82	**5.71** s
	CEGP	**1.00**	**1.00**	0.29	$3.50{\cdot}10^{7}$	0.04	24.75	**0.26**	**0.10**	9654.60 s
	CEM	**1.00**	**1.00**	0.07	$2.50{\cdot}10^{8}$	0.02	12.37	0.35	0.20	1639.16 s
	WACH	**1.00**	**1.00**	0.00	$2.65{\cdot}10^{8}$	0.03	-15.09	0.74	0.37	1600.28 s
	ARTELT	0.00	-	-	-	-	-	-	-	- s
	PPCEF	**1.00**	**1.00**	**1.00**	$6.47{\cdot}10^{7}$	0.07	**32.42**	0.90	0.23	12.44 s
Blobs	CBCE	**1.00**	**1.00**	0.27	1.02	0.03	-35.52	0.95	0.72	**0.13** s
	CEGP	**1.00**	**1.00**	0.00	2.43	-0.07	-9.08	**0.30**	**0.25**	1295.36 s
	CEM	0.96	**1.00**	0.00	3.51	-0.12	-14.95	0.46	0.45	512.56 s
	WACH	**1.00**	**1.00**	0.04	2.24	-0.06	-9.52	0.51	0.38	441.59 s
	ARTELT	**1.00**	**1.00**	0.00	2.11	-0.07	-3.51	0.39	0.33	6.62 s
	PPCEF	**1.00**	**1.00**	**1.00**	1.01	0.04	**3.00**	0.69	0.50	3.22 s
Digits	CBCE	**1.00**	**1.00**	0.18	1.02	0.04	23.72	16.28	3.09	**0.51** s
	CEGP	**1.00**	**1.00**	0.11	1.09	0.01	-0.39	2.53	**0.63**	1945.67 s
	CEM	**1.00**	0.98	0.01	1.23	-0.03	-86.77	5.28	1.38	852.05 s
	WACH	**1.00**	**1.00**	0.08	1.20	0.00	-34.97	**2.47**	1.20	651.00 s
	ARTELT	0.80	0.93	0.04	1.69	0.01	-54.72	3.30	2.43	238.28 s
	PPCEF	**1.00**	**1.00**	**1.00**	1.12	0.03	**44.42**	8.27	1.33	8.68 s
Wine	CBCE	**1.00**	**1.00**	0.37	1.06	0.05	2.13	3.38	1.12	**0.01** s
	CEGP	**1.00**	**1.00**	0.01	1.08	0.05	-0.15	0.82	0.32	191.09 s
	CEM	**1.00**	**1.00**	0.00	1.35	-0.02	-12.94	1.20	0.63	81.33 s
	WACH	**1.00**	**1.00**	0.01	1.27	0.00	-9.41	1.57	0.78	50.74 s
	ARTELT	**1.00**	0.97	0.01	1.33	0.02	-11.73	0.68	0.65	0.96 s
	PPCEF	**1.00**	**1.00**	**1.00**	1.01	0.09	**9.72**	1.65	0.53	2.03 s

Classification Models For the experiments, we include Logistic Regression (LR), 3-layer Multilayer Perceptron (MLP), and Neural Oblivious Decision Ensemble (NODE) catering to both linear and non-linear scenarios. LR aligns with linear assumptions prevalent in some baseline methods, MLP allows for the assessment of behaviors in non-linear model contexts, and NODE stands as an example of a complex ensemble of neural decision trees. This triple-model approach facilitates a thorough evaluation across varied model complexities. Crucially, all models are differentiable, which is essential in the context of our method.

Experiments Details For every combination of the classification model and dataset, we trained both the classification model and a Normalizing Flow as the density estimator, following the approach detailed in Section 4.2. We opted for the Masked Autoregressive Flow (MAF) architecture [19] as our choice for the Normalizing Flow. This decision was based on experimental findings indicating MAF's superior performance in accurately fitting data distributions. For a deeper analysis of these results, including in-depth model performance metrics like accuracy, please refer to the Appendix [33]. See Section C for a detailed exploration and Tab. 9 for specific performance figures. The final step involved generating counterfactual explanations for the entire set of test samples.

