Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review

Sahil Verma
vsahil@cs.washington.edu
University of Washington
Seattle, WA, USA

Varich Boonsanong
varieb@cs.washington.edu
University of Washington
Seattle, WA, USA

Minh Hoang
minh257@cs.washington.edu
University of Washington
Seattle, WA, USA

Keegan E. Hines
keegan@arthur.ai
Arthur AI
Washington D.C., USA

John P. Dickerson
john@arthur.ai
Arthur AI
Washington D.C., USA

Chirag Shah
chirags@uw.edu
University of Washington
Seattle, WA, USA

ABSTRACT

Machine learning plays a role in many deployed decision systems, often in ways that are difficult or impossible to understand by human stakeholders. Explaining, in a human-understandable way, the relationship between the input and output of machine learning models is essential to the development of trustworthy machine learning based systems. A burgeoning body of research seeks to define the goals and methods of explainability in machine learning. In this paper, we seek to review and categorize research on counterfactual explanations, a specific class of explanation that provides a link between what could have happened had input to a model been changed in a particular way. Modern approaches to counterfactual explainability in machine learning draw connections to the established legal doctrine in many countries, making them appealing to fielded systems in high-impact areas such as finance and healthcare. Thus, we design a rubric with desirable properties of counterfactual explanation algorithms and comprehensively evaluate all currently proposed algorithms against that rubric. Our rubric provides easy comparison and comprehension of the advantages and disadvantages of different approaches and serves as an introduction to major research themes in this field. We also identify gaps and discuss promising research directions in the space of counterfactual explainability.

1 INTRODUCTION

Machine learning is increasingly accepted as an effective tool to enable large-scale automation in many domains. In lieu of hand-designed rules, algorithms are able to learn from data to discover patterns and support decisions. Those decisions can, and do, directly or indirectly impact humans; high-profile cases include applications in credit lending [281], talent sourcing [275], parole [295], and medical treatment [93]. The nascent Fairness, Accountability, Transparency, and Ethics (FATE) in machine learning community has emerged as a multi-disciplinary group of researchers and industry practitioners interested in developing techniques to detect bias in machine learning models, develop algorithms to counteract that bias, generate human-comprehensible explanations for the machine decisions, hold organizations responsible for unfair decisions, etc. Human-understandable explanations for machine-produced decisions are advantageous in several ways. For example, focusing on a use case of applicants applying for loans, the benefits would include:

- An explanation can be beneficial to the applicant whose life is impacted by the decision. For example, it helps an applicant understand which of their attributes were strong drivers in determining a decision.
- Various forms of explanations can serve as a proxy for transparency in the system, which could increase its trustworthiness.
- Further, it can help an applicant challenge a decision if they feel an unfair treatment has been meted out, e.g., if one’s race was crucial in determining the outcome. This can also be useful for organizations to check for bias in their algorithms.
- In some instances, an explanation provides the applicant with feedback that they can act upon to receive the desired outcome at a future time.
- Explanations can help the machine learning model developers identify, detect, and fix bugs and other performance issues.
- Explanations help adhere to laws surrounding machine-produced decisions, e.g., GDPR [62].

Explainability in machine learning is broadly about using inherently interpretable and transparent models or generating post-hoc explanations for opaque models. Examples of the former include linear/logistic regression, decision trees, rule sets, etc. Examples of the latter include random forests, support vector machines (SVMs), and neural networks. Post-hoc explanation approaches can either be model-specific or model-agnostic. Explanations by feature importance and model simplification are two broad kinds of model-specific approaches. Model-agnostic approaches can be categorized into visual explanations, local explanations, feature importance, and model simplification.

Feature importance finds the most influential features contributing to the model’s overall accuracy or for a particular decision, e.g., SHAP [205], QII [70]. Model simplification finds an interpretable model that imitates the opaque model closely. Dependency plots are a popular kind of visual explanation, e.g., Partial Dependence Plots [106], Accumulated Local Effects Plot [16], Individual Conditional Expectation [118]. They plot the change in the model’s prediction as one or multiple features are changed. Local explanations differ from other methods because they only explain a single prediction. Local explanations can be further categorized into approximation and example-based approaches. Approximation approaches sample new datapoints in the vicinity of the datapoint whose prediction from the model needs to be explained (hereafter
called the explainee datapoint, and then fit a linear model (e.g., LIME [261]) or extracts a rule set from them (e.g., Anchors [262]). Example-based approaches seek to find datapoints in the vicinity of the explainee datapoint. They either offer explanations in the form of datapoints that have the same prediction as the explainee datapoint or the datapoints whose prediction differs from the explainee datapoint. Note that the latter kind of datapoints are still close to the explainee datapoint and are termed as “counterfactual explanations” (CFE).

Recall the use case of applicants applying for a loan. For an individual whose loan request has been denied, counterfactual explanations provide them with actionable feedback that could help them make changes to their features in order to transition to the desirable side of the decision boundary, i.e., get the loan. This feedback is termed as an algorithmic recourse. Unlike several other explainability techniques, CFEs (or recourses) do not explicitly answer the “why” the model made a prediction; instead, they provide suggestions to achieve the desired outcome. CFEs are also applicable to black-box models (when only the predict function of the model is accessible), and therefore place no restrictions on model complexity and do not require model disclosure. They also do not necessarily approximate the underlying model, producing accurate feedback. Owing to their intuitive nature, CFEs are also amenable to legal frameworks (see appendix C).

In this work, we collect, review and categorize more than 350 recent papers that propose algorithms to generate counterfactual explanations for machine learning models. Many of these methods have focused on datasets that are either tabular or image-based. We describe our methodology for collecting papers for this survey in appendix B. We describe recent research themes in this field and categorize the collected papers among a fixed set of desiderata for effective counterfactual explanations (see table 1). The contributions of this review paper are:

(1) We examine a set of more than 350 recent papers on the same set of parameters to allow for an easy comparison of the techniques these papers propose and the assumptions they work under.

(2) The categorization of the papers achieved by this evaluation helps a researcher or a developer choose the most appropriate algorithm given the set of assumptions they have and the speed and quality of the generation they want to achieve.

(3) Comprehensive and lucid introduction for beginners in the area of counterfactual explanations for machine learning.

2 BACKGROUND

This section gives the background about the social implications of machine learning, explainability research in machine learning, and some prior studies about counterfactual explanations.

2.1 Social Implications of Machine Learning

Establishing fairness and making an automated tool’s decision explainable are two broad ways in which we can ensure equitable social implications of machine learning. Fairness research aims at developing algorithms that can ensure that the decisions produced by the system are not biased against a particular demographic group of individuals, which are defined with respect to sensitive or protected features, such as race, sex, and religion. Anti-discrimination laws make it illegal to use sensitive features as the basis of any decision (see Appendix C). Biased decisions can also attract widespread criticism and are therefore crucial to avoid [123, 177]. Fairness has been captured in several notions based on a demographic grouping or individual capacity. Verma and Rubin [317] have enumerated and intuitively explained many fairness definitions using a unifying dataset. Dunkelau and Leuschel [88] provide an extensive overview of the major categorization of research efforts in ensuring fair machine learning and enlists important works in all categories. Explainable machine learning has also seen interest from other communities, specifically healthcare [300], having huge social implications. Several works have summarized and reviewed other research in explainable machine learning [3, 51, 127].

2.2 Explainability in Machine Learning

This section gives some concrete examples that emphasize the importance of explainability and give further details of the research in this area. In a real-world example, the US military trained a classifier to distinguish enemy tanks from friendly tanks. Although the classifier performed well on the training and test dataset, its performance was abysmal on the battlefield. Later, it was found that the photos of friendly tanks were taken on sunny days, while for enemy tanks, photos clicked only on overcast days were available [127]. The classifier found it much easier to use the difference between the background as the distinguishing feature. In a similar case, a husky was classified as a wolf because of the presence of snow in the background, which the classifier had learned as a feature associated with wolves [261]. The use of an explainability technique helped discover these issues. The explainability problem can be divided into model explanation and outcome explanation problems [127].

Model explanation searches for an interpretable and transparent global explanation of the original model. Various papers have developed techniques to explain neural networks and tree ensembles using single decision tree [65, 83, 184] and rule sets [14, 76]. Some approaches are model-agnostic, such as Golden Eye and PALM [139, 185, 357].

Outcome explanation needs to provide an explanation for a specific prediction from the model. This explanation need not be a global explanation or explain the internal logic of the model. Model-specific approaches for deep neural networks (CAM, Grad-CAM [274, 355]), and model agnostic approaches (LIME, MES [261, 307]) have been proposed. These are either feature attribution or model simplification methods. Example-based approaches are another kind of explainability technique used to explain a particular outcome. This work focuses on counterfactual explanations (CFEs), which is an example-based approach.

By definition, CFEs are applicable to supervised machine learning setups where the desired prediction has not been obtained for a datapoint. The majority of research in this area has applied CFEs to classification settings, which consists of several labeled datapoints that are given as input to the model, and the goal is to learn a function mapping from the input datapoints (with, say, m features) to labels. In classification, the labels are discrete values. \(X^m\) is used to denote the input space of the features, and \(Y\) is used to denote the output space of the labels. The learned function is the mapping
\( f : \mathcal{X}^n \rightarrow \mathcal{Y} \), which is used to predict labels for unseen datapoints in the future.

