Diffusion-Guided Counterfactual Generation for Model Explainability

Nishtha Madaan*
Indian Institute of Technology
New Delhi, India
nishthaa.madaan@gmail.com

Srikanta Bedathur
Indian Institute of Technology
New Delhi, India
srikanta@cse.iitd.ac.in

Abstract

Generating counterfactual explanations is one of the most effective approaches for uncovering the inner workings of black-box neural network models and building user trust. While remarkable strides have been made in generative modeling using diffusion models in domains like vision, their utility in generating counterfactual explanations in structured modalities remains unexplored. In this paper, we introduce *Structured Counterfactual Diffuser* or SCD, the first plug-and-play framework leveraging diffusion for generating counterfactual explanations in structured data. SCD learns the underlying data distribution via a diffusion model which is then guided at test time to generate counterfactuals for any arbitrary black-box model, input, and desired prediction. Our experiments show that our counterfactuals not only exhibit high plausibility compared to the existing state-of-the-art but also show significantly better proximity and diversity.

1 Introduction

As AI models become more capable and widespread, the issue of trust becomes critical (Doshi-Velez & Kim, 2017). While traditional software is transparent—allowing tracing its control flow and easily resolving trust concerns—modern AI is built upon neural networks that are not transparent. Their underlying control flow is not understood, making it difficult to trust in high-risk settings such as loan or hiring decisions. Although the remarkable power and flexibility of neural networks have allowed building systems that achieve capabilities not possible with traditional software alone OpenAI (2023); Ramesh et al. (2022), this lack of transparency and trust becomes a significant hurdle in realizing the full potential of neural networks Ribeiro et al. (2016); Lundberg & Lee (2017); Wachter et al. (2017).

To address concerns about trust, one needs to answer *why* a model behaves in a certain way. One of the most promising directions to answer this is via *what-if* scenarios or counterfactuals Wachter et al. (2017). While Wachter et al. (2017) originally introduced the idea of counterfactual explanations, the idea has gained significant attention in recent years Mothilal et al. (2020); Karimi et al. (2019); Yang et al. (2022); Ross et al. (2020); Madaan et al. (2021). Ideally, counterfactuals should possess the following characteristics: *1)* they should maintain *proximity* to the original input, *2)* they should attain the desired counterfactual label to ensure its *validity*, *3)* they should be *diverse* and capture a wide range of distinct scenarios and *4)* they should be *plausible*. While proximity, validity, and diversity criteria have been studied extensively, there has been little focus on the plausibility of the generated counterfactuals, i.e., ensuring that the generated counterfactuals are realistic and conform

*Nishtha Madaan is a researcher at IBM Research. This work was done as part of PhD at IIT Delhi.

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

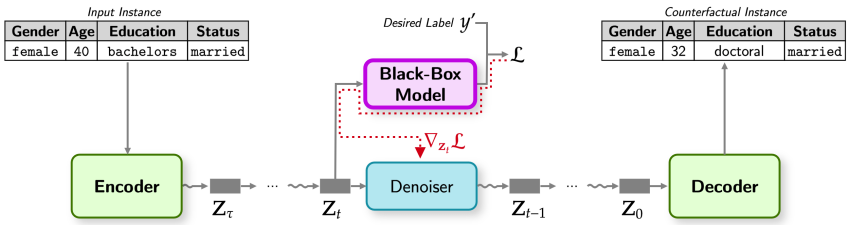

Figure 1: **Overview of Our Counterfactual Generation Process.** The process starts by encoding the given human-readable instance or row into an embedding by performing a look-up on a dictionary of learned embeddings. Next, we iteratively apply denoising steps while incorporating the gradient information from the given black-box model to minimize the disparity between the model's prediction and the desired label. At the end of the denoising process, we obtain an embedding which is then decoded via a reverse look-up on the dictionary to obtain the counterfactual instance.

to the underlying data distribution. Previous works have approached plausibility in a minimal sense, e.g., enforcing values to lie in legal ranges or applying user-designed constraints Mothilal et al. (2020); Karimi et al. (2019).

