Interpretable Machine Learning: Moving From Mythos to Diagnostics

Valerie Chen [*1] **Jeffrey Li** [*2] **Joon Sik Kim** [†1] **Gregory Plumb** [†1] **Ameet Talwalkar** [1 3]

Abstract

Despite years of progress 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 stated as *consumers' use cases*. To address this gap, we argue for the IML community to embrace a *diagnostic* vision for the field. Instead of viewing IML methods as a panacea for a variety of overly broad use cases, we emphasize the need to systematically connect IML methods to narrower–yet better defined–target use cases. To formalize this vision, we propose a taxonomy including both methods and use cases, helping to conceptualize the current gaps between the two. Then, to connect these two sides, we describe a three-step workflow to enable researchers and consumers to define and validate IML methods as useful diagnostics. Eventually, by applying this workflow, a more complete version of the taxonomy will allow consumers to find relevant methods for their target use cases and researchers to identify motivating use cases for their methods.

1. Introduction

The emergence of machine learning as a society-changing technology in the last decade has triggered concerns about our inability to understand the reasoning of increasingly complex models. The field of Interpretable Machine Learning (IML)[1] grew out of these concerns, with the goal of empowering various stakeholders to tackle use cases such as building trust in models, performing model debugging, and generally informing real human-decision making (Bhatt et al., 2020; Lipton, 2018; Gilpin et al., 2018).

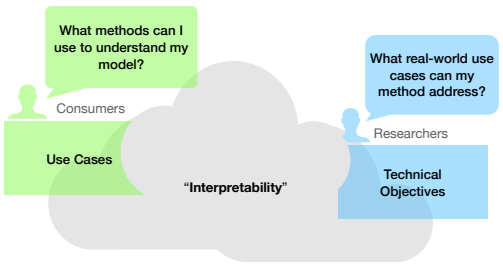

Figure 1. Currently, IML researchers focus more on technical objectives while consumers focus on use cases. Often, a lack of explicit connections remains between the two, making proper usage and development of IML methods difficult for both parties.

However, despite the flurry of IML methodological development over the last several years, a stark disconnect characterizes the current overall approach: IML methods typically optimize diverse but narrow technical objectives, yet their claimed use-cases remain broad and often underspecified. Echoing similar critiques about the field made (Lipton, 2018), it has thus remained difficult for the field to sufficiently evaluate these claims and thus to translate methodological advances into widespread practical impact.

In this paper, we outline a path forward for the ML community to address this disconnect and foster more widespread adoption, focusing on two key principles:

1. Embrace a "diagnostic" vision for IML. Instead of aiming to provide complete solutions for ill-defined problems such as "debugging" and "trust", we argue that the field of IML should focus on the important, if less grandiose, goal of developing a suite of rigorously-tested diagnostic tools. In treating IML methods as future diagnostics, we view each as providing a targeted, well-specified insight into a model's behavior. In this sense, these methods should then be used alongside and in a manner similar to more classical statistical diagnostics (e.g., error bars, hypothesis tests, methods for outlier detection), for which clearer guidelines exist for when and how to apply them.[2]

[*,†]Equal contribution [1]Carnegie Mellon University [2]University of Washington [3]Determined AI. Correspondence to: Valerie Chen <valeriechen@cmu.edu>, Jeffrey Li <jwl2162@cs.washington.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]The literature sometimes differentiates *interpretable* ML (i.e., designing models which are understandable by-design) and *explainable* ML (i.e., producing post-hoc explanations for models) (Rudin, 2019). 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 (though it may affect the quality of the results). Thus, we see this distinction as orthogonal to our paper.

[2]Under this vision, we treat existing IML methods as *potential* diagnostics, emphasizing their need to be more rigorously-tested.

