Towards Connecting Use Cases and Methods in Interpretable Machine Learning

Valerie Chen[1*], **Jeffrey Li**[2*], **Joon Sik Kim**[1†], **Gregory Plumb**[1†], **Ameet Talwalkar** [1,3]
[1]Carnegie Mellon University, [2]University of Washington, [3]Determined AI
valeriechen@cmu.edu, jwl2162@cs.washington.edu

Abstract

Despite increasing interest in the field of Interpretable Machine Learning (IML), a significant gap persists between the technical objectives targeted by *researchers' methods* and the high-level goals of *consumers' use cases*. In this work, we synthesize foundational work on IML methods and evaluation into an actionable taxonomy. This taxonomy serves as a tool to conceptualize the gap between researchers and consumers, illustrated by the lack of connections between its methods and use cases components. It also provides the foundation from which we describe a three-step workflow to better enable researchers and consumers to work together to discover what types of methods are useful for what use cases. Eventually, by building on the results generated from this workflow, a more complete version of the taxonomy will increasingly allow consumers to find relevant methods for their target use cases and researchers to identify applicable use cases for their proposed methods.

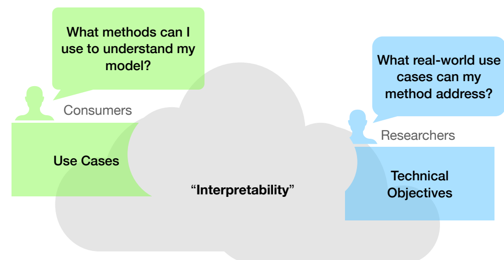

Figure 1: In their current IML practices, researchers focus more on technical objectives while consumers focus on use cases. Often, there remains a lack of explicit connections between the two, making proper usage and development of IML methods difficult for both parties.

1 Introduction

Interpretable Machine Learning (IML)[1] is motivated by a variety of real-world use cases including: helping data scientists debug and select models [29]; aiding domain experts in understanding data-driven decisions [13]; and enabling end-users to gain trust in the systems that affect their lives [9]. However, there remain fundamental disconnects (Figure 1) between the goals of IML researchers and IML consumers.

IML researchers typically develop methods that optimize quantifiable *technical objectives*, e.g., maximizing notions of faithfulness or adherence to some desirable axioms [37, 54, 8]. In contrast, consumers' use cases tend to target more holistic, less technically well-defined end-objectives, e.g., practical goals surrounding improving, understanding, and trusting ML systems. This inherent goal mismatch makes it difficult to pair consumer needs with appropriate methods [5, 11], or, in other words, to determine exactly how *useful* IML can be for addressing particular use cases. Resultingly, this gap is hindering the more widespread deployment of IML.

Towards characterizing and also closing this gap between research and practice, we propose a taxonomy that synthesizes multiple foundational works on IML methods and evaluations (see Section 2). The taxonomy (as shown at an abstract level in Figure 2 and discussed in more depth in Section 3) not only serves as a template for building an explicit mapping between the goals of methods and use cases, but also as a tool to unify studies of IML's usefulness in real-world settings, organizing them in a way that can assist both researchers and consumers increasingly over time.

However, as an acknowledgment of its own incompleteness, the current taxonomy emphasizes the need for all parties to work together to expand its coverage and refine connections within it (Figure 2, left). In particular, it highlights two key deficiencies in the current state of the field. First, as indicated by the sparsity of the use cases part of the taxonomy, it calls for collaborative efforts to define better-specified use cases and to group them in an adequately nuanced way. Second, the current taxonomy points out the difficulty present when attempting to narrow down potential methods for real applications. Since few studies have clearly

*Equal Contribution
†Equal Contribution
[1]The literature sometimes differentiates *interpretable* machine learning (i.e., designing models which are understandable by-design) and *explainable* machine learning (i.e., producing post-hoc explanations for a model). We do not make this distinction and instead refer to interpretability as the general goal of better understanding machine learning models through explanations. We emphasize that whether an explanation is produced by-design or by a post-hoc method does not affect *how* it should be used or evaluated but that it may affect the *quality* of the results.

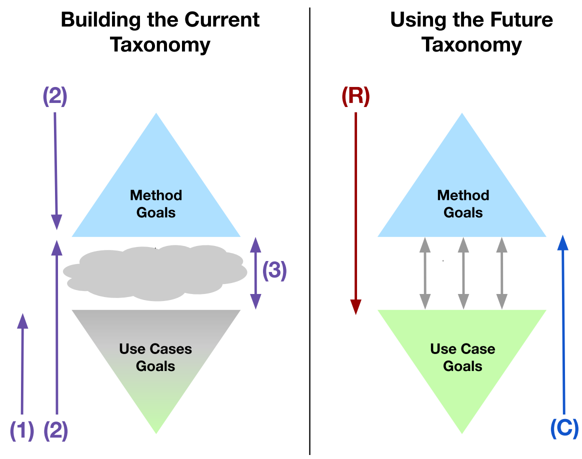

Figure 2: (Left) In this work, we focus on how researchers and consumers can work together to both establish a better use case organization (denoted by grayed area of "Use Case Goals" part of our taxonomy) and further connections through the current gap between methods and use cases (denoted by the cloud). We discuss the steps (1)-(3) to do this in Section 4.1-Section 4.3. (Right) As the two sides of the taxonomy are increasingly connected to one another, both researchers (R) and consumers (C) can begin to make use of the taxonomy to find use cases for their methods and methods for their use cases, respectively.

demonstrated what methods are useful for what kinds of use cases, the default assumption is often that all methods are potentially useful candidates for a given use case.

We discuss best practices for conducting future studies in a principled manner to begin addressing both deficiencies. More specifically, doing so requires careful considerations at each of these 3 steps of the IML workflow in the context of our taxonomy (Figure 2, left):

(1) *Problem Definition*, where researchers work with consumers to define a well-specified *target use case*, in what we call the *consumer-researcher handshake*. This involves navigating and extending the use cases portion of the taxonomy.

(2) *Method Selection*, where researchers and consumers identify potential IML methods for a target use case by navigating the methods part of the taxonomy *and/or* leveraging previously established connections between similar use cases and methods.