### 2.3 History of Counterfactual Explanations

Counterfactual explanations have a long history in other fields like philosophy, psychology, and the social sciences. Philosophers like David Lewis published articles on the ideas of counterfactuals back in 1973 \cite{Lewis1973}. Woodward \cite{Woodward2003} said that a satisfactory explanation must follow patterns of counterfactual dependence. Psychologists have demonstrated that counterfactuals elicit causal reasoning in humans \cite{Asher2012, Fischhoff1977, Lorch1970}. Philosophers have also validated the concept of causal thinking due to counterfactuals \cite{Atkins2009, Woodward2003}.

Studies have compared the likeability of CFEs with other explanation approaches. Binns et al. \cite{Binns2017} and Dodge et al. \cite{Dodge2019} performed user studies that showed that users prefer CFEs over case-based reasoning, which is another example-based approach. The work by Fernández-Loría et al. \cite{Fernandez-Lorija2017} provides three interesting examples where the feature importance explanation methods fail to capture the underlying model, whereas CFEs do. Asher et al. \cite{Asher2012} argue that the partiality and locality of CFEs make them epistemically accessible and an adequate form of explanations.

### 3 COUNTERFACTUAL EXPLANATIONS

This section illustrates counterfactual explanations by giving an example and then outlines the major aspects of the problem.

#### 3.1 An Example

Suppose Alice walks into a bank and seeks a home mortgage loan. The decision is impacted in large part by a machine learning classifier that considers Alice’s feature vector of \{Income, CreditScore, Education, Age\}. Unfortunately, Alice is denied the loan she seeks and is left wondering: (1) why the loan was denied? and (2) what can she do differently so that the loan will be approved in the future? The former question might be answered with explanations like: “CreditScore was too low”, and is similar to the majority of traditional explainability methods. The latter question forms the basis of a counterfactual explanation: what small changes could be made to Alice’s feature vector in order to end up on the other side of the classifier’s decision boundary? Let us suppose the bank provides Alice with exactly this advice (through a CFE) of what she might change in order to be approved next time. A possible counterfactual recommended by the system might be to increase her Income by $10K or get a new master’s degree or a combination of both. The answer to the former question does not tell Alice what action to take, while the CFE explicitly helps her. Figure 1 illustrates how the datapoint representing an individual, which originally got classified in the negative class, can take two paths to cross the decision boundary into the positive class region.

The assumption in a CFE is that the underlying classifier would not change when the applicant applies in the future. And if the assumption holds, the counterfactual guarantees the desired outcome in the future time.

#### 3.2 Desiderata and Major Themes of Research

The previous example alludes to many desirable properties of an effective counterfactual explanation. For Alice, the counterfactual should quantify a relatively small change, which will lead to the desired alternative outcome. Alice might need to increase her income by $10K to get approved for a loan, and even though an increase of $50K would do the job, it is most pragmatic for her if she can make the smallest possible change. Additionally, Alice might care about a simpler explanation – it is easier for her to focus on changing a few things (such as only Income) instead of trying to change many features. Alice certainly also cares that the counterfactual she receives is giving her advice, which is realistic and actionable. It would be of little use if the recommendation were to decrease her age by ten years.

These desiderata, among others, have set the stage for recent developments in the field of counterfactual explainability. As we describe in this section, major themes of research have sought to incorporate increasingly complex constraints on counterfactuals, all in the spirit of ensuring the resulting explanation is truly actionable and helpful. Development in this field has focused on addressing these desiderata in a way that is generalizable across algorithms and is computationally efficient.

(1) **Validity:** Wachter et al. \cite{Wachter2017} first proposed counterfactual explanations in 2017. They posed CFE as an optimization problem. Equation (1) states the optimization objective, which is to minimize the distance between the counterfactual (\(x'\)) and the original datapoint (\(x\)) subject to the constraint that the output of the classifier on the counterfactual is the desired label (\(y' \in \mathcal{Y}\)). Converting the objective into a differentiable, unconstrained form yields two terms (see Equation (2)). The first term encourages the output of the classifier on the counterfactual to be close to the desired class, and the second term forces the counterfactual to be close to the original datapoint. A metric \(d\) is used to measure the distance between two datapoints \(x, x' \in \mathcal{X}\), which can be the L1/L2 distance, or quadratic distance, or distance functions which take as input the CDF of the features \cite{Wachter2017}, or pairwise feature costs as perceived by users \cite{Zafar2018}. Thus, this original definition already emphasized that an effective counterfactual must be small change relative to the starting point.
\[
\begin{align*}
\text{arg min } d(x, x') & \text{ subject to } f(x') = y' \quad (1) \\
\text{arg min } \max_{x'} \lambda (f(x') - y')^2 + d(x,x') & \quad (2)
\end{align*}
\]

A counterfactual that indeed is classified in the desired class is a valid counterfactual. As illustrated in fig. 1, the points shown in red and green are valid counterfactuals, as they are in the positive class region. The distance to the red counterfactual is smaller than the distance to the green counterfactual.

(2) **Actionability:** An important consideration while making a recommendation is about which features are mutable (e.g., income, age) and which are not (e.g., race, country of origin). A recommended counterfactual should never change the immutable features. In fact, if a change to a legally sensitive feature produces a change in prediction, it shows inherent bias in the model. Several papers have also mentioned that an applicant might have a preference order amongst the mutable features (which can also be hidden.) The optimization problem is modified to take this into account. We might call the set of actionable features \( \mathcal{A} \), and update our loss function to be,

\[
\text{arg min } \max_{x' \in \mathcal{A}} \lambda (f(x') - y')^2 + d(x,x') 
\]

(3) **Sparsity:** There can be a trade-off between the number of features changed and the total amount of change made to obtain the counterfactual. A counterfactual ideally should change a smaller number of features in order to be the most effective. It has been argued that people find it easier to understand shorter explanations [218, 227], making sparsity an important consideration. We update our loss function to include a penalty function that encourages sparsity in the difference between the modified and the original datapoint, \( g(x' - x) \), e.g., L0/L1 norm.

\[
\text{arg min } \max_{x' \in \mathcal{A}} \lambda (f(x') - y')^2 + d(x,x') + g(x' - x) 
\]

(4) **Data Manifold closeness:** It would be hard to trust a counterfactual if it resulted in a combination of features that were utterly unlike any observations the classifier has seen before. In this sense, the counterfactual would be “unrealistic”, not easy to realize, and anomalous to the training datapoints [40]. Therefore, a generated counterfactual should be realistic in the sense that it is near the training data and adheres to observed correlations among the features. Many papers have proposed various ways of quantifying this. We might update our loss function to include a penalty for adhering to the data manifold defined by the training set \( X \), denoted by \( l(x',X) \)

\[
\text{arg min } \max_{x' \in \mathcal{A}} \lambda (f(x') - y')^2 + d(x,x') + g(x' - x) + l(x',X) 
\]

In fig. 1, the region between the dashed lines shows the data manifold. There are two possible paths to cross the decision boundary for the blue datapoint. The shorter, red path takes it to a counterfactual that is outside the data manifold, whereas a bit longer, the green path takes it to a counterfactual that follows the data manifold. Adding the data manifold loss term encourages the algorithm to choose the green path over the red path, even if it is slightly longer.

(5) **Causality:** Features in a dataset are rarely independent, therefore, changing one feature in the real world affects other features. For example, getting a new educational degree necessitates increasing the individual’s age by at least some amount. In order to be realistic and actionable, a counterfactual should maintain any known causal relations between features. Generally, our loss function now accounts for (1) counterfactual validity, (2) sparsity in feature vector (and actionability of features); (3) similarity to the training data; and (4) causal relations.

The following research themes are not added as terms in the optimization objective; they are properties of the algorithm generating the CFEs.

(6) **Amortized inference:** Generating a counterfactual is expensive, which involves solving an optimization process for each datapoint. Mahajan et al. [210] proposed generative technique for “amortized inference” of CFEs. Learning to predict a CFE allows the algorithm to quickly compute a counterfactual (or several) for any new input \( x \), without requiring to solve an optimization problem. Verma et al. [316] proposed another approach that uses RL to generate amortized CFEs.

(7) **Black-box access:** If a CFE generating approach can work with the black-box access to an ML model, i.e., with only accessing its ‘predict’ function, it can then be used in settings where the access to the ML model cannot be given due to proprietary or legal reasons. Dandl et al. [67] propose a genetic algorithm and Verma et al. [316] propose a RL-based algorithm to this end.

(8) **Model Agnosticism:** A closely linked concept is model agnosticism. An approach that is model agnostic can work with different kinds of ML models and hence is more desirable than a model-specific approach. An approach that requires black-box access to the model is model-agnostic by definition.

### 3.3 Relationship to other related terms

Out of the papers collected, different terminology often captures the basic idea of counterfactual explanations, although subtle differences exist between the terms. Several terms worth noting include:

- **Algorithmic Recourse:** Ustun et al. [310] point out that counterfactuals do not take into account the actionability of the prescribed changes, which recourse does. Works taking a causal view of the problem further fortify this claim [168, 169]. Recent papers in counterfactual generation take actionability and feasibility of the prescribed changes, and therefore the difference with recourse has blurred. In this work, we use the term counterfactual explanation, its abbreviation CFE, and recourse interchangeably.

- **Inverse classification:** Inverse classification aims to perturb an input in a meaningful way in order to classify it into its desired class [4, 189]. Such an approach prescribes the actions to be taken in order to get the desired classification. Therefore inverse classification has the same goals as CFEs.