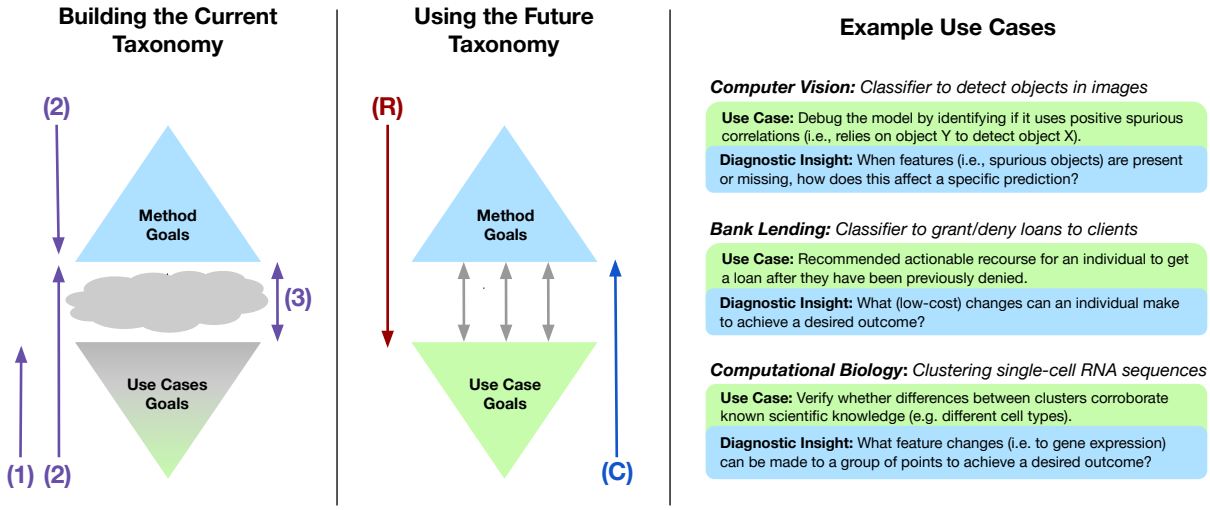

Figure 2. (Left) We focus on how researchers and consumers can work together to both establish a better use case organization (i.e., "Use Case Goals") and further connections through the current gap between methods and use cases (i.e., the cloud) by following steps (1)-(3) in our proposed workflow. (Middle) As the two sides are increasingly connected to one another, researchers (R) and consumers (C) can make use of the taxonomy to find use cases for their methods and methods for their use cases, respectively. (Right) We highlight how three different potential diagnostics may provide useful insights for three use cases. In fact, the desired diagnostic information in each use case maps to a different Technical Objective (local feature attribution, local counterfactual, and global counterfactual, respectively) in our taxonomy (Figure 3). When we later discuss a more concrete workflow for filling in the taxonomy we expand on the computer vision setting as a running example.

2. Rigorously evaluate and establish potential IML diagnostics. Currently, IML researchers typically develop and evaluate methods by focusing on quantifiable technical objectives, e.g., maximizing various notions of faithfulness or adherence to some desirable axioms (Lundberg & Lee, 2017; Sundararajan et al., 2017; Bach et al., 2015). However, while these IML methods generally target seemingly relevant aspects of a model's behavior, it is imperative to carefully measure their effectiveness on concrete use cases in order to demonstrate their utility as practical diagnostics.

Motivated by these principles, we first illustrate our diagnostic vision via an incomplete taxonomy that synthesizes foundational works on IML methods and evaluation. 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 potential IML diagnostics and specific use cases, but also as a tool to unify studies of IML's usefulness in real-world settings (concrete examples shown in Figure 2, right).

However, the incompleteness of the current taxonomy emphasizes the need for researchers and consumers to work together to expand its coverage and refine connections within it. More specifically, doing so requires careful considerations at each of these 3 steps of the IML workflow in the context of our taxonomy as shown in Figure 2 (left):

(1) *Problem Definition*, where researchers work with consumers to define a well-specified *target use case* (TUC).

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

(3) *Method Evaluation*, where they test whether selected methods can meet TUCs.

