Interpretable Machine Learning

MOVING FROM MYTHOS TO DIAGNOSTICS

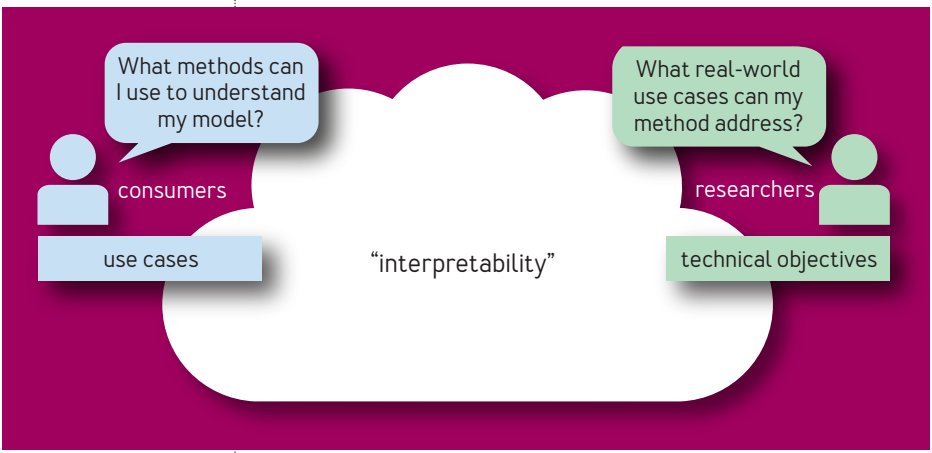
VALERIE CHEN, JEFFREY LI, JOON SIK KIM, GREGORY PLUMB, AMEET TALWALKAR

The emergence of machine learning as a society-changing technology in the past decade has triggered concerns about people's inability to understand the reasoning of increasingly complex models. The field of IML (interpretable machine learning) 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.[7,10,17]

Yet despite the flurry of IML methodological development over the past several years, a stark disconnect characterizes the current overall approach. As shown in figure 1, IML researchers develop methods that typically optimize for diverse but narrow technical objectives, yet their claimed use cases for consumers remain broad and often underspecified. Echoing similar critiques about the field,[17] it has thus remained difficult to evaluate these claims sufficiently and to translate methodological advances into widespread practical impact.

This article outlines a path forward for the ML

FIGURE 1: **THE GAP BETWEEN IML CONSUMERS AND RESEARCHERS**



community to address this disconnect and foster more widespread adoption, focusing on two key principles:

➡ *Embrace a "diagnostic" vision for IML.* Instead of aiming to provide complete solutions for ill-defined problems, such as "debugging" and "trust," the field of IML should focus on the important, if less grandiose, goal of developing a suite of rigorously tested diagnostic tools. By treating IML methods as diagnostics, each can be viewed 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), which have clearer guidelines for when and how to apply them. Under this vision, existing IML methods should be treated as *potential* diagnostics until they are rigorously tested.

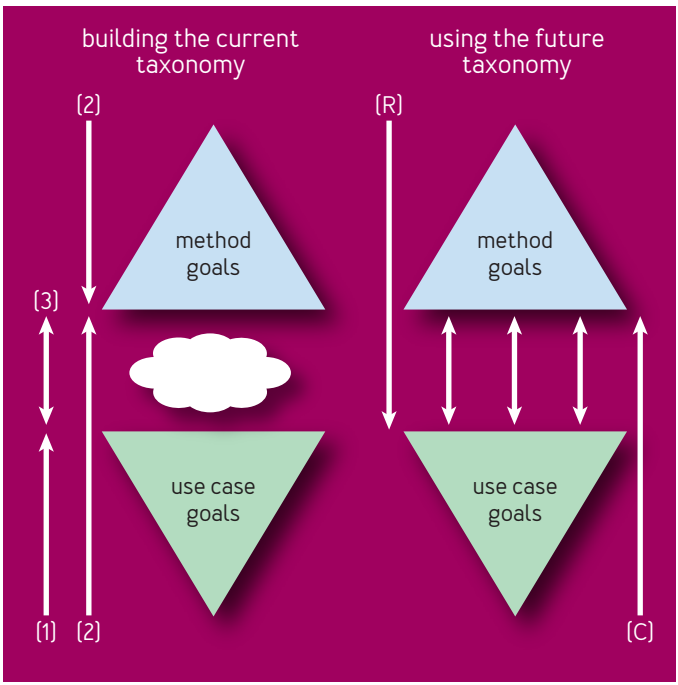
➡ *Rigorously evaluate and establish potential IML diagnostics.* IML researchers typically develop and evaluate methods by focusing on quantifiable technical objectives (e.g., maximizing various notions of faithfulness or adherence to some desirable axioms[4,18,24]). While these IML methods generally target seemingly relevant aspects of a model's behavior, it is imperative to measure their effectiveness on concrete use cases in order to demonstrate their utility as practical diagnostics.