Recently, in the visual domain, diffusion models (Ho et al., 2020) have been successfully used to acquire the underlying data distribution for generating plausible counterfactual explanations Augustin et al. (2022); Jeanneret et al. (2022); Sanchez & Tsaftaris (2022); 202 (2023). However, in the domain of tabular or structured data, counterfactual explanation methods have largely ignored these recent advances in diffusion modeling raising another important question: *"Can diffusion models, which are known for their remarkable generation capabilities in vision, help generate high-quality plausible counterfactuals in the structured domain?"*

To answer this question, in this work, we propose a novel counterfactual explainer called *Structured Counterfactual Diffuser* or *SCD*. SCD is the first plug-and-play framework leveraging diffusion modeling for generating counterfactual explanations for structured data. SCD works by learning the underlying data distribution via a diffusion model Li et al. (2022); Ho et al. (2020). At test time, the diffusion model is used to perform guided iterative denoising to generate counterfactuals for any given input and black-box model in a plug-and-play manner. In experiments, we show that our counterfactual explainer not only exhibits high plausibility compared to the state-of-the-art approaches but also shows significantly better proximity and diversity scores of the generated counterfactuals.

2 Preliminaries

Structured Data. A table or structured data consists of rows or instances. Each instance is a tuple with a value for each column or attribute. The entire space of such instances can be described as $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_C$. Here, C denotes the number of columns or attributes in the table, and each $\mathcal{X}_c$ denotes the space of possible values for column c. For example, a possible instance from a 4-column table is `[female, 40, doctoral, married]`. Here, $\mathcal{X}_1$ can represent gender categories, $\mathcal{X}_2$ can represent the possible age values, and so forth. We will use $\mathbf{x}$ to denote an instance and $\mathbf{x}^c$ to denote c-th column or attribute within the instance.

Black-Box Model. A black-box model is a model $f : \mathcal{X} \to \mathcal{Y}$ that maps an input instance $\mathbf{x} \in \mathcal{X}$ to a label $y \in \mathcal{Y}$.

2.1 Structured Counterfactual Explanations

As highlighted by Wachter et al. (2017), counterfactuals help identify alternative scenarios where a slight change in the original input $\mathbf{x}$ to a counterfactual input $\mathbf{x}'$ would have changed

the outcome from y to y' by a black-box model f. By analyzing the change in prediction on counterfactual inputs, one can uncover if the model is making decisions based on potentially problematic or undesired criteria.

Counterfactual Explainer. Formally, a counterfactual explainer can be described as a system or framework that, given an input $\mathbf{x}$, a model f, and a counterfactual label y' (where y' is different from the original label y), produces a set of B counterfactuals $\mathbf{X}'$.

$$\mathbf{X}' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_B\} = \text{CounterfactualExplainer}(f, \mathbf{x}, y').$$

Here, each counterfactual $\mathbf{x}'_b \in \mathbf{X}'$ should achieve the counterfactual label y' on the given black-box model f with minimal change to the original input $\mathbf{x}$.

Desired Characteristics of Counterfactuals. While Wachter et al. (2017) originally introduced the proximity desiderata i.e., the counterfactual should be close to the original input $\mathbf{x}$, Mothilal et al. (2020) introduced the desiderata of diversity. Plausibility, on the other hand, has not been given much attention in the community. Some existing works primarily focus on only keeping generated values within legal ranges, disregarding the complex relationships that values of various columns have (Karimi et al., 2019) or require costly user-defined plausibility constraints (Mothilal et al., 2020). In this work, we take a significant step forward in alleviating this concern.

3 SCD: Structured Counterfactual Diffuser

In this section, we present our proposed model *Structured Counterfactual Diffuser* or *SCD*. SCD learns a diffusion model through training on a structured dataset or table $\mathcal{D}$. Via training on $\mathcal{D}$, SCD learns about the underlying data distribution which enables it to generate plausible counterfactuals. Once the diffusion model is trained, SCD can be used in a plug-and-play manner to obtain counterfactual explanations for any given black-box model. We now describe SCD in detail.

Row Embedding. To train the diffusion model, we first map the raw human-readable instances or rows $\mathbf{x}$ of the table $\mathcal{D}$ into embeddings. The diffusion model shall be trained to model the distribution in this embedding space. We maintain a learned dictionary of embeddings $\text{Embedding}_c : \mathcal{X}_c \to \mathbb{R}^d$ for each column c. To encode a row, we lookup the embedding for each of the C columns and concatenate these embeddings to obtain a row embedding $\mathbf{z}$ as follows:

$$\mathbf{z} = [\text{Embedding}_1(\mathbf{x}^1), \ldots, \text{Embedding}_C(\mathbf{x}^C)] \in \mathbb{R}^{C \times d}$$

where d is the size of the embedding per column.