Reference Methods Our analysis includes several significant baselines, each selected for its relevance to the field. We first consider the method developed by Artelt and Hammer [3], notable for its focus on probabilistically plausible counterfactuals. Additionally, we evaluate the approach by Wachter et al. [31], widely recognized as a foundational baseline in counterfactual explanations research. To provide both endogenous and exogenous counterfactual explanations, we compare three methods: Case-Based Counterfactual Explainer (CBCE) [14], Contrastive Explanation Method (CEM) [5], and Counterfactual Explanations Guided by Prototypes (CEGP) [15].

Metrics Following related works, we chose a comprehensive set of metrics to assess the performance of counterfactual explanation methods. We include two success metrics: *coverage*, evaluating the method's ability to generate explanations across all instances, and *validity*, assessing the efficacy of counterfactuals in altering the model's decision. In terms of proximity, we measure the *L1* and *L2* distances to quantify the closeness between original instances and their counterfactuals. We evaluate plausibility using a combination of metrics. First of all, we measure the *Local Outlier Factor (LOF)* score, which, when significantly greater than 1, indicates an outlier, with values closer to 1 suggesting normalcy, highlighting anomalies through local density deviations. Secondly, we utilize *Isolation Forest*, which

assigns scores between -0.5 and 0.5, with values approaching -0.5 identifying anomalies due to the ease of isolation and scores above 0 indicating normal observations. We further access counterfactuals using *probabilistic plausibility* metric, the proportion of CFs meeting the criterion defined in Eq. 3c. Moreover, we calculate *log density*, which gauges the logarithmic probability density of counterfactuals under the target class, with higher values indicating greater plausibility. Finally, we calculate *time* metric representing time in seconds needed for the method to process the whole test dataset.

5.1 Method Intuition via Toy Example

In our illustrative example, we present the counterfactual generation process using the Moons dataset under a Logistic Regression model, as depicted in Figure 1. The initial observation is represented by a black dot, with intermediary observations during the optimization process (after every 150 iteration steps) shown as orange dots and the final counterfactual outcome marked by a red dot. The probability distributions are indicated by contour lines, with the filled red contour denoting the region exceeding the desired density threshold. The blue line illustrates the decision boundary of the classifier. This visualization effectively demonstrates how our method navigates toward the target classification and probabilistic plausibility regions, adjusting its trajectory to surpass the classifier's decision boundary by a predefined margin ϵ upon achieving the required density level.

5.2 Probabilistically Plausibile Counterfactual Explanations Methods Comparison

In this section, we conduct a focused comparison of our approach, PPCEF, against the method by Artelt and Hammer [3], which is the primary reference in the realm of probabilistically plausible counterfactual explanations. For that purpose, we utilize the datasets, metrics, and classifiers described in the previous section. The evaluation is centered on assessing and validating the accuracy of both methods in generating counterfactuals, their plausibility, and their proximity to original instances.

The results are presented in Tab. 1. Firstly, we can observe that our method always returns the results that are probabilistically plausible. That is not the case for Artelt's method, which struggles in high-dimensional datasets like Heloc (23 dimensions) or Digits (64 dimensions), doesn't support non-linear classifiers like MLPs, and wasn't able to consistently fulfill the probabilistic plausibility criterion. Secondly, in terms of distances, Artelt's method returns better results, which is expected due to the trade-off between distance and plausibility, i.e., the more plausible observations, the farther away they usually are. However, the results are not clearly worse, especially in terms of L2 distance, meaning PPCEF can balance both desired properties of counterfactuals. Thirdly, the log density values of the observations produced by PPCEF method are significantly better. Fourthly, our analysis using Local Outlier Factor (LOF) and Isolation Forest (IsoForest) metrics indicates that our methods generate inliers (except for Audit and Heloc, where almost all methods struggle to obtain reasonable values of LOF), whereas Artelt's method underperforms and can sometimes result in outliers. Fifthly, our method turned out to be significantly faster, with the speed up around x2-10 on relatively small datasets. Finally, our method was almost always able to produce valid counterfactual explanations for MLP and NODE, contrary to Artelt (see results in Tab. 2 and detailed results in Tab. 6 and 7 in Appendix [33]). It's worth mentioning that PPCEF almost always returned probabilistically plausible observations, which, in case of non-valid observations, might still be valuable insight for the final user, contrary to the lack of a response at all.