(3) *Method Evaluation*, where researchers work with consumers to test whether selected methods can meet target use cases. These connections (or lack thereof) can be formalized by refining the taxonomy, which will improve method selection (step 2 above) for subsequent use cases.

We believe that the IML community needs to perform such studies to populate the taxonomy for all real-world use cases. An increasingly complete version of the taxonomy would then (a) allow consumers to more directly find methods that have been demonstrated to be good for their use cases and (b) help researchers to better ground their technical work in real applications (Figure 2, right).

Our main contributions are outlined as follows.

- In Section 3, we build upon several important works in IML to propose a general-purpose taxonomy for defining and connecting its methods and use cases. This taxonomy presents a snapshot of where the field currently stands as well as the important gaps that remain, namely an inadequate use case organization and lack of explicit method-to-use-case connections.
- In Section 4, towards closing these gaps, we define a 3-step workflow for how researchers and consumers can produce more principled studies for IML on real use cases. By contextualizing important past and new ideas in terms of our taxonomy, we outline "API-level" desiderata and some best practices for all steps. We note our most novel recommendations are for Step (2), where we outline how our taxonomy can be used to guide method selection and highlight oft- overlooked details regarding (im)proper method setup and usage.
- To clarify and contextualize the proposed workflow, we also present a hypothetical running example focused on the common IML use case of model debugging, i.e., to help ensure that a model is correct, fair, or legal.[2]

2 Related Work

Many previous works have laid foundations towards critically analyzing and better organizing the field of IML. We now overview related papers based on their main goals and discuss how our work builds upon their positions and observations. As a whole, we see our work as synthesizing these separate threads to build a unifying taxonomy and to establish a workflow for more-principled future studies.

Disconnects between IML use cases and methods: Several previous works have called into question the current usages of IML methods for real use cases. [35] made such a critique at a conceptual level, observing that the stated motivations of IML are both highly variable and potentially discordant with proposed methods. Such gaps have been further crystallized through specific user perspectives [26, 11, 29]. These works all conduct interviews with IML consumers in order to describe their current use cases and needs. They emphasize the importance of understanding human perspectives for researchers in the field to make more applicable progress, a principle which is also shared as the main focus of [39]. More recently, similar gaps between methods and use cases have been identified in a public policy setting [5].

A further set of works explores negative consequences that arise from the disconnect between use cases and technical objectives in specific settings. For instance, [10] finds that counterfactual explanations are not always useful when presented to end-recipients of ML decisions, pointing out

[2]Model debugging is chosen as it is arguably the most common [26, 11] and most well-grounded set of consumer use cases. Moreover, it is a natural starting point due to the versatile nature of its assumed consumer, data scientists, who have both substantial ML knowledge and domain expertise.

that their ability to be successfully acted upon relies on several key hidden assumptions. [9] conducts a human study on IML for AI-human teams, demonstrating that the use of popular explanations can actually lead to over-trust in ML decisions and without necessarily improving upon simpler techniques such as displaying model confidence. Finally, [29] questions the effectiveness of existing explanation methods for model debugging, finding that several data scientists were not able to accurately describe the intended purposes of the popular explanations used in their experiment, while others simply trusted methods on the basis of their popularity and open-source availability.

Overall, these works provide the core motivation for our paper, which is that IML methods and use cases lack a unifying framework and effective evaluation process. Our proposed taxonomy takes a step towards bridging these gaps.

IML Use Case and Methods are not Monolithic: Towards addressing its own critiques of IML, [35] enumerates several desiderata (e.g., trust, causality, transferability) and notions of interpretability present in the IML literature. One central claim of this work is that interpretability is often treated as a monolithic concept, leading to ill-defined research claims. Our position is that this is an issue that still persists in the field. We aim to further alleviate this problem by proposing a taxonomy to both encourage properly defining both use cases and methods and also to connect them to one another.

Numerous papers have surveyed IML to better contextualize the diverse work in the field [24, 22, 40, 41, 7]. One approach that aims to provide a more actionable organization moving forward is [53], which introduces a fact sheet for researchers to fill in when developing new methods as a way to systematize comparisons of various explanation methods. In our work, we propose the use of similar categorizations of methods to prescribe best practices for more fundamental aspects of the IML workflow. Specifically, one of our contributions is to clarify how different types of methods can be compared to one another in a consistent manner.

More Principled Evaluations: Cautionary findings [1, 33, 4] about the lack of stability and robustness of explanation methods call into question their usefulness in practical settings, motivating the need for a more solid foundation for evaluating them. Further, [58, 28] have discussed the importance of separating nuanced dimensions of evaluation which are often conflated, i.e. faithfulness, plausibility / persuasiveness / usefulness, and generalizability.

Finally, [19] builds towards more principled matchings between research claims and evaluation strategies. They propose a taxonomy of usefulness evaluations based upon the nature of human-involvement and type of tasks used, creating the categories of application-, human-, and functionally-grounded evaluations. In our work, we discuss best practices for the entire IML workflow, of which evaluations is the last step. Further, we extend their taxonomy to consider separating evaluations of faithfulness to the model and usefulness for the target use case.

3 Taxonomy Overview

IML is far from a monolithic concept, so a key challenge is to identify and reconcile the many possible *method* goals and *use case* goals that one might encounter. We propose a taxonomy of these various goal statements, based on current practices and discourse, that organizes methods at the top-end and use case goals at the bottom-end (Figure 3). In between, we use a "cloud" to illustrate the overall lack of well-studied connections between these two sides.

We now describe both ends in greater depth individually, before discussing how the taxonomy can guide studies establishing connections between them in Section 4.

3.1 Method Goals

IML methods come in many forms, each with the goal of providing a specific *insight* into a given model. The top-end of our taxonomy aims to differentiate between these various perspectives based on three factors commonly discussed in existing literature [7, 24, 19].