In Section 4, we provide an extensive discussion about best practices for this IML workflow to flesh out this taxonomy and deliver rigorously-tested diagnostics to consumers. Ultimately, we envision an increasingly complete taxonomy that (i) allows consumers to find suitable IML methods for their use cases; and (ii) helps researchers to ground their technical work in real applications (Figure 2, middle).

2. Background

An increasingly diverse set of methods has been recently proposed and broadly classified as part of IML. However, multiple concerns have been expressed in light of this rapid development, focused on IML's underlying foundations and the gap between research and practice.

Critiques of the field's foundations: (Lipton, 2018) provided an early critique, highlighting that the stated moti-

vations of IML were both highly variable, and potentially discordant with proposed methods. (Krishnan, 2019) added to these arguments from a philosophical angle, positing that interpretability as a unifying concept is both unclear and of questionable usefulness. Instead, as they argue, more focus should be placed on the actual end-goals, for which IML is one possible solution.

Gaps between research and practice: Multiple works have also highlighted important gaps between existing methods and their practical usefulness. Some have demonstrated a lack of stability/robustness of popular approaches (Adebayo et al., 2018; Laugel et al., 2019; Alvarez-Melis & Jaakkola, 2018). Others discuss how common IML methods can fail to help humans in the real-world, both through pointing out hidden assumptions and dangers (Barocas et al., 2020; Rudin, 2019) as well as conducting case-studies with users (Bansal et al., 2020; Kaur et al., 2020).

More recently, many review papers (Gilpin et al., 2018; Mohseni et al., 2019; Murdoch et al., 2019; Arya et al., 2019) have attempted to clean up and organize aspects of IML, but largely do not address these issues head-on. In contrast, our proposed re-framing of IML methods as diagnostic tools follows naturally from these concerns. Notably we embrace the seeming shortcomings of IML methods as providing merely "facts" (Krishnan, 2019) or "summary statistics" (Rudin, 2019) about a model, and instead focus on the practical questions of when and how these methods can be practically useful.

3. A Diagnostic Vision for IML

We think of a diagnostic as a tool that provides some insight about a model. As an analogy, consider the suite of diagnostic tools at a doctor's disposal that similarly provides insight about a patient. An x-ray could be useful for identifying bone fractures while a heart rate monitor would be helpful for identifying an irregular heart rhythm. Importantly, neither tool enables the doctor to broadly "understand" a person's health. However, each can be useful *if applied properly to a well-scoped problem*. Similarly, rigorously establishing connections between IML methods and well-defined use cases is imperative for the IML community.

To begin such a pursuit, we start by first identifying and reconciling the many method goals and use case goals that one might encounter currently. Based on contemporary practices and discourse, we propose a taxonomy that organizes separately the method goals at the top-end and use case goals at the bottom-end (Figure 3). While our diagnostic vision for the field ideally involves a robust set of connections between these two sides, we use a "cloud" to illustrate the current overall lack of well-established diagnostics.

3.1. Method Goals

Each IML method provides a specific type of insight into a given model. Based on these types of insights, we first provide a hierarchical organization which divides the set of existing IML methods into 8 method clusters. In the diagnostic vision, we think of each method cluster broadly as a class of diagnostics that addresses a Technical Objective (TO). Then, we describe in more detail each TO in a way that allows one to specify individual method objectives.

3.1.1. Hierarchical Organization

The top-end of our taxonomy aims to differentiate between the various perspectives explanations provide based on three factors commonly discussed in existing literature (Arya et al., 2019; Guidotti et al., 2018; Doshi-Velez & Kim, 2017). We discuss these further in Appendix A.

At the leaf nodes are *technical objectives* (TOs), classes of goals that are precise enough to be generally linked to a *method cluster* that most directly addresses them. In total, there are 8 TOs/method clusters which captures a large portion of the goals of current 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 it is frequently possible for that method to, in an ad-hoc fashion, be adapted to address a different TO.