3.1 Diffusion Modeling

Via diffusion modeling, we seek to learn a distribution $p_\theta(\mathbf{z})$ over the row embeddings. In diffusion modeling, the distribution $p_\theta(\mathbf{z})$ consists of T denoising steps:

$$p_\theta(\mathbf{z}_0) = \int p(\mathbf{z}_T) \prod_{t=T,\ldots,1} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, t) d\mathbf{z}_{1:T}$$

Here, $p(\mathbf{z}_T)$ represents standard Gaussian, the sequence $\mathbf{z}_T, \ldots, \mathbf{z}_1$ consists of iteratively cleaner samples, finally producing the desired sample $\mathbf{z}_0$; and $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, t)$ is a one-step denoising distribution. The $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, t)$ is parametrized in the following manner:

$$\mathcal{N}\left(\gamma_{1,t}\hat{\mathbf{z}}_0 + \gamma_{2,t}\mathbf{z}_t, \beta_t \mathbf{I}\right)$$

where $\hat{\mathbf{z}}_0 = g_\theta(\mathbf{z}_t, t)$, and the coefficients $\gamma_{1,t}$ and $\gamma_{2,t}$ are given by:

$$\gamma_{1,t} = \frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}, \qquad \gamma_{2,t} = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}$$

Employing standard notations, we utilize a variance schedule $\beta_1, \ldots, \beta_T$, where $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$. We use a cosine schedule in our implementation.

Learning: The training procedure involves first introducing noise to the input $\mathbf{z}_0$, creating its noisy version $\mathbf{z}_t$.

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t, \qquad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Subsequently, a neural network predictor is trained that takes $\mathbf{z}_t$ as input and aims to predict the original input $\mathbf{z}_0$ by generating a prediction $\hat{\mathbf{z}}_0 = g_\theta(\mathbf{z}_t, t)$. The learning objective is $\mathcal{L}_{\text{diffusion}}(\theta) = \mathcal{E}(\hat{\mathbf{z}}_0, \mathbf{z}_0)$ where $\mathcal{E}$ is an error function.

3.2 Generating Counterfactuals via Guided Diffusion

Given the trained denoising distribution $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, t)$, we are now ready to generate counterfactuals for a black-box model f given an input instance $\mathbf{x}$ and a desired label y'. The process works by performing guided diffusion starting from the embedding of the given input instance. For this, we first encode $\mathbf{x}$ to its row embedding $\mathbf{z} \in \mathbb{R}^{C \times d}$. Since we seek to sample B counterfactuals, we copy the row embedding B times and stack the copies together to construct an embedding $\mathbf{Z} \in \mathbb{R}^{B \times C \times d}$. Next, we add Gaussian noise to $\mathbf{Z}$ to facilitate diversity among the B generated samples.

$$\mathbf{Z}'_\tau \leftarrow \sqrt{\bar{\alpha}_\tau}\mathbf{Z} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t, \qquad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Next, we perform τ guided diffusion steps. We iteratively and alternatingly apply the following two steps: *1) Denoising Step:* This step involves sampling $\mathbf{Z}'_{t-1} \sim p_\theta(\mathbf{Z}'_{t-1}|\mathbf{Z}'_t, t)$. *2) Guiding Step:* This step involves performing a gradient step on $\mathbf{Z}'_{t-1}$ with respect to a guiding loss $\mathcal{L}$ as: $\mathbf{Z}'_{t-1} \leftarrow \mathbf{Z}'_{t-1} - \eta\nabla_{\mathbf{Z}'_{t-1}}\mathcal{L}$, where η is the step size for the update. One of the things that $\mathcal{L}$ measures is how well the black-box model f produces the counterfactual label y' on the samples $\mathbf{Z}'_{t-1}$ of the current step. We describe the exact formulation of $\mathcal{L}$ in detail in a later section. From this iterative process, we obtain a series of progressively cleaned embeddings $\mathbf{Z}'_\tau, \ldots, \mathbf{Z}'_0$. Next, we take the generated $\mathbf{Z}'_0$, perform reverse look-up using the learned embeddings and obtain the human-readable counterfactual instances $\mathbf{X}' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_B\}$. In Fig. 1, we illustrate this process.