These two principles motivated us to first illustrate our diagnostic vision via an incomplete taxonomy that synthesizes foundational works on IML methods and evaluation. The taxonomy (shown at an abstract level in the left side of figure 2) serves as not only a template for building an explicit mapping between potential IML diagnostics and specific use cases, but also a tool to unify studies of IML's usefulness in real-world settings. Further, the incompleteness of the current taxonomy emphasizes the need for researchers and consumers to work together to expand the coverage of the use case organization (i.e., in the "Use Case Goals"), and to establish connections between methods and use cases by following the proposed workflow below.

(1) Problem definition, where researchers work with consumers to define a well-specified target use case.
(2) Method selection, where they 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

2

FIGURE 2: **ABSTRACTED VERSIONS OF OUR TAXONOMY**



consumers to test whether selected methods can meet target use cases.

Then, the latter part of this article includes an extensive discussion about best practices for this IML workflow to flesh out the taxonomy and deliver rigorously tested diagnostics to consumers. Ultimately, there could be an increasingly complete taxonomy that allows consumers (C) to find suitable IML methods for their use cases and helps researchers (R) to ground their technical work in real applications (as seen on the right side of figure 2). For instance, Table 1 highlights concrete examples

TABLE 1: **EXAMPLE USE CASES**

COMPUTER VISION: CLASSIFIER TO DETECT OBJECTS IN IMAGES	
Use Case:	Debug the model by identifying if it uses positive spurious correlations (i.e., relies on object Y to detect object X).
Diagnostic Insight:	When features (i.e., spurious objects) are present or missing, how does this affect a specific prediction?
BANK LENDING: CLASSIFIER TO GRANT/DENY LOANS TO CLIENTS	
Use Case:	Recommend actionable recourse for an individual to get a loan after they have been previously denied.
Diagnostic Insight:	What (low-cost) changes can an individual make to achieve a desired outcome?
COMPUTATIONAL BIOLOGY: CLUSTERING TO ANALYZE SINGLE-CELL RNA SEQUENCES	
Use Case:	Verify whether differences between clusters corroborate known scientific knowledge (e.g., different cell types).
Diagnostic Insight:	What feature changes (i.e., to gene expression) can be made to a group of points to achieve a desired outcome?

of how three different potential diagnostics, each corresponding to different types of IML methods (local feature attribution, local counterfactual, and global counterfactual, respectively), may provide useful insights for three use cases. In particular, the computer vision use case from Table 1 is expanded upon as a running example.

BACKGROUND

An increasingly diverse set of methods has been recently proposed and broadly classified as part of IML. Multiple concerns have been expressed, however, 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

Zachary C. Lipton provided an early critique, highlighting that the stated motivations of IML were both highly variable and potentially discordant with proposed methods.[17] Maya Krishnan added to these arguments from a philosophical angle, positing that interpretability as a unifying concept is both unclear and of questionable usefulness.[15] Instead, 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 claimed practical usefulness. Some have demonstrated a lack of stability/robustness in popular approaches.[1,2,16] Others, meanwhile, discuss how common IML methods can fail to help humans in the real world, both through pointing out hidden assumptions and dangers,[6,21] as well as conducting case studies with users.[5,14]

More recently, many review papers[3,10,19,20] have attempted to clean up and organize aspects of IML but largely do not address these issues head on. In contrast, the reframing of IML methods as diagnostic tools proposed here follows naturally from these concerns. Notably, this article embraces the seeming shortcomings of IML methods as providing merely "facts"[15] or "summary statistics"[21] about a model, and instead focuses on the practical questions of when and how these methods can be useful.