- **Contrastive explanation:** Contrastive explanations generate explanations of the form “an input x is classified as y because features \( f_1, f_2, \ldots, f_k \) are present and \( f_a, \ldots, f_r \) are absent”. The features that are minimally sufficient for a classification are called pertinent positives, and the features whose absence is necessary for the final classification are termed pertinent negatives. To generate both pertinent positives and pertinent negatives, one needs...
to solve the optimization problem to find the minimum perturbations needed to maintain the same class label or change it, respectively. Therefore contrastive explanations (specifically pertinent negatives) are related to CFEs.

- **Adversarial learning**: Adversarial learning is closely related, but the terms are not interchangeable. Adversarial learning aims to generate the least amount of change in a given input to classify it differently, often with the goal of far-exceeding the decision boundary and resulting in a highly-confident misclassification. While the optimization problem is similar to the one posed in a counterfactual generation, the desiderata are different. For example, in adversarial learning (often applied to images), the goal is an imperceptible change in the input image. This is often at odds with the CFE’s goal of sparsity and parsimony (though single-pixel attacks are an exception). Further, notions of data manifold and actionability/causality are rarely considerations in adversarial learning. A few works point to the similarity and synergy between the two domains: Paweleczyk et al. [239] explore the connection between the optimization objectives and results of the adversarial and CFE generating techniques. Freiesleben [105] state that the differences in the desired class label and distance from the original datapoint distinguish CFEs from adversarial examples. Elliott et al. [91] propose generating semantically meaningful adversarial perturbations to generate CFEs for images. Browne and Swift [41] point out that the constraint of producing plausible datapoints distinguishes CFEs from adversarial examples.

#### 4 ASSESSMENT OF THE APPROACHES ON COUNTERFACTUAL PROPERTIES

For easy comprehension and comparison, we identify several properties that are important for a counterfactual generation algorithm. For all the collected papers which propose an algorithm to generate counterfactual explanations, we assess the algorithm they propose against these properties. The results are presented in Table 1. For papers that do not propose new algorithms and discuss related aspects of counterfactual explanations or modifications to previous methods are mentioned in section 5.3. The methodology we used to collect the papers is given in appendix B.

**4.1 Properties of counterfactual algorithms**

This section expounds on the key properties of a counterfactual explanation generation algorithm. The properties form the columns of Table 1.

1. **Model access**: The counterfactual generation algorithms require different levels of access to the underlying model for which they generate counterfactuals. We identify three distinct access levels - access to complete model internals, access to gradients, and access to only the prediction function (black-box). Access to the complete model internals is required when the algorithm uses a solver-based method like, mixed integer programming [164, 167, 168, 267, 310] or if they operate on decision trees [48, 97, 203, 221, 302] which requires access to all internal nodes of the tree. A majority of the methods use a gradient-based algorithm to solve the optimization objective, modifying the loss function proposed by Wachter et al. [324], but this is restricted to differentiable models only. Black-box approaches use gradient-free optimization algorithms such as Nelder-Mead [124], growing spheres [191], FISTA [79, 311], ASP [32], or genetic algorithms [67, 189, 278] to solve the optimization problem. Finally, some approaches do not cast the goal into an optimization problem and solve it using heuristics [126, 173, 254, 334]. Poyiadzi et al. [247] propose FACE, which uses Dijkstra’s algorithm [80] to find the shortest path between existing training datapoints to find counterfactual for a given input. Hence, this method does not generate new datapoints. Fraunhofer IOSB et al. [104] and Blanchart [35] divide the feature space into ‘pure’ regions where all datapoints (by sampling) belong to one class and then use graph traversing techniques to find the closest CFEs.

Distinct from the three levels of model access, there exist approaches that propose new training routines. Ross et al. [265] propose adding adversarial loss during training of the ML model to give a higher probability of having a recourse for the training datapoints. (After training, any CFE generating method can be used.) Guo et al. [130] propose CounterNet, a novel architecture that predicts the class and generates the CFE of a datapoint when trained from scratch. [277] train a sum-product network that acts as both a classifier and density estimator and uses that to generate CFEs.

2. **Model agnostic**: This column describes the domain of models a given algorithm can operate on. For example, gradient-based algorithms can only handle differentiable models, and the algorithms based on solvers require linear or piece-wise linear models [164, 167, 168, 267, 310], some algorithms are model-specific and only work for those models like tree ensembles [97, 164, 203, 302]. Black-box methods have no restriction on the underlying model and are, therefore, model-agnostic.

3. **Optimization amortization**: Among the collected papers, the proposed algorithm mostly returned a single counterfactual for a given input datapoint. Therefore these algorithms require solving the optimization problem for each counterfactual that was generated, that too, for every input datapoint. A smaller number of the methods are able to generate multiple counterfactuals (generally diverse by some metric of diversity) for a single input datapoint; therefore, they require to be run once per input to get several counterfactuals [48, 67, 97, 126, 167, 210, 224, 267, 278]. Mahajan et al. [210]’s approach learns the mapping of datapoints to counterfactuals using a variational auto-encoder (VAE) [82]. Therefore, once the VAE is trained, it can generate multiple counterfactuals for all input datapoints, without solving the optimization problem separately and is thus very fast. Verma et al. [316] and Samoilescu et al. [270] train a reinforcement learning model to learn the actions that need to be taken to generate CFEs for a data distribution. Hence, these approaches are also amortized. [344] trains a CGAN to synthesize CFEs with umbrella sampling; hence, their approach is also amortized. Van Looveren et al. [312] also train aGAN-based model that is amortized. Schleich et al. [272] partially evaluate (amortize) the classifier for the static features, hence speeding up the CFE generation. We report two aspects of optimization amortization in the table.
• **Amortized Inference:** This column is marked Yes if the algorithm can generate counterfactuals for multiple input data points without optimizing separately for them; otherwise, it is marked No.

• **Multiple counterfactuals (CF):** This column is marked Yes if the algorithm can generate multiple counterfactuals for a single input datapoint; otherwise, it is marked No.

(4) **Counterfactual (CF) attributes:** These columns evaluate algorithms on sparsity, data manifold adherence, and causality. Among the collected papers, methods using solvers explicitly constrain sparsity [167, 310], black-box methods constrain L0 norm of counterfactual and the input datapoint [67, 191]. Gradient-based methods typically use the L1 norm of counterfactual and the input datapoint. Some of the methods change only a fixed number of features [173, 334], change features iteratively [160, 193, 273, 316], or flip the minimum possible split nodes in the decision tree [126] to induce sparsity. Some methods also induce sparsity post-hoc [191, 224]. This is done by sorting the features in ascending order of relative change and greedily restoring their values to match the values in the input datapoint until the prediction for the CFE is still different from the input datapoint.

Adherence to the data manifold has been addressed using several different approaches, like training VAEs on the data distribution [78, 159, 210, 311], constraining the distance of a counterfactual from the k nearest training datapoints [67, 89, 164], directly sampling points from the latent space of a VAE trained on the data, and then passing the points through the decoder [243], using an ensemble of model to capture the predictive entropy [273], using an Kernel Density Estimator (KDE) to estimate PDF of underlying data manifold [109], using cycle consistency loss in GAN [312], mapping back to the data domain [193], using a combination of existing datapoints [173], using Gaussian Mixture Models to approximate the probability of in-distributionness [19], or by using feature correlations [20], or by simply not generating any new datapoint [247].

The relation between different features is represented by a directed graph between them, which is termed as a causal graph [244]. Out of the papers that have addressed this concern, most require access to the complete causal graph [168, 169] (which is rarely available in the real world), while Duong et al. [89], Mahajan et al. [210], Verma et al. [316], Yang et al. [344] can work with partial causal graphs.

These three properties are reported in the table.

• **Sparsity:** This column is marked No if the algorithm does not consider sparsity, else it specifies the sparsity constraint.

• **Data manifold:** This column is marked Yes if the algorithm forces the generated counterfactuals to be close to the data manifold by some mechanism; otherwise, it is marked No.

• **Causal relation:** This column is marked Yes if the algorithm considers the causal relations between features when generating counterfactuals; otherwise, it is marked No.

(5) **Counterfactual (CF) optimization (opt.) problem attributes:** These are a few attributes of the optimization problem.

Out of the papers that consider feature actionability, most classify the features into immutable and mutable types. Karimi et al. [168] and Lash et al. [189] categorize the features into immutable, mutable, and actionable types. Actionable features are a subset of mutable features. They point out that certain features are mutable but not directly actionable by the individual, e.g., CreditScore cannot be directly changed; it changes as an effect of changes in other features like income, credit amount. Mahajan et al. [210] uses an oracle to learn the user preferences for changing features (among mutable features) and can also learn hidden preferences.

Most tabular datasets have both continuous and categorical features. Performing arithmetic over continuous features is natural, but handling categorical variables in gradient-based algorithms can be complicated. Some algorithms cannot handle categorical variables and filter them out [191, 203]. Wachter et al. [324] proposed clamping all categorical features to each of their values, thus spawning many processes (one for each value of each categorical feature), leading to scalability issues. Some approaches convert categorical features to one-hot encoding and then treat them as numerical features. In this case, maintaining one-hotness can be challenging. Some use a different distance function for categorical features, which is generally an indicator function (1 if a different value, else 0). [109] use Markov chain transitions to encode categorical distances. Yang et al. [344] use Gaussian mixture models to normalize the continuous features and Gumbel-Softmax to relax categorical features into continuous ones. Genetic algorithms, evolutionary algorithms, and SMT solvers can naturally handle categorical features. We report these properties in the table.

• **Feature preference:** This column is marked Yes if the algorithm considers feature actionability, otherwise marked No.

• **Categorical distance function:** This column is marked No if the algorithm does not use a separate distance function for categorical variables, else it specifies the distance function.