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 be considered towards fully parametrizing meaningfully different instantiations of the same broader goal. These important technical details, taken together with the TO, allow one to 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, as well as measures of how well any method addresses a particular instantiation of the TO.

3.1.2. Technical Objectives

We next overview the TOs (and their technical details) that correspond to various method clusters. Due to the overlaps in content, we group together local and global versions of the same general method type/objective (for more extensive details and examples of specific methods for each, see Appendix B).

Feature attribution address when features are present (or missing), how does this affect the the model's prediction(s) (i.e. how "important" each feature is to the model's predic-

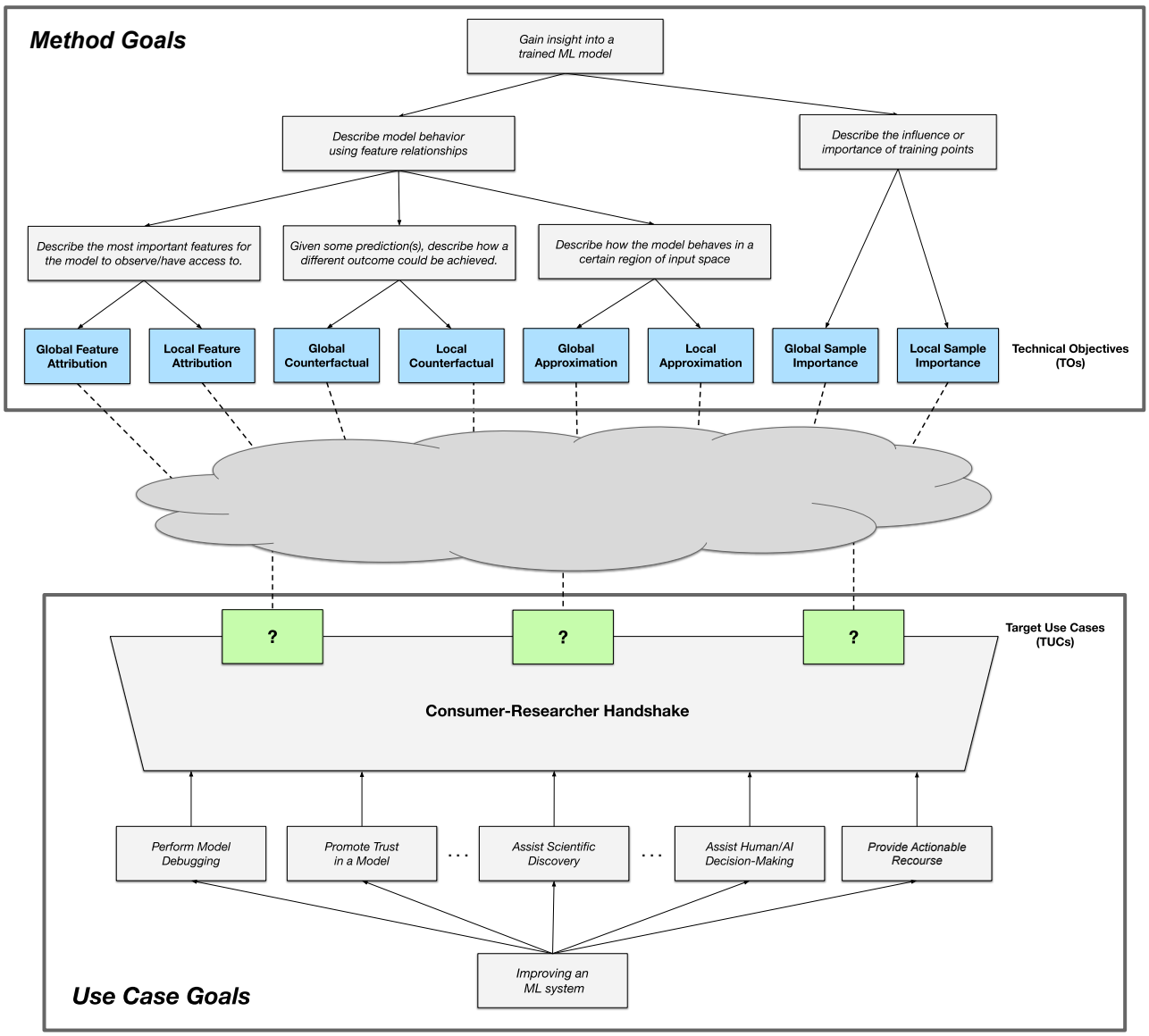

Figure 3. Our taxonomy consists of a hierarchical organization of both existing method and use case goals. Moving forwards, the goal for researchers and consumers is to conduct principled studies to both refine the current organization of use cases by defining more well-specified target use cases (green) and to establish explicit connections between these targets and technical objectives (blue).

tion(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 domain specific (e.g., all black pixels for grayscale images or the mean input in tabular data). Thus, the technical details are both the precise *notion of "importance"* and the choice of the *baseline input*. 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.

Counterfactual explanations address what "low cost" modification can be applied to data point(s) to achieve a desired 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).

Approximation methods address how can one summarize the model by approximating its predictions in a region, either locally around a data point, 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. For local approximation, a canonical metric is local fidelity, which measures how well the method predicts within a certain neighborhood of a data point. For global approximation, a proxy metric is coverage, which measures how many data points the explanation applies to.