A DIAGNOSTIC VISION FOR IML

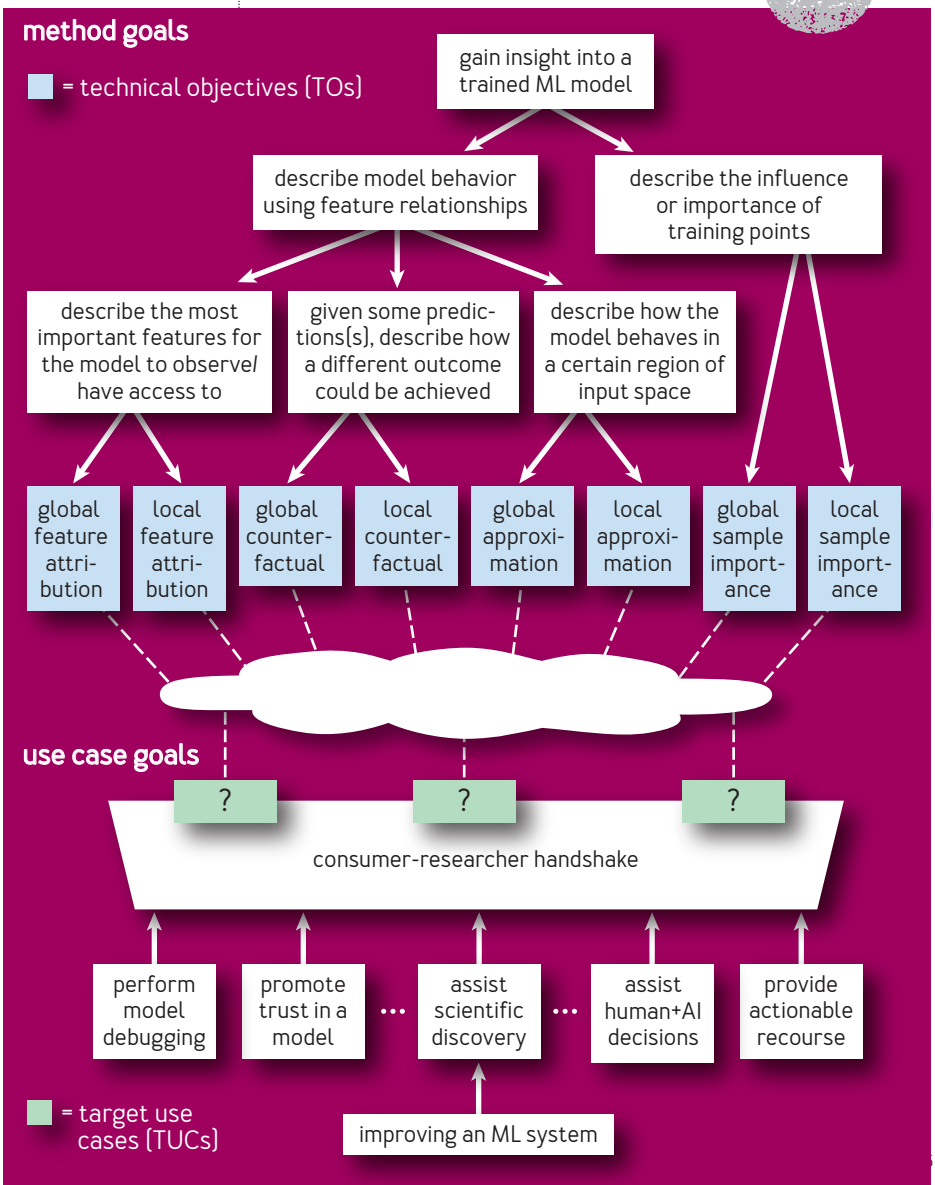
In our vision, a diagnostic is a tool that provides some

actionable insight about a model. As an analogy, consider the suite of diagnostic tools at a doctor's disposal that similarly provides various insights 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 rhythm. Importantly, neither tool enables the doctor to broadly "understand" a person's health, but each can be useful if applied properly to a well-scoped problem. A similarly rigorous approach to establishing connections between IML methods and well-defined use cases is imperative for the IML community.