5 EVALUATION OF COUNTERFACTUAL GENERATION ALGORITHMS

This section lists the common datasets used to evaluate counterfactual generation algorithms and the metrics on which they are typically evaluated and compared.

5.1 Commonly used datasets for evaluation

The datasets used in the evaluation in the papers we review can be categorized into tabular and image datasets. Not all methods support image datasets. Some of the papers also used synthetic datasets for evaluating their algorithms, but we skip those in this review since they were generated for a specific paper and also might not be available. Common datasets in the literature include:

• **Image:** MNIST [194], EMNIST [60], CelebA [200], CheXpert [152], ImageNet [77], ISIC Skin Lesion [59], ADNI [225], ChestX-ray8 [326].

---

1 It considers global and local feature importance, not preference.
2 All features are converted to polytope type.
3 Does not generate new datapoints
4 The distance is calculated in latent space.
5 It considers feature importance not user preference.
6 Maybe partially as it uses cycle consistency loss.
Table 1: Assessment of the collected papers on the key properties, which are important for readily comparing and comprehending the differences and limitations of different counterfactual algorithms. Papers are sorted chronologically. Details about the full table is given in appendix A.

| Year | Paper | Model access | Model domain | Assumptions | Optimization amortization | CF attributes | CF opt. problem attributes |
|------|-------|--------------|--------------|-------------|--------------------------|---------------|-----------------------------|
|      |       |              |              |             | Amortized Inference | Multiple CPEs | Sparsity | Data manifold | Causal relation | Feature preference | Categorical dist. func |
| 2017 | [189] | Black-box    | Agnostic     | No          | No           | No               | Yes       | No          | -           | -                | -                  |
|      | [324] | Gradients    | Differentiable | No          | No           | L1              | No        | No          | No          | -                | -                  |
|      | [302] | Complete     | Tree ensemble | No          | No           | No               | No        | No          | No          | -                | -                  |
|      | [191] | Black-box    | Agnostic     | No          | No           | L0 and post-hoc acceptor | No        | No          | No          | -                | -                  |
| 2018 | [126] | Black-box    | Agnostic     | No          | Yes          | Flips min. split nodes | No        | No          | No          | Indicator        | N.A. 2              |
|      | [78]  | Gradients    | Differentiable | No          | No           | L1              | Yes        | No          | No          | -                | -                  |
|      | [124] | Black-box    | Agnostic     | No          | No           | No               | No        | No          | No          | No                | No 1               |
|      | [267] | Complete     | Linear       | No          | Yes          | L1              | No        | No          | No          | No                | N.A. 2              |
|      | [310] | Complete     | Linear       | No          | No           | Hard constraint | No        | No          | Yes         | -                | -                  |
|      | [278] | Black-box    | Agnostic     | No          | Yes          | No               | No        | No          | Yes         | Indicator        | -                  |
| 2019 | [79]  | Black-box or gradient | Differentiable | No          | No           | L1              | Yes        | No          | No          | -                | -                  |
|      | [254] | Black-box    | Agnostic     | No          | No           | No               | No        | No          | No          | -                | -                  |
|      | [159] | Gradients    | Differentiable | No          | No           | No               | Yes        | No          | No          | -                | -                  |
|      | [250] | Gradients    | Differentiable | No          | No           | No               | No        | No          | No          | -                | -                  |
|      | [334, 335] | Black-box | Agnostic      | No          | No           | Changes one feature | No        | No          | No          | -                | -                  |
|      | [224] | Gradients    | Differentiable | No          | Yes          | L1 and post-hoc acceptor | No        | No          | No          | Indicator        | -                  |
|      | [247] | Black-box    | Agnostic     | No          | No           | No               | Yes 3      | No          | No          | -                | -                  |
|      | [311] | Black-box or gradient | Differentiable | No          | No           | L1              | Yes        | No          | No          | Embedding        | -                  |
|      | [210] | Gradients    | Differentiable | Yes          | Yes          | No               | Yes        | Yes         | Yes         | -                | -                  |
|      | [167] | Complete     | Linear       | No          | Yes          | Hard constraint | No        | No          | Yes         | Indicator        | -                  |
|      | [243] | Gradients    | Differentiable | No          | No           | No               | Yes        | No          | Yes         | N.A. 4           | -                  |
| 2020 | [175] | Black-box    | Agnostic     | No          | No           | Yes              | Yes        | No          | No          | No                | -                  |
|      | [168] | Complete     | Linear and causal graph | No          | No           | L1              | Yes        | Yes         | Yes         | -                | -                  |
|      | [169] | Gradients    | Differentiable | No          | No           | No               | Yes        | Yes         | Yes         | -                | -                  |
|      | [193] | Gradients    | Differentiable | No          | No           | Iteratively      | Yes        | No          | No 5        | -                | -                  |
|      | [67]  | Black-box    | Agnostic     | No          | Yes          | L0              | Yes        | No          | Yes         | Indicator        | -                  |
|      | [164] | Complete     | Linear and tree ensemble | No          | No           | No               | Yes        | No          | Yes         | -                | -                  |
|      | [97]  | Complete     | Random Forest | No          | Yes          | L1              | No        | No          | No          | -                | -                  |
|      | [202, 203] | Complete | Tree ensemble | No          | No           | L1              | No        | No          | No          | No                | -                  |

- **Tabular:** Adult income, German credit, Student Performance, Breast cancer, Default of credit, Shopping, Iris, Wine, Spam-bee, Coveryte, ICU [87], LendingClub [294], Give Me Some Credit [162], COMPAS [155], LSAT [36], Pima diabetes [283], HELOC/FICO [100], Fannie Mae [208], Portuguese Bank [223], Sangiovese [209], Bail dataset [158], Simple-BN [210], AllState [150], WiDS Dataathon [149], Home Credit Default Risk [125], German Housing [102], HospitalTriage [142], MIMIC-IV [157], Freddie Mac [206], UK unsecured personal loans [43], insurance dataset [179], BPIC2017 [145].

5.2 Metrics for evaluation of counterfactual generation algorithms

Most of the counterfactual generation algorithms are evaluated on the desirable properties of counterfactuals. Counterfactuals are
considered actionable feedback to individuals who have received undesirable outcomes from automated decision-makers, and therefore, a user study can be considered a gold standard. The ease of acting on a recommended counterfactual is thus measured by using quantifiable proxies:

1. **Validity**: Validity measures the ratio of the counterfactuals that actually have the desired class label to the total number of counterfactuals generated. Higher validity is preferable. Most papers report it.
2. **Proximity**: Proximity measures the distance of a counterfactual from the input datapoint. For counterfactuals to be easy to act upon, they should be close to the input datapoint. Distance metrics like the L1 norm, L2 norm, Mahalanobis distance are common. To handle the variability of range among different

| Year | Paper | Model access | Model domain | Assumptions | Optimization amortization | CF attributes | CF opt. problem attributes |
|------|-------|--------------|--------------|-------------|---------------------------|---------------|----------------------------|
| 2021 | [312] Gradient | Differentiable | Yes | No | L1 | No<sup>6</sup> | No | No | - |
|      | [48, 134] Complete Decision Tree | No | Yes | L1 | No | No | Yes | - |
|      | [166] Complete Linear | No | Yes | Iteratively | No | Yes | No | - |
|      | [275] Gradients Differentiable | No | No | Iteratively | Yes | No | Yes | - |
|      | [227] Black-box Agnostic | No | Yes | Gower | No | Yes | Yes | Gower |
|      | [42] Black-box Agnostic | No | No | Yes | Yes | No | No | Indicator |
|      | [89] Black-box Agnostic | No | No | No | No | Yes | No | Latent space |
|      | [228] Complete Linear | No | Yes | Hard constraint | Yes | No | Yes | - |
|      | [20] Complete Linear | No | No | No | No | Yes | No | - |
|      | [272] Black-box or complete Agnostic if black-box | No | Yes | L0/L1 | No | Yes | Yes | Indicator |
|      | [230] Black-box or gradient Agnostic if black-box | Yes | No | L1 | Yes | No | Yes | - |
|      | [35] Complete Tree ensemble | Yes | No | Yes | No | No | Yes | - |
|      | [270] Black-box Agnostic | Yes | Yes | L0/L1 | Yes | No | Yes | Indicator |
|      | [316] Black-box Agnostic | Yes | Yes | Iteratively | Yes | Yes | Yes | - |
|      | [238] Complete Tree ensemble | No | No | No | No | Yes | No | Gower |
|      | [221] Complete Linear | No | Yes | Hard constraint | No | No | Yes | Indicator |
|      | [104] Black-box Agnostic | Yes | Yes | No | No | No | No | - |
|      | [344] Black-box Agnostic | Yes | Yes | No | Yes | Yes | No | Not sure |
|      | [160] Gradient Differentiable | No | No | No | No | No | No | - |
|      | [109] Black-box Agnostic | No | No | L1 | Yes | No | No | Markov Chains |
|      | [259] Black-box Agnostic | Partially | Yes | Hard constraint | No | No | Yes | Gower |
|      | [130] Training from scratch | Differentiable | Yes | No | No | No | No | - |
|      | [340] Gradient | Differentiable | No | No | No | Yes | Yes | No | - |
|      | [343] Black-box Agnostic | No | Might | Yes | No | No | Yes | - |
|      | [258] Black-box Agnostic | Yes | Might | Yes | No | No | Yes | Indicator |
|      | [277] Training from scratch | Differentiable | No | No | No | Yes | No | - |

features, some papers standardize them in pre-processing or divide L1 norm by median absolute deviation of respective features [224, 267, 324], or divide L1 norm by the range of the respective features [67, 167, 168]. Some papers term proximity as the average distance of the generated counterfactuals from the input. Lower values of average distance are preferable.