1. *Explanation representation*. Model explanations are typically given in terms of either *feature relationships* between inputs and outputs or *training examples*.
2. *Type of feature relationships*. In the context of explanations based on feature relationships, there are three distinct approaches for explaining different aspects of the model's reasoning: *approximation*, *feature attribution*, and *counterfactual*. As a note, due to there being less focus from the IML community on training examples-based explanations, we consider one main grouping along that branch, *sample importance* explanations.
3. *Explanation scale*. Explanations vary in terms of the scale of the desired insights, either in terms of how *local* (i.e. for an individual instance) or how *global* (i.e. for a well defined region of the input space) their scope aims to be.

As branches in the method goals part of the taxonomy are followed downwards along these three factors, one encounters progressively more well-specified goal statements, and eventually reaches what we call *technical objectives* (TOs). TOs are classes of goals which are precise enough to be generally linked to a *method cluster* that most directly addresses them. In total, our taxonomy centers around 8 TOs/method clusters which organize a large portion of the goals of existing IML methods. We note a few important nuances regarding our characterization of TOs.

First, although TOs and method cluster are bijective in our proposed taxonomy, it is important to explicitly distinguish these two concepts because of the potential for *cross-cluster adaptation*. This notion arises because, although a method can be assigned to a method cluster by determining which TO it most closely optimizes for, it is frequently possible for that method to, in an ad-hoc fashion, be adapted to address a different TO. In Section 4.2, we present specific examples, discussing the assumptions made when performing such adaptations and how they should be carried out.

Second, we emphasize that each TO should be thought of as defining a *class* of related goals. Indeed, for a given TO, we hypothesize some of the key *technical detail(s)* that must

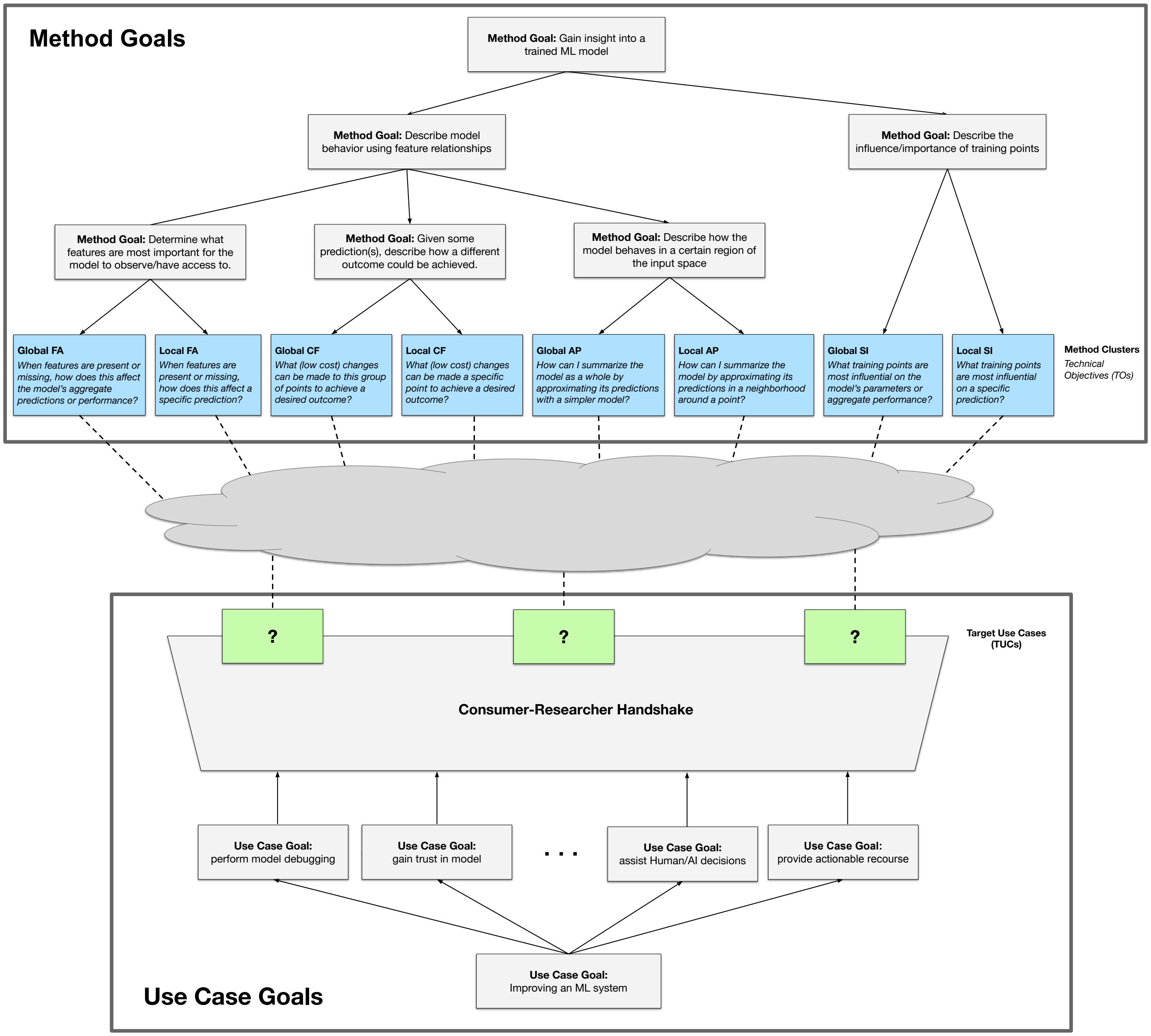

Figure 3: Our taxonomy consists of a hierarchical organization of both method goals and use case goals, which are connected in the middle via a cloud that signifies the lack of connections between technical objectives (blue) and target use cases (green). The goal is for researchers and consumers to conduct principled studies to both refine the current organization of use cases (i.e. to define well-specified targets) and to establish explicit connections between the two sides. (Note: FA = feature attribution, CF = counterfactual, AP = approximation, and SI = sample importance)

be considered towards fully parametrizing meaningfully different instantiations of the same broader goal. For example, a fully specified local approximation objective requires the identification of both *what form of simple approximation is desired* and *where the approximation is supposed to hold*. These important technical details, taken together with the TO, allow one to then define individual *proxy metrics* that reflect the desired properties of one's explanations. Proxy metrics can then serve as tractable objective functions for individual methods to optimize and also as measures to evaluate how well any method addresses a particular instantiation of the TO (see Section 4.3 for more details).