To begin such a pursuit, let's identify and reconcile the many method goals and use case goals that you might currently encounter. Based on contemporary practices and discourse, let's consider a taxonomy that organizes separate hierarchies for the method goals at the top end and use case goals at the bottom end (as illustrated in figure 3). While the diagnostic vision for the field ideally involves a clearly defined set of use cases and a robust set of connections between these two sides, a cloud is used to illustrate the current overall lack of well-established diagnostics. Moving forward, the goal for researchers and consumers is to conduct principled studies focused on filling in both gaps. First, they should work to refine the current organization of use cases, consisting of an incomplete list of commonly discussed broad goals, by defining more well-specified target use cases (shown in green) via the consumer-researcher handshake. Second, they should aim to establish explicit connections between these targets and technical objectives (shown in blue).

3

FIGURE 3: **TAXONOMY FOR THE DIAGNOSTIC VISION**



Method goals

Each IML method provides a specific type of insight into a given model. The forms of these insights help provide a hierarchical organization that divides the set of existing IML methods into eight method clusters. In the diagnostic vision, each method cluster is thought of broadly as a class of diagnostics that addresses a TO (technical objective). Later, each TO is described in a way that allows individual method goals to be specified.

Hierarchical organization

The top end of the taxonomy aims to differentiate between the various perspectives that explanations provide, based on three factors commonly discussed in existing literature:[3,9,11]

- *Explanation representation.* Model explanations are typically given in terms of either *feature relationships* between inputs and outputs or *training examples*.
- *Types 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: (1) *feature attribution*; (2) *counterfactual*; and (3) *approximation*. Note that because the IML community focuses less on generating example-based explanations, we consider one main grouping along that branch: *sample importance explanations*.
- *Explanation scale.* Explanations vary in terms of the scale of the desired insights, with their scope ranging from *local* (i.e., for an individual instance) to *global* (i.e., for a well-defined region of the input space).

At the leaf nodes are the 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 eight TOs/method clusters that capture a large portion of the goals of existing IML methods. There are a couple of important nuances regarding the characterization of TOs.

➡ *First,* although TOs and method clusters are one-to-one in the 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 a method to, in an ad hoc fashion, be adapted to address a different TO.

➡ *Second,* each TO should be thought of as defining a class of related goals. Indeed, for a given TO, we hypothesize what some of the key *technical detail(s)* are that must be considered toward fully parametrizing meaningfully different instantiations of the same broader goal. These important technical details, taken together with the TO, allow you to define individual *proxy metrics* that reflect the desired properties of your explanations. Proxy metrics can then serve as tractable objective functions for individual methods to optimize for, as well as measures of how well any method addresses a particular instantiation of the TO.

Technical objectives

The following is an overview of the TOs (and their technical details) that correspond to various method clusters. Because of the overlaps in content, local and global versions of the same general method type/objective are grouped together. (For more details and examples of specific methods for each, see our longer-form paper,

"Interpretable Machine Learning: Moving from Mythos to Diagnostics," by Chen, et al.[8]).

➡ *Feature attribution explanations* address how the model's prediction(s) are affected when features are present (or missing), 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) changes 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 to 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 details of both the definition of the *region* and the simple function's *model family*. 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 which training points most influence a model's prediction for either an individual point or the model as a whole. Technical details differ from method to method, so 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 for data distribution shifts, and recovering high accuracy with the samples considered important.

How do by-design methods fit in?

While they do not have a corresponding method cluster in this taxonomy, it is important to discuss another family of IML methods that propose models that are themselves *interpretable by design.*[21] The differentiating property of these models from the post-hoc methods referenced in the above section is that the TO(s) of these approaches is intrinsically tied to the model family itself; hence, the models are interpretable by design only in that they satisfy said TO(s). That said, by-design methods also fit into this framework and should be viewed as a different way to answer the same TOs in the taxonomy. When by-design methods are proposed or used, they should clearly specify which TOs they intend to address.