5.3 Counterfactual Explanations Methods Comparison

In this comparative analysis, we evaluate our method against well-established reference methods, with a particular focus on the impact of integrating probabilistically plausible conditions into the optimization process. Our primary objective is to assess our method's performance in terms of validity, plausibility, proximity metrics, and processing efficiency. We also explore whether methods not specifically designed for plausibility can still produce plausible counterfactuals across various classifiers such as Logistic Regression (LR), Multilayer Perceptron (MLP), and Neural Oblivious Decision Ensembles (NODE).

Results presented in Tab. 1 and Tab. 2 indicate that our model excels in validity, plausibility (considering both probabilistic formulation and outlier metrics), and processing times while maintaining reasonable distances compared to competing approaches across all datasets and classification methods. Specifically, Tab. 2 presents the evaluation results for two selected high-dimensional datasets (one for binary classification problem and one for multiclass problem) using two advanced classifiers, demonstrating that our method consistently produces valid results not only with a shallow model, such as LR but also with deeper models, including MLP and NODE. In contrast, the majority of existing methods encounter difficulties in producing valid counterfactual explanations for the Multilayer Perceptron. We conducted a comprehensive evaluation utilizing all methods and datasets mentioned earlier, applying three different classifiers. Detailed outcomes are presented in Appendix A [33]. Particularly, results for Logistic Regression are shown in Tab. 5, while findings for the MLP and NODE classifiers are detailed in Tab. 6 and 7, respectively.

Furthermore, our hypothesis that reference methods could inadvertently yield plausible outcomes without targeted optimization was not confirmed. In terms of proximity, CEGP achieves the most favorable outcomes, with our method typically ranking closely behind. This demonstrates our method's effectiveness in balancing proximity and plausibility constraints. Notably, our method's computational time efficiency closely parallels the CBCE method, which does not involve an optimization process. This efficiency is due to our batching strategy, which processes all datasets collectively, as opposed to the case-by-case optimization typical of other methods. Summarizing, our method generates probabilistically plausible counterfactuals with exceptional efficiency and minimal compromise on proximity. Its ability to process high-dimensional data quickly makes it ideal for resource-constrained, real-world applications.

6 Method Analysis

In this section, we delve into the analysis of two pivotal components of our proposed method: the loss function and the regularization hyperparameter λ. Adhering to the experimental framework established in the earlier sections, these studies are conducted specifically using the Logistic Regression model. Our focus is on evaluating the impact of these elements on the method's overall performance and efficacy.

6.1 Loss Function Ablation Study

In this ablation study, we examined the influence of discriminative loss function selection on the effectiveness of our proposed method. While Binary Cross Entropy (BCE) and Cross Entropy (CE) losses are conventional choices for binary and multiclass problems, respectively,

Table 2: Analysis of Counterfactual Methods Across Classification Models. We offer a detailed comparison of our method and other well-established reference methods across two classification models: a 3-layer Multilayer Perceptron (MLP), and a Neural Oblivious Decision Ensemble (NODE). The results emphasize the efficacy of our method in producing valid and plausible counterfactuals across various models, including those that are deeper and more complex.