Sample importance methods address what training points are influential on a model's prediction for either an individual point or the model as a whole. 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, detecting points responsible data distribution shifts, and recovering high accuracy with the samples considered important.

3.1.3. 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 IML methods called "interpretable by-design" methods (Rudin, 2019). The differentiating property of these models from the post-hoc methods that we reference above is that the TO(s) of these approaches is intrinsically tied to the model family itself, hence the models are interpretable by design. 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.

3.2. Use Case Goals

Currently, much of the discourse on IML use cases surrounds differentiating fairly broad goals, such as model debugging, gaining trust of various stakeholders, providing actionable recourse, assisting in scientific/causal discovery, and aiding Human/AI teams (Bhatt et al., 2020; Lipton, 2018; Gilpin et al., 2018) (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. That is, to show that an IML method is an effective diagnostic, specific use cases must be identified and demonstrated (Krishnan, 2019). 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 IML 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 IML method that will succeed on those real or simulated evaluations.

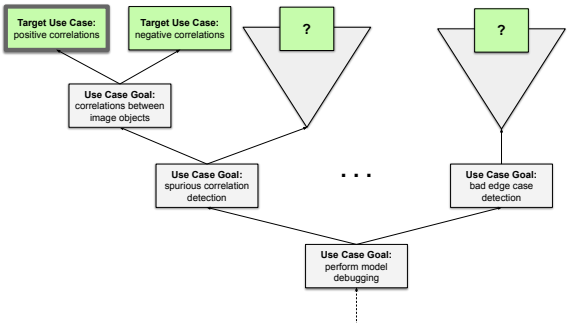

Figure 4. A hypothetical version of the use cases part of our taxonomy as produced by the consumer-researcher handshake in our running example. The identified TUC is highlighted by the box with the thicker border.

4. A Workflow for Establishing Diagnostics

We now turn to how a diagnostic vision for IML can be more fully realized, discussing how methods can be established as diagnostics, thus filling gaps in the existing taxonomy. Specifically, we define an ideal workflow for consumer-researcher teams to conduct future studies about IML methods 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 provide a running example that builds on the Computer Vision model debugging example from Figure 2 (right). Model debugging is not only a common (Hong et al., 2020; Bhatt et al., 2020), but also well-grounded consumer use case. It is a natural starting point due to the versatile nature of its assumed consumer, data scientists, typically has both substantial ML knowledge and domain expertise, minimizing the communication gap between the data scientist and the IML researcher.