3. **Sparsity**: Shorter explanations are more comprehensible to humans [218], therefore, counterfactuals ideally should prescribe a change in a small number of features. Although a consensus on a hard cap on the number of modified features has not been reached, Keane and Smyth [173] cap a sparse counterfactual to at most two feature changes.

4. **Counterfactual generation time**: Intuitively, this measures the time required to generate counterfactuals. This metric can be averaged over the generation of a counterfactual for a batch of
input datapoints or for the generation of multiple counterfactuals for a single input datapoint.

5. Diversity: Some algorithms support the generation of multiple counterfactuals for a single input datapoint. The purpose of providing multiple counterfactuals is to increase the ease for applicants to reach at least one counterfactual state. Therefore, the recommended counterfactuals should be diverse, allowing applicants to choose the easiest one. If an algorithm is strongly enforcing sparsity, there could be many different sparse subsets of the features that could be changed. Therefore, having a diverse set of counterfactuals is useful. Diversity is encouraged by maximizing the distance between the multiple counterfactuals by adding it as a term in the optimization objective [67, 224] or as a hard constraint [167, 221, 310], or by minimizing the mutual information between all pairs of modified features [193]. Mothidal et al. [224] reported diversity as the feature-wise distance between each pair of counterfactuals. A higher value of diversity is preferable.

6. Closeness to the training data: Recent papers have considered the actionability and realism of the modified features by grounding them in the training data distribution. This has been captured by measuring the average distance to the k-nearest datapoints [67], or measuring the local outlier factor [164], or measuring the reconstruction error from a VAE trained on the training data [210, 311], or measuring the PDF of such datapoints using KDE [109], or measuring the maximum mean discrepancy (MMD) between the original and counterfactual points [312]. A lower value of the distance and reconstruction error is preferable.

7. Causal constraint satisfaction (feasibility): This metric captures how realistic the modifications in the counterfactuals are by ensuring that they satisfy the causal relation between features. Mahajan et al. [210] evaluated their algorithm on this metric.

IM1 and IM2: Van Looveren and Klaise [311] proposed two interpretability metrics specifically for algorithms that use auto-encoders. Let the counterfactual class be $t$, and the original class be $o$. $AE_t$ is the auto-encoder trained on training instances of class $t$, and $AE_o$ is the auto-encoder trained on training instances of class $o$. Let $AE$ be the auto-encoder trained on the full training dataset (all classes).

$$\text{IM1} = \frac{\|x_{cf} - AE_t(x_{cf})\|^2_2}{\|x_{cf} - AE_o(x_{cf})\|^2_2 + \epsilon} \quad (6)$$

$$\text{IM2} = \frac{\|AE_t(x_{cf}) - AE(x_{cf})\|^2_2}{\|x_{cf}\|_1 + \epsilon} \quad (7)$$

A lower value of $\text{IM1}$ implies that the counterfactual ($x_{cf}$) can be better reconstructed by the auto-encoder trained on the counterfactual class ($AE_t$) compared to the auto-encoder trained on the original class ($AE_o$). Thus implying that the counterfactual is closer to the data manifold of the counterfactual class. A lower value of $\text{IM2}$ implies that the reconstruction from the auto-encoder trained on the counterfactual class and the auto-encoder trained on all classes is similar. Therefore, a lower value of $\text{IM1}$ and $\text{IM2}$ means a more interpretable counterfactual.

9. Label Variation Score and Oracle Score: Hvilshøj et al. [147] point out that the previous metrics are unable to detect out-of-distribution CFEs (especially for high dimensional datasets) and propose two new metrics. Label Variation Score applies when each datapoint has multiple labels, and the intuition is that CFE for a particular label should not affect the predictions for other labels (unless they are highly correlated).

$$\text{LVS} = \sum_{l \in L} d_{dis}[p_l(x), p_l(CFE(x))] \quad (8)$$

where $L$ is the total number of labels for a datapoint and $p_l$ is the predicted probability for the specific label $l$, and $d_{dis}$ measures the divergence between the predicted probability of label $l$ for the original datapoint $x$ and its CFE. Oracle Score is similar to validity, however, with an additional classifier trained on the same dataset as the original classifier. The intuition is that if a CFE is more like an adversarial example for a classifier, the CFE would not be classified in the desired class by the other classifier, and hence we use the prediction from the additional classifier as the ground truth validity.

Some of the reviewed papers did not evaluate their algorithm on any of the above metrics. They only showed a couple of example inputs and respective CFEs, details about which are available in the full table (see appendix A).

5.3 Other works

This section enlists works that talk about the desirable properties of counterfactuals or point to their issues. We also talk about works that propose minor modifications to previous similar approaches.

Works exploring desirable CFE properties: Sokol and Flach [286] list several desirable properties of counterfactuals inspired from Miller [218] and state how the method of flipping logical conditions in a decision tree satisfies most of them. Laugel et al. [190] enlist proximity, connectedness, and stability as three desirable properties of a CFE and propose the metrics to measure them.

Works pointing to issues with CFEs: Laugel et al. [192] says that if the explanation is not based on training data, but the artifacts of non-robustness of the classifier, it is unjustified. They define justified explanations to be connected to training data by a continuous set of datapoints, termed $E$-chainability. Barocas et al. [28] state five reasons that have led to the success of counterfactual explanations and also point out the overlooked assumptions. They mention the unavoidable conflicts which arise due to the need for privacy invasion in order to generate helpful explanations. Kasirzadeh and Smart [171] provide philosophical insight into the implicit assumptions and choices made when generating CFEs.

Causal CFEs: Downs et al. [86] propose using conditional subspace VAEs (CSVAE), a variant of VAEs, to generate CFEs that obey correlations between features, causal relations between features, and personal preferences. This method builds a probabilistic data model of the training data using a CSVAE and uses it to generate CFEs. However, these CFEs are not with respect to a specific ML model. Crupi et al. [66] propose a technique that can be used with any counterfactual generation approach to make causality abiding CFEs. von Kügelgen et al. [321] extend Karimi et al. [169]’s work...
to the setting where unobserved confounders may be present in the causal setting. de Lara et al. [71] show that optimal transport-based methods are an approximation of Pearl’s CFEs and hence can be used to generate causal CFEs. Beckers [31] delve further into the integration of causality, actual causation, and CFEs.

**CFE for specific models:** Albini et al. [11] propose a CFE generation approach targeted for Bayesian network classifiers. Artelt and Hammer [18, 19] enlists the counterfactual optimization problem formulation for several model-specific cases, like generalized linear model, gaussian naive bayes, and mention the general algorithm to solve them. Koopman and Renooij [180] propose a BFS-based technique for generating CFEs for Bayesian networks.

**Works considering multi-agent scenarios of CFEs:** Tsirtsis with the pertinent positives and pertinent negatives [78] of the CFEs. Guidotti and Ruggieri [128] propose LORELEY that extends LORE [126] to generate CFEs quantitatively compare 10 CFE generating approaches using 22 Benchmark and dataset curation: Huang et al. [27] propose using trust scores to measure the out-of-distributionness of the CFEs. Delaney et al. [75] propose using trust scores to measure the out-of-distributionness of the CFEs. Guidotti and Ruggieri [128] propose using an ensemble of base CFE explainers to generate diverse CFEs.

**Global CFEs:** Rawal and Lakkaraju [258] propose AReS to generate rule lists that act as global CFEs. Ley et al. [197] and Kenamori et al. [165] propose computationally more efficient implementation of Rawal and Lakkaraju [258]'s work. Carrizosa et al. [49] propose a mixed integer quadratic model to generate CFEs for a group of datapoints. Koo et al. [179] propose generating CFEs for a set of datapoints using lagrangian and subgradient methods. Pedapati et al. [245] propose a technique to train a globally interpretable model (for a black-box model) such that this model is consistent with the pertinent positives and pertinent negatives [78] of the training datapoints used to train the original model.

**Works proposing modifications to previous approaches:** Chen et al. [57] and De Toni et al. [72] use RL to generate CFE as was also proposed by Verma et al. [316]. Rasouli and Chieh Yu [252] propose a genetic algorithm to generate CFEs as was also proposed by Dandl et al. [67]. Hashemi and Fathi [137] propose to use genetic algorithm for CFE generation similar to Dandl et al. [67]'s work. Monteiro and Reynoso-Meza [222] propose extending Dandl et al. [67]'s approach using U-NSGA-III evolutionary algorithm. Barr et al. [29] extend Mahajan et al. [210]’s work by interpolating between the input and CFE datapoint to generate CFEs closer to the input datapoint. Sajja et al. [269] propose using a semi-supervised autoencoder instead of the traditional unsupervised autoencoder to generate CFEs close to the training data manifold. Huang et al. [145] propose LORELEY that extends LORE [126] to generate CFEs for multi-class classification problems and account for flow constraints. Wijekoon et al. [337] use feature importance provided by LIME to assist the case-based reasoning approach to generate CFEs. Delaney et al. [75] propose using trust scores to measure the out-of-distributionness of the CFEs. Guidotti and Ruggieri [128] propose using an ensemble of base CFE explainers to generate diverse CFEs.

**Benchmark and dataset curation:** Mazzine and Martens [214] quantitatively compare 10 CFE generating approaches using 22 datasets and nine metrics. Pawelczyk et al. [240] and Artelt [17] have developed extensible toolboxes where several CFE approaches can be plugged in and compared on specific datasets.