3.2.1 Guiding Loss

We now describe the terms in our guiding loss $\mathcal{L}$. Following Mothilal et al. (2020), we include 3 terms in our loss capturing validity, proximity, and diversity of the samples. Formally this loss can be described as:

$$\mathcal{L}(\mathbf{Z}', \mathbf{x}, f, y') = \lambda_{\text{validity}}\mathcal{L}_{\text{validity}}(\mathbf{Z}', f, y') + \lambda_{\text{proximity}}\mathcal{L}_{\text{proximity}}(\mathbf{Z}, \mathbf{Z}') + \lambda_{\text{diversity}}\mathcal{L}_{\text{diversity}}(\mathbf{Z}')$$

Validity Loss. We use the cross-entropy loss of the black-box model f with respect to the desired prediction y' as our validity loss.

$$\mathcal{L}_{\text{validity}}(\mathbf{Z}', f, y') = \text{CrossEntropy}(f(\mathbf{Z}'), \texttt{target} = y').$$

Proximity Loss. We use a simple L2 loss between $\mathbf{Z}$ the embedding of the original input and $\mathbf{Z}'$ the generated embedding at the current step of the guided diffusion.

$$\mathcal{L}_{\text{proximity}}(\mathbf{Z}, \mathbf{Z}') = ||\mathbf{Z} - \mathbf{Z}'||^2.$$

Diversity Loss. We use the negative of L2 loss between all pairs of counterfactual instances:

$$\mathcal{L}_{\text{diversity}}(\mathbf{Z}') = \frac{-2}{B(B-1)} \sum_{i=1}^{B-1} \sum_{j=i+1}^{B} ||\mathbf{z}'_i - \mathbf{z}'_j||^2.$$

Table 1: Comparison of plausibility, proximity, diversity, and validity scores of SCD and DiCE on various datasets. For validity, proximity, and diversity scores, higher is better. For the plausibility score, lower is better since it captures the negative log-likelihood of the generated samples.

Dataset	Plausibility (↓)		Proximity (↑)		Diversity (↑)		Validity (↑)	
	DiCE	SCD	DiCE	SCD	DiCE	SCD	DiCE	SCD
Adult Income	121.0	**21.21**	0.5764	**0.6173**	0.3837	**0.4008**	**0.9776**	0.7511
UCI Bank	166.7	**42.37**	0.2141	**0.3000**	0.4165	**0.5498**	**0.9686**	0.8600
Housing Price	109.5	**42.91**	0.3055	**0.3417**	0.4289	**0.5986**	**0.9908**	0.8526

Table 2: **Counterfactual Samples in Adult Income Dataset.** Given the input row with the original label "$\leq 50K$", we ask our method SCD and the baseline DiCE to generate counterfactual instances that flip the label to "$> 50K$" with respect to a black-box income predictor. We note that SCD generates plausible samples while DiCE struggles. Specifically, we note that DiCE creates counterfactuals containing *Divorced* and *Husband* within the same row which is contradictory and impossible (highlighted in red). In comparison, SCD creates plausible counterfactuals where the Marital Status and Relationship columns correctly conform with each other (highlighted in green).

Method	Age	Workclass	Education	Ed. No.	Marital Status	Occupation	Relationship
Input	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family
Ours	31	Self-emp-inc	Bachelors	13	Married-civ-spouse	Adm-clerical	Wife
	34	Self-emp-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
	39	Federal-gov	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband
DiCE	39	State-gov	Bachelors	16	Divorced	Transport-moving	Husband
	39	State-gov	Bachelors	16	Divorced	Transport-moving	Husband
	39	Without-pay	Some-college	16	Divorced	Transport-moving	Husband

4 Experiments

Datasets. In experiments, we evaluate the quality of generated counterfactuals on three datasets: Adult Income Dataset Frank (2010), UCI Bank Dataset Moro & Cortez (2012) and Housing Price Dataset Pace & Barry (1997).