Method Clusters We next discuss the method clusters corresponding to the various TOs. Due to the overlaps in content, we organize our discussion by grouping together local and global versions of the same general method type. In each grouping, we first highlight aspects that are common to both local and global clusters before separately detailing specific examples of existing work in each. Additionally, we give examples of the technical detail(s) that are required to specify individual objectives within each TO and examples of associated proxy metrics.

Feature attribution methods address the question of how features present or missing in the input(s) to a model affect the model's prediction(s), that is how "important" each feature is to the model's prediction(s). Often, measures of importance are defined based on how the model's prediction(s) change relative to its prediction for some baseline input. The baseline input is sometimes implicit and is typically domain specific (e.g. all black pixels for grayscale images or the mean input in tabular data). Thus, the technical details here are both the precise *notion of "importance"* as well as the choice of the *baseline input*.

Local feature attribution methods like SHAP [37] attribute the change in the conditional expectation of the model output conditioned on the features of interest, with respect to an explicit baseline input. Therefore the output explanation differs significantly based on the baseline input chosen. Other methods such as Grad-CAM [51] and Integrated Gradients [54], treat gradients and their variants as the importance values, and thus are usually restricted to deep neural networks. The latter, like SHAP, also requires carefully choosing an explicit baseline input. Separately, a family of methods, such as L2X [16], use mutual information between the features and labels to learn the importance values. Relevant proxy metrics typically measure how much the model prediction changes for different types of perturbations applied to the individual (or the training data) according to the "importance" values as computed by each method [8, 4, 6, 27].

Global feature attribution methods include traditional feature selection approaches from classical statistics [14] or model specific approaches tailored to specific classes such as decision trees and tree ensembles [20, 12]. Because these approaches are either computationally expensive or model-specific, more recent methods focus on aggregating local feature attributions, such as how [57] estimates the Shapley-based metric, SPVIM. Proxy metrics might integrate over the domain at the individual-level to derive a global-level measure.

Counterfactual explanations identify a "low cost" modification that can be applied to data point(s) to get a different prediction. The most common technical detail is the specific measure of *cost* and the most common proxy metric is how often the counterfactual changes the model's prediction(s).

Local Counterfactual methods includes POLARIS [60], which finds stable counterfactual points (i.e. where the larger region around it also has a different prediction). Meanwhile, FACE [46] tries to find a counterfactual that is on the data distribution and is thus realistic to change into, which might be an important requirement for real-world applications (i.e. the feature that represents a person's "income" cannot be easily doubled for a potentially better mortgage rate if that person does not have the capacity to do so). As such, another set of proxy metrics typically tries to capture how real or feasible the proposed changes are with the amount of cost incurred for individual instances, as well as the distribution of these costs over different sub-groups of the data [55, 42]. Other local counterfactual works include [18, 23, 15].

Global counterfactual explanations finds a modification that can be applied to a whole group of points. For example, ELDR [45] identifies which features (genes in its original medical use case) differentiate different clusters of data (cell types), and AReS [47] aims to do this to detect model bias. One important proxy metric to consider for global methods is coverage [45], which measures the degree to which the explanations capture all of the differences between different cluster of points.

Approximation methods aim to use a simple function to approximate the model's behavior as accurately as possible in a region, either locally around a data point or globally around as many points as possible or across a specific region of the input space. These methods require the technical detail of both what that *region* is and what the simple function's *model family* is.

Local approximation methods are most well known by its canonical method, LIME [48], which weights data points drawn uniformly from an interpretable feature representation using their similarity to the point being explained. Other methods such as MAPLE [44] leverage the structure of the underlying data distribution to generate local approximations. One canonical proxy metric is local fidelity [44, 34], which measures how well the approximation method predicts within a certain neighborhood of data points.

Global approximation methods include distillation [21], which leverage the more intuitive representation of models such as shallow decision trees to approximate a more complex model's decision process. Another method, Generalized additive model (GAM), and its variant (GA2M) [36] benefit from being able to represent a prediction in terms of univariate features and pairwise interactions. Finally, a third model type includes falling rule lists [56], decision sets [32], and anchors [49], which create lists of if-then rules on the features that best replicate the model's decision process. A canonical proxy metric is coverage [49], which in this context measures how many data points are covered by the explanation.

Sample importance methods aim to understand how either model's prediction on an individual point or the model as a whole is impacted by changes in the training data. Technical details differ from method to method, so currently it is difficult to identify a uniform axis of variation. These methods can be evaluated with proxy metrics that represent the usefulness of the provided explanations, through simulated experiments of finding corrupted data points [59], detecting points responsible data distribution shifts [31], and recovering high accuracy with the samples considered important [30].

Local sample importance methods include influence functions [17, 31], which compute the effect of removing or perturbing a training point on the resulting model's loss for a particular test point. Meanwhile, representer point selection [59] decomposes the model prediction value on the test point in terms of the neural network activations of each training sample, computing a similar notion of influence but in a different manner. Such methods have been shown to be effective for dataset debugging [59] and detecting vulnerable examples for dataset poisoning [31].

Global sample importance methods, on the other hand, compute the effect of removing or perturbing a training point on the model's learned parameters and does not require the specification of a test point. Influence functions [31] do this by approximating Hessian of the loss for the training point, while representer point selection [59] explicitly decomposes the weights as the linear combination of the training point activations.

How do by-design methods fit in? While they do not have a corresponding method cluster in our taxonomy, it is important to discuss another family of explanation methods called "interpretable by-design" methods [50]. Although there have been multiple "interpretable by-design" models proposed [2, 36, 25, 3, 38], the differentiating property of these models is that the TO(s) of these approaches is intrinsically tied to the model family itself, hence the models are interpretable by design. In contrast, the post-hoc methods we reference above are aimed at explaining models that were designed agnostic of any particular TO relating to interpretability. That said, by-design methods also fit into our framework and should be viewed as a different way to answer the same TOs in our taxonomy. When by-design methods are proposed or used, they should clearly specify which TO(s) they are intending to address.