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 who wants to debug their image-based object detection model. The team needs to identify a TUC that is more specific 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 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. By considering the data scientist's concern, they first narrow the goal from model debugging to detecting spurious correlations (SCs). 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 SCs 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 IML methods might be appropriate. This does assume that IML methods are necessary, that is the team should have demonstrated that the TUC presents challenges to more "trivial" or conventional diagnostics. For example, (Bansal et al., 2020) found model confidence to be a competitive baseline against dedicated interpretability approaches for AI-human decision making teams.

If non-IML diagnostics are unsuccessful, there are two ways the taxonomy can be used to select methods. First, researchers and consumers can, as a default, traverse the methods part of the taxonomy to hypothesize the TOs (and thus respective 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. If a method is being proposed, the method should be mapped to the appropriate method cluster and the same selection process should follow. Second, the team can also navigate starting from the use cases part, 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 are often 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 (and in more depth in Appendix B.1).

First, one might try to use feature importance weights, via SHAP (Lundberg & Lee, 2017), 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 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 (Simonyan et al., 2013) 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

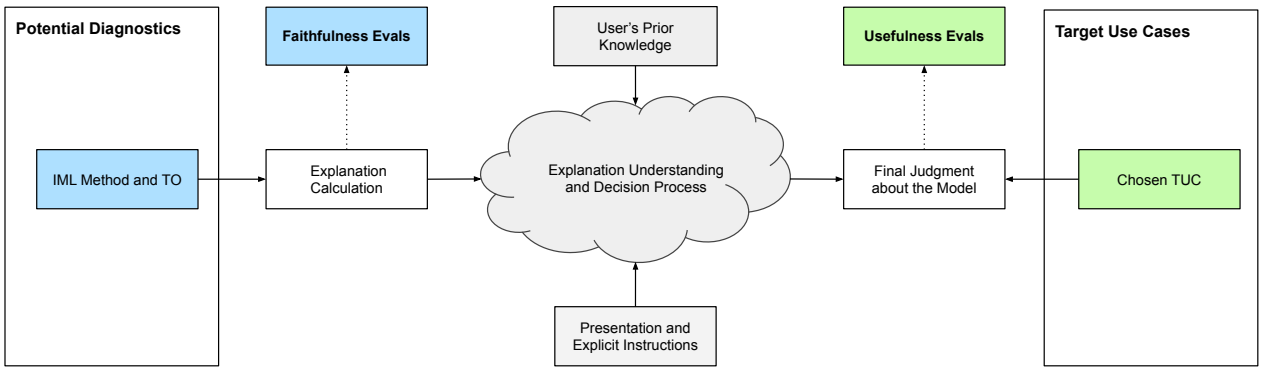

Figure 5. An overview of where different types of IML evaluations (faithfulness and usefulness) fall within the overall pipeline of IML applications. We highlight differences in goals of these evaluations as well as the various moving components that affect them. Colored boxes denote components that need to be defined (IML method and TO, Chosen TUC), while gray boxes to denote components that require more careful study (Explanation Understanding and Decision Process).

measured by the gradient is not necessarily indicative of the model's behavior near a baseline input that is farther away.

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

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. To do so, 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 (Selvaraju et al., 2017), 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 common 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). While both may play important roles, as we discuss further in this section, they target fundamentally different claims.

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 cases part. Next, using Figure 5 as a guide, we clarify differences between these two types of evaluations and how they can be carried out.

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 (Doshi-Velez & Kim, 2017), 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. An explanation can

also be faultily compared as a result of unfair or biased settings of technical details. As an example, although GAMs (Hastie & Tibshirani, 1990) 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 established, these proxies can then be used more confidently to help rule out bad set-ups before performing expensive usefulness evaluations.