Use case goals

Much of the current discourse on IML use cases surrounds

differentiating fairly broad goals, such as debugging models, gaining trust of various stakeholders, and providing actionable recourse to users (figure 3). While this level of categorization represents a good start, it is of limited utility because it treats 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, nor solely, solvable by IML itself. Rather, IML is but one potential set of tools that must be proven to be useful. That is, to show that an IML method is an effective diagnostic, specific use cases must be identified and demonstrated.[15]

Secondly, each broad goal really includes multiple separate technical problems, crossed with many possible practical settings and constraints. It is unlikely that a given IML method will be equally useful across the board for all of these subproblems and domains.

Thus, claims of practical usefulness should ideally be specified down to the level of an adequately defined TUC (target use case). Like TOs on the methods side, TUCs correspond to learning a specific relevant characteristic about the underlying model (e.g., a certain property or notion of model behavior). Unlike a TO, however, they represent real-world problems that, while they can be evaluated, often might not be amenable to direct optimization.

For example, you can set up evaluations to determine whether an IML method is useful for identifying a particular kind of bug in the model (e.g., positive spurious correlations), but it is not so obvious how to optimize an IML method that will succeed on those evaluations.

A WORKFLOW FOR ESTABLISHING DIAGNOSTICS

Let's turn now 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, an ideal workflow is defined for consumer-researcher teams to conduct future studies about IML methods. It describes 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 both to teams who wish to study existing IML methods and to those proposing new ones.

A running example helps contextualize this discussion, building on the computer vision model debugging example from table 1. Model debugging is not only a common consumer use case,[7,13] but also a well-grounded one. It is a natural starting point because of the versatile nature of its assumed consumer, data scientists, who typically have both substantial ML knowledge and domain expertise, minimizing the communication gap between the data scientist and the IML researcher.

Step 1: Problem Definition

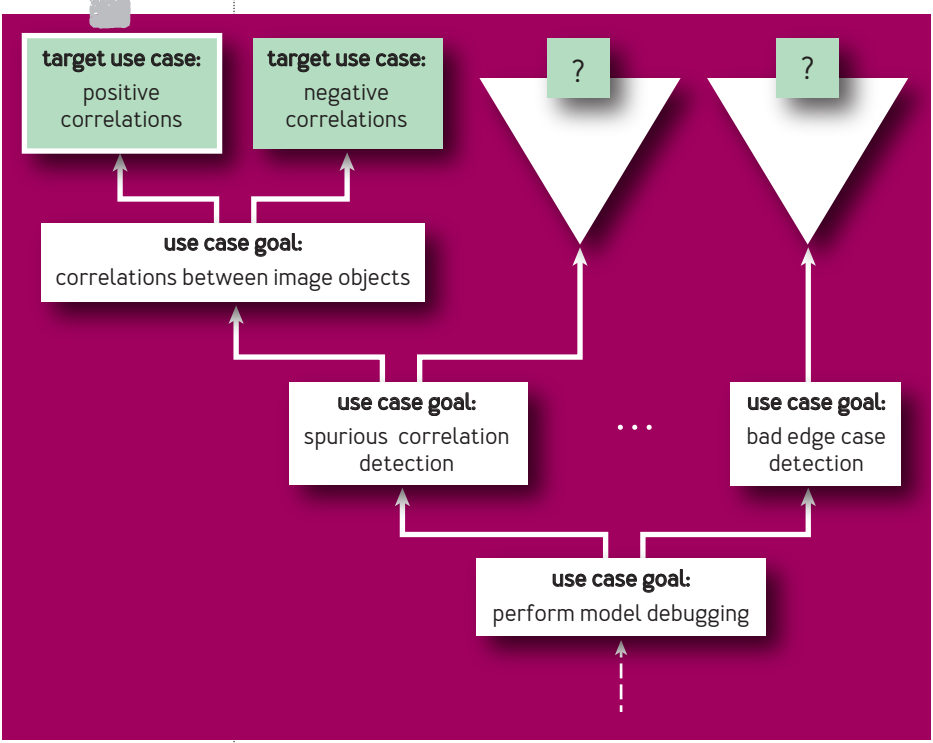
An important first step for any principled study is to define a well-specified TUC. This process is called the *consumer-researcher handshake* (figure 3), where researchers work with consumers to progressively refine the latter's real-world problems into relevant TUCs. In this process, some helpful pieces of information include: the data available, the ML pipeline used, and the domain knowledge required to perform evaluations. Ultimately, a more fleshed-out