Dataset	Method	Cov. ↑	Val. ↑	Prob. Plaus. ↑	LOF	IsoForest	Log Dens. ↑	L1 ↓	L2 ↓	Time ↓
					MLP					
Heloc	CBCE	**1.00**	**0.94**	0.54	1.09	0.08	28.85	2.87	0.82	**6.47** s
	CEGP	0.94	0.63	0.05	$4.15 \cdot 10^8$	0.01	-3.28	1.25	0.43	31309.33 s
	CEM	**1.00**	0.86	0.01	$7.71 \cdot 10^8$	-0.01	-89.39	1.32	0.58	6938.45 s
	WACH	**0.99**	0.81	0.00	$1.34 \cdot 10^8$	-0.06	-161.68	3.11	0.90	23392.40 s
	ARTELT	-	-	-	-	-	-	-	-	- s
	PPCEF	**1.00**	0.92	**1.00**	$1.42 \cdot 10^8$	0.07	**32.07**	**1.18**	**0.31**	25.32 s
Digits	CBCE	**1.00**	**1.00**	0.18	1.02	0.04	23.66	16.29	3.09	**0.54** s
	CEGP	0.95	0.46	0.02	1.24	-0.02	-138.62	6.39	**1.42**	2523.28 s
	CEM	**1.00**	0.42	0.01	1.44	-0.06	-481.57	**6.34**	1.76	1260.54 s
	WACH	**1.00**	0.72	0.00	1.50	-0.07	-516.44	11.04	2.13	3342.38 s
	ARTELT	-	-	-	-	-	-	-	-	- s
	PPCEF	**1.00**	**1.00**	**0.98**	1.13	0.03	**43.87**	8.78	**1.42**	25.09 s
					NODE					
Heloc	CBCE	**1.00**	**1.00**	0.55	1.09	0.08	28.88	2.85	0.82	**17.53** s
	CEM	0.94	**1.00**	0.10	1.35	0.05	9.00	**0.47**	0.29	14772.66 s
	WACH	0.96	**1.00**	0.10	$2.12 \cdot 10^8$	0.05	10.75	0.85	0.36	37254.33 s
	ARTELT	-	-	-	-	-	-	-	-	- s
	PPCEF	**1.00**	0.94	**1.00**	1.08	0.09	**31.85**	1.02	**0.28**	126.05 s
Digits	CBCE	**1.00**	**1.00**	0.18	1.02	0.04	24.00	16.27	3.09	**3.12** s
	CEM	**1.00**	**1.00**	0.03	1.32	-0.02	-39.458	4.07	1.44	5451.835 s
	WACH	**1.00**	**1.00**	0.16	1.12	0.02	7.02	**2.93**	**1.13**	15376.44 s
	ARTELT	-	-	-	-	-	-	-	-	- s
	PPCEF	**1.00**	**1.00**	**1.00**	1.15	0.02	**43.97**	7.76	1.36	69.45 s

Table 3: Ablation Study on Loss Function Selection.

Dataset	Loss	Cov.	Val.	PP	L1	L2	LD
Moons	Ours	**1.00**	**1.00**	**1.00**	**0.45**	**0.36**	1.69
	BCE	**1.00**	**1.00**	**0.99**	0.89	0.69	**1.74**
Law	Ours	**1.00**	**1.00**	**1.00**	**0.37**	**0.23**	**2.05**
	BCE	**1.00**	**1.00**	**0.98**	0.97	0.60	1.67
Audit	Ours	**1.00**	**0.99**	**0.99**	**2.04**	**0.79**	51.64
	BCE	**1.00**	**0.99**	**0.98**	3.01	1.25	**52.54**
Heloc	Ours	**1.00**	**0.99**	**0.99**	**0.85**	**0.23**	**37.50**
	BCE	**1.00**	**0.97**	**0.99**	1.91	0.54	34.50
Blobs	Ours	**1.00**	**1.00**	**1.00**	**0.69**	**0.50**	**3.00**
	CE	**1.00**	**1.00**	0.93	0.82	0.60	2.85
Digits	Ours	**1.00**	**1.00**	**1.00**	**8.27**	**1.33**	**44.42**
	CE	**1.00**	**1.00**	**1.00**	12.67	2.13	44.18
Wine	Ours	**1.00**	**1.00**	**1.00**	**1.65**	**0.53**	**9.72**
	CE	**1.00**	**1.00**	**0.99**	3.87	1.29	9.29

we compared them against our proposed discriminative loss function to understand their impacts on the results. The findings, detailed in Tab. 3, reveal a notable distinction in distance metrics. Our method, using the specialized loss function, demonstrated significantly better proximity to original observations compared to BCE and CE. This improvement is attributed to our loss function's design, which zeroes the classification component of the loss upon surpassing by ϵ a classification threshold. This allows for more rapid convergence to closer counterfactuals, while CE, by continually seeking points with higher classification confidence, tends to push counterfactuals further from the original samples. Consequently, this affects the final values in proximity metrics, underscoring the advantage of our approach in generating more proximate and plausible counterfactuals.