Usefulness evaluations, in contrast to faithfulness, measure a user's success in applying explanations on the specified TUC. Since they are ultimately an evaluation of what one *does* with an explanation, usefulness depends crucially on factors such as users' prior knowledge, such as domain and ML/IML experience. Referring to the terminology from (Doshi-Velez & Kim, 2017), 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 particular, to successfully utilize explanations in practice, we would need to study how this process might differ depending on the presentation of the explanation and explicit instructions that are provided.

As highlighted by the cloud in Figure 5, how exactly users translate explanation calculations (in their minds) to their final judgments remains murky. This motivates further research relating to better understanding *what users understand explanations to tell them* and *how they act upon these understandings*. Then, when establishing new diagnostics, these assumptions/limitations should be clearly spelled out for when researchers use the method in a future study and when the consumers deploy the method.

Motivated by these challenges, we suggest researchers consider another type of usefulness evaluation called *simulation evaluation*. Simulation evaluation is an algorithmic evaluation on a simulated version of the real task where success and failure is distilled by a domain expert into a measurable quantity (as illustrated in the running example). This type of evaluation is still based on the real task, but is easier and potentially more reliable to run than user studies. By simulating the users and their decision-making process algorithmically, thus controlling some noisier aspects of usefulness evaluation, researchers may be able to better understand why their methods are "failing". Is it because of the algorithm itself, or the actual decision process users take?

Overall, success on these various levels of evaluations provides evidence for establishing a connection between the method in question and the TUC. Specifically, the team should check to see if the proxy metrics considered earlier were correlated to success on the TUC. If so, this would provide evidence for whether the proxy metrics considered should be used again in future studies, connecting faithfulness and usefulness evaluations.

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 team first conducts 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 they want 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 explanation.

Second, the team runs a human study with multiple models where they know the ground truth of which ones use SCs. 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 SCs. 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.

5. Conclusion

Towards a diagnostic vision for IML, we presented a taxonomy as a way to clarify and begin bridging the gap between methods and use cases. 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. 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, middle). 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 Ribeiro, 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

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

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

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

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. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

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

Bansal, G., Wu, T., Zhu, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., and Weld, D. S. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. *arXiv preprint arXiv:2006.14779*, 2020.

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

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

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

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

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892, 2018.

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

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning, 2017. URL https://arxiv.org/abs/1702.08608.

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

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

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

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

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

Hong, S. R., Hullman, J., and Bertini, E. Human factors in model interpretability: Industry practices, challenges, and needs. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26, 2020.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. 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., 2019. URL https://proceedings.neurips.cc/paper/2019/file/fe4b8556000d0f0cae99daa5c5c5a410-Paper.pdf.

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., and Wortman Vaughan, J. 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*, pp. 1–14, 2020.

Kim, B., Koyejo, O., Khanna, R., et al. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, pp. 2280–2288, 2016.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894, 2017.

Krishnan, M. Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology*, pp. 1–16, 2019.

Lakkaraju, H., Bach, S. H., and Leskovec, J. 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*, pp. 1675–1684, 2016.

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

Li, J., Nagarajan, V., Plumb, G., and Talwalkar, A. A learning theoretic perspective on local explainability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7aL-OtQrBWD.

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

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

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

Mohseni, S., Zarei, N., and Ragan, E. A multidisciplinary survey and framework for design and evaluation of explainable ai systems. arxiv. *Human-Computer Interaction*, 2019.

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

Pawelczyk, M., Broelemann, K., and Kasneci, G. Learning model-agnostic counterfactual explanations for tabular data. *Proceedings of The Web Conference 2020*, Apr 2020. doi: 10.1145/3366423.3380087. URL http://dx.doi.org/10.1145/3366423.3380087.

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

Plumb, G., Terhorst, J., Sankararaman, S., and Talwalkar, A. 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*, pp. 7762–7771. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/plumb20a.html.

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

Rawal, K. and Lakkaraju, H. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems*, 33, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. " 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*, pp. 1135–1144, 2016.