Black-Box Model. For each dataset, we train a classifier to act as the black-box model that a counterfactual explainer would seek to explain. The architecture is a simple 2-layer MLP that takes the concatenated embeddings of columns of a row as input and tries to predict a class label. For each dataset, the classification task that the black-box model is trained to perform is as follows: *1) Adult Income Dataset:* Given a row as input, the black-box model predicts whether the income exceeds 50K per year or not. *2) UCI Bank Dataset:* Given a row describing attributes of a client, the black-box model predicts if the client will subscribe to a term deposit or not. *3) Housing Price Dataset:* Given a row as input, the black-box model predicts whether the house price is greater than $200K or not.

4.1 Metrics

We consider the following metrics for evaluating the generated counterfactuals. *1) Validity Score:* We compute the validity score of the generated counterfactuals in $\mathbf{X}'$ by checking if they result in the desired label with respect to the black-box model. *2) Proximity Score:* We compute proximity score as the average fraction of matching values between the generated counterfactuals in $\mathbf{X}'$ and the original input $\mathbf{x}$. *3) Diversity Score:* We compute the diversity score of the generated counterfactuals in $\mathbf{X}'$ as the mean of the distances between each pair of samples. *4) Plausibility:* The goal is to evaluate how likely is the generated counterfactual under the true data distribution. We learn a model of the desired distribution by learning an auto-regressive model p_ϕ over the tokens or values in the instances. To compute the plausibility score, we compute the negative log-likelihood of each generated counterfactual

$\mathbf{x}'_b \in \mathbf{X}'$ using p_ϕ:

$$\text{Plausibility} = -\frac{1}{B}\sum_{b=1}^{B}\log p_\phi(\mathbf{x}'_b) = -\frac{1}{B}\sum_{b=1}^{B}\sum_{n=1}^{N}\log p_\phi(\mathbf{x}'_{b,n}|\mathbf{x}'_{b,1},\ldots,\mathbf{x}'_{b,n-1})$$

where a lower negative log-likelihood is desired for a more plausible counterfactual.

4.2 Benefits of SCD in Counterfactual Generation

In Table 1, we compare our model SCD and our baseline, DiCE. It is remarkable that our model produces counterfactuals that are significantly more plausible than those generated by DiCE. In fact, the negative log-likelihood of our samples are 21.21, 42.37, and 42.91 while DiCE yields significantly worse results attaining 121.0, 166.7, and 109.5 on the 3 datasets, respectively. Our higher plausibility is also evidenced by our generated counterfactual samples in Table 2. We can see that our model coherent values for the columns *Marital Status* and *Relationship* while the baseline DiCE produces contradictory values e.g., *Divorced* and *Husband* within the same row. This highlights the advantage of using a diffusion model that learns complex relationships to constrain the generated counterfactuals to be plausible.

Furthermore, our results show significant improvements in the diversity and proximity scores over the baseline, achieving approximately 0.10-0.17 higher diversity and 0.04-0.10 higher proximity scores relative to DiCE. Our validity score, i.e., the fraction of generated counterfactuals that attain the desired label, is about 0.1 lower than the baseline. While this is a slight decline, it is not a significant concern since it is straightforward to remove the counterfactuals that do not attain the desired label via post-processing. Furthermore, some worsening of the validity score may be expected since SCD constrains the samples to be plausible while DiCE does not.

5 Related Work

Various studies have pursued counterfactual explanations Wachter et al. (2017); Mothilal et al. (2020); Yang et al. (2022); Karimi et al. (2020); Guidotti et al. (2019). However, none of them directly and properly tackle the problem of generating plausible counterfactuals. In the image domain, several works attempt to generate counterfactuals using diffusion models Augustin et al. (2022); Jeanneret et al. (2022); Sanchez & Tsaftaris (2022); 202 (2023). This is another line of works focusing on contrastive explanations Dhurandhar et al. (2019); Jacovi et al. (2021), however, these do not leverage diffusion modeling, like ours. However, while these are based on the image domain, the utility of diffusion models for counterfactual explanation in the tabular domain has remained unexplored. In the language domain, there has been a significant number of works for counterfactual generation Wu et al. (2021); Madaan et al. (2021, 2023); Ross et al. (2020); Boreiko et al. (2022); Howard et al. (2022). However, these have primarily relied on auto-regressive LLMs and not diffusion models. Although Li et al. (2022) pursues diffusion-based language modeling, it does not pursue the task of counterfactual explanation and also does not deal with the tabular domain. Additionally, there has also been interest in the domain of search and retrieval for generating counterfactual explanations Xu et al. (2023).