**Various uncategorized works:** State [288] talk about generating CFEs with real-world constraints on features and adaptability with updating ML models using constraint logic programming. Tahoun and Kassis [291] propose to disentangle actions from feature modifications to address the lack of intervention data and appropriate action costs. The users should already describe the actions they are willing to take, and a model should just choose the minimum cost action that generates the CFE. Lucic et al. [201] propose a CFE approach to provide a lower and upper bound for the feature values that get a low prediction error from the ML model for a datapoint that originally had a high prediction error. Korikov and Beck [181], Korikov et al. [182] show how CFEs can be generated by using the generalization of inverse combinatorial optimization and solve it under two objectives. Pawelczyk et al. [241] provide a general upper bound on the cost of counterfactual explanations under the phenomenon of predictive multiplicity, wherein more than one trained models have the same test accuracy and there is no clear winner among them. Fdez-Sánchez et al. [95] propose a hierarchical decompositions-based method to obtain CFEs for multi-class classification problems. Bertossi [32] and Medeiros Raimundo et al. [215] propose brute force approaches to generate CFEs.

### 6 COUNTERFACTUAL EXPLANATIONS FOR OTHER DATA MODALITIES

Since we restrict this survey to the papers that generate CFEs for tabular data, in this section we point the readers to the papers that propose algorithms targeted towards other data modalities:

(1) **Image data:** [1, 8, 9, 12, 13, 27, 69, 91, 96, 101, 115, 122, 129, 133, 138, 146, 148, 153, 154, 174, 175, 188, 198, 199, 217, 235, 236, 246, 264, 271, 284, 299, 312, 313, 318, 325, 336, 345, 347, 353].

(2) **Text data:** [38, 54, 160, 207, 251, 255, 263, 301, 345–347].

(3) **Speech data:** [351].

(4) **Time-series data:** [24, 74, 144, 170, 290, 305, 312, 329, 330].

(5) **Graph data for graph neural networks:** [2, 25, 26, 92, 204, 232, 332]. A survey for CFE on graph neural networks: [248].

(6) **Agent action (e.g. Reinforcement Learning or Planning):** [39, 237, 289].

(7) **Recommender systems:** [73, 116, 117, 161, 276, 293, 303, 341, 354, 356].

(8) **Functional data:** [50, 183] and **Behavioral data:** [251].

### 7 OTHER APPLICATIONS OF COUNTERFACTUAL EXPLANATIONS

Here we refer the readers to other applications where counterfactual explanations are being used apart from explaining ML models:

(1) **Anomaly and data-drift detection:** Hinder and Hammer [140] propose to use CFEs to explain data drift. Sulem et al. [290] propose to use CFEs to explain anomalies in time-series datasets. Ravi et al. [256] wrote a survey on the explainability techniques for convolutional auto-encoders for anomaly detection of images. Haldar et al. [135] propose to use CFEs to explain anomaly detection when using autoencoders. Antoran et al. [15] use
CFEs to find changes in a datapoint that would help a classifier have a higher confidence in its prediction.

(2) **Training dataset debugging:** Yousefzadeh and O’Leary [349] propose to use CFEs to debug ML models by diagnosing the behavior and using synthetic data to alter the decision boundaries. Qi and Chelmis [249] propose to use CFEs to debug potentially mislabeled datasets. Gan et al. [111] propose to use CFEs to detect bugs in financial models. Han and Ghosh [136] propose finding a minimal subset of training datapoints that are responsible for a particular prediction and hence can be used to debug training datasets.

(3) **Data augmentation:** Yuan et al. [350] propose to use CFEs to augment training data that is used to predict market volatility based on earning calls. Temraz and Keane [296] propose using CFEs to augment training data to tackle the class imbalance problem. Mehedi Hasan and Talbert [216], Rasouli and Yu [253] propose using CFEs for data augmentation of tabular datasets for increased robustness. Temraz et al. [297] propose using CFEs to generate data points that can be used to train ML models that predict crop growth (affected by climate change).

(4) **Drug designing:** Nguyen et al. [231] use CFEs to find changes in a drug and protein molecule that will increase their affinity for each other. They use multi-agent RL to this end.

(5) **ML model bias detection:** [94, 226, 310].

(6) **Various applications:** Mazzine et al. [213] propose to use CFEs in employment services to help job seekers get personalized advice for increasing their propensity for getting recommended for a job and to help the ML developers to detect potential bias and other issues in their ML model. Sadler et al. [268] propose to use CFEs for community detection in social networks. Fujisawa et al. [108] propose to use CFEs to understand interactive dimensionality reduction. Tsiaikamaki and Ragos [304] propose to use CFEs for providing actionable suggestions to improve student performance in a university course. Cong et al. [63] propose a CFE approach to explain why a test set fails the Kolmogorov-Smirnov test. Marchezini et al. [211] propose to use CFE for altering both observational and latent variables to reason about mental health. Yao et al. [348] propose to use counterfactuals for evaluating the explanations for recommender systems. Gupta et al. [131] use CFEs to propose changes to constraint satisfaction problems that have no solutions. Teofilii et al. [298] propose using CFEs to explain entity resolution models. Artelt et al. [21] use CFEs to explain the differences between the learning of a pair of models. Frohberg and Binder [107] propose a new dataset, CRASS, to test reasoning and natural language understanding of LLMs.

There has been one case of real-world deployment of CFEs in a hiring platform, Hired. Nemirovsky et al. [229] use a GAN-based approach [230] to suggest changes in features like expected salary, years of experience, and skills to candidates in order to get them approved by the Hired Marketplace ML model.

**8 OPEN QUESTIONS AND RESEARCH PROGRESS FOR SOLVING THEM**

In the first version of this survey paper, we delineated the open questions and challenges yet to be tackled by the existing works pertaining to CFEs [315]. In this version, we supplement this section with the research progress made towards solving them and new research challenges.

**Research Challenge 1.** Unify counterfactual explanations with traditional “explainable AI.”

Although counterfactual explanations have been credited to eliciting causal thinking and providing actionable feedback to users, they do not tell which feature(s) was the principal reason for the original decision and why. It would be nice if, along with giving actionable feedback, counterfactual explanations also gave the reason for the original decision, which can help applicants understand the model’s logic. This is addressed by traditional “explainable AI” methods like LIME [261], Anchors [262], Grad-CAM [274].

**Progress:** Guidotti et al. [126] have attempted this unification, as they first learn a local decision tree and then interpret the inversion of decision nodes of the tree as counterfactual explanations. However, they do not show the CFEs they generate, and their technique also misses other desiderata of counterfactuals (see section 3.2).

**Research Challenge 2.** Provide counterfactual explanations as discrete and sequential steps of actions.

Most counterfactual generation approaches return the modified datapoint, which would receive the desired classification. The modified datapoint (state) reflects the idea of instantaneous and continuous actions, but in the real world, actions are discrete and often sequential. Therefore the counterfactual generation process must take the discreteness of actions into account and provide a series of actions that would take the individual from the current state to the modified state, which has the desired class label.
Wang et al. [327] propose refining the CFEs for different feature sets. Along the lines of an incomplete causal graph, counterfactual explanations can also work with partial causal graphs. Kanamori et al. [166] propose a mixed-integer based programming method and Verma et al. [316] propose an RL-based method that generates ordered sequences of actions as a CFE.

**Research Challenge 3.** *Extend counterfactual explanations beyond classification.*

Progress: Recent work has been extending counterfactual explanations to different tasks and model architectures. Spooner et al. [287] propose a Bayesian optimization-based technique for generating CFEs for regression problems. Numeroso and Bacciu [252] propose an RL-based approach for generating CFEs for graph neural networks, which are used to predict chemical molecule properties. Delaney et al. [74] propose a case-based reasoning approach to generate CFEs for a time-series classifier. See Section 6 and Section 7 for a list of all the approaches.

**Research Challenge 4.** *Counterfactual explanations as an interactive service to the applicants.*

Counterfactual explanations should be provided as an interactive interface, where an individual can come at regular intervals, inform the system of the modified state, and get updated instructions to achieve the counterfactual state. This can help when the individual could not precisely follow the earlier advice for various reasons.

Progress: Hohman et al. [141] developed an interactive user-interface for providing explanations to data scientists. They found out that data scientists used interactivity as the primary mechanism for exploring, comparing, and explaining predictions. Sokol and Flach [285] propose to enhance ML explanations with a voice-assisted interactive service. Akula et al. [9] propose an approach that explains an ML model using an interactive sequence of CFEs. Wang et al. [327] propose refining the CFEs for different feature change costs based on user interactions.

**Research Challenge 5.** *The ability of counterfactual explanations to work with incomplete—or missing—causal graphs.*

Incorporating causality in the counterfactual generation is essential for the CFEs to be grounded in reality. Complete causal graphs and structural equations are rarely available in the real world, and therefore the algorithm should be able to work with incomplete causal graphs.

Progress: Mahajan et al. [210]’s approach was the first to be compatible with incomplete causal graphs. Now other works like Galhotra et al. [116], Verma et al. [316], Schleich et al. [272], Yang et al. [344] can also work with partial causal graphs.

**Research Challenge 6.** *The ability of counterfactual explanations to work with missing feature values.*

Along the lines of an incomplete causal graph, counterfactual explanation algorithms should also be able to handle missing feature values, which often happens in the real world [112].

**Research Challenge 7.** *Scalability and throughput of counterfactual explanations generation.*

As we see in table 1, most approaches need to solve an optimization problem to generate one counterfactual explanation. Some papers generate multiple counterfactuals while optimizing once, but they still need to optimize separately for different input datapoints. However, for industrial deployment, the generation should be more scalable.