For instance, decision trees are commonly deemed interpretable by-design, but we can use our taxonomy to clarify in what sense they are interpretable. Indeed, shallow decision trees might be well-suited for addressing TOs related to global feature attribution or global approximation goals, as both give insights on impactful features and coarse simulatability. However, a decision tree would not lend itself easily to other types of insights, such as providing a simple and accurate local approximation. With smaller decision trees, the step-wise nature of the function makes it difficult to make local approximations for points near the discontinuities.

3.2 Use Case Goals

While IML literature has previously proposed multiple taxonomies for methods, fewer attempts have been made to more thoroughly explore a concrete organization of use cases. Currently, much of the discourse surrounds differentiating fairly broad goals, such as: model debugging, gaining trust of various stakeholders, providing actionable recourse, and assisting human/AI decision making [5, 35] (Figure 3).

While this represents a good start, it is of limited utility to treat each of these categories as monolithic problems for IML to solve. For one, these problems are complex and should not be assumed to be completely solvable by IML itself. Rather, IML is but one potential set of tools that must be demonstrated to be useful. Secondly, each broad goal really includes multiple separate technical problems, crossed with many possible practical settings and constraints. It is likely that a given IML method will not be equally useful across the board for all of these sub-problems and domains.

Thus, claims of practical usefulness should ideally be specified down to the level of an adequately defined *target use case* (TUC). TUCs, like TOs on the methods side, correspond to learning a specific relevant characteristic about the underlying model (e.g. a certain property or notion of model behavior). However, unlike a TO, they represent real-world problems that while evaluable, often might not be amenable to direct optimization. For example, one can set up real or simulated evaluations (see Section 4.3) to determine whether an explanation method is useful for identifying a particular kind of bug in the model (e.g. spurious positive correlations), but it is not so obvious how to optimize an explanation method that will succeed on those real or simulated evaluations. In the next section, we provide further discussion of both what a TUC should be (via a running example) and how potential connections to methods can be made.

4 Building Taxonomy

We now discuss how consumer-researcher teams can build off of and fill in the gaps of the current taxonomy. Specifically, we define an ideal workflow for them to conduct future studies, describing how the taxonomy can guide best practices for each of the three key steps: (1) Problem Definition, (2) Method Selection, and (3) Method Evaluation. This workflow applies to both teams who wish to study existing IML methods and those who are proposing new ones.

To help contextualize this discussion, we also describe a hypothetical running example focused on the application of IML towards model debugging. For this broad use case, the consumer is assumed to be a data scientist with ML knowledge. This assumption reduces the communication gap between herself and the IML researcher, allowing for the usage of many kinds of explanation methods while minimizing the role of external factors that are usually murkier to define (e.g., who is using the explanation, how it is presented to them, what they are told to do with it, and their presumed level of domain or ML expertise).

4.1 Step 1: Problem Definition

As motivated by Section 3.2, we argue that an important first step for any principled study is to define a well-specified TUC. We call this the *consumer-researcher handshake* (Figure 3), where researchers work with consumers to progressively refine the latter's real-world problems into relevant TUCs. In this process, some helpful pieces of information that should be discussed include: the data available, the ML pipeline used, the domain knowledge required to perform evaluations, etc. Ultimately, a more flushed out taxonomy will help researchers to have more concrete use cases at hand to motivate their method development and consumers to have more realistic guidance on what IML can and cannot do for them.

Running Example: *Consider a data scientist whose holistic use case for IML is to perform model debugging for an image-based object detection model. The team needs to identify a TUC that is more nuanced than "perform model debugging" by identifying exactly what the notion of "bug" is that the IML method should detect. As shown in a hypothetical version of the use cases part of the taxonomy (Figure 4), the umbrella of model debugging includes sub-problems*

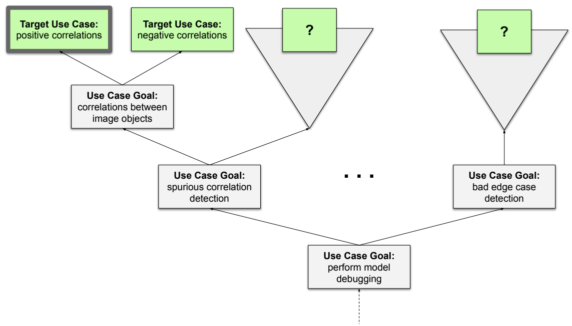

Figure 4: A hypothetical version of the use cases part of our taxonomy as produced by the consumer-researcher handshake in our running example. The identified TUC is highlighted by the box with the thicker border.

such as detecting spurious correlations and identifying bad edge case behavior. Through the consumer-researcher handshake, it arises that the data scientist is concerned the model might not be making correct decisions based on the actual target objects, but rather relying on correlated objects which also happen to be present. For example, the model might be using the presence of a person as an indicator that there is a tennis racket in the image, instead of the racket itself.

This information allows the team to navigate the portion of the taxonomy in Figure 4. Here, by considering the data scientist's concern, they first narrow the goal from model debugging to detecting spurious correlations. Then, by also taking into account the specific setting (i.e. the presence of the tennis racket at the same time as the tennis player), they are able to arrive at a further specified use case of detecting spurious correlations between two positively correlated objects. In this case, the team takes care to differentiate this from the analogous problem of detecting reliance on negatively correlated objects, reasoning that the latter is fundamentally different (i.e., it is harder to tell that the output depends on an object or not if the co-occurrences are rare in the first place).

4.2 Step 2: Method Selection

After a TUC has been properly defined, the next step is to consider which explanation methods might be appropriate.[3] There are two ways the taxonomy can be used to select methods.[4] First, researchers and consumers can, as a default, traverse the methods part of the taxonomy to hypothesize the TOs (and thus the associated method clusters) that might best align with the TUC. Doing so should rely on the researcher's best judgment in applying prior knowledge and intuition about various method types to try to narrow down the set of potential TOs. Second, the team can also navigate starting from the use cases portion, leveraging and expanding on connections established by previous studies. Naturally, if some methods have already been shown to work well on a TUC, then those (or similar) methods provide immediate baselines when studying the same (or similar) use cases.