taxonomy will help researchers have more concrete use cases at hand to motivate their method development, and consumers will have more realistic guidance on what IML can and cannot do for them.

4

Running example: Consider a data scientist who wants to debug her image-based object detection model. She hopes to leverage the expertise of an IML researcher, but as shown in a hypothetical version of the use cases part of

FIGURE 4: **CONSUMER-RESEARCHER HANDSHAKE FOR OUR RUNNING EXAMPLE**



the taxonomy (figure 4), the umbrella of model debugging includes several subproblems, such as detecting spurious correlations and identifying bad edge-case behavior. Thus, the team of researcher and data scientist needs to identify a TUC that is more specific than "perform model debugging" by identifying exactly what notion of "bug" the IML method should detect. Through the consumer-researcher handshake, it arises that the data scientist is concerned that the model might not be making correct decisions based on the actual target objects, but rather is relying on correlated objects that 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 relevant branches of the taxonomy. 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 (marked by the white border in figure 4). 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 whether the output depends on an object if the co-occurrences are rare in the first place).

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. found model confidence to be a competitive baseline against dedicated interpretability approaches for AI/human decision-making teams.[5]

If non-IML diagnostics are unsuccessful, the taxonomy can be used in two ways to select methods. First, researchers and consumers can, as a default, traverse the methods part of the taxonomy to identify 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, it 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 the section about method goals, 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 may be 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, as illustrated via the following two examples (and in more depth in Chen, et al.[8]).

First, you might try to use "feature importance weights," via SHAP (Shapley additive explanations),[18] 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. This is not necessarily guaranteed by SHAP, however, 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, you can think of saliency maps via vanilla gradients[23] 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 identifying which parts of the image are affecting the prediction the most. This adaptation, however, 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.

Running example: In this scenario, suppose that there have been no previously established results for detecting positive spurious correlations. The team follows the methods part of the taxonomy to generate hypotheses for which types of local explanations best suit 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 believe 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. To do so, they are assuming that the most changed features are also the most important for 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, the feature attribution method that the team selects for comparison is Grad-CAM (gradient-weighted class activation mapping),[22] which also produces a saliency map.

Step 3: Method Evaluation

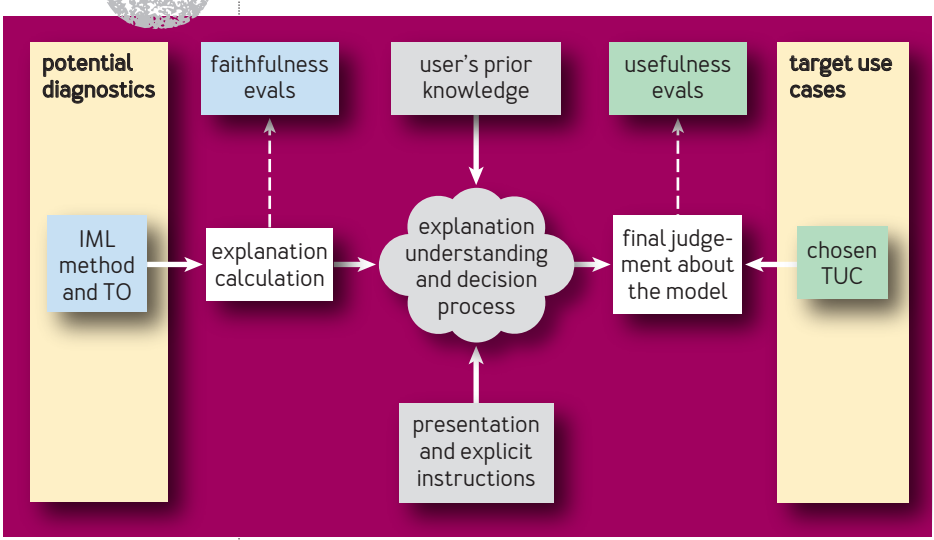
Once appropriate method(s) have been chosen, the last step is to evaluate them. Evaluation is the crucial step

of testing whether proposed methods can actually help address the specified TUC. Yet despite its importance, this step is often carried out in manners incongruent with the properties it claims 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, they target fundamentally different claims.