Progress: Mahajan et al. [210] learn a VAE which can generate multiple CFEs for any given input datapoint after training. Therefore, their approach is highly scalable and is termed as “amortized inference”. Verma et al. [316] proposed an RL-based technique, FastAR, that also generates amortized CFEs. Van Looveren et al. [312], Samoilescu et al. [270], [344], Rawal and Lakkaraju [258], and Nemirovsky et al. [230] also propose approaches to this end.

**Research Challenge 8.** *Counterfactual explanations should account for bias in the classifier.*

Counterfactuals potentially capture and reflect the bias in the models. To underscore this as a possibility, Ustun et al. [310] experimented on the difference in the difficulty of attaining a counterfactual state across genders, which clearly showed a significant difference. More work must be done to find how equally easy counterfactual explanations can be provided across different demographic groups, or how adjustments should be made to the prescribed changes to account for the bias.

Progress: Rawal and Lakkaraju [258] generate recourse rules for a subgroup that they use to detect model biases. Gupta et al. [132] propose adding a regularizer while training a classifier that encourages the classifier to maintain a similar distance of the decision boundary from different demographic groups, thereby facilitating the opportunity of equal recourse across demographic groups (which is their definition of fairness). von Kügelgen et al. [322] extend this fairness notion when the distance between the recourse is measured in a causal manner. Galhotra et al. [110] propose LEWIS that uses CFEs to identify racial bias in COMPAS and gender in Adult datasets. Dash et al. [69] propose using CFEs to detect bias in image classifiers and counterfactual regularizer to counteract that bias.

**Research Challenge 9.** *Generate robust counterfactual explanations*. [99, 219].

Counterfactual explanation optimization problems force the modified datapoint to obtain the desired class label. However, the modified datapoint could be labeled either in a robust manner or due to the classifier’s non-robustness, e.g., an overfitted classifier. Laugel et al. [190] term this as the stability property of a counterfactual. There are three kinds of robustness needs: 1) robustness to model changes when models are retrained, for example, 2) robustness to the input datapoint (two individuals with a slight change in features should be given similar CFEs), and 3) robustness to small changes in the attained CFE (a CFE with minor changes to the originally suggested CFE should also be accepted).

Progress: Slack et al. [282] underscore this challenge by showing that small perturbations in the input datapoints can result in drastically different CFEs. Rawal et al. [257] further emphasize
this challenge by empirically demonstrating the invalidation of already prescribed recourses when the ML model gets retrained on datasets with temporal or geospatial distribution shifts. Artelt et al. [22] evaluate the robustness of closest CFEs when contrasted with CFEs generated with the data manifold constraint. Bueff et al. [43] propose the framework to measure the robustness of models by purposing generated CFEs as adversarial attack datasets. Vir-golin and Fracaros [320] empirically show that non-robust CFEs encounter a higher cost of change when adverse perturbations are applied to the datapoint, thus concluding that robustness in CFEs should be considered. Upadhyay et al. [309] propose a technique named ROAR that uses adversarial training to generate recourses robust to changes in an ML model that is retrained on a distributionally shifted training dataset. Dominguez-Olmedo et al. [84] show that the CFEs that just cross the decision boundary are usually non-robust and formulate an optimization problem that generates robust recourse for linear models and neural networks. Pawelczyk et al. [242] propose a technique named PROBE that generates robust CFEs while letting the users decide the trade-off between the CFE validation risk and its cost. Black et al. [34] argue that robust CFEs should have high confidence neighborhoods with small Lipschitz constants, and propose a Stable Neighbor Search algorithm to that end. Bui et al. [44] propose an algorithm to generate robust CFEs by considering a distribution over the parameters of the model if retrained. Dutta et al. [90] propose counterfactual stability (the lower bound of the predicted class probability for the sampled datapoints in the neighborhood of a given CFE) as a metric for filtering robust CFEs. Bajaj et al. [26] propose a technique to generate robust CFEs for graph neural networks.

**Research Challenge 10. Counterfactual explanations should handle dynamics (data drift, classifier update, applicant’s utility function changing, etc.)**

All counterfactual explanation papers we review assume that the underlying black box is monotonic and does not change over time. However, this might not be true; credit card companies and banks update their models as frequently as 12-18 months [113]. Therefore counterfactual explanation algorithms should take data drift, the dynamism and non-monotonicity of the classifier into account.

**Research Challenge 11. Counterfactual explanations should capture the applicant’s preferences.**

Along with the distinction between mutable and immutable features (finitely classified into actionable, mutable, and immutable), counterfactual explanations should also capture preferences specific to an applicant. This is important because the ease of changing different features can differ across applicants.

**Progress:** Mahajan et al. [210] captures the applicant’s preferences using an oracle, but that is expensive and is still a challenge. Rawal and Lakkaraju [258] use the Bradley-Terry model to learn the pairwise cost for each feature pair and hence the preference among them. Yadav et al. [343] argue that assuming each user’s cost of changing different features is the same is unrealistic. They propose asking for the user’s cost function or computing the expectation by sampling cost functions from a distribution.

**Research Challenge 12. Counterfactual explanations should also inform the applicants about what must not change**

Suppose a CFE advises someone to increase their income but does not tell that their length of last employment should not decrease. To increase their income, the applicant who switches to a higher-paying job may find themselves in a worse position than earlier. Thus by failing to disclose what must not change, an explanation may lead the applicant to an unsuccessful state [28]. This corroborates RC4, whereby an applicant might be able to interact with a platform to see the effect of a potential real-world action they are considering taking to achieve the counterfactual state.

**Research Challenge 13. Preserving model privacy.**

Privacy attacks on ML models can come in two major forms: member inference and model extraction. Both of these privacy attacks can be enhanced due to the provision of CFEs. Aïvodji et al. [7] empirically demonstrate that adversaries can train a surrogate model with very high fidelity to the original model (i.e., model extraction attack) with as few as 1,000 queries to the model (which is required during CFE generation). The problem is further aggravated when diverse CFEs are provided. Shokri et al. [279] have demonstrated that gradient-based explanations methods leak a lot of information and make the models vulnerable to membership inference attacks. Miura et al. [220] propose MEGEX, a data-free model extraction attack that learns a surrogate model without access to its training data by training a generative model. Wang et al. [328] propose using the CFE of a CFE to train a surrogate model and show that it is more efficient in model extraction when compared to [7].

**Research Challenge 14. Guarding against fairwashing.**

Aïvodji et al. [5] and Aïvodji et al. [6] have pointed out the risk of an adversary using model explanations to rationalize a model’s decisions and obscure its bias. It remains to be seen if the fair recourse approaches can guard against fairwashing.

**Research Challenge 15. CFE interpretability with engineered features** [272].

Most current CFE approaches assume that the features they change are directly input to the ML model. This might not be the case – it is known that model developers use highly engineered features for training the ML models. In this light, approaches need to be developed that take feature engineering into account (potentially a non-differentiable step). Approaches that work with black-box access will naturally be able to work in this setting.

**Research Challenge 16. Handling of categorical features in counterfactual explanations**

Different papers have come up with various methods to handle categorical features, like converting them to one-hot encoding and then enforcing the sum of those columns to be 1 using regularization or a hard constraint, or clamping an optimization problem to a specific categorical value, or leaving them to be automatically handled by genetic approaches and SMT solvers. Measuring distance in categorical features is also not obvious. Some papers use an indicator function, which equates to 1 for unequal values and 0 if the same; other papers convert to one-hot encoding and use standard distance metrics like L1/L2 norm, or use the distance in
Markov chains [102]. Therefore none of the methods developed to handle categorical features are obvious; future research must consider this and develop appropriate methods.

**Research Challenge 17. Evaluate counterfactual explanations using a user study.**

The evaluation for counterfactual explanations must be done using a user study because evaluation proxies (see section 5) might not be able to precisely capture the psychological and other intricacies of human cognition on the ease of actionability of a counterfactual. Keane et al. [172] emphasize the importance of user studies in the context of CFEs. **Progress:** Förster et al. [103] conduct a user study with 144 participants to understand the format of explanation they prefer. They conclude that users prefer concrete, consistent, relevant explanations, and lengthy explanations if they are concrete. Förster et al. [102] conduct a user study with 46 participants who were asked to rate the realism of the CFEs generated by theirs and a baseline approach. Using statistical tests, they concluded that the CFEs generated by their approach were perceived to be more real and typical. Rawal and Lakkaraju [258] conduct a user study with 21 participants who were asked to detect a bias in the recourse summaries for demographic groups. Kanamori et al. [165] conduct a user study with 35 participants to compare their global CFE generating technique with that of Rawal and Lakkaraju [258]. Singh et al. [280] conduct a user study with 54 participants and found that most users prefer specific directives over generic and non-directive explanations. Warren et al. [351] conduct a user study with 127 participants and found that counterfactual explanations elicited higher trust and satisfaction than causal explanations. Yacoby et al. [342] conduct a user study with 8 U.S. state court judges to understand their response to CFEs from pretrial risk assessment instruments (PRAI). They conclude that judges ignored the CFEs and focused on the factual features of the defendant. Kuhl et al. [186] conduct a user study with 74 users in an interactive game setting and found that users benefit less from receiving computationally plausible CFEs than the closest CFEs (measured using feature distance). Zhang et al. [352] conduct a user study with 200 users to check their understanding of global, local, and CF explanations. Cai et al. [47] conduct a user study on 1070 participants to understand how users perceive explanations when provided examples from the desired class vs. when provided examples from all other classes.