In either case, an important –yet subtle– choice must then be made for each method: exactly how its resulting explanations should be interpreted, i.e. which TO is being addressed. As discussed in Section 3.1, a method belonging to a specific cluster may most naturally address the associated TO, but it is also possible, and indeed commonplace, to attempt *cross-cluster adaptation* for addressing other TOs. Unfortunately, while such adaptations are perhaps useful at times, they often are currently performed in an ad hoc fashion. Specifically, the differences between the technical details of each TO are often overlooked in the adaptation process, which we illustrate next via two examples.

First, one might try to use feature importance weights, via SHAP [37], as linear coefficients in a local approximation. Such an adaptation assumes that the notion of local "importance" also can reflect linear interactions with features on the desired approximation region. However, this is not necessarily guaranteed by a method like SHAP, which instead enforces a different set of game-theoretic desiderata on the importance values and may be set up to consider a quite disparate set of perturbations compared to the target approximation region.

Conversely, one can think of saliency maps via vanilla gradients [52] as an adaptation in the opposite direction. These saliency maps, a local approximation where the effective neighborhood region is extremely small, are more popularly used to address local feature attribution objectives such as to identify which parts of the image are affecting the prediction the most. However, this adaptation carries an underlying assumption that the pixels with the largest gradients are also the most "important". This approximation may not be accurate because the local shape measured by the gradient is not necessarily indicative of the model's behavior near a baseline input that is farther away.

While these are two specific examples, Table 1 provides a more general overview of necessary considerations for making cross-cluster adaptations, focusing on the side of the taxonomy containing local feature-based objectives due to their more common study and usage. In addition to the adaptations in the table, it is possible to perform more general adaptations (a) from local explanations to a global explanations by strategically aggregating the local ones or (b) from sample-based explanations to feature-based explanations by utilizing the features of the relevant sample. One can see that it is not always obvious how to fully carry out an adaptation that guarantees validity on an instantiation of a new TO.

Running example: *In this scenario, suppose that there have not been previously established results for detecting positive spurious correlations. The team follows the methods part of the taxonomy to generate hypotheses for which types*

[3] Before considering potentially more complicated explanation methods, the team should ideally also demonstrate that their TUC presents challenges to more "trivial" or conventional diagnostics. For example, [9] found model confidence to be a competitive baseline against dedicated interpretability approaches for AI-human decision making teams.

[4] If a method is being proposed, the method should be mapped to the appropriate cluster and the same selection process should follow for identifying relevant baselines.

Table 1: We discuss the considerations and limitations in cross-cluster adaptations of one method cluster to answer a TO of another cluster, specifically for local explanations that are feature attributions (FA), counterfactuals (CF), and approximations (AP).

FA → AP	It is unclear how one should map FA scores to the "parameters" in an approximation. For instance, one might attempt to use importance scores as linear coefficients, but this will not work in general when they were not meant for this purpose.
FA → CF	One could possibly adapt FA methods to do CFs, for example, by changing the most important features to their baseline input. However, it is likely that the resulting point is not very close to the original or not very realistic and, as a result, may do poorly on the "low cost" part of the CF objective.
AP → CF	One could adapt APs by computing a CF on the surrogate model, which might be easier than on the complex full model. However, there is no guarantee these CFs hold exactly on the original model given the surrogate model is an approximation.
AP → FA	One could adapt APs to derive FA scores by simply using weights derived from a surrogate model, say the coefficients of a linear approximation. Its success would depend on how close the intended baseline input of the FA is to the neighborhood region used by the approximation.
CF → FA	One could use the CF perturbation in feature space and derive FA scores by saying the features that are changed are the most important. However, this also depends on matching the intended baseline input(s) and the point(s) one generates the CF for.
CF → AP	A single CF for a single original point is likely insufficient to approximate the function for non-trivial data dimensions. However, it may be possible for one to use a diverse set of CFs for the same point.

of local explanations best suits their needs for understanding individual images. They decide against approximation based objectives, because as the inputs vary in pixel space, simple approximations are unlikely to hold or be semantically meaningful across continuous local neighborhoods. They choose feature attribution because they hypothesize that visualizing the features that the model deems most important would be useful for detecting these types of spurious correlations.

The team proposes a method in the local counterfactual method cluster that identifies the super-pixels that must change in order to flip the prediction from "tennis racket" to "no tennis racket". By "visualizing" the counterfactual explanation like a saliency map, the team performs a cross-cluster adaptation to interpret the counterfactual as a feature attribution explanation. By following the CF → FA in Table 1, they are assuming that the most changed features are also the most important to detecting the tennis racket. They reason that a feature attribution explanation would be a more intuitive format for the data scientist for this TUC. In terms of comparison, a feature attribution method that the team selects for comparison is Grad-CAM [51], which also produces a saliency map.

4.3 Step 3: Method Evaluation

Once appropriate method(s) have been chosen, the last step is to evaluate them. Evaluation is the crucial step of testing whether proposed methods can actually help address the specified TUC. However, evaluations are often carried out in manners incongruent with the properties they claim to test. One canonical mistake is that the evaluation of an explanation's *faithfulness* (i.e. ability to meet a specified TO) is often problematically conflated with the evaluation of its *usefulness* (i.e., applicability for addressing practical TUCs).

Our taxonomy addresses this mistake by mapping these evaluations to its different components: faithfulness corresponds to meeting objectives of a specific TO in the methods part and usefulness corresponds to meeting the TUC in the use case part. Next, we discuss how to perform both types of evaluation in more detail.

Faithfulness Evaluations are performed with respect to a proxy metric specified using the relevant technical details from the target TO class. For example, if the goal was to show the usefulness of an approximation-based explanation adapted as a counterfactual, the faithfulness evaluation should be with respect to a counterfactual proxy metric. Referring to the terminology from [19], these types of evaluations are called *functionally-grounded*, that is involving automated proxy tasks and no humans. While such evaluations are easiest to carry out, they come with key limitations.