6.2 Regularization Hyperparameter λ Analysis

To evaluate the impact of the regularization hyperparameter λ on the fulfillment of validity and probabilistic plausibility conditions, we conducted a focused hyperparameter sensitivity analysis. While λ theoretically should extend to infinity, practical considerations necessitate setting a feasible value. Our objective is to identify an optimal λ that not only guarantees condition fulfillment but also to understand its influence on other metrics. Experiments were carried out on the Moons and Law datasets, exploring λ values within the set $\{1, 2, 5, 10, 100, 1000\}$. The results in Tab. 4 indicate that moderate values of λ, like 5 or 10, deliver satisfactory outcomes, while values around 100 or more almost invariably guarantee the fulfillment of the conditions, leading us to adopt the value of 100 for all preceding experiments. This experiment confirms the expected trade-off: higher strictness in counterfactual conditions leads to decreased proximity metrics, requiring larger deviations from the original data point.

Table 4: Ablation Study on Regularization Hyperparameter λ.

Dataset	λ	Cov.	Val.	PP	L1	L2	LD
Moons	1	1.00	0.46	0.78	0.43	0.34	1.61
	2	1.00	0.95	0.92	0.43	0.34	1.63
	5	1.00	0.99	0.98	0.43	0.34	1.66
	10	1.00	0.99	1.00	0.44	0.35	1.70
	100	1.00	1.00	1.00	0.45	0.36	1.70
	1000	1.00	1.00	1.00	0.45	0.36	1.70
Law	1	1.00	0.48	0.98	0.19	0.12	1.85
	2	1.00	0.99	0.99	0.28	0.18	1.88
	5	1.00	1.00	1.00	0.29	0.18	1.94
	10	1.00	1.00	1.00	0.30	0.18	2.00
	100	1.00	1.00	1.00	0.34	0.21	2.08
	1000	1.00	1.00	1.00	0.38	0.22	2.09

7 Conclusions

In this work, we present PPCEF, a novel method for generating counterfactual explanations that utilize normalizing flows as density estimators within an unconstrained optimization framework. This technique adeptly balances essential factors such as distance, validity, and probabilistic plausibility in the counterfactuals it produces. Notably, PPCEF is computationally efficient and capable of handling large datasets, making it highly applicable in real-world scenarios. The method's flexible design allows for future enhancements, including other desirable counterfactual attributes like actionability or sparsity, and to generate plausible counterfactuals in label-scarce environments.

Acknowledgements

Patryk Wielopolski, Oleksii Furman, and Maciej Zieba's work was supported by the National Science Centre (Poland) Grant No. 2021/43/B/ST6/02853, and Jerzy Stefanowski's work was supported by the National Science Centre (Poland) grant No. 2023/51/B/ST6/00545. Moreover, we gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016636.

References

[1] A. B. Arrieta and J. D. Ser. Plausible counterfactuals: Auditing deep learning classifiers with realistic adversarial examples. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–7. IEEE, 2020.

[2] A. Artelt and B. Hammer. On the computation of counterfactual explanations - A survey. *CoRR*, abs/1911.07749, 2019.

[3] A. Artelt and B. Hammer. Convex density constraints for computing plausible counterfactual explanations. In *Artificial Neural Networks and Machine Learning - ICANN 2020 - 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15-18, 2020, Proceedings, Part I*, volume 12396 of *Lecture Notes in Computer Science*, pages 353–365. Springer, 2020.

[4] M. Augustin, V. Boreiko, F. Croce, and M. Hein. Diffusion visual counterfactual explanations. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[5] A. Dhurandhar, P. Chen, R. Luss, 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 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 590–601, 2018.