6 Conclusion

In this paper, we introduced a novel counterfactual explainer called *Structured Counterfactual Diffuser* (SCD) for structured data aimed at producing highly plausible counterfactuals. Our technique leverages a diffusion model to learn complex relationships among various attributes of structured data. Via guided diffusion, our model not only exhibits high plausibility compared to the existing state-of-the-art but also shows significant improvement in proximity and diversity, while also maintaining high validity.

References

Diffusion-based visual counterfactual explanations – towards systematic quantitative evaluation. 2023. URL https://api.semanticscholar.org/CorpusID:260866076.

Augustin, M., Boreiko, V., Croce, F., and Hein, M. Diffusion visual counterfactual explanations. *Advances in Neural Information Processing Systems*, 35:364–377, 2022.

Boreiko, V., Augustin, M., Croce, F., Berens, P., and Hein, M. Sparse visual counterfactual explanations in image space. *ArXiv*, abs/2205.07972, 2022. URL https://api.semanticscholar.org/CorpusID:248834482.

Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P.-Y., Shanmugam, K., and Puri, R. Model agnostic contrastive explanations for structured data. *ArXiv*, abs/1906.00117, 2019. URL https://api.semanticscholar.org/CorpusID:173990728.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:11319376.

Frank, A. Uci machine learning repository. *http://archive. ics. uci. edu/ml*, 2010.

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., and Turini, F. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34:14–23, 2019. URL https://api.semanticscholar.org/CorpusID:210931542.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Howard, P., Singer, G., Lal, V., Choi, Y., and Swayamdipta, S. Neurocounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. *ArXiv*, abs/2210.12365, 2022. URL https://api.semanticscholar.org/CorpusID:253098636.

Jacovi, A., Swayamdipta, S., Ravfogel, S., Elazar, Y., Choi, Y., and Goldberg, Y. Contrastive explanations for model interpretability. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://api.semanticscholar.org/CorpusID:232092617.

Jeanneret, G., Simon, L., and Jurie, F. Diffusion models for counterfactual explanations. In *Asian Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusID:247779169.

Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. Model-agnostic counterfactual explanations for consequential decisions. *ArXiv*, abs/1905.11190, 2019. URL https://api.semanticscholar.org/CorpusID:166227893.

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

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343, 2022.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Madaan, N., Padhi, I., Panwar, N., and Saha, D. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 15, pp. 13516–13524, 2021.

Madaan, N., Saha, D., and Bedathur, S. Counterfactual sentence generation with plug-and-play perturbation. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 306–315. IEEE, 2023.

Moro, S., R. P. and Cortez, P. Bank Marketing. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.

Mothilal, R. K., Sharma, A., and Tan, C. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Pace, R. K. and Barry, R. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. URL https://api.semanticscholar.org/CorpusID:248097655.

Ribeiro, M. T., Singh, S., and Guestrin, C. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

Ross, A., Marasović, A., and Peters, M. E. Explaining nlp models via minimal contrastive editing (mice). *arXiv preprint arXiv:2012.13985*, 2020.

Sanchez, P. and Tsaftaris, S. A. Diffusion causal models for counterfactual estimation. In *CLEaR*, 2022. URL https://api.semanticscholar.org/CorpusID:247011291.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Wu, T., Ribeiro, M. T., Heer, J., and Weld, D. S. Polyjuice: Automated, general-purpose counterfactual generation. *arXiv preprint arXiv:2101.00288*, 2021.

Xu, Z., Lamba, H., Ai, Q., Tetreault, J., and Jaimes, A. Counterfactual editing for search result explanation. *ArXiv*, abs/2301.10389, 2023. URL https://api.semanticscholar.org/CorpusID:256231426.

Yang, W., Li, J., Xiong, C., and Hoi, S. C. Mace: An efficient model-agnostic framework for counterfactual explanation. *arXiv preprint arXiv:2205.15540*, 2022.