In general, one should expect that a method would perform well at least on a proxy for its selected TO and, naturally, those methods which do not directly target this specific proxy will likely not perform as well. That being said, an explanation can be faultily compared as a result of unfair or biased settings of technical details. As an example, although GAMs [25] and linear models both provide local approximations, comparing these methods only in the context of fidelity ignores the fact that GAMs potentially generate more "complicated" explanations.

Further, while faithfulness evaluations can act as a first-step sanity check before running more costly usefulness evaluations, showing that a method is faithful to the model alone is not conclusive of the method's *real-world* usefulness until a direct link is established between the corresponding proxy and TUC. Once these links are ideally established, these proxies can then be used more confidently to help rule out bad set-ups before performing expensive usefulness evaluations.

Usefulness evaluations are, in contrast to faithfulness, more grounded in terms of real-world considerations and measuring success on the specified TUC is likely to rely on human domain knowledge of the task. Referring to the terminology from [19], users' perspectives can be incorporated through studies on real humans performing simplified or actual tasks (i.e. *human-grounded* or *application-grounded* evaluations respectively).

In addition to these approaches, we suggest another layer of usefulness evaluations, called *simulation evaluation*, which is an algorithmic evaluation of a simulated version of the real task defined by the domain expert. Here, the domain expert should distill what success and failure look like in the real task into a measurable quantity in the simulation (as done in [31, 43]). This type of evaluation can still be considered a usefulness evaluation because it is based on the real task. However, it is also easier and potentially more reliable to run than user studies, because the users and their decisions regarding how to use an explanation are simulated algorithmically. Overall, success on these various levels of evaluations provides evidence for establishing a connection between the method in question and the TUC.

Running example: *The team first performs respective local feature attribution faithfulness evaluations for both methods using the notions of importance that each defines. For example, for the proposed method, the team ensures that each generated explanation faithfully carries out its intended TO of identifying the effect of the presence or missingness of a super-pixel. However, good performance on any proxy metric does not conclusively imply good performance on the actual TUC, so they turn to usefulness evaluation.*

The first type of usefulness evaluation that the team conducts is a simulation evaluation, where a set of datasets is created that contains either an (artificially induced) positive correlation between a pair of objects or no such correlations. By carefully controlling the training and validation distributions, they can automatically verify whether or not a model has learned the problematic behavior she wants to detect. Then, they can define a scoring function for the explanations (i.e., how much attention they pay to the spurious object) and measure how well that score correlates with the ground truth for each candidate explanation.

Second, the team runs a human study with multiple models where they know the ground truth of which ones use spurious correlations. They score data scientists based on whether they are able to use each explanation generated by the counterfactual versus Grad-CAM to correctly identify models which use spurious correlations. If the methods are successful on the human studies, the team has demonstrated the connection between them and the TUC of detecting positively correlated objects. The team then checks to see if the proxy metrics considered earlier were actually correlated to success on the TUC, looking for evidence whether this proxy metric can be used again in future studies.

5 Conclusion

We presented a taxonomy as a way to clarify and begin bridging the gap between use cases and methods of IML. Further, we discussed best practices for how the taxonomy can be used and refined over time by researchers and consumers to better establish what methods are useful for what use cases. In illustrating this ideal workflow, we focused on a running example centered around model debugging, where the intended IML consumer is a data scientist or machine learning engineer. It would be a valuable direction moving forward to consider a broader set of variables – such as other consumers of IML, particularly those with less assumed ML knowledge – that may potentially call for new perspectives and conceptual developments.

As the taxonomy is flushed out via more studies by consumer-researcher teams, our vision is that it will be increasingly useful for both parties individually (Figure 2, right). On one hand, researchers can better ground and test their methods by using established connections to identify relevant TUCs and baselines for evaluation. On the other hand, consumers can better identify methods for their real-world applications, starting by leveraging a more complete version of the use cases part of the taxonomy. We hope that our discussions promote better practices in discovering, testing, and applying new and existing IML methods moving forward.

6 Acknowledgements

We would like to thank David Alvarez-Melis, Maruan Al-Shedivat, Kasun Amarasinghe, Wenbo Cui, Lisa Dunlap, Boyang Fu, Rayid Ghani, Hoda Heidari, Oscar Li, Zack Lipton, Adam Perer, Marco Ribiero, Kit Rodolfa, Sriram Sankararaman, Mukund Sundararajan, and Chih-Kuan Yeh for their valuable feedback. This work was supported in part by DARPA FA875017C0141, the National Science Foundation grants IIS1705121, IIS1838017 and IIS2046613, an Amazon Web Services Award, a Carnegie Bosch Institute Research Award, a Facebook Faculty Research Award, funding from Booz Allen Hamilton, and a Block Center Grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, the National Science Foundation, or any other funding agency.

References

[1] Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 9505–9515.

[2] Agarwal, R.; Frosst, N.; Zhang, X.; Caruana, R.; and Hinton, G. E. 2020. Neural additive models: Interpretable machine learning with neural nets. *arXiv:2004.13912* .

[3] Al-Shedivat, M.; Dubey, A.; and Xing, E. 2020. Contextual explanation networks. *Journal of Machine Learning Research* 21(194): 1–44.

[4] Alvarez-Melis, D.; and Jaakkola, T. S. 2018. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049* .

[5] Amarasinghe, K.; Rodolfa, K.; Lamba, H.; and Ghani, R. 2020. Explainable Machine Learning for Public Policy: Use Cases, Gaps, and Research Directions. *arXiv preprint arXiv:2010.14374* .

[6] Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2018. Towards better understanding of gradient-based

attribution methods for Deep Neural Networks. In *International Conference on Learning Representations*.

[7] Arya, V.; Bellamy, R. K.; Chen, P.-Y.; Dhurandhar, A.; Hind, M.; Hoffman, S. C.; Houde, S.; Liao, Q. V.; Luss, R.; Mojsilović, A.; et al. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012* .

[8] Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7): e0130140.

[9] Bansal, G.; Wu, T.; Zhu, J.; Fok, R.; Nushi, B.; Kamar, E.; Ribeiro, M. T.; and Weld, D. S. 2020. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. *arXiv preprint arXiv:2006.14779* .

[10] Barocas, S.; Selbst, A. D.; and Raghavan, M. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 80–89.