Considering these evaluations within the overall pipeline of an IML application (as shown in figure 5) addresses this kind of mistake. It first highlights differences in goals of these evaluations by connecting back to the taxonomy presented in this article; faithfulness

5

FIGURE 5: **EVALUATION PROCEDURES WITHIN IML PIPELINES**



corresponds to meeting objectives of a specific TO in the methods part, and usefulness corresponds to meeting the TUC in the use cases part. Then, it also lays out the various moving components that affect each type, with gray boxes denoting components that require more careful study. This serves to ground how each may be carried out, which we discuss in greater detail next.

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 were 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 and Kim,[9] 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, you should expect that a method would perform well at least on a proxy for its selected TO, and, naturally, those methods that do not directly target this specific proxy will likely not perform as well. An explanation's performance can also be faultily compared with another's as a result of unfair or biased settings of technical details. As an example, although GAMs (generalized additive models)[12] 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 setups before performing expensive usefulness evaluations.

Usefulness evaluations, in contrast to faithfulness, measure a user's success in applying explanations to the specified TUC. Since they are ultimately an evaluation of what a user does with an explanation, usefulness depends crucially on factors, such as the user's prior knowledge—for example, their domain and ML/IML experience. Again, using terminology from Doshi-Velez and Kim,[9] 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, as part of conducting usefulness studies, you would need to consider how users might act differently depending on the presentation of the explanation and explicit instructions that are provided.

As highlighted by the cloud in figure 5, exactly how 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 consumers deploy the method.

Motivated by these challenges, researchers might want to also consider another type of usefulness evaluation: *simulation evaluation*. This is an algorithmic evaluation on a simulated version of the real task where success and failure are 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 users' actual decision-making process?

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 separate local feature attribution faithfulness evaluations for both methods using the respective 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 absence of a super-pixel. Good performance on any proxy metric, however, does not conclusively imply*

good performance on the actual TUC, so the team turns to usefulness evaluation.

The team first conducts a simulation evaluation, where datasets are created that contain either (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 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 spurious correlations. They score data scientists based on whether they are able to use each explanation generated by the counterfactual versus Grad-CAM to identify models that 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.

CONCLUSION

Assuming a diagnostic vision for IML, the taxonomy presented here is a way to clarify and begin bridging the gap between methods and use cases. Further, this article discusses best practices for how the taxonomy can be used and refined over time by researchers and consumers to establish which methods are useful for which use cases. As the taxonomy is fleshed out via more studies by consumer-researcher teams, our vision is that it will be increasingly

useful for both parties individually (figure 2, right). Overall, the goal is to promote better practices in discovering, testing, and applying new and existing IML methods moving forward.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B. 2018. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 9525-9536; https://dl.acm.org/doi/10.5555/3327546.3327621.
2. Alvarez-Melis, D., Jaakkola, T. 2018. On the robustness of interpretability methods. arXiv:1806.08049; https://arxiv.org/abs/1806.08049.
3. 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:1909.03012; https://arxiv.org/pdf/1909.03012.pdf.
4. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., Samek, W. 2015. On pixelwise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE* 10(7): e0130140; https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140.
5. Bansal, G., Wu, T., Zhu, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M.T., Weld, D S. 2020. Does the whole exceed its parts? The effect of AI explanations on complementary team performance. arXiv:2006.14779; https://arxiv.org/pdf/2006.14779.pdf.