Ribeiro, M. T., Singh, S., and Guestrin, C. Anchors: High-precision model-agnostic explanations. 2018.

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

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

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

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328, 2017.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, Jan 2019. doi: 10.1145/3287560.3287566. URL http://dx.doi.org/10.1145/3287560.3287566.

Wang, F. and Rudin, C. Falling rule lists. In *Artificial Intelligence and Statistics*, pp. 1013–1022, 2015.

Williamson, B. and Feng, J. 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*, pp. 10282–10291. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/williamson20a.html.

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

Zhang, X., Solar-Lezama, A., and Singh, R. 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*, pp. 4874–4885. Curran Associates, Inc., 2018.

A. Taxonomy of Method Goals

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: *feature attribution*, *counterfactual*, and *approximation*. As a note, due to there being less focus from the IML community on training example-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, with their scope ranging from how *local* (i.e. for an individual instance) to *global* (i.e. for a well defined region of the input space).

B. Method Cluster Details

Feature attribution methods address the question of how features present (or missing) in the input(s) affect the model's prediction(s) (i.e. 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 (Lundberg & Lee, 2017) 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 (Selvaraju et al., 2017) and Integrated Gradients (Sundararajan et al., 2017), 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 (Chen et al., 2018), 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 (Bach et al., 2015; Alvarez-Melis & Jaakkola, 2018; Ancona et al., 2018; Hooker et al., 2019).

Global feature attribution methods include traditional feature selection approaches from classical statistics (Chandrashekar & Sahin, 2014) or model specific approaches tailored to specific classes such as decision trees and tree ensembles (Friedman, 2001; Breiman, 2001). Because these approaches are either computationally expensive or model-specific, more recent methods focus on aggregating local feature attributions, such as how (Williamson & Feng, 2020) 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 (Zhang et al., 2018), which finds stable counterfactual points (i.e. where the larger region around it also has a different prediction). Meanwhile, FACE (Poyiadzi et al., 2020) 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 (Ustun et al., 2019; Pawelczyk et al., 2020).

Global counterfactual explanations finds a modification that can be applied to a whole group of points. For example, ELDR (Plumb et al., 2020) identifies which features (genes in its original medical use case) differentiate different clusters of data (cell types), and AReS (Rawal & Lakkaraju, 2020) aims to do this to detect model bias. One important proxy metric to consider for global methods is coverage (Plumb et al., 2020), 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 (Ribeiro et al., 2016), which weights data points drawn uniformly from an interpretable feature representation using their similarity to the point being explained. Other methods such as MAPLE (Plumb et al., 2018) leverage the structure of the underlying data distri-

bution to generate local approximations. One canonical proxy metric is local fidelity (Plumb et al., 2018; Li et al., 2021), which measures how well the approximation method predicts within a certain neighborhood of data points.

Global approximation methods include distillation (Frosst & Hinton, 2017), 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) (Lou et al., 2013) 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 (Wang & Rudin, 2015), decision sets (Lakkaraju et al., 2016), and Anchors (Ribeiro et al., 2018), which create lists of if-then rules on the features that best replicate the model's decision process. A canonical proxy metric is coverage (Ribeiro et al., 2018), which in this context measures how many data points the explanation applies to.

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 (Yeh et al., 2018), detecting points responsible data distribution shifts (Koh & Liang, 2017), and recovering high accuracy with the samples considered important (Kim et al., 2016).

Local sample importance methods include influence functions (Cook, 1977; Koh & Liang, 2017), 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 (Yeh et al., 2018) 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 (Yeh et al., 2018) and detecting vulnerable examples for dataset poisoning (Koh & Liang, 2017).

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 (Koh & Liang, 2017) do this by approximating the Hessian of the loss for the training point, while representer point selection (Yeh et al., 2018) explicitly decomposes the weights as the linear combination of the training point activations.

B.1. More on Cross-cluster Adaptation

Table 1. We discuss important limitations and assumptions one should consider when performing 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.
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.