**Research Challenge 18. Counterfactual explanations should be integrated with data visualization interfaces.**

Counterfactual explanations will directly interact with consumers with varying technical knowledge levels; therefore, counterfactual generation algorithms should be integrated with visualization interfaces. We already know that visualization can influence human behavior [64], and a collaboration between machine learning and HCI communities could help address this challenge.

**Progress:** Cheng et al. [58], Gomez et al. [119, 120], Leung et al. [195], Wexler et al. [333] have developed interactive graphical user interfaces for displaying CFEs. DECE [58] also summarizes CFEs for subgroups that can help detect model biases, if any. Tamagnini et al. [292] develop a visualization tool for CFEs for text classification models. Hohman et al. [141] also build a visual interactive user interface for providing model explanations.

**Research Challenge 19. Generating optimal recourses when considering a multi-agent scenario.**

O’Brien and Kim [233] demonstrate the non-optimality of recourses generated when a single agent’s interest is considered in a multi-agent scenario like the prisoner’s dilemma. In the real world, an agent’s actions affect other agents, hence generating recourses that consider the interests of multiple agents would be useful.

**Research Challenge 20. Incentivize users to improve features in non-manipulative ways.**

An approach that provides a recourse to users might want to prevent the “gamification” of the model (when users manipulate simple features like the purpose of a loan to get approved). This also protects the ML models from adversarial robustness attacks.

**Progress:** Chen et al. [56] propose the optimization objective for linear classification models when the goal is to develop an accurate model that encourages actual feature improvement for users. They categorize features into three categories: improvement, manipulative, and immutable. Users should be encouraged to change the improvement features, not the manipulative ones when optimizing for recourse. König et al. [187] suggest using causality to generate meaningful recourses and prevent gamification of the model.

**Research Challenge 21. Strengthen the ties between machine learning and regulatory communities.**

A joint statement between the machine learning community and regulatory community (OCC, Federal Reserve, FTC, CFPB) acknowledging successes and limitations of where counterfactual explanations will be adequate for legal and consumer-facing needs and would improve the adoption and use of counterfactual explanations in critical software.

**Progress:** Reed et al. [260] talk about how regulation and policies need to adapt to how ML models can explain their decisions.

### 9 CONCLUSIONS

In this paper, we collected and reviewed more than 350 papers which proposed various algorithmic solutions to finding counterfactual explanations for the decisions produced by automated systems, specifically automated by machine learning. Evaluating all the papers on the same rubric helps in quickly understanding the peculiarities of different approaches and the advantages, and disadvantages of each of them, which can also help organizations choose the algorithm best suited to their application constraints. This has also helped us readily identify the gaps, which will be beneficial to researchers scouring for open problems in this space and quickly sifting the large body of literature. We hope this paper can also be the starting point for people wanting to get an introduction to the broad area of counterfactual explanations and guide them to proper resources for things they might be interested in.

**Acknowledgments.** We thank Jason Wittenbach, Aditya Kusupati, Divyat Mahajan, Jessica Dai, Soumye Singhal, Harsh Vardhan, and Jesse Michel for helpful comments.
Kareem L. Jordan and Tina L. Freiburger. 2015. The Effect of Race/Ethnicity
The US White House. 2022. Blueprint for an AI bill of rights. https:
Xing Han and Joydeep Ghosh. 2021. Model-Agnostic Explanations using Mini-
Paul Jacob, Éloi Zablocki, Hédi Ben-Younes, Mickaël Chen, Patrick Pérez, and
Allstate Insurance. 2011. Allstate Claim Prediction Challenge. https:
Masoud Hashemi and Ali Fathi. 2020. PermuteAttack: Counterfactual Expla-
Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán. 2021. Concept Drift. https:
Frieder Hvilshoj, Alexandros Josifidis, and Ira Assent. 2021. ECINN Efficient
Frederik Hvilshoj, Alexandros Josifidis, and Ira Assent. 2021. On Quantitative
Benedikt Höltgen, Lena Schüt, Jan M. Brauner, and Yarin Gal. 2021. DeDUCE. Generating Counterfactual Explanations Efficiently. https:
Global Women in Data Science Conference The Global Open Source Severity of Illness Score Consortium. 2020. WIDS Datathon 2020. https:
Allstate Insurance. 2011. Allstate Claim Prediction Challenge. https:
France intelligence artificielle. [n. d.]. RAPPORT DE SYNTHESE FRANCE INTELLIGENCE ARTIFICIELLE. https://www.econome.gouv.fr/files/files/PDF/
Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciucaes-Iluc, Chris Chute, Henriik Marklund, Behzad Haghigho, Robyn Ball, Katie Shipan-
Rickie Jones, David B. Larson, Curtis P. Langlotz, Bhavik N Patel, Matthew P. Lungen, and Andrew Y. Ng. 2019. CleXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. https:
Paul Jacob, Eloi Zablocki, Hedi Ben-Younes, Maickel Chen, Patrick Perez, and Matthieu Cord. [n. d.]. STEEX: Steering Counterfactual Explanations with Semantics. https:
Guillaume Jaennerot, Loïc Simon, and Frédéric Jurie. 2021. Diffusion Models for Counterfactual Explanations. https:
Lauren Kirchner Jeff Larson, Surya Mattu and Julia Angwin. 2016. UCI Machine Learning Repository. https://github.com/propublica/compsa-analysis/
Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Cavedon, and Roger Mark. 2021. MIMIC-IV. DataSHIELD: The Uncertainty Library. Springer International Publishing, Cham, 389–394.
Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Cavedon, and Roger Mark. 2021. MIMIC-IV. DataSHIELD: The Uncertainty Library. Springer International Publishing, Cham, 389–394.
Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Fines, John P. Dickerson, and Chirag Shah.
A  FULL TABLE
Initially, we categorized the set of papers with more columns and in a much larger table. We selected the most critical columns and put them in table 1. The full table is available here.

B  METHODOLOGY
B.1 How we collected the paper to review?
We collected a set of more than 350 papers. This section provides the exact procedure used to arrive at this set of papers. For the first version of this survey paper, we had started from a seed set of papers recommended by other people [210, 224, 250, 310, 324] followed by snowballing their references. For this updated (second) version of the paper, we collected papers that cited the first paper that proposed CFES for ML, i.e., Wachter et al. [324] and the first version of this CFE survey paper [314]. For an even complete search, we searched for “counterfactual explanations”, “recourse”, and “inverse classification” on two popular search engines for scholarly articles, Semantic Scholar and Google Scholar. We looked for papers published in the last five years on both search engines. This is a reasonable time frame since the paper that started the discussion of counterfactual explanations in the context of machine learning (specifically for tabular data) was published in 2017 [324]. We collect papers that were published before 31st May 2022. The papers we collected were published at conferences like KDD, IJCAI, PAcT, AAAI, WWW, NeurIPS, WHI, or uploaded to Arxiv.

B.2 Scope of the review
Even though the first paper we reviewed was published online in 2017, and most other papers we review cite it [324] as the seminal paper that started the discussion around counterfactual explanations, we do not claim that this is an entirely new idea. Communities from data mining [98, 212], causal inference [244], and even software engineering [55] have explored similar ideas to identify the principal cause of a prediction, an effect, and a bug, respectively. Even before the emergence of counterfactual explanations in applied fields, they have been the topic of discussion in fields like social sciences [218], philosophy [176, 196, 266], psychology [45, 46, 163]. In this review paper, we restrict our discussion to recent papers that discuss counterfactual explanations in machine learning, specifically classification settings. These papers have been inspired by the emerging trend of FATE and the legal requirements pertaining to explainability in tasks automated by machine learning algorithms.

C  BURGEONING LEGAL FRAMEWORKS AROUND EXPLANATIONS IN AI
To increase the accountability of automated decision systems—specifically, AI systems—laws and regulations regarding the decisions produced by such systems have been proposed and implemented across the globe [85]. The most recent version of the European Union’s General Data Protection Regulation (GDPR), enforced starting on May 25, 2018, offered a right to information about the existence, logic, and envisaged consequences of such a system [121]. This also includes the right to not be a subject of an automated decision-making system. Although the closeness of this law to “right to explanation” is debatable and ambiguous [323], the official interpretation by Working Party for Article 29 has concluded that the GDPR requires explanations of specific decisions, and therefore counterfactual explanations are apt. In the US, the Equal Credit Opportunity Act (ECOA) and the Fair Credit Reporting Act (FCRA) require the creditor to inform the reasons for an adverse action, such as rejection of a loan request [52, 53]. They generally compare the applicant’s feature to the average value in the population to arrive at the principal reasons. Government reports from the United Kingdom [234] and France [151, 319] also touched on the issue of explainability in AI systems. In the US, Defense Advanced Research Projects Agency (DARPA) launched the Explainable AI (XAI) program in 2016 to encourage research into designing explainable models, understanding the psychological requirements of explanations, and the design of explanation interfaces [68]. The European Union has taken similar initiatives as well [61, 308]. The US White House recently put forward the Blueprint for an AI Bill of Rights [143] to modulate decisions from automated systems. The Bill outlines five principles for operating such systems: 1) safe and effective systems, 2) algorithmic discrimination protections, 3) data privacy, 4) explanations for decisions made using such systems, and 5) discussion about human alternatives. While many techniques have been proposed for explainable machine learning, it is yet unclear if and how these specific techniques can help address the letter of the law. Future collaboration between AI researchers, regulators, the legal community, and consumer watchdog groups will help ensure the development of trustworthy AI.