6. Barocas, S., Selbst, A. D., Raghavan, M. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 80-89; https://dl.acm.org/doi/abs/10.1145/3351095.3372830.
7. Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J.M.F., Eckersley, P. 2020. Explainable machine learning in deployment. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 648-657; https://dl.acm.org/doi/abs/10.1145/3351095.3375624.
8. Chen, V., Li, J., Kim, J.S., Plumb, G., Talwalkar, A. 2021. Interpretable Machine Learning: Moving from Mythos to Diagnostics. arXiv:2103.06254.
9. Doshi-Velez, F., Kim, B. 2017. Towards a rigorous science of interpretable machine learning. arXiv:1702.08608; https://arxiv.org/pdf/1702.08608.pdf.
10. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L. 2018. Explaining explanations: an overview of interpretability of machine learning. In *Fifth IEEE International Conference on Data Science and Advanced Analytics*; https://ieeexplore.ieee.org/document/8631448.
11. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys* 51(5), 1–42; https://dl.acm.org/doi/10.1145/3236009.
12. Hastie, T.J., Tibshirani, R.J. 1990. Generalized Additive Models. *Monographs on Statistics and Applied Probability*, 43. Chapman and Hall/CRC.

13. Hong, S.R., Hullman, J., Bertini, E. 2020. Human factors in model interpretability: industry practices, challenges, and needs. In *Proceedings of the ACM on Human-Computer Interaction* 4(CSCW1), 1–26; https://dl.acm.org/doi/10.1145/3392878.
14. Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., Wortman Vaughan, J. 2020. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1-14; https://dl.acm.org/doi/abs/10.1145/3313831.3376219.
15. Krishnan, M. 2020. Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology* 33, 487–502; https://link.springer.com/article/10.1007/s13347-019-00372-9.
16. Laugel, T., Lesot, M.-J., Marsala, C., Detyniecki, M. 2019. Issues with post-hoc counterfactual explanations: a discussion. arXiv:1906.04774; https://arxiv.org/pdf/1906.04774.pdf.
17. Lipton, Z.C. 2018. The mythos of model interpretability. *ACM Queue* 16(3), 31–57; https://queue.acm.org/detail.cfm?id=3241340.
18. Lundberg, S.M., Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* 30; https://papers.nips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.
19. Mohseni, S., Zarei, N., Ragan, E. 2020. A multidisciplinary survey and framework for design and evaluation

of explainable AI systems. *ACM Transactions on Interactive Intelligence Systems* 1(1); https://arxiv.org/pdf/1811.11839.pdf.

20. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B. 2019. Interpretable machine learning: definitions, methods, and applications. In *Proceedings of the National Academy of Sciences* 116(44), 22071-22080; https://www.pnas.org/content/116/44/22071.
21. Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 206–215; https://www.nature.com/articles/s42256-019-0048-x.
22. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. 2017. Grad-CAM: visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, 618-626; https://ieeexplore.ieee.org/document/8237336.
23. Simonyan, K., Vedaldi, A., Zisserman, A. 2013. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034; https://arxiv.org/abs/1312.6034.
24. Sundararajan, M., Taly, A., Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*; http://proceedings.mlr.press/v70/sundararajan17a.html.

Valerie Chen *is a Ph.D. student in the Machine Learning Department at Carnegie Mellon University. Her research is on interpretability techniques to aid human decision-making and,*

more broadly, as a way to study the societal impact of machine learning.

Jeffrey Li *is a Computer Science Ph.D. student at the University of Washington. He is interested in topics that address challenges limiting the deployment of machine learning in practice, including learning from weak sources of supervision and interpretable machine learning.*

Joon Sik Kim *is a Ph.D. student in the Machine Learning Department at Carnegie Mellon University. He is interested in methods that can facilitate the understanding of complex machine learning models and their implications for model interpretability and fairness.*

Gregory Plumb *is a Ph.D. student in the Machine Learning Department at Carnegie Mellon University. His research focuses on explainable machine learning, with an emphasis on developing novel techniques for model debugging.*

Ameet Talwalkar *is an assistant professor in the Machine Learning Department at Carnegie Mellon University. His current work is motivated by the goal of democratizing machine learning, with a focus on topics related to automation, interpretability, and distributed learning.*

CONTENTS