[11] Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M. F.; and Eckersley, P. 2020. Explainable Machine Learning in Deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, 648–657. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369367. doi:10.1145/3351095.3375624. URL https://doi.org/10.1145/3351095.3375624.

[12] Breiman, L. 2001. Random forests. *Machine learning* 45(1): 5–32.

[13] Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1721–1730.

[14] Chandrashekar, G.; and Sahin, F. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40(1): 16–28.

[15] Chang, C.-H.; Creager, E.; Goldenberg, A.; and Duvenaud, D. 2019. Explaining Image Classifiers by Counterfactual Generation. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=B1MXz20cYQ.

[16] Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *International Conference on Machine Learning*, 883–892.

[17] Cook, R. D. 1977. Detection of influential observation in linear regression. *Technometrics* 19(1): 15–18.

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

[19] Doshi-Velez, F.; and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. URL https://arxiv.org/abs/1702.08608.

[20] Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

[21] Frosst, N.; and Hinton, G. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* .

[22] Gilpin, L. H.; Bau, D.; Yuan, B. Z.; Bajwa, A.; Specter, M.; and Kagal, L. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, 80–89. IEEE.

[23] Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Counterfactual Visual Explanations. In *International Conference on Machine Learning*, 2376–2384.

[24] Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51(5): 1–42.

[25] Hastie, T. J.; and Tibshirani, R. J. 1990. *Generalized additive models*, volume 43. CRC press.

[26] Hong, S. R.; Hullman, J.; and Bertini, E. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction* 4(CSCW1): 1–26.

[27] Hooker, S.; Erhan, D.; Kindermans, P.-J.; and Kim, B. 2019. A Benchmark for Interpretability Methods in Deep Neural Networks. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2019/file/fe4b8556000d0f0cae99daa5c5c5a410-Paper.pdf.

[28] Jacovi, A.; and Goldberg, Y. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4198–4205. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.386. URL https://www.aclweb.org/anthology/2020.acl-main.386.

[29] Kaur, H.; Nori, H.; Jenkins, S.; Caruana, R.; Wallach, H.; and Wortman Vaughan, J. 2020. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14.

[30] Kim, B.; Koyejo, O.; Khanna, R.; et al. 2016. Examples are not enough, learn to criticize! Criticism for Interpretability. In *NIPS*, 2280–2288.

[31] Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning*, 1885–1894.

[32] Lakkaraju, H.; Bach, S. H.; and Leskovec, J. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1675–1684.

[33] Laugel, T.; Lesot, M.-J.; Marsala, C.; and Detyniecki, M. 2019. Issues with post-hoc counterfactual explanations: a discussion. *arXiv preprint arXiv:1906.04774* .

[34] Li, J.; Nagarajan, V.; Plumb, G.; and Talwalkar, A. 2021. A Learning Theoretic Perspective on Local Explainability. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=7aL-OtQrBWD.

[35] Lipton, Z. C. 2018. The mythos of model interpretability. *Queue* 16(3): 31–57.

[36] Lou, Y.; Caruana, R.; Gehrke, J.; and Hooker, G. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 623–631.

[37] Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 4765–4774.

[38] Melis, D. A.; and Jaakkola, T. 2018. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, 7775–7784.

[39] Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267: 1–38.

[40] Mohseni, S.; Zarei, N.; and Ragan, E. 2019. A Multidisciplinary survey and framework for design and evaluation of explainable AI systems. arXiv. *Human-Computer Interaction* .

[41] Murdoch, W. J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; and Yu, B. 2019. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592* .

[42] Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. *Proceedings of The Web Conference 2020* doi:10.1145/3366423.3380087. URL http://dx.doi.org/10.1145/3366423.3380087.

[43] Plumb, G.; Al-Shedivat, M.; Cabrera, A. A.; Perer, A.; Xing, E.; and Talwalkar, A. 2020. Regularizing Black-box Models for Improved Interpretability. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 10526–10536. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/file/770f8e448d07586afbf77bb59f698587-Paper.pdf.

[44] Plumb, G.; Molitor, D.; and Talwalkar, A. S. 2018. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, 2515–2524.

[45] Plumb, G.; Terhorst, J.; Sankararaman, S.; and Talwalkar, A. 2020. Explaining Groups of Points in Low-Dimensional Representations. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 7762–7771. PMLR. URL http://proceedings.mlr.press/v119/plumb20a.html.

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

[47] Rawal, K.; and Lakkaraju, H. 2020. Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses. *Advances in Neural Information Processing Systems* 33.

[48] Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

[49] Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-Precision Model-Agnostic Explanations.

[50] Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5): 206–215.

[51] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

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

[53] Sokol, K.; and Flach, P. 2020. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 56–67.

[54] Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*, 3319–3328.

[55] Ustun, B.; Spangher, A.; and Liu, Y. 2019. Actionable Recourse in Linear Classification. *Proceedings of the Conference on Fairness, Accountability, and Transparency* doi:10.1145/3287560.3287566. URL http://dx.doi.org/10.1145/3287560.3287566.

[56] Wang, F.; and Rudin, C. 2015. Falling rule lists. In *Artificial Intelligence and Statistics*, 1013–1022.

[57] Williamson, B.; and Feng, J. 2020. Efficient nonparametric statistical inference on population feature importance using Shapley values. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 10282–10291. PMLR. URL http://proceedings.mlr.press/v119/williamson20a.html.

[58] Yang, F.; Du, M.; and Hu, X. 2019. Evaluating Explanation Without Ground Truth in Interpretable Machine Learning. URL https://arxiv.org/abs/1907.06831.

[59] Yeh, C.-K.; Kim, J.; Yen, I. E.-H.; and Ravikumar, P. K. 2018. Representer point selection for explaining deep neural networks. In *Advances in neural information processing systems*, 9291–9301.

[60] Zhang, X.; Solar-Lezama, A.; and Singh, R. 2018. Interpreting Neural Network Judgments via Minimal, Stable, and Symbolic Corrections. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 4874–4885. Curran Associates, Inc. URL http://papers.nips.cc/paper/7736-interpreting-neural-network-judgments-via-minimal-stable-and-symbolic-corrections.pdf.