[6] L. Dinh, D. Krueger, and Y. Bengio. NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[7] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[8] A. Dombrowski, J. E. Gerken, K. Müller, and P. Kessel. Diffeomorphic counterfactuals with generative models. *CoRR*, abs/2206.05075, 2022. doi: 10.48550/ARXIV.2206.05075.

[9] T. D. Duong, Q. Li, and G. Xu. Ceflow: A robust and efficient counterfactual explanation framework for tabular data using normalizing flows. In *Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25-28, 2023, Proceedings, Part II*, volume 13936 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2023.

[10] R. Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 04 2022.

[11] G. Jeanneret, L. Simon, and F. Jurie. Diffusion models for counterfactual explanations. In *Computer Vision - ACCV 2022 - 16th Asian Conference on Computer Vision, Macao, China, December 4-8, 2022, Proceedings, Part VII*, volume 13847 of *Lecture Notes in Computer Science*, pages 219–237. Springer, 2022.

[12] J. Jiang, J. Lan, F. Leofante, A. Rago, and F. Toni. Provably robust and plausible counterfactual explanations for neural networks via robust optimisation. *CoRR*, abs/2309.12545, 2023.

[13] K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura. DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2855–2862. ijcai.org, 2020.

[14] M. T. Keane and B. Smyth. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *Case-Based Reasoning Research and Development - 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8-12, 2020, Proceedings*, volume 12311 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 2020.

[15] A. V. Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*, volume 12976 of *Lecture Notes in Computer Science*, pages 650–665. Springer, 2021.

[16] D. Mahajan, C. Tan, and A. Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *CoRR*, abs/1912.03277, 2019.

[17] S. Mertes, T. Huber, K. Weitz, A. Heimerl, and E. André. Ganterfactual—counterfactual explanations for medical non-experts using generative adversarial learning. *Frontiers in Artificial Intelligence*, 2022.

[18] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pages 607–617. ACM, 2020.

[19] G. Papamakarios, I. Murray, and T. Pavlakou. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2338–2347, 2017.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[21] J. Pearl, M. Glymour, and N. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. ISBN 9781119186847.

[22] S. Popov, S. Morozov, and A. Babenko. Neural oblivious decision ensembles for deep learning on tabular data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[23] R. Poyiadzi, K. Sokol, R. Santos-Rodríguez, T. D. Bie, and P. A. Flach. FACE: feasible and actionable counterfactual explanations. In *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, pages 344–350. ACM, 2020.

[24] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[25] C. Russell. Efficient search for diverse coherent explanations. In danah boyd and J. H. Morgenstern, editors, *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 20–28. ACM, 2019.

[26] G. Shakhnarovich, T. Darrell, and P. Indyk. Nearest-neighbor methods in learning and vision. *IEEE Trans. Neural Networks*, 19(2):377, 2008.

[27] B. Smyth and M. T. Keane. A few good counterfactuals: Generating interpretable, plausible and diverse counterfactual explanations. In *Case-Based Reasoning Research and Development - 30th International Conference, ICCBR 2022, Nancy, France, September 12-15, 2022, Proceedings*, volume 13405 of *Lecture Notes in Computer Science*, pages 18–32. Springer, 2022.

[28] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[29] N. Vercheval and A. Pizurica. Hierarchical variational autoencoders for visual counterfactuals. In *2021 IEEE International Conference on Image Processing, ICIP 2021, Anchorage, AK, USA, September 19-22, 2021*, pages 2513–2517. IEEE, 2021.

[30] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[31] S. Wachter, B. D. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017.

[32] G. P. Wellawatte, A. Seshadri, and A. D. White. Model agnostic generation of counterfactual explanations for molecules. *Chem. Sci.*, 2022.

[33] P. Wielopolski, O. Furman, J. Stefanowski, and M. Zieba. Probabilistically plausible counterfactual explanations with normalizing flows. *CoRR*, abs/2405.17640, 2024. doi: 10.48550/ARXIV.2405.17640.

[34] L. F. Wightman. Lsac national longitudinal bar passage study. lsac research report series. Technical report, Law School Admission Council, Newtown, PA